

ELEKTRONIKA PRAKTYCZNA + FTP

● Międzynarodowy magazyn elektroników konstruktorów ● Październik ● 10/2019 ●



inspirujące, użyteczne projekty

Bramka SMS na bazie Raspberry Pi z dołączonym modułem GSM • Rower elektryczny • Monitor połączenia z Internetem zbudowany na Raspberry Pi • Wzmacniacz lampowy na lampach EL34 • microStation – mała stacja pogodowa • Radiowy przedłużacz DS18B20 • VL53L1X – laserowy sensor. Czujnik odległości działający w technologii lidar • Sterownik wejść mostka H • Przetwornik DAC z końcówką mocy dla Raspberry Pi • Przedwzmacniacz na lampie Nutube 6P1 • Konwerter USB-UART z ekstenderem • Sterownik wentylatora wyciągu łazienkowego (kuchennego)

podzespoły, sprzęt, aplikacje

Mikrokontrolery specjalizowane od Nuvoton • Faulhaber rozszerza gamę silników krokowych. Ogromne przyspieszenie przy wykorzystaniu magnesu dyskowego • SOLIDWORKS PCB – unikatowy mariaż narzędzi MCAD i EDA • BeagleBone AI

tutoriale

Bezpieczeństwo funkcjonalne: normy, analizy i sposoby jego uzyskiwania w platformach mikrokontrolerowych • Symulacja i pomiar czyli LTspice i Analog Discovery 2 w rękach konstruktora. Filtr antyaliasingowy

kursy

ISIXRTOS v3 mini system operacyjny dla mikrokontrolerów rodziny M0/M3/M4/M7. Obsługa wyświetlacza graficznego • RISC-V – budujemy własny mikrokontroler • AudioDSP zestaw z procesorem Sigma DSP ADAU1701. Konfiguracja DSP procesorem zewnętrznym • Systemy dla Internetu Rzeczy. Praca z jednoczesną komunikacją radiową BLE 5 oraz IEEE 802.15.4

MIKROKONTROLERY DO ZADAŃ SPECJALNYCH

TEMAT NUMERU

16 zł (w tym 8% VAT) • PRICE: 8 EUR • Nakład 27000 egz.

ISSN 1230-3526 Indeks 357677



9 771230 352191



Ogranicz czas i koszt przestoju



RS PRO to globalna marka produktów dla sektora przemysłowego, której zaufali inżynierowie z całego świata.



Wysoka jakość



Konkurencyjna cena



Szeroki wybór



Znak jakości RS PRO



Więcej na pl.rs-online.com/rspro

Demo Day

360 DEGREES OF LOW-POWER

Sopot, 15.10.2019



Zaproszenie

do udziału w Demo Day

„360 Degrees of Low-Power”

który odbędzie się

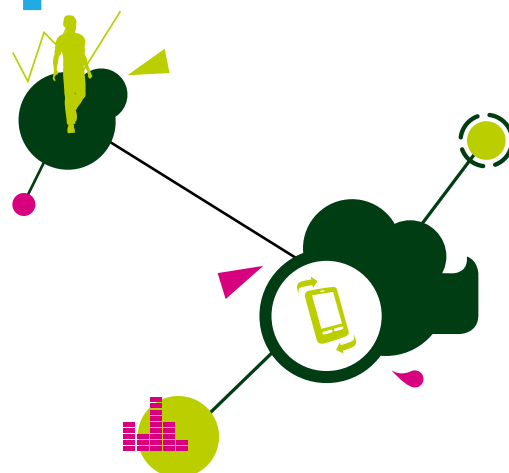
15.10.2019 w Sopocie

**Prezentacje poświęcimy 12 innowacyjnym
tematom, w tym:**

- * **budowa urządzeń low-power,**
- * **rozpoznawanie obrazów z AI,**
- * **MEMS Machine Learning,**
- * **sensory Time-of-Flight,**
- * **podzespoły dla USB-C,**
- * **ochrona własności intelektualnej z NFC,**
- * **bezpieczne aplikacje z STM32,**
- * **systemy trackingowe LoRa,**
- * **Bluetooth 5.0.**

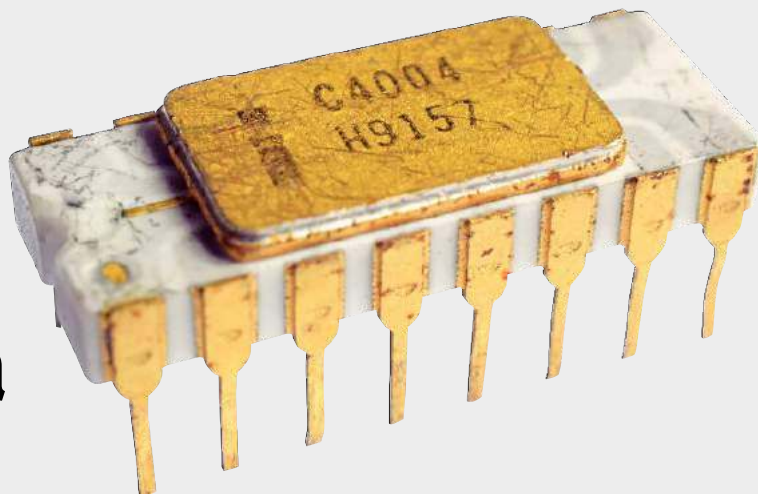
Zarejestruj się na stronie

STPOLAND.com



life.augmented

Krótką historia na jubileusz 50-lecia mikroprocesora



Nie do wiary, to już pół wieku?! Wprawdzie pierwszy mikroprocesor, słynny 4004, Intel wprowadził na rynek wiosną 1971 roku, to jednak koncepcja tego układu, wbrew amerykańsko-centrycznej wizji świata, zrodziła się wcześniej w Japonii. W czerwcu 1969 r. Dr Masatoshi Shima z firmy Busicom zawiązał w Santa Clara i przedstawił w firmie Intel zamówienie na wyprodukowanie zestawu układów logicznych do kalkulatora planowanego do produkcji w Busicom. Schematy, które przywiózł ze sobą Japończyk, były rewolucyjne – zrywały z programowaniem wszytym w hardware, proponując architekturę komputerową z jednostką arytmetyczną o dostępie swobodnym, realizującą zadania logiczne zarządzane przez software. Koncepcja architektury logicznej, rozdzielającej dane od adresów i zawierającej blok CPU (Central Processor Unit), blok pamięci ROM, blok pamięci RAM i blok interfejsów peryferyjnych, zakładała de facto budowę małego komputera na układach scalonych. Właśnie 4-bitową jednostką centralną z tego projektu stał się pierwszy na świecie mikroprocesor oznaczony jako 4004. Japończycy wiedzieli, co chcą osiągnąć, ale nie mieli do tego technologii. Mieli ją Amerykanie w firmie Intel, która wówczas istniała dopiero od niespełna roku, czyli według obecnych pojęć była start-upem. Skąd w start-upie najnowocześniejsza na świecie technologia? Otóż przywędrowała w głowach założycieli Intela, a wcześniej pracowników firmy Fairchild – Gordona Moore’a i Roberta Noyce’a, za którymi przyszli do Intela czołowi technolodzy Fairchilda – A. Grove i L.L. Vadász. Była to technologia SGT (Silicon Gate Technology), opracowana w 1967 roku w firmie Fairchild przez Federico Faggin, który również przeszedł do Intela w roku 1970.

Intel zaczął się od zdrady Fairchilda przez jego najlepszych technologów, której przyczyną bezpośrednią były niezaspokojone ambicje Roberta Noyce’a – chciał zostać prezesem Fairchilda, a właściciele firmy mieli inne plany. Nie mogę się powstrzymać, by nie napisać w tym miejscu paru zdań o historii kilku zdrad, które stały



Gordon Moore



Robert Noyce

się motorem postępu technologicznego uwieńczonego powstaniem mikroprocesora.

Zaczął się od „zdradzieckiej ósemki”, tj. przejścia w 1957 roku ośmiu najzdolniejszych uczniów odkrywcy tranzystora Williama Shockleya z jego firmy Shockley Semiconductor Laboratory do firmy Fairchild. Tej ósemce przewodził Robert Noyce, który 11 lat później wraz z Gordonem Moore’em odszedł „na swoje”, zakładając firmę Intel. Obie „zdrady” miały swoje przyczyny. W 1957 r. R. Noyce z siedmioma kolegami mieli dość apodyktycznego szefa (W. Shockley był geniuszem – laureatem Nagrody Nobla za odkrycie tranzystora – ale człowiekiem bardzo trudnym we współżyciu) i chcieli rozwiązać prace nad technologią krzemową, z czym nie zgadzał się W. Shockley. Przejście „zdradzieckiej ósemki” do Fairchilda uczyniło z tej firmy, produkującej wcześniej przyrządy optyczne, światową potęgę w technologii półprzewodnikowej. Na początku lat sześćdziesiątych dwie firmy przewodziły światowej mikroelektronice: Fairchild w segmencie układów scalonych analogowych i Texas Instruments w segmencie układów scalonych cyfrowych, przy czym w innowacjach technologicznych zdecydowanym liderem światowym był Fairchild. Wynalazkiem fundamentalnym, bez którego nie byłaby możliwa produkcja układów scalonych, okazała się technologia planarna, użyta przez R. Noyce’a w roku 1959 do wyprodukowania pierwszego układu scalonego. Układy scalone produkowane w pierwszej połowie lat sześćdziesiątych bazowały na tranzystorach bipolarnych, chociaż już w 1963 r. były dobrze znane i opisane właściwości tranzystora MOS, w szczególności inwertera CMOS i doskonale zdawano sobie sprawę z kolosalnej przewagi tych tranzystorów nad bipolarnymi w zastosowaniu do układów scalonych. Jednak rozwój technologii układów scalonych MOS napotkał na przykrą przeszkodę w postaci ich niestabilności czasowej. W rozwiązaniu tego problemu



William Shockley



Bruce E. Deal

największe zasługi ma dr Bruce Deal pracujący w firmie Fairchild. Pierwsze układy scalone w technologii PMOS z bramką aluminiową nie dość, że mogły po jakimś czasie przestać pracować (niestabilność), to wymagały zasilania napięciem 24 V i były dość powolne. Dopiero opracowanie technologii NMOS z tak zwaną samocentrującą bramką polikrzemową (SGT) zwiększyło szybkość działania układów MOS około 5-krotnie i pozwoliło zmniejszyć napięcie zasilania do poziomu kompatybilnego ze standardem zasilania (+5 V) bardzo wówczas popularnych układów cyfrowych TTL.

Autorzy „drugiej zdrady” – R. Noyce i G. Moore – mieli gotową technologię do ekspansywnego rozwoju produkcji układów scalonych LSI (zawierających ponad 1000 tranzystorów) w ich własnej firmie Intel. Zaczęli od pamięci RAM 1 kbit, a wizyta dr. Shimy z zamówieniem na mikroprocesor otworzyła przed ich firmą fantastyczne perspektywy. Trzeba było tylko przeciągnąć z Fairchilda do Intela twórcę technologii NMOS Silicon-gate Federico Faggina, który okazał się również wybitnie uzdolnionym projektantem na poziomie elementarnych układów logicznych. Japoński pomysł wyartykułowany na poziomie schematów blokowych architektury został doprowadzony do postaci projektów układu 4004 na poziomach logicznym, elektrycznym i technologicznym. Dokonał tego Federico Faggin i w topologii ścieżek czipa tego układu wpisał swoje inicjały FF, uważając siebie, nie bez podstaw, za twórcę pierwszego mikroprocesora.

Można też mówić o „trzeciej zdradzie”, a może po prostu o amerykańskiej mobilności, gdyż po kilku latach F. Faggin odszedł z Intela i założył firmę Zilog produkującą kultowy mikroprocesor Z80, stosowany w legendarnych minikomputerach ZX Spectrum, Amstrad, Commodore 128 (jako drugi procesor).

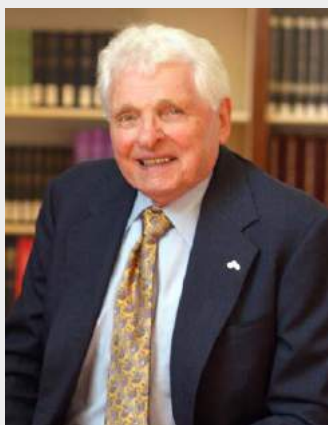
Bohater tej opowieści – mikroprocesor 4004 – nie odegrał wielkiej roli. Japończycy zastosowali go w swoim kalkulatorze i użyto go jeszcze w kilku innych aplikacjach, ale dopiero jego następca 8-bitowy Intel 8008 i kolejny 8080 na dobre otworzyły nieograniczone możliwości aplikacji i rozwoju elektroniki.

Układ 4004 zawierał 2300 tranzystorów i miał ścieżki o szerokości 10 μm . Po 50 latach w technologii CMOS wymiar charakterystyczny jest 1000-krotnie mniejszy, czyli wynosi 10 nm, a gęstość upakowania jest milion razy większa, czyli układy scalone zawierają miliardy tranzystorów. A jednak podstawowe idee nie zmieniły się – nadal rzeźbimy ścieżki w krzemie i budujemy wielkie systemy z klocków znanych ponad 50 lat temu, tj. z inwerterów CMOS.

Jak już tyle napisałem, to może jeszcze dodam wątek polski i mój osobisty. W 1973 roku na IV Konferencji Mikroelektroniki w Toruniu gościliśmy czterech przedstawicieli firmy Fairchild, wśród nich był wspomniany już wcześniej dr Bruce Deal i dr Harry Sello (uczeń i ulubieniec W. Shockleya). Nikt tak pięknie nie potrafił opowiadać o pierwszym mikroprocesorze jak Harry Sello. Nie była to wizyta przypadkowa. Od wielu miesięcy



Federico Faggin



Harry Sello

uzgadniane były 2 licencje CEMI (Centrum Mikroelektroniki na Śluzewcu w Warszawie, gdzie obecnie jest galeria handlowa) z Fairchildem, w tym jedna na układy scalone MOS LSI. Tak się złożyło, że organizatorzy konferencji powierzyli mi eskortowanie dwóch Amerykanów w podróży pociągiem z Torunia do Warszawy. Jednym z nich był dr B. Deal – mój guru, gdyż mój doktorat i habilitacja, którą wówczas robiłem, dotyczyły niestabilności struktur MOS, tj. tematyki, w której dr Deal był światowym autorytetem. Rozmowa z nim była dla mnie wielką gratką. Okazało się szybko, że ja znam każdy jego artykuł, a on nie wie nic o moich pracach. Żartuję, rzecz jasna, choć przecież miałem publikacje w pismach światowych, m.in. w amerykańskim „Journal of Electrochemical Society”, a w artykułach Philipa powoływano się na moją metodę diagnostyczną. W połowie drogi do Warszawy, gdzieś w okolicach Kutna, stało się jednak jasne, że dr Deal nie tylko nie słyszał o moich pracach, ale w ogóle nie zna żadnych prac poza swoimi. Wtedy zrozumiałem, że poza Silicon Valley jest pustynia, którą nie warto się interesować.

A z dr. Harrym Sello, który w Toruniu miał porywający wykład na temat mikroprocesora, choć ciągle reprezentował firmę Fairchild, spotkałem się po latach. Około 1990 r. dr H. Sello wizytował Instytut Technologii Materiałów Elektronicznych, którym wówczas kierowałem. Poza rozmową biznesową oddawaliśmy się nostalgicznym wspomnieniom konferencji w Toruniu i oplakiwaliśmy okrutny los firmy Fairchild, która z potęgi światowej stoczyła się do niebytu. My friend Harry (z Amerykanami można się zaprzyjaźnić błyskawicznie) wspominał m.in. swoją misję we Włoszech. Otóż wśród wielu nietrafionych działań zarządu Fairchilda była nieudana próba zbudowania joint venture z Włochami. Na paroletnią „zsyłkę” do Włoch wysłano Harry'ego, o czym chętnie wspominał, opowiadając różne anegdoty. Na przykład taką: „zwolniłem z pracy jegomościa, który uprawiał seks w miejscu pracy. Następnego dnia złożyła mi wizytę jego żona i zapytała mnie najgorszymi wyzwiskami. A ja wszystko zrozumiałem. Byłem zachwycony, że po 2 latach pobytu tak dobrze opanowałem język włoski”. Taki był Harry. Zmarł 2 lata temu i już nikt na świecie nie opowie Wam tak jak on o pierwszych mikroprocesorach.

Niestaw Kurczak

PS A z licencji Fairchilda w 1973 roku nic nie wyszło. Gdy w Pentagonie dowiedzieli się, że kraj komunistyczny dostanie najnowszą technologię (2–3 lata po USA), to złapali się za głowy i prezes Fairchilda Lester Hogan zapłacił za to stołkiem prezesowskim, a gotowy już kontrakt został unieważniony. Młodym Czytelnikom wyjaśniam, że w czasach zimnej wojny w USA ustalano normy sygnowane przez COCOM (Coordinating Committee for Multilateral Export Control), które precyzyjnie określały, jakie technologie są dostępne dla krajów komunistycznych, tak aby utrzymywać stałe opóźnienie technologiczne tych krajów wobec USA od 10 do 15 lat. Normy te obowiązywały cały Zachód i nie były to żarty. Osobiście znałem biznesmena z zachodniej Europy, który z nami niejawnie współpracował i w USA czekał na niego wyrok 150 lat. Dodam dla jeszcze większej jasności, że nasz Wielki Brat (ZSRR) też stosował wobec nas embargo technologiczne, choć czynił to obłudnie pod płaszczykiem koordynacji prac rozwojowych z planami militarnymi w ramach Układu Warszawskiego. Może kiedyś napiszę w jakich warunkach moje pokolenie Polaków podejmowało wysiłki, by nadążyć w technologii, łamiąc embargo amerykańskie w tajemniczy przed ZSRR. Kluczową rolę w tych działaniach motywowanych patriotyzmem odgrywały służby, którym szczególną satysfakcję sprawiało wyprowadzanie w pole Radzianów, jak pogardliwie nazywali „bratnie” służby ZSRR. Tym ludziom Najjaśniejsza odpłaciła w ostatnich latach odebraniem im emerytur. Ale to temat na inną okazję.

Nie przeocz

Nowe podzespoły.....	8
BeagleBone AI.....	14
Koktajl niusów.....	15
Energetab 2019.....	18
Dodaj do obserwowanych.....	19

Projekty soft

Bramka SMS na bazie Raspberry Pi z dołączonym modułem GSM.....	22
Rower elektryczny (2).....	26
Monitor połączenia z Internetem zbudowany na Raspberry Pi.....	30

Projekty

Wzmacniacz lampowy na lampach EL34.....	34
microStation – mała stacja pogodowa.....	40
Radiowy przedłużacz DS18B20.....	47

Moduły w aplikacjach

VL53L1X – laserowy sensor. Czujnik odległości działający w technologii lidar.....	53
--	----

Miniprojekty

Sterownik wejść mostka H.....	57
Przetwornik DAC z końcówką mocy dla Raspberry Pi.....	59
Przedwzmacniacz na lampie Nutube 6P1.....	62
Konwerter USB-UART z ekstenderem.....	64

Projekty czytelników

Sterownik wentylatora wyciągu łazienkowego (kuchennego).....	66
---	----

Temat numeru: Mikrokontrolery

Mikrokontrolery do zadań specjalnych.....	72
Mikrokontrolery specjalizowane od Nuvoton.....	79
Bezpieczeństwo funkcjonalne: normy, analizy i sposoby jego uzyskiwania w platformach mikrokontrolerowych.....	80

Prezentacje

Faulhaber rozszerza gamę silników krokowych. Ogromne przyspieszenie przy wykorzystaniu magnesu dyskowego.....	85
SOLIDWORKS PCB – unikatowy mariaż narzędzi MCAD i EDA.....	86

Notatnik konstruktora

Symulacja i pomiar czyli LTspice i Analog Discovery 2 w rękach konstruktora (3). Filtr antyaliasingowy.....	88
--	----



Coraz trudniej jest nam wyobrazić sobie sytuację, w której nie mamy połączenia z Internetem. Gdy nie możemy połączyć się z jakąś stroną w Internecie zazwyczaj obwiniamy (nie bezzasadnie) zdalny serwer, ale co w momencie gdy to nasze łącze płata nam figla? Jeszcze większy problem mamy w przypadku zastosowania licznych urządzeń Internetu Rzeczy, czy systemów automatyki domowej, które uzależnione są od serwerów znajdujących się w Chmurze.

Wybór konstruktora:

Elementy indukcyjne i transformatory.....	98
INDEL.....	106
Noratel.....	107
AET.....	108
Telzam.....	109

Kursy

ISIXRTOS v3 mini system operacyjny dla mikrokontrolerów rodziny M0/M3/M4/M7 (7). Obsługa wyświetlacza graficznego.....	111
RISC-V – budujemy własny mikrokontroler (2).....	116
AudioDSP zestaw z procesorem Sigma DSP ADAU1701 (7). Konfiguracja DSP procesorem zewnętrznym.....	124
Systemy dla Internetu Rzeczy (30). Praca z jednoczesną komunikacją radiową BLE 5 oraz IEEE 802.15.4.....	130
Od wydawcy.....	4
Konkurs Microchip.....	29
Prenumerata.....	110
Kramik i rynek.....	137
Hity następnego numeru.....	138
Konkurs elektronika – by żyło się łatwiej.....	139



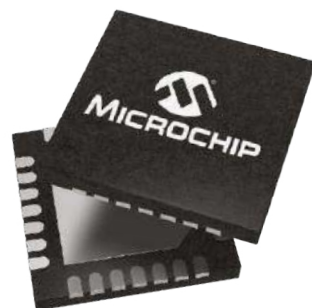
Wysokiej klasy układy analogowe od lidera rynku mikrokontrolerów?

Jeśli postrzegasz firmę Microchip tylko jako dostawcę MCU, mamy zamiar zmienić twoje myślenie!

Ocena pozycji firmy Microchip nie byłaby kompletna bez uwzględnienia naszych rozwiązań analogowych. Nasza historia jako wiodącego dostawcy rozwiązań zapewniającego wszechstronne wsparcie projektowe i szeroką gamę produktów nie obejmuje wyłącznie mikrokontrolerów.

Oferujemy również wydajne, łatwe w implementacji produkty analogowe, rozwiązania dla sygnałów mieszanych, zarządzania energią, a także układy interfejsowe I do zarządzania ciepłem. Razem szeroka oferta Microchip może być wykorzystywana w wielu aplikacjach o różnych wymaganiach w zakresie parametrów i wydajności.

Zapewniamy wydajność, elastyczność i pewność optymalnego wyboru rozwiązania dla swojego projektu, niezależnie od istniejących ograniczeń projektowych. Skorzystaj z naszego doświadczenia i kompletnych rozwiązań systemowych, aby zaoszczędzić czas i uprościć pracę nad projektem.



Zaplanuj własny sukces na
www.microchip.com/Real-Analog



nowe podzespoły

Z kilkuset nowości wybraliśmy te, których nie wolno przeoczyć. Bieżące nowości można śledzić na www.elektronikaB2B.pl

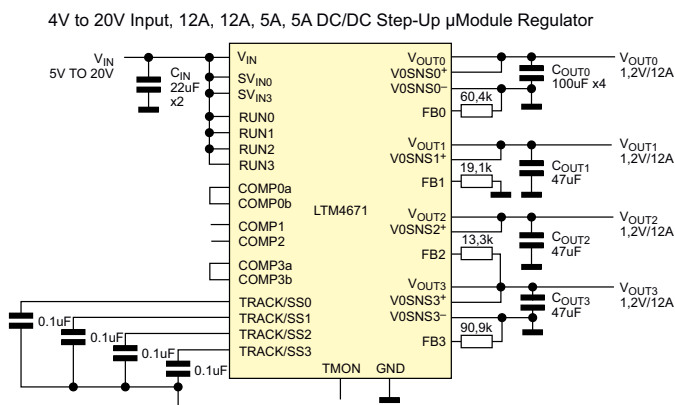


Miniaturowe odbiorniki IR o dużej czułości i skutecznym tłumieniu szumów

Do oferty firmy Vishay wchodzi dwie nowe serie zminiaturyzowanych odbiorników IR do pilotów zdalnego sterowania, charakteryzujących się, w stosunku do wcześniejszych modeli, zwiększoną czułością, skuteczniejszym tłumieniem szumów i dokładniejszą kontrolą szerokości impulsu, zapewniającą mniejszą bitową stopę błędów dla niemal wszystkich kodów. Są one zamykane w dwóch typach 4-wyprowadzeniowych obudów: Heimdall (seria TSOP95xxx) i Panhead (TSOP96xxx).

Nowe odbiorniki są mniej podatne od wcześniejszych wersji na interferencje optyczne ze strony m.in. lamp CFL oraz na emisję radiową w cz. z pobliskich anten Wi-Fi. Występują w wersjach o częstotliwości nośnej od 30 do 56 kHz i 5 różnych współczynnikach AGC dla krótkich i długich paczek danych. Pracują z napięciem zasilania z zakresu od 2,0 do 3,6 V przy typowym poborze prądu 0,37 mA.

<http://bit.ly/2kulw4r>



4-wyściowy regulator DC-DC step-down do zasilania układów FPGA, DSP i ASIC

LTM4671 to 4-wyściowy regulator DC-DC step-down zaprojektowany do zasilania układów FPGA, DSP i ASIC, zawierający wewnątrz miniaturowej obudowy BGA niezbędne kontrolery, tranzystory mocy i cewki. W najprostszej konfiguracji wymaga dołączenia jedynie kilku rezystorów i kondensatorów. Układ akceptuje napięcia wejściowe z zakresu od 3,1 do 20 V. Zawiera dwa wyjścia 12-ampierowe o zakresie programowania napięcia od 0,6 do 3,3 V i dwa wyjścia 5-ampierowe o zakresie 0,6...5,5 V. Programowanie napięcia wyjściowego w każdym z kanałów odbywa się poprzez dobór wartości pojedynczego zewnętrznego rezystora. Dokładność stabilizacji napięć wynosi $\pm 1,5\%$ w całym dopuszczalnym zakresie napięć wejściowych i prądów obciążenia. LTM4671 zawiera zabezpieczenie nadnapięciowe, nadprądowe i termiczne. Jest zamykany w obudowie BGA o wymiarach 16×9,5×4,72 mm.



<http://bit.ly/2kFCHzZ>

REKLAMA

BORNICO

Teraz większe MOŻLIWOŚCI

bornico.com.pl

- montaż kontraktowy elektroniki
- projektowanie urządzeń i systemów

Zakład Elektroniczny BORNICO

ul. Matęczyńska 25
26-600 Radom
tel. +48 48 365 58 22
bornico@bornico.com.pl



Renesas RX72M MCUs with EtherCAT Support for Industrial Equipment



Mikrokontrolery do układów napędowych z wbudowanym kontrolerem EtherCAT slave

Standard EtherCAT w przemysłowych sieciach Ethernet staje się coraz popularniejszy, a do jego implementacji wykorzystywane są specjalizowane mikrokontrolery i układy SoC wysokiej klasy. Renesas

Electronics wprowadza na rynek grupę mikrokontrolerów RX72M do zastosowań przemysłowych w urządzeniach wymagających funkcji sterowania i komunikacyjnych, jak roboty, sterowniki PLC, bramki dostępowe i moduły wejść/wyjść. Wyróżniają się one wbudowanym kontrolerem EtherCAT slave. Stanowią rozszerzenie rodziny mikrokontrolerów RX o dużej mocy obliczeniowej i dużej pamięci wbudowanej.

Mikrokontrolery grupy RX72M zapewniają moc obliczeniową 1396 CoreMark przy częstotliwości taktowania 240 MHz (EEMBC Benchmarks). Są pierwszymi mikrokontrolerami rodziny RX z jednostką FPU o podwójnej precyzji. Umożliwiają równoczesne wykonywanie programu aplikacyjnego i obsługę komunikacji EtherCAT. Zawierają 4 MB pamięci Flash oraz dużą wewnętrzną pamięć SRAM o pojemności 1 MB, pozwalającą na obsługę np. stosów TCP/IP, web serwerów czy systemów plików bez konieczności dołączania pamięci zewnętrznej. Elastyczność tych układów otwiera również drogę do przyszłych rozszerzeń funkcjonalnych, np. OPC UA, bez dołączania dodatkowej pamięci. Pamięć Flash, podzielona na dwa banki po 2 MB, umożliwia np. uruchamianie aplikacji z jednego banku przy równoczesnym zapisie w tle informacji do drugiego banku.

Pozostałe cechy:

- szybka pamięć Flash o częstotliwości taktowania do 120 MHz,
- akceleratory funkcji trygonometrycznych (sin, cos, arctan, hypot),
- funkcje zapisu rejestrów wspierające precyzyjne operacje sterowania silnikiem,
- sprzętowy moduł kryptograficzny,
- sprzętowe funkcje ochrony pamięci i kluczy zapobiegające kopiowaniu bez autoryzacji,
- wersje w obudowach LQFP-176, BGA-176 i BGA-224.

<http://bit.ly/2kumspt>

Wysokoprądowa cewka SMD odporna na silne wibracje do 50 g

Panasonic Industry Europe wprowadza na rynek motoryzacyjny nową cewkę wysokoprądową ETQP-5M2R5YSK, wyróżniającą się odpornością na silne wibracje mogące sięgać nawet 50 g. Cewka jest zamknięta w obudowie SMD o powierzchni montażowej 8,5×8,0 mm i dzięki swoim właściwościom jest zalecana do zastosowań



w sterownikach ECU, pozwalając na ich integrowanie wewnątrz obudów silników. Dzięki zastosowaniu rdzenia ze stopu ferromagnetycznego, zapewnia stabilną pracę przy pracy z dużym prądem i w szerokim zakresie temperatur otoczenia do +150°C. Uzyskała kwalifikację AEC-Q200. Wykazuje jedynie minimalny przydzźwięk akustyczny.

Ważniejsze dane techniczne modelu ETQP5M2R5YSK:

- indukcyjność: 2,45 mH @ 100 kHz,
- rezystancja DC: 7,4 mΩ @ +20°C,
- dopuszczalny prąd DC: 12 A,
- odporność na wibracje: przyspieszenie do 50 g/amplituda do 5 mm,
- zakres częstotliwości pracy: 5...2000 Hz,
- maksymalna temperatura pracy: +150°C.

<http://bit.ly/2kFDuAX>

3-woltowe superkondensatory o pojemności 650...3000 F

Tecate Group wprowadza do oferty rodzinę 3-woltowych superkondensatorów TPLH zamykanych w obudowach Ø60 mm z gwintowanymi wyprowadzeniami osiowymi. Są one produkowane na zakres pojemności od 650 do 3000 F. Mogą stanowić bezpośrednie zamienniki wcześniejszych



wersji o napięciu znamionowym 2,7 V. Do ich kluczowych zalet należą małe gabaryty i duża gęstość energii, wyjątkowo mała rezystancja ESR oraz duża odporność na udary i wibracje. Superkondensatory TPLH mogą być stosowane jako źródła zasilania awaryjnego w zasilaczach UPS, pojazdach AGV i energetyce. Są zdadne do pracy w zakresie temperatur otoczenia od -40 do +65°C. Producent określa ich żywotność na 10 lat.

<http://bit.ly/2m9oVGb>

Miniaturowe kontrolery diod LED eliminujące przydzźwięk akustyczny

Dzięki opatentowanej metodzie sterowania Pre-Emptive Boost (PEB), miniaturowe kontrolery diod LED serii A8060x eliminują problem przydzźwięku akustycznego, wytwarzanego przez wyjściowe kondensatory ceramiczne podczas regulacji jasności sygnałem PWM. Drugą ważną zaletą jest bardzo szeroki zakres regulacji jasności, wynoszący 15000:1 przy sterowaniu sygnałem PWM o częstotliwości



REKLAMA



Obudowy plastikowe: standardowe i miniaturowe

Ponad 5000 różnych stylów obudowy:
hammfg.com/electronics/small-case

+44 1256 812812

sales@hammondmfg.eu





200 Hz oraz 150000:1 przy sterowaniu jednocześnie sygnałem PWM i analogowym. Są to układy zamykane w obudowach QFN-24 o powierzchni 4×4 mm, zawierające wbudowany kontroler i tranzystor MOSFET w przypadku ALT80600 i A80603. Z kolei A80601 i A80602 zawierają sam kontroler i współpracują z zewnętrznym tranzystorem, pozwalając na uzyskanie większej mocy wyjściowej. Wszystkie cztery układy mogą pracować w topologii Boost i SEPIC. Zostały zaprojektowane pod kątem obniżenia emisji elektromagnetycznej; zapewnia to programowalna przez użytkownika częstotliwość pracy, kontrolowany współczynnik slew-rate i programowalny dither. Wyjście sygnału zegarowego umożliwia synchronizowanie częstotliwości pracy z innymi przetwornicami impulsowymi w celu obniżenia poziomu generowanych zaburzeń EMI w całym systemie.

Kontrolery serii A8060x nadają się idealnie do zastosowań w elektronice samochodowej. Pracują z napięciem zasilania od 3,9 do 40 V (start przy min. 4,5 V). Są odporne na wahania napięcia m.in. przy rozruchu silnika, typowe dla instalacji samochodowych. ALT80600 i A80603 zawierają po 4 wyjścia, każde o maksymalnym prądzie 120 mA, zapewniające maksymalny prąd obciążenia układu 480 mA. A80601 i A80602 zapewniają sumaryczną wydajność prądową 840 mA i zawierają odpowiednio 4 oraz 6 wyjść.

<http://bit.ly/2lOhTWZ>

Czujniki prądowe Halla z kwalifikacją AEC-Q100

Firma Allegro MicroSystems dodała do oferty serię czujników prądowych ACS71240 pracujących w oparciu o efekt Halla, charakteryzujących się małymi gabarytami i energooszczędną pracą. Są to czujniki



z kwalifikacją AEC-Q100, przeznaczone do zastosowań w układach ładowania samochodów o napędzie elektrycznym, przetwornicach DC-DC i przemysłowych układach napędowych. Mogą zastępować równoważne układy pomiarowe z rezystorami bocznikowymi, wzmacniaczami operacyjnymi, transformatorami prądowymi i izolatorami napięcia wszędzie tam, gdzie liczy się powierzchnia płytki drukowanej. Pracują z napięciem zasilania 3,3 i 5 V. W porównaniu z popularnym czujnikiem ACS711 zapewniają:

- odporność sygnału wyjściowego na obce pola magnetyczne (wytwarzane np. przez pobliskie silniki lub przewody wysokoprądowe),
- mniejszy błąd pomiaru (typ. < ±1,5%),
- odporność sygnału wyjściowego na szumy występujące na liniach zasilania.

Czujniki serii ACS71240 wyróżniają się również bardzo krótkim czasem odpowiedzi wyjścia sygnalizującego przekroczenie zakresu pomiarowego, wynoszącym 1,5 μs. Jest to ważny parametr wpływający na bezpieczeństwo systemu, umożliwiający szybkie wykrycie zwarcia mogącego prowadzić do uszkodzenia tranzystorów MOSFET/IGBT w układach impulsowych dużej mocy. Dostępne są wersje w dwóch wariantach obudów: QFN (3×3 mm) do współpracy z liniami zasilania o napięciu do 120 VDC oraz SOIC-8 o izolacji 2400 V rms i napięciu pracy do 420 VDC/297 V rms.

<http://bit.ly/2lOhTWZ>

The future of test and measurement is here



Najszybszy na rynku 12-bitowy przetwornik A/C do zastosowań pomiarowych i wojskowych

Firma Texas Instruments wprowadziła do oferty najszybszy obecnie na rynku 12-bitowy przetwornik A/C zaprojektowany do zastosowań w aplikacjach wojskowych, radarach, oscyloskopach oraz w aparaturze pomiarowej do systemów 5G. ADC12DJ5200RF charakteryzuje się pasmem analogowym 8 GHz, szerszym o 20% od przetworników konkurencyjnych. Pozwala to na bezpośrednią

REKLAMA

ZAJRZYJ NA TE STRONY



www.piekarz.pl
części elektroniczne
sprzedaz@piekarz.pl tel. 22 599 49 70



www.gamma.pl
PODZESPOŁY ELEKTRONICZNE

digitalizację sygnałów o dużych częstotliwościach bez potrzeby korzystania z dodatkowych mieszaczy, których zastosowanie wiąże się z większym poborem mocy i większą wymaganą powierzchnią montażową.

Przy 12-bitowej rozdzielczości, w trybie dwukanałowym układ zapewnia szybkość próbkowania do 5,2 GSps i pasmo IBW (*instantaneous bandwidth*) równe 2,6 GHz. W trybie jednokanałowym maksymalna szybkość próbkowania i pasmo IBW wynoszą 10,4 GSps i 5,2 GHz.

Pozostałe cechy:

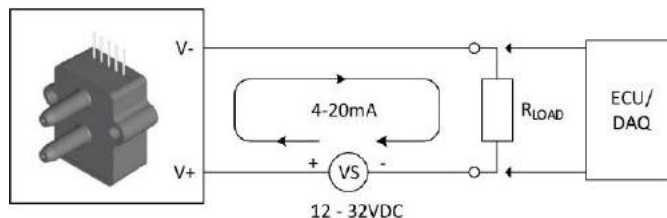
- podłoga szumowa: -151,8 dBFS/Hz (tryb dwukanałowy),
- ENOB: 8,6 (tryb dwukanałowy, $f_{IN} = 2,4$ GHz),
- ponad 100-krotnie mniejszy współczynnik CER (code error rate) od przetworników konkurencyjnych,
- pobór mocy 4 W, mniejszy o 20% od przetworników konkurencyjnych,
- interfejs JESD204C.

ADC12DJ5200RF jest zamykany w obudowie FCBGA-144 o wymiarach 10×10 mm. Zapewnia kompatybilność pod względem rozkładu wyprowadzeń z przetwornikami ADC12DJ3200, ADC12DJ2700 i ADC08DJ3200 z oferty TI, pozwalając łatwo zwiększyć maksymalną szybkość próbkowania z 2,7 do 10,4 GSps przy jedynie minimalnych zmianach układowych.

<http://bit.ly/2kuoTZ9>

Seria czujników ciśnienia 1..150 psi z wyjściem prądowym 4...20 mA

Firma All Sensors wprowadziła do oferty nową serię cyfrowych czujników ciśnienia TLAX do montażu na płytach drukowanych, realizujących pomiary różnicowe i manometryczne. Występują one w kilkunastu wersjach



o zakresach pomiarowych od 0,5 do 30 inH₂O (1...150 psi). Mogą znaleźć zastosowanie w aplikacjach HVAC i przemyśle. Zawierają wyjście prądowe 4...20 mA z cyfrową kompensacją temperatury. Są zabezpieczone przed odwróceniem polaryzacji. Zastosowana technologia CoBeam2 zmniejsza podatność na odkształcenia obudowy, zapewniając bardzo dobrą stabilność długoterminową. Czujniki TLAX charakteryzują się całkowitym błędem pomiaru wynoszącym $\pm 2\%$ FS w całym zakresie dopuszczalnych temperatur otoczenia od 0 do +50°C. Ich dryft długoterminowy wynosi $\pm 1\%$ SF w pierwszym roku pracy.

Pozostałe parametry:

- napięcie zasilania: od 12 do 32 VDC,
- pobór prądu: 22 mA,
- czas odpowiedzi (0...100% FS): 10 s,
- wymiary: 27,4×21,6×26,2 mm (z portami procesowymi).

<http://bit.ly/2kan0R0>

Fotoakustyczny czujnik stężenia CO₂ o zakresie 0...10000 ppm i 10-letniej niezawodności

Firma Infineon dodaje do oferty czujników stężenia CO₂ nowy model XENSIV PAS210 korzystający z nowo opracowanej, fotoakustycznej metody pomiaru (PAS – *photoacoustic spectroscopy*). Może on znaleźć zastosowanie do pomiaru jakości powietrza



REKLAMA

COMPUTER CONTROLS

Computer Controls zaprasza na:

Evertiq Expo

17.10.2019 Kraków

Roadshow Altium Designer

05.11.2019 Warszawa

06.11.2019 Wrocław

07.11.2019 Kraków

Seminarium MCAD – ECAD

20.11.2019 Warszawa

szczegóły wkrótce na: www.ccontrols.pl

Oficjalny
Dystrybutor

Altium

SOLIDWORKS

arm KEIL

Computer Controls Sp. z o.o.
Bielsko-Biała, ul. Budowlanych 1

tel.: +48 (33) 485 94 90

e-mail: info@ccontrols.pl
www.ccontrols.pl



wewnątrz pomieszczeń, w stacjach pogodowych, oczyszczaczach powietrza i termostatach. W odróżnieniu od tradycyjnych czujników CO₂ charakteryzuje się małymi gabarytami i niską ceną. Wewnętrzny mikrofon MEMS o dużej czułości (XENSIV IM69D130), pełniący funkcję detektora, pozwala na zmniejszenie objętości obudowy o ponad 75% w stosunku do innych tego typu czujników.



XENSIV PAS210, bazujący na wielu patentach związanych z technologią PAS, poza pomiarem stężenia CO₂ umożliwi w przyszłości pomiar stężenia również innych rodzajów gazów. W obecnej wersji zawiera przetwornik fotoakustyczny, mikrokontroler do realizacji algorytmów i przetwarzania sygnałów oraz tranzystor MOSFET do sterowania źródła IR. Struktura przetwornika fotoakustycznego obejmuje źródło promieniowania IR, wąskopasmowy filtr optyczny, pracujący na długości fali 4,2 mm specyficznej dla CO₂ oraz detektor rejestrujący zmiany ciśnienia wywołane przez molekuly CO₂. Wynik pomiaru jest konwertowany przez mikrokontroler i wyprowadzany bezpośrednio w ppm za pośrednictwem interfejsu I²C, UART lub PWM. PAS210 pokrywa zakres pomiarowy od 0 do 10000 ppm i zapewnia dokładność ± 30 ppm lub $\pm 3\%$ odczytu. W impulsowym trybie pracy deklarowana żywotność czujnika wynosi 10 lat.

<http://bit.ly/2IN71bU>

Małogabarytowe 16- i 24-bitowe przetworniki A/C o szybkości próbkowania do 153,6 kSps

Microchip wprowadza do sprzedaży rodzinę małogabarytowych, 16 i 24-bitowych przetworników A/C delta-sigma o oznaczeniach



odpowiednio MCP346x i MCP356x, mogących pracować z szybkością próbkowania do 153,6 kSps. Są one zamykane w obudowach UQFN-20 o powierzchni 3 × 3 mm, dzięki czemu mogą znaleźć zastosowanie również w urządzeniach przenośnych. Zapewniają znacznie większą szybkość próbkowania niż w przypadku innych tego typu przetworników, dla których jest to najczęściej kilka kSps. Mogą dzięki temu znaleźć zastosowanie w precyzyjnych aplikacjach z przetwornikami pomiarowymi i nadajnikami wymagających pracy z dużymi prędkościami transmisji danych. Wbudowany oscylator, czujnik temperatury i źródło prądowe pozwalają ograniczyć liczbę elementów współpracujących.

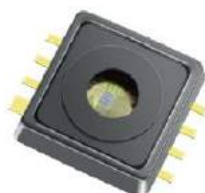
Przetworniki MCP346x i MCP356x występują w wersjach z 2, 4 i 8 kanałami asymetrycznymi oraz 2, 4 i 8 kanałami różnicowymi. Ich ceny hurtowe wynoszą od 1,46 do 3,83 USD. Do 24-kanałowego przetwornika MCP3564 dostępny jest zestaw ewaluacyjny (ozn. ADM00583).

<http://bit.ly/2IKmsBN>



Precyzyjne czujniki ciśnienia bezwzględne MAP o zakresie pomiarowym od 10 do 400 kPa

Na targach Sensor+Test firma Infineon Technologies zaprezentowała nową serię czujników ciśnienia bezwzględne XENSIV KP276 przeznaczonych do zastosowań jako samochodowe czujniki MAP (manifold air pressure). Są to czujniki z wyjściem cyfrowym, pokrywające zakresy pomiarowe od 10 do 400 kPa, zapewniające bardzo dużą dokładność i szybkość pomiaru.



MAP to ważny parametr pozwalający na obliczenie właściwego współczynnika mieszanki powietrza i paliwa dostarczanej do silnika. Duża dokładność pomiaru pozwala zwiększyć efektywność spalania i ograniczyć emisję szkodliwych związków chemicznych. Dodatkowo, wynik pomiaru ciśnienia MAP może być wykorzystywany do diagnozy uszkodzeń zaworu układu recyrkulacji spalin.

Czujniki XENSIV KP276 charakteryzują się błędem pomiaru na poziomie 0,77% FS przez cały czas życia. Ponadto, planowana jest produkcja bardziej precyzyjnych wersji o dokładności 0,5% FS, zapewniającej zgodność z wymogami standardu ISO26262. Dzięki wbudowanemu układowi przetwarzania sygnału do zewnętrznego czujnika temperatury NTC, wartości ciśnienia i temperatury mogą być wyprowadzane za pomocą tylko jednego interfejsu cyfrowego. Mała długość ramki protokołu SENT (846 μ s) i krótki czas inicjalizacji czujnika NTC (typ. 10 ms) skracają czas pomiaru i transmisji danych w porównaniu z wcześniejszymi czujnikami z oferty Infineon.

Czujniki XENSIV KP276 są zamykane w obudowach SMD-8, odpornych na ciężkie warunki pracy występujące w aplikacjach samochodowych. Mogą pracować w zakresie temperatur otoczenia od -40 do +150°C i są odporne na działanie agresywnych mediów, takich jak jod i kondensat gazów wydechowych.

<http://bit.ly/2IN71bU>



REKLAMA



UNISTREAM[®]

Zaawansowany PLC z wirtualnym panelem HMI



www.elmark.com.pl

ELMARK Automatyka Sp. z o.o.
tel. 22 541 84 60
sterowniki@elmark.com.pl

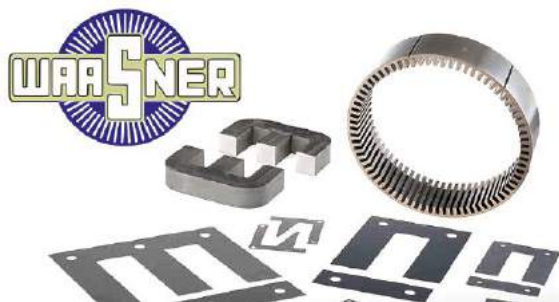




Wszelchstronne techniki połączeń

Szeroki asortyment złącz do interfejsów komunikacyjnych (USB, ETH, D-SUB) a także wiele typów złącz rastrowych typu płytka-płytkę lub przewód-płytkę, złącza okrągłe z IP68, złącza hermetyczne, listwy kołkowe a także złącza i podstawki do baterii oraz złącza dla przemysłu motoryzacyjnego i złącza audio-video znajduje się w ofercie włoskiej firmy EURO-CONN. Firma oferuje możliwość dostarczenia standardowych i niestandardowych produktów zaprojektowanych zgodnie ze specyfikacjami klientów. Siostrzana marka EURO-BLOCK oferuje szeroki asortyment listew połączeniowych (terminal-blocks), konkurujących jakością i ceną z produktami uznanych światowych producentów. Ścisła kontrola procesu produkcji gwarantuje, że komponenty nie odbiegają parametrami od, uznawanych nierzadko za standardy przemysłowe, wzorców.

<http://www.contrans.pl>



Komponenty do transformatorów

Firma WAASNER zajmuje się wytwarzaniem kształtek i rdzeni transformatorowych. Katalog produktów obejmuje kształtki EI, UI, 3UI, EE oraz kształtki do rdzeni spawanych. Oferta zawiera również kształtki pakietowane, gotowe rdzenie zwijane oraz toroidalne. Producent szczyci się posiadaniem nowoczesnego parku maszynowego i oprzyrządowania pozwalającego na szybkie przygotowanie komponentu ściśle według specyfikacji klienta. Kształtki produkowane są z blach wg norm EN10106 i EN10107.

<http://www.contrans.pl>

Nowa seria

Przekaźniki nadzorcze RPN

nadzór temperatury

nadzór napięcia



nadzór prądu

relpol® S.A.

11 różnych wykonan RPN – sprawdź



Biuro Obsługi Klienta
tel. 68 47 90 822, 850, sprzedaz@relpol.com.pl
www.relpol.com.pl

BeagleBone AI

Komputery BeagleBone to bardzo popularna platforma SBC, ustępująca sukcesowi chyba tylko Raspberry Pi. Od zawsze kierowana była do nieco innych zastosowań niż bardzo uniwersalne RPI i widać to także po najnowszej edycji tego miniaturowego komputerka. BeagleBone AI – bo o nim mowa – to płytka przeznaczona do aplikacji realizujących algorytmy sztucznej inteligencji. Dlaczego zdecydowano się na taki krok i na czym polega to przystosowanie pod kątem AI? Wyjaśniamy w artykule.

Rozwiązania wykorzystujące algorytmy sztucznej inteligencji bardzo silnie popularyzują się w ostatnim czasie – nawet nie latach, co miesiącach. Stąd powstaje coraz więcej różnego rodzaju platform, które wspierają tego typu aplikacje. BeagleBone, jako rodzina miniaturowych komputerów do różnych zastosowań, od dawna znana jest z tego, że szybko otwiera się na nowe rodzaje rozwiązań, czego efektami są np. BeagleBone Black Industrial dla rynku przemysłowego oraz BeagleBone Black Wireless i BeagleBone Blue dla rozwiązań w pełni bezprzewodowych. Najnowsze dziecko w rodzinie, BeagleBone AI pozwala wkroczyć na rynek sztucznej inteligencji.

Wiele rdzeni

O ile większość jednostek dotychczasowych odmian komputerków serii BeagleBone różniła się od podstawowego (aktualnego) modelu, BeagleBone Black, głównie zestawem atrybutów, odróżniających je właściwie tylko pod jednym lub dwoma względami od tego co w modelu podstawowym, BeagleBone AI jest platformą zupełnie inną. Pasuje do rodziny rozmiarami i kompatybilnością ze sprzętem i oprogramowaniem przygotowanym dla reszty „rodzeństwa”, ale poza tym sprawia wrażenie brata pochodzącego od innego ojca, i to poznanego na wakacjach na innym kontynencie.

O ile wszystkie pozostałe płytki BeagleBone, jakie obecnie są w sprzedaży, bazują na układzie AM3358 z pojedynczym rdzeniem ARM Cortex-A8, BeagleBone AI ma dwa rdzenie i to mocniejsze. Jego CPU to AM5729, wyposażony w dwa ARMy Cortex-A15. Układ jest też taktowany szybszym zegarem – 1,5 GHz, zamiast 1 GHz jak u rodzeństwa. To sprawia, że niejako płytka ta może bardziej konkurować z Raspberry Pi, które na przestrzeni lat też znacznie zwiększyło swoją moc obliczeniową.

To jednak nie wystarczy, by moduł uznać za zaprojektowany pod kątem sztucznej inteligencji. Kluczem jest tu wykorzystanie

dotychczasowych rdzeni, znanych z procesorów sygnałowych. W BeagleBone AI znalazły się dwa rdzenie TI C66x, które mają stanowić podstawę do realizacji większości operacji typowych dla algorytmów AI. A biorąc pod uwagę, jakie zadania najczęściej powierza się sztucznej inteligencji, twórcy znaleźli miejsce na płytce na jeszcze cztery rdzenie EVE, które wspierają wizualizację w systemach wbudowanych, obsługują zoptymalizowany interfejs API OpenCL do uczenia maszynowego TIDL (Texas Instruments Deep Learning API). Wzmocniono także pozostały zestaw rdzeni, jakie standardowo znajdują się na płytkach BeagleBone. Zamiast dwóch znalazły się cztery jednostki czasu rzeczywistego PRU, zamiast jednego rdzenia ARM Cortex-M3 wbudowano cztery rdzenie mikrokontrolera ARM Cortex-M4, a zamiast jednego rdzenia graficznego SGX PowerVR, wbudowano takie dwa.

Pozostałe usprawnienia

Tak wiele rdzeni pozwoliło na obsłużenie dużej liczby wyprowadzeń. BeagleBone AI ma 7 pinów analogowych, które można wykorzystać np. do obsługi 4-przewodowego interfejsu ekranu dotykowego. Wyprowadzenia analogowe można też ustawić do trybu pracy cyfrowej, co daje w sumie 72 dostępne wyprowadzenia cyfrowe.

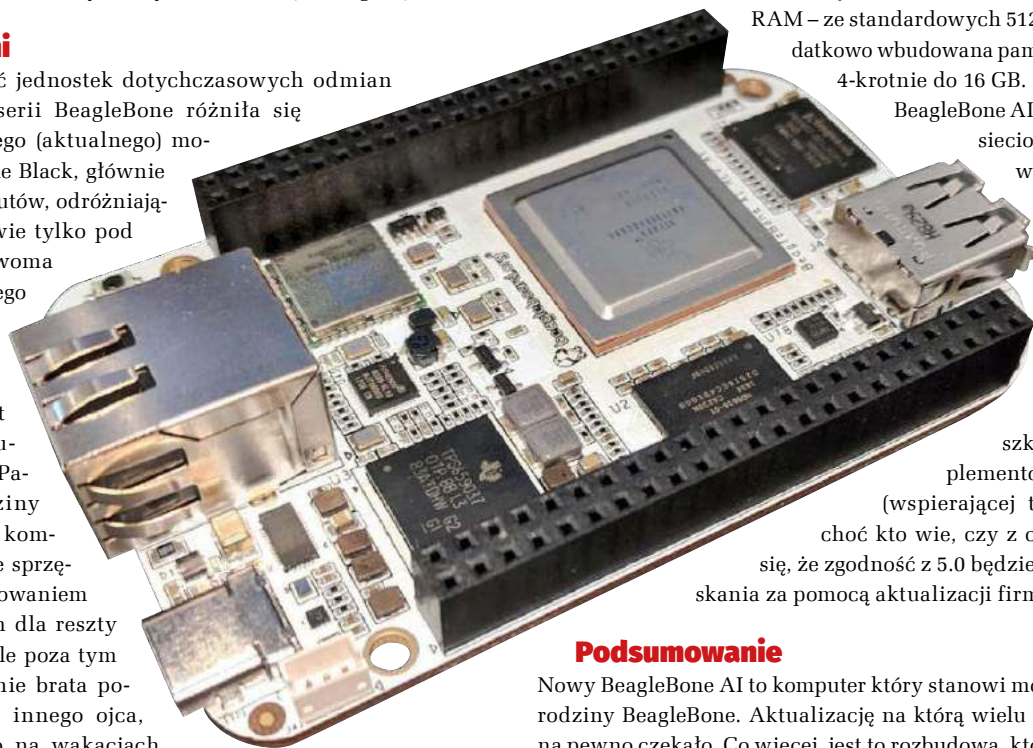
Aby znacznie większa moc CPU mogła być sensownie wykorzystana, zwiększono ilość wbudowanej pamięci RAM – ze standardowych 512 MB do 1 GB. Dodatkowo wbudowana pamięć Flash wzrosła 4-krotnie do 16 GB.

BeagleBone AI zawiera zarówno sieciowy interfejs przewodowy jak i bezprzewodowy. Oprócz gigabitowego Ethernetu obsługuje też Bluetooth 4.2 i dwuzakresowe Wi-Fi. Trochę szkoda, że nie zaimplementowano wersji 5.0 (wspierającej topologię kraty), choć kto wie, czy z czasem nie okaże się, że zgodność z 5.0 będzie możliwa do uzyskania za pomocą aktualizacji firmware'u.

Podsumowanie

Nowy BeagleBone AI to komputer który stanowi mocną aktualizację rodziny BeagleBone. Aktualizację na którą wielu miłośników serii na pewno czekało. Co więcej, jest to rozbudowa, która pozwala temu komputerowi wybiec w przyszłość i przegonić konkurencję, jeśli chodzi o wsparcie dla AI, a więc dla klasy rozwiązań, które teraz rozwijają się najszybciej. W naszym przekonaniu to w ogólności bardzo dobra decyzja, choć oczywiście pojawiają się obawy, czy cena nie będzie odstraszać. I choć BeagleBone Black kosztuje teraz około 190 zł (plus VAT), a bezprzewodowy BeagleBone Black Wireless 270 zł, to ponad 370 zł za model AI może zniechęcić osoby, które chciały potestować możliwości tej płytki. Niemniej osoby, które zainteresowane są realizacją algorytmów uczenia maszynowego w sprzęcie, powinny zainteresować się nowym BeagleBone AI.

Marcin Karbowniczek, EP



koktajl niusów



Radio w służbie komputerów (aczkolwiek dawno temu)

Commodore 64, Atari, Amiga – komputery lat osiemdziesiątych XX wieku znane zwłaszcza tym, którym było dane żyć w tym czasie. Zaskakujące były metody wprowadzania danych do tych komputerów. Sposobów było minimum trzy: przez ręczne wprowadzanie znak po znaku kodów ze specjalnych gazet (zwykle w języku BASIC), przez nagrane uprzednio kasety, które należało odtworzyć na dedykowanym magnetofonie, oraz przez nagrane uprzednio dyskietki, które wystarczyło włożyć do stosownych stacji. Jednak nie bez powodu występuje tu słowo minimum, ponieważ istniał jeszcze czwarty, choć mniej znany sposób wprowadzania danych, w dodatku znany jedynie w Europie. Sposobem tym były fale radiowe, a dokładniej dedykowane audycje, w ramach których dane przesyłano w oryginalnym formacie. Po raz pierwszy dokonali tego Holendrzy w 1980 roku w ramach audycji Hobbyscope, po czym trzy lata później zaczęła to czynić również Wielka Brytania w ramach audycji Datarama. Z kolei dwa lata później śladem tym podążyła również fińska radiostacja YLE. Bez względu jednak na to, wystarczyło nagrać audycje na kasety i wczytać z niej dane do komputera, albo też (jak czyniły bardziej doświadczane osoby) bezpośrednio zintegrować radio z komputerem (zwłaszcza przez magnetofon). Wszystko to wymagało warunków odbioru, co było dość istotne, gdyż audycja była nadawana przeważnie tylko raz (a chciało się uniknąć przekłamań psujących działanie zarejestrowanych danych, których efekt działania byłby zupełnie inny, gdyby je uwzględnić) Oczywiście przesyłano w ten sposób różne rzeczy, jedną z nich przedstawia fotografia.



HDMI jak USB (ale na odwrót)

Wtyków standardu USB, a dokładniej ich rodzaju A i związanych z nimi rozmiarów (full, mini i micro), z pewnością nikomu

przedstawić nie trzeba. Jednak nie tylko w jego przypadku tak jest – standard HDMI także odznacza się identycznymi rozmiarami. Inaczej niż dla pierwszego standardu, nie podaje się ich nazw, stosując w zamian rodzaje takie jak dla USB. Wszystko to ma na celu uniknięcie potencjalnych sporów sądowych z firmą Intel (twórca standardu USB), która mogłaby wytoczyć sprawę o bezpodstawne skopiowanie przyjętych przez siebie oznaczeń (stąd nie wymienia się również rodzaju B w HDMI).



SMS wiecznie żywy (choć w innej postaci)

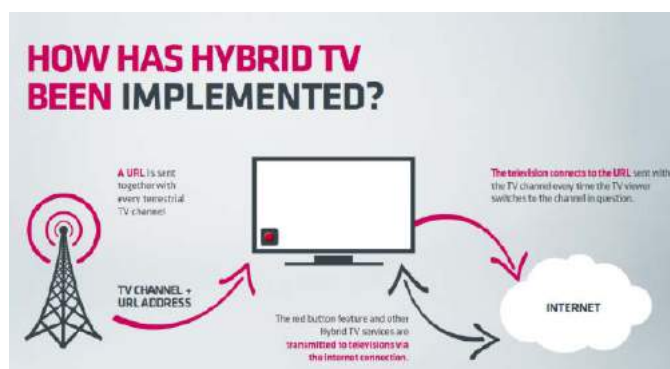
SMS pomimo spadku popularności, nie jest usługą przeznaczoną do wygaszenia. Jednak nie chodzi tu o zmianę jego dotychczasowych zastosowań, jak np. w przypadku komunikatora Gadu-Gadu, która miałaby odbudować popularność. Zamiast tego chodzi o zmianę strony technicznej tej usługi, mającej na celu jej uwspółcześnienie przez wprowadzenie rozwiązań stosowanych w takich aplikacjach jak Messenger, WhatsApp czy Viber. Krótko mówiąc: błyskawiczne wiadomości, czaty grupowe, przesyłanie dźwięków, obrazków i plików video wysokiej rozdzielczości czy transfery plików dowolnych formatów bez ograniczeń rozmiarów – wszystko co od dawna oferują wspomniane aplikacje. Z tą jednak różnicą, że w ramach telefonii GSM, bez konieczności posługiwania się dodatkowymi rzeczami (tak jak to ma miejsce teraz). Efektem jest usługa RCS – Rich Communication Service będąca następcą SMS (i zarazem nowym wcieleniem), która już teraz obsługiwana jest przez wiele telefonów i oferowana przez wielu operatorów na świecie. W Polsce jeszcze nie została uruchomiona (<http://bit.ly/2mOI2FU>) – niemniej jest to tylko kwestia czasu (na chwilę obecną, firma Orange analizuje zasadność wprowadzenia tej usługi na terenie naszego kraju). Tym samym, parafrazując słowa Francuzów: „Umarł SMS, niech żyje SMS (jako RCS)!”.

Zielone ekrany kineskopów

Obserwując wiele starych monitorów monochromatycznych i oscyloskopów analogowych można zauważyć, że wszelkie wyświetlane na nich przebiegi, napisy, obrazy i inne formy przekazu odznaczają się kolorem zielonym. Co sprawia, że są one tego koloru? Otóż związek chemiczny o wzorze $Zn_2SiO_4:Mn$, czyli krzemian cynku aktywowany manganem, świecący na zielono pod wpływem padającej



wiązki elektronowej i promieniowania ultrafioletowego. Słowem: jest to tani w produkcji luminofor w postaci proszku, który był nakładany na wewnętrzne strony ekranów kineskopów wchodzących w skład wspomnianych urządzeń.



Telewizja hybrydowa, a dostęp do niej

Z telewizją hybrydową (zwaną też HbbTV) z pewnością miała do czynienia niejedna osoba - wystarczy wspomnieć pierwsze jej emisje przez Telewizję Polską w 2012 roku, przy okazji Olimpiady Letniej w Londynie, czy fakt, że dostępne obecnie na rynku modele telewizorów oferują jej obsługę w standardzie (niczym pilot zdalnego sterowania, który jest do nich obowiązkowo dołączany). Jednak w jaki sposób telewizor wie jak do niej dostać? Jak się okazuje, droga prowadzi przez sygnał naziemnej telewizji DVB-T, a dokładniej przez adres URL, który jest w jego ramach nadawany. Słowem: wystarczy podawać ten adres w strumieniu pojedynczego kanału (nadawanego w ramach multipleksu), aby taki telewizor pobrał go, a następnie wszedł w niego jak w adres dowolnej strony internetowej w przeglądarce (i nie ma w tym słowa przesady – telewizja hybrydowa bowiem to też HTML).

O czcionce teletekstu słów kilka

Używany w teletekstach krój czcionki (widoczny zwłaszcza na starych kineskopowych telewizorach) nie jest przypadkowy. Jest to krój opracowany w 1986 roku przez holenderskiego naukowca Florisa L. van Nesa z Uniwersytetu Technicznego w Eindhoven, będący efektem prac nad polepszeniem czytelności usługi na ówczesnych odbiornikach telewizyjnych (okazało się bowiem, że stosowana wcześniej czcionka była niewystarczająca wraz z polepszaniem się w nich jakości obrazu). Krój ten okazał się na tyle dobry, że został szybko zaadoptowany przez producentów dekoderek teletekstu, w tym przez, holenderskiego Philipsa, który w tym względzie dominował – dzięki swojej pozycji na rynku, zdecydował o jego upowszechnieniu na świecie. Niemniej, jest on sukcesywnie wypierany, ponieważ

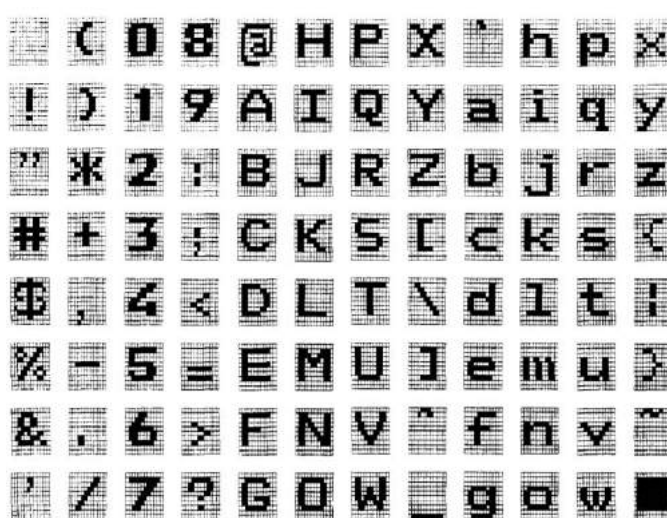


Fig. 5. The basic set of IPO-Normal 12 x 10 dot-matrix characters (copyrighted). The complete IPO-Normal set is now protected under the rules of the International Design Registration effected under the Geneva Protocol of 1975.

coraz częściej telewizję odbiera się na wyświetlaczach nie będących kineskopami, podczas gdy zaprojektowany krój był dedykowany wyłącznie tym ostatnim. Skoro tak, to można zastosować znacznie wygodniejszą czcionkę, pasującą dokładnie pod te wyświetlacze (i tak też jest czynione w wielu programach komputerowych, które obsługują teletext).



Polska

Niemcy

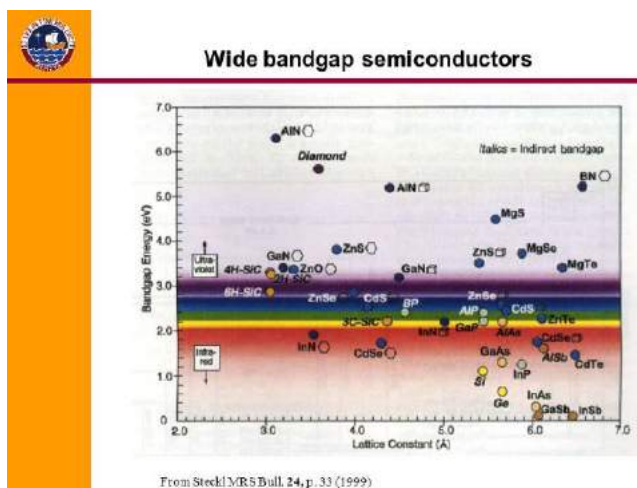
Stare (lecz nieoryginalne) sygnalizacje świetlne

Przejeżdżając samochodem przez polskie miasta można jeszcze na wielu skrzyżowaniach spotkać stare, charakterystyczne sygnalizatory (jak na fotografii po lewej stronie), których okres powstania przypadł na czasy PRL. Jednak czy wiecie, że jest to kopia sygnalizatorów (z prawej strony fotografii), które stworzyła zachodnioniemiecka firma Signalbau Huber (obecnie Swarco Traffic Systems) w latach sześćdziesiątych i siedemdziesiątych XX wieku? Jak się okazuje, jest to tylko „podróbka” niemieckiego rozwiązania, która inaczej niż oryginał, wykonana została z gorszej jakości materiałów, jako że funkcjonujący w czasach komunizmu przemysł nie był w stanie zaoferować lepszej jakości. Co więcej, „podróbki” tej dokonano jeszcze za czasów Władysława Gomułki, który choć był wrogo nastawiony do Niemiec Zachodnich, to jednak po cichu i tak skorzystał z ich pomysłu (jednak niechętnie się do takich rzeczy przyznawał – wystarczy wspomnieć jak było w przypadku kupna licencji na włoski samochód Fiat 125 za jego rządów).



Dolina krzemowa – etymologia

Kto tak naprawdę stoi za określeniem „dolina krzemowa”? Otóż amerykański dziennikarz Donald C. Hoefler (na fotografii), który współpracował z wieloma firmami z branży elektronicznej (m.in. z Fairchild Semiconductor i RCA Corp.) i pisał artykuły prasowe na ich temat. Po raz pierwszy użył tego sformułowania 11 stycznia 1971 roku na łamach tygodnika „Electronics News”, w serii artykułów na temat rozwoju przemysłu elektronicznego w Kalifornii, opatrzonych tym właśnie sformułowaniem (poczynił to dzięki sugestii jego dobrego przyjaciela Ralpha Vaersta). Tym samym jest to określenie stosunkowo młode – ma ono bowiem dopiero nieco ponad 48 lat!

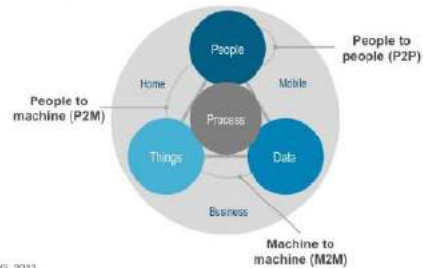


Półprzewodniki „szersze” niż zwykle

Dotychczas pod definicją półprzewodnika rozumiano materiały, których przerwa energetyczna (między pasmem przewodnictwa, a pasmem walencyjnym) jest dużo mniejsza od izolatorów. Jednak od niedawna rozumie się pod nią również materiały o przerwie energetycznej jedynie ciut mniejszej od przerwy w izolatorach. Mowa o tzw. półprzewodnikach z szeroką przerwą energetyczną, które łączą w sobie cechy tradycyjnych półprzewodników i izolatorów. Materiały te powstały jako odpowiedź na potrzebę zwiększenia wytrzymałości termicznej oraz prądowej dotychczas stosowanych półprzewodników, przy jednoczesnym zachowaniu stosowanych dotychczas niskich napięć sterowania oraz dużych częstotliwości pracy. Przykładowe z nich przedstawia rysunek, są umieszczone powyżej fioletowego paska (*lattice constant* oznacza stałą sieci krystalicznej, którą wyraża się w angstromach tj. w 10^{-10} m).

The Internet of Everything Connects the Unconnected

IoT brings together **people, process, data, and things** to make networked connections more relevant and valuable than ever before—turning information into actions that create new capabilities, richer experiences, and unprecedented economic opportunity for businesses, individuals, and countries.

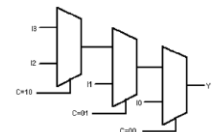


Source: Cisco IBSG, 2013.

Internet rzeczy (a nawet więcej)!

Jest Internet Rzeczy (*Internet of Things* – IoT) skupiający się na rzeczach, ale też jest i tzw. Internet Wszechrzeczy (*Internet of Everything* – IoE) obejmujący rzeczy, ludzi, dane i procesy, czyli tak naprawdę wszystko! (chodzi wyłącznie o to, co dla kogo jest przeznaczone).

4:1 Multiplexer Using if-else statements:
This implementation gives priority encoder structure.



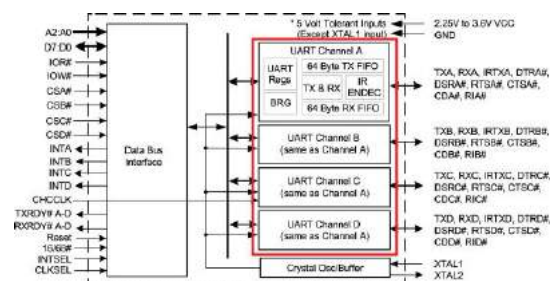
```

always@ (I0 or I1 or I2 or I3 or C1) begin
    if ((C1,C0) == 2'b00) Y = I0;
    else if ((C1,C0) == 2'b01) Y = I1;
    else if ((C1,C0) == 2'b10) Y = I2;
    else if ((C1,C0) == 2'b11) Y = I3;
end
endmodule

```

Warunek „if-else if”, a układy cyfrowe

Czym jest warunek „if-else if” (spotykany w wielu językach programowania) z punktu widzenia programowalnych układów cyfrowych (lub też z punktu pojedynczych bramek logicznych)? Otóż co najmniej jednym multiplekserem. Dowodem niech będzie rysunek ukazujący przykładowy taki warunek w postaci opisu Verilog oraz w postaci odpowiadającej siatki wspomnianego elementu (jak można zauważyć, warunek wyraża składnia zbliżona do języka C, co nie jest przypadkiem – wspomniany opis bowiem jest właśnie nim inspirowany).



QUART czyli 4*UART (choć nie tylko)!

Jest UART (*Universal Asynchronous Receiver and Transmitter*) do pojedynczej transmisji szeregowej, ale jest też tzw. QUART (*Quad Universal Asynchronous Receiver and Transmitter*) do poczwórnej takiej transmisji. Stanowi zestaw czterech układów UART traktowanych jako 4 odrębne kanały transmisji, tj. 4 pary Rx i Tx (rozwiązanie stosowane w celu zwiększenia przepustowości danych po porcie szeregowym). Jednak wbrew oczekiwaniom, nie wymyślono określić dla innych liczb takich układów. Tym samym (co może niektórych zasmucić) nie znajdziemy Bi UART-u czy BUART-u dla dwóch układów UART, Octo UART-u, czy QUART-u dla ośmiu, czy wreszcie Hex UART-u czy Hex UART-u dla szesnastu (choć nie oznacza to, że nie można tak tego teraz nazywać – kto wie czy za jakiś czas takie nazewnictwo nie przyjmie się oficjalnie?).

Energetab 2019



W dniach 17-19 września odbyły się 32. Międzynarodowe Energetyczne Targi Bielskie „ENERGETAB”, w których uczestniczyła także redakcja Elektroniki Praktycznej. Energetab to największe w Polsce targi nowoczesnych urządzeń, aparatury i technologii dla przemysłu energetycznego. W tym roku w trakcie wydarzenia swoje produkty zaprezentowało ponad 720 wystawców, pochodzących z 25 państw świata. Zajęta powierzchnia targowa wyniosła 35 tysięcy metrów kwadratowych, w tym część wewnątrz hali, a część w pawilonach namiotowych oraz na terenach otwartych.

Tematy przewodnie

W tym roku do tematów, które pojawiały się częściej niż w latach ubiegłych zaliczyć można energetykę odnawialną i różnego rodzaju instalacje IoT. Poruszano wiele zagadnień związanych z systemami bezpieczeństwa, fotowoltaiką i z elektromobilnością. Cały czas popularnością cieszą się też zagadnienia związane z oświetleniem LED-owym. Tematy te pojawiały się również wśród prelekcji i seminariów, jakie odbyły się w trakcie trwania targów.

Nagrody

W trakcie Targów przyznano nagrody i wyróżnienia dla najciekawszych produktów.

Statuetka „Złotego Lwa” Fundacji im. Kazimierza Szpotańskiego została wręczona za Tester przekładników prądowych typ mini TP, zgłoszony przez firmę ZP-BE Energopomiar-Elektryka Sp. z o.o.

Puchar Ministra Energii został wręczony za Rozłącznik napowietrzny RPZ-24, zgłoszony przez Instytut Energetyki Zakład Doświadczalny w Białymstoku.

Puchar Prezesa Polskiego Towarzystwa Przesyłu i Rozdziału Energii Elektrycznej został wręczony za produkt „Projekt Pylon - katalog słupów kratowych dla linii 110 kV”, zgłoszony przez firmę Energa Invest Sp. z o.o.

Nagroda Prezydenta Miasta Bielska-Białej została przyznana za produkt „Smartpole charger – stacja ładowania samochodów elektrycznych zintegrowana w kompozytowym słupie oświetleniowym”, zgłoszony przez firmę Alumast S.A.

Złoty Medal Polskich Sieci Elektroenergetycznych S.A. został przyznany za Układ monitoringu stanu urządzeń i systemów SWICOM, zgłoszony przez firmę ABB Sp. z o.o.

Poza tym przyznano następujące Medale 32. Międzynarodowych Energetycznych Targów Bielskich ENERGETAB 2019 oraz Medale PGE Energia Odnawialna S.A.

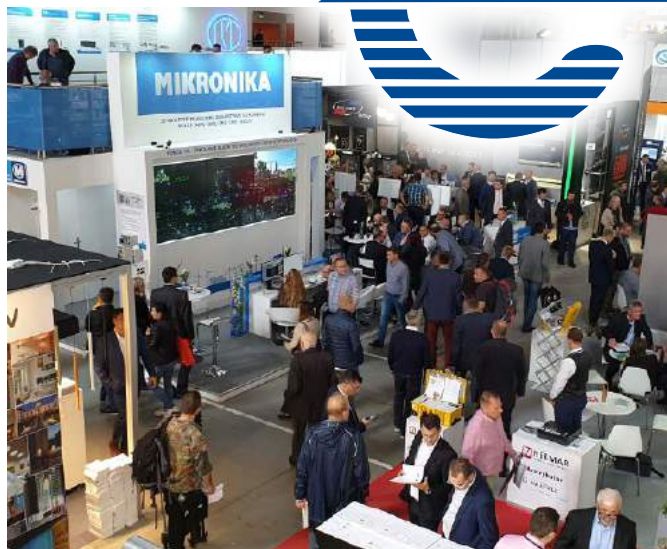
Medale złote:

- za produkt „Seria fundamentów prefabrykowanych typu FPAw2 dla wąskotrzonowych jedno- i dwutorowych słupów 110 kV serii EJ 24 i ED 24”, zgłoszony przez firmę Energoprojekt Kraków.
- za produkt „SPS-Smart Power Stations. Inteligentna Stacja Transformatorowa z Magazynem Energii”, zgłoszony przez firmę ZPUE S.A.

Medale srebrne:

- za Przekładnik napięciowy małej mocy UR-56, zgłoszony przez Sieć Badawczą Łukasiewicz - Instytut Tele- i Radiotechniczny
- za produkt „Konstrukcje wolnostojące pod moduły fotowoltaiczne”, zgłoszony przez firmę Corab Sp. z o.o.

Medale brązowe



- za Rozłącznik próżniowy napowietrzny typu RU-V, zgłoszony przez firmę JM-Tronic Sp. z o.o.
- za Monitor Energii Elektrycznej MEW-01, zgłoszony przez firmę ZAMEL Sp. z o.o.

Puchar Izby Gospodarczej Energetyki i Ochrony Środowiska przyznano za Gazowy osadzak do betonu i stali R-RAWL-SC40 II, zgłoszony przez firmę Koelner Polska Sp. z o.o.

Statuetkę „Złoty Volt” Polskiej Izby Gospodarczej Elektrotechniki nadano za Analizator jakości izolacji MIC-15k1, zgłoszony przez firmę SONEL S.A.

Statuetkę Polskiego Stowarzyszenia Branży Elektroenergetycznej wręczono za produkt „BiTservo UV2XSLCYK-J FR”, zgłoszony przez Zakłady Kablowe BITNER Sp. z o.o.

Ponadto komisja konkursowa wyróżniła Medalem Prezesa SEP Energoelektroniczny przekształtnik SVG OPTIVAR do kompensacji mocy biernej i poprawy jakości energii elektrycznej, zgłoszony przez firmę Elma Energia Sp. z o.o.

Wyróżnienia Honorowe 32. Międzynarodowych Energetycznych Targów Bielskich Energetab 2019 przyznano:

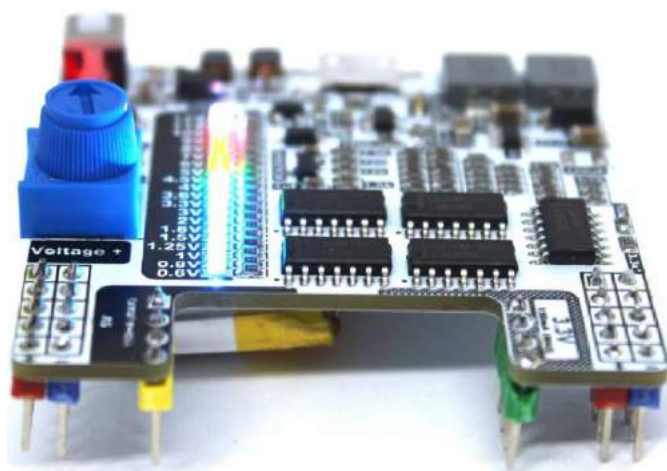
- za Przekaznik pomocniczy PWS-3/S oraz PWS-1/S, zgłoszony przez firmę Energotest Sp. z o.o.
- za produkt „Słupy wąskotrzonowe”, zgłoszony przez firmę ARINET Sp. z o.o.
- za Aktywny Filtr Harmonicznych, zgłoszony przez firmę Rabbit Sp. z o.o.
- za produkt „Rodzina routerów inQL”, zgłoszony przez firmę Grupa WB
- za produkt „Technologia komunikacji bezprzewodowej IQRF Seria 7x-Dxx”, zgłoszony przez firmę IQRF Tech s.r.o.
- za produkt „Seria innowacyjnych monolityczno-prefabrykowanych fundamentów słupów energetycznych 400 kV”, zgłoszony przez firmę Enprom Sp. z o.o.
- za produkt „Seria ORANGE - narzędzia akumulatorowe do pracy pod napięciem do 1 kV”, zgłoszony przez firmę Klauke Polska Sp. z o.o.
- za produkt „Zabezpieczenie i sterownik polowy REX640”, zgłoszony przez firmę ABB Sp. z o.o.

Kolejna, 33. edycja targów Energetab odbędzie się w Bielsku-Białej w dniach od 15 do 17 września 2020 roku.

Marcin Karbowniczek, EP

dodaj do obserwowanych

Przedstawiamy redakcyjny wybór najciekawszych projektów spośród ostatnio anonsowanych w internecie. Są to projekty na różnych etapach realizacji. Warto się zapoznać z projektami zakończonymi i śledzić realizację projektów niegotowych, by czerpać z nich inspirację do własnych prac.



OPEN Power – otwarty zasilacz laboratoryjny do płytek stykowych dla amatorów

OPEN Power to zasilacz warsztatowy dla hobbystów i amatorów. Są podłączane do płytki stykowej i zasilane z USB lub ogniwa litowo-polimerowego. Moduł ma zapewnić twórcom, studentom i hobbystom przenośne, wszechstronne, intuicyjne i tanie źródła zasilania dla wszystkich ich projektów. Zamyśłem twórców urządzenia nie jest zastąpienie normalnego zasilacza laboratoryjnego czy oscyloskopu, ale zapewnienie hobbystom taniego i dostępnego narzędzia do prototypowania układów.

Celem projektu OPEN Power jest stworzenie prototypów trzech modeli kompaktowych zasilaczy warsztatowych, kompatybilnych z płytkami stykowymi. Poszczególne modele mają się różnić parametrami i ilością wbudowanych funkcji.

Prototyp 1: Prosta i tania konstrukcja, zapewniająca stałe napięcie lub stały prąd. Napięcie regulowane może być w zakresie od 0,6 V do 12 V z diodowym wskaźnikiem poziomu napięcia. Zasilanie przez USB z funkcją podtrzymywania zasilania z ogniwa litowo-polimerowego. Szacowany koszt – poniżej 2,65 dolara.

Prototyp 2: Parametry takie jak dla pierwszego prototypu plus dodatkowo szyna zasilania ujemnego dla układów symetrycznych. Monitorowanie napięcia i prądu realizowane jest przez wyświetlacz OLED. Szacowany koszt – poniżej 5,50 dolara.

Prototyp 3: Ten moduł wykorzystuje scaloną przetwornicę, aby uzyskać napięcie do 13 V. Stabilizator liniowy o niskim spadku napięcia pozwala na regulację napięcia wyjściowego. Dzięki takiej konstrukcji urządzenie daje napięcie od 0 V do 12 V. Układ ten może pracować tak w trybie stałego napięcia jak i stałego prądu. Wyposażony w wyświetlacz OLED moduł będzie pozwalał również na śledzenie zmian pobieranej mocy. Dodatkowo, dzięki wbudowanemu gniazdu

na kartę SD, układ może pracować jako system akwizycji danych, a także oscyloskop lub analizator logiczny. Szacowany koszt systemu – poniżej 15,50 dolarów.

<http://bit.ly/2mfM8X2>



Otwarte, precyzyjne lutownice USB-C do elementów SMD w oparciu o grotty Weller'a

Niemalże równocześnie zaprezentowano dwa bardzo do siebie podobne projekty lutownic precyzyjnych. Są to RT-Pen (zdjęcie powyżej u góry) oraz Solder Ninja (zdjęcie powyżej na dole). Oba urządzenia wykorzystują grotty firmy Weller z serii RT. Są to miniaturowe grotty, zintegrowane z grzałką i czujnikiem temperatury, dedykowane do stacji lutowniczej WMRP. Ich zaletą jest doskonała jakość, szeroki wybór kształtów i relatywnie niska cena (grot kosztuje około 100 zł) w porównaniu do ceny całej stacji lutowniczej tej klasy.

Obie lutownice zasilane są poprzez port USB-C i wykorzystują technologię power delivery (USB-PD), co pozwala im pobierać z odpowiedniego źródła dużą moc, potrzebną do zasilania grotu lutownicy. Przyciski sterowania, do regulacji temperatury oraz włączania i wyłączania lutownicy znajdują się na jej obudowie, od razu pod naszymi palcami, co pozwala na sterowanie urządzeniem jedną ręką, niemalże w dowolnej pozycji.

RT-Pen (zdjęcie powyżej) korzysta z USB-PD pobierając do 3 A przy napięciu 9 V poprzez złącze USB-C. Pozwala to rozgrzać grot od 25°C do 350°C w około 6 sekund, doskonale utrzymując temperaturę. Lutownica posiada dwa przyciski sterujące i trzy diody LED. Zaprogramowane są trzy tryby pracy – z niską, średnią i wysoką temperaturą, pomiędzy którymi przełączamy się z pomocą przycisków. Nad pracą urządzenia czuwa mikrokontroler STM32F042 wraz z kontrolerem USB-PD – STUSB4500.

O Solder Ninja wiadomo w momencie pisania tego artykułu trochę mniej. Autor póki co podał informację, że również wykorzystuje mikrokontroler z rodziny STM32, korzysta on również z dedykowanego układu do konwersji temperatury z termopary typu K (MAX31855),

**PEN
Power**



jaka znajduje się w grocie. Urządzenie posiada niewielki wyświetlacz, który pokazuje dokładną temperaturę grotu. Kontroler USB-PD (FUSB302) zapewnia do 18 W mocy, co pozwala nagrzać grot do 300°C w około 10 sekund.

<http://bit.ly/2kBrAYH>

<http://bit.ly/2m8kTOe>



Konkurs Beyond Earth organizowany przez NASA i Fairchild Tropical Garden

Fairchild Tropical Botanic Garden we współpracy z NASA, stworzyło konkurs dedykowany do amerykańskich uczniów, a dotyczący wszystkich zagadnień związanych z uprawą roślin w przestrzeni kosmicznej.

„Obecne statki kosmiczne mają minimalną przestrzeń dla astronautów do życia i pracy, pozostawiając bardzo mało miejsca na cokolwiek innego. Na pokładzie statku kosmicznego środowiska do uprawy roślin muszą być małe, co ogranicza ilość produkowanej żywności” opisują problem do rozwiązania organizatorzy. Wyzwaniem jest stworzenie nowych wzorów i strategii, aby lepiej wykorzystać dostępną przestrzeń do uprawy roślin. Zespoły mają za zadanie zaprojektować, stworzyć i przetestować system, który efektywnie wykorzystuje przestrzeń do uprawy w trzech wymiarach, aby zmaksymalizować produkcję żywności.

Konkurs jest otwarty dla zespołów ze szkół średnich, wyższych i zawodowych ze Stanów Zjednoczonych. Projekty należy przesłać do 3 lutego 2020 r., a zwycięzcy zostaną ogłoszeni 14 lutego 2020 r. Zwycięzcy liczyć mogą na 500 dolarów nagrody. Mimo że jest to konkurs dedykowany tylko dla mieszkańców USA, warto monitorować postępy i oglądać nadysyłane prace (wszystkie widoczne są na portalu Instructables na bieżąco).

<http://bit.ly/2kESVcH>

Interaktywne dzwonki wietrzne, wykorzystujące Raspberry Pi 3

Perpetual Chimes (lit. Wieczne Dzwonki) to zestaw „rozszerzonych” dzwonek wietrznych, które oferują eskapistyczne wrażenia. Autorce chodziło o stworzenie zupełnie obcego i surrealistycznego krajobrazu dźwiękowego z wykorzystaniem szeroko znanego instrumentu. Ponieważ w pomieszczeniu, gdzie mogą działać dzwonki, nie ma wiatru, dzwonki wymagają interakcji ludzkiej – należy delikatnie je stukać lub machać nimi, aby wydobyły z siebie zupełnie niespodziewane odgłosy.

Dzwonki wykorzystują przewodzące, metalowe rurki nie do bezpośredniego wydawania dźwięków, a do sterowania Raspberry Pi. Program uruchomiony na tym komputerze jedynopłytkowym pełni rolę samplera. Po dotknięciu jednego z dzwonek, zwraca on obwód na jednym z wejść GPIO minikomputera, a ten odtwarza załadowaną wcześniej próbkę dźwięku.

<http://bit.ly/2l64F8c>



Europejski konkurs Astro Pi, edycja 2019/20

ESA Education wraz z Fundacją Raspberry Pi ogłosiło rozpoczęcie europejskiego konkursu Astro Pi 2019-2020. European Astro Pi Challenge to szkolny projekt prowadzony przez Europejską Agencję Kosmiczną we współpracy z Raspberry Pi Foundation. Konkurs ten daje młodym ludziom możliwość napisania kodu, który uruchomiony zostanie na komputerach Raspberry Pi, znajdujących się na pokładzie Międzynarodowej Stacji Kosmicznej. „Jesteśmy zachwyceni sukcesem programu i jesteśmy bardzo dumni z młodych ludzi, którzy biorą udział w tym programie, a kod, który piszą, działać będzie w przestrzeni kosmicznej! Tegorocznym ambasadorem European Astro Pi Challenge jest astronauta ESA Luca Parmitano” informuje ESA.

Europejskie wyzwanie AstroPi jest podzielone na dwie kategorie – Mission Zero i Mission Space Lab.

Mission Zero oferuje zespołom od 2 do 4 uczestników w wieku do 14 lat możliwość uruchomienia kodu na ISS. Zespoły piszą prosty program do wyświetlania komunikatu i odczytów temperatury na komputerze Astro Pi, dostępnym dla astronautów w trakcie wykonywania codziennych zadań. Nie jest wymagany żaden specjalny sprzęt ani wcześniejsze umiejętności programowania, a wszystkie wpisy zgodne z regułami wyzwania mają zagwarantowane uruchomienie programu w kosmosie! Termin składania wniosków upływa 20 marca 2020 r.

Mission Space Lab jest dedykowane dla zespołów złożonych z od 2 do 6 uczniów w wieku od 11 do 19 lat i oferuje uczestnikom możliwość przeprowadzenia eksperymentów naukowych na stacji kosmicznej. Konkurs polega na zaprojektowaniu i zaprogramowaniu eksperymentu do uruchomienia na komputerze Astro Pi. Najlepsze eksperymenty zostaną uruchomione na ISS, a zespoły będą miały okazję analizować zebrane dane i raportować o wynikach. Dziesięć zespołów, które opracują najlepsze raporty, zostanie wybranych jako zwycięzcy Astro Pi Mission Space Lab!

<http://bit.ly/2mDVD2J>

Kobold K2 – procesor RISC zbudowany z dyskretnych elementów TTL

Opisywana konstrukcja to nie pierwsze podejście autora do konstrukcji retro-komputera z dyskretnych elementów logicznych. Pierwszym był projekt nazwany po prostu Kobold – 16-bitowy komputer z 20 bitami adresu i wyświetlaczem wideo, który składać miał się z zaledwie kilku elementów TTL i układów pamięci. Komputer miał używać zestaw instrukcji przypominający ten z jakiego korzystała Motorola 68000 czy komputer PDP-11. „Odniosłem wrażenie, że projekt zmierza w złym kierunku. Pracowałem nad assemblerem i zaplątałem się w złożoność mikro kodu. Nie podobało mi się również to, że do dekodowania mikro kodu potrzebnych było tyle elementów. Postanowiłem więc





dokonać ogromnej zmiany w projekcie” pisze o genezie projektu Kobold K2 sam autor.

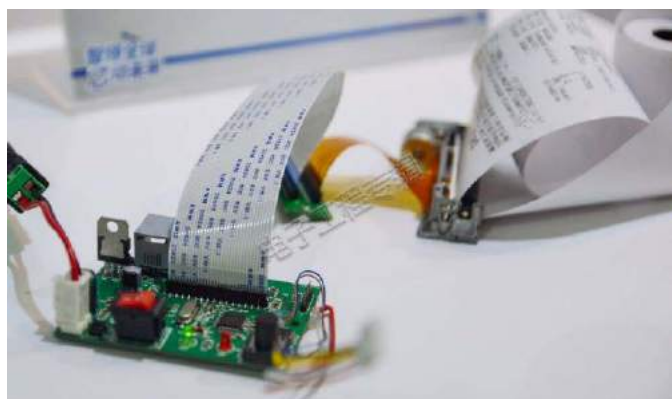
Kobold K2 to prosty procesor RISC, składający się z elementów logicznych TTL. Kompletny CPU mieścić ma się na jednej płytce drukowanej. Główne cechy procesora:

- 16-bitowa architektura z 16-bitową szyną danych,
- osiem rejestrów,
- możliwość adresowania do dwóch megabajtów pamięci
- mikrokod nie jest już używany, instrukcje będą RISC
- Wszystkie instrukcje wymagają dwóch cykli (pobieranie, wykonywanie).

Aby z CPU K2 uczynić pełnoprawny komputer, procesor uzupełniony będzie o następujące subsystemy na płycie głównej:

- system wideo – tekstowy 80×25 znaków i graficzny, w pełnym kolorze z dwoma warstwami,
- system audio,
- przestrzeń dyskową – do 32 megabajtów,
- kontroler IO.

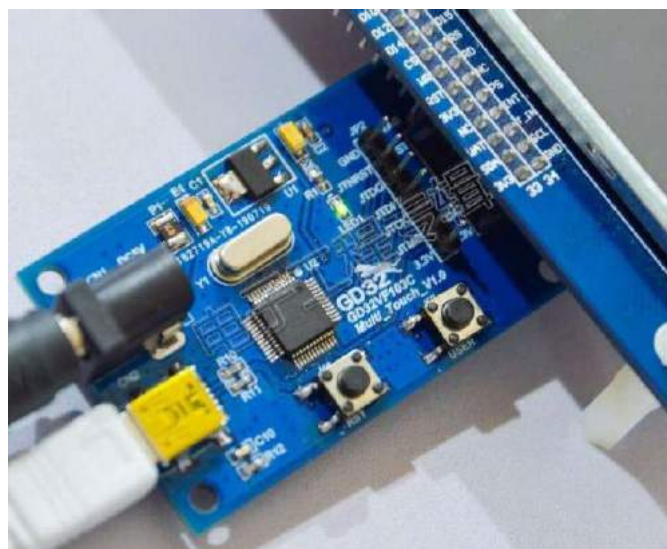
<http://bit.ly/2kFkTVG>



Nowe mikrokontrolery RISC-V na rynku i w laboratoriach

Zestaw instrukcji RISC-V zaczął powstawać w 2010 roku na uczelni w Berkeley. Miała to być otwarta architektura procesora, oparta o zasady RISC. Zadanie udało się zrealizować i opublikowano otwartą dokumentację, która dostępna jest bez opłat licencyjnych i innych. Brak opłat odróżnia RISC-V od innych architektur, takich jak ARM, które dostępne są komercyjnie.

Architektura RISC-V jeszcze niedawno była akademicką ciekawostką, jednak teraz, gdy doczekała się fizycznej implementacji w dostępnych w handlu procesorach, rynek zaczyna się nią coraz bardziej interesować, dlatego też warto zwracać uwagę na nowe układy, które



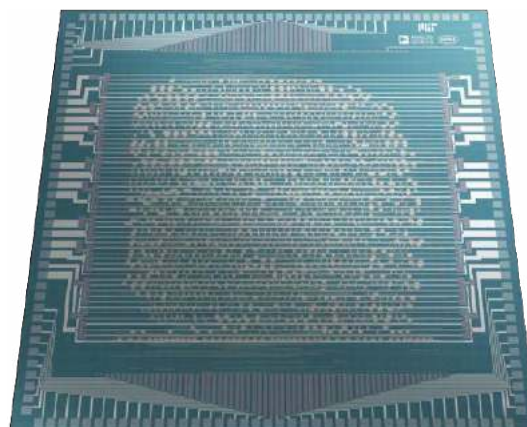
ją wykorzystują i pokazują się na rynku. Pojawiają się także pierwsze produkty korzystające z tych mikrokontrolerów.

Firma GigaDevices poinformowała niedawno o wprowadzeniu układu GD32VF103 do sprzedaży. Jest to rodzina mikrokontrolerów ogólnego przeznaczenia oparta na RISC-V. Układ ten posiada od 16 do 128 kB pamięci Flash oraz od 8 do 32 kB pamięci SRAM. Mikrokontrolery te wyposażono w dwa wielokanałowe kontrolery DMA, 12-bitowe przetworniki ADC (do 2,6 MSPS), 12-bitowe przetworniki DAC, liczne interfejsy wyjściowe (takie jak CAN czy USB OTG) oraz gFlash, czyli szyfrowaną unikatowym ID układu pamięć Flash w układzie. Na rynku pojawiły się także pierwsze implementacje tego mikrokontrolera, takie jak pokazana powyżej drukarka termiczna.

Architektura RISC-V ma jednakże nadal ogromne zastosowanie w sferach akademickich na całym świecie. Naukowcy MIT we współpracy z firmą Analog Devices wykorzystali RISC-V jako podstawę stworzenia kompletnego procesora wykonanego z... nanorurek węglowych. Ten unikatowy materiał może także być półprzewodnikiem, co wykorzystano w tym projekcie. Na zdjęciu poniżej zaprezentowano strukturę tego układu. Jakkolwiek procesor RISC-V z nanorurek węglowych nie trafi w najbliższym czasie do sprzedaży (o ile w ogóle) to pokazuje doskonale jak szerokie zastosowania są tego otwartego zestawu instrukcji. Warto obserwować rozwój układów opartych na RISC-V w najbliższej przyszłości, gdyż tego rodzaju mikrokontrolery z uwagi na brak opłat licencyjnych i w pełni otwartą architekturę rdzenia mogą w wielu zastosowaniach okazać się prawdziwym hitem.

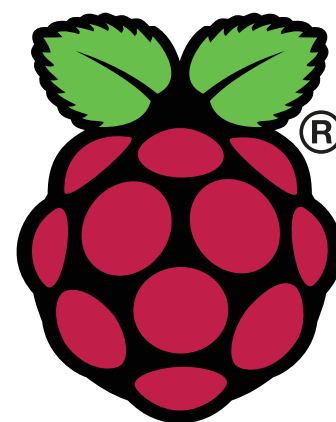
<https://ubm.io/2mbvJD8>

<http://bit.ly/2m9Jtyi>





Bramka SMS na bazie Raspberry Pi z dołączonym modułem GSM



SMS-y jeszcze do niedawna były szalenie popularne. Są to krótkie wiadomości tekstowe przesyłane z telefonu na telefon poprzez sieć komórkową. Obecnie są wypierane przez łączność wykorzystującą Internet (komunikatory internetowe, chaty i inne), jednakże nadal pełnią istotną rolę w systemie komórkowym. Możliwość wysyłania tego rodzaju wiadomości z poziomu komputera, bez wykorzystania telefonu komórkowego, jest niezwykle istotna, szczególnie gdy osoba, z którą chcemy się skontaktować nie korzysta z innych systemów komunikacji tekstowej, które wykorzystują połączenie z Internetem.

W poniższym artykule opowiemy, jak można wykorzystać komputer jednopłytkowy z serii Raspberry Pi z dołączonym modemem GSM do wysyłania krótkich wiadomości tekstowych – popularnych SMS-ów. Do tego popularnego minikomputera można dołączyć, na wiele sposobów, modem GSM, pozwalający na komunikowanie się z siecią telefonii komórkowej. W artykule opisany jest sposób podłączania popularnych modemów na USB do minikomputera Raspberry Pi oraz przedstawiona jest biblioteka, dedykowana

do wysyłania i odbierania SMS-ów za pomocą tego minikomputera z modemem GSM.

Tego rodzaju system może mieć wiele zastosowań – samodzielnie, bądź w połączeniu z innymi urządzeniami, czy systemami, takimi jak na przykład centralka alarmowa czy system automatyki domowej. W podsumowaniu artykułu przedstawimy kilka potencjalnych zastosowań takiego systemu.

Modemy GSM

Aby dodać funkcjonalność GSM do komputera jednopłytkowego, takiego jak Raspberry Pi, musimy go wyposażać w modem GSM. Na rynku dostępnych jest wiele urządzeń tego rodzaju – przyjrzyjmy się im, aby wiedzieć jaki mamy wybór, projektując naszą bramkę SMS. Modemy GSM podłączane są do minikomputerów na kilka sposobów. Typowo, poprzez interfejs szeregowy (UART), który jest wyprowadzony na 40 pinowym złączu GPIO, ale także modem może być podłączony poprzez mostek UART-USB lub jako 'dongle' – zintegrowany modem w postaci urządzenia podpinanego bezpośrednio do portu USB komputera.

Jeśli chodzi o modemy GSM podłączane do sprzętowego interfejsu UART na rynku dostępnych jest wiele zintegrowanych modułów – w postaci modułów HAT oraz samodzielnych elementów, do zabudowania w swoich projektach. Moduły typu HAT, czyli dołączane



Fotografia 1. Moduł typu HAT z modemem GSM oraz odbiornikiem GPS (czerwone PCB) na Raspberry Pi (zielone PCB).

do złącza GPIO minikomputera Raspberry Pi, są najprostsze do użycia w projektach hobbystycznych – wystarczy zamontować moduł na płytce komputera. Przykład takiego rozwiązania pokazano na **fotografii 1**. Wszystkie elementy niezbędne dla działania modemu GSM znajdują się w takim module. Także antena lub gniazdo do podłączenia zewnętrznej anteny. Jeśli nie chcemy korzystać z modułu tego rodzaju, możemy zbudować własny w oparciu o zintegrowany modem GSM o odpowiadających nam parametrach (2G, 3G, 4G) i z potrzebnymi nam dodatkowymi funkcjami. Modem wymaga zainstalowania karty SIM lub tzw. *embedded SIM* (eSIM), czyli karty w postaci układu scalonego, lutowanego na płytce.

Osobną klasę sprzętów stanowią popularne modemy GSM na USB. Ich podłączanie do komputera z systemem operacyjnym Windows jest bardzo proste, niestety w przypadku Linuksa, który standardowo instalowany jest na komputerach jedнопłytkowych, nie jest to już takie trywialne. Istnieje wiele przewodników dostępnych w sieci, wykorzystujących np. skrypt *sakis3g*, ale uruchomienie niektórych modemów wymaga większego nakładu pracy. Poniżej przedstawimy uniwersalny sposób na podłączenie tego rodzaju modemu do Raspberry Pi pracującego z systemem operacyjnym Raspbian.

Główny problem z podłączaniem tego rodzaju dongla do Raspberry Pi polega na tym, że większość modemów USB zgłasza się jako dwa urządzenia – urządzenie pamięci masowej USB oraz modem USB. Po podłączeniu do Raspberry Pi urządzenie zwykle znajduje się w trybie pamięci USB. Istnieje specjalny program o nazwie *usb_modeswitch*, który może zostać wykorzystany do przełączania trybów zgłaszania się dongla USB. Drugą brakującą część to sposób na połączenie się z siecią komórkową za pomocą modemu USB z poziomu minikomputera Raspberry Pi. W tym celu wykorzystane zostanie klasyczne oprogramowanie *ppp* i *wvdial*.

W pierwszej kolejności musimy zainstalować program do przełączania trybów urządzenia USB. W tym celu korzystamy z *apt-get*; w terminalu systemu wpisujemy:

```
sudo apt-get install usb-modeswitch
```

Po zainstalowaniu oprogramowania musimy ustalić numery VID oraz PID (odpowiednio Vendor ID i Product ID – ID producenta oraz ID produktu) naszego modemu GSM, aby móc odpowiednio skonfigurować *usb-modeswitch*. W tym celu podłączamy modem GSM do komputera i resetujemy Raspberry Pi. Najlepiej robić to bez podłączonego Internetu do minikomputera. Po uruchomieniu systemu operacyjnego w terminalu wpisujemy:

```
lsusb
```

Uruchomimy wbudowane w Linuksa narzędzie do wyświetlania informacji o systemowych szynach USB i podłączonych do nich urządzeniach. Komenda zwróci nam listę urządzeń USB podłączonych do komputera, tak jak pokazano na **rysunku 2**. System wykrywa poprawnie modem USB, który zgłasza się swoimi numerami VID oraz PID, rozdzielonymi dwukropkiem (podkreślone pomarańczową linią). Możemy teraz zresetować komputer, aby zainstalowany wcześniej *usb_modeswitch* zmienił tryb działania dołączonego urządzenia oraz ponownie wyświetlić podłączone urządzenia USB. Numer PID urządzenia zmienił się. Zapamiętane numery VID i PID urządzenia w obu trybach posłużą nam do konfiguracji *usb_modeswitch*, tak, aby urządzenie uruchamiało się za każdym razem jako modem GSM. W tym celu musimy wypakować, na przykład do folderu */tmp*, odpowiedni plik konfiguracyjny. W tym celu w terminalu wpisujemy:

```
cd /tmp
tar -xzf /usr/share/usb-modeswitch/configPack.tar.gz 19d2\:2000
```

```
Bus 001 Device 002: ID 0424:2514 Standard Microsystems Corp. USB 2.0 Hub
Bus 001 Device 003: ID 0bda:0136 Realtek Semiconductor Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 001 Device 006: ID 19d2:2000 ZTE WCDMA Technologies MSM MF627/MF628/MF628+/MF636+ HSDPA/HSUPA
Bus 001 Device 004: ID 046d:c50c Logitech, Inc. Cordless Desktop S510
```

Rysunek 2. Przykładowa lista, drukowana na ekranie przez polecenie *lsusb*. Kolorem pomarańczowym podkreślono VID:PID modemu GSM w trybie pamięci USB

```
# ZTE devices
```

```
TargetVendor= 0x19d2
TargetProductList="0001,0002,0015,0016,0017,0031,0037,0052,0055,0063,0064,0066,0091,0108,0117,0128,0157,2002,2003"

MessageContent="555342431234567800000000000061e00000000000000000000000000000"
MessageContent2="55534243123456790000000000000061b00000002000000000000000000000"
MessageContent3="55534243123456702000000080000c85010101180101010101000000000000"

NeedResponse=1
```

Rysunek 3. Przykładowa zawartość pliku konfiguracyjnego usb_modeswitch dla VID:PID 19d2:2000

gdzie VID i PID (19d3:2000 w powyższym przykładzie) zastępujemy numerami identyfikacyjnymi, jakie odpowiadają naszemu urządzeniu. Możemy teraz otworzyć wypakowany plik konfiguracyjny z pomocą naszego preferowanego edytora tekstowego (większość użytkowników Raspberry Pi używa vi albo nano). Na **rysunku 3** widzimy przykładową zawartość takiego pliku dla podanych powyżej VID i PID. Zapamiętajmy lub zapiszmy dane zaznaczone na niebiesko w na tym rysunku, będą nam potrzebne do konfiguracji programu.

Po zapoznaniu się z danymi dla naszego urządzenia możemy otworzyć plik konfiguracyjny `/etc/usb_modeswitch.conf` i dodać do niego informacje, jakie uzyskaliśmy powyżej. Po edycji plik powinien wyglądać następująco (dla powyższych przykładowych danych):

```
DefaultVendor=0x19d2
DefaultProduct=0x2000
```

```
TargetVendor=0x19d2
TargetProduct=0x2002
```

```
MessageContent="555342431234567800000000000000  
61e000000000000000000000000000000000"  
MessageContent2="555342431234567900000000000000  
61b000000002000000000000000000000000"  
MessageContent3="5553424312345670200000000800  
00c850101011801010101010000000000000"
```

W ten sposób określamy, jakie ma być docelowe VID i PID podłączonego urządzenia – ma to być modem GSM.

Po skonfigurowaniu urządzenia tak, aby uruchamiało się każdorazowo jako modem GSM, musimy skonfigurować łączenie się z siecią GSM. W tym celu instalujemy wymienione wcześniej w tekście programy ppp oraz wvdial. Do instalacji używamy klasycznie apt-get. W terminalu wpisujemy:

```
sudo apt-get install ppp wvdial
```

Po zainstalowaniu potrzebnego oprogramowania możemy przystąpić do konfiguracji *wvdial*. W tym celu otwieramy do edycji plik konfiguracyjny */etc/wvdial.conf*. W pliku tym wpisujemy następujący blok konfiguracji:

```
[Dialer 3gconnect]
Init1 = ATZ
Init2 = ATQ0 V1 E1 S0=0 &C1 &D2 +FCLASS=0
Init3 = AT+CGDCONT=1,"IP","APN"
Stupid Mode = 1
Modem Type = Analog Modem
ISDN = 0
Phone = *99#
Modem = /dev/gsmmodem
Username = { }
Password = { }
Baud = 460800
```

W ten sposób skonfigurowaliśmy tzw. dialer, czyli moduł łączący nas z Internetem. W powyższej konfiguracji zamiast zaznaczonego na żółto *APN* podać musimy nazwę sieci pakietowej, z jakiej korzystamy do łączenia się z Internetem. Numer telefonu, zaznaczony na żółto – *99# – jest typowym numerem dostępowym w internecie

```

Listing 1. Biblioteka do połączenia z modemem GSM
import serial, time, logging
from curses import ascii

# logowanie działania aplikacji
logger = logging.getLogger(__name__)

def openModem(modem, time):
    """Otwarcie połączenia z modemem"""
    logger.info("Opening " + modem)
    serialPort = serial.Serial(modem, 460800, timeout=5)
    if serialPort.isOpen():
        logger.info("Modem opened.")
        return serialPort
    else:
        logger.error("Could not open modem")
        return -1

def closeModem(modem):
    """Zamknięcie połączenia z modemem"""
    if modem.isOpen():
        modem.close()
        logger.info("Modem closed.")
    else:
        logger.error("Modem is not opened.")

def checkModem(modem):
    """Sprawdzenie połączenia z modemem"""
    modem.write("AT\r")
    modem.readline()
    status = modem.readline()
    if status.startswith("OK"):
        logger.info("AT OK")
        return True
    else:
        logger.error("AT ERROR")
        return False

def setModemTextMode(modem):
    """Ustawienie modemu w trybie tekstowym
    (do wysyłania wiadomości SMS)"""
    modem.write("AT+CMGF=1\r")
    modem.readline()
    status = modem.readline()
    if status.startswith("OK"):
        logger.info("Modem set in text mode")
        return True
    else:
        return False

def sendSMS(modem, phoneNumber, text):
    """Wysłanie wiadomości SMS"""
    modem.write('AT+CMGS="%s\r' % phoneNumber)
    modem.write(text)
    modem.write(ascii.ctrl('z'))
    time.sleep(2)

    modem.readline()
    modem.readline()
    modem.readline()
    cmgi = modem.readline()
    modem.readline()
    modem.readline()
    status = modem.readline()
    if status.startswith("OK"):
        logger.info("SMS sent.")
        return True
    else:
        logger.error("Error sending SMS")
        return False

def deleteSMS(modem, messageIndex):
    """Skasowanie wiadomości SMS z pamięci"""
    modem.write("AT+CMGD=%s\r" % messageIndex)
    modem.readline()
    status = modem.readline()

    if status.startswith("OK"):
        logger.info("SMS deleted")
        return True
    else:
        return False

def readSMS(modem, messageIndex):
    """Odczytanie wiadomości SMS"""
    modem.write("AT+CMGR=%s\r" % messageIndex)
    time.sleep(2)

    modem.readline()
    header = modem.readline()
    logger.info(header)
    body = modem.readline()
    modem.readline()
    status = modem.readline()

```

mobilnym, ale istnieje możliwość jego edycji, gdyby nasz dostawca usług wykorzystywał inny. Jeśli nasze połączenie wymaga podania nazwy użytkownika oraz hasła (odpowiednio *Username* i *Password*), to wpisujemy je w nawiasy klamrowe w pliku konfiguracyjnym.

Po uzupełnieniu wszystkich danych możemy przystąpić do łączenia się z siecią. Aby to zrobić musimy w pierwszej kolejności skonfigurować dongle do pracy jako modem USB, a następnie wykorzystać skonfigurowany przed chwilą dialer do połączenia się z siecią. W tym celu w terminalu wpisujemy:

```
sudo usb_modeswitch -c /etc/usb_modeswitch.conf
wvdial 3gconnect
```

Komendy AT

Do komunikacji z większością modemów wykorzystuje się tak zwane komendy AT, zwane czasami komendami Hayes. Zarys tego języka poleceń opracowany został na początku lat '80 XX wieku przez Dennisa Hayes, dla opracowanej przez jego firmę Hayes Microcomputer Products, rodziny modemów do przesyłu danych przez sieć telefoniczną, Smartmodem. Pierwsze urządzenie, które korzystało z tego zestawu komend, był Smartmodem 300 (pracujący z prędkością 300 bodów, czyli około 300 bajtów na sekundę) wypuszczony na rynek w 1981 roku. Od tego czasu komendy te ewoluowały i znalazły swoje zastosowanie w szerokiej gamie modemów, także GSM (zestaw komend AT dla modemów GSM opisany jest w specyfikacji 3GPP TS 27.007).

Nazwa komend AT pochodzi od słowa *attention*, czyli „uwaga”. Te dwa znaki rozpoczynają każdą komendę wysyłaną do modemu. Komendy dla modemów GSM opisane są w specyfikacji 3GPP TS 27.007, a dodatkowo w 3GPP TS 27.005 znajduje się opis komend wykorzystywanych do wysyłania SMSów. Od typowych komend AT, używanych w przeszłości, odróżnia je wykorzystanie znaku „+” po AT w większości komend. Wykorzystanie tego znaku wprowadzone zostało do komend AT w połowie lat '90 XX wieku, wraz ze standardem komend dla modemów V.250.

Przykładowo, komenda AT+CPIN=1234 wprowadzi do karty SIM numer PIN – 1234, z kolei np. AT+CSQ pozwala na pomiar siły sygnału – modem zwraca siłę odbieranego sygnału (RSSI) oraz odsetek błędów bitowych (BER) w danym momencie. Dokładne informacje na temat zaimplementowanych w konkretnym modemie komend, znaleźć można w jego dokumentacji. Jakkolwiek specyfikacja 3GPP mówi o tym, jakie zestawy komend powinny być zaimplementowane, to producenci różnie je implementują, lub nie implementują ich wcale.

Biblioteka do połączenia z modemem GSM

Biblioteka, która zaprezentowana jest na **listingu 1** jest dedykowana do stosowania w skryptach języka Python. Korzysta z komend AT, aby kontrolować pracę modemu – wysłać i odbierać wiadomości tekstowe. W bibliotece zdefiniowano szereg funkcji, które używane są do komunikacji z modemem GSM w celu wysłania czy odebrania wiadomości tekstowej. Komentarze w **listingu 1** powinny dostatecznie tłumaczyć zastosowanie poszczególnych funkcji. Jako parametr *modem* podajemy adres sprzętowy portu szeregowego do którego dołączony jest nasz modem. W przypadku dongla na USB będzie to, na przykład, */dev/ttyUSB1*.

Stosując opisywaną bibliotekę do wysyłania wiadomości SMS w pierwszej kolejności należy połączyć się z modemem, wykorzystując funkcję *openModem*. Następnie trzeba przełączyć modem do trybu tekstowego korzystając z funkcji *setModemTextMode* i już można wysłać SMS-y z pomocą funkcji *sendSMS*. Po zakończonej pracy z modemem nie zapominajmy zamknąć połączenia funkcją *closeModem*.

```
Listing 1. cd
logger.info("Status is: " + status.strip())
if status.startswith("OK"):
    logger.info("SMS read.")
    headerParts = header.split(",")
    if len(headerParts) > 1:
        return "-".join([headerParts[1].strip("'"), body])
    else:
        return body
else:
    logger.error("Error reading SMS")

def readLineFromModem(modem):
    """Odczytanie jednej linii z interfejsu UART"""
    return modem.readline()

def flushBuffer(modem):
    """Wyczyszczenie buforu interfejsu UART"""
    modem.readlines()

def getMessageIndex(newMessageString):
    """Pobierz ilość SMSów (aktualny indeks w liście)"""
    newSMSIndicationParts = newMessageString.strip().split(":")
    logger.debug(newSMSIndicationParts[1])
    messageIndex = newSMSIndicationParts[1].split(",")[1]
    logger.debug(messageIndex)
    return messageIndex
```

Przykłady wykorzystania bramki GSM

Podstawowym zastosowaniem opisanego wyżej systemu jest oczywiście wysyłanie i odbieranie SMS-ów. Taka funkcjonalność przydatna jest w szerokiej klasie programów. Bibliotekę można zastosować do stworzenia bramki SMS – urządzenia, które pozwoli nam wysłać z poziomu komputera wiadomości tekstowe. Jeżeli uzupełnimy program napisany w Pythonie o interfejs webowy, na przykład napisany z wykorzystaniem frameworku *Flask*, możemy stworzyć prostą bramkę SMS, dostępną jako strona w naszej wewnętrznej domowej sieci.

Biblioteka może posłużyć też do rozbudowy innych aplikacji o możliwość komunikacji za pomocą wiadomości SMS. W szczególności w systemach automatyki domowej może to być wartościowe – pozwala z jednej strony np. sterować poszczególnymi urządzeniami w domu z pomocą wiadomości SMS lub wysłać SMS-owe alerty

Z systemu SMS skorzystać można też w samochodzie. Raspberry Pi w naszym pojeździe może np. komunikować się z interfejsem diagnostycznym OBD-II, aby zbierać istotne parametry pracy silnika w naszym pojeździe, a następnie wysłać nam SMS-y, jeżeli któreś z nich wychodzą poza normę, lub też auto zgłasza jakieś usterki z pomocą kodów błędów. Jeśli do takiego urządzenia dodamy np. moduł nawigacji satelitarnej (GPS), który często jest instalowany na wspólnych modułach HAT z modemami GSM, z łatwością stworzyć możemy system śledzenia pozycji naszego auta, przydatny w razie kradzieży pojazdu... lub gdy zapomnimy, gdzie go zaparkowaliśmy.

Nikodem Czechowski

REKLAMA

OBWODY DRUKOWANE

- płytki jednostronne i dwustronne
- płytki na podłożu aluminiowym
- testy elektryczne płytek
- pokrycia płytek: cyna lub cyna/ołów



WYKONANIE ZŁOŻYŃ
FALDRUK

Faldruk s.c.



Rower elektryczny (2)

W pierwszym artykule poznaliśmy historię rowerów elektrycznych oraz dowiedzieliśmy się jak wyglądają teraz i jakie komponenty są potrzebne do ich budowy. W tej części pokażę jak zamontować napęd elektryczny do zwykłego roweru. Do prawie każdego roweru można dodać taki napęd, nie jest to bardzo skomplikowane, większość osób powinna dać sobie z tym radę.

Modyfikacji zostanie poddany zwykły rower, widoczny na fotografii tytułowej. Głównie założenie jest takie, że będzie pojazdem, który umożliwi dojazdy do pracy, wyjazdy rekreacyjne itp., lecz dalej będzie to rower. Dzięki modyfikacji pozwoli na pokonanie znacznie większych dystansów, w o wiele krótszym czasie, przy włożonym podobnym wysiłku, w stosunku do wersji podstawowej.

Do tego projektu użyję gotowego zestawu, który składa się z następujących komponentów:

- Silnik. Typu HUB, przekładniowy, zaopleciony w obręcz, wystarczy tylko nałożyć na felgę taśmę ochronną, dętkę oraz oponę.
- Sterownik. Jest modulem, który łączy wszystkie komponenty zestawu.
- Klamki hamulcowe z czujnikami. Dzięki sygnałom z czujników, sterownik wyłącza silnik gdy naciskamy hamulec.
- Komputer pokładowy. Zastępuje licznik rowerowy i dodatkowo pozwala

zmieniać tryby wspomagania oraz pokazuje stopień naładowania baterii. Komputer zamontowany na kierownicy pokazuje **fotografia 1**.

- Manetka przyspieszenia, również widoczna na fotografii 1.
- Czujnik PAS – czujnik pedałowania.

W zestawie brakuje jeszcze trzech bardzo ważnych elementów.

Akumulator

Pierwszym elementem, o który musimy uzupełnić zestaw, jest akumulator. W większości rowerów elektrycznych stosuje się akumulatory wykonane w technologii litowo-jonowej (li-ion). Ja również użyję akumulatora tego typu, takiego jak na **fotografii 2**. Jego napięcie znamionowe wynosi 36 V, a pojemność 20 Ah, co daje około 720 Wh przy pełnym naładowaniu. Taka ilość energii przekłada się na przynajmniej 50...60 km zasięgu na najwyższym stopniu wspomagania napędem.



Fotografia 1. Kierownica roweru po konwersji, z zamontowaną manetką i komputerkiem pokładowym



Fotografia 2. Akumulator w obudowie, do napędu elektrycznego

Jeśli kupujemy akumulator, to najlepiej w gotowej obudowie, np. popularnej obudowie bidonowej, którą montujemy w miejscu koszyka na bidon. Jest to bardzo korzystna opcja dla lekkiego roweru. Akumulator w takiej obudowie ma ważną zaletę - można go bez problemu zdemontować z roweru. Jest to nieocenione w sytuacji, gdy nie mamy możliwości ładowania akumulatora w miejscu, w którym przechowujemy rower.

Niektóre akumulatory (szczególnie te nietypowe, wykonywane na zamówienie) nie mają żadnej obudowy, są tylko zabezpieczone rękawem termokurczliwym. W takiej sytuacji sami musimy wykonać obudowę do akumulatora. Często skrzynkę z akumulatorem i elektroniką montuje się w trójkącie ramy. Jeśli zdecydujemy się wykonać ją własnoręcznie to możemy użyć do tego celu sklejki, dibondu czy laminatu. Taką usługę można również zlecić

firmie, która wykona skrzynkę z laminatu o dowolnym kształcie, jednak będzie się to wiązało ze znacznym kosztem.

Akumulator z reguły nie pasuje idealnie do pojemnika. Aby go sztywno umocować, najczęściej stosuje się zwykłą budowlaną piankę montażową nakładaną dookoła pakietu. Takie rozwiązanie nie mocuje baterii całkiem sztywno, zastygła pianka amortyzuje większość drobnych drgań. Przy większych akumulatorach pianka może się okazać za słaba, aby utrzymać dużą masę, ale do akumulatorów poniżej 6 kg jest odpowiednia. W przypadku zastosowania pianki, na początku używania naszego „nowego” roweru warto co jakiś czas sprawdzić, czy akumulator nie jest obluźniany. Wbrew pozorom, takie rozwiązanie pozwala na demontaż akumulatora. W sytuacji awaryjnej należy zwyczajnie wyciąć piankę nożem montażowym.

Jest jeszcze jedna metoda mocowania akumulatora w rowerze, ale wymaga zamontowania bagażnika. Jeśli akumulator ma kształt prostokąta podobnego wymiarami do kształtu bagażnika to można użyć zwykłej rowerowej torby na bagażnik, w której zostanie umieszczony. Jednak najpierw akumulator musi być umieszczony w dodatkowej skrzyneczce, która ograniczy ryzyko uszkodzenia w razie upadku. Całość musi być przymocowana na sztywno do bagażnika, można w tym celu użyć dużych opasek zaciskowych. Ten typ mocowania w niektórych przypadkach może być nieodpowiedni, ponieważ, dodaje więcej masy na tył pojazdu oraz podnosi środek ciężkości. Może to skutkować dziwnymi odczuciami podczas jazdy i troszkę mniejszą stabilnością, ale zaletą jest bardzo dobry kamuflaż akumulatora i trzeba być na prawdę spostrzegawczym, żeby zauważyć że nie jest to zwykły rower.

Blokada obrotu osi silnika

Kolejnym elementem, o który musimy uzupełnić zestaw jest blokada osi silnika.

W opisywanym rowerze, którego napęd dysponuje stosunkowo małą mocą, prawdopodobnie wystarczyłaby podkładka z tak zwanym „uchem”, która znajduje się w zestawie razem z silnikiem. Zdecydowałem, że jednak założę podstawową blokadę obrotu, aby mieć pewność że nic złego się nie stanie. Zamontowaną blokadę pokazuje **fotografia 3**. Jest to element bardzo potrzebny, jeśli mamy do czynienia z konstrukcjami o większej mocy i poświęcę mu więcej uwagi w kolejnej części artykułu opisującej „mocniejszy” rower. Brak tego elementu może doprowadzić do urwania haków w ramie lub do pęknięcia osi silnika i są to niebezpieczne sytuacje.

Ładowarka

Ostatnim elementem, który potrzebujemy jest ładowarka. Zwykle brakuje jej w zestawie do konwersji, czasami można kupić ją razem z akumulatorem. Zasadniczymi parametrami jakimi trzeba się kierować przy zakupie ładowarki, przeznaczonej do baterii typu li-ion, jest prąd ładowania oraz napięcie końcowe ładowania (napięcie maksymalne akumulatora). Te dwa parametry muszą być odpowiednio dobrane do każdego akumulatora. Napięcie końcowe powinno być podane w parametrach akumulatora. Jeśli nie ma takiej informacji to wartość tę można wyliczyć na podstawie ilości sekcji szeregowych, które składają się na pakiet akumulatora. Nie należy mylić napięcia znamionowego akumulatora z napięciem końcowym ładowania.

Prąd ładowania przekłada się na czas ładowania. Czas można oszacować dzieląc pojemność akumulatora podaną w Ah przez wartość prądu ładowania. Do wyniku końcowego należy dodać przynajmniej 30 min., ponieważ pod koniec ładowania wartość prądu sukcesywnie spada.

Na przykład: mamy baterię o pojemności 17 Ah i ładowarkę, która ładuje prądem 3,5 A. Wynik obliczeń daje prawie 5 godzin, więc realnie akumulator powinien



Fotografia 3. Zamontowana blokada obrotu osi silnika



Fotografia 4. Podkładka dostosowana do kształtu osi silnika

naładować się od 0 do 100% w czasie około 5,5 godziny. Im większym prądem ładowania dysponuje ładowarka tym szybciej naładuje akumulator, lecz szybkie ładowanie przyczynia się do skracania żywotności naszego akumulatora. Optymalny czas ładowania powinien wynosić przynajmniej 4 godziny.

Montaż zestawu w rowerze

Gdy mamy wszystkie potrzebne komponenty, możemy przystąpić do działania. Zaczynamy od tylnego koła z silnikiem, na początek zakładam opaskę na felgę, potem oponę z dętka i na koniec pompuję. Gdy mam całe koło, łatwiej jest zamontować wolnobieg (lub kasetę) oraz ewentualnie tarcze hamulcową. Gotowe koło wystarczy zamontować w rowerze, należy tylko pamiętać o podkładkach które są fabrycznie na silniku, zwykle są po dwie z każdej strony osi. Jedną jest zwykła okrągła podkładka, a drugą jest podkładka z tak zwanym „uchem” pokazana na **fotografii 4**. Jest dopasowana do kształtu osi silnika (otwór nie jest w całości okrągły) i teoretycznie powinna zabezpieczać przed obrotem osi. Musi być zamontowana od strony wewnętrznej haków, przy silniku, a samo ucho musi być skierowane w dół i na zewnątrz. Zwykła podkładka ma swoje miejsce pomiędzy nakrętką a ramą, pokazuje to **fotografia 5**.

Na tym kończy się montaż silnika. Czasami zdarza się, że koło nie jest na środku ramy (jest przesunięte w którąś stronę), przeważnie silniki są zaplatane na najpopularniejszy wzór, który pasuje do większości rowerów, ale zdarzają się wyjątki. Jeśli mamy do czynienia z takim przypadkiem to zwykle wystarczy dodać z którejś strony podkładkę dystansującą na oś silnika. Jeśli przesunięcie jest duże to może być wymagane przeplecenie obręczy.

Gdy silnik jest na miejscu można zamontować resztę komponentów. Będą potrzebne podstawowe narzędzia (klucze imbusowe, śrubokręty itp.), oraz dwa specjalistyczne, o ile zdecydujemy się na montaż czujnika pedałowania. Dostępne są dwa rodzaje czujników. Do montażu jednego z nich należy zdemonstrować jedną korbę i potrzebny byłby specjalny ściągacz. Drugi czujnik (popularniejszy, również ja taki zastosowałem) ma taką okrągłą blaszkę, którą umieszcza się pomiędzy miską suportu a ramą. Potrzebny jest wtedy klucz do suportu oraz ściągacz do korby. Zamontowany czujnik pokazuje **fotografia 6**.

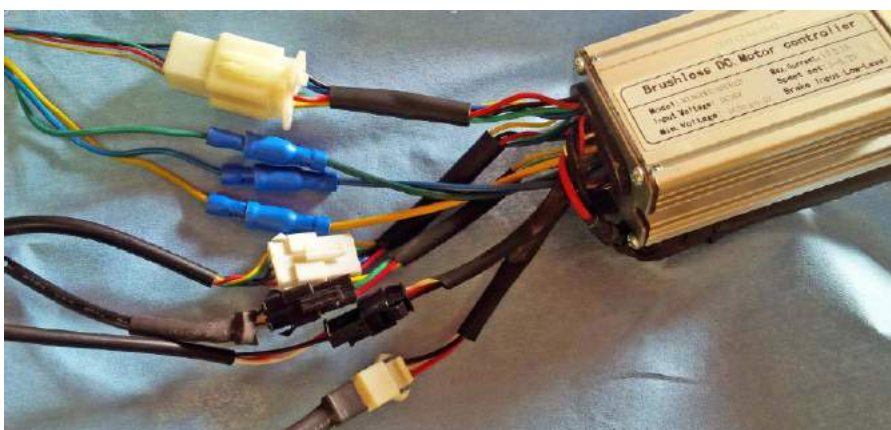
Pozostałe elementy są montowane na kierownicy i nie powinny przysporzyć większego problemu. Czasami trudno jest zmieścić wszystko w odpowiedniej pozycji, zdarzają się problemy z manetką przerzutki, ale można ją umieścić bliżej środka, ponieważ nie zmienia się biegów zbyt często jeśli mamy dodatkowy napęd.



Fotografia 5. Sposób zamocowania silnika



Fotografia 6. Zamontowany czujnik pedałowania



Fotografia 7. Połączenia w instalacji sterownika

Teraz trzeba podłączyć całą instalację. Jeżeli kompletowaliśmy cały zestaw, to wszystkie połączenia należy wykonać zgodnie z dołączonymi instrukcjami. Jeśli mamy gotowy zestaw to wystarczy połączyć odpowiednie złącza zgodnie z kolorami i rodzajami. Powinno to być dość proste i raczej trudno o popełnienie błędu, przykład połączeń pokazuje **fotografia 7**. Inne rozwiązanie, które jest preferowane, polega na nie stosowaniu złączek, tylko

polutowaniu wszystkich połączeń. Złącza, nawet wyglądające jak hermetyczne, po długotrwałym działaniu wstrząsów i wilgoci mogą utracić szczelność, a wtedy styki zaczną się utleniać i doprowadzi to do braku połączenia i dziwnych problemów z działaniem całości. Niezależnie od metody wszystkie przewody należy przypiąć do ramy opaskami zaciskowymi.

Miejsce montażu sterownika zależy od rodzaju roweru i miejsca zamontowania



Fotografia 8. Sterownik zamontowany pod bagażnikiem



Fotografia 9. Akumulator zamontowany w torbie na bagażniku

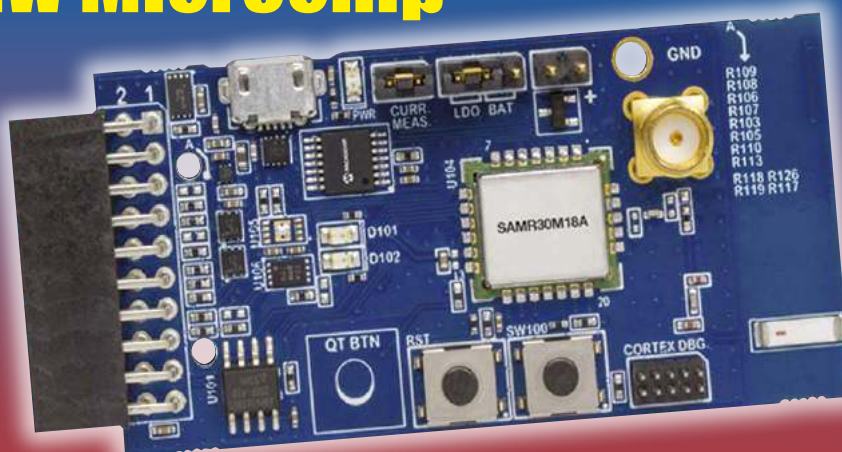
akumulatora. Jeśli mamy skrzynkę na cały trójkąt ramy to można w niej umieścić sterownik. Czasami jest w zestawie specjalny pojemnik na sterownik, który mocuje się w miejscu koszyka na bidon. Można też zamocować sterownik bez osłony (sterownik jest przeważnie wodoodporny jeśli jest zamontowany przewodami do dołu), ale psuje to estetykę i nie jest to zalecane rozwiązanie. Można również zamontować sterownik na bagażniku, jeżeli takowy posiadamy w rowerze. Niewielki sterownik zmieści się w przestrzeni pod bagażnikiem, wystarczy umieścić go w rękawie termokurczliwym i przypiąć opaskami, jak na **fotografii 8**.

Na koniec zamontuję akumulator, wybiorę opcję łatwiejszą i najmniej rzucającą się w oczy, czyli montaż w torbie na bagażniku. Sam wykonałem ze sklejk skrzynkę dla akumulatora. Jeśli ktoś chciałby wykonać taką skrzyneczkę ale nie ma odpowiedniego warsztatu, to może skorzystać z usługi cięcia na wymiar oferowanej w niektórych sklepach budowlano-remontowych. Ceny są naprawdę przystępne, koszt materiału to ok. 10...15 zł (sklejka o grubości 10 mm). Torbę mocuje zgodnie z instrukcją i wycinam dziurki na dnie torby, aby zamocować na sztywno akumulator przy pomocy białej opaski zaciskowej (**fotografia 9**). Ostatni etap polega na umieszczeniu akumulatora w torbie i przypięciu go opaską. Teraz czas wybrać się na jazdę testową.

Bartek Plonka
prototyp-mail@o2.pl

KONKURS

Wygraj zestaw Microchip SAM R30M Xplained Pro



Firma Microchip organizuje konkurs dla czytelników Elektroniki Praktycznej, w ramach którego do wygrania jest zestaw Microchip SAM R30M Xplained Pro. Platforma ta została zaprojektowana by testować możliwości układu SAMR30M18A. Zestaw SAM R30M Xplained Pro to najmniejszy na rynku moduł zgodny z IEEE 802.14.4, który łączy w sobie mikrokontroler

o ultra niskim poborze mocy oraz obwody radiowe pracujące na częstotliwościach poniżej 1 GHz. Zestaw jest wspierany przez środowisko Atmel Studio, która zawiera liczne przykładowe aplikacje oraz przez programator Atmel ICE. Aby wziąć udział w konkursie należy zarejestrować się pod adresem <http://bit.ly/2norxRb>.



Monitor połączenia z Internetem zbudowany na Raspberry Pi

Coraz trudniej jest nam wyobrazić sobie sytuację, w której nie mamy połączenia z Internetem. Gdy nie możemy połączyć się z jakąś stroną w Internecie zazwyczaj obwiniamy (nie bezzasadnie) zdalny serwer, ale co w momencie gdy to nasze łącze płata nam figła? Jeszcze większy problem mamy w przypadku zastosowania licznych urządzeń Internetu Rzeczy, czy systemów automatyki domowej, które uzależnione są od serwerów znajdujących się w Chmurze.

W poniższym artykule opisujemy prosty monitor połączenia z Internetem wykonany z użyciem komputera jednopłytkowego Raspberry Pi. System taki będzie przydatny wszystkim, którzy mają problem ze stabilnością łącza. Urządzenie zapewnia wizualny podgląd kondycji połączenia sieciowego w czasie rzeczywistym. Dzięki niemu, bez pomocy komputera, można sprawdzić stan połączenia z Internetem.

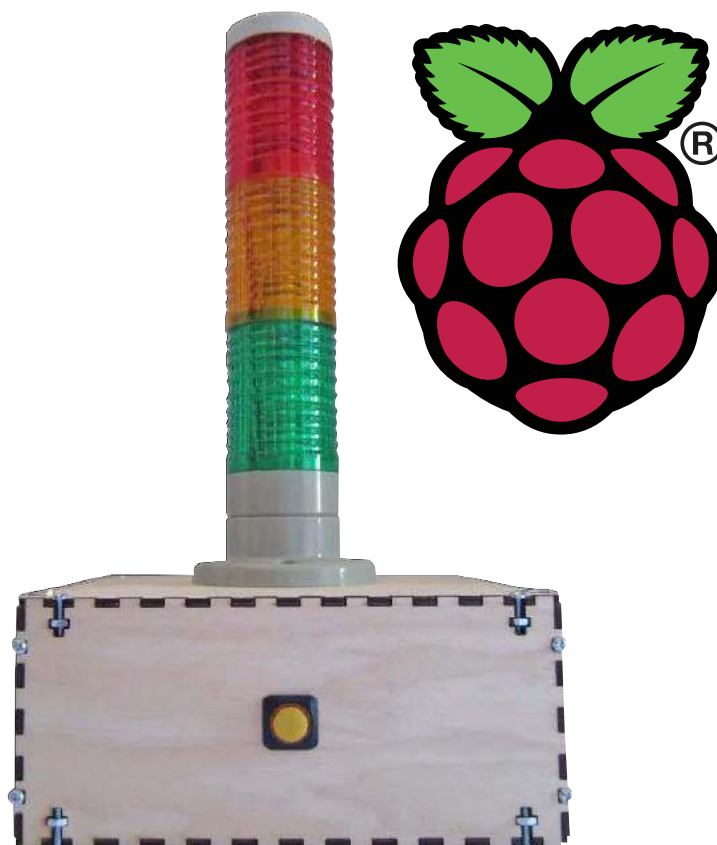
Zasada działania

W urządzeniu zastosowano przemysłowy sygnalizator w postaci trójkolorowej kolumny świetlnej. Pracą kolumny steruje komputer jednopłytkowy Raspberry Pi. Skrypt zainstalowany na minikomputerze pracującym pod kontrolą systemu operacyjnego Linuks, regularnie mierzy czas odpowiedzi szeregu stron internetowych z wykorzystaniem komendy *ping*. W momencie, gdy ponad 50%

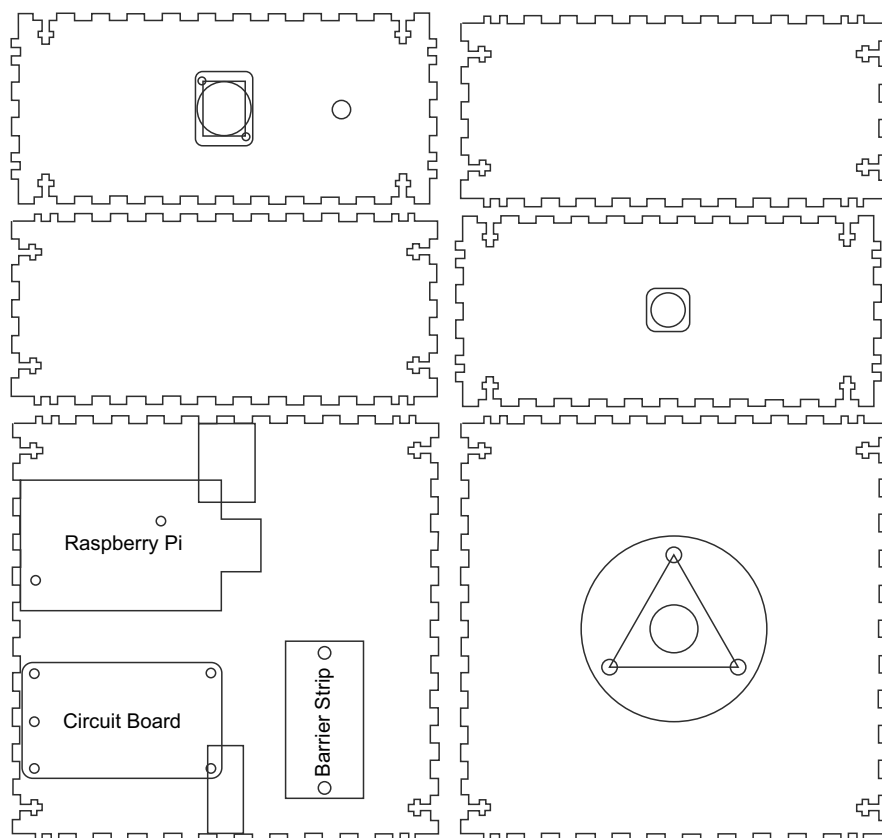
zapytań wysyłanych poprzez *ping* dochodzi do adresatów, system zapala zielone światło (**rysunek 1a**), w momencie, gdy odsetek poprawnych *ping*ów mieści się pomiędzy 1% a 50%, zapalony zostaje żółty sygnalizator (**rysunek 1b**), a jeżeli wszystkie *ping*i nie dochodzą do celu, to zapalony zostaje światło czerwone (**rysunek 1c**).

Potrzebne elementy

Do budowy opisanego urządzenia potrzebnych będzie kilka elementów. Przede wszystkim potrzebny jest komputer jednopłytkowy Raspberry Pi – w dowolnej wersji, autor wykorzystał pierwszą wersję tego minikomputera, model B (z 512 MB pamięci RAM). Dodatkowo do komputera potrzebna jest



Rysunek 1. Trzy kolory sygnalizatora, dla różnych odsetków udanych *ping*ów stron w Internecie: a) >50%, zielony, b) 1...50%, żółty, c) 0%, czerwony



Rysunek 2. Rozkład cięcia arkusza sklejk do wykonania elementów obudowy

karta SD – rekomendowana jest karta o pojemności co najmniej 8 GB oraz karta Wi-Fi w postaci modułu podłączanego na USB. Jeżeli komputer podłączany jest do sieci z pomocą połączenia kablowego, lub korzystamy

z modelu Raspberry Pi, które wyposażone jest w bezprzewodowy interfejs sieciowy, zewnętrzna karta Wi-Fi jest nam zbędna.

Potrzebny jest również sygnalizator. W tym projekcie najlepiej sprawdzi się

sygnalizator kolumnowy z diodami LED lub żarówkami na 12 V. Do jego sterowania potrzebny będzie również transoptor LTV847 lub inny, kompatybilny, umożliwiający sterowanie napięciem 12 V oraz trzy oporniki 18 Ω i jeden opornik 470 Ω . Potrzebny będzie też przycisk chwilowy, który będzie służył do sterowania pracą komputera jedнопłytkowego, aby możliwe było jego poprawne wyłączenie.

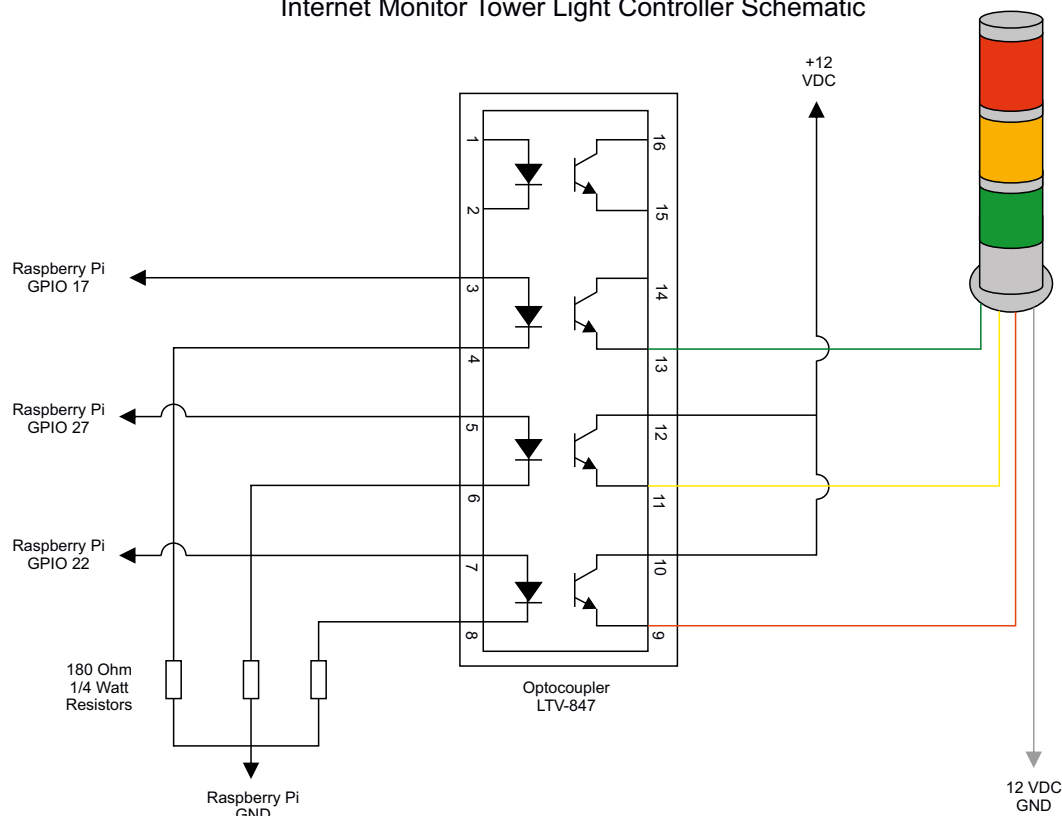
Oprócz powyższych elementów, do zestawienia opisywanego sygnalizatora potrzebne będą kable do połączeń wszystkich elementów, płytka uniwersalna oraz dwa zasilacze – 5 V z wtyczką microUSB, do zasilania Raspberry Pi oraz 12 V zakończony zwykłą wtyczką zasilaczową, do zasilania sygnalizatora kolumnowego.

Obudowa

Autor projektu włożył wiele wysiłku w zbudowanie estetycznej obudowy dla tego sygnalizatora. Obudowa wycięta została ze sklejki liściastej o grubości 3 mm. Na **rysunku 2** zaprezentowano rozkład cięcia na arkuszu sklejki o wymiarach ok. 40×40 cm. Dokładny projekt cięcia, dedykowany do wykorzystania z ploterem laserowym, pobrać można na stronie źródłowej.

Poszczególne elementy systemu przykręcone są do jego podstawy i osłonięte ściankami bocznymi obudowy, które łączą się z podstawą za pomocą ząbków. Sygnalizator kolumnowy zainstalowany jest na górnej płycie obudowy.

Internet Monitor Tower Light Controller Schematic



Rysunek 3. Schemat połączeń pomiędzy Raspberry Pi, a sterownikiem kolumny świetlnej

Schemat połączeń

Na **rysunku 3** zaprezentowano schemat połączeń pomiędzy Raspberry Pi, a sterownikiem kolumny świetlnej. Sygnalizator zasilany 12 V i Raspberry Pi są od siebie oddzielone galwanicznie poprzez transoptor. Pozwala to na zabezpieczenie komputera jednopłytkowego przed podwyższonym napięciem, a także umożliwia ewentualnie zasilanie oświetlenia w sygnalizatorze wyższym napięciem, np. 24 V.

Do komputera podłączony jest też pojedynczy przycisk, który służy do sterowania Raspberry Pi. Jego naciśnięcie powoduje zamknięcie systemu operacyjnego i wyłączenie komputera. Przykładowy przycisk, jaki wykorzystano w układzie, zaprezentowano na **rysunku 4**. Przycisk jest podświetlany, w związku z czym wykorzystuje dwie linie GPIO minikomputera. Jedna linia steruje diodą LED wbudowaną w przycisk zasilania – jest podłączona poprzez opornik 470 Ω do jednego z portów GPIO i do masy. Należy pamiętać o podłączeniu diody w kierunku przewodzenia. Przycisk podłączony jest do innej linii GPIO, naciśnięcie przycisku powoduje zwarcie linii do masy.

Oprogramowanie – skrypt do monitorowania połączenia

Za monitorowanie połączenia Internetowego i obsługę sygnalizatora kolumnowego odpowiedzialny jest skrypt `rpi-internet-monitor.py`, który znajdować będzie się w folderze `/home/pi/python_programs/`. Aby pobrać folder z repozytorium w internecie i umieścić go w tym folderze w terminalu wpisujemy: `wget https://s3-us-west-1.amazonaws.com/talk2bruce/instructables/rpi-internet-monitor/rpi-internet-monitor.py` `mv rpi-internet-monitor.py /home/pi/python_programs/rpi-internet-monitor.py`

Treść skryptu znajduje się w **listingu 1**. Działanie programu jest bardzo proste. W nieskończonej pętli `while` cyklicznie uruchamiana jest funkcja do realizacji pingów. Na podstawie jej wartości zwrotnej system zapala światło odpowiedniego koloru. Pingi realizowane są z wykorzystaniem systemowego narzędzia `ping`. Program konstruuje polecenie:

```
/bin/ping -c 1 nazwa_strony
```

Polecenie to zostaje uruchomione, jako osobny proces z wykorzystaniem biblioteki `subprocess`, która została wcześniej zaimportowana. Skrypt wychwytuje zwracaną informację z polecenia – w przypadku błędu, spowodowanego nieudanym pingiem, wychwytuje on błąd, po czym zwraca zero. W przypadku, gdy ping zakończy się poprawnie, funkcja ta zwraca jeden. Pozwala to na proste obliczanie odsetka udanych pingów w dalszej części programu.



Rysunek 4. Przycisk zastosowany w urządzeniu

```
Listing 1. Skrypt do monitorowania połączenia
#!/usr/bin/env python

import subprocess
import sys
import time
import RPi.GPIO as GPIO

GPIO_GREEN_LIGHT = 17 # linia IO podłączona do zielonego światła
GPIO_AMBER_LIGHT = 27 # linia IO podłączona do żółtego światła
GPIO_RED_LIGHT = 22 # linia IO podłączona do czerwonego światła

DELAY_BETWEEN_PINGS = 1 # opóźnienie (w sekundach) pomiędzy pingami
DELAY_BETWEEN_TESTS = 120 # opóźnienie (w sekundach) pomiędzy testami

SITES = ["google.com", "comcast.com"] # lista domen do testowania

# drukowanie na ekranie wiadomości debugowych, jeżeli opcja jest włączona
def debug_message(debug_indicator, output_message):
    if debug_indicator:
        print output_message

# wykorzystanie komendy ping z linii poleceń systemu operacyjnego
def ping(site):
    cmd = "/bin/ping -c 1 " + site
    try:
        output = subprocess.check_output(cmd,
            stderr=subprocess.STDOUT, shell=True)
    except subprocess.CalledProcessError, e:
        debug_message(debug, site + ": nieosiągalny")
        return 0
    else:
        debug_message(debug, site + ": osiągalny")
        return 1

# funkcja realizująca ping do listy stron i
# obliczająca odsetek udanych pingów
def ping_sites(site_list, wait_time, times):
    successful_pings = 0
    attempted_pings = times * len(site_list)
    for t in range(0, times):
        for s in site_list:
            successful_pings += ping(s)
            time.sleep(wait_time)
    debug_message(debug, "Odsetek udanych pingów: " +
        str(int(100 * (successful_pings / float(attempted_pings)))) + "%")
    # funkcja zwraca odsetek udanych pingów
    return successful_pings / float(attempted_pings)

# włączenie żółtego światła
def lamp_amber_on():
    debug_message(debug, ">>> Czerwone OFF; Żółte ON; Zielone OFF")
    GPIO.output(GPIO_RED_LIGHT, False)
    GPIO.output(GPIO_AMBER_LIGHT, True)
    GPIO.output(GPIO_GREEN_LIGHT, False)

# włączenie zielonego światła
def lamp_green_on():
    debug_message(debug, ">>> Czerwone OFF; Żółte OFF; Zielone ON")
    GPIO.output(GPIO_RED_LIGHT, False)
    GPIO.output(GPIO_AMBER_LIGHT, False)
    GPIO.output(GPIO_GREEN_LIGHT, True)

# włączenie czerwonego światła
def lamp_red_on():
    debug_message(debug, ">>> Czerwone ON; Żółte OFF; Zielone OFF")
    GPIO.output(GPIO_RED_LIGHT, True)
    GPIO.output(GPIO_AMBER_LIGHT, False)
    GPIO.output(GPIO_GREEN_LIGHT, False)

# wyłączenie wszystkich świateł
def lamp_all_off():
    debug_message(debug, ">>> Czerwone OFF; Żółte OFF; Zielone OFF")
```

Oprogramowanie – skrypt do obsługi przycisku zasilania

Drugim istotnym skryptem, jest moduł obsługujący przycisk zasilania. Ten prosty skrypt przedstawiono na **listingu 2**. Aby pobrać i przenieść skrypt do tego samego folderu, co w przypadku pierwszego prezentowanego tutaj modułu, w linii komend wpisujemy:

```
wget https://s3-us-west-1.amazonaws.com/talk2bruce/instructables/rpi-internet-monitor/rpi-halt-btn.py
mv rpi-halt-btn.py /home/pi/python_programs/
```

Skrypt wysyła do systemu operacyjnego polecenie *halt*, które rozpoczyna zamykanie systemu operacyjnego. Jest poprzedzone poleceniem *sudo*, które oznacza, że komenda zatrzymania systemu wydawana jest z prawami roota (administratora systemu operacyjnego). Jednakże, aby do tego doszło, układ musi wykryć na linii GPIO podłączonej do przycisku, zbocze opadające. Oczekiwanie można przezwyciężać, nie wyłączając komputera, poprzez naciśnięcie dowolnego przycisku na klawiaturze.

Konfiguracja systemu operacyjnego do pracy automatycznej

Ten krok polega na skonfigurowaniu systemu do automatycznego uruchamiania skryptów, do monitorowania połączenia Internetowego i obsługi przycisku zasilania. Skrypty muszą zostać uruchomione automatycznie, po uruchomieniu systemu operacyjnego.

Aby dodać skrypt do obsługi przycisku zasilania do automatycznego uruchamiania przy starcie systemu, musimy edytować plik *rc.local*. W tym celu w terminalu wpisujemy: `sudo nano /etc/rc.local` i dodajemy poniższe polecenie na samym dole pliku przed wierszem, w którym znajduje się wpis „exit 0”.

```
python /home/pi/python_programs/rpi-halt-btn.py
```

Aby skonfigurować system do uruchamiania programu do monitorowania sieci, musimy uruchomić skrypt Pythona, gdy połączenie Wi-Fi jest uruchomione na Raspberry Pi. W tym celu musimy dodać wpis w pliku */etc/network/interfaces*. Plik edytujemy za pomocą *nano*, wpisując w terminalu:

```
sudo nano / etc / network / interfaces
```

i na końcu pliku dodajemy polecenie:

```
post-up python /home/pi/python_programs/rpi-internet-monitor.py
```

Spowoduje to uruchomienie się skryptu po połączeniu komputera z siecią Wi-Fi. Te dwa wpisy kończą konfigurację komputera jednopłytkowego do pracy jako monitor połączenia sieciowego.

Teraz możemy wyłączyć Raspberry Pi za pomocą polecenia:

```
sudo halt
```

Listing 1. cd.

```
GPIO.output(GPIO_RED_LIGHT, False)
GPIO.output(GPIO_AMBER_LIGHT, False)
GPIO.output(GPIO_GREEN_LIGHT, False)

# test lamp - mruganie lampami w sekwencji, pięciokrotnie
def lamp_test():
    debug_message(debug, "Testowanie lamp")
    # jedna dziesiąta sekundy opóźnienia w testach
    TEST_DELAY = 0.1
    for i in range(0, 5):
        time.sleep(TEST_DELAY)
        lamp_red_on()
        time.sleep(TEST_DELAY)
        lamp_amber_on()
        time.sleep(TEST_DELAY)
        lamp_green_on()
    lamp_all_off()
    debug_message(debug, "Testowanie lamp zakończone")

# Tutaj zaczyna się główna część programu

# sprawdzamy czy użytkownik chce wypisywać informacje debugowe
debug = False
if len(sys.argv) > 1:
    if sys.argv[1] == "-debug":
        debug = True
    else:
        print "nieznane polecenie: " + sys.argv[1]
        sys.exit(1)

# konfiguracja pinów GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setup(GPIO_GREEN_LIGHT, GPIO.OUT)
GPIO.setup(GPIO_AMBER_LIGHT, GPIO.OUT)
GPIO.setup(GPIO_RED_LIGHT, GPIO.OUT)

# mrugnięcie lampami, sygnalizujące, że program uruchamia się
lamp_all_off()
lamp_test()
time.sleep(0.5)
# po zakończeniu testów, zapala się żółta lampka
lamp_amber_on()

# główna pętla programu: uruchamianie pingów,
# zapalanie lampki, opóźnienie i powtarzanie w pętli
test = 0
while True:
    test+=1
    debug_message(debug, "----- Test " + str(test) + " -----")
    success = ping_sites(SITES, DELAY_BETWEEN_PINGS, 2)
    if success == 0:
        lamp_red_on()
    elif success <= .50:
        lamp_amber_on()
    else:
        lamp_green_on()

    debug_message(debug, "Oczekiwanie " +
        str(DELAY_BETWEEN_TESTS) + " sekund do kolejnego testu.")

    time.sleep(DELAY_BETWEEN_TESTS)
```

Po wyłączeniu minikomputera należy odłączyć wszystkie kable, które dodaliśmy (monitor, klawiatura, mysz etc) zostawiając w środku kartę SD. Możemy teraz zamontować Raspberry

Pi w urządzeniu i podłączyć do niego sygnalizator kolumnowy i inne elementy.

Nikodem Czechowski

<http://bit.ly/2kIdg0L>

Listing 2. Skrypt do obsługi przycisku zasilania

```
#!/usr/bin/env python

import RPi.GPIO as GPIO
import subprocess, time

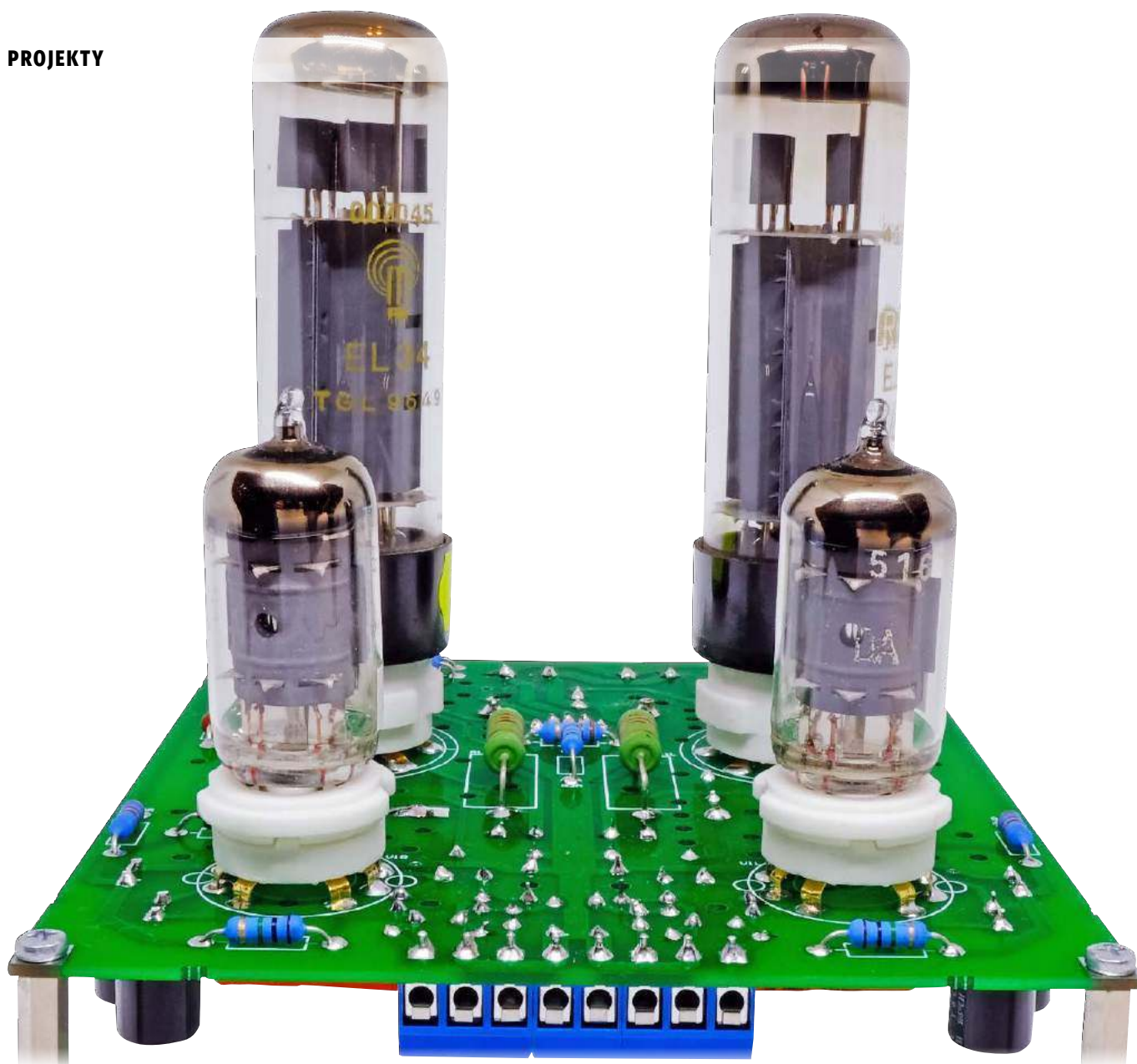
# linia GPIO podłączona do przycisku
HALT_SWITCH_GPIO_PIN = 24
# linia GPIO podłączona do diody sygnalizacyjnej w przycisku
HALT_SWITCH_LED_GPIO_PIN = 23

# konfiguracja linii GPIO
GPIO.setmode(GPIO.BCM)
# ustawienie linii GPIO diody LED jako wyjścia
GPIO.setup(HALT_SWITCH_LED_GPIO_PIN, GPIO.OUT)
# ustawienie linii GPIO przycisku jako wejścia z podciągnięciem do góry
GPIO.setup(HALT_SWITCH_GPIO_PIN, GPIO.IN, pull_up_down=GPIO.PUD_UP)
# ustawienie linii GPIO diody LED w stanie wysokim - zapalenie diody
GPIO.output(HALT_SWITCH_LED_GPIO_PIN, True)

print("rpi-halt-btn: uruchomiono, trwa oczekiwanie na naciśnięcie przycisku")

# oczekiwanie na naciśnięcie przycisku
# gdy zostanie on naciśnięty, skrypt wysyła
# do systemu operacyjnego polecenie zatrzymania
try:
    # oczekiwanie na zbocze opadające na linii GPIO przycisku
    GPIO.wait_for_edge(HALT_SWITCH_GPIO_PIN, GPIO.FALLING)
except KeyboardInterrupt:
    print("Zatrzymano przez użytkownika")
    GPIO.cleanup()

print("rpi-internet-monitor: Naciśnięto przycisk - trwa wyłączanie systemu")
GPIO.output(HALT_SWITCH_LED_GPIO_PIN, False)
time.sleep(0.5)
GPIO.output(HALT_SWITCH_LED_GPIO_PIN, True)
# wysłanie do systemu komendy sudo, a następnie halt
subprocess.call(["sudo", "halt"])
```



Wzmacniacz lampowy na lampach EL34

Wzmacniacze lampowe nieprzerwanie cieszą się dużą popularnością w kręgach DiY. Opisana konstrukcja nie spodoba się audiofilom, gdyż w torze audio zawiera tylko pentody i to na dodatek pracujące, a jakże w trybie pentodowym – czasami warto zrobić coś wbrew oklepanym standardom. Pozostałe osoby, które po prostu słuchają muzyki i lubią zbudować coś samodzielnie zachęcam do eksperymentu. W konstrukcji zastosowano łatwo dostępne typowe lampy EF86 i EL34 oraz ogólnie dostępne typowe transformatory.

Zasada działania

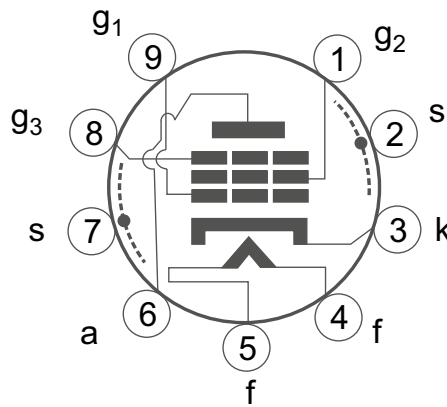
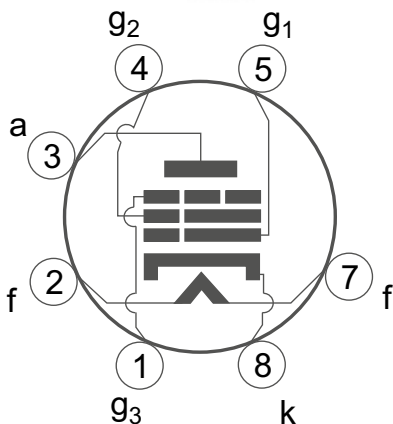
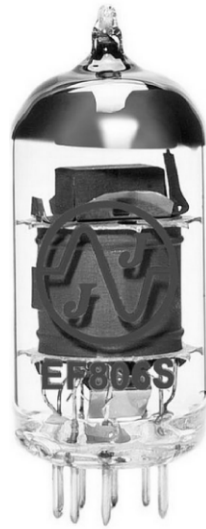
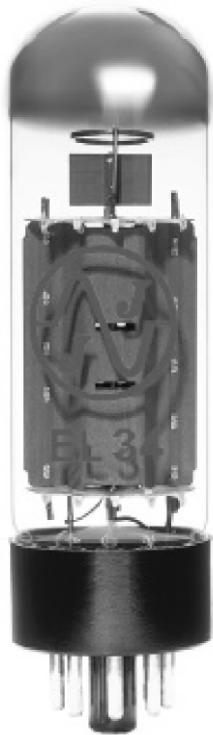
Wzmacniacz jest zbudowany w topologii SE (Single-Ended – z pojedynczym elementem wzmacniającym. Niestety skrót

SE nie doczekał się sensownego polskiego tłumaczenia).

Podstawowe założenia, które miała spełniać konstrukcja to osiągnięcie mocy

ok. 7...9 W na kanał, czyli większej niż w popularnym układzie EL84SE, oraz zachowanie prostej konstrukcji i zastosowanie łatwo dostępnych elementów. W każdym kanale pracują po dwie lampy. W stopniu wzmocnienia napięciowego użyto niskoszumnej pentody małej mocy EF86 lub EF806 (rosyjski zamiennik 6Ж32P, cyr. 6Ж32П), a w stopniu mocy pentody EL34. Przykładowy wygląd oraz układ wyprowadzeń lamp przedstawia **rysunek 1**.

Schemat ideowy wzmacniacza pokazuje **rysunek 2**. Elementy kanału lewego mają oznaczenia zakończone literą L, kanału prawego literą R, zaś elementy zasilacza



Rysunek 1. Lampy EL34 i EF86(EF806) wygląd i schemat wyprowadzeń

anodowego wyróżnione są literką A, a obwód żarzenia literką H. Sygnał wejściowy ze złącza J1L/R, po separacji składowej stałej w C1L podany jest na siatkę pentody V1L. Stopień wzmocnienia napięciowego jest klasyczny, niskoszumna pentoda zapewnia odpowiedni poziom sygnału sterującego i pozwala wygospodarować zapas wzmocnienia na obciążenie wzmacniacza globalną pętlą sprzężenia zwrotnego przez R6L, R5L, C2L. Ujemne sprzężenie pozwala wyrównać pasmo przenoszenia i zmniejszyć zniekształcenia wzmacniacza. Wzmocniony wstępnie sygnał doprowadzony jest do lampy mocy V2L, pracującej z polaryzacją automatyczną zapewnianą przez R11L, CE4L. Z anody lampy poprzez złącze J3L sygnał podany jest na transformator wyjściowy. W modelu zastosowano transformator LOSE25-9 $R_a=2,5\text{ k}\Omega$, możliwe jest zastosowanie innych typów $R_a=2,5\text{--}3,0\text{ k}\Omega$ po niewielkich doświadczalnych korektach R6L, C2R.

Zasilacz napięcia anodowego UA jest wspólny dla obu kanałów i składa się

z mostka prostowniczego na szybkich diodach D1...D4 UF4007, kondensatorów filtrujących CE1A/CE2A oraz obwodu filtracji aktywnej opartej na tranzystorze Q1A typu P6NK60ZFP, kondensator CE3A zapewnia „łagodny” start zasilacza anodowego, eliminując konieczność opóźnianego załączania napięcia anodowego. Ze względu na wydzielane ciepło tranzystor Q1 zamontowany jest na radiatorze. Kondensator CE4A zapewnia dodatkową filtrację zasilania, stopnie wstępne zasilane są przez własne filtry odprężające R8L/C1L. Żarzenie lamp wykorzystuje napięcie przemienne, rezystory R1, 2H ustalają potencjał grzejników względem masy, obwód R3H, D1H zasilają diodę LED sygnalizującą pracę wzmacniacza. Napięcie anodowe (240 VAC/0,3 A) z transformatora doprowadzone jest do złącza JA1, napięcie żarzenia (6,3 VAC/4 A) do złącza JH1.

Montaż wzmacniacza

Wzmacniacz wykonany jest na dwustronnym obwodzie drukowanym, zawierającym

Dodatkowe materiały do pobrania ze strony www.media.avt.pl

W ofercie AVT* AVT-5718

Podstawowe parametry:

- moc wyjściowa ok. 7...9 W/8 Ω
- bazuje na lampach EL34 i EL86
- zbudowany w topologii Single-Ended

Projekty pokrewne na www.media.avt.pl:

- AAVT-5642 Lampowy regulator barwy dźwięku (EP 9/2018)
- AVT-5634 Lampowy przedwzmacniacz gramofonowy (EP 8/2018)
- AVT-5537 Sterownik wzmacniacza lampowego (EP 5/2016)
- AVT-5493 Stereofoniczny przedwzmacniacz lampowy zasilany niskim napięciem (EP 3/2015)

* Uwaga! Elektroniczne zestawy do samodzielnego montażu.

Wymagana umiejętność lutowania!

Podstawową wersją zestawu jest wersja [B] nazywana potocznie KIT-em (z ang. zestaw). Zestaw w wersji [B] zawiera elementy elektroniczne (w tym [UK] – jeśli występuje w projekcie), które należy samodzielnie wlutować w dołączoną płytkę drukowaną (PCB). Wykaz elementów znajduje się w dokumentacji, która jest podłączona w opisie kitu.

Mając na uwadze różne potrzeby naszych klientów, oferujemy dodatkowe wersje:

- wersja [C] – zmontowany, uruchomiony i przetestowany zestaw [B] (elementy wlutowane w płytkę PCB)
 - wersja [A] – płytkę drukowaną bez elementów i dokumentacji Kity w których występuje układ scalony wymagający zaprogramowania, mają następujące dodatkowe wersje:
 - wersja [A+] – płytkę drukowaną [A] + zaprogramowany układ [UK] i dokumentacja
 - wersja [UK] – zaprogramowany układ
- Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik pdf! Podczas składania zamówienia upewnij się, którą wersję zamawiasz!
- <http://sklep.avt.pl>, osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: kity@avt.pl.

Wykaz elementów:

Rezystory:

- R10L, R10R, R2L, R2R, R4L, R4R: 1,8 k Ω
- R6L, R6R: 5,6 k Ω
- R11L, R11R: 120 $\Omega/5\text{ W}$
- R12L, R12R: 470 $\Omega/0,5\text{ W}$
- R1A, R3H: 2,7 k Ω
- R1H, R2H: 47 $\Omega/0,5\text{ W}$
- R1L, R1R, R9L, R9R: 470 k Ω
- R2A: 33 k $\Omega/2\text{ W}$
- R3A, R7L, R7R: 1 M $\Omega/2\text{ W}$
- R3L, R3R: 220 k $\Omega/1\text{ W}$
- R5L, R5R: 470 Ω
- R8L, R8R: 33 k $\Omega/1\text{ W}$

Kondensatory:

- C1L, C1R: 1 $\mu\text{F}/50\text{ V}$ foliowy 5 mm
- C2L, C2R: 1 nF/50 V foliowy 5 mm
- C3L, C3R: 220 nF/400 V foliowy 25/30 mm
- CE1A, CE2A, CE3A: 100 $\mu\text{F}/400\text{ V}$ elektrolityczny
- CE1L, CE1R, CE3L, CE3R, CE4A: 10 $\mu\text{F}/400\text{ V}$ elektrolityczny
- CE2L, CE2R: 220 $\mu\text{F}/10\text{ V}$ elektrolityczny
- CE4L, CE4R: 220 $\mu\text{F}/35\text{ V}$ elektrolityczny

Półprzewodniki:

- D1A, D1H, D2A, D3A, D4A, D5A: UF4007
- Q1A: P6NK60

Inne:

- V1L, V1R: EF86 + podstawka
- V2L, V2R: EL34 + podstawka
- J1H, J3L, J3R, JA1: ARK2/300
- J1L, J1R, J2L, J2R: ARK2/500
- J2H: Złącze KK proste 2 piny
- HS: Radiator HS142

REKLAMA

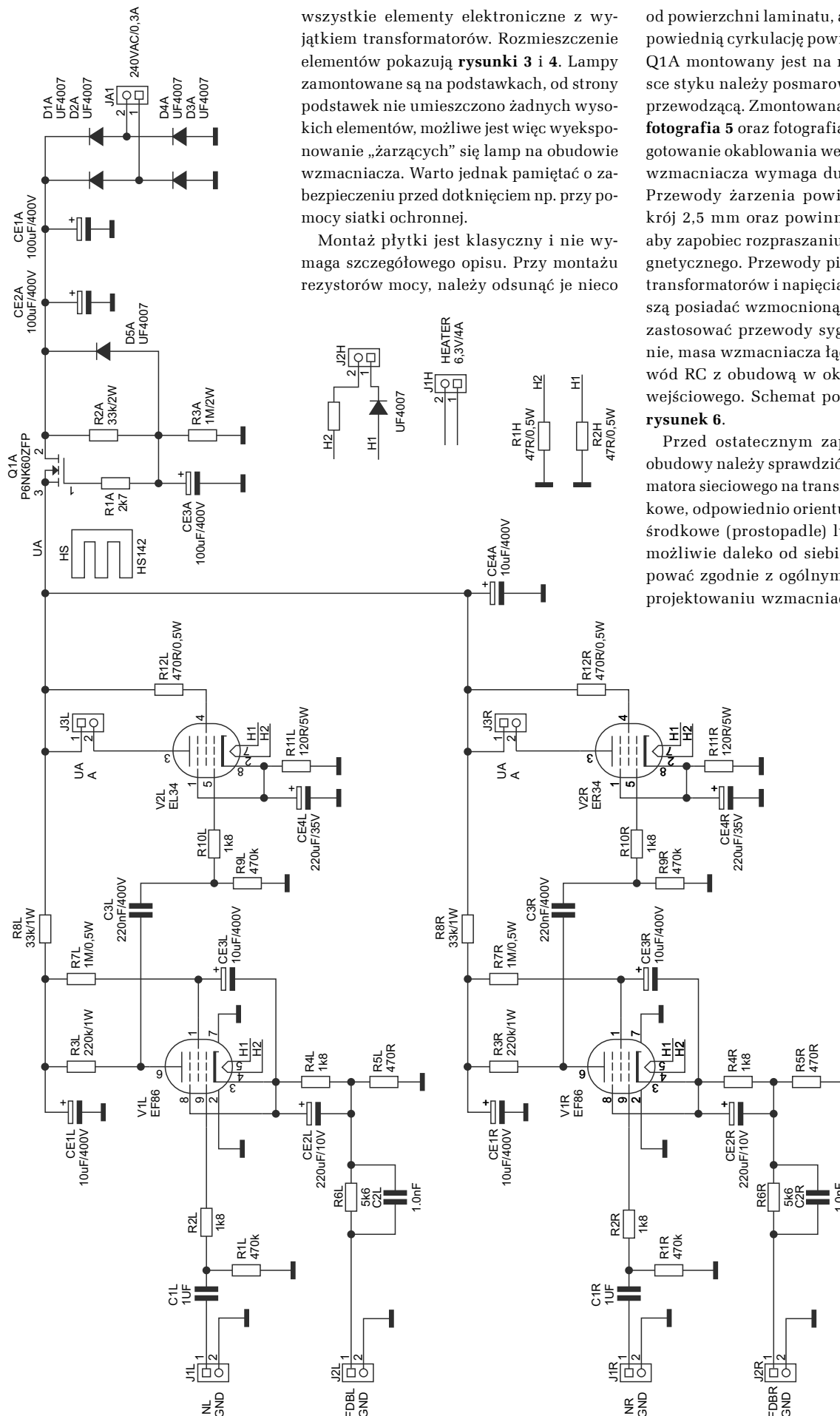
Specjalistyczne szkolenia dla elektroników i automatyków



TECHDAYS

techdays@techdays.pl
TECHDAYS.PL

CERTYFIKOWANY PARTNER SZKOLENIOWY



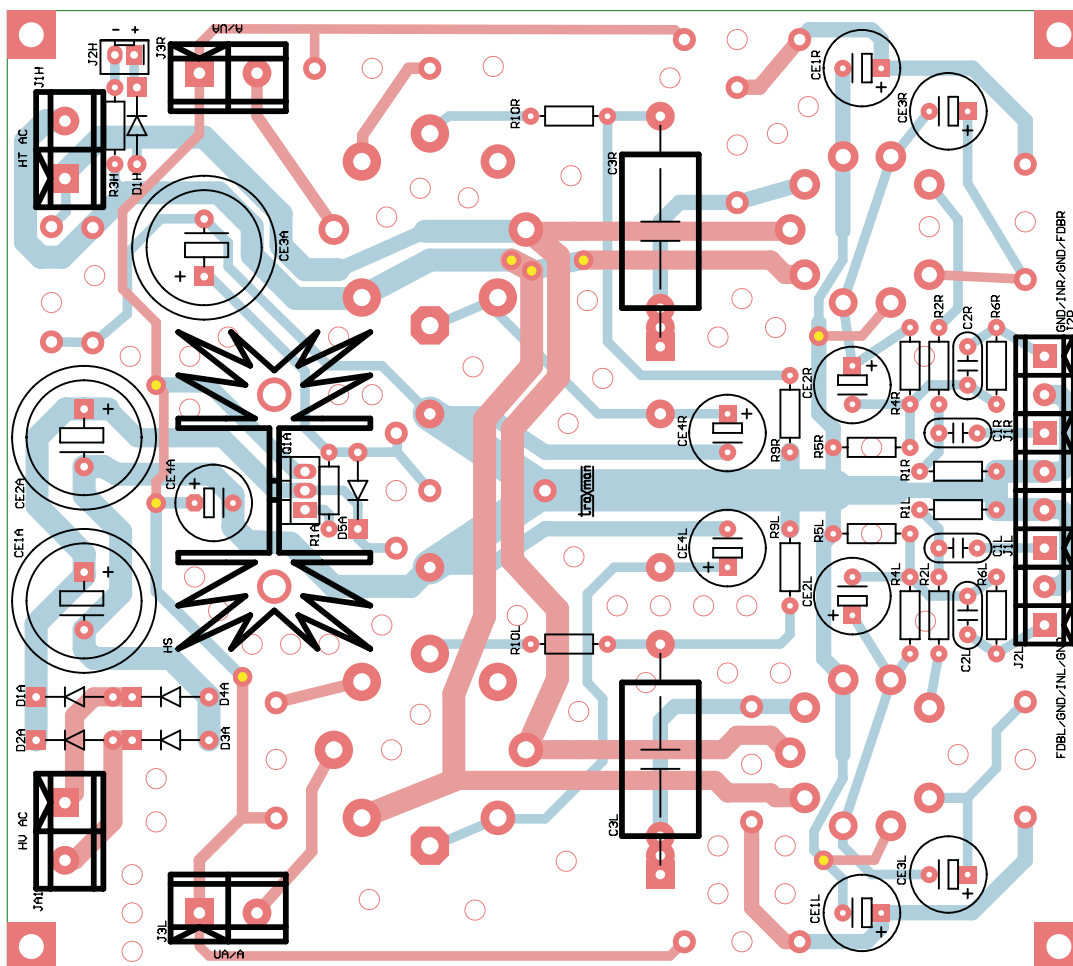
Rysunek 2. Schemat wzmacniacza

wszystkie elementy elektroniczne z wyjątkiem transformatorów. Rozmieszczenie elementów pokazują **rysunki 3 i 4**. Lampy zamontowane są na podstawkach, od strony podstawek nie umieszczono żadnych wysokich elementów, możliwe jest więc wyeksponowanie „żarzących” się lamp na obudowie wzmacniacza. Warto jednak pamiętać o zabezpieczeniu przed dotknięciem np. przy pomocy siatki ochronnej.

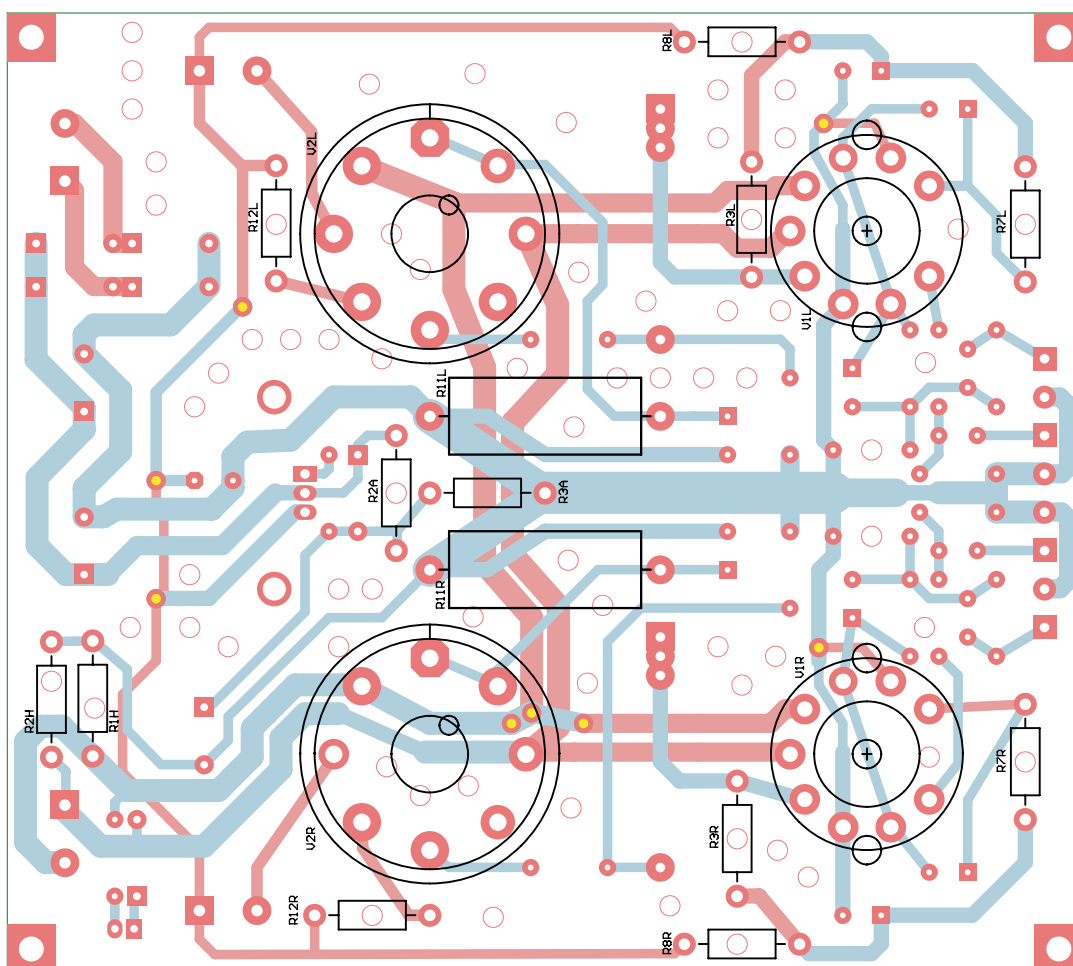
Montaż płytki jest klasyczny i nie wymaga szczegółowego opisu. Przy montażu rezystorów mocy, należy odsunąć je nieco

od powierzchni laminatu, aby zapewnić odpowiednią cyrkulację powietrza, tranzystor Q1A montowany jest na radiatorze, miejsce styku należy posmarować pastą termoprzewodzącą. Zmontowaną płytkę pokazuje **fotografia 5** oraz fotografia tytułowa. Przygotowanie okablowania wewnątrz obudowy wzmacniacza wymaga dużej staranności. Przewody żarzenia powinny mieć przekrój 2,5 mm oraz powinny być skręcone, aby zapobiec rozpraszaniu pola elektromagnetycznego. Przewody pierwotnych stron transformatorów i napięcia anodowego muszą posiadać wzmocnioną izolację. Należy zastosować przewody sygnałowe w ekranie, masa wzmacniacza łączy się przed obwód RC z obudową w okolicach gniazda wejściowego. Schemat połączeń pokazuje **rysunek 6**.

Przed ostatecznym zaprojektowaniem obudowy należy sprawdzić wpływ transformatora sieciowego na transformatory głośnikowe, odpowiednio orientując ich kolumny środkowe (prostopadle) lub odsuwając je możliwie daleko od siebie. Należy postępować zgodnie z ogólnymi regułami przy projektowaniu wzmacniaczy lampowych.



Rysunek 3. Schemat płytki PCB i rozmieszczenie elementów, strona TOP



Rysunek 4. Schemat płytki PCB i rozmieszczenie elementów, strona BOTTOM



Fotografia 5. Zmontowana płytki

Szczególną uwagę należy zwrócić na dokładność napięcia żarzenia, przy niedokładnie wykonanym uzwojeniu zdarza się, że napięcie jest zbyt wysokie (nawet ponad 7 V), co skutkuje przeżarzeniem lamp, zwiększeniem poziomu przydźwięku pochodzącego od upływu grzejnika katody i oczywiście skróceniem trwałości lamp. W takim przypadku należy włączyć w szeregu z uzwojeniem żarzenia dwa małe

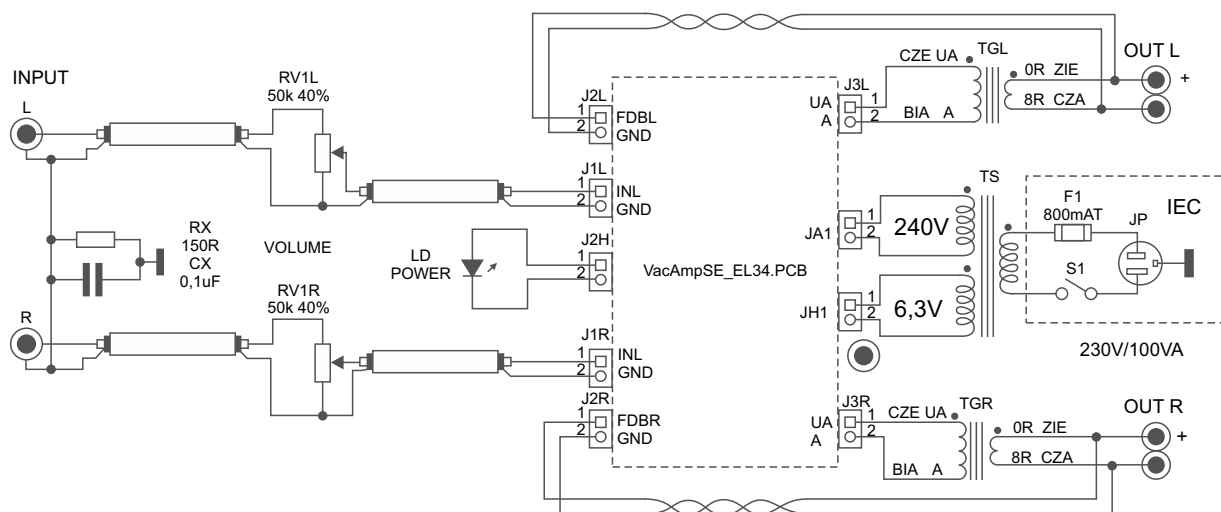
jednakowe rezystory o odpowiedniej mocy korygujące napięcie. Najlepiej jednak zaopatrzyć się w dobrej jakości transformator sieciowy.

Uruchomienie

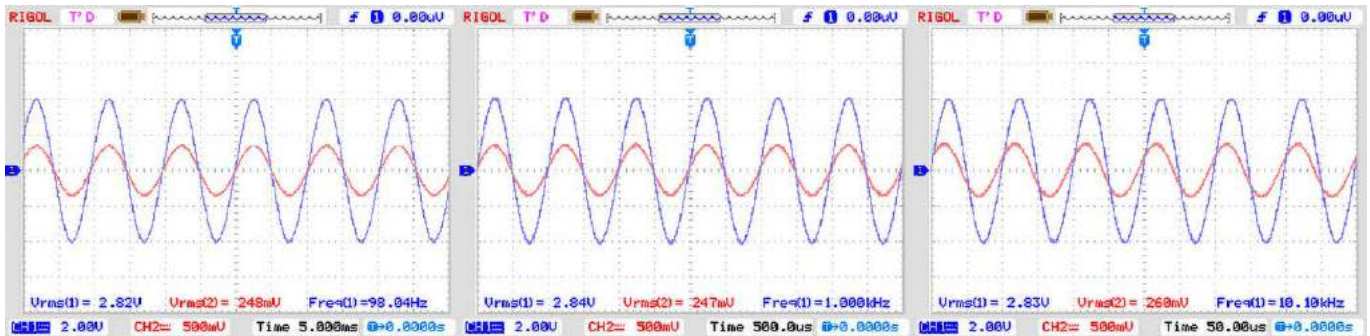
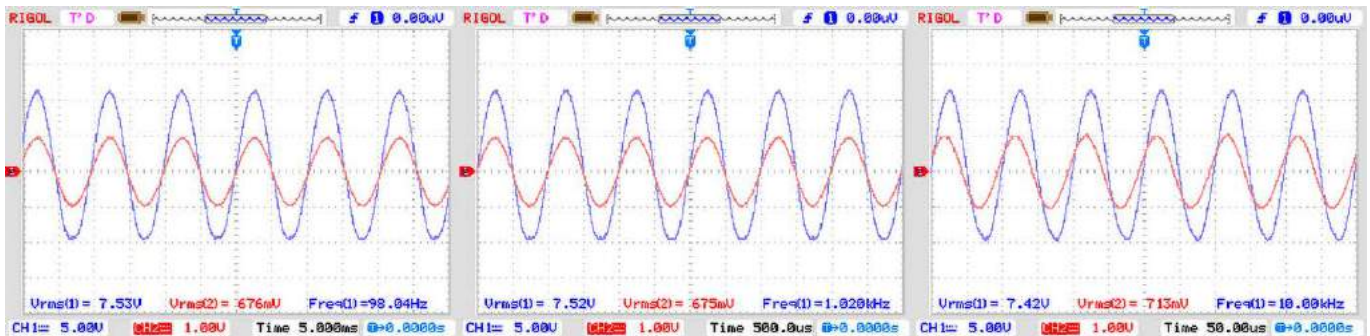
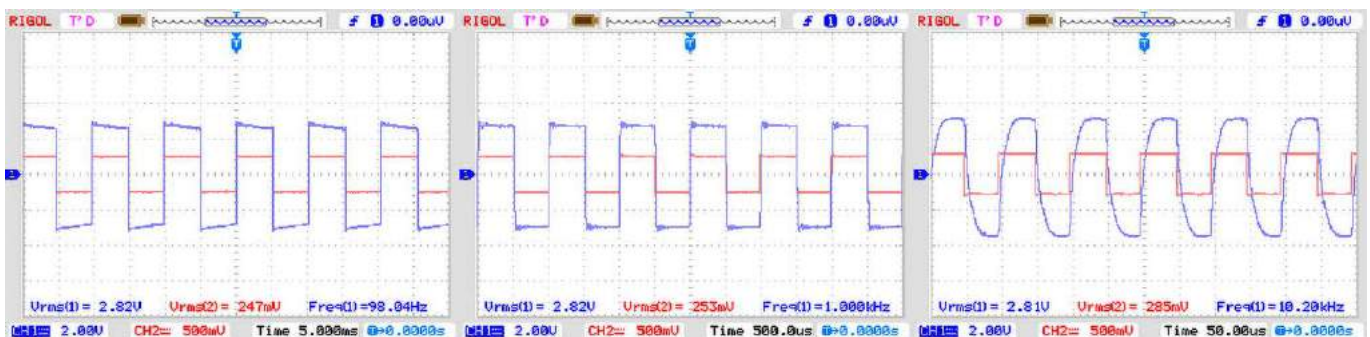
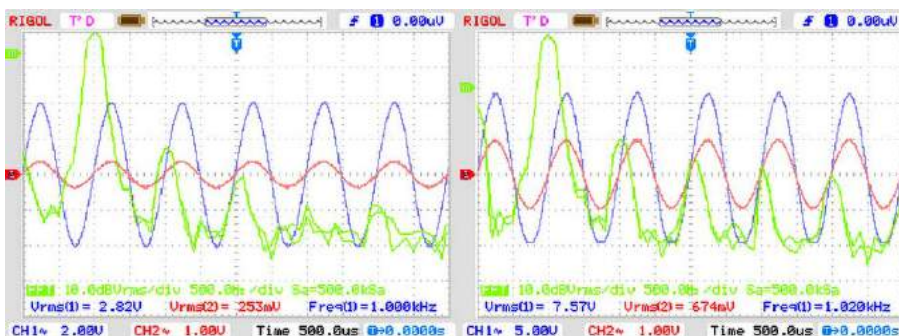
Uruchomienie wzmacniacza, po kontroli wizualnej montażu i upewnieniu się o braku zwarcie i prawidłowym rozmieszczeniu elementów, odbywa się etapami.

Po obciążeniu transformatorów wyjściowych rezystorem $8\ \Omega/20\ W$ i podłączeniu do wyjść oscyloskopu, a do wejść generatora funkcji (minimalny poziom sygnału), należy w pierwszej kolejności podać napięcie żarzenia. Jego wartość powinna wynosić $6,3\ V \pm 5\%$ przy pobieranym prądzie $3,5\ A \pm 5\%$. W drugiej kolejności z zasilacza regulowanego lub autotransformatora należy podać napięcie anodowe, przy wartości ok. 50 V należy sprawdzić czy wzmacniacz ma poprawnie podpięte sprzężenie zwrotne, jeżeli układ wzbudzi się należy zamienić połączenie jednego z uzwojeń transformatora głośnikowego, tak aby sprzężenie zwrotne zmniejszało wzmocnienie. Jeżeli wzmacniacz jest stabilny, można dalej zwiększać napięcie anodowe. Przy wartości 240 VAC należy skontrolować spadek napięcia na rezystorach katodowych R11L/R lamp EL34, który powinien wynosić ok. 13 V. Wartość napięcia anodowego powinna być zbliżona do 300 V, napięcia zasilającego stopień wstępny na kondensatorach CE1L/R, ok. 260 V, na katodach EF86 ok. 2,2 V. Napięcia podano z dokładnością 5%.

Jeżeli wzmacniacz został uruchomiony statycznie to nadszedł czas na bardziej szczegółowe pomiary. Po zwiększeniuysterowania, warto sprawdzić przebiegi wyjściowe dla sterowania przebiegiem sinusoidalnym przy mocy 1 W/7 W dla kilku częstotliwości np. 100 Hz/1 kHz/10 kHz. Dla mniejszej mocy warto ocenić przenoszenie przebiegu prostokątnego, na jego podstawie należy skorygować wartości R6L/R, C2L/R, aby otrzymać prawidłowe odtworzenie przebiegu niezależnie od zastosowanego transformatora wyjściowego. Przebiegi otrzymane podczas pomiarów modelu pokazują rysunki 7a, 7b, 7c, 7d. W modelu, przy mocy 7 W, zniekształcenia THD+N utrzymują się poniżej 4% w paśmie 100 Hz/10 kHz. Pasmo przeniesienia wynosi 20 Hz...20 kHz przy obciążeniu znamionowym i mocy 7 W. Otrzymane



Rysunek 6. Schemat połączeń płytki z elementami całego wzmacniacza

Rysunek 7a. Pomiary dla mocy 1 W/8 Ω Rysunek 7b. Pomiary dla mocy 7 W/8 Ω Rysunek 7c. Pomiary dla mocy 1 W/8 Ω przebieg prostokątny

Rysunek 7d. Orientacyjny rozkład harmonicznych dla 1 W i 7 W

wyniki mogą być różne w zależności od zastosowanych transformatorów i stanu lamp. Jeżeli wszystko przebiegło pomyślnie, można przełączyć się na sygnał audio, podłączyć głośniki i cieszyć się muzyką, nie naprężając się otoczeniu...

UWAGA! We wzmacniaczu występują wysokie napięcia oraz wysoka temperatura, niebezpieczne dla zdrowia i życia. Uruchamianie należy przeprowadzić z zachowaniem szczególnej ostrożności.

Adam Tatuś, EP

REKLAMA

już w październiku nowa odsłona strony
www.ep.com.pl

microStation – mała stacja pogodowa

Pogoda potrafi być bardzo kapryśna. Nie mając gwarancji dobrej pogody trudno planować dłuższe eskapady, ale jest pewien gadżet, który powinien poprawić nieco nasze szanse na dobrą zabawę na świeżym powietrzu. Mowa o podręcznej stacji pogodowej, która pokaże bieżący stan pogody oraz poinformuje o prognozie na podstawie bieżących parametrów pogodowych.

Schemat ideowy urządzenia nazwanego, z uwagi na miniaturowe wymiary, microStation, pokazano na **rysunku 1**. Jest to bardzo prosty system mikroprocesorowy, którego „sercem” jest niewielki mikrokontroler firmy Microchip (dawniej Atmel) typu ATtiny84 realizujący całą, założoną funkcjonalność urządzenia. Mikrokontroler ten steruje pracą niewielkiego graficznego wyświetlacza OLED o organizacji 128×64 pikseli, używając do tego celu programowej implementacji interfejsu SPI, w który wyposażony jest zastosowany wyświetlacz. Obsługuje także scalony czujnik temperatury, wilgotności i ciśnienia atmosferycznego pod postacią układu firmy Bosch Sensortec o oznaczeniu BME280, używając do tego celu programowej

implementacji interfejsu I²C, w który wyposażony jest wspomniany czujnik.

Czujnik BME280

Układ BME280 stanowi główny element stacji pogodowej, dostarczając wszystkie interesujące nas dane pogodowe, w związku z czym warto przyrzeć się bliżej temu ciekawemu peryferium.

Układ BME280 charakteryzuje się następującymi, wybranymi cechami:

- pomiar ciśnienia atmosferycznego (300...1100 hPa), temperatury otoczenia (-40...85°C) i wilgotności (0...100%),
- szeroki zakres napięć zasilania 1,7...3,6 V,
- niski pobór prądu rzędu 3,6 μ A/pomiar,
- wysoka dokładność pomiaru oraz niski poziom szumów,
- szybka magistrala I²C z dopuszczalną prędkością sygnału zegarowego dochodzącą do 3,4 MHz,
- szybka magistrala SPI z dopuszczalną prędkością sygnału zegarowego dochodzącą do 10 MHz,
- małe wymiary obudowy typu LGA 2,5×2,5 mm.

Powyższe parametry idealnie wpisują się w założenia naszego projektu, a jedynym problemem jaki możemy napotkać stosując wspomniany element jest jego dość kłopotliwa, jeśli chodzi o montaż, obudowa LGA, wymagająca sporej wprawy w lutowaniu oraz odpowiedniego sprzętu lutowniczego. Obsługa peryferiów tego typu polega na zapisie/odczytanie wielu, specjalnych rejestrów konfiguracyjnych lub też rejestrów danych, przy udziale których, po pierwsze możemy zainicjować proces pomiarowy, a po drugie, dokonać odczytu interesujących nas wartości ciśnienia atmosferycznego, wilgotności i temperatury otoczenia. Będą to wartości „surowe”, wymagające odpowiedniego przeliczenia. Każdy element BME280 przechodzi na etapie produkcji proces kalibracji, który zapewnia osiągnięcie założonej dokładności pomiarów niezależnie od właściwości elementu piezo-rezystancyjnego, który stanowi w nim przetwornik



ciśnienia na napięcie. Proces ten kończy się ustaleniem szeregu (dokładnie osiemnastu) specjalnych współczynników korekcyjnych (zapisanych w nieulotnej pamięci EEPROM elementu), dzięki którym możliwe staje się obliczenie skompensowanej wartości ciśnienia atmosferycznego, wilgotności i temperatury. Jest to dość typowe rozwiązanie stosowane przez wielu producentów w przypadku elementów tego rodzaju, które przechodzą proces kalibracji na ostatnim etapie produkcji. Zatem, pierwszą czynnością, jaką należy wykonać w przypadku obsługi układu BME280 jest odczyt osiemnastu, 16- i 8-bitowych rejestrów, które przechowują wartości współczynników korekcyjnych. Następnie należy wysłać do układu BME280 rozkaz inicjujący pomiary, by po pewnym czasie odczytać nieskompensowane wartości interesujących nas parametrów. Dalej, na podstawie dość skomplikowanych wzorów dostarczonych przez producenta układu, można obliczyć skompensowane wartości ciśnienia atmosferycznego, wilgotności i temperatury otoczenia. Jest to dość niezrozumiałe podejście, ale stosowane przez większość producentów scalonych barometrów. Przecież korzystając z faktu, że współczynniki korekcyjne znajdują się w nieulotnej pamięci układu, można całą procedurę obliczeń położyć po stronie elementu, a użytkownikowi udostępnić gotowe wyniki, co znacznie ułatwiłoby obsługę takiego rodzaju peryferium. Na szczęście firma Bosch Sensortec dostarcza gotowy driver do obsługi swojego czujnika, co znacznie upraszcza proces implementacji własnego oprogramowania.

Program

Szczegóły implementacyjny mojej wersji oprogramowania zacznę od pliku

Dodatkowe materiały do pobrania ze strony www.media.avt.pl

W ofercie AVT* AVT-5722

Podstawowe parametry:

- pomiar ciśnienia atmosferycznego (300...1100 hPa),
- pomiar temperatury otoczenia (-40...85°C),
- pomiar wilgotności (0...100%),
- prezentacja wyników na wyświetlaczu OLED,
- zasilanie 5 V (USB),
- pobór prądu (z akumulatora) ok. 7,5 mA, lub 0,7 mA w stanie Standby,
- niewielkie wymiary.

Projekty pokrewne na www.media.avt.pl:

- AVT-5668 Moduł czujnika temperatury, wilgotności, ciśnienia z interfejsem I²C (EP 3/2019)
- AVT-5654 Amatorska stacja pogodowa (EP 12/2018)
- AVT-5639 Bezprzewodowy czujnik warunków atmosferycznych (EP 10/2018)
- AVT-5605 WiStation – domowa stacja pogodowa z prognozą pogody (EP 9/2017)

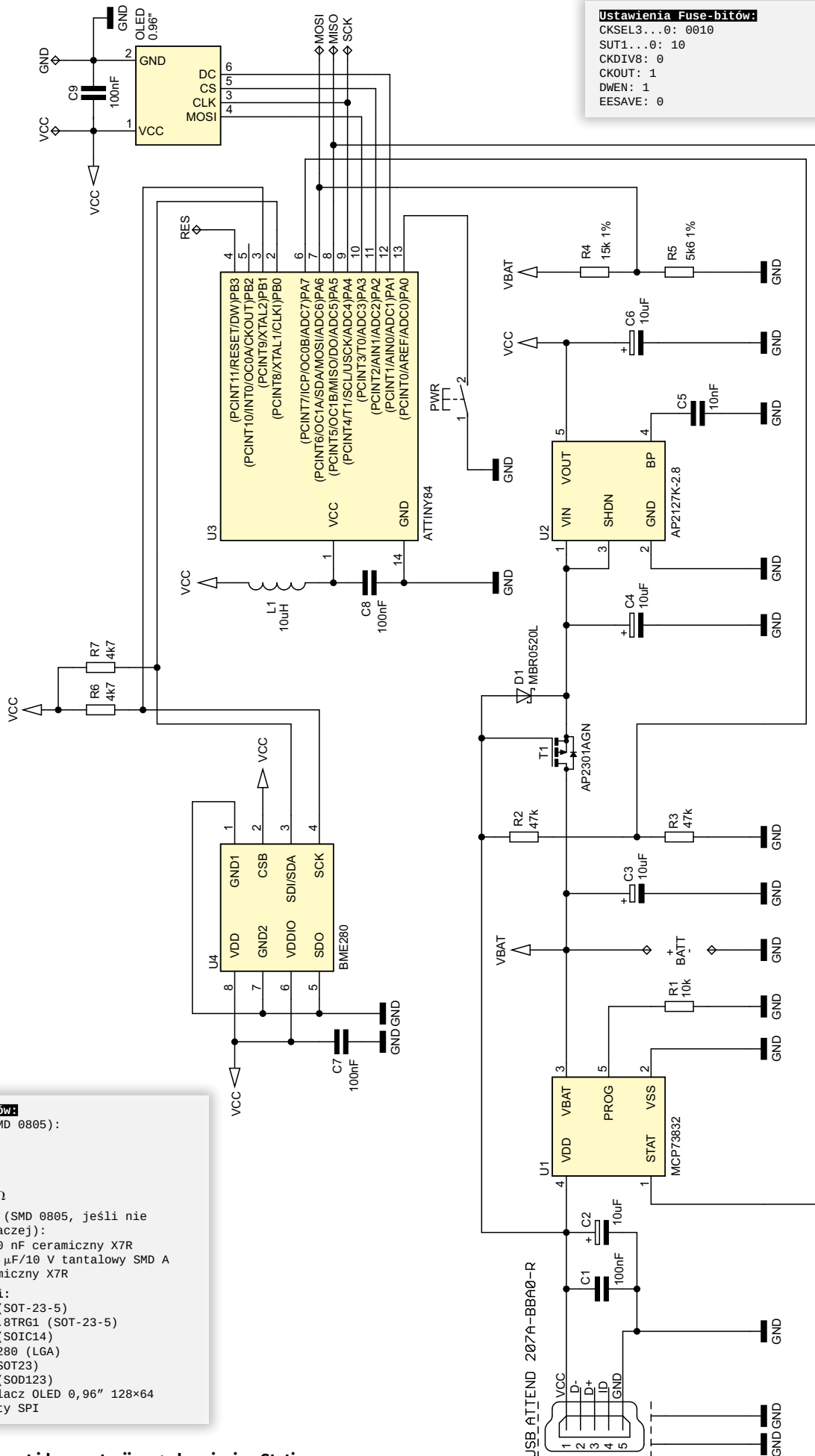
*** Uwaga! Elektroniczne zestawy do samodzielnego montażu. Wymagana umiejętność lutowania!**

Podstawową wersją zestawu jest wersja [B] nazywana potocznie KIT-em (z ang. zestaw). Zestaw w wersji [B] zawiera elementy elektroniczne (w tym [UK] – jeśli występuje w projekcie), które należy samodzielnie wlutować w dołączoną płytkę drukowaną (PCB). Wykaz elementów znajduje się w dokumentacji, która jest podlinkowana w opisie kitu.

Mając na uwadze różne potrzeby naszych klientów, oferujemy dodatkowe wersje:

- wersja [C] – zmontowany, uruchomiony i przetestowany zestaw [B] (elementy wlutowane w płytkę PCB)
- wersja [A] – płytkę drukowaną bez elementów i dokumentacji Kity, w których występuje układ scalony wymagający zaprogramowania, mają następujące dodatkowe wersje:
- wersja [A+] – płytkę drukowaną [A] + zaprogramowany układ [UK] i dokumentacja
- wersja [UK] – zaprogramowany układ

Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik pdf! Podczas składania zamówienia upewnij się, którą wersję zamawiasz! <http://sklep.avt.pl>. W przypadku braku dostępności na <http://sklep.avt.pl>, osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: kity@avt.pl.

**Ustawienia Fuse-bitów:**

```

CKSEL3...0: 0010
SUT1...0: 10
CKDIV8: 0
CKOUT: 1
DWEN: 1
EESAVE: 0

```

Wykaz elementów:**Rezystory: (SMD 0805):**

R1: 10 kΩ
 R2, R3: 47 kΩ
 R4: 15 kΩ 1%
 R5: 5,6 kΩ 1%
 R6, R7: 4,7 kΩ

**Kondensatory: (SMD 0805, jeśli nie
 zaznaczono inaczej):**
 C1, C7..C9: 100 nF ceramiczny X7R
 C2...C4, C6: 10 μF/10 V tantalowy SMD A
 C5: 10 nF ceramiczny X7R

Półprzewodniki:

U1: MCP73832 (SOT-23-5)
 U2: AP2127K-2.8TRG1 (SOT-23-5)
 U3: ATtiny84 (SOIC14)
 U4: Bosch BME280 (LGA)
 T1: LGE2301 (SOT23)
 D1: MBR0520L (SOD123)
 OLED: wyświetlacz OLED 0,96" 128×64
 niebiesko/żółty SPI

Rysunek 1. Schemat ideowy stacji pogodowej microStation

nagłówkowego związanego z obsługą elementu BME280, którego zawartość pokazuje **listing 1**. Został napisany w taki sposób, by bez zagłębiania się w dokumentację czujnika można było zorientować się w funkcjonalności poszczególnych rejestrów konfiguracyjnych/danych.

Dalej, potrzebujemy kilka prostych funkcji narzędziowych, dzięki którym (i przy udziale

Listing 1. Plik nagłówkowy do obsługi układu BME280
 //Adresy sprzętowe i2c układu BME280
 #define BME280_WRITE_ADDR 0xEC
 #define BME280_READ_ADDR 0xED

//Rejestry współczynników korekcyjnych
 #define BME280_DIG_T1_REG 0x88
 #define BME280_DIG_T2_REG 0x8A
 #define BME280_DIG_T3_REG 0x8C

#define BME280_DIG_P1_REG 0x8E
 #define BME280_DIG_P2_REG 0x90
 #define BME280_DIG_P3_REG 0x92
 #define BME280_DIG_P4_REG 0x94
 #define BME280_DIG_P5_REG 0x96
 #define BME280_DIG_P6_REG 0x98
 #define BME280_DIG_P7_REG 0x9A
 #define BME280_DIG_P8_REG 0x9C
 #define BME280_DIG_P9_REG 0x9E

#define BME280_DIG_H1_REG 0xA1
 #define BME280_DIG_H2_REG 0xE1
 #define BME280_DIG_H3_REG 0xE3
 #define BME280_DIG_H4_REG 0xE4
 #define BME280_DIG_H5_REG 0xE5
 #define BME280_DIG_H6_REG 0xE7

//Rejestry identyfikacyjne układu
 #define BME280_CHIPID_REG 0xD0
 #define BME280_VERSION_REG 0xD1

//Rejestr resetu software-owego
 #define BME280_SOFTRESET_REG 0xE0
 #define D0_RESET 0xB6

//Rejestr ustawień oversamplingu dla
 //pomiaru wilgotności
 #define BME280_CONTROLHUMID_REG 0xF2
 #define H_OVERSMP_SKIPPED (0<<0)
 #define H_OVERSMP_X1 (1<<0)
 #define H_OVERSMP_X2 (2<<0)
 #define H_OVERSMP_X4 (3<<0)
 #define H_OVERSMP_X8 (4<<0)
 #define H_OVERSMP_X16 (5<<0)

//Rejestr statusowy
 #define BME280_STATUS_REG 0xF3
 #define IS_MEASURING (1<<3)
 #define IM_UPDATE (1<<0)

//Rejestr ustawień oversamplingu dla
 //pomiaru temperatury i ciśnienia oraz
 //ustawień trybu pracy

#define BME280_CONTROL_MEASURE 0xF4
 #define T_OVERSMP_SKIPPED (0<<5)
 #define T_OVERSMP_X1 (1<<5)
 #define T_OVERSMP_X2 (2<<5)
 #define T_OVERSMP_X4 (3<<5)
 #define T_OVERSMP_X8 (4<<5)
 #define T_OVERSMP_X16 (5<<5)
 #define P_OVERSMP_SKIPPED (0<<2)
 #define P_OVERSMP_X1 (1<<2)
 #define P_OVERSMP_X2 (2<<2)
 #define P_OVERSMP_X4 (3<<2)
 #define P_OVERSMP_X8 (4<<2)
 #define P_OVERSMP_X16 (5<<2)
 #define MODE_SLEEP (0<<0)
 #define MODE_FORCED (1<<0)
 #define MODE_NORMAL (3<<0)

//Rejestr konfiguracyjny ustawień czasu
 //Standby (istotny w normalnym trybie pracy)
 //oraz ustawień filtra IIR

#define BME280_CONFIG_REG 0xF5
 #define STBY_TIME_0_5 (0<<5)
 #define STBY_TIME_62_5 (1<<5)
 #define STBY_TIME_125 (2<<5)
 #define STBY_TIME_250 (3<<5)
 #define STBY_TIME_500 (4<<5)
 #define STBY_TIME_1000 (5<<5)
 #define STBY_TIME_10 (6<<5)
 #define STBY_TIME_20 (7<<5)
 #define FILTER_OFF (0<<2)
 #define FILTER_2 (1<<2)
 #define FILTER_4 (2<<2)
 #define FILTER_8 (3<<2)
 #define FILTER_16 (4<<2)

//Rejestry danych: ciśnienia,
 //temperatury i wilgotności
 #define BME280_PRESSUREDATA_REG 0xF7
 #define BME280_TEMPDATA_REG 0xFA
 #define BME280_HUMIDDATA_REG 0xFD

Listing 2. Funkcje umożliwiające zapis i odczyt 8-bitowego rejestru
 //Zapis wartości „regValue” do 8-bitowego rejestru
 //znajdującego się pod adresem „regAddress”
 void BME280writeRegister8(uint8_t regAddress, uint8_t regValue)
 {
 i2cStart();
 i2cWriteByte(BME280_WRITE_ADDR);
 i2cWriteByte(regAddress);
 i2cWriteByte(regValue);
 i2cStop();
 }
 //Odczyt wartości 8-bitowego rejestru
 //znajdującego się pod adresem „regAddress”
 uint8_t BME280readRegister8(uint8_t regAddress)
 {
 uint8_t readByte;
 i2cStart();
 i2cWriteByte(BME280_WRITE_ADDR);
 i2cWriteByte(regAddress);
 i2cRepeatStart();
 i2cWriteByte(BME280_READ_ADDR);
 readByte = i2cReadByte(NACK);
 i2cStop();
 return readByte;
 }

Listing 3. Funkcja umożliwiająca odczyt wartości 16-bitowego rejestru
 //Odczyt wartości 16-bitowego rejestru
 //znajdującego się pod adresem „regAddress”
 //(format Little endian)
 uint16_t BME280readRegister16(uint8_t regAddress)
 {
 uint16_t readWord;
 i2cStart();
 i2cWriteByte(BME280_WRITE_ADDR);
 i2cWriteByte(regAddress);
 i2cRepeatStart();
 i2cWriteByte(BME280_READ_ADDR);
 //Odczytany, młodszy bajt słowa
 readWord = i2cReadByte(ACK);
 //Odczytany, starszy bajt słowa
 readWord |= i2cReadByte(NACK)<<8;
 i2cStop();
 return readWord;
 }

Listing 4. Funkcja umożliwiająca odczyt wartości 24-bitowego rejestru danych
 //Odczyt wartości 24-bitowego rejestru
 //znajdującego się pod adresem „regAddress”
 //(format Big endian, z przesunięciem 4 bity w prawo)
 uint32_t BME280readRegister24(uint8_t regAddress)
 {
 uint32_t readLong;
 i2cStart();
 i2cWriteByte(BME280_WRITE_ADDR);
 i2cWriteByte(regAddress);
 i2cRepeatStart();
 i2cWriteByte(BME280_READ_ADDR);
 //Odczytany bajt słowa MSB
 readLong = (uint32_t) i2cReadByte(ACK)<<12;
 //Odczytany bajt słowa LSB
 readLong |= (uint32_t) i2cReadByte(ACK)<<4;
 //Odczytany bajt słowa XLSB
 readLong |= (uint32_t) i2cReadByte(NACK)>>4;
 i2cStop();
 return readLong;
 }

Listing 5. Typ danych odpowiedzialny za przechowywanie wartości współczynników korekcyjnych
 //Struktura globalna przechowująca
 //współczynniki korekcyjne układu BME280
 struct
 {
 //Współczynniki temperatury
 uint16_t T1;
 int16_t T2;
 int16_t T3;
 //Współczynniki ciśnienia
 uint16_t P1;
 int16_t P2;
 int16_t P3;
 int16_t P4;
 int16_t P5;
 int16_t P6;
 int16_t P7;
 int16_t P8;
 int16_t P9;
 //Współczynniki wilgotności
 uint8_t H1;
 int16_t H2;
 uint8_t H3;
 int16_t H4;
 int16_t H5;
 int8_t H6;
 } Cfs;

magistrali I²C) będziemy mogli operować na rejestrach konfiguracyjnych oraz danych naszego peryferium. Jednak tym razem nie możemy ograniczyć się wyłącznie do dwóch funkcji umożliwiających zapis i odczyt, gdyż rejestry układu BME280 występują w trzech konfiguracjach: jako 8-, 16- i 24-bitowe i to zarówno w notacji little endian, jak i big endian. Funkcje umożliwiające zapis

i odczyt 8-bitowego rejestru układu BME280 prezentuje **listing 2**. Funkcję umożliwiającą odczyt wartości 16-bitowego rejestru układu BME280 zapisanego w notacji little endian (takimi rejestrami są niektóre z rejestrów przechowujących wartości współczynników korekcyjnych) prezentuje **listing 3**. Na koniec prosta funkcja narzędziowa umożliwiająca odczyt wartości 24-bitowego rejestru układu

BME280 zapisanego w notacji big endian (takimi rejestrami są rejestry „surowych” danych). Kod funkcji pokazuje **listing 4**.

Mając funkcje narzędziowe operujące na rejestrach układu BME280 pora na przedstawienie podstawowych funkcji obsługi. Zanim do nich przejdę niezbędne jest wprowadzenie nowego, strukturalnego typu danych, dzięki któremu możliwe stanie się odczytanie i przechowywanie wszystkich współczynników korekcyjnych. Wspomniany typ danych pokazano na **listingu 5**. Funkcję umożliwiającą odczyt wszystkich współczynników korekcyjnych układu BME280 prezentuje **listing 6**. Na **listingu 7**, pokazano funkcję inicjalizacyjną czujnika, a **listingi 8 i 9** prezentują dwie proste funkcje sprawdzające czy czujnik jest w trakcie odczytywania współczynników korekcyjnych (a dokładnie w trakcie przepisywania wartości współczynników korekcyjnych z wbudowanej, nieulotnej pamięci NVM do rejestrów konfiguracyjnych układu) lub w trakcie wykonywania pomiarów (zainicjowanych odrębną funkcją). Kod funkcji, która inicjuje proces wykonania pomiarów przez układ BME280 wygląda jak na **listingu 10**. Konieczność wprowadzenia takiej funkcji wynika z faktu, że układ BME280 został wcześniej skonfigurowany do pracy w trybie forced, czyli takim, w którym pomiary wykonywane są na żądanie, zaś po ich wykonaniu układ automatycznie przechodzi do stanu uśpienia o bardzo niskim poborze mocy (<0,1 μ A). Oczywiście może pracować również w normalnym trybie pracy podczas którego cyklicznie wykonuje pomiary bez potrzeby każdorazowej inicjacji procesu pomiarowego, jednak wtedy pobiera znacznie więcej energii, gdyż pomiędzy pomiarami nie przechodzi do stanu uśpienia (pozostaje w stanie Standby, którego czas konfigurujemy dzięki stosownym rejestrów konfiguracyjnym). Na koniec dość skomplikowane funkcje umożliwiające odczyt rzeczywistych (skompensowanych) wartości ciśnienia atmosferycznego, wilgotności i temperatury. Kod funkcji pokazano na **listingu 11, 12 i 13**. Zostały one napisane na podstawie API dostarczanego

Listing 6. Funkcja umożliwiająca odczyt wszystkich współczynników korekcyjnych

```
void BME280readCoeffs(void)
{
    Cfs.T1 = (uint16_t) BME280readRegister16(BME280_DIG_T1_REG);
    Cfs.T2 = (int16_t) BME280readRegister16(BME280_DIG_T2_REG);
    Cfs.T3 = (int16_t) BME280readRegister16(BME280_DIG_T3_REG);

    Cfs.P1 = (uint16_t) BME280readRegister16(BME280_DIG_P1_REG);
    Cfs.P2 = (int16_t) BME280readRegister16(BME280_DIG_P2_REG);
    Cfs.P3 = (int16_t) BME280readRegister16(BME280_DIG_P3_REG);
    Cfs.P4 = (int16_t) BME280readRegister16(BME280_DIG_P4_REG);
    Cfs.P5 = (int16_t) BME280readRegister16(BME280_DIG_P5_REG);
    Cfs.P6 = (int16_t) BME280readRegister16(BME280_DIG_P6_REG);
    Cfs.P7 = (int16_t) BME280readRegister16(BME280_DIG_P7_REG);
    Cfs.P8 = (int16_t) BME280readRegister16(BME280_DIG_P8_REG);
    Cfs.P9 = (int16_t) BME280readRegister16(BME280_DIG_P9_REG);

    Cfs.H1 = (uint8_t) BME280readRegister8(BME280_DIG_H1_REG);
    Cfs.H2 = (int16_t) BME280readRegister16(BME280_DIG_H2_REG);
    Cfs.H3 = (uint8_t) BME280readRegister8(BME280_DIG_H3_REG);

    Cfs.H4 = (int16_t) ((BME280readRegister8(BME280_DIG_H4_REG) << 4) |
        (BME280readRegister8(BME280_DIG_H5_REG) & 0xF));
    Cfs.H5 = (int16_t) ((BME280readRegister8(BME280_DIG_H5_REG + 1) << 4) |
        (BME280readRegister8(BME280_DIG_H5_REG) >> 4));

    Cfs.H6 = (int8_t) BME280readRegister8(BME280_DIG_H6_REG);
}
```

Listing 7. Funkcja inicjalizacyjna układu BME280

```
void BME280init(void)
{
    //Inicjalizacja interfejsu i2c
    i2cInit();
    //Reset programowy układu BME280
    BME280writeRegister8(BME280_SOFTRESET_REG, DO_RESET);
    _delay_ms(10);
    //Oczekiwanie, aż układ BME280 przepisze wszystkie
    //współczynniki korekcyjne z nieulotnej pamięci NVM
    //do stosownych rejestrów konfiguracyjnych
    //(po włączeniu zasilania)
    while(BME280isReadingCalibData());
    _delay_ms(10);
    //Odczyt współczynników korekcyjnych układu BME280
    //do pamięci RAM mikrokontrolera
    BME280readCoeffs();
    //Ustawiamy oversampling x1 dla pomiaru wilgotności
    BME280writeRegister8(BME280_CONTROLHUMID_REG, H_OVERSMP_X1);
    //Ustawiamy oversampling x1 dla pomiaru
    //temperatury i ciśnienia. Włączamy tryb forced układu BME280.
    BME280writeRegister8(BME280_CONTROL_MEASURE, T_OVERSMP_X1 |
        P_OVERSMP_X1 | MODE_FORCED);
}
```

Listing 8. Funkcja sprawdzająca czy układ jest w trakcie odczytywania współczynników korekcyjnych

```
uint8_t BME280isReadingCalibData(void)
{
    uint8_t Status = BME280readRegister8(BME280_STATUS_REG);
    return (Status & IM_UPDATE) != 0;
}
```

Listing 9. Funkcja sprawdzająca czy układ jest w trakcie wykonywania pomiarów

```
uint8_t BME280isMeasuring(void)
{
    uint8_t Status = BME280readRegister8(BME280_STATUS_REG);
    return (Status & IS_MEASURING) != 0;
}
```

Listing 10. Funkcja inicjalizująca proces wykonywania pomiarów (w trybie forced)

```
void BME280takeMeasure(void)
{
    //Inicjalizacja nowego pomiaru (niezbędna w trybie FORCED)
    BME280writeRegister8(BME280_CONTROL_MEASURE, T_OVERSMP_X1 |
        P_OVERSMP_X1 | MODE_FORCED);
    //Oczekiwanie na zakończenie pomiaru
    while(BME280isMeasuring());
    _delay_ms(5);
}
```

REKLAMA



Specjalistyczne szkolenia dla elektroników i automatyków

STM32

TECHDAYS

techdays@techdays.pl
TECHDAYS.PL

CERTYFIKOWANY PARTNER SZKOLENIOWY

przez firmę Bosch Sensortec. Korzystają z globalnej zmiennej `int32_t Tfine` przechowującej temperaturę otoczenia w wysokiej rozdzielczości (do kompensacji ciśnienia i wilgotności). Dość szczegółowy opis układu BME280 był podyktowany tym, że jest bardzo ciekawy element a oprócz drivera firmy Bosch nie ma w Internecie zbyt wielu sprawdzonych przykładów oprogramowania.

Prognozowanie pogody

Stacja pogodowa została wyposażona w pewną istotną funkcjonalność – umożliwia prognozowanie pogody. Aby zrozumieć zasadę działania zastosowanego mechanizmu należy omówić podstawowe zagadnienia związane z pojęciem ciśnienia atmosferycznego. Z definicji, ciśnienie atmosferyczne jest stosunkiem wartości siły, z jaką słup powietrza atmosferycznego naciska na powierzchnię Ziemi, do powierzchni, na jaką ten słup naciska. Zatem ciśnienie atmosferyczne w górach jest niższe, a na nizinach wyższe, ponieważ słup powietrza ma w tych rejonach różne wysokości. Zależność ta ma w przybliżeniu charakter wykładniczy, co pokazuje **rysunek 2**.

Wartość ciśnienia atmosferycznego dla standardowych warunków pogodowych dla wysokości h (n.p.m.) możemy wyznaczyć z uproszczonej zależności opisanej wzorem:

$$P_h = P_0 e^{-\frac{h}{7990}}$$

gdzie:

P_h – ciśnienie na wysokości h n.p.m. dla standardowych warunków pogodowych

P_0 – ciśnienie na poziomie morza równe 1013,25 hPa

h – wysokość n.p.m. w metrach

Właśnie na podstawie średniej wartości ciśnienia atmosferycznego na Ziemi na poziomie morza wprowadzono (dla pogodowych warunków standardowych), nieużywaną już, jednostkę atmosfery równą 1013,25 hPa. Jak wiadomo, ciśnienie atmosferyczne ulega ciągłym zmianom, które zależą od zmiany warunków pogodowych, w związku z czym na podstawie zmian ciśnienia na założonej wysokości n.p.m. możemy z pewnym prawdopodobieństwem określić tendencję zmian pogody. Dla przykładu, w umiarkowanych szerokościach geograficznych powolne obniżanie się ciśnienia oznacza zbliżanie się niżu, czyli pogorszenie pogody (wystąpienie chmur, opadów, w lecie – ochłodzenia, w zimie – ocieplenia) zaś wzrost ciśnienia łączy się z poprawą pogody (ustąpieniem mgły/zachmurzenia, osłabieniem wiatru itp.). Ciśnienie wyższe o kilka hPa

Listing 11. Funkcja zwracająca wartość rzeczywistego ciśnienia atmosferycznego

```
//Funkcja zwracająca zmierzone ciśnienie
//w formie liczby całkowitej bez znaku.
//Jednostką jest hPa

uint16_t readPressure(void)
{
    int32_t Var1;
    int32_t Var2;
    int32_t Var3;
    int32_t Var4;
    uint32_t Var5;
    uint32_t Pressure;
    int32_t rawPressure = BME280readRegister24(BME280_PRESSUREDATA_REG);

    Var1 = (((int32_t)Tfine) / 2) - (int32_t)64000;
    Var2 = (((Var1 / 4) * (Var1 / 4)) / 2048) * ((int32_t)Cfs.P6);
    Var2 = Var2 + ((Var1 * ((int32_t)Cfs.P5)) * 2);
    Var2 = (Var2 / 4) + (((int32_t)Cfs.P4) * 65536);
    Var3 = (Cfs.P3 * ((Var1 / 4) * (Var1 / 4)) / 8192) / 8;
    Var4 = (((int32_t)Cfs.P2) * Var1) / 2;
    Var1 = (Var3 + Var4) / 262144;
    Var1 = (((32768 + Var1) * ((int32_t)Cfs.P1)) / 32768;

    //Uniknięcie dzielenia przez zero
    if(Var1)
    {
        Var5 = (uint32_t)((uint32_t)1048576) - rawPressure;
        Pressure = ((uint32_t)(Var5 - (uint32_t)(Var2 / 4096))) * 3125;
        if (Pressure < 0x80000000) Pressure = (Pressure <= 1) / ((uint32_t)Var1);
        else Pressure = (Pressure / (uint32_t)Var1) * 2;

        Var1 = (((int32_t)Cfs.P9) * ((int32_t)((Pressure / 8) *
        (Pressure / 8) / 8192))) / 4096;

        Var2 = (((int32_t)(Pressure / 4)) * ((int32_t)Cfs.P8)) / 8192;
        Pressure = (uint32_t)((int32_t)Pressure + ((Var1 + Var2 + Cfs.P7) / 16));
        if(Pressure < 30000) Pressure = 30000;
        else if(Pressure > 110000) Pressure = 110000;
    }
    else Pressure = 30000;
    //Jednostką jest hPa
    return Pressure/100;
}
```

Listing 12. Funkcja zwracająca wartość temperatury otoczenia

```
//Funkcja zwracająca zmierzona temperaturę
// w formie liczby całkowitej ze znakiem.
//Jednostką jest 0.01°C

int16_t readTemperature(void)
{
    int32_t Var1;
    int32_t Var2;
    int16_t Temperature;
    int32_t rawTemperature = BME280readRegister24(BME280_TEMPDATA_REG);

    Var1 = (int32_t)((rawTemperature / 8) - ((int32_t)Cfs.T1 * 2));
    Var1 = (Var1 * ((int32_t)Cfs.T2)) / 2048;
    Var2 = (int32_t)((rawTemperature / 16) - ((int32_t)Cfs.T1));
    Var2 = (((Var2 * Var2) / 4096) * ((int32_t)Cfs.T3)) / 16384;
    Tfine = Var1 + Var2;
    Temperature = (Tfine * 5 + 128) / 256;

    return Temperature; //Jednostką jest 0.01°C
}
```

Listing 13. Funkcja zwracająca wartość wilgotności

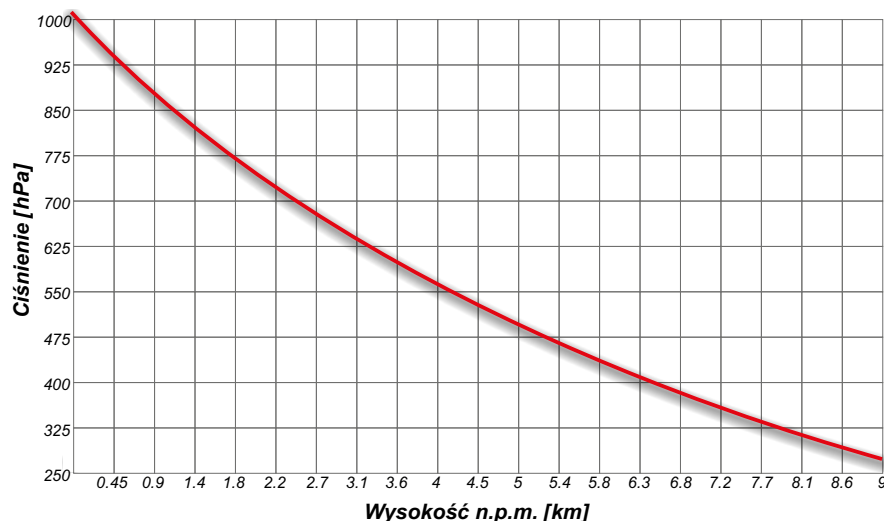
```
//Funkcja zwracająca zmierzona wilgotność
//w formie liczby całkowitej bez znaku.
//Jednostką jest %

uint8_t readHumidity(void)
{
    int32_t Var1;
    int32_t Var2;
    int32_t Var3;
    int32_t Var4;
    int32_t Var5;
    uint32_t Humidity;
    int32_t rawHumidity = BME280readRegister16(BME280_HUMIDDATA_REG);

    //Zamiana z Little endian na Big endian,
    //bo wilgotność jest w formacie Big endian
    rawHumidity = (rawHumidity>>8)|(rawHumidity<<8);

    Var1 = Tfine - ((int32_t)76800);
    Var2 = (int32_t)(rawHumidity * 16384);
    Var3 = (int32_t)((int32_t)Cfs.H4 * 1048576);
    Var4 = ((int32_t)Cfs.H5) * Var1;
    Var5 = ((Var2 - Var3) - Var4) + (int32_t)16384 / 32768;
    Var2 = (Var1 * ((int32_t)Cfs.H6)) / 1024;
    Var3 = (Var1 * ((int32_t)Cfs.H3)) / 2048;
    Var4 = ((Var2 * (Var3 + (int32_t)32768)) / 1024) + (int32_t)2097152;
    Var2 = ((Var4 * ((int32_t)Cfs.H2)) + 8192) / 16384;
    Var3 = Var5 * Var2;
    Var4 = ((Var3 / 32768) * (Var3 / 32768)) / 128;
    Var5 = Var3 - ((Var4 * ((int32_t)Cfs.H1)) / 16);
    Var5 = (Var5 < 0 ? 0 : Var5);
    Var5 = (Var5 > 419430400 ? 419430400 : Var5);
    Humidity = (uint32_t)(Var5 / 4096);
    if (Humidity > 102400) Humidity = 102400;

    return Humidity/1024; //Jednostką jest %
}
```



Rysunek 2. Wykres zależności bezwzględnego ciśnienia atmosferycznego od wysokości n.p.m.

od ciśnienia p_n dla danego rejonu zwiastuje utrwalenie się ładnej pogody, jednak z drugiej strony, szybki wzrost ciśnienia może także zwiastować burzę. Co ciekawe, na podstawie gradientu zmian ciśnienia można dość dokładnie określić spodziewaną siłę wiatru. Pomimo tych, wydawałoby się prostych zależności, przewidywanie zmian pogody jest procesem bardzo skomplikowanym (zwłaszcza przygotowywanie prognoz długookresowych) i obarczonym pewnym błędem, o czym niejednokrotnie możemy się przekonać słuchając prognoz pogody. Jednak dla prostych aplikacji możemy posilkować się algorytmami, z jakich korzysta większość elektronicznych stacji pogodowych, a które oparte są tylko na śledzeniu zmian ciśnienia atmosferycznego. Z uwagi na charakter zachodzących zmian prognozowanie pogody jest znacznie utrudnione dla obszarów położonych wysoko n.p.m. Na koniec, warto podkreślić istotną różnicę pomiędzy wartością bezwzględnego ciśnienia atmosferycznego dla danej miejscowości (ciśnienia tam panującego) a wartością ciśnienia podawaną w prognozach pogody. Ostatnie jest ciśnieniem, jakie wystąpiłoby danego dnia w danej miejscowości, gdyby znajdowała się ona na poziomie morza, czyli jest to bezwzględne ciśnienie atmosferyczne dla tego rejonu przeliczone dla poziomu morza. Tego typu konwersja ułatwia zorientowanie się, co do warunków pogodowych dla różnych wysokości n.p.m.

W nieskomplikowanych konstrukcjach stacji pogodowych, które do prognozowania pogody wykorzystują wyłącznie wartość mierzonego ciśnienia atmosferycznego, korzysta

się z dwóch rodzajów algorytmów. Pierwszy z nich śledzi zmiany ciśnienia atmosferycznego i po upływie pewnego czasu (zwykle 6 do 12 godzin) na podstawie gradientu zmian jest w stanie określić oczekiwany stan pogody. Drugi z algorytmów, z którego korzysta nasze urządzenie, oblicza wzorcowe ciśnienie dla danej wysokości n.p.m. (na której to się znajdujemy) dla dobrych warunków pogodowych (wspomniany wcześniej wzór na p_h) i na podstawie różnicy ciśnienia mierzonego i wzorcowego określa prognozę pogody niwelując potrzebę oczekiwania na zmiany gradientu ciśnienia. Oczywiście, prognozy takie mogą być obciążone dużym błędem, lecz wykonanie bardziej wiarygodnej prognozy pogody wymagałoby dysponowania większą liczbą wielkości wejściowych. Dla naszego urządzenia przyjęto zależności pokazane w tabeli 1. Aby zapewnić realizację funkcjonalności prognozowania pogody nasze urządzenie pozwala ustawić wysokość nad poziomem morza, na jakiej się znajduje. W tym celu przewidziano odpowiednią funkcjonalność systemu Menu.

Budowa

Kilka słów komentarza wymaga blok zasilający, zbudowany z zastosowaniem specjalizowanego kontrolera ładowania akumulatorów li-ion i li-pol pod postacią układu MCP73832 firmy Microchip. Integruje w sobie kompletny system ładowania, którego podstawowe cechy to:

- szeroki zakres napięć zasilania 3,75...6 V,
- programowany prąd szybkiego ładowania 15...500 mA (rezystor R1),

- możliwość wyboru wartości prądu ładowania wstępnego,
- możliwość wyboru poziomu naładowania akumulatora,
- wbudowany mechanizm wykrywania podłączonego akumulatora,
- trójstanowe wyjście statusu procesu ładowania STAT,
- automatyczne przejście do trybu power-down,

Stosując układ w aplikacji prostej ładowarki jedynym „zmartwieniem” użytkownika jest ustawienie prądu szybkiego ładowania, poprzez dobór rezystora podłączonego pomiędzy wyprowadzenie PROG a masę. Prąd ten dobieramy według poniższej zależności:

$$I_{\text{REG}} = \frac{1000V}{R_{\text{PROG}}}$$

gdzie:

I_{REG} – wyrażono w mA,

R_{PROG} – w kΩ.

W naszym urządzeniu rezystor RPROG(R1) ma wartość 10 kΩ, co ustawia prąd ładowania szybkiego na wartość 100 mA. Taka wartość została celowo wybrana. Do zasilania naszego urządzenia przewidziano podłączenie go do portu USB komputera (lub popularnego zasilacza sieciowego z portem USB przeznaczonego do ładowania telefonów komórkowych), który zwykle pozwala na maksymalny pobór prądu rzędu 500 mA w trybie high-power, zaś typowo 100 mA.

Dodatkowego wyjaśnienia wymaga opcjonalny układ współdzielenia obciążenia zbudowany przy użyciu tranzystora T1 typu MOSFET z kanałem P, diody Schottky'ego D1 oraz rezystorów R2 i R3. Takie rozwiązanie było konieczne, aby nie zaburzać działania kontrolera ładowania, ponieważ odbiornik pobierając nieustannie prąd z ogniwa, a więc z układu nadzorującego, nie pozwoli na skuteczną detekcję końca procesu ładowania. W przypadku obecności napięcia USB (czyli stanu wysokiego na bramce tranzystora) tranzystor T1 przechodzi w stan wyłączenia, odłączając tym samym ładowany akumulator

REKLAMA

Tabela 1. Zależności opisujące prognozowanie pogody, zastosowane w urządzeniu

Różnica pomiędzy P_h a P_{comp}	Symbol pogody
$\text{Diff} \geq 9$	Słoneczko
$-9 < \text{Diff} < 9$	Zachmurzenie
$\text{Diff} \leq -9$	Deszcz/Śnieg

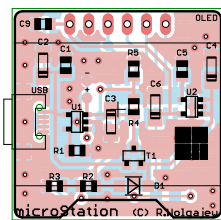
od obciążenia. W tym czasie obciążenie, jakim jest stacja pogodowa, zostaje zasilone bezpośrednio z napięcia portu USB poprzez diodę D1. W przypadku odłączenia napięcia zasilającego USB, bramka tranzystora T1 zostaje ściągnięta do masy powodując przewodzenie tranzystora, a tym samym zasilanie urządzenia z akumulatora. Wtedy dioda D1 zabezpiecza przed przepływem prądu wstecznego tj. z akumulatora w kierunku źródła napięcia zasilającego (USB). Wyjście z układu ładowania wprowadzono na wejście stabilizatora LDO typu AP2127K-2.8TRG1, który zapewnia stały poziom napięcia zasilającego system mikroprocesorowy (2,8 V) niezależnie od stanu układu ładowania.

Aplikacja urządzenia „microStation” korzysta z wbudowanego w strukturę mikrokontrolera przetwornika ADC (wejście ADC6), przy udziale którego (poprzez prosty dzielnik napięciowy R4, R5) mierzony jest poziom napięcia akumulatora zasilającego co pozwala na orientacyjne określenie stanu jego naładowania. Jest to oczywiście rozwiązanie dość proste, gdyż dokładne określenie stanu naładowania akumulatora wymagałoby kontrolowania wielkości gromadzonego i traconego ładunku, jednak w tak prostym systemie jest w zupełności wystarczające.

Montaż

Schemat montażowy urządzenia microStation pokazano na **rysunkach 3 i 4**.

Niewielki, dwustronny obwód drukowany z przewagą elementów SMD wymiarami zbliżony jest do wymiarów płytki drukowanej zastosowanego modułu wyświetlacza OLED (28×28 mm). Montaż urządzenia rozpoczynamy od warstwy BOTTOM, gdzie przylutowujemy układ BME280. Proces ten najłatwiej wykonać przy użyciu stacji lutowniczej na gorące powietrze (tzw. Hot Air) i odpowiednich stopów lutowniczych, gdyż obudowa tego elementu nie pozwala na zastosowanie typowej stacji lutowniczej.



Radiowy przedłużacz DS18B20

Jeden z najpopularniejszych czujników temperatury z interfejsem cyfrowym, jakim niewątpliwie jest DS18B20, musi być połączony z układem nadrzędnym za pomocą dwóch lub trzech przewodów. Prezentowany układ pozwala zrealizować połączenie bezprzewodowo – drogą radiową.

Dodatkowe materiały do pobrania ze strony www.media.avt.pl

W ofercie AVT* AVT-5720

Podstawowe parametry:

- realizuje połączenie bezprzewodowe magistrali 1-Wire dla czujnika DS18B20,
- reaguje tylko na komendę 0xBE, czyli żądanie o zawartość „scratchpada”,
- współpracuje z systemami zasilanymi napięciem 3,3 lub 5 V,
- komunikacja radiowa w paśmie ISM o częstotliwości 868 MHz,
- wymaga zasilania 3,3 V

Projekty pokrewne na www.media.avt.pl:

- AVT-5630 Bezprzewodowy przedłużacz sygnalizatora (EP 7/2018)
- AVT-5590 Zdalny włącznik radiowy (EP 6/2017)
- AVT-1838 Uniwersalny przedłużacz pilotów RTV (EP 01/2015)
- AVT-1840 Włącznik 230 V sterowany dowolnym pilotem na podczerwień (EP 11/2014)
- AVT-1815 4-kanałowy przełącznik sterowany dowolnym pilotem IR (EP 8/2014)
- AVT-5455 Zdalny włącznik dwukanałowy (EP 6/2014)
- AVT-5407 Włącznik sterowany radiowo (EP 8/2013)
- AVT-5290 3-kanałowa aparatura do zdalnego sterowania modeli (EP 5/2011)
- AVT-1520 Zdalny włącznik radiowy (EP 4/2009)
- AVT-2851 Zdalne sterowanie (nie tylko) RC5 (Edw 1/2008)
- AVT-1308 Zdalny włącznik 4 urządzeń (EP 8/2006)
- AVT-390 8-kanałowy przełącznik sterowany pilotem RC5/SIRC (EP 4/2006)
- AVT-559 Radiowy przedłużacz pilotów (EP 2-3/2004)
- AVT-517 Radiowy system zdalnego sterowania z kanałem zwrotnym (EP 7-8/2003)
- AVT-2482 Czerokanałowe zdalne sterowanie (Edw 4/2001)

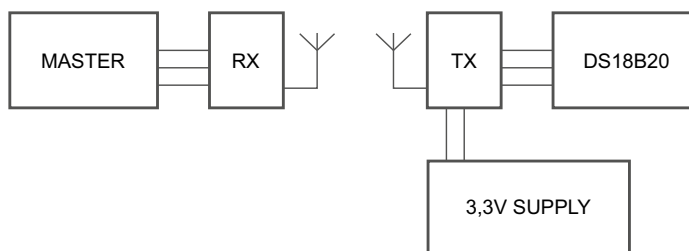
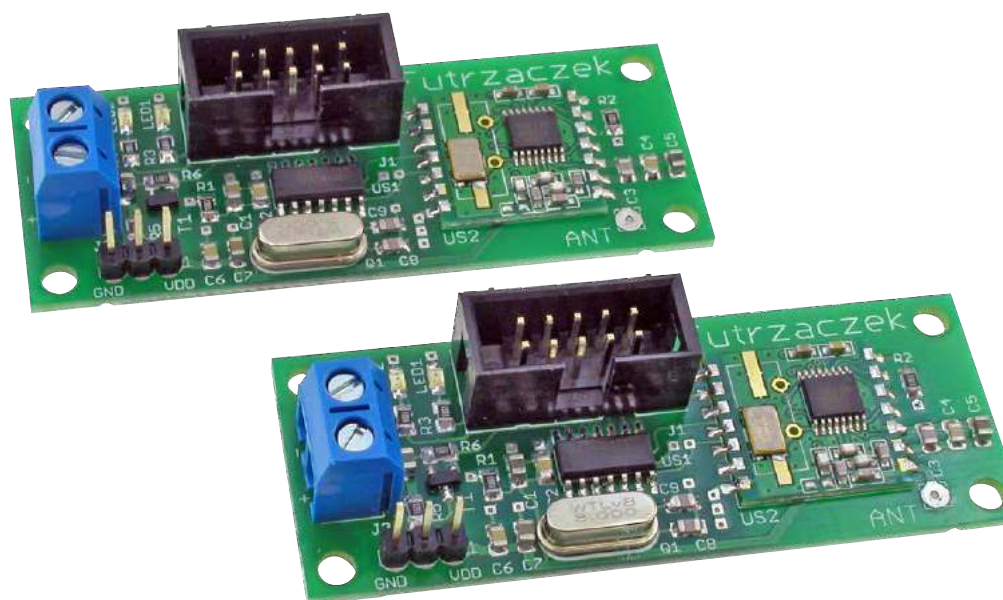
Uwaga! Elektroniczne zestawy do samodzielnego montażu. Wymagana umiejętność lutowania!

Podstawową wersją zestawu jest wersja [B] nazywana potocznie KIT-em (z ang. zestaw). Zestaw w wersji [B] zawiera elementy elektroniczne (w tym [UK] – jeśli występuje w projekcie), które należy samodzielnie włożyć w dołączoną płytkę drukowaną (PCB). Wykaz elementów znajduje się w dokumentacji, która jest podlinkowana w opisie kitu. Mając na uwadze różne potrzeby naszych klientów, oferujemy dodatkowe wersje:

- wersja [C] – zmontowany, uruchomiony i przetestowany zestaw
- [B] (elementy wmontowane w płytkę PCB)
- wersja [A] – płytkę drukowaną bez elementów i dokumentacji
- Kity w których występuje układ scalony wymagający zaprogramowania, mają następujące dodatkowe wersje:
- wersja [Av] – płytkę drukowaną [A] + zaprogramowany układ [UK] i dokumentacja
- wersja [UK] – zaprogramowany układ

Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik pdf! Podczas składania zamówienia upewnij się, która wersję zamawiasz! <http://sklep.avt.pl>. W przypadku braku dostępności na <http://sklep.avt.pl>, osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: kity@avt.pl.

Wyobraźmy sobie sytuację, w której życzeniem klienta jest panel sterujący, np. wentylacją, umieszczony w jednym kącie pomieszczenia, a czujnik temperatury w zupełnie innym. Na poprowadzenie



Rysunek 1. Schematyczne przedstawienie użycia opisanych modułów

dodatkowych przewodów nie wyraża zgody. Jeśli w miejscu zamontowania czujnika znajduje się źródło zasilania, np. 12 VDC dla systemu automatycznych rolet, to nasz układ będzie idealnym rozwiązaniem do wybrnięcia z sytuacji. Idea działania jest prosta: jeden moduł (dalej nazywany nadajnikiem, TX) cyklicznie komunikuje się z układem DS18B20 i pozyskane informacje wysyła drogą radiową. Drugi moduł (odbiorca, RX) odbiera te informacje i przechowuje je w pamięci. Podłączony jest w miejsce układu DS18B20 w urządzeniu nadrzędnym. Układ nadrzędny może w każdej chwili, przy użyciu zapytania zgodnego z protokołem 1-wire, zażądać informacji, a wtedy odbiorca przekaże dane, które otrzymał z nadajnika. Schemat blokowy takiego systemu pokazuje rysunek 1.

Ponieważ założono, że chodzi o „przedłużenie” jednego czujnika, układ reaguje tylko na jedną komendę: 0xBE, czyli żądanie o zawartość scratchpada (czyli 9 bajtowy blok danych zawierający podstawowe informacje,

w tym wynik pomiaru). Pierwsze dwa bajty niosą informację o temperaturze, odczytanie tych wartości wystarczy do obliczenia wyniku. W ten sposób działają najprostsze algorytmy komunikacji z DS18B20, a jeden z nich (napisany w Arduino) można zobaczyć na [listingu 1](#). Wszystkie komendy,

REKLAMA

Specjalistyczne szkolenia dla elektroników i automatyków

TECHDAYS

techdays@techdays.pl
TECHDAYS.PL

CERTYFIKOWANY PARTNER SZKOLENIOWY

```
Listing 1. Przykładowy kod odpowiadający
za komunikację z czujnikiem DS18B20
newWire.reset();
newWire.write(0xCC, 0); //skip ROM
newWire.write(0x44, 0); //convert T
delay(750);
newWire.reset();
newWire.write(0xCC, 0); //skip ROM
newWire.write(0xBE, 0); //read scratchpad
data[0] = newWire.read();
data[1] = newWire.read();
temp = (data[0] + (data[1] * 256)) / 16;
```

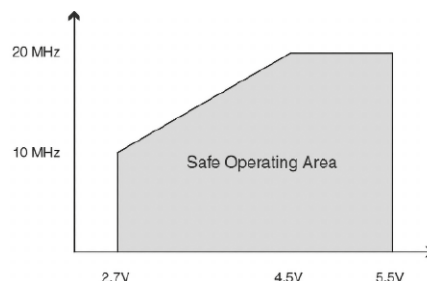
poza zresetowaniem magistrali oraz wspominanym wcześniej odczytem *scratchpada*, są przez układ pomijane.

W układzie nie zaimplementowano mechanizmów sprawdzających wiarygodność danych otrzymanych drogą radiową. Podczas testów nie zdarzyły się przekłamania, które nie byłyby łatwe do wykrycia – prędzej łączność została całkowicie zerwana, co skutkowało wyzerowaniem bajtów *scratchpada*. Jeżeli jednak dana aplikacja wymaga wysokiej wiarygodności odczytywanych danych, należałoby dodać sprawdzanie sumy kontrolnej, która znajduje się w ostatnim bajcie *scratchpada*.

Budowa

System przedłużający magistralę 1-wire drogą radiową składa się z dwóch modułów – nadawczego i odbiorczego – o identycznej budowie od strony elektrycznej. Różnica tkwi w oprogramowaniu. Schemat ideowy pojedynczego modułu znajduje się na **rysunku 2**. Układem zarządzającym pracą całego modułu, w tym komunikacją radiową, jest mikrokontroler Attiny24A. Jego zasilanie odsprzęgają dwa kondensatory, C1 i C2, umiejscowione blisko odpowiednich wyprowadzeń. Dodatkowy rezystor R1 podciąga wyprowadzenie RESET do napięcia +3,3 V, którym zasilany jest cały moduł.

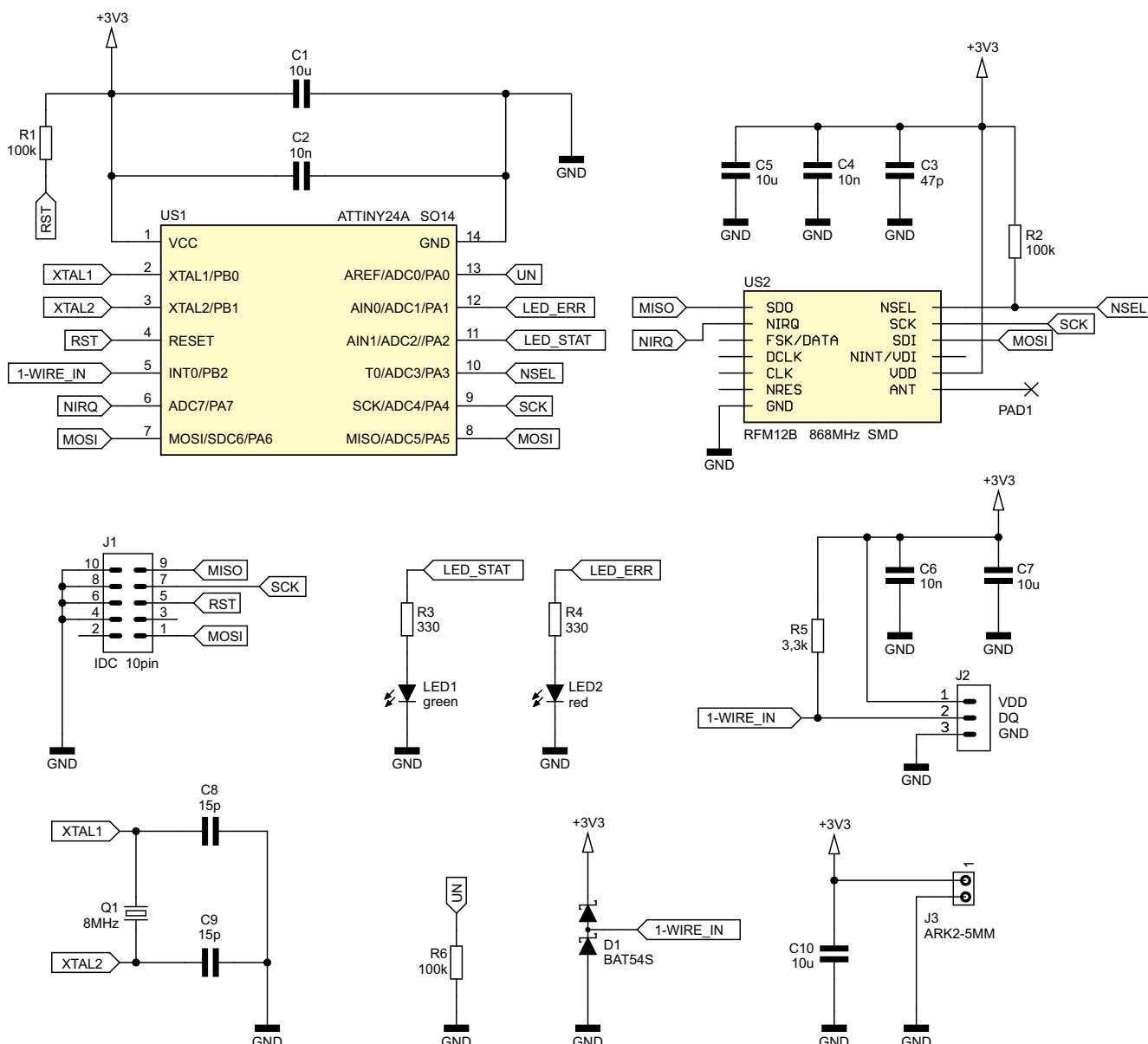
Magistrala 1-wire jest restrykcyjna pod względem zależności czasowych. Dlatego zdecydowano się na stabilizację zegara mikrokontrolera przy użyciu zewnętrznego rezonatora kwarcowego o częstotliwości 8 MHz. Taka, a nie wyższa, wartość częstotliwości została wymuszona relatywnie niskim napięciem zasilania – w tych warunkach,



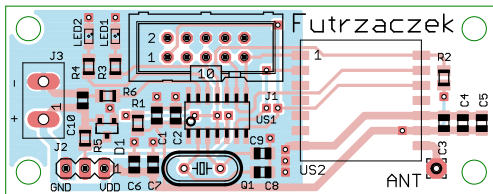
Rysunek 3. Maksymalna częstotliwość zegara w funkcji napięcia zasilającego dla Attiny24

częstotliwość sygnału zegarowego nie może być większa niż ok. 10 MHz, na co wskazuje wykres z **rysunku 3**.

Komunikację bezprzewodową w paśmie ISM na częstotliwości 868 MHz realizuje gotowy układ RFM12B. Komunikuje się z mikrokontrolerem przy użyciu magistrali SPI. Ponieważ te same wyprowadzenia służą do programowania pamięci Flash w Attiny24, eliminacja kolizji została



Rysunek 2. Schemat jednego modułu radiowego „przedłużacza” DS18B20



Rysunek 4. Schemat płytki PCB wraz z rozmieszczeniem elementów



Rysunek 5. Konfiguracja fusebitów w programie BitBurner

zrealizowana poprzez dodanie rezystora R2, który podciąga wyprowadzenie NSEL do dodatniej linii zasilania. Oznacza to, że wyprowadzenia dedykowane dla SPI są domyślnie w stanie wysokiej impedancji. Mikrokontroler w odpowiednich momentach ściąga potencjał tej linii do zera, co inicjuje możliwość komunikacji z RFM12B. Wyprowadzenie anteny znajduje się na polu lutowniczym PAD1. Uznano, że wygodniej będzie dolutować zewnętrzną antenę w postaci odcinka przewodu o odpowiedniej długości niż borykać się z umiejscowieniem w obudowie płytki powiększonej o antenę wewnętrzną.

Złącze J1 służy do programowania pamięci Flash mikrokontrolera. Układ wyprowadzeń jest zgodny z typowym standardem ISP KANDA, do którego dedykowana jest większość niewielkich programatorów na złącze USB. Do wskazywania stanu pracy modułu służą dwie diody LED. W zależności od tego, czy jest to nadajnik, czy też odbiornik, ich świecenie bądź miganie ma różne znaczenie, co zostanie szerzej opisane w dalszej części artykułu.

Czujnik typu DS18B20 (w przypadku modułu nadajnika) lub dedykowany układ nadrzędny (w przypadku odbiornika) podłącza się do złącza J2. Jeżeli w systemie znajduje się rezystor podciągający na magistrali

1-wire, o odpowiedniej wartości, R5 należy wymontować.

Podwójna dioda Schottky typu BAT54S została użyta do zabezpieczenia delikatnego wejścia mikrokontrolera przed impulsami, które mogą indukować się w przewodzie sygnałowym. Przez większość czasu jest on obciążony relatywnie wysoką impedancją (rzędu kΩ), więc mogą występować zakłócenia.

Cały układ pracuje pod napięciem 3,3 V, lecz zadziała poprawnie z układem nadrzędnym, zasilanym napięciem 5 V. Rezystor podciągający powinien być wówczas dołączony do wyższego napięcia. Dzięki temu, w stanie spoczynku, potencjał linii 1-wire wyniesie ok. 3,6 V (3,3 V napięcia zasilającego + 0,3 V napięcie przewodzenia górnej diody D1). Taka wartość zostanie poprawnie zinterpretowana jako logiczna „1” zarówno przez moduł odbiornika, jak i układ nadrzędny, o ile próg detekcji tego poziomu logicznego leży dostatecznie nisko. Przykładowo, dla układu ATmega328 zasilanego napięciem 5 V wynosi 3 V, czyli cały system zadziała poprawnie.

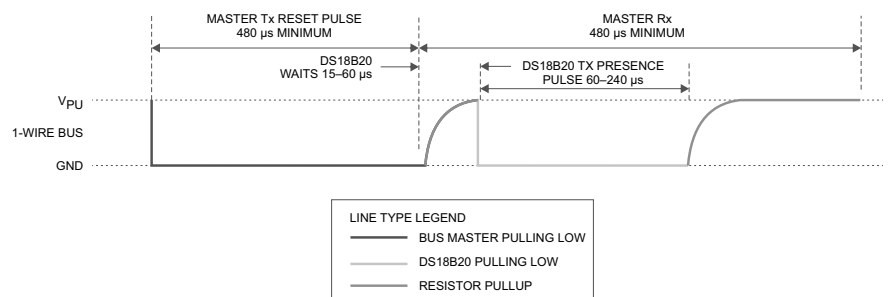
Zasilanie dla modułu może zostać dołączone w dwóch miejscach: zarówno do złącza

śrubowego typu ARK (J3), jak i szpilkowego (J2). Nie przewidziano stabilizatora napięcia, ponieważ zmniejszyłoby to uniwersalność płytki – w systemie nadrzędnym prawdopodobnie będzie odpowiednie napięcie 3,3 V. W przeciwnym razie do dyspozycji będzie napięcie zdecydowanie wyższe, które trzeba będzie obniżyć dodatkową przetwornicą impulsową (dla zmniejszenia strat mocy).

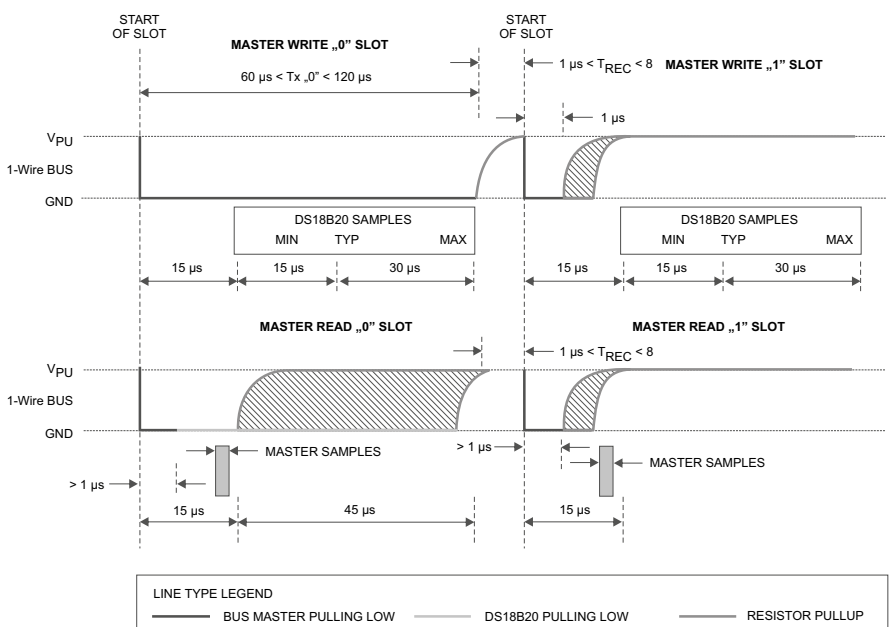
Montaż i uruchomienie

Oba moduły, nadajnik i odbiornik, realizowane są na tej samej płytce drukowanej o wymiarach 25×65 mm. Schemat płytki PCB wraz z rozmieszczeniem elementów pokazuje rysunek 4. Obsada elementów dla obu modułów jest identyczna, z ewentualną różnicą w postaci wspomnianego wcześniej rezystora R5. Po przylutowaniu elementów w obudowach do montażu powierzchniowego (SMD), na płytce powinien znaleźć się kwarc XTAL1 oraz wszystkie złącza. Na samym końcu polecam przylutować moduł RFM12B, aby nie uległ uszkodzeniu podczas montażu pozostałych części.

Prawidłowo zmontowane moduły nie wymagają czynności uruchomieniowych. Należy pamiętać o zamontowaniu anten. W najprostszym wykonaniu, funkcję anteny może pełnić odcinek przewodu w izolacji



Rysunek 6. Przebieg czasowy komendy inicjującej czujnik



Rysunek 7. Transmisja do przesyłu komend i danych

Wykaz elementów: (taki sam dla nadajnika i odbiornika)

Rezystory:

R1, R2, R6: 100 kΩ SMD0805
R3, R4: 330 Ω SMD0805
R5: 3,3 kΩ SMD0805 (opis w tekście)

Kondensatory:

C1, C5, C7, C10: 10 µF/10 V SMD0805
C2, C4, C6: 10 nF SMD0805
C3: 47 pF SMD0805
C8, C9: 15 pF SMD0805

Półprzewodniki:

D1: BAT54S SOT23
LED1: LED zielona SMD0805
LED2: LED czerwona SMD0805
US1: Attiny24A S014
US2: RFM12B 868 MHz SMD

Inne:

J1: IDC 10pin 2,54 mm proste
J2: goldpin męski 3 pin 2,54 mm
J3: ARK2/500

Listing 2. Obsługa urządzenia nadrzędnego, kluczowe fragmenty kodu

```
//sygnał RESET
uint8_t reset_pulse(void){ //1 - układ obecny, 0 - błąd
    uint8_t presence;
    LOW_1WIRE;
    CLEAR_1WIRE;
    _delay_us(480); //nadać sygnał RESET, min. 480us
    SET_1WIRE; //zwolnij linię
    _delay_us(60); //odczekać na odpowiedź, min. 60us
    //weryfikacja odpowiedzi
    if(IS_LOW){presence = 1;} else {presence = 0;}
    _delay_us(440); //odczekać na zanik odpowiedzi
    //weryfikacja po zaniku
    if(IS_HIGH && presence){presence = 1;}
    else {presence = 0;}
    return presence;
}

//nadanie jednego bitu
void send_bit(uint8_t bit){
    LOW_1WIRE;
    CLEAR_1WIRE;
    _delay_us(1); //nadać 1 krótkim impulsem
    if(bit == 1){ SET_1WIRE; }
    _delay_us(70); //nadać 0 dłuższym impulsem
    SET_1WIRE;
}

//odbiór jednego bitu
uint8_t read_bit(void){
    uint8_t bit;
    LOW_1WIRE;
    //wystaw zbocze opadające na krótką chwilę
    CLEAR_1WIRE;
    _delay_us(1);
    SET_1WIRE; //zwolnij linię
    //odczekać na ew. „przeciągnięcie” impulsu przez czujnik
    _delay_us(10);
    if(IS_HIGH){bit = 1;} else {bit = 0;} //sprawdź stan
    _delay_us(40); //uzupełnienie całego
    return bit;
}

//nadanie jednego bajtu
void send_byte(uint8_t byte){
    uint8_t i=8;
    while(i--){
        send_bit(byte&1);
        byte>>=1;
    }
}

//odbiór jednego bajtu
uint8_t read_byte(void){
    uint8_t byte = 0;
    uint8_t i=8;
    while(i--){
        byte>>=1;
        byte|=(read_bit()<<7);
    }
    return(byte);
}
```

o długości ok. 17 cm – będzie anteną półfalową dla częstotliwości 868 MHz.

Po podłączeniu zasilania o napięciu 3,3 V (dobrze filtrowane i stabilizowane), mikrokontrolery w obu modułach należy zaprogramować wsadom pamięci Flash. Ponadto, należy zmienić ustawienia *fusebitów* na wartości:

Low Fuse = 0xFD

High Fuse = 0xDD

Szczegółowa konfiguracja jest widoczna na **rysunku 5**, gdzie znajduje się zrzut okna programu BitBurner. Ustawienie progu zadziałania *BOD* na 2,7 V zapobiegnie powstaniu błędów podczas uruchamiania, kiedy

napięcie zasilające narasta zbyt wolno. Po tych czynnościach, moduły są gotowe do działania. Średni pobór prądu wynosi 27 mA dla modułu nadawczego oraz 14 mA dla modułu odbiorczego.

Eksploracja

Po włączeniu zasilania nadajnika, z dołączonym czujnikiem DS18B20, co 750 ms następuje odczyt 9 bajtów *scratchpada*. Dwa pierwsze bajty zawierają informację o temperaturze z najwyższą dostępną rozdzielczością, tj. 12 bitów. Jednocześnie, co 100 ms uruchamiany jest nadajnik, który wysyła

zawartość *scratchpada*. Tak częste transmisje służą eliminacji przypadkowego przekłamania, którego wyeliminowanie przy użyciu mechanizmu zwrotnych potwierdzeń byłoby skomplikowane w realizacji, a jednym z założeń projektu była prostota działania.

Diody LED1 i LED2 sygnalizują stan pracy modułu nadajnika. Opis sygnalizacji prezentuje **tabela 1**. Błąd komunikacji z dołączonym układem DS18B20 jest rozpoznawany poprzez brak prawidłowego sygnału obecności (*presence*), wystawianego przez czujnik po zresetowaniu magistrali.

Odbiornik cały czas nasłuchuje danych emitowanych przez nadajnik oraz reaguje na zapytania układu nadrzędnego wysyłane magistralą 1-wire. Jego działanie różni się od modułu nadawczego, dlatego diody LED również mają odmienne znaczenie – **tabela 2**. Dioda LED1 zaświeca się, kiedy przez min. 1,5 s nie zarejestrowano transmisji z nadajnika. Transmisja realizowana jest co 100 ms i każde poprawnie odebrane dane powodują zerowanie licznika czasu. Odbiornik nie weryfikuje poprawności odebranych danych, jest to zadanie układu nadrzędnego. Obserwując mignięcia diody LED2 można zauważyć, że komunikacja odbiornika z układem nadrzędnym przebiega prawidłowo. Opisany układ nie reaguje na takie komendy jak *Search ROM (0xF0)* czy *Write Scratchpad (0x4E)*. Realizuje tylko podstawowy cel, czyli umożliwia odczyt danych z czujnika DS18B20 za pośrednictwem drogi radiowej.

Dla dociekliwych

Magistrala 1-wire stawia ostre wymagania czasowe wobec dołączonych układów. Bezpośrednie przesyłanie stanów logicznych drogą radiową jest niemożliwe, ponieważ w krytycznych przypadkach, decyzja o stanie magistrali musi zostać podjęta w ciągu pojedynczych mikrosekund. Użyte moduły RFM12B są do tego zdecydowanie zbyt wolne (choćby ze względu na konieczność przełączania pomiędzy nadawaniem a odbiorem), toteż zdecydowano o emulacji urządzenia 1-wire po stronie odbiornika. Zanim przejdziemy

Tabela 3. Opis danych bloku *scratchpad*

Numer bajtu	Znaczenie
0	Temperatura – młodsza część
1	Temperatura – starsza część
2	Górny próg alarmu temperatury (z EEPROM)
3	Dolny próg alarmu temperatury (z EEPROM)
4	Bajt konfiguracyjny
5	Wartość stała = 255 (0xFF)
6	Zarezerwowany
7	Wartość stała = 16 (0x10)
8	CRC (suma kontrolna)

Tabela 1. Opis stanów modułu nadawczego sygnalizowanych diodami LED

Dioda	Sygnalizacja	Opis
LED1	Krótkie mignięcia	DS18B20 działa prawidłowo
	Świecenie ciągłe	Błąd komunikacji z DS18B20
LED2	Bardzo krótkie mignięcia	Praca nadajnika radiowego

Tabela 2. Opis stanów modułu odbiorczego sygnalizowanych diodami LED

Dioda	Sygnalizacja	Opis
LED1	Świecenie ciągłe	Brak sygnału z nadajnika
LED2	Bardzo krótkie mignięcia	Wysłanie <i>scratchpada</i> po magistrali 1-wire

Listing 3. Emulacja urządzenia podrzędnego, kluczowe fragmenty

```

//definicje wymagań czasowych dot. magistrali 1-wire [us]
#define TIME_RESET_PULSE 480
#define TIME_WAIT_AFTER_RESET_PULSE 30
#define TIME_PRESENCE_PULSE 120
#define TIME_ZERO_MIN 60
#define TIME_ZERO_MAX 120
#define TIME_ZERO_SEND 15

//definicje czasów odmierzanym Timerem1
#define TCNT1_SAMPLE 65535-30
#define TCNT1_WAIT_AFTER_RESET_PULSE 65535-30
#define TCNT1_PRESENCE_PULSE 65535-120

//definicje czasów timeout odmierzanym Timerem0
#define TIMEOUT_15MS 255-234

//definicje kolejnych stanów układu
//stan spoczynkowy - oczekiwanie na sygnał RESET
#define STATE_IDLE 0
//oczekiwanie na zakończenie impulsu RESET od mastera
#define STATE_RESET_WAIT 1
#define STATE_SEND_PRESENCE 2 //wysyłanie impulsu PRESENCE
#define STATE_AFTER_RESET 3 //oczekiwanie na rozkazy po resecie
#define STATE_SEND 4 //wysyłanie bajtów na magistralę
volatile uint8_t state = STATE_IDLE;

volatile uint8_t read_bit_cnt = 0; //licznik odczytanych bitów
volatile uint8_t read_byte = 0; //bajt tworzony z kolejnych bitów
volatile uint8_t read_byte_ready = 0; //gotowy, odczytany bajt
//licznik do odbioru kolejnych bajtów scratchpada
volatile uint8_t rec_cnt;

volatile uint8_t send_bit_cnt = 0; //numer bitu do wysyłki
volatile uint8_t send_byte_cnt = 0; //numer bajtu do wysyłki
volatile uint8_t send_scratch[10] = {0}; //kompletny scratchpad

// INT0 do obsługi 1-wire
ISR(INT0_vect)
{
    if(IS_L0){ //jeżeli jest stan niski na linii 1-wire
        //jeżeli trwa wysyłka bajtów scratchpada
        if(state == STATE_SEND){
            LED_STAT_1; //załącz diodę STAT
            //jeżeli ma zostać nadane zero
            if((send_scratch[send_byte_cnt] & (1<<send_bit_cnt)) == 0){
                WIRE_LO;
                TCNT1 = 0;
                //przeciagnij impuls do 15us
                while(TCNT1 < TIME_ZERO_SEND);
                WIRE_HI;
            }
            send_bit_cnt++; //przesun wysyłany bit o 1
            //po wysłaniu całego bitu, przejdź do następnego bajtu
            if(send_bit_cnt > 7){ send_byte_cnt++; send_bit_cnt = 0; }
            //po wysłaniu wszystkich bajtów, wróć do stanu IDLE
            if(send_byte_cnt > 8){ state = STATE_IDLE; LED_STAT_0; }
        }
        //w stanie spoczynku wyczyść Timer1 i
        if(state == STATE_IDLE){ TCNT1 = 0; LED_STAT_0; } wyłącz diodę STAT
        //jeżeli magistrala jest zainicjowana,
        //odmierzaj czas do zbierania próbek
        if(state == STATE_AFTER_RESET){ TCNT1 = TCNT1_SAMPLE; }
        TCNT0 = TIMEOUT_15MS; //wyzeruj licznik timeout
    }

    if(IS_HI){ //jeżeli jest stan wysoki
        //jeżeli wykryto impuls RESET o prawidłowej długości (od mastera)
        if(TCNT1 >= TIME_RESET_PULSE && state == STATE_IDLE){
            //przełącz stan na oczekiwanie do wygenerowania własnego
            state = STATE_RESET_WAIT; impulsu (presence)
            //oczekiwanie na wygenerowanie własnego impulsu
            TCNT1 = TCNT1_WAIT_AFTER_RESET_PULSE;
            //wyzeruj licznik odebranych bitów
            read_bit_cnt = 0;
            read_byte = 0; //wyzeruj odebrany bajt
            //wyzeruj licznik timeout
            TCNT0 = TIMEOUT_15MS;
        }
    }
}

// Tim1 do odmierzanego czasu nadawanych
//i odbieranych impulsów
ISR(TIM1_OVF_vect){
    //po pełnym resecie: układ jest gotowy do odbioru komendy
    if(state == STATE_AFTER_RESET){
        TCNT0 = TIMEOUT_15MS; //wyzeruj licznik timeout
        //master wysłał 1, jeżeli 0 to nie rób nic
        if(IS_HI){ read_byte |= (1<< read_bit_cnt); }
        //zainkrementuj licznik odebranych bitów
        read_bit_cnt++;
        //jeżeli zebrano cały bajt
        if(read_bit_cnt > 7){
            //wyzeruj licznik odebranych bitów
            read_bit_cnt = 0;
            //przekopiu gotowy bajt
            read_byte_ready = read_byte;
            //wyzeruj odebrany bajt
            read_byte = 0;
            //jeżeli odebrano komendę odczytu scratchpada
            if(read_byte_ready == 0xBE){
                state = STATE_SEND;
                send_bit_cnt = 0;
                send_byte_cnt = 0;
            }
        }
    }
    //po wysłaniu sygnału presence
    if(state == STATE_SEND_PRESENCE){

```

do omówienia kodów źródłowych, zastanówmy się nad zasadą działania tej magistrali. Można tu wyróżnić dwa typy przesyłanych informacji. Pierwszym jest inicjalizacja (*reset*) wszystkich urządzeń podrzędnych, którego przebieg czasowy jest widoczny na **rysunku 6**. Najpierw układ nadrzędny wymusza stan niski na linii, zaś potem czynią to układy podrzędne, po odczekaniu pewnego czasu. Taka „odpowiedź” jest dla układu nadrzędnego znakiem, że ma podłączonych działających „rozmówców”. Zatem należy sprawdzić stan linii 1-wire po min. 60 μ s (powinien być niski), a potem jeszcze raz, po min. 240 μ s – wtedy stan logiczny powinien wrócić do wysokiego. Cała procedura musi trwać min. 960 μ s.

Do przesyłu komend i danych stosowany jest inny rodzaj transmisji, szczegóły przedstawia **rysunek 7**. Jest różny w zależności od kierunku transmisji. Urządzenie nadrzędne zawsze rozpoczyna bit. Kiedy nadaje, długość trwania stanu niskiego determinuje znaczenie bitu:

- poniżej 15 μ s, jest to logiczna „1”
- od 60 μ s do 120 μ s, jest to logiczne „0”

Układy podrzędne sprawdzają stan linii po 15...60 μ s (typowo 30 μ s) od zarejestrowania zbocza opadającego. Krótki impuls zdąży się w tym czasie zakończyć i stan linii wyniesie powróci do „1”. Dłuższy impuls zostanie zaś zinterpretowany w tym momencie jako „0”.

Wysyłka bitów przez urządzenie podrzędne (poprzedzona odpowiednią komendą) polega na tym, że urządzenie nadrzędne wystawia zbocza opadające. Jeżeli układ podrzędny ma wysłać „1”, wówczas nie robi nic i po ok. 10...13 μ s od rozpoczęcia impulsu, stan logiczny linii zdąży powrócić do stanu wysokiego, co rejestruje układ nadrzędny. Jeżeli zaś ma zostać wysłane „0”, układ podrzędny „przeciaga” uchwycione zbocze opadające do 15 μ s, po czym zwalnia linię. Układ nadrzędny, który w tym czasie sprawdza stan linii, odczyta „0”. Co istotne, to układ nadrzędny decyduje o tym, ile bitów odczyta. Cały *scratchpad* składa się z 9 bajtów (**tabela 3**), lecz nic nie stoi na przeszkodzie, by odczytać tylko pierwsze dwa – tyle wystarczy do odczytu temperatury. Użyteczny może być również ostatni bajt – zawiera sumę kontrolną (*CRC*), dzięki której można dowiedzieć się, czy otrzymana informacja jest poprawna.

Obsługa magistrali 1-wire, którą trzeba było zaimplementować w nadajniku, jest dobrze znana i opisana w wielu miejscach. Ponieważ pracy nadajnika nie zaburzają jakiegokolwiek przerwania zewnętrzne, można było zrealizować ją za pomocą zwykłych opóźnień typu *delay()*. Długości trwania opóźnień dobrano doświadczalnie, za pomocą oscyloskopu, aby zgadzały się z tymi, których wymaga nota katalogowa układu. Kod został napisany w języku C, istotne fragmenty pokazuje **listing 2**.

Listing 3. cd.

```

//przejdź w tryb odbioru
state = STATE_AFTER_RESET;
WIRE_HI; //zwołnij linię
}
//po odczekaniu, nadaj sygnał presence
if(state == STATE_RESET_WAIT) {
    state = STATE_SEND_PRESENCE;
    TCNT1 = TCNT1_PRESENCE_PULSE;
    WIRE_LO;
}
}
// Tim0 do odmierzenia czasu timeout
ISR(TIM0_OVF_vect){
    //jeżeli doszło do przepełnienia tego
    //licznika, wróć do stanu IDLE
    state = STATE_IDLE;
}
int main(void) //pętla główna
{
    //przerwania od przepełnienia
    TIMSK1 |= (1 << TOIE1);
    //Timer1 odlicza co 1µs, preskaler 8
    TCCR1B |= (1 << CS11);
    TCNT1 = 0;
    //przerwania od przepełnienia
    TIMSK0 |= (1 << TOIE0);
    //Timer0 odlicza co 64µs, preskaler 256
    TCCR0B |= (1 << CS02);
    TCNT0 = 0;
    //przerwania zboczem narastającym i opadającym od INT0
    MCUCR |= (1 << ISC00);
    GIMSK |= (1 << INT0);

    sei();
}

```

Problem stanowiła emulacja urządzenia podrzędnego. Tutaj konieczne było użycie przerwania sprzętowego do wykrywania zboczy opadających. Ponadto, w przerwanach nie należy używać funkcji opóźniających, więc odmierzenie czasu trzeba było zrealizować z zastosowaniem liczników, zwłaszcza, że równocześnie może trwać odbiór danych z nadajnika. Rozwiązano to za pomocą prostej maszyny stanów, której najważniejsze fragmenty kodu znajdują się na **listingu 3**.

Widoczna jest obsługa przerwań, deklaracja zmiennych i stałych oraz inicjalizacja liczników i przerwania zewnętrznego.

W stanie spoczynku (*IDLE*), układ oczekuje na zbocze opadające. W momencie detekcji zerowany jest *Timer1*. Kiedy wystąpi zbocze narastające, sprawdzany jest stan licznika. Jeżeli był to prawidłowy impuls (tj. o długości min. 480 µs), układ przechodzi do generacji impulsu obecności (*presence*) po odczekaniu ustalonego czasu.

Odbiór danych jest możliwy po wykonaniu pełnej sekwencji inicjalizacji magistrali. Na zboczach opadających jest zadawana wartość licznika *Timer1*, który, w momencie przepełnienia, sprawdza stan linii 1-wire, odczytując w ten sposób wartość odebranego bitu. Jeżeli odebrana sekwencja bitów jest bajtem komendy odczytu zawartości *scratchpad*, układ przechodzi w stan gotowości do wysyłki danych. Każde zbocze opadające generowane przez urządzenie nadrzędne na linii 1-wire, jest analizowane pod kątem ewentualnego przedłużenia do długości odpowiadającej bitowi o wartości „0”.

Licznik *Timer0* odpowiada za automatyczny powrót układu do stanu spoczynku. Podczas komunikacji z układem nadrzędnym, jego wartość jest ustawiana tak, aby uległ przepełnieniu po 15 ms. Jeżeli na linii 1-wire nic się nie wydarzy, to układ przejdzie do stanu początkowego (*idle*), więc jego działanie można porównać do układu *watchdog*. Ten licznik przepełnia się cyklicznie również w stanie oczekiwania, więc przywróci układ do tego stanu w razie niespodziewanego pojawienia się stanu niskiego na linii – spowodowanego np. zakłóceniem.

Przedstawione programy można rozbudować np. o obsługę komendy porównania ROM (*Match ROM*), która stanowi bazę umożliwiającą prawidłowy odczyt temperatury z czujnika oraz emulację jego dwóch najważniejszych funkcji.

Michał Kurzela, EP

REKLAMA

Klub Aplikantów Próbek

to inicjatywa redakcji „Elektroniki Praktycznej”. W kontaktach z firmami redakcja często otrzymuje do przetestowania próbki podzespołów, modułów, a nawet całych urządzeń elektronicznych. Są to zwykle najnowsze typy/modele produktów na rynku. Z chęci podzielenia się z Czytelnikami tymi próbkami zrodziła się inicjatywa pod nazwą Klub Aplikantów Próbek. Członkiem KAP staje się każdy, kto zgłosi chęć przetestowania próbki. Wykaz i krótki opis próbek, którymi dysponuje redakcja EP, można znaleźć na stronie www.ep.com.pl/KAP.

Wystarczy wybrać rodzaj próbek i zwrócić się majlmem (na adres: SzeF Pracowni Konstrukcyjnej grzegorz.becker@ep.com).

pl) z prośbą o przesłanie bezpłatnych próbek, podając ich nazwę i adres wysyłki. Warto dopisać jaki jest plan zastosowania tych próbek.

Nie jest to konieczne, ale może mieć znaczenie przy podziale próbek w przypadku większej liczby zgłoszeń. Mile widziane, choć nieobowiązkowe, jest też przysłanie do redakcji EP opisu wykonanej aplikacji próbek, oczywiście po jej wykonaniu z zastosowaniem otrzymanej próbki. Autorom przysłanych opisów przyznamy punkty, które będą im dawały pierwszeństwo przy ubieganiu się o kolejne próbki. Najciekawsze opisy aplikacji opublikujemy na forum ep.com.pl lub na łamach „Elektroniki Praktycznej”.

Dla pełnej jasności jeszcze raz podkreślamy, że próbki przekazujemy bezpłatnie i nie trzeba ich zwracać do redakcji. Z uwagi na ograniczoną liczbę dostępnych próbek i niemałe zainteresowanie nimi, prosimy o opisanie swojego pomysłu na projekt na naszym forum internetowym, w dziale poświęconym Klubowi Aplikantów Próbek <https://forum.ep.com.pl/viewforum.php?f=80>. Ponadto, by zwiększyć swoje szanse na bycie wybranym do realizacji projektu w oparciu o nasze próbki, należy polubić fanpage Elektroniki Praktycznej na Facebooku (<https://facebook.com/ElektronikaPraktyczna>) oraz udostępnić post, w którym opisujemy rozdawane próbki. W przypadku podobnie interesujących pomysłów na projekty, będziemy uwzględniać to jako dodatkowe kryterium wyboru.



www.ep.com.pl/kap



VL53L1X – laserowy sensor

Czujnik odległości działający w technologii lidar

VL53L1X to miniaturowy laserowy dalmierz o zasięgu do 4 m i wysokiej powtarzalności pomiarów. Może znaleźć zastosowanie do pomiarów odległości, zliczania obiektów czy jako prosty detektor gestów. W artykule omówiono sposób działania czujnika oraz możliwości wykorzystania bibliotek i przykładowych programów.

Obudowa sensora VL53L1X ma wymiary 4,9×2,5×1,56 mm, a jej wygląd został pokazany na **fotografii 1**. Wewnątrz mieści się kompletna struktura sensora. Najważniejsze bloki funkcjonalne są zaprezentowane na **rysunku 2**. Dioda LED symbolizuje półprzewodnikowy laser podczerwony małej mocy, wysyłający impulsy pomiarowe. Po odbiciu od obiektu światło lasera pada na matrycę diodową (SPAD). Częścią czujnika jest mikrokontroler, który na podstawie różnicy czasu pomiędzy wyemitowaniem impulsu światła a jego powrotem do matrycy, oblicza odległość sensora od obiektu. Mikrokontroler reaguje na polecenia sterujące przesyłane magistralą I²C.



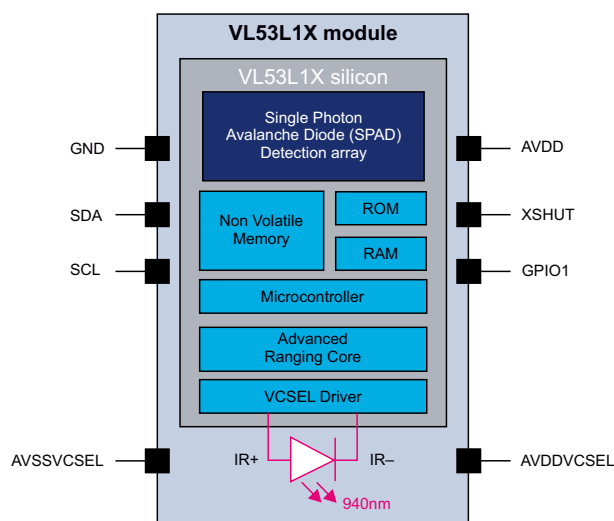
Fotografia 1. Wygląd sensora VL53L1X

Budowa matrycy odbiorczej umożliwia użycie sensora nie tylko do pomiaru odległości. Możliwe jest śledzenie ruchu obiektu i wykrywanie jego kierunku. Pozwala to wykorzystać sensor do zliczania przesuwających

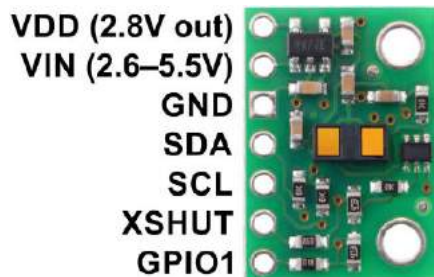
się w polu widzenia przedmiotów lub osób. Na tej samej zasadzie można zbudować prosty detektor gestów reagujący na ruch dłonią w prawą lub lewą stronę.

Najważniejsze parametry sensora VL53L1X:

- trzy zakresy pomiarowe dla odległości od 4 cm do 4 m
- maksymalne pole widzenia (FoV) wynosi 27°
- dioda laserowa klasy 1 (bezpieczna dla wzroku) emituje światło podczerwone o długości fali 940 nm



Rysunek 2. Wewnętrzna struktura sensora



Rysunek 3. Wyprowadzenia modułu VL53L1X Pololu

- do detekcji użyto matrycy 16×16 czułych fotodiod. Obszar odbiorczy matrycy można zmniejszyć (minimum 4×4 fotodiody) co zawęży pole widzenia FoV sensora
- komunikacja poprzez magistralę I2C
- napięcie zasilania od 2,6...3,5 V

Budowa

Do testów i eksperymentów najwygodniej posłużyć się modulem z zamontowanym sensorem VL53L1X, stabilizatorem i układami przesuwania poziomów. Na potrzeby artykułu skorzystano z modułu firmy Pololu (rysunek 3), oraz z płytki NUCLEO-F411RE. W tabeli 1 zestawiono wszystkie konieczne połączenia.

Dokumentacja i biblioteki

Ze względu na budowę sensora, w którym pracuje wewnętrzny procesor, do sterowania VL53L1X nie wykorzystuje się rejestrów ale procedury zebrane w specjalnej bibliotece. Pliki biblioteki producent, firma ST, zamieściła w spakowanym pakiecie STSW-IMG007 dostępnym do pobrania pod adresem [1]. Pakiet oprócz biblioteki zawiera dokument będący opisem (API) jej stosowania.

Dokumentację techniczną samego czujnika można pobrać pod adresem [2]. Przykładowe oprogramowanie wykorzystujące bibliotekę czujnika VL53L1X jest dostępne tu [3].

W pakiecie STSW-IMG009 znajdują się pliki prostego programu do pomiaru odległości, jakości oświetlenia i odczytania statusu. Oprócz tego pakiet zawiera „okrojona” na potrzeby projektu wersję biblioteki wraz z dokumentacją. Pakiet STSW-IMG010 zawiera przykładowy projekt pokazujący jak zastosować czujnik VL53L1X do zliczania obiektów albo ludzi. Program pozwala nie tylko zliczać obiekty, ale także określać kierunek ich ruchu. Na tej samej zasadzie można zbudować oprogramowanie do detekcji gestów, reagujące na ruch dłoni. Pliki biblioteki napisane są w języku C i można je zaadoptować dla różnych platform programistycznych i kontrolerów. Przykładowe programy jako bazę sprzętową wykorzystują zestaw firmowy ST do testowania czujnika czyli płytę VL53L1X Expansion Board (X-NUCLEO-53L1A1) oraz NUCLEO-F401. W dalszej części artykułu pokażę jak przystosować udostępnione przykłady oprogramowania do pracy z modulem sensora VL53L1X Pololu oraz dowolną płytką NUCLEO (w opisie użyję NUCLEO-F411).

Tworzenie szkieletu oprogramowania oraz dodanie biblioteki VL53L1XM

Szkielet oprogramowania zostanie utworzony przy pomocy konfiguratora STM32CubeMX dla środowiska programistycznego SW4STM32. Dalszy opis dotyczyć będzie płytki NUCLEO-F411, która komunikuje się z modulem VL53L1X Pololu za pośrednictwem magistrali I2C oraz zasilania modułu napięciem stabilizowanym 3,3 V. Wykorzystanie innego typu płytki NUCLEO lub DISCOVERY wymaga tylko zmiany ustawienia połączenia wraz z oznaczeniami portów i numerami złącz zebrano w tabeli 1.

Kolejne kroki z STM32CubeMX będą wyglądać następująco:

- Wybrać docelową płytkę, dla której ma być generowany kod, np. NUCLEO-F411 z zachowaniem predefiniowanych ustawień.
- Ustawienia zegarów: taktowanie z wewnętrznego generatora HSI RC 16 MHz, HCLK 84 MHz
- Na zakładce Pinout & Configuration → Connectivity zaznaczyć jako aktywną magistralę I2C1, pozostawiając domyślne ustawienia parametrów.
- Ustawić funkcję portu PB9 jako linię I2C1_SDA a portu PB8 jako linię I2C1_SCL.
- Ustawić funkcję portu PA4 jako GPIO_EXTI4 z etykietą VL53L1X_INT. Port można połączyć z wyjściem przerwań sensora. Jest to opcjonalna możliwość nie wykorzystywana w dalszych przykładach.
- Na zakładce Pinout & Configuration → Connectivity włączyć port UART2 z szybkością transmisji 115200 bodów. Tym portem będą wysyłane komunikaty tekstowe o pracy oprogramowania.
- Korzystając z zakładki Project Manager wybrać nazwę i katalog docelowy do zapisu wygenerowanych plików projektu oraz ustawić środowisko programistyczne np. SW4STM32.

Po wygenerowaniu szkieletu kodu otwieramy nasze środowisko programistyczne, w tym opisie będzie to SW4STM32. W kolejnych krokach do wygenerowanego kodu trzeba dodać pliki biblioteki sensora VL53L1X. Można przekopiować pliki biblioteczne

z jednego z udostępnionych przez firmę ST przykładów. Ja posłużyłem się przykładem STSW-IMG009. W dalszym opisie będą podawane położenia plików do przekopiowania z wewnętrznej struktury tego pakietu.

Kolejne kroki, które trzeba wykonać dodając pliki biblioteki VL53L1X:

- w drzewie projektu i katalogu Src tworzymy podkatalog dla biblioteki. Ja nadałem mu nazwę VL53L1X_Lib. Do utworzonego podkatalogu będą przekopiowywane kolejne pliki,
- z STSW-IMG009\Example\Src\plik vl53l1_platform.c,
- z STSW-IMG009\Example\Inc\pliki: vl53l1_platform.h, vl53l1_error_codes.h, vl53l1_types.h, vl53l1_platform_log.h, vl53l1_platform_user_config.h,
- z STSW-IMG009\API\core\pliki: vl53l1_api.c, vl53l1_api.h.

Należy dokonać zmian w zaimportowanym pliku vl53l1_platform.c. Zależnie od użytej płytki trzeba dostosować deklarację typu bibliotek HAL. W przypadku NUCLEO-F411 zmiana będzie wyglądała tak:

```
z: #include „stm32xxx_hal.h” na:
#include „stm32f4xx_hal.h”
```

Usunąć lub zakomentować:

```
//#include „X-NUCLEO-53L1A1.h”
```

Ostatnia konieczna zmiana polega na wskazaniu I2C1 jako magistrali przypisanej do komunikacji z sensorem VL53L1X. Należy zastąpić:

```
#define VL53L0X_pI2CHandle (&hi2c1)
```

na:

```
extern I2C_HandleTypeDef hi2c1;
#define XNUCLEO53L1A1_hi2c hi2c1
```

Na koniec należy podać w ustawieniach kompilacji ścieżkę dostępu do utworzonego katalogu biblioteki VL53L1X_Lib. W środowisku SW4STM32 robi to się poprzez Properties → Settings → MCU GCC Compiler → Includes → Include Paths.

W dalszym kroku możemy do tak utworzonego projektu dodawać kod pobrany np. z przykładowych projektów.

Odczyt danych z sensora

Korzystając z przykładu STSW-IMG009 można zastosować sensor VL53L1X do pracy

Tabela 1. Opis połączeń pomiędzy czujnikiem i płytką Nucleo

VL53L1X Pololu	Nucleo	Opis
VIN	AVDD (3,3V) CN10-7	Zasilanie +3,3 V
GND	GND CN10-9	Masa
SDA	PB9 CN10-5	Linia SDA magistrali I2C
SCL	PB8 CN10-3	Linia SCL magistrali I2C
GPIO1	PA4 CN7-32	Wyjście przerwania z czujnika (do opcjonalnego wykorzystania)

Listing 1. Deklaracja niezbędnych zmiennych i prototyp funkcji printf

```

/* USER CODE BEGIN PV */
uint16_t dev=0x52;
int status=0;
uint8_t byteData;
uint16_t wordData, Distance,
uint16_t SignalRate, AmbientRate, SpadNum;
uint8_t RangeStatus, dataReady;
/* USER CODE END PV */
/* USER CODE BEGIN 0 */
//int fputc(int ch, FILE *f)
#define PUTCHAR_PROTOTYPE int __io_putchar (int ch)

PUTCHAR_PROTOTYPE
{
    HAL_UART_Transmit(&huart2, (uint8_t*)&ch, 1, 0xFFFF);
    return ch;
}
/* USER CODE END 0 */

```

Listing 2. Inicjalizacja czujnika

```

/* Those basic I2C read functions can be used */
/* to check your own I2C functions */
status = VL53L1X_RdByte(dev, 0x010F, &byteData);
printf("VL53L1X Model_ID: %X\n", byteData);
status = VL53L1X_RdByte(dev, 0x0110, &byteData);
printf("VL53L1X Module_Type: %X\n", byteData);
status = VL53L1X_RdWord(dev, 0x010F, &wordData);
printf("VL53L1X: %X\n", wordData);
while(sensorState==0)
{
    status = VL53L1X_BootState(dev, &sensorState);
    HAL_Delay(2);
}
printf("Chip booted\n");
/* This function must be called */
/* to initialize the sensor with the default setting */
status = VL53L1X_SensorInit(dev);
/* Optional functions to be used */
/* to change the main ranging parameters */
/* according the application requirements */
/* to get the best ranging performances */
/* 1=short, 2=long */
status = VL53L1X_SetDistanceMode(dev, 2);
/* in ms possible values [20, 50, 100, 200, 500] */
status = VL53L1X_SetTimingBudgetInMs(dev, 100);
/* in ms, IM must be >= TB */
status = VL53L1X_SetInterMeasurementInMs(dev, 100);
/* offset compensation in mm */
// status = VL53L1X_SetOffset(dev, 20);
/* minimum ROI 4, 4 */
// status = VL53L1X_SetROI(dev, 16, 16);
/* may take few second to perform the offset cal */
status = VL53L1X_CalibrateOffset(dev, 140, &offset);
/* may take few second to perform the xtalk cal */
// status = VL53L1X_CalibrateXtalk(dev, 1000, &xtalk);
printf("VL53L1X Ultra Lite Driver Example running ...n");
/* This function has to be called to enable the ranging */
status = VL53L1X_StartRanging(dev);

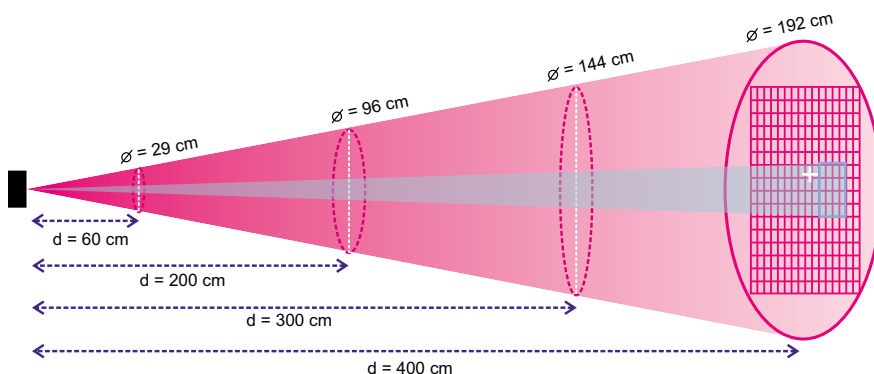
```

Listing 3. Odczyt danych z czujnika

```

/* USER CODE BEGIN WHILE */
while (1)
{
    while (dataReady == 0){
        status = VL53L1X_CheckForDataReady(dev, &dataReady);
        HAL_Delay(2);
    }
    dataReady = 0;
    status = VL53L1X_GetRangeStatus(dev, &RangeStatus);
    status = VL53L1X_GetDistance(dev, &Distance);
    status = VL53L1X_GetSignalRate(dev, &SignalRate);
    status = VL53L1X_GetAmbientRate(dev, &AmbientRate);
    status = VL53L1X_GetSpadNb(dev, &SpadNum);
    /* clear interrupt has to be called to enable next interrupt */
    status = VL53L1X_ClearInterrupt(dev);
    printf("%u, %u, %u, %u, %u\n", RangeStatus, Distance,
    SignalRate, AmbientRate, SpadNum);
    /* USER CODE END WHILE */
}

```



Rysunek 4. Pole widzenia sensora

jako dalmierz i poprzez port UART2 wysyłać sformatowane jako tekst dane pomiarowe. Na **listingu 1** pokazana została deklaracja niezbędnych zmiennych i prototyp funkcji printf do wysyłania danych portem UART2. **Listing 2** rozpoczyna się testem podłączenia sensora realizowanym poprzez wysłanie zapytania o jego typ. Następne komendy służą do inicjacji sensora w trybie długiego zasięgu (dystans do 4 metrów), na koniec włączana jest funkcja cyklicznych pomiarów. Funkcje kalibracji zostały zakomentowane ponieważ sensor powinien być skalibrowany na etapie produkcji i bez zmiany warunków jego użytkowania, powtórna kalibracja nie jest potrzebna. Na **listingu 3** w pętli głównej programu odbywa się ciągły odczyt z sensora. Najpierw program oczekuje, aż sensor będzie gotów przesłać nowe dane. Potem w serii kolejnych pobrań odczytywane są kolejne parametry: statusu pomiarów, zmierzona odległość w mm, parametr obrazujący jakość sygnału, poziom oświetlenia, ilość aktywnych diod matrycy SPAD. Na koniec odczytane dane są formatowane i wysyłane portem UART2. Następnie cały cykl rozpoczyna się na nowo.

Dwie strefy odczytu

Przykład zawarty w pakiecie STSW-IMG010 pokazuje jak stworzyć oprogramowanie do zliczania obiektów lub do detekcji gestów – przesuwania dłoni. Działanie programu wiąże się z dwu strefowym odczytem matrycy SPAD, więc zanim przejdziemy do listingów warto napisać kilka słów o tej opcji działania sensora. Zagadnienie dokładnie opisuje dokument UM2555, który można znaleźć na stronach firmy ST.

Na **rysunku 4** pokazano jak w funkcji odległości zmienia się pole widzenia sensora. Jeżeli chcemy to pole zawęzić zmniejszamy ilość aktywnych fotodiod, przez określenie wymiarów zwartego prostokątnego obszaru, które mają utworzyć. Najmniejszy dopuszczalny obszar to 4×4. Określenie rozmiarów nowego aktywnego obszaru odbywa się poprzez wywołanie funkcji API VL53L1X_SetROI(). Pokazana na **rysunku 5** tabela odzwierciedla lokalizację wszystkich fotodiod matrycy oraz ich numerację. Zaznaczony na żółto prostokąt oznacza aktywną strefę składającą się z 6×11 fotodiod. Teraz przy pomocy funkcji VL53L1X_SetROICenter() należy określić pozycję diody znajdującej się w centrum. Na **rysunku 5** jest to fioletowy obszar wokół numeru fotodiody 223. Jeżeli geometryczny środek obszaru przypada pomiędzy rzędami, należy podać numer elementu znajdującego się bezpośrednio ponad i bezpośrednio z prawej strony pozycji środkowej. Od tej chwili sensor reaguje tylko na impulsy świetlne docierające do tej strefy. Jeżeli teraz przy pomocy funkcji VL53L1X_SetROICenter() zmienimy lokalizację punktu

```
/* Table of SPAD locations. Each SPAD has a number which is not obvious.
*
* 128,136,144,152,160,168,176,184, 192,200,208,216,224,232,240,248
* 129,137,145,153,161,169,177,185, 193,201,209,217,225,233,241,249
* 130,138,146,154,162,170,178,186, 194,202,210,218,226,234,242,250
* 131,139,147,155,163,171,179,187, 195,203,211,219,227,235,243,251
* 132,140,148,156,164,172,180,188, 196,204,212,220,228,236,244,252
* 133,141,149,157,165,173,181,189, 197,205,213,221,229,237,245,253
* 134,142,150,158,166,174,182,190, 198,206,214,222,230,238,246,254
* 135,143,151,159,167,175,183,191, 199,207,215,223,231,239,247,255

* 127,119,111,103, 95, 87, 79, 71, 63, 55, 47, 39, 31, 23, 15, 7
* 126,118,110,102, 94, 86, 78, 70, 62, 54, 46, 38, 30, 22, 14, 6
* 125,117,109,101, 93, 85, 77, 69, 61, 53, 45, 37, 29, 21, 13, 5
* 124,116,108,100, 92, 84, 76, 68, 60, 52, 44, 36, 28, 20, 12, 4
* 123,115,107, 99, 91, 83, 75, 67, 59, 51, 43, 35, 27, 19, 11, 3
* 122,114,106, 98, 90, 82, 74, 66, 58, 50, 42, 34, 26, 18, 10, 2
* 121,113,105, 97, 89, 81, 73, 65, 57, 49, 41, 33, 25, 17, 9, 1
* 120,112,104, 96, 88, 80, 72, 64, 56, 48, 40, 32, 24, 16, 8, 0 - Pin 1*/
```

Rysunek 5. Tabela SPAD pozycji fotodiod

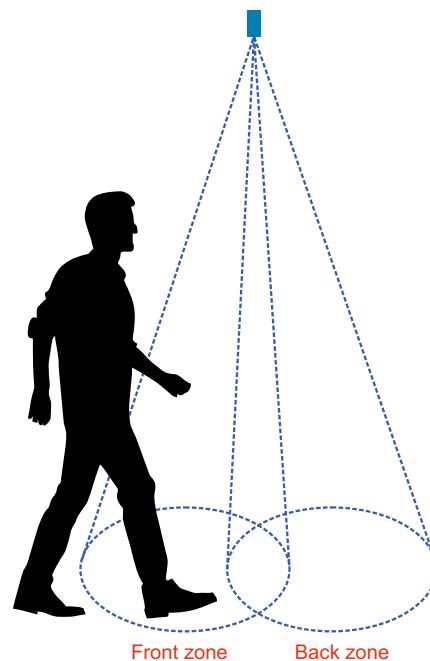
centralnego np. na 159, sensor będzie reagował na impulsy świetlne z obszaru 6×11 fotodiod wokół nowego punktu centralnego drugiej strefy. Dokonując przełączeń między dwiema lokalizacjami punktu centralnego, sensor naprzemiennie kontroluje sytuację w dwu strefach.

Użycie sensora do zliczania osób

Na **rysunku 6** pokazano jak użyć sensor VL53L1X do zliczania liczby osób. Sensor pracuje z dwiema strefami odczytu. Każdy przekraczający strefy w kolejności pierwsza-druga zostanie zliczony jako osoba wchodząca. Z kolei przekraczając strefy w kolejności druga-pierwsza, osoba zostanie zliczona jako wychodząca.

Inicjacja sensora jest podobna do tej z poprzedniego przykładu. Na **listingu 4** pokazano główne różnice: wprowadzenie tablicy center[2] do przechowania dwu lokalizacji punktów centralnych stref, zmiana parametrów czasowych oraz zaprogramowanie wymiarów stref.

Pętla główna programu pokazana jest na **listingu 5**. Tak jak w poprzednim przykładzie odczyt danych z sensora następuje, gdy zgłosi on gotowość. Po odczycie odległości następuje wyzerowanie przerwania i po krótkiej pauzie ustawiana jest nowa pozycja punktu środkowego drugiej strefy. Następnie wywoływana jest procedura ProcessPeopleCountingData(), której algorytm badania obecności obiektu w obu strefach określa, w którą stronę obiekt poruszał się i odpowiednio modyfikuje stan licznika PplCounter. Po uaktualnieniu licznika stref Zone i wysłaniu portem UART2 informacji o stanie operacji i zawartości licznika PplCounter, program wraca na początek pętli głównej.



Rysunek 6. Zliczanie osób

W tym przykładzie UART został zaprogramowany do pracy z szybkością 460800 baudów. Program może reagować i odpowiednio zliczać ruchy dłoni o ile gesty nie są wykonywane na tle sylwetki. W czasie testów zauważyłem, że na działanie sensora negatywnie wpływa intensywne oświetlenie. Według dokumentacji sensor jest mniej podatny na zakłócenia gdy wybierze się mniejszy zasięg

jego działania i ograniczy się intensywność oświetlenia.

Ryszard Szymaniak

Linki:

- [1] <http://bit.ly/2mvNjSD>
- [2] <http://bit.ly/2m9vGHR>
- [3] <http://bit.ly/2kZlhzg>

```
Listing 4. Przygotowanie do odczytu danych z dwóch stref
/* USER CODE BEGIN PV */
/* these are the spad center of the 2 8*16 zones */
int center[2] = {167,231};
...
/* USER CODE END PV */
...
/* in ms possible values [20, 50, 100, 200, 500] */
status += VL53L1X_SetTimingBudgetInMs(dev, 20);
status += VL53L1X_SetInterMeasurementInMs(dev, 20);
/* minimum ROI 4,4 */
status += VL53L1X_SetROI(dev, 8, 16);
if (status != 0) {
    printf("Error in Initialization or configuration of the device\n");
    return (-1);
}
printf("Start counting people ... \n");
```

```
Listing 5. Główna pętla programu realizującego funkcję zliczania osób
/* USER CODE BEGIN WHILE */
while (1){
    while (dataReady == 0) {
        status = VL53L1X_CheckForDataReady(dev, &dataReady);
        HAL_Delay(2);
    }

    dataReady = 0;
    status += VL53L1X_GetRangeStatus(dev, &RangeStatus);
    status += VL53L1X_GetDistance(dev, &Distance);
    /* clear interrupt has to be called to enable next interrupt*/
    status += VL53L1X_ClearInterrupt(dev);
    if (status != 0) {
        printf("Error in operating the device\n");
        return (-1);
    }

    HAL_Delay(10);
    status = VL53L1X_SetROI(dev, center[Zone]);
    if (status != 0) {
        printf("Error in changing the center of the ROI\n");
        return (-1);
    }

    // inject the new ranged distance in the people counting algorithm
    PplCounter = ProcessPeopleCountingData(Distance, Zone);
    Zone++;
    Zone = Zone%2;
    printf("%d, %d, %d, %d, %d\n", Zone, RangeStatus, Distance, PplCounter);
    /* USER CODE END WHILE */
}
```

Sterownik wejść mostka H

Zastosowanie silników prądu stałego o mocy większej niż przewidział producent danego urządzenia, może zakończyć się niepowodzeniem z uwagi na niedostateczną wydajność prądową wbudowanego w urządzenie mostka H. Przedstawiony układ „konwertuje” sygnały z wyjścia mostka H na postać przystosowaną do sterowania innym tego typu układem, o większej mocy.

Aby dobrze wyjaśnić zastosowanie układu warto opisać praktyczny przykład jego zastosowania.

Odbiornik radiowy zdalnego sterowania, który został pozyskany z zabawki, ma posłużyć do budowy znacznie większego pojazdu. W odbiorniku mostek H, sterujący oryginalnym silnikiem, ma niewielką wydajność prądową, a do tego został zintegrowany z głównym układem scalonym, więc wyodrębnienie poszczególnych sygnałów sterujących jest niemożliwe. Opisany układ „zamieni” dwuprzewodowy sygnał, który zasilal oryginalny silniczek, na tróprzewodowy, który może wysterować mostek H o większym dopuszczalnym prądzie, np. L298N.

Dodatkowe materiały do pobrania ze strony www.media.avt.pl

W ofercie AVT* AVT-5723

Podstawowe parametry:

- wejścia przystosowane do wykrywania kierunku obrotów zadanych mostkiem H,
- wyjścia typu TTL (0/5 V) do sterowania mostkiem H: A, B, ENABLE,
- maksymalna częstotliwość pracy ok. 50 kHz
- zasilanie stałym napięciem 8...35 V
- pobór prądu ok. 7 mA

Wykaz elementów:

R1, R2, R6...R8, R12: 3,3 kΩ SMD0805
R3, R4, R9, R10: 1 kΩ SMD0805 (opis w tekście)
R5, R11: 1 MΩ SMD0805
C1, C4: 47 μF/25 V THT raster 2 mm
C2, C3, C5...C7: 100 nF SMD0805
D1, D2: Zener 3,3 V 0,5 W MiniMELF
D3: SS14 lub podobna
US1: LM393 S08
US2, US3: CD4093 S014
US4: 78M05 DPAK
J1: goldpin 3 pin 2,54 mm THT
J2: goldpin 5 pin 2,54 mm THT

* Uwaga! Elektroniczne zestawy do samodzielnego montażu wymagają umiejętności lutowniczych!

Podstawowa wersja zestawu jest wersja [B] nazywana potocznie KIT-em (z ang. zestaw). Zestaw w wersji [B] zawiera elementy elektroniczne (w tym UK) – jeśli występuje w projekcie), które należy samodzielnie wylutować w dołączoną płytkę drukowaną (PCB). Wykaz elementów znajduje się w dokumentacji, która jest podlinkowana w opisie kitu. Mając na uwadze różne potrzeby naszych klientów, oferujemy dodatkowe wersje:

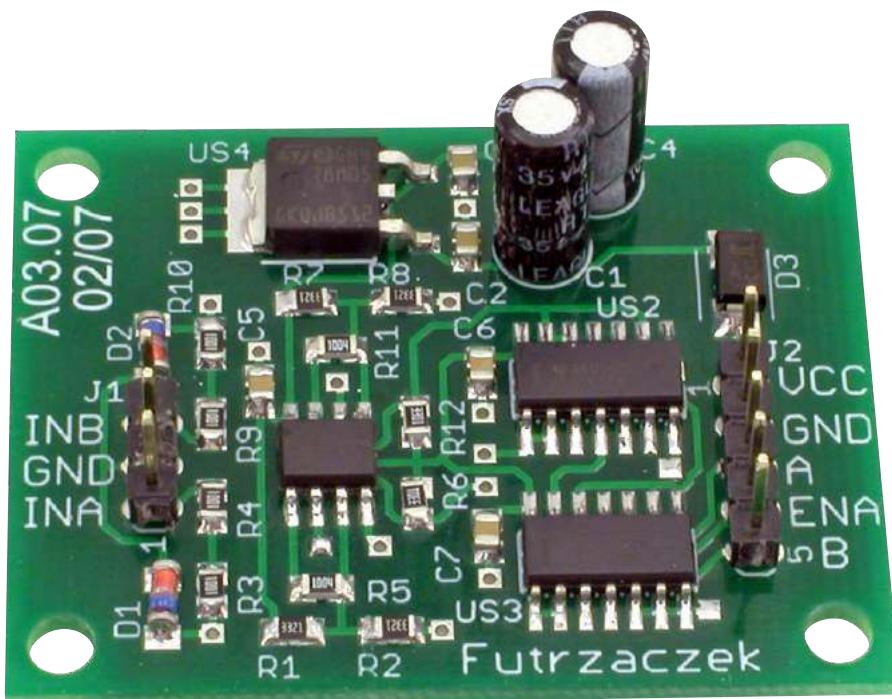
- wersja [C] – zmontowany, uruchomiony i przetestowany zestaw [B] (elementy wylutowane w płytkę PCB)
- wersja [A] – płytkę drukowaną bez elementów i dokumentacji

Kity w których występuje układ scalony wymagający zaprogramowania, mają następujące dodatkowe wersje:

- wersja [A*] – płytkę drukowaną [A] + zaprogramowany układ [UK] i dokumentacja
- wersja [UK] – zaprogramowany układ

Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik pdf! Podczas składania zamówienia upewnij się, którą wersję zamawiasz!

<http://sklep.avt.pl>. W przypadku braku dostępności na <http://sklep.avt.pl>, osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: kity@avt.pl.

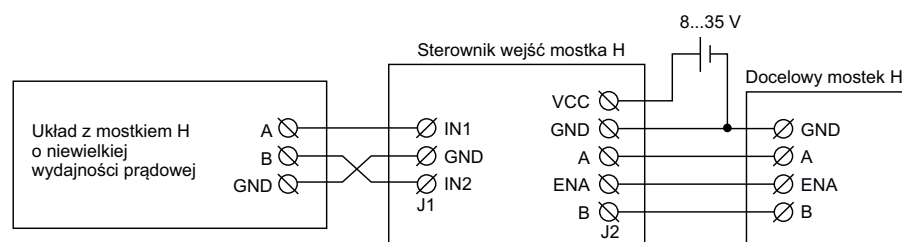


Schemat blokowy podłączenia sterownika prezentuje **rysunek 1**. Wychodzące z niego sygnały to A, B, ENABLE (kierunek + aktywacja wyjść). **Tabela 1** przedstawia tablicę prawdy, która wiąże stany logiczne wejść i wyjść układu. Szczególną uwagę warto zwrócić na sytuację w najniższym wierszu, kiedy na obu wejściach pojawia się napięcie. Taki stan jest niedopuszczalny dla niektórych mostków i zostanie potraktowany tak samo, jak brak napięcia na obu wejściach.

Budowa

Schemat ideowy układu znajduje się na **rysunku 2**. Wyodrębnić można dwa komparatory oraz prosty blok cyfrowy, złożony z kilku bramek NAND.

Komparatory, zawarte w strukturze układu LM393, mają za zadanie określić, czy napięcie wchodzące do układu z mostka H można określić jako „stan niski”, czy też „wysoki”. Napięcie odniesienia to podzielone na pół stabilizowane napięcie zasilające część cyfrową, czyli ok. 2,5 V. Rezystory 1 MΩ w pętli dodatniego sprzężenia zwrotnego wytwarzają niewielką histerezę, co zapobiega powstawianiu niepożądanych oscylacji przy wolnozmiennych zboczach sygnału wejściowego. Diody Zenera ograniczają napięcie wejściowe do wartości ok. 3,3 V. Daje to margines równy ok. 1,7 V od napięcia zasilającego te komparatory. Taki margines jest wymagany przez producenta tego układu w jego notcie katalogowej – maksymalna



Rysunek 1. Schemat blokowy podłączenia sterownika

Tabela 1. Tablica prawdy, która wiąże stany logiczne wejść i wyjść układu

Wejście INA	Wejście INB	Wyjście A	Wyjście B	Wyjście ENA
L	L	L	L	L
L	H	L	H	H
H	L	H	L	H
H	H	L	L	L

H – stan wysoki (na wejściu: powyżej 3,2 V, na wyjściu: ok. 5 V)

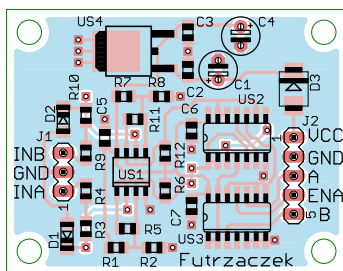
L – stan niski (na wejściu: poniżej 2 V, na wyjściu: ok. 0 V)

wartość napięcia podawanego na wejście nie może być wyższa niż napięcie zasilania pomniejszone o 1,5 V.

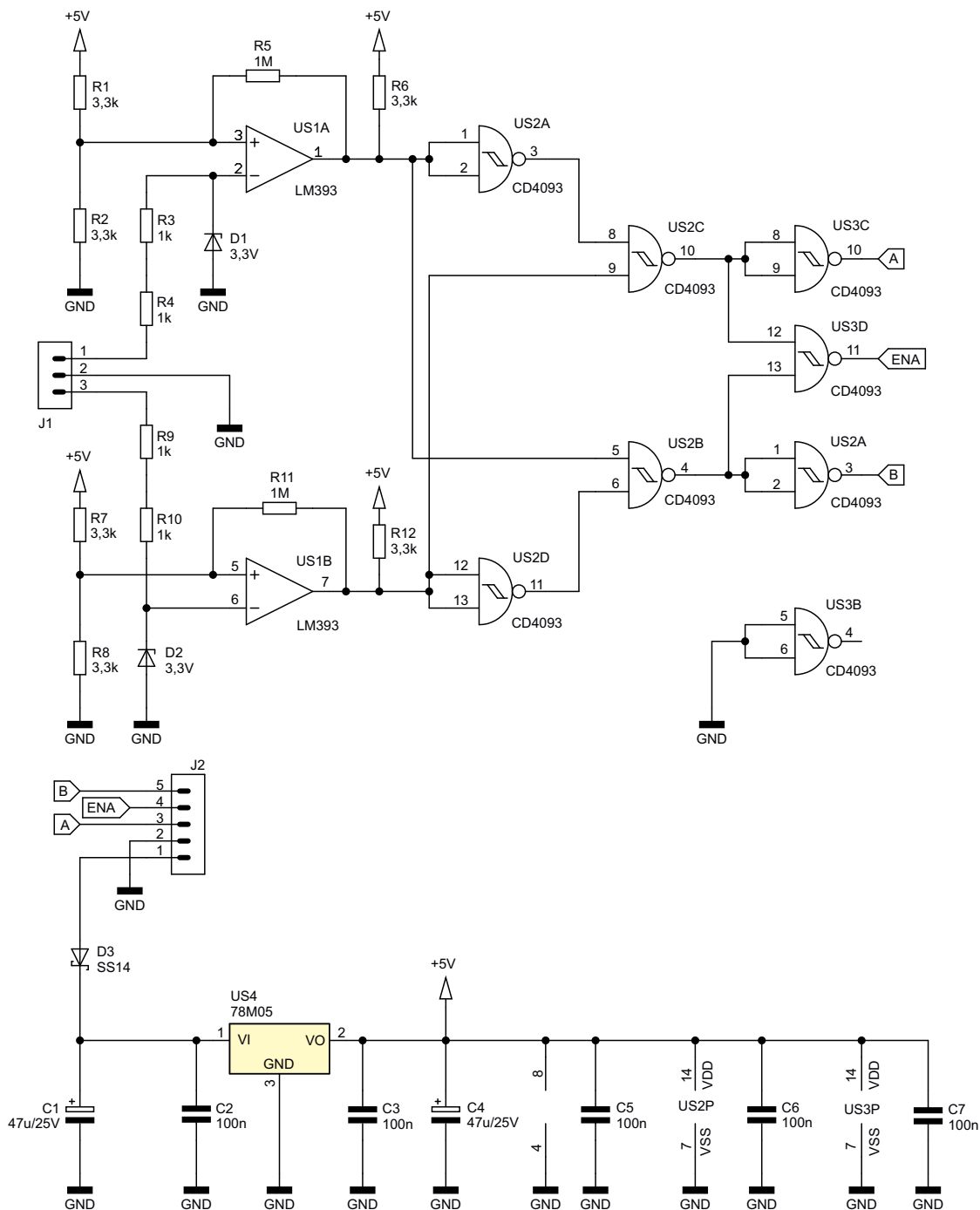
Wartości elementów zostały tak dobrane, że minimalne napięcie, które jest interpretowane jako „wysokie”, wynosi ok. 3,2 V. Sterujący układem mostek H może mieć relatywnie wysoki spadek napięcia w „niskim” stanie wyjścia – spowodowany obecnością np. układów mierzących prąd obciążenia – dlatego przyjęto pewien „zapas” napięcia. Z kolei, stan „niski” jest rozpoznawany przy napięciu ok. 2 V lub mniej.

Maksymalne napięcie, jakie może być podane na zaciski złącza J1, jest zdeterminowane głównie przez maksymalną moc, jaką można wytracić na rezystorach ograniczających prądy diod Zenera. Przyjmując 0,125 W na jeden rezystor w obudowie 0805, daje to maksymalne napięcie wejściowe ok. 25 V, co powinno wystarczyć w większości sytuacji. Gdyby jednak zaszła konieczność sprzęgnięcia układu ze źródłem sygnału o wyższym napięciu wyjściowym, należy zwiększyć odpowiednio ich rezystancję.

Wyjścia komparatorów są typu „otwarty kolektor”, dlatego rezystory R6 i R12 wymuszają na nich potencjał 5 V w momencie zatkania wbudowanych tranzystorów.



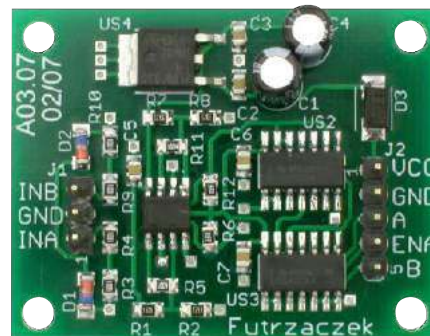
Rysunek 3. Schemat płytki PCB wraz z rozmieszczeniem elementów (skala 1:1)



Rysunek 2. Schemat ideowy układu

Bramki cyfrowe NAND mają za zadanie załączyć tylko jedno wyjście sterujące kierunkiem (A lub B) w sytuacji, gdy tylko jedno wejście jest w stanie wysokim. Dwa takie same poziomy napięć na wejściach nie powodują uaktywnienia wyjść. Dodatkowo, układ steruje wyjściem aktywującym mostek H (ENA) w sytuacji, gdy którekolwiek z wyjść – A lub B – jest w stanie wysokim.

Napięcie zasilające jest stabilizowane i wynosi 5 V. Dzięki zastosowaniu „nadmiarowego” stabilizatora, układ może być zasilany z napięcia o wartości ok. 35 V bez dodawania jakichkolwiek radiatorów. Dioda D3 chroni układ przed zniszczeniem w razie omyłkowej zamiany biegunów źródła zasilającego.



Fotografia 4. Widok zmontowanej płytki

Montaż i uruchomienie

Układ został zmontowany na dwustronnej płytce drukowanej o wymiarach 35×45 mm, której schemat montażowy i wzór ścieżek

przedstawia **rysunek 3**. Większość elementów montowana jest powierzchniowo i to od nich należy rozpocząć lutowanie. Złącza typu gold-pin i kondensatory elektrolityczne warto zostawić na koniec. Gotowy układ pokazuje **fotografia 4**. Otwory montażowe znajdują się w odległości 3 mm od krawędzi płytki.

Prawidłowo zmontowany układ jest od razu gotowy do działania i nie wymaga jakichkolwiek czynności uruchomieniowych.

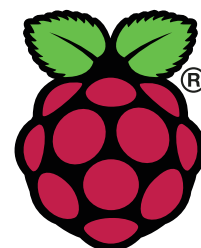
Napięcie zasilania powinno wynosić od 8 V do 35 V, a pobór prądu (przy nieobciążonych wyjściach) to ok. 7 mA.

Dolna częstotliwość pracy jest równa zero – układ może trwać w ustalonej konfiguracji dowolnie długi czas. Natomiast górna częstotliwość, przy której wypełnienie impulsów jest przenoszone poprawnie, to ok. 50 kHz. Ta informacja jest istotna z punktu widzenia zastosowania PWM do regulacji mocy silnika.

Jeżeli zachodzi konieczność współpracy ze źródłem sygnału, które już po załączeniu wyjścia daje napięcie mniejsze od wspomnianej wcześniej wartości 3,2 V (ponieważ jest np. zasilane z dwóch baterii AA), można zmodyfikować progi przerzutu wejściowych komparatorów poprzez odpowiednie zmniejszenie wartości R2 i R8.

Michał Kurzela, EP

Przetwornik DAC z końcówką mocy dla Raspberry Pi



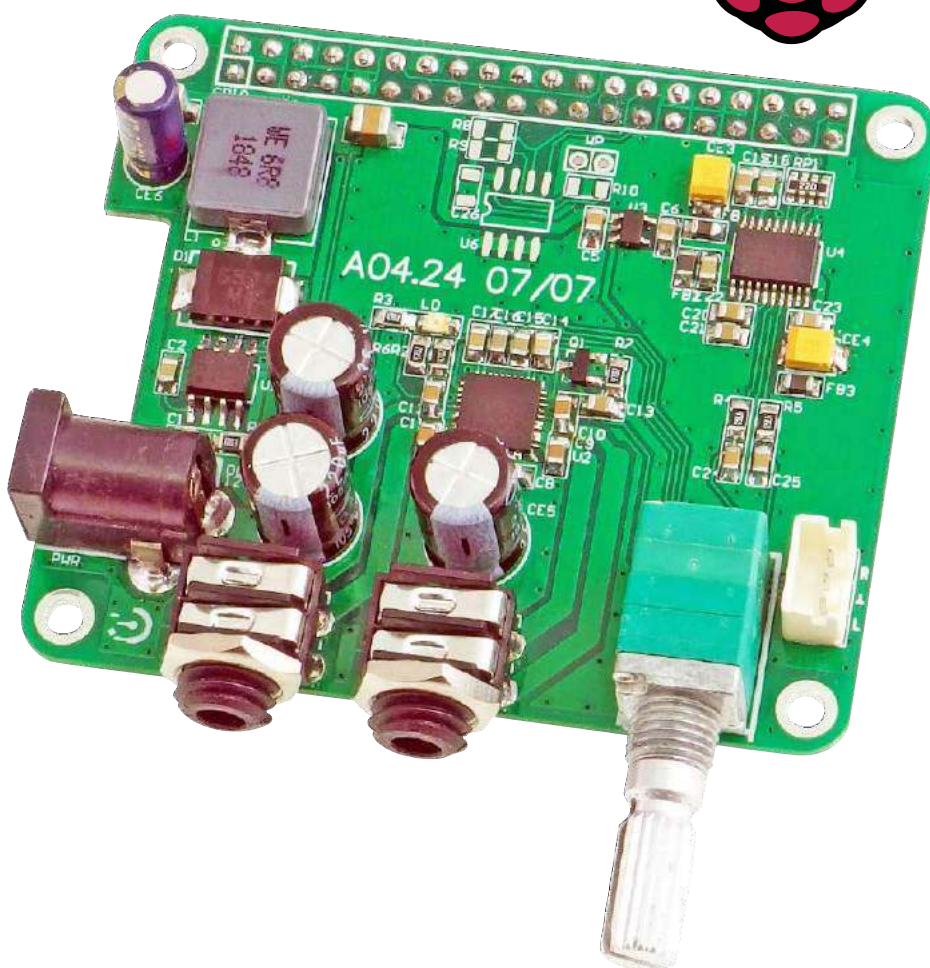
Przedstawiony moduł HAT posiada wbudowany przetwornik DAC audio oraz końcówkę mocy $2 \times 8\text{ W}$ ($8\ \Omega$) ułatwiającą realizację wbudowanych projektów audio takich jak odtwarzacze multimedialne lub info-kioski, bez potrzeby składania aplikacji z kilku oddzielnych modułów. HAT posiada także wbudowany zasilacz 5 V/1,5 A dla Raspberry co umożliwia zastosowanie jednego wspólnego zasilania 12 V dla całego systemu.

Budowa i działanie

Schemat modułu prezentuje **rysunek 1**. Moduł składa się z trzech bloków funkcjonalnych: przetwornicy ADP2302 obniżającej napięcie zasilania z 12 V na 5 V dla zasilania Raspberry Pi, cyfrowej stereofonicznej końcówki mocy SSM3302, układu DAC audio typu PCM5100A. Opcjonalnie na płycie pozostawiono miejsce na pamięć 24LC32 dla spełnienia wymogów nakładki HAT.

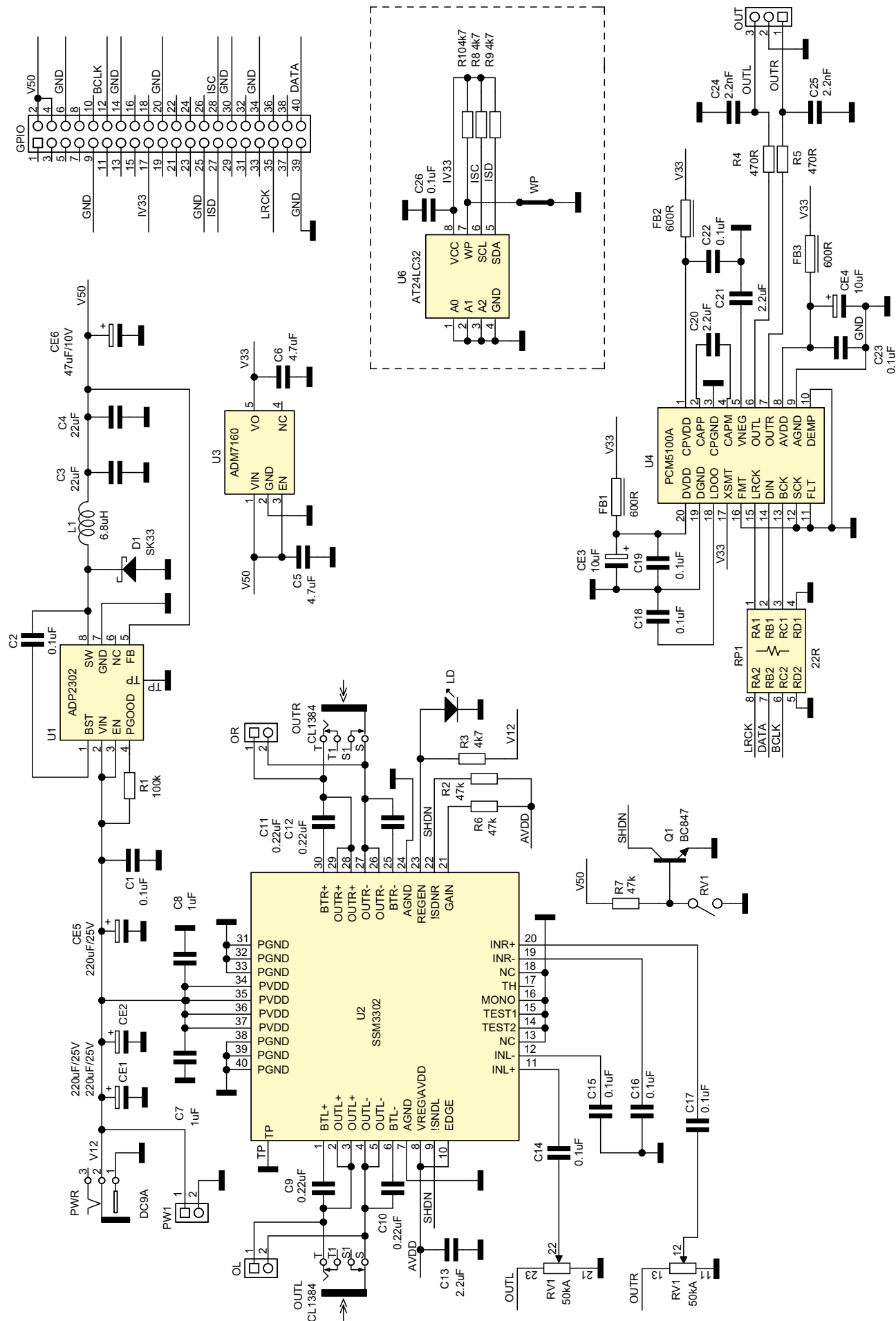
Moduł wymaga zasilania napięciem z przedziału 9...15 V o mocy min. 25 W przez złącze PWR. Elementy CE1, CE2, CE5 filtrują napięcie a przetwornica U1 typu ADP2302 zapewnia na wyjściu napięcie 5 V, które poprzez złącze GPIO zasila Raspberry PI. Wydajność 5 V ustalona jest na maksimum, czyli 1,5 A co zaspokaja potrzeby komputerka w większości zastosowań. Z napięcia 5 V, zasilany jest stabilizator U3 typu ADM7160, dostarczający napięcia 3,3 V dla przetwornika DAC audio. Napięcie 5 V jest również wykorzystane w obwodzie odwracającego klucza Q1, sterującego wyłączeniem końcówki mocy ze styku wbudowanego w potencjometr głośności.

Układ stereofonicznej końcówki mocy U2 typu SSM3302, zasilany jest bezpośrednio z napięcia doprowadzonego do złącza



PWR. Moc wyjściowa układu przy zasilaniu 12 V wynosi $2 \times 8\text{ W}$ na obciążeniu $8\ \Omega$ przy zniekształceniach poniżej 1%. Dzięki pracy wzmacniacza w klasie D i konfiguracji mostkowej, zbędne są kondensatory sprzęgające w obwodzie głośnika. Zastosowany tryb modulacji przy krótkich przewodach (do 20 cm) nie wymaga stosowania dolno-przepustowego filtra wyjściowego. Wysoka sprawność SSM3302 dochodząca do 91% w sporej ilości aplikacji pozwala na rezygnację z radiatora. Wzmocniony sygnał zasilający głośniki dostępny jest na gniazdach

mini jack 3,5 mm (OUTL, OUTR). Kondensatory C9...C12 są elementami obwodów polaryzacji tranzystorów końcowych, C7, C8, C13 odsprężają zasilanie. Dioda LD sygnalizuje obecność zasilania i polaryzuje wejście aktywujące wbudowany stabilizator AVDD. Wejściowy sygnał audio separowany jest kondensatorami C14 i C17. Różnicowe wejścia audio SSM3302 pracują w trybie niesymetrycznym, wyprowadzenia wejść odwracających połączone są z masą układu kondensatorami C15 i C16. Poziom sygnał regulowany jest potencjometrem RV1.



Rysunek 1. Schemat modułu

Dodatkowe materiały do pobrania ze strony www.media.avt.pl

W ofercie AVT* AVT-5721

Podstawowe parametry:

- wbudowany przetwornik DAC audio,
- końcówka mocy 2x8 W (8 Ω),
- wbudowany zasilacz 5 V dla Raspberry,
- zasilanie napięciem 9...15 V, min 3 A,
- przystosowany do zamontowania bezpośrednio na płytce Raspberry Pi.

***Uwaga! Elektroniczne zestawy do samodzielnego montażu.**

Wymagana umiejętność lutowania!

Podstawową wersją zestawu jest wersja [B] nazywana potocznie KIT-em (z ang. zestaw). Zestaw w wersji [B] zawiera elementy elektroniczne (w tym [UK] – jeśli występuje w projekcie), które należy samodzielnie wlutować w dołączoną płytkę drukowaną (PCB). Wykaz elementów znajduje się w dokumentacji, która jest podlinkowana w opisie kitu.

Mając na uwadze różne potrzeby naszych klientów, oferujemy dodatkowe wersje:

- wersja [C] – zmontowany, uruchomiony i przetestowany zestaw
- [B] (elementy wlutowane w płytkę PCB)
- wersja [A] – płytkę drukowaną bez elementów i dokumentacji

Kity w których występuje układ scalony wymagający zaprogramowania, mają następujące dodatkowe wersje:

- wersja [A*] – płytkę drukowaną [A] + zaprogramowany układ [UK] i dokumentacja
- wersja [UK] – zaprogramowany układ

Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik pdf! Podczas składania zamówienia upewnij się, którą wersję zamawiasz!

<http://sklep.avt.pl>. W przypadku braku dostępności

na <http://sklep.avt.pl>, osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: kity@avt.pl.

Za przetwarzanie sygnału I²S z interfejsu Raspberry Pi odpowiada układ U4 typu PCM5100A. Aplikacja U4 składa się w zasadzie z kondensatorów odsprężających i dławików FB1...3 filtrujących zasilanie. Wyjściowy sygnał analogowy filtrowany jest dolnoprzepustowo przez R4, R5, C24 i C25 i podawany na wyjście linowe o nie-regulowanym poziomie OUT oraz doprowadzony jest do potencjometru RV1.

Złącza wyjść OUTL, OUTR i zasilania PWR są powielone na opcjonalne listwy kołkowe OL, OR i PW1. Decyzja o montażu odpowiedniego kompletu zależy od aplikacji. Uzupełnieniem modułu jest pamięć EEPROM U6 typu 23LC32 wraz z elementami towarzyszącymi C26, R8...10 i zworą WP blokującą zapis zgodnie z wymogami dla modułów HAT Raspberry Pi. W modelu pamięć nie jest wykorzystana i nie jest wlutowana.

Montaż

Sposób montażu jest klasyczny i nie wymaga szczegółowego opisu. Rozmieszczenie elementów pokazuje **rysunek 2**. Po zmontowaniu i sprawdzeniu poprawności, ze szczególnym zwróceniem uwagi na poprawne przylutowanie padów termicznych układów U1, U2, należy moduł zasilic z napięcia 9...15 V. Najlepiej użyć zasilacza laboratoryjnego i zwiększając powoli napięcie należy sprawdzić obecność napięć 5 V i 3,3 V, dodatkowo dioda LD powinna świecić. Zmiana

położenia styku potencjometru RV1 powinna zmieniać stan sygnału SHDN. Układ końcówki mocy można sprawdzić z przy pomocy generatora sygnałowego podając na złącze OUT poprzez rezystor 10 kΩ, sygnał testowy 1 kHz/1 V_{pp}. Powinien być słyszalny w podłączonych głośnikach, a poziom powinien zmieniać się w zależności od położenia suwaka RV1. Jeżeli zasilanie i wzmacniacz działają, można podłączyć moduł do Raspberry Pi i do docelowego zasilacza w celu konfiguracji i ostatecznego sprawdzenia modułu.

Uruchomienie

Moduł jest zgodny z Hifiberry_DAC i tak należy skonfigurować system. Najlepszym wyborem jest Raspbian gdyż posiada wbudowaną obsługę programową (zgodną z Hifi-Berry). Na wszelki wypadek należy zaktualizować system wpisując polecenia:

```
sudo rpi-update
sync
sudo reboot
```

Następnie należy usunąć z pliku /boot/config.txt linię:

```
dtoverlay=hifiberry-dac
```

Teraz należy załączyć obsługę DAC, dodając wpisy w pliku /boot/config.txt oraz skonfigurować opcjonalnie ALSA tworząc plik /etc/asound.conf z zawartością:

```
pcm.!default {
    type hw card 0
}
ctl.!default {
    type hw card 0
}
```

Następnie należy zrestartować RPI. Po uruchomieniu warto sprawdzić poprawność konfiguracji poleceniem:

```
sudo aplay -l
```

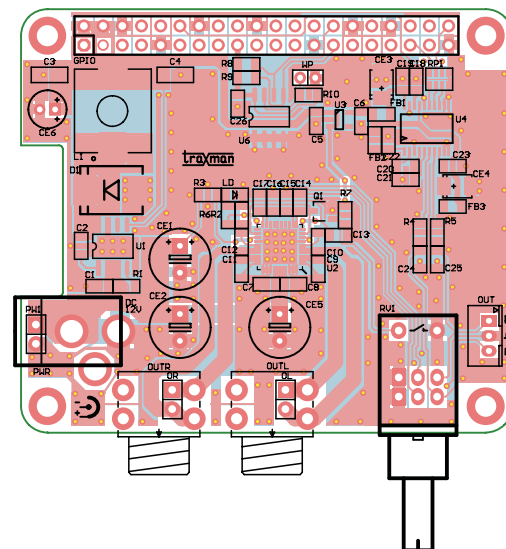
DAC powinien pojawić się na liście dostępnych urządzeń odtwarzających:

```
**** List of PLAYBACK Hardware Devices ****
```

```
card 0: sndrpihifiberry [snd_rpi_hifiberry_dac], device 0:
HifiBerry DAC HiFi pcm5102a-hifi-0 []
```

```
Subdevices: 1/1
```

```
Subdevice #0: subdevice #0
```



Rysunek 2. Schemat płytki PCB z rozmieszczeniem elementów

Wykaz elementów:

Rezystory:

R1: 100 kΩ SMD0805
R2, R6, R7: 47 kΩ SMD0805
R3, R8, R9, R10: 4,7 kΩ SMD0805
R4, R5: 470 Ω SMD0805
RP1: drabinka 4x22 Ω SMD0805
RV1, RV1: potencjometr stereo z wyłącznikiem 2x50 kΩ log.

Kondensatory:

C1, C2, C14...C19, C22, C23, C26: 100 nF SMD0805
C3, C4: 22 μF SMD1206
C5, C6: 4,7 μF SMD0805
C7, C8: 1 μF SMD0805
C9, C10, C11, C12: 220 nF SMD0805
C13, C20, C21: 2,2 μF SMD0805
C24, C25: 2,2 nF SMD0805
CE1, CE2, CE5: 220 μF / 25 V
CE3, CE4: 10 μF tantalowy SMA
CE6: 47 μF / 10 V

Półprzewodniki:

D1: SK33 SMC_D
L1: 6,8 μH HPI1040
LD: LED zielona SMD0805
Q1: BC847
U1: ADP2302 S08
U2: SSM3302 LFCSP40
U3: ADM7160 SOT-23-5
U4: PCM5100A SSOP20
U6: AT24LC32 S08

Pozostałe:

FB1, FB2, FB3: 600R SMD0805
GPIO: IDC40
OL, OR, PW1: SIP2
OUT: Złącze B3B 2 mm
OUTL, OUTR: Gniazdo jack 3,5 mono
PWR: Złącze zasilania
WP: SIP2 + zworka

Dla sprawdzenia działania można użyć pliku *flac/wav/mp3* i odtwarzacza VLC Player, Amarok lub wygenerować sygnał testowy przy pomocy np. Audacity. Miłego odsłuchu...

Adam Tatuś, EP

REKLAMA

Elektronice dyskutuj na <https://forum.ep.com.pl>

Przedwzmacniacz na lampie Nutube 6P1

W artykule przedstawiono aplikację lampy Korg Nutube 6P1 w przedwzmacniaczu liniowym. Dzięki Nutube możliwe jest zbudowanie przedwzmacniacza lampowego bez borykania się z problemami typowymi dla klasycznych lamp, jakimi są m. in. wysokie napięcia anodowe i temperatury układu. Układ może nadać „lampowego” charakteru dźwięku w torze audio, niezależnie od tego, czy będzie włączony w tor instrumentu, kształtując jego brzmienie, czy przy odtwarzaniu już nagranych materiałów.

Dodatkowe materiały do pobrania ze strony www.media.avt.pl

W ofercie AVT* AVT-5719

Podstawowe parametry:

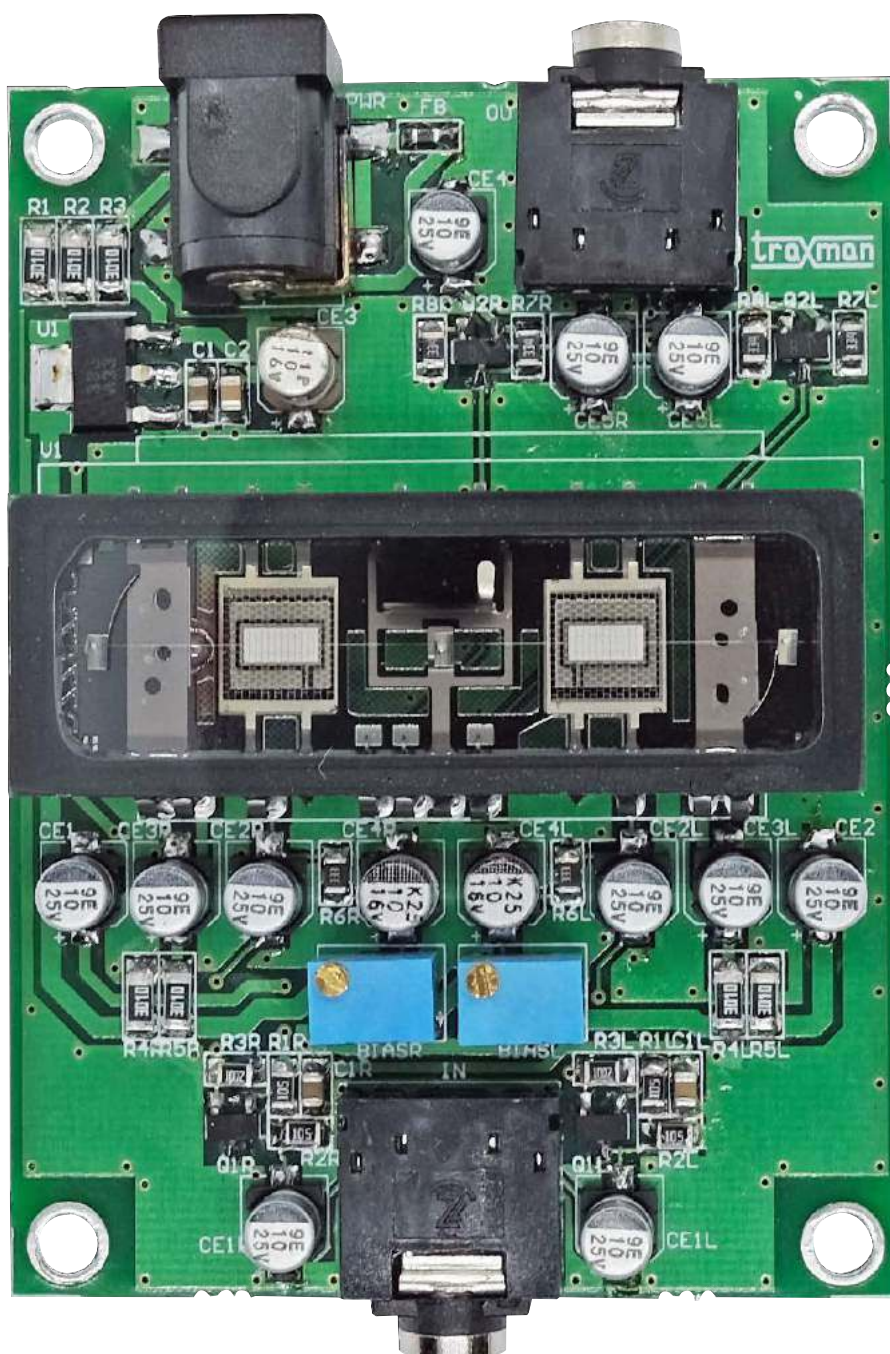
- stereofoniczny wzmacniacz liniowy,
- prosta i bezpieczna konstrukcja,
- zasilanie 12 V/100 mA.

Wykaz elementów:

R1L, R3, R4L, R4R, R5L, R5R: 300 Ω SMD1206
R1L, R1R, R2L, R2R: 1 M Ω SMD0805
R3L, R3R: 10 k Ω SMD0805
R6L, R6R: 33 k Ω SMD0805
R7L, R7R, R8L, R8R: 330 k Ω SMD0805
BIASL, BIASR: pot. wieloobrotowy pionowy 10 k Ω
C1, C2, C1L, C1R: 1 μ F SMD0805
CE1, CE2, CE3, CE4, CE1L, CE1R, CE2L, CE2R, CE3L, CE3R, CE4L, CE4R, CE5L, CE5R: 10 μ F/25 V
Q1L, Q1R, Q2L, Q2R: 2SK208
U1: NCP117ST33 SOT223
V1: NuTube6P1 Korg
FB: koralek ferrytowy 600 Ω /8,2 A
IN, OUT: gniazdo minijack 3,5 mm stereo
PWR: złącze zasilania DC 2,1 mm

Uwaga! Elektroniczne zestawy do samodzielnego montażu. Wymagana umiejętność lutowania!
Podstawową wersją zestawu jest wersja [B] nazywana potocznie KIT-em (z ang. zestaw). Zestaw w wersji [B] zawiera elementy elektroniczne (w tym UK) – jeśli występuje w projekcie, które należy samodzielnie wylutować w dołączoną płytkę drukowaną (PCB). Wykaz elementów znajduje się w dokumentacji, która jest podlinkowana w opisie kitu. Mając na uwadze różne potrzeby naszych klientów, oferujemy dodatkowe wersje:

- wersja [C] – zmontowany, uruchomiony i przetestowany zestaw [B] (elementy wylutowane w płytkę PCB)
 - wersja [A] – płytkę drukowaną bez elementów i dokumentacji. Kity w których występuje układ scalony wymagający zaprogramowania, mają następujące dodatkowe wersje:
 - wersja [A+] – płytkę drukowaną [A] + zaprogramowany układ [UK] i dokumentacja
 - wersja [UK] – zaprogramowany układ
- Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik pdf! Podczas składania zamówienia upewnij się, którą wersję zamawiasz!
<http://sklep.avt.pl>. W przypadku braku dostępności na <http://sklep.avt.pl>, osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: kity@avt.pl.



Budowa i działanie

Prezentowany wzmacniacz liniowy jest hybrydą elementów dyskretnych i lampy 6P1 (fotografia 1). Jest prosty w montażu i uruchomieniu, omija problemy spotykane w klasycznych układach lampowych. Schemat wzmacniacza pokazuje rysunek 2. Układ oparty jest na fabrycznej aplikacji proponowanej przez producenta lampy, firmę Korg. Elementy kanału lewego oznaczone są zakończeniem L, prawego R, elementy wspólne nie mają dodatkowych oznaczeń.

Sygnał wejściowy z gniazda IN typu Mini-Jack doprowadzony jest do stopnia na tranzystorze połowym Q1L. Jego zadaniem jest

zapewnienie niskiej rezystancji sterującej lampą niezależnie od parametrów podłączonego źródła sygnału. Następnie sygnał podawany jest na siatkę lampy V1. Lampa polaryzowana jest poprzez obwód BIASL, R6L ustalający punkt pracy. Żarzenie lampy stabilizowane jest przez U1 typu NCP117, a obniżenie i filtracja do 0,7 V zrealizowana jest na rezystorach R4L, R5L. Rezystory R1...3 zmniejszają napięcie wejściowe stabilizatora. Wzmocnione przez lampę napięcie doprowadzone jest bezpośrednio do stopnia na tranzystorze Q2L zapewniającego dopasowanie impedancji. Sygnał wyjściowy dostępny jest na gnieździe OUT.

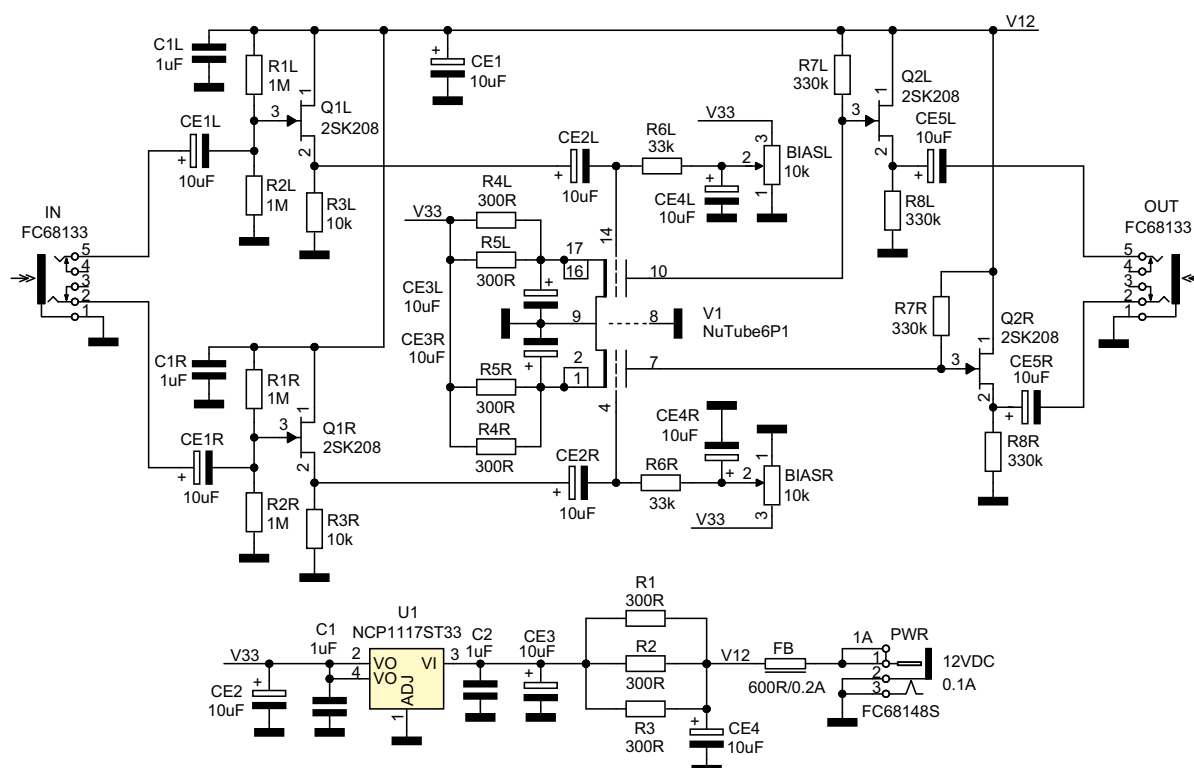
Układ zasilany jest poprzez gniazdo PWR poprzez typowy wtyk zasilający 2,1 mm. Napięcie zasilania wynosi 12 V, pobór prądu nie przekracza 100 mA. Układ musi być zasilany napięciem stabilizowanym, najlepiej stabilizatorem liniowym. Jeżeli planujemy zastosowania przenośne układ można zasiląć z baterii 9 V lub małego akumulatora żelowego 12 V.

Montaż i uruchomienie

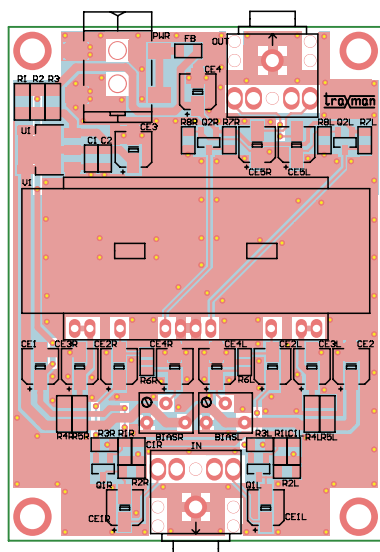
Wszystkie elementy przedwzmacniacza umieszczone są na dwustronnej płytce drukowanej. Schemat płytki, wraz z rozmieszczeniem elementów pokazuje rysunek 3.



Fotografia 1. Lampa Nutube 6P1



Rysunek 2. Schemat wzmacniacza liniowego



Rysunek 3. Schemat płytki wraz z rozmieszczeniem elementów

Montaż płytki jest typowy i nie wymaga opisu. Podczas lutowania lampy należy zachować ostrożność, ponieważ łatwo uszkodzić kruche wyprowadzenia tego nietypowego elementu. Przed montażem warto podkleić lampę taśmą dwustronną do powierzchni płytki. Zmontowany układ widoczny jest na fotografii tytułowej.

Wzmacniacz wymaga przeprowadzenia regulacji. Potrzebne będą generator i oscyloskop lub analizator widma. Po doprowadzeniu zasilania 12 V o wydajności przynajmniej 100 mA należy sprawdzić wartość napięcia żarzenia na kondensatorach CE3L, CE3R, która powinna zawierać się w przedziale 0,6...0,8 V. Następnie do wejścia należy doprowadzić sygnał 1 kHz/100 mV z generatora sygnałowego. Podłączając sondę oscyloskopu lub analizatora widma do wyjścia układu, potencjometrami BIASL i BIASR należy ustawić najniższe zniekształcenia sygnału wyjściowego. Teraz

pozostaje tylko podłączyć przedwzmacniacz słuchawkowy do docelowego zestawu audio i cieszyć się dźwiękiem...

Adam Tatuś, EP

Więcej informacji:

Wyłącznym dystrybutorem oryginalnych lamp Korg Nutube 6P1 jest

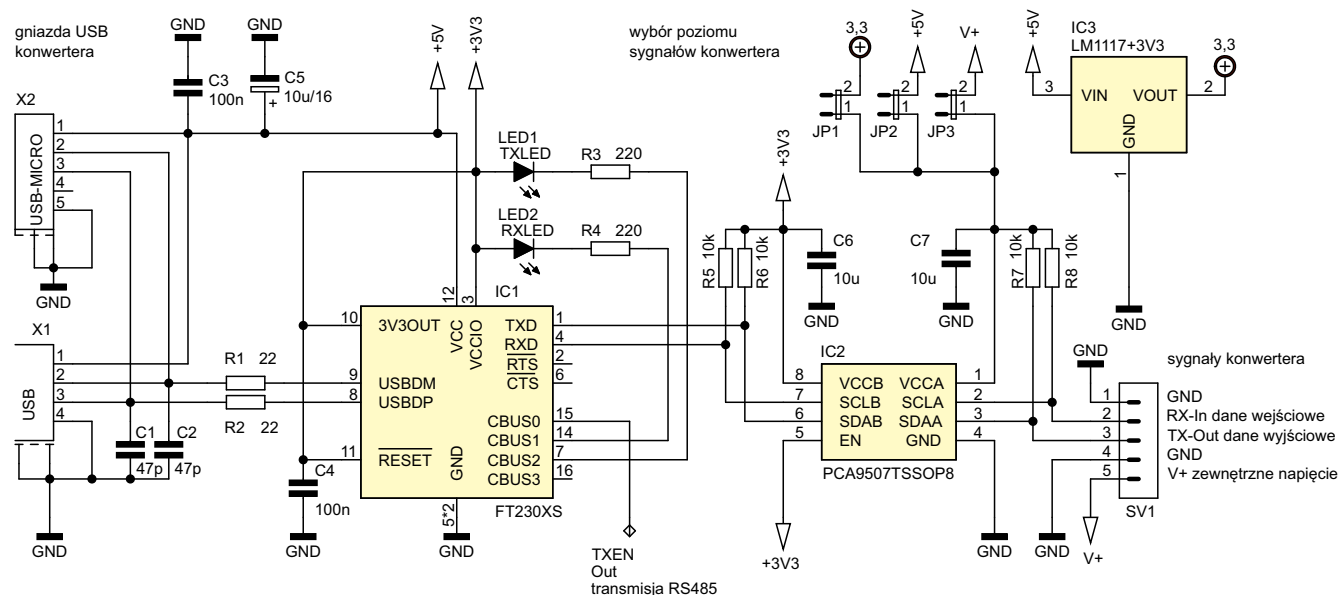
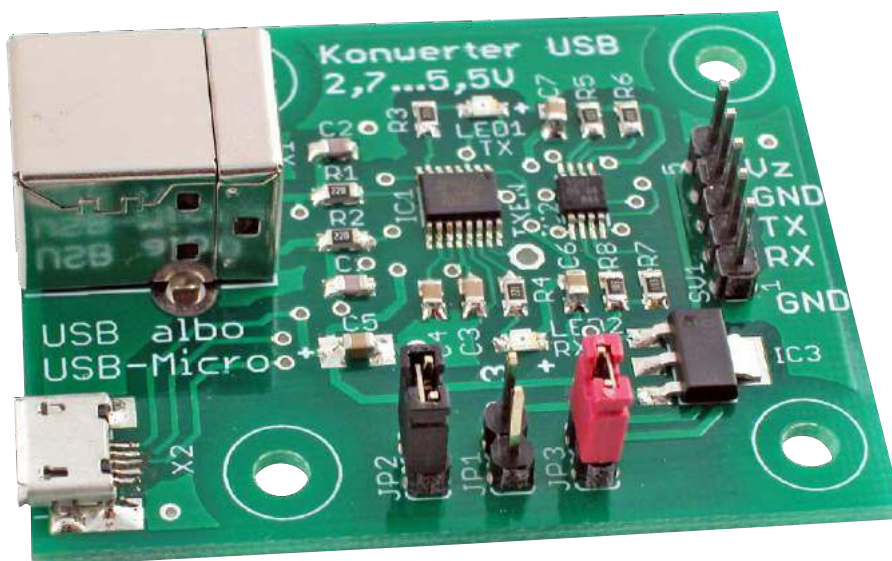
RS Components Sp. z o.o.

02-785 Warszawa, ul. Puławska 303
tel. 22 22 3 11 11, bok@rspoland.com
<https://pl.rs-online.com>



Konwerter USB-UART z ekstenderem

Konwerter pozwala podłączyć system mikroprocesorowy z interfejsem UART z komputerem wyposażonym w port USB. Umożliwia komunikację, przesył danych, programowanie procesora bezpośrednio w systemie itp. Konwerter może być stosowany z układami zasilanymi różnymi napięciami, czyli o różnych poziomach napięcia dla logicznej jedynki.



Rysunek 1. Schemat ideowy konwertera USB-UART

Budowa i działanie

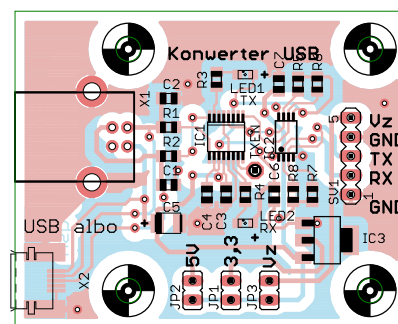
Konwerter podłączony do dowolnego portu USB komputera, zostanie rozpoznany jako dodatkowy (wirtualny) port szeregowy. Umożliwia to sterownik konwertera zainstalowany w systemie operacyjnym komputera. Część sprzętowa przekształca dane przesyłane liniami UART RxD i TxD na transmisję USB. Schemat ideowy konwertera pokazany został na **rysunku 1**.

Jako mostek USB wykorzystany został powszechnie stosowany układ FT230XS. Transmisja USB podawana jest na wejście układu przez jedno z dwu równolegle połączonych gniazd X1, X2. **Uwaga! W tym samym czasie do portu USB komputera może być podłączone tylko jedno z gniazd. Próba jednoczesnego użycia obydwu gniazd konwertera, doprowadzi do zakłócenia transmisji a nawet do elektrycznego uszkodzenia**

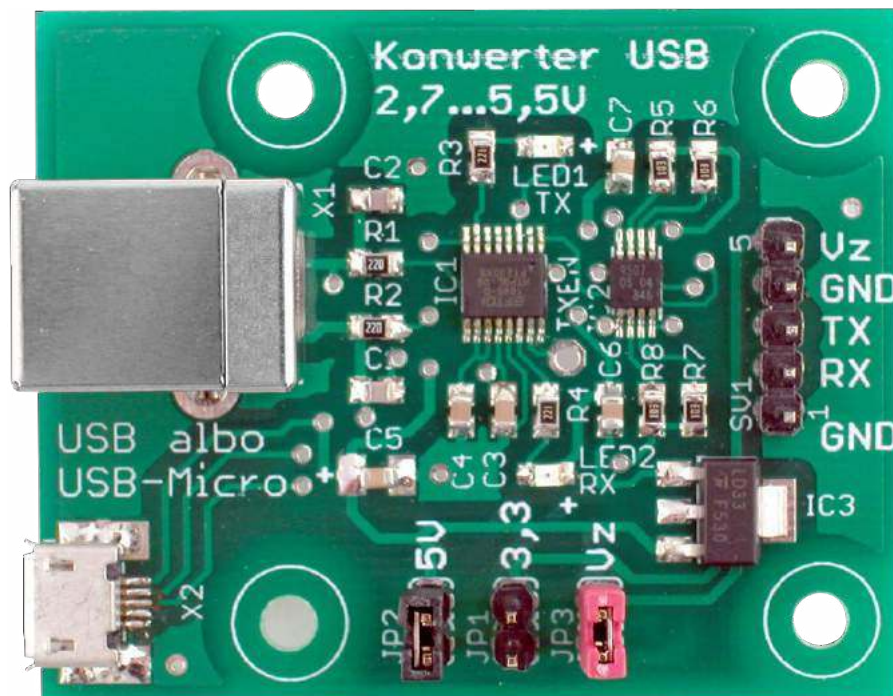
portu komputera. Układ IC1 zasilany jest napięciem 5 V pobieranym z portu USB komputera. Wewnętrzny stabilizator wytwarza napięcie 3,3 V służące do zasilania jego buforów, zarówno sygnałowych TxD, RxD jak i sygnalizacyjnych do których podłączone są diody LED. Świecenie diod informuje o przepływie danych odbieranych i wysyłanych poprzez port USB.

Bufory sygnałowe TxD i RxD podłączone są do wyprowadzeń konwertera poziomu IC2. Jego zadaniem jest elektryczne dopasowanie poziomu sygnałów UART doprowadzanych z dołączanego poprzez złącze SV1 systemu procesorowego. W roli konwertera „występuje” układ PCA9507. To trochę nietypowe rozwiązanie, ponieważ układ scalony został pierwotnie zaprojektowany do pracy z magistralami PC jako ich przedłużacz (ekstender). Posiada wewnętrzne układy

regeneracji zboczy sygnałów oraz możliwość konwersji poziomów pomiędzy wyprowadzeniami strony A i B. Strona B konwertera IC2 zasilana jest napięciem 3,3 V podawanym na wyprowadzenie VCCB. Jest to napięcie wykorzystywane do zasilania buforów



Rysunek 2. Schemat płytki PCB wraz z rozmieszczeniem elementów



Fotografia 3. Zmontowany układ

sygnałowych mostka IC1. Zasilanie strony A konwertera IC2 podawane jest na wyprowadzenie VCCA. Jego poziom powinien być dopasowany do poziomu sygnałów UART. Wyboru dokonuje się ręcznie ustawiając jedną ze zwór JP1, JP2, JP3. Ta ostatnia przyłącza jako napięcie odniesienia V+, napięcie pobierane z podłączonego układu procesorowego. Wartość napięcia V+ powinna mieścić się w przedziale 2,7...5,5 V.

Uruchomienie i eksploatacja

Schemat płytki PCB wraz z rozmieszczeniem elementów pokazuje rysunek 2. Montaż wykonujemy zgodnie z ogólnymi zasadami wzorując się na fotografii 3. Gdy konwerter będzie gotowy, podłączamy go do komputera kablem USB, tylko poprzez jedno z gniazd: USB-B lub mikro USB. System komputera powinien rozpoznać dodatkowy port szeregowy. Jeśli tak się nie stanie oznacza to, że nie jest zainstalowany sterownik obsługujący układ FT230XS. Zazwyczaj system Windows samodzielnie instaluje potrzebny sterownik korzystając z dostępu do sieci internet. W przypadku problemów trzeba ze strony firmy FTDI <http://bit.ly/2kqd3zb> pobrać odpowiedni plik i w standardowy sposób zainstalować sterownik w systemie.

Sygnały UART systemu z mikroprocesorem podłączamy poprzez złącze SV1. Linie

TxD UART dołączanego systemu należy podłączyć z wyprowadzeniem SV1-2 (opisanym na rysunku 1 jako RX-In, dane wejściowe). Linie RxD UART dołączanego systemu łączymy z SV1-3 (opisanym na rysunku 1 jako TX-Out dane wyjściowe). Masę dołączanego systemu łączymy z masą konwertera poprzez wyprowadzenie SV1-1 lub SV1-4.

W przypadku systemów zasilanych napięciem 5 V (większość popularnych mikrokontrolerów ATMEL) poziom sygnałów logicznych interfejsów UART będzie równy 0 V oraz 5 V. W tym przypadku trzeba na interfejsie zewrzeć zworę JP2. Gdy system zasilany jest napięciem 3,3 V (np. mikrokontroler STM32) poziom sygnałów logicznych interfejsów UART będzie wynosił 0 V oraz 3,3 V. W tym przypadku trzeba na interfejsie zewrzeć zworę JP1. W przypadku systemów zasilanych innym napięciem np. 2,8 V należy zewrzeć zworę JP3 a do wyprowadzenia SV1-5 (V+) podłączyć napięcie zasilania systemu.

Testy

Po zmontowaniu prototypu przeprowadziłem testy możliwości konwertera USB-UART. Do testów użyłem płytkę Nucleo-F446 z oprogramowaniem pracującym jak echo, odsyłającym UART-em odebraną transmisję. Wyprowadzenie konwertera SV1-2 (RX-In) połączyłem z portem płytki

Dodatkowe materiały do pobrania ze strony www.media.avt.pl

W ofercie AVT* AVT-5717

Podstawowe parametry:

- zamontowane dwa typy gniazd USB-B i mikro USB,
- wybieranie zworkami poziomu dopasowania sygnałów: 5 V, 3,3 V lub dowolnego w zakresie 2,7...5,5 V,
- możliwość zasilania systemu procesorowego z konwertera, napięciem 5 V lub 3,3 V,
- możliwość podłączania przewodów sygnałowych o długości nawet kilku metrów,
- maksymalna szybkość transmisji wynosi 512000 bódów,
- zasilanie konwertera z portu USB.

Wykaz elementów:

R1, R2: 22 Ω SMD0805
R3, R4: 220 Ω SMD0805
R5, R6, R7, R8: 10 kΩ SMD0805
C1, C2: 47 pF SMD0805
C3, C4: 100 nF SMD0805
C5, C6, C7: 10 μF SMD0805
IC1: FT230XS SSOP-16
IC2: PCA9507 TSSOP-8
IC3: LM1117 3V3 SOT-223-4
LED1, LED2: dioda led SMD0805
X1: gniazdo mikro USB
X2: gniazdo USB-B
SV1: 5x1 goldpiny proste
JP1, JP2, JP3: 2x1 goldpiny proste

* Uwaga! Elektroniczne zestawy do samodzielnego montażu.

Wymagana umiejętność lutowania!
Podstawową wersją zestawu jest wersja [B] nazywana potocznie KIT-em (z ang. zestaw). Zestaw w wersji [B] zawiera elementy elektroniczne (w tym [UK] – jeśli występuje w projekcie), które należy samodzielnie wstawić w dołączoną płytkę drukowaną (PCB). Wykaz elementów znajduje się w dokumentacji, która jest podlinkowana w opisie kitu. Mając na uwadze różne potrzeby naszych klientów, oferujemy dodatkowe wersje:

- wersja [C] – zmontowany, uruchomiony i przetestowany zestaw
 - wersja [B] (elementy wstawiane w płytkę PCB)
 - wersja [A] – płytka drukowana bez elementów i dokumentacji
 - wersja [A+] – płytka drukowana z dodatkowymi elementami
 - wersja [A+] – płytka drukowana [A] + zaprogramowany układ [UK] i dokumentacja
 - wersja [UK] – zaprogramowany układ
- Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik pdf! Podczas składania zamówienia upewnij się, którą wersję zamawiasz! <http://sklep.avt.pl>. W przypadku braku dostępności na <http://sklep.avt.pl>, osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: kity@avt.pl.

Nucleo PA9 (UART1 TX), wyprowadzenie SV1-3 (TX-Out) z PA10 (UART1 RX), SV1-1 (GND) z jednym z wyprowadzeń masy na płytce Nucleo.

Zależnie od testu zmieniałem:

- szybkość transmisji 256000 lub 512000 bódów,
- poziom sygnałów 5 V lub 3,3 V,
- długość przewodów pomiędzy płytką konwertera a płytką Nucleo: 30 cm lub 5 m.

Rezultaty testów pozwoliły stwierdzić, że przy szybkościach transmisji równych lub mniejszych niż 256000 bódów, konwerter pracuje bez problemu. Dla szybkości 512000 bódów i podłączeniach pomiędzy konwerterem a płytką Nucleo o długości 30 cm także wszystko działa prawidłowo. Konwerter zawodzi przy kombinacji dużej szybkości transmisji, długich przewodów i przy poziomach logicznych 0...5 V.

Ryszard Szymaniak

REKLAMA

<https://m.ep.com.pl>

Dział „Projekty Czytelników” zawiera opisy projektów nadesłanych do redakcji EP przez Czytelników. Redakcja nie bierze odpowiedzialności za prawidłowe działanie opisywanych układów, gdyż nie testujemy ich laboratoryjnie, chociaż sprawdzamy poprawność konstrukcji. Prosimy o nadsyłanie własnych projektów z modelami (do zwrotu). Do artykułu należy dołączyć podpisane oświadczenie, że artykuł jest własnym opracowaniem autora i nie był dotychczas nigdzie publikowany. Honorarium za publikację w tym dziale wynosi 250,- zł (brutto) za 1 stronę w EP. Przesyłanych tekstów nie zwracamy. Redakcja zastrzega sobie prawo do dokonywania skrótów.

**PROJEKT
NR 246**



Uwaga! W urządzeniu występuje napięcie niebezpieczne dla życia! Należy zachować ostrożność podczas uruchamiania układu. Gałka enkodera powinna być wykonana z materiału, który nie przewodzi prądu.

Sterownik wentylatora wyciągu łazienkowego (kuchennego)

Ręczne sterowanie wentylatorem wyciągu w łazience czy kuchni nie jest zbyt wygodne. W łazience czy toalecie można podłączyć wentylator pod oświetlenie, jednak to rozwiązanie ma kilka wad. Wentylator wyłączy się natychmiast gdy wyjdziemy z łazienki po kąpeli, gdy w łazience jest jeszcze duża wilgotność. W kuchni nie ma innego rozsądnego rozwiązania niż sterowanie ręczne. Zaprezentowany sterownik automatycznie steruje wentylatorem na podstawie zmian wilgotności, czasowo po włączeniu/wyłączeniu oświetlenia oraz umożliwia sterowanie ręczne. Złącze karty sieciowej Wi-Fi umożliwia podłączenie sterownika do systemu inteligentnego domu.

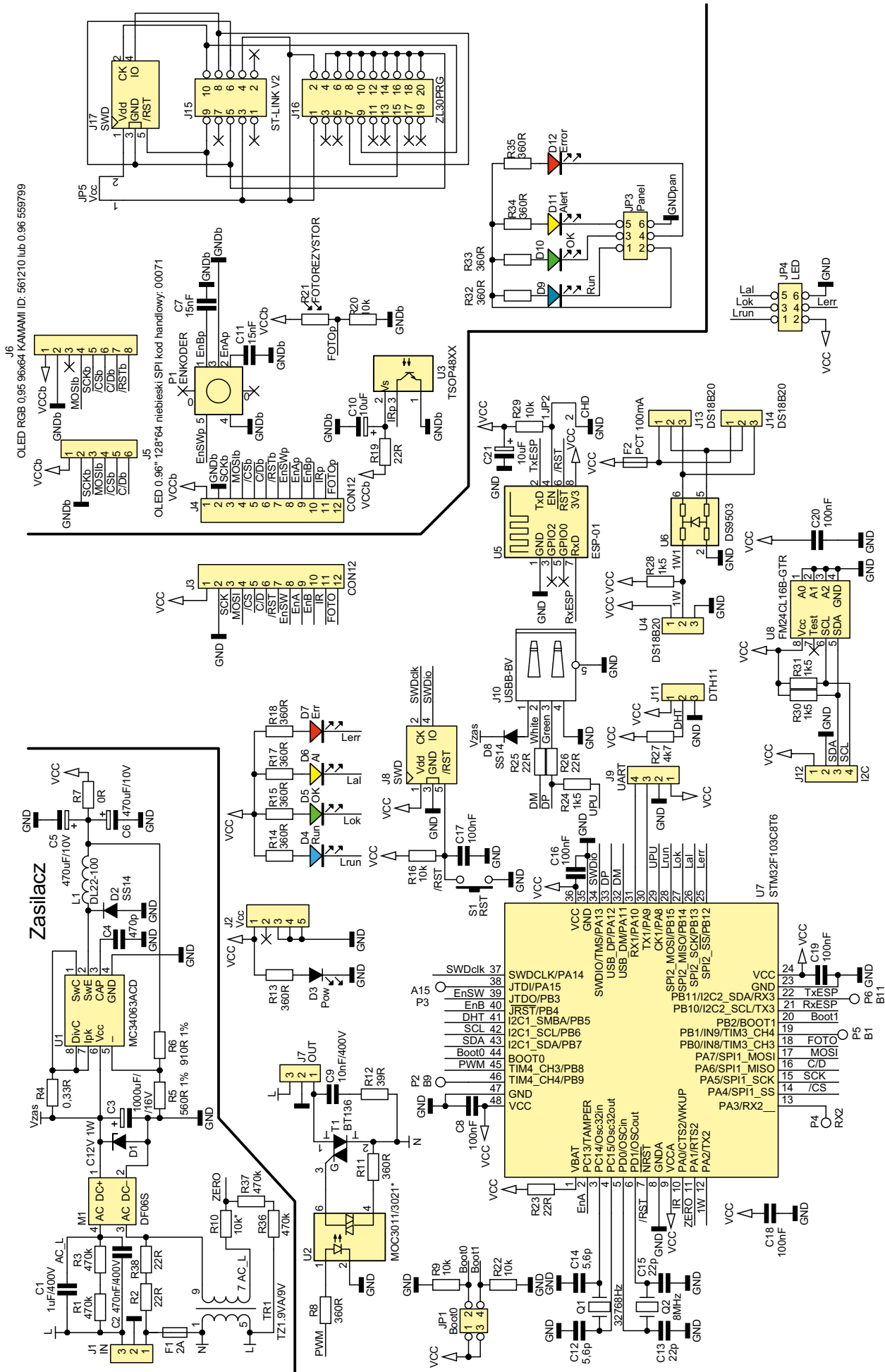
Budowa i działanie

Schemat układu prezentuje **rysunek 1**. Urządzenie zasilane jest stabilizatorem impulsowym zbudowanym na układzie

U1 (MC34063ACD) pracującym w typowej aplikacji. Stabilizator może być zasilany z transformatora lub z zasilacza beztransformatorowego. Zasilacz beztransformatorowy

spawdzi się w sytuacji, gdy nie korzystamy z łącza USB i pobór prądu jest niewielki (brak modułu Wi-Fi). Stabilizowane napięcie 3,3 V zasilą wszystkie obwody sterownika.

Mikrokontrolerem sterującym jest ARM STM32F103. Na płycie przewidziano kwarce do taktowania μC oraz zegara RTC. W aktualnej wersji oprogramowania μC jest taktowany wewnętrznym generatorem RC i kwarce nie są używane ale wyklucza to możliwość uruchomienia USB (wymagany kwarc). Elementem sterującym fazowo silnikiem wentylatora jest triak sterowany optotriakiem. Triak jest zabezpieczony szybkim bezpiecznikiem F1. Przebiegi napięcia sieciowego



Rysunek 1. Schemat układu

przez zero, wymagane przy sterowaniu fazowym, jest wykrywane przez GPIO PA2 (przerwanie EXTI1). W przypadku zasilacza beztransformatorem montowane są rezystory R36, R37, w przypadku zasilacza z transformatorem – montowany jest R10. Sterownik wykrywa zarówno narastanie jak i opadanie napięcia sieciowego (reakcja na oba zbocza sygnału). Diody D4... D7 sygnalizują stan pracy sterownika).

Do komunikacji z użytkownikiem zastosowano enkoder i monochromatyczny, dwukolorowy, wyświetlacz OLED o rozdzielczości 128×64. Odbiornik podczerwieni (U3) umożliwia zdalne sterowanie urządzeniem ale ta opcja jest przewidziana do przyszłych zastosowań i nie została jeszcze oprogramowana. Wilgotność powietrza mierzy czujnik DTH11 podłączony do złącza J11. Istnieje możliwość pomiaru temperatury termometrami DS18B20 podłączonymi do J13, J14 i/ lub U4 wlotowego w płytkę sterownika. U6 (DS9503)

zabezpiecza wejście µC przed uszkodzeniem przepięciami, które mogą pojawić się na przewodach łączących termometry ze sterownikiem. Zasilanie czujników zewnętrznych jest zabezpieczone bezpiecznikiem polimerowym F2. Włączenie oświetlenia wykrywa fotorezystor R21 tworzący wraz z R20 dzielnik napięcia. Umożliwia to pomiar intensywności oświetlenia bo wyłączenie oświetlenia np. w łazience nie musi oznaczać, że jest całkowicie ciemno, przykładowo światło może wpadać przez szybę w drzwiach. Sterownik wyposażono w pamięć FRAM (U8) ale nie

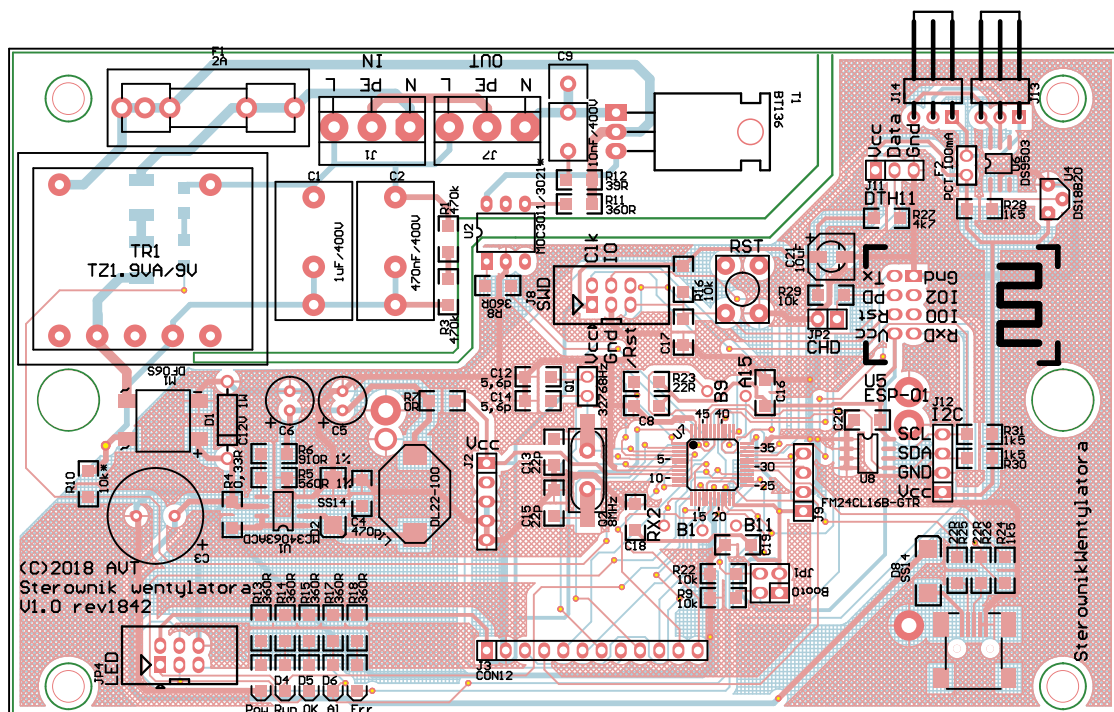
jest ona używana, ustawienia zapisywane są w EEPROM emulowanym w pamięci FLASH µC.

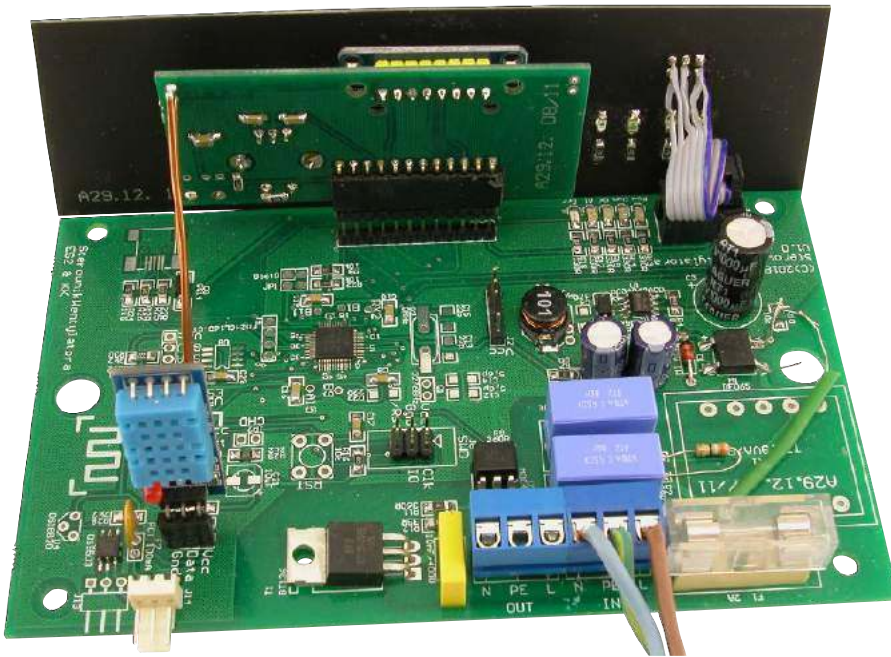
Montaż i uruchomienie

Schemat płytki wraz z rozmieszczeniem elementów pokazano na **rysunku 2**. Na początku montażu należy wybrać sposób zasilania: transformatorem czy beztransformatorem. Zależnie od opcji nie montuje się transformatora lub elementów C1, C2, R1... R3. Następnie należy zamontować elementy stabilizatora impulsowego i uruchomić go.

Gdy pracuje poprawnie można zamontować pozostałe elementy. Nie ma potrzeby montowania kwarców i kondensatorów do nich, odbiornika podczerwieni i pamięci FRAM.

Złącze J8 służy do zaprogramowania µC, do tego celu wystarczy ST-Link V2. Jeżeli w sterowniku zastosowano zasilacz beztransformatorem, podczas programowania, należy zastosować izolowany galwanicznie interfejs USB, transformator separacyjny 230 V/230 V lub zasilic sterownik z izolowanego galwanicznie zasilacza 7...12 V. Jeżeli czujnik wilgotności będzie zamontowany





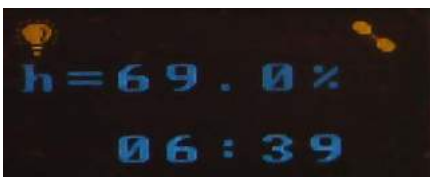
Fotografia 3. Widok zmontowanego układu

wewnątrz sterownika należy zadbać o odpowiednie otwory w obudowie. Ze względu na to, że elementy sterownika mogą się nagrzewać (transformator, triak) zaleca się zamontowanie czujnika poza sterownikiem.

Płyta czołowa sterownika jest wlutowana w płytę bazową z wykorzystaniem kątownego gniazda GOLDPIN oraz listwy o wysokości 20 mm. Aby ustabilizować konstrukcję, płyta czołowa jest wzmocniona dwoma drutami wlutowanymi w przygotowane do tego celu pola na płytkach (fotografia 3). Dzięki temu płyta czołowa nie ugina się podczas naciskania przycisku enkodera. Diody LED na płycie czołowej połączone są z płytą bazową z użyciem gniazda i wtyku IDC6FT. Jeżeli



Fotografia 4. Ekran powitalny



Fotografia 5. Wygląd ekranu głównego



Fotografia 6. Ekran konfiguracji poziomów wilgotności

zamontowane zostaną LED zarówno na płycie czołowej jak i na płycie głównej to należy zwiększyć wartość rezystorów ograniczających prąd do 750 Ω , zarówno na płycie czołowej jak i na głównej. Wymiary płytek i rozmieszczenie otworów są przystosowane do obudowy KM-50.

Obsługa urządzenia

Jeśli montaż przeprowadzono prawidłowo, po włączeniu sterownika pokaże się ekran powitalny (fotografia 4), a po chwili ekran główny (fotografia 5). Przy pierwszym uruchomieniu sterownik należy skonfigurować. W tym celu kręcąc enkoderem wybieramy ekran konfiguracji poziomów wilgotności, przy których nastąpi załączenie i wyłączenie wentylatora (fotografia 6). W drugim wierszu wyświetlana jest aktualna wilgotność. W trzecim i czwartym, poziomy, przy których nastąpi włączenie (ON) i wyłączenie (Off) wentylatora. Wciśnięcie enkodera wywoła miganie trzeciego wiersza wyświetlacza umożliwiając ustawienie wartości wilgotności, przy której wentylator zostanie włączony. Kręcąc enkoderem można zwiększać i zmniejszać wartość parametru. Kolejne naciśnięcie enkodera pozwoli zmienić parametr określający poziom wilgotności, przy której nastąpi wyłączenie wentylatora. Program dba o to, aby pomiędzy poziomami włączenia i wyłączenia była różnica (histereza) co najmniej 10%. Trzecie naciśnięcie enkodera wyłącza ustawianie parametrów. Kręcąc enkoderem można wybrać inny ekran konfiguracyjny. Warto zaznaczyć, że pozostawienie aktywnego ekranu konfiguracyjnego lub zmiany parametru na dłużej niż minutę, spowoduje powrót do ekranu głównego.

Kolejnym ekranem jest ustawienie poziomu oświetlenia, który jest interpretowany

Podstawowe parametry:

- pobór prądu ok 0,24 W (zasilanie 12 V 20 mA),
- max. obciążenie: 230 W (2 kW z radiatorem),
- parametry wyświetlane na wyświetlaczu oled,
- sterowanie przy pomocy enkodera przyciskiem,
- kryteria zał/wył: wilgotność, oświetlenie, manualnie,
- pasująca obudowa KM-50

Projekty pokrewne na www.media.avt.pl:

- AVT-5698 Sterownik wentylatorów 12 V dużej mocy (EP 8/2019)
- AVT-5653 Sterownik mikro wentylatora (EP 11/2018)
- Uniwersalny driver silnika małej mocy (EP 3/2018)
- AVT-1981 Sterownik wentylatora z płynną zmianą obrotów (EP 1/2018)
- AVT-5612 Dwukierunkowy regulator obrotów silnika prądu stałego (EP 12/2017)
- AVT-5565 Sterownik silnika do napędu (EP 10/2016)
- AVT-478 Regulator obrotów wentylatorów 12 V (EP 8/2016)
- AVT-1855 Sterownik wentylatora z czujnikiem wilgotności powietrza (EP 5/2015)
- AVT-3082 Zasilacz - sterownik miniwiertarki DC (Edw 1/2014)
- AVT-1724 Uniwersalny sterownik silników DC (EP 2/2013)
- AVT-1596 Regulator obrotów wentylatora (EP 10/2010)
- AVT-1519 Sterownik silnika do modeli RC (EP 4/2009)
- AVT-1469 Generator PWM - regulator mocy silnika DC (EP 8/2008)
- AVT-2871 Bi-motor driver (Edw 7/2008)
- AVT-1444 Dwukierunkowy regulator obrotów silników prądu stałego (EP 12/2006)
- AVT-1564 Sterownik wentylatora 12 V (EP 8/2001)

Wykaz elementów:

Rezystory SMD 1206:

- R4: 0,33 Ω
- R7: 0 Ω
- R24, R28, R30, R31: 1,5 Ω
- R27: 4,7 k Ω
- R10*: 10 k Ω
- R9, R16, R20, R22, R29: 10 k Ω
- R19, R23, R25, R26: 22 Ω
- R12: 39 Ω
- R8, R11, R13...R15, R17, R18, R32...R35: 360 Ω
- R1, R3: 70 k Ω
- R36*, R37*: 470 k Ω

Rezystory SMD 1812:

- R2, R38: 22 Ω

Kondensatory SMD1206:

- C12, C14: 5,6 pF
- C7, C11: 15 nF
- C13*, C15*: 22 pF
- C8, C16, C17, C18, C19, C20: 100 nF
- C4: 470 pF
- C10, C21: 10 μ F

Kondensatory THT:

- C1: 1 μ F/400 V
- C9: 10 nF/400 V
- C2: 470 nF/400 V
- C5, C6: 470 μ F/10 V elektrolityczny
- C3: 1000 μ F/16 V elektrolityczny
- C2: 470 nF/400 V elektrolityczny
- C5, C6: 470 μ F/10 V elektrolityczny

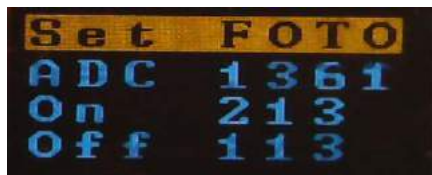
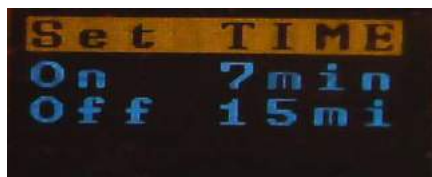
Półprzewodniki:

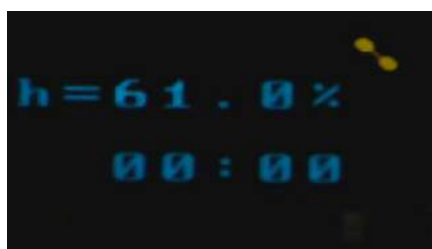
- U7: STM32F103C8T6
- U3: TSOP48XX
- U1: MC34063ACD SO-8
- U2: MOC3020/3021 DIP6
- T1: BT136 TO-220
- U6*: DS9503 SO-6
- U8*: FM24CL16B-GTR SO-8
- D2, D8: SS14 D021
- M1: DF06S
- R21: FOTOREZYSTOR 10 k Ω
- D6, D11: Dioda LED żółta SMD1206
- D7, D12: Dioda LED czerwona SMD1206
- D3, D4, D9: Dioda LED niebieska SMD1206
- D5, D10: Dioda LED Zielona SMD1206
- D1: C12V 1W THT
- J5: OLED 0,96" 128x64 niebieski SPI
- J6*: OLED RGB 0,95 96x64 KAMAMI
- U4*: DS18B20

Pozostałe:

- F1: Oprawa ZH32 wraz z szybkim bezp. 2 A
- F2: PCT 100 mA
- Q2*: Kwar 8 MHz
- Q1*: Kwar 32768 Hz
- L1: Dławik DL22-100
- J11: Czujnik DTH11
- U5*: ESP-01
- IMPULSATOR: Enkoder impulsowy z przyciskiem
- JP1: Goldpin 2x2
- JP2: Goldpin 1x2
- JP3, JP4: 821-1-06-S1 + T812-1-06 + taśma FLAT
- J13, J14: SN25-W3K
- J1, J7: ARK3
- S1: Micoswitch 5x7
- TR1*: TZ1.9VA/9 V
- J10*: USB-B-BV

* - nie montować (patrz tekst)


Fotografia 7. Ekran ustawiania poziomu oświetlenia

Fotografia 8. Ekran ustawień czasów

Fotografia 9. Ekran wyłączenia automatycznej akcji

Fotografia 10. Wygląd ekranu w stanie spoczynku

jako włączenie światła oraz poziomu traktowanego jako wyłączenie oświetlenia (**fotografia 7**). Drugi wiersz pokazuje wartość odczytaną (po uśrednieniu) z przetwornika ADC. Ustawianie poziomu załączenia i wyłączenia przebiega tak samo jak w przypadku wilgotności. W kolejnym ekranie (**fotografia 8**) można ustawić czas (On) po którym, od wykrycia włączenia oświetlenia, nastąpi załączenie wentylatora oraz jak długo po zgaśnięciu światła (Off) wentylator ma jeszcze pracować. Ustawienia te zapobiegają niepotrzebnemu włączaniu się wentylatora gdy wchodzimy na chwilę do łazienki oraz odprowadzić parę czy wywietrzyć pomieszczenie po jego dłuższym użytkowaniu. Kolejny ekran (**fotografia 9**) pozwala wyłączyć automatyczną reakcję na oświetlenie i/lub wilgotność. Wyłączenie automatyki spowoduje, że ekran przypisany do wybranej opcji będzie przyciemniony. Ostatni ekran wyświetla informacje diagnostyczne. Ich znaczenie można poznać analizując kod źródłowy programu.

Po skonfigurowaniu można przejść do użytkowania sterownika. Wygląd ekranu w stanie spoczynku przedstawia **fotografia 10**. Włączenie oświetlenia sygnalizuje

Listing 1. Podstawowe definicje

```
#define DEF_TIM_FAN_MAN_MAX (10*60*60*1000) // 10 godzin
#define DEF_TIM_FAN_ON_MAN (30 * 60 * 1000) // krok 30 minut
#define DEF_TIMEOUT_MENU (60*1000)
#define DEF_TIM_DEMO (5 * 1000)
#define DEF_EEPROM_TIMEOUT (10 * 1000) // 10 sekund
```

Listing 2. Fragment kodu pokazujący sposób użycia DMA do komunikacji z wyświetlaczem

```
HAL_GPIO_WritePin( LCD_C_D_GPIO_Port, LCD_C_D_Pin, GPIO_PIN_SET );
HAL_GPIO_WritePin( LCD_CS_GPIO_Port, LCD_CS_Pin, GPIO_PIN_RESET );
ssd1306_busy = 1;
HAL_SPI_Transmit_DMA( &hspi1, ssd1306buffer, SSD1306_LCDHEIGHT * SSD1306_LCDWIDTH / 8 );
```

Listing 3. Kod procedury uruchamianej przerwaniem z DMA

```
//Handler przerwania DMA:
void DMA1_Channel13_IRQHandler(void)
{
    /* USER CODE BEGIN DMA1_Channel13_IRQn 0 */
    DMA_HandleTypeDef *hdma = &hdma_spi1_tx;
    uint32_t flag_it = hdma->DmaBaseAddress->ISR;
    uint32_t source_it = hdma->Instance->CCR;

    /* Transfer Complete Interrupt management */
    if (((flag_it & (DMA_FLAG_TC1 << hdma->ChannelIndex)) != RESET)
        && ((source_it & DMA_IT_TC) != RESET)){
        extern void ssd1306_DMA_End();
        ssd1306_DMA_End();
    }

    /* USER CODE END DMA1_Channel13_IRQn 0 */
    HAL_DMA_IRQHandler(&hdma_spi1_tx);
    /* USER CODE BEGIN DMA1_Channel13_IRQn 1 */
    /* USER CODE END DMA1_Channel13_IRQn 1 */
}
```

Listing 4. Dodatkowa funkcja wywoływana przerwaniem z DMA

```
void ssd1306_DMA_End(){
    // Timer przy wartosci 1 wywola „ssd1306_CS_End()”
    extern uint8_t TimLcdCs;
    TimLcdCs = 1;
}
```

Listing 5. Dekrementowanie wirtualnego timera w przerwaniu systemowym

```
void Irq1ms() {
    //----- dezaktywacja, przez DMA, strobu LCD -----//
    extern void ssd1306_CS_End();
    if ( TimLcdCs == 1 ) ssd1306_CS_End();
    if ( TimLcdCs ) TimLcdCs--;
}
```

symbol żarówki w lewym górnym rogu wyświetlacza, ponadto rozpocznie się odliczanie czasu do włączenia wentylatora. Gdy wentylator zostanie załączony odliczany będzie maksymalny dopuszczalny czas pracy (10 godzin). Po zgaśnięciu światła wentylator będzie nadal pracował przez czas ustawiony w menu „Set Time”. Na wyświetlaczu będzie widać upływ czasu do wyłączenia wentylatora. Jeśli przyczyną załączenia wentylatora będzie wzrost wilgotności na ekranie zobaczymy symbol „Rh”. W każdej chwili można wyłączyć wentylator naciskając przycisk enkodera. Możliwe jest też ręczne jego włączenie. Wentylator jest załączany na określony czas, który można regulować enkoderem w zakresie od 30 minut do 10 godzin, z krokiem 30 minut. Regulacja możliwa jest przez 60 sekund od włączenia wentylatora.

Program sterujący

W artykule zostaną opisane tylko wybrane fragmenty programu, pełne kody źródłowe dostępne są w materiałach dodatkowych. Definicjami z **listingu 1** można regulować maksymalny czas pracy wentylatora, krok czasu w trybie ręcznym, czas nieaktywności użytkownika, po którym program wraca z ekranu

ustawień do ekranu głównego, czas wyświetlania napisu powitalnego oraz czas po, którym, od zakończenia regulacji, nastawy zostaną zapamiętane w pamięci nieulotnej.

Funkcje obsługi czujnika DHT11 zostały przeniesione z Arduino. W przeciwieństwie do AVR w ARM można wykonać obsługę czujnika bez potrzeby zawieszania przerwań. Oczywiście lepiej byłoby do tego celu użyć UART (protokół DHT11 jest oparty o 1-Wire) ale w tym zastosowaniu nie wydawało się to celowe. Nie modyfikowałem też nadużywanych w Arduino typów FLOAT na INT ponieważ ARM jest na tyle szybki, że czasy wykonywania operacji na FLOAT nie mają większego znaczenia, tym bardziej, że wykonywane są stosunkowo rzadko.

Interesujące może być emulowanie pamięci EEPROM w FLASH. Zastosowano bibliotekę dostarczoną przez STMicroelectronics. Używając biblioteki trzeba pamiętać, aby w IDE zmniejszyć ilość dostępnej pamięci FLASH. Aby biblioteka działała, przed jej użyciem należy odblokować dostęp do zapisu pamięci FLASH, umieszczając poniższy kod:

```
HAL_FLASH_Unlock();
EE_Init();
```

Bardziej eleganckim rozwiązaniem byłoby odblokowywanie pamięci przed procedurą zapisu i blokowanie po jej zapisaniu.

Procedury obsługi wyświetlacza także zostały przeniesione z Arduino, z tym, że transmisja używa mechanizmu DMA (**listing 2**). Pojawia się problem zdjęcia stanu linii CS. Co prawda DMA może generować przerwanie, gdy wyśle połowę i wszystkie dane (**listing 3**). Niestety nie można po zakończeniu transferu od razu zdjąć stanu CS, ponieważ SPI jeszcze transmituje dane. Takie natychmiastowe zdjęcie stanu powodowało obciążenie 4...6 bitów danych. Aby tego uniknąć, przerwanie DMA wywołuje funkcję z **listingu 4**. Funkcja ta ustawia wirtualny timer „TimLcdCs”, który jest dekrementowany w przerwaniu systemowym co 1 ms (**listing 5**). W przerwaniu, gdy TimLcdCs osiągnie wartość 1, wywołana zostanie funkcja dezaktywująca stan linii CS. W `ssd1306_DMA_End()` przypisanie

zmiennej „TimLcdCs” wartości 1 jest trochę niebezpieczne. Może się zdarzyć, że zaraz po przerwaniu DMA wykona się przerwanie 1ms. W konsekwencji stan linii CS zostanie zdjęty zbyt szybko. Może to powodować niekontrolowane zachowanie się ostatnich ośmiu pikseli na wyświetlaczu. Ze względu na to, że szansa na wystąpienie tego zjawiska jest bardzo mała oraz to, że wyświetlacz odświeżany jest 10 razy na sekundę nie modyfikowałem kodu. Rozwiązania są dwa, albo w przerwaniu `ssd1306_DMA_End()` do zmiennej „TimLcdCs” wpisywać liczbę 2 albo lepsze – uruchomić dodatkowy timer odliczający μ s, który odczeka wymagany czas i zdejmie stan linii CS wyświetlacza. Należy przyznać, że obsługa wyświetlacza z interfejsem I²C jest dużo prostsza, uruchamia się DMA i to wszystko.

Program obsługuje „UsageFault_Handler”, „BusFault_Handler”,

„MemManage_Handler”, „HardFault_Handler” i inne błędy sygnalizując je diodą ERROR. W programie można wprowadzić szereg zmian. Można oprogramować moduł karty Wi-Fi umożliwiając współpracę z systemem inteligentnego budynku. Pisanie obsługi Wi-Fi ułatwić może uruchomienie interfejsu USB lub UART1 dostępnego na J9. Można też dodać obsługę 1-Wire a co za tym idzie, pomiar temperatury (termometr w DHT11 nie jest zbyt dokładny). W programie, od reakcji na bezwzględne wartości wilgotności, lepiej sprawdziłby się algorytm reagujący na wzrost/spadek wilgotności w zadanym czasie. Program steruje tylko diodami RUN i ERR (Error). Można dodać obsługę diody AL (Alert) oraz OK. Alert mógłby informować o awarii czujnika DHT11, OK o poprawnej pracy wszystkich czujników.

SaS

sas@elportal.pl

REKLAMA

E-prenumerata to:

- najszybszy dostęp do nowego wydania magazynu
- wygodne archiwum na www.avt.pl
- hipertekstowy spis treści i wyszukiwarka
- wbudowane linki – klikasz i jesteś na odpowiedniej stronie WWW

Chcesz otrzymywać dodatek
Niezbędnik Elektronika?
Zamów prenumeratę
drukowaną na
www.avt.pl/prenumerata

Zamów e-prenumeratę (.pdf) na
www.avt.pl/prenumerata/elektroniczne

e-prenumerata roczna z rabatem
15% – 91,80 zł

e-prenumerata dwuletnia z rabatem
30% – 151,20 zł



Prenumeratorzy wersji drukowanej za równoległe e-wydania płacą tylko 20% ceny: 21,60 zł/rok i 43,20 zł/2 lata



Mikrokontrolery do zadań specjalnych

W artykule przedstawiamy przegląd mikrokontrolerów, które wyróżniają się na rynku przez swoją specjalizację. Przyzwyczailiśmy się do tego, że wewnętrzne wyposażenie mikrokontrolerów jest imponująco bogate, że ich moce obliczeniowe i pojemności pamięci minimalizują ograniczenia w działaniach programistów, a praktycznie każdy dostępny model MCU doskonale nadaje się do każdej, nawet bardzo wymagającej aplikacji. Czy rzeczywiście rozwiązania uniwersalne są optymalne dla każdej aplikacji?

Współcześnie produkowane mikrokontrolery bazują głównie na architekturach ARM Cortex-M, tylko niewielka liczba producentów wprowadza na rynek pojedyncze nowe modele układów bazujących na rdzeniach będących natywnymi opracowaniami. Są to głównie mniej lub bardziej udoskonalane rdzenie znane na rynku od lat '90 ubiegłego wieku, jak PIC, AVR, a nawet – ciągle żywy! – rdzeń 8051. Trend konwersji architektur MCU w kierunku Cortex-M jest bardzo silny, do tego poparty niskimi (i malejącymi) cenami nowych układów, produkowanych zazwyczaj w „gęstych” technologiach półprzewodnikowych pozwalających na minimalizację powierzchni krzemu.

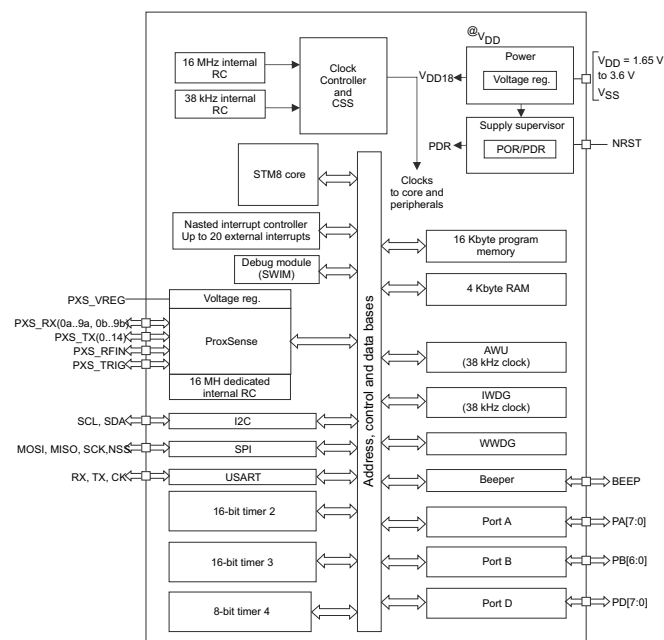
Podobnej unifikacji są poddawane także peryferia integrowane w mikrokontrolerach: producenci integrują w oferowanych układach dużą liczbę modułów peryferyjnych, standardem nawet w niewielkich mikrokontrolerach jest kilka interfejsów komunikacyjnych, timery, przetwornik A/C itp. elementy, niegdyś wymagające bardzo precyzyjnego doboru typu mikrokontrolera do wymogów aplikacji.

W intensywnym trendzie homogenizacji rynku mikrokontrolerów i maksymalizacji ich uniwersalności łatwo przegapić inny, znacznie bardziej interesujący: uwzględnianie przez producentów wymogów specyficznych aplikacji, wymagających wyspecjalizowanych interfejsów

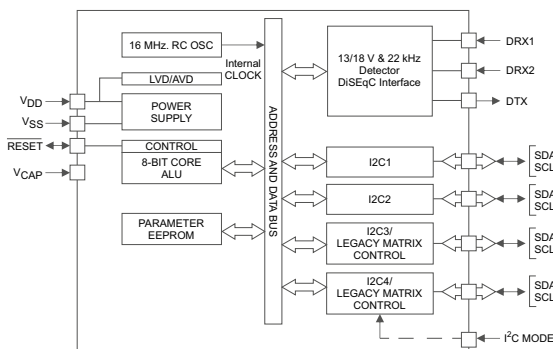
sprzętowych. Kilka wybranych przykładów takich mikrokontrolerów „specjalnych” przedstawiamy w artykule. Kryterium „specjalności” prezentowanych układów wynika z subiektywnej oceny autora, z myślą o czytelnikach EP skupiamy się na rozwiązaniach najbardziej spektakularnych, przygotowanych przez największych rynkowych graczy.

STM8T – ProxSense do usług

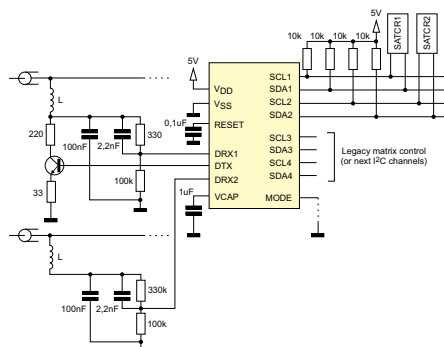
Są to 8-bitowe mikrokontrolery firmy STMicroelectronics, bazujące na rozwiązaniach znanych z produkowanej od wielu lat rodziny STM8. Wyróżniają się one na rynku wbudowanym interfejsem ProxSense,



Rysunek 1. Schemat blokowy mikrokontrolerów STM8T



Rysunek 2. Wewnętrzna budowa mikrokontrolera STM8SPLNB1



Rysunek 3. Schemat aplikacyjny mikrokontrolera STM8SPLNB1

k który umożliwia budowę zaawansowanych nastawników, przełączników i sensorów pojemnościowych o maksymalnej liczbie pól czujników wynoszącej 300. Interfejs ProxSense jest jednym z najdoskonalszych obecnie opracowań bazujących na sensorach pojemnościowych, przeznaczonym do wykrywania obecności i ruchu różnych obiektów w niewielkiej odległości od pola sensorów. Ma wbudowany zaawansowany system autokalibracji oraz kompensacji zmian środowiskowych, dzięki czemu jego niezawodność w realnych aplikacjach jest bardzo wysoka.

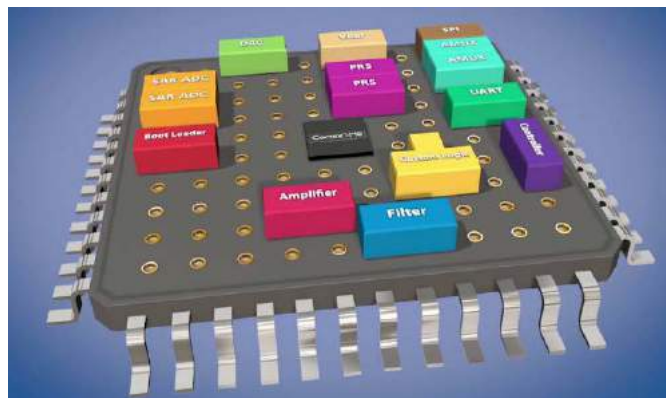
Mikrokontrolery STM8T – poza standardowym rdzeniem – wyposażono także w interfejsy komunikacyjne SPI, I²C i UART, timery oraz pamięć Flash o pojemności 16 kB.

Schemat blokowy mikrokontrolerów STM8T pokazano na **rysunku 1**.

STM8S – specjalizacja w satelitach

Mikrokontroler STM8SPLNB1 z rodziny STM8S jest konstrukcyjnie zoptymalizowany do stosowania w odbiorczych systemach satelitarnych wyposażonych w konwertery zgodne ze standardem DiSEqC. Wewnętrzna budowa STM8SPLNB1 jest mało mikrokontrolerowa (**rysunek 2**) przede wszystkim ze względu na brak wewnętrznych uniwersalnych timerów czy interfejsów komunikacyjnych, poza widocznymi na schemacie blokowym 4 kanałami I²C oraz sprzętowym transceiverem DiSEqC. Interfejsy te służą do konwersji protokołów komunikacyjnych pomiędzy konwerterami satelitarnymi i routerami w systemach odbiorczych.

Prezentowany układ jest dostarczany z oprogramowaniem przygotowanym przez firmę STMicroelectronics, jego działanie można



modyfikować poprzez zmianę tablicy konfiguracyjnej przechowywanej w wewnętrznej pamięci EEPROM.

Schemat aplikacyjny mikrokontrolera STM8SPLNB1 pokazano na **rysunku 3**.

STSPIN32 – kontrolery i sterowniki silników

Układy STSPIN32 firmy STMicroelectronics to zintegrowane sterowniki silników elektrycznych, produkowane w technologii SiP (System in Package), składające się z:

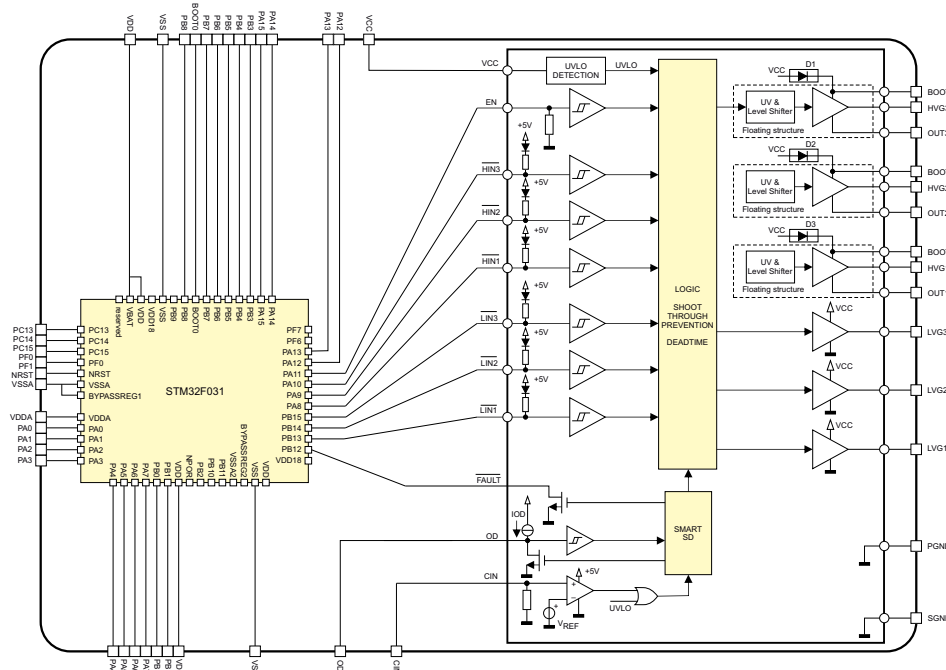
- mikrokontrolera z rdzeniem Cortex-M0,
- driverów bramek dołączanych z zewnątrz tranzystorów mocy,
- modułów pomocniczych, których konfiguracja zależy od typu układu.

Schemat blokowy układu STSPIN32F060x pokazano na **rysunku 4**.

Układy z tej rodziny charakteryzują się integracją jednego z najpopularniejszych mikrokontrolerów z Cortex-M0 z oferty STMicroelectronics – STM32F031 – którego wybrane peryferia są dostępne dla użytkowników. Indywidualne parametry driverów wyjściowych (wydajność prądowa, szybkość przepompowywania ładunku, napięcie blokady itd.), zakresy napięć zasilających itp. cechy i parametry są zależne od konkretnego typu układu, które są przez producenta optymalizowane pod kątem sterowania różnymi typami silników.

Poszerzone zakresy temperatur

Rodzina STM8AF to kolejne układy bazujące na 8-bitowym rdzeniu STM8, ale tym razem zachowawcza, jeśli chodzi o wyposażenie:



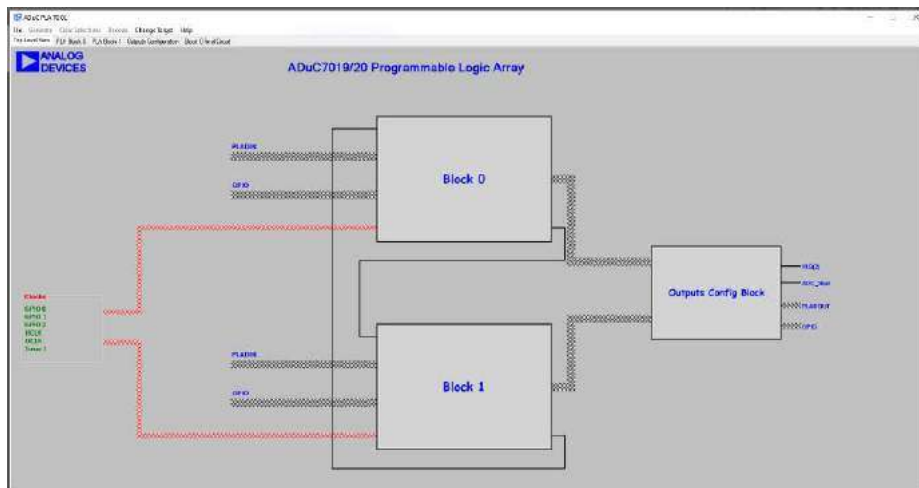
Rysunek 4. Schemat blokowy układu STSPIN32F060x

„specjalność” przejawia się w rozszerzonym zakresie temperatur otoczenia aż do $+150^{\circ}\text{C}$, co czyni je zgodnymi ze standardem AEC-Q100 grade 0.

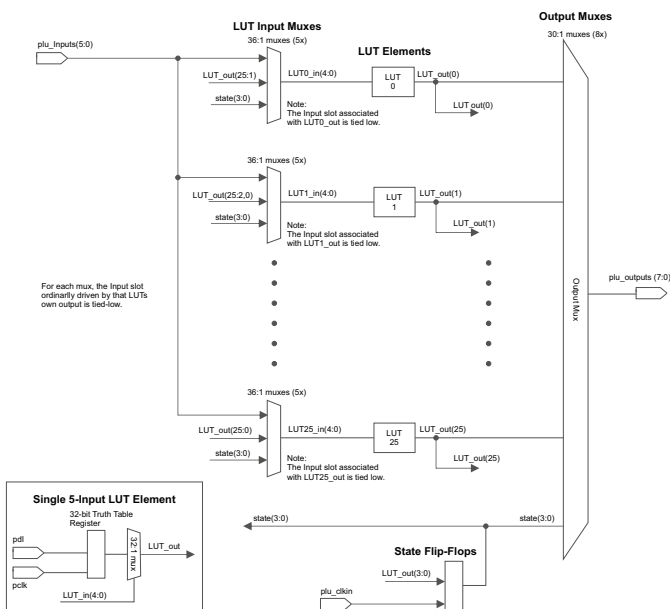
Podobne cechy mają mikrokontrolery:

- S912Z, S9S08G, LPC11xx, LPC12xx firmy NXP,
- TLE984x oraz rodzina Aurix firmy Infineon,
- SPC58 z oferty STMicroelectronics,
- wybrane modele PIC18K/F oraz PIC24H/JEP, ATtiny16x produkowane przez Microchip,
- wybrane modele MSP430F z oferty Texas Instruments.

Firma NXP pobiła także rekord dolnej dopuszczalnej temperatury otoczenia.



Rysunek 6. Okno programu narzędziowego ADuC PLA Tool



Rysunek 5. Schemat blokowy modułu PLU (Programmable Logic Unit) stosowanego w mikrokontrolerach LPC55S6X

którą przesunęła aż do -65°C , radzą sobie z nią wybrane modele mikrokontrolerów z rodziny LPC111x.

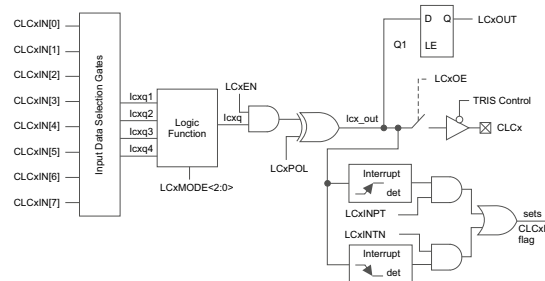
Wewnętrzna logika konfigurowalna

Kilku producentów mikrokontrolerów dostrzegło konieczność wyposażenia produkowanych mikrokontrolerów w bloki logiki konfigurowalnej, która pozwala w prosty sposób optymalizować pracę mikrokontrolera lub jego interfejsów do specyfiki aplikacji. Przykładem tak wyposażonych mikrokontrolerów jest rodzina LPC55S6X firmy NXP, którą wyposażono w konfigurowalny moduł PLU (*Programmable Logic Unit*), którego schemat blokowy pokazano na **rysunku 5**. Budowa tego modułu odpowiada architekturze CPLD bazującej na tablicach LUT z 6 wejściami.

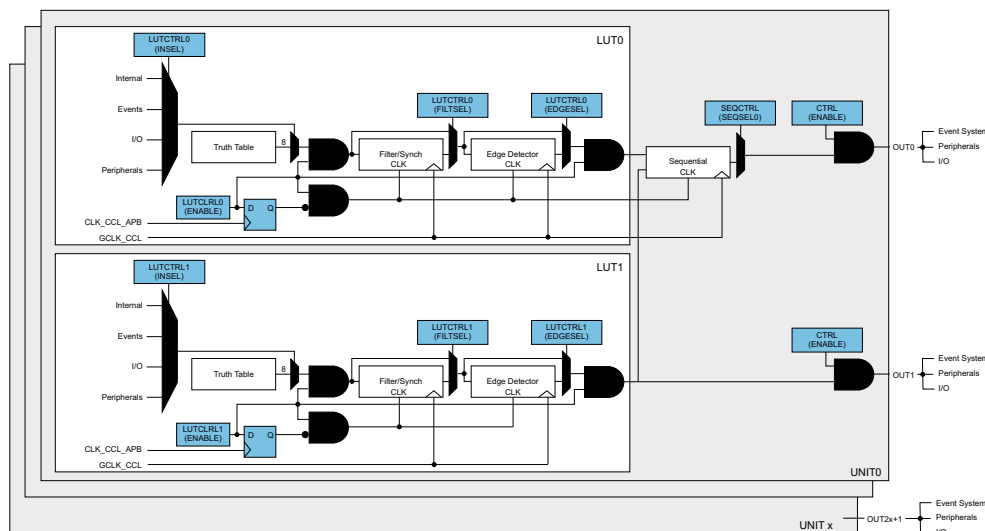
Podobne rozwiązanie oferuje konstruktorom firma

Analog Devices w wybranych modelach mikrokontrolerów ADuC701x/702x/712x, które wyposażono w moduły PLA (*Programmable Logic Array*) o konstrukcji także podobnej do CPLD o 16 komórkach konfigurowalnych. Do konfiguracji bloku PLA producent zaleca używanie programu narzędziowego ADuC PLA Tool (**rysunek 6**), za pomocą którego można wygenerować plik w języku C, zawierający opis wartości odpowiednich rejestrów mikrokontrolera.

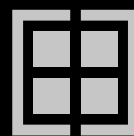
Także Microchip docenił zalety wyposażania mikrokontrolerów w małe bloki logiki konfigurowalnej i kilkanaście modeli mikrokontrolerów PIC10F33x/PIC16F17xx/18xxx/PIC18/dsPIC33 oraz PIC32MM wyposażyć w moduł o nazwie CLC (*Configurable Logic Cell*). Schemat blokowy takiego modułu, zastosowanego w mikrokontrolerach PIC10(L)F320/322 pokazano na **rysunku 7**. Do konfiguracji CLC



Rysunek 7. Schemat blokowy modułu CLC (Configurable Logic Cell) stosowanego w mikrokontrolerach rodziny PIC



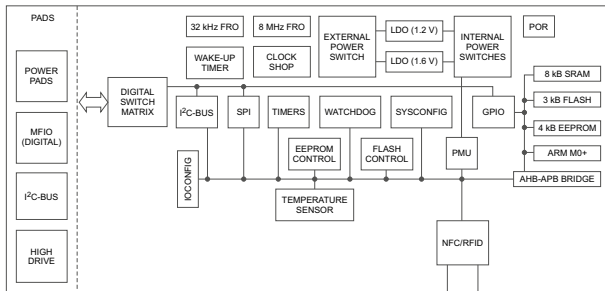
Rysunek 8. Schemat blokowy modułu CCL (Configurable Custom Logic) stosowanego w mikrokontrolerach serii Atmega oraz ATSAM



RUTRONIK
ELECTRONICS WORLDWIDE

RUTRONIK 24
next generation e-commerce

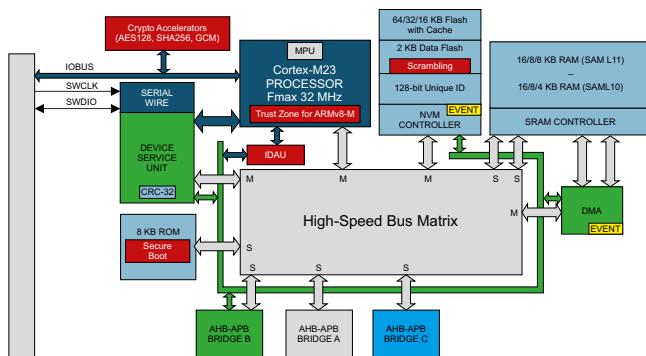
B2B-Shop: rutronik24.com



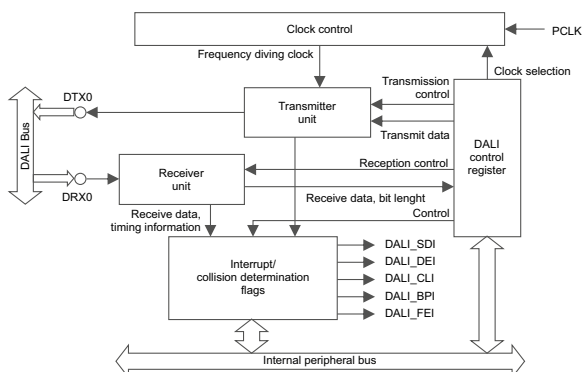
Rysunek 9. Schemat blokowy układu LPC8N04 ze zintegrowanym blokiem NFC

producent zaleca używanie graficznego narzędzia zintegrowanego w programie MPLAB Code Configurator.

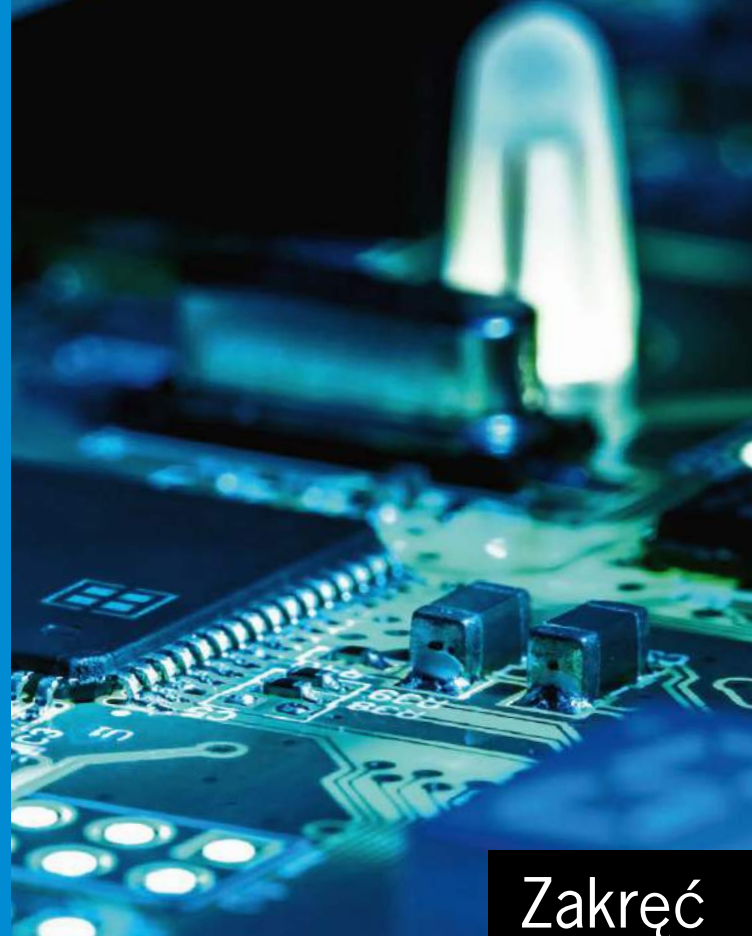
Podobne rozwiązania, ale nazwane CCL (*Configurable Custom Logic*), firma Microchip zastosowała w rodzinach mikrokontrolerów: ATmega4809, AML10/11/21/22, SAMD5x oraz SAME5x. Uproszczony schemat blokowy podsystemu CCL firmy Microchip pokazano na **rysunku 8**.



Rysunek 10. Fragment schematu blokowego z elementami „bezpiecznymi” w CPU w mikrokontrolerach S1JA firmy Renesas



Rysunek 11. Schemat blokowy sprzętowego interfejsu DALI stosowanego w mikrokontrolerach S128 firmy Renesas



Zakręć kołem

Elementy high-tech do Twoich innowacji

Jako jeden z wiodących dystrybutorów elementów elektronicznych, oferujemy na całym świecie szeroki wachlarz produktów, kompetentne wsparcie techniczne podczas opracowywania i projektowania produktów, indywidualne rozwiązania logistyczne oraz kompleksowe usługi serwisowe.

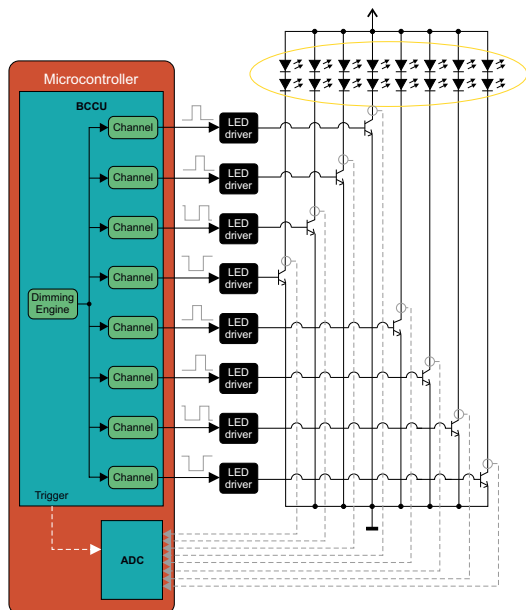
- Półprzewodniki
- Elementy elektroniczne bierne
- Elementy elektromechaniczne
- Wyświetlacze i tablice
- Technologie pamięci
- Technologie bezprzewodowe

Informacje o RUTRONIK:
+48 (32) 461 2000 | www.rutronik.com



Committed to excellence

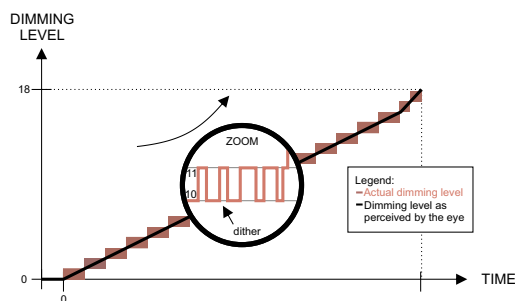
Consult | Components | Logistics | Quality



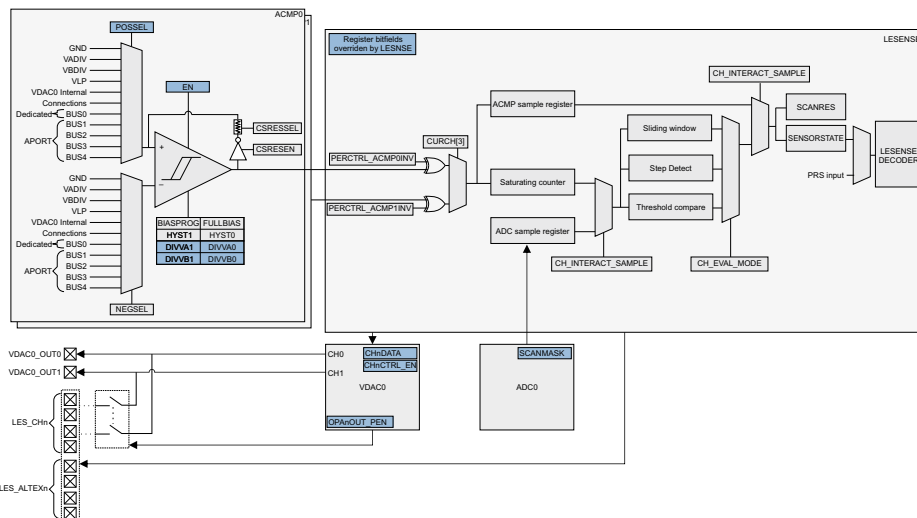
Rysunek 12. Działanie modułów BCCU stosowanych w mikrokontrolerach rodziny XMC firmy Infineon

NFC z MCU

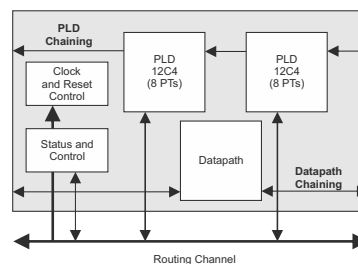
LPC8N04 należy do produkowanej przez NXP rodziny małych układów z rdzeniem Cortex-M0 o nazwie LPC800. Układ LPC8N04 wyposażono w zintegrowany tag NFC (zgodny z ISO14443 type A), który jest dołączony do wewnętrznej magistrali systemowej (rysunek 9). Do mikrokontrolera wystarczy dołączyć antenę NFC i można korzystać (oczywiście po napisaniu odpowiedniego oprogramowania) z zalet tego interfejsu.



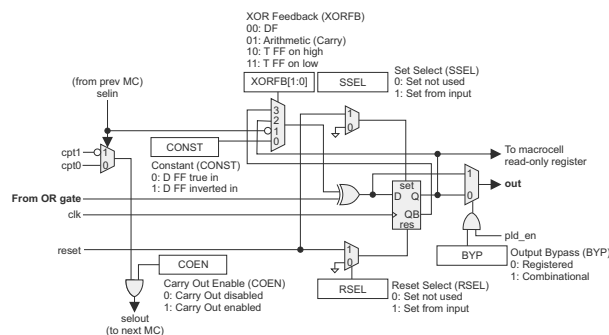
Rysunek 13. Działanie systemu rozmywania stromych zboczy pomiędzy kolejnymi dyskretnymi stopniami jasności LED



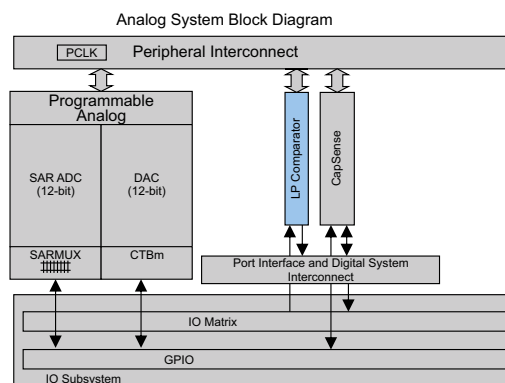
Rysunek 14. Konfigurowalny interfejs LESENSE stosowany w mikrokontrolerach rodziny EFM32GG11 firmy Silicon Labs



Rysunek 15. Budowa bloków UDB stosowanych w mikrokontrolerach PSoc 6 firmy Cypress



Rysunek 16. Komórki PLD, z których składają się bloki UDB



Rysunek 17. Schemat blokowy części analogowej stosowanej w mikrokontrolerach PSoc 6

TrustZone w mikrokontrolerach dla aplikacji IoT

Rosnące wymagania związane z bezpieczeństwem aplikacji działających na współczesnych mikrokontrolerach (które coraz częściej są dołączone do sieci internetowej) spowodowały, że kilka firm wprowadziło układy wyposażone w bezpieczne rdzenie firmy ARM – Cortex-M23 lub Cortex-M33.

Obydwa rdzenie bazują na nowej architekturze ARMv8-M, przy czym Cortex-M23 jest odpowiednikiem Cortex-M0+ w wersji „bezpiecznej”, natomiast Cortex-M33 jest „bezpiecznym” odpowiednikiem rdzenia Cortex-M4. Rdzenie wyposażono w rozszerzenie o nazwie TrustZone, udostępniające aplikacjom dodatkowe instrukcje przydatne w aplikacjach o wymaganym podwyższonym stopniu ochrony danych. Rdzeń Cortex-M23 został użyty w rodzinach mikrokontrolerów: SAML10 i SAML11 firmy Microchip, S1JA firmy Renesas (fragment schematu blokowego z elementami „bezpiecznymi” w CPU pokazano na rysunku 10), NuMicro M23

firmy Nuvoton oraz GD32 firmy GigaDevice. Rdzeń Cortex-M33 został użyty w rodzinach mikrokontrolerów: STM32L5 firmy ST-Microelectronics, LPC5500 z oferty NXP oraz modułach SiP nRF91xx firmy Nordic.

DALI – steruj oświetleniem

Rodzina mikrokontrolerów Synergy S128 (Cortex-M0+) firmy Renesas została – jako jedna z niewielu na rynku – wyposażona w sprzętowy interfejs DALI (*Digital Addressable Lighting Interface*) zgodny ze standardem IEC 62386 version 2. Schemat blokowy tego interfejsu pokazano na **rysunku 11**, jak można zauważyć realizuje on znaczną część protokołu komunikacyjnego DALI.

W odróżnieniu od często proponowanych przez innych producentów mikrokontrolerów programowych implementacji stosu komunikacyjnego DALI, rozwiązanie zastosowane w rodzinie S128 pozwala na odciążenie CPU, dzięki czemu stosunkowo prosty (mało wydajny) rdzeń mikrokontrolera zachowuje moc obliczeniową na potrzeby aplikacji.

Sprzętowe sterowanie LED-ami

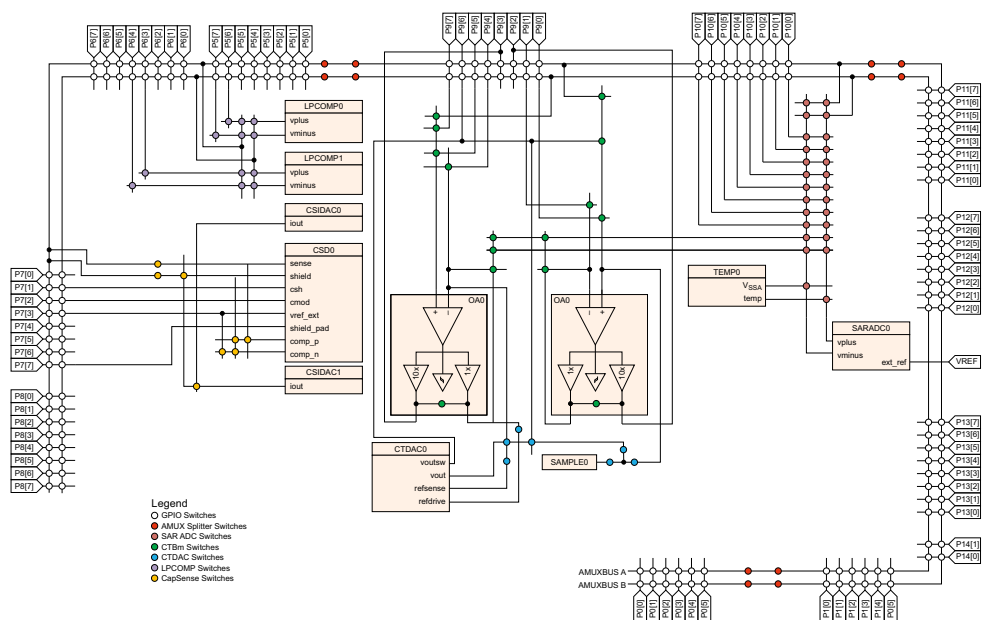
Z myślą o wygodnym sterowaniu LED-ami w aplikacjach sygnalizacyjnych i oświetleniowych firma Infineon wyposaża wybrane mikrokontrolery z rodziny XMC1000 (XMC1200, XMC1202, XMC1302, XMC1402 oraz XMC1404) w moduły BCCU (*Brightness and Color Control Unit*), które pozwalają m.in. na inteligentne sterowanie kolorem emitowanym przez LED-y RGB i RGBW. Interfejs BCCU obsługuje do 8 kanałów LED jednocześnie (**rysunek 12**), każdy kanał może automatycznie sprzętowo obsługiwać rozmywanie stromych zboczy pomiędzy kolejnymi dyskretnymi stopniami jasności LED (**rysunek 13**), zwiększając liniowość regulacji jasności świecenia odbieranej przez ludzkie oko.

Wszystko dla energooszczędności: konfigurowalny interfejs multisensorowy

Na bardzo interesujący pomysł wpadli inżynierowie firmy Silicon Labs: mikrokontrolery z rodziny EFM32GG11 wyposażyli w konfigurowalny interfejs LESENSE (*Low Energy Sensor Interface*) – **rysunek 14**, który może automatycznie monitorować 16 kanałów z dołączonymi czujnikami (pojemnościowymi, rezystancyjnymi, LC, cyfrowymi). Moduł LESENSE, po wykryciu zadanego wcześniej wzorca budzi CPU z trybu oszczędzania energii, co umożliwia programowe obsłużenie wykrytego zdarzenia. Wzorzec wybudzający CPU może składać się z sekwencji zdefiniowanych wartości, jego poprawność jest wykrywana przez dekodery zintegrowane w LESENSE bez udziału CPU. Prezentowany moduł jest natywnym opracowaniem firmy Silicon Labs, która specjalizuje się w ograniczaniu poboru energii przez produkowane przez siebie mikrokontrolery.

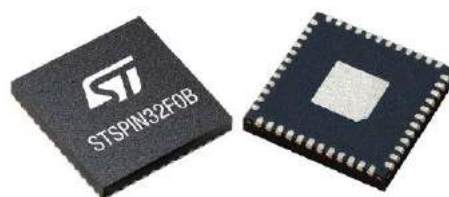
Konfigurowalne peryferia analogowe i cyfrowe

Nieco wcześniej zostały wspomniane w artykule rodziny mikrokontrolerów wyposażonych w konfigurowalne bloki cyfrowe o stosunkowo niewielkiej złożoności. Prekursorem takich rozwiązań jest firma Cypress Semiconductor, która w mikrokontrolerach PSoc (*Programmable System on Chip*) od wielu lat stosuje dwa bloki konfigurowalne: analogowe i cyfrowe. Takie rozwiązanie pozwala na implementację



Rysunek 18. Schemat ilustrujący budowę toru trasującego drogę sygnałów analogowych wewnątrz mikrokontrolera

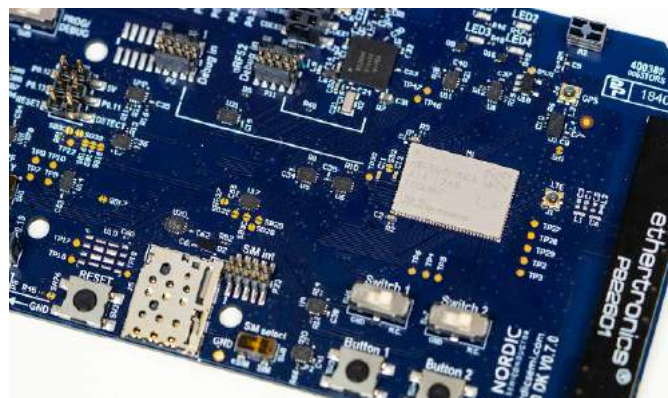
Programmable 3-phase BLDC motor controller
All-in-one solution for power tools



zarówno własnych torów analogowych i cyfrowych, jak i poszerzanie liczby lub możliwości bloków peryferyjnych zaimplementowanych w sprzęcie mikrokontrolerów.

W najnowszej rodzinie mikrokontrolerów PSoc 6 zaimplementowano 12 bloków UDB (*Universal Digital Blocks* – budowę pokazano na **rysunku 15**), składających się z komórek PLD, której schemat pokazano na **rysunku 16**. Budowa jest bardzo bliska klasycznym rozwiązaniom stosowanym w układach CPLD.

Na **rysunku 17** pokazano schemat blokowy części analogowej mikrokontrolerów PSoc 6 z modulem *Programmable Analog*, który można funkcjonalnie skonfigurować w zależności od bieżących potrzeb aplikacji. Na **rysunku 18** pokazano schemat ilustrujący budowę toru trasującego drogę sygnałów analogowych wewnątrz mikrokontrolera oraz możliwości doprowadzenia go do/z pinów układów. Podobne



rozwiązania firma Cypress udostępnia w mikrokontrolerach z rodzin PSoc 4, PSoc 5LP, a także nadal produkowanych PSoc 1 (M8C) i PSoc 3 (rdzeń 8051). Konfiguracja bloków peryferyjnych jest możliwa przy użyciu dostępnego bezpłatnie środowiska programistycznego PSoc Creator oraz PSoc Designer (dla starszych typów mikrokontrolerów).

Do systemów pomiarowych

Ofertę, która nie ma na rynku konkurencji, przygotowała firma Analog Devices. W ofercie produkcyjnej tej firmy są dostępne m.in.:

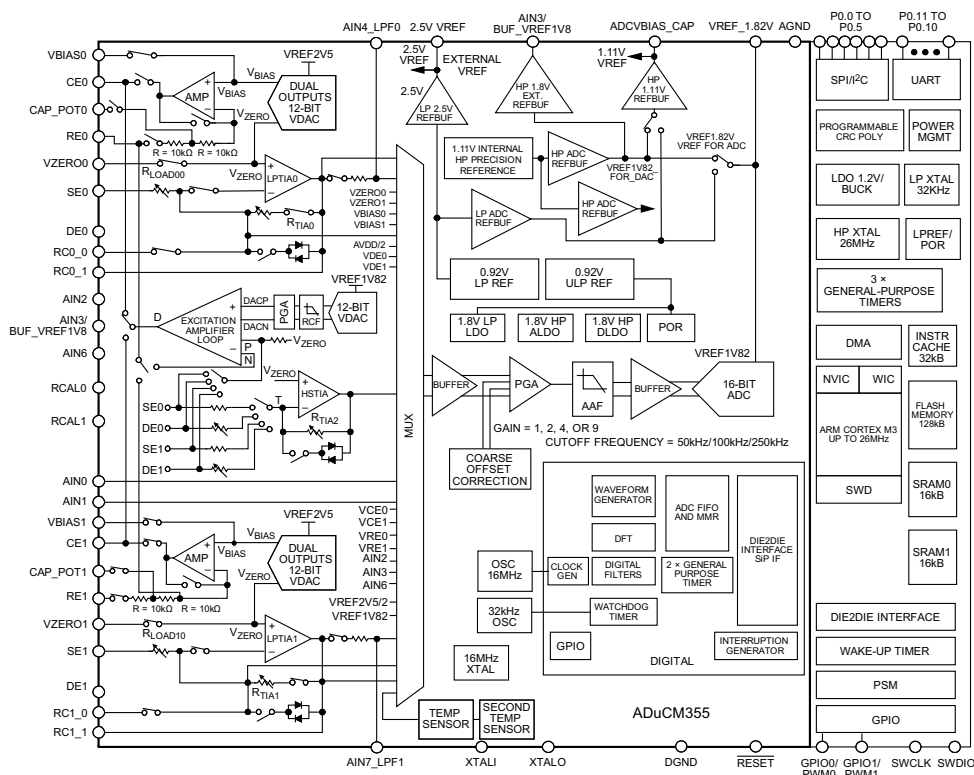
- wyspecjalizowany mikrokontroler ADuCM355 (Cortex-M3) do budowy urządzeń z sensorami chemicznymi (**rysunek 19**),
- mikrokontrolery ADuCM320i oraz ADuCM322, który powstał z myślą o pracy w systemach światłowodowych jako support dla transceiverów laserowych, w czym pomaga mu wbudowany sprzętowy interfejs MDIO,
- ADuCM350, który ma wbudowany zaawansowany analogowy interfejs wejściowy z systemem kondycjonowania sygnału analogowego (**rysunek 20**), a powstał z myślą o budowie liczników mediów, w czym pomaga mu zintegrowany interfejs LCD,
- mikrokontrolery ADuCM330WFS/331WFS (Cortex-M3), przeznaczone do systemów pomiarowych z czujnikami z wyjściami mostkowymi, przy czym zintegrowany przetwornik pomiarowy A/C ma rozdzielczość użytkową 20 bitów.

To oczywiście nie są wszystkie typy mikrokontrolerów do systemów pomiarowych produkowane przez Analog Devices, ale omówienie ich specyficznych możliwości i wyposażenia – ze względu na dużą liczbę ważkich szczegółów – wymaga osobnego artykułu. Pewne jest, że w ofercie Analog Devices można znaleźć najbardziej zaawansowane i dopracowane mikrokontrolery pomiarowe, zarówno z rdzeniami 8051, jak nowszymi: ARM7 oraz Cortex-M3/M4.

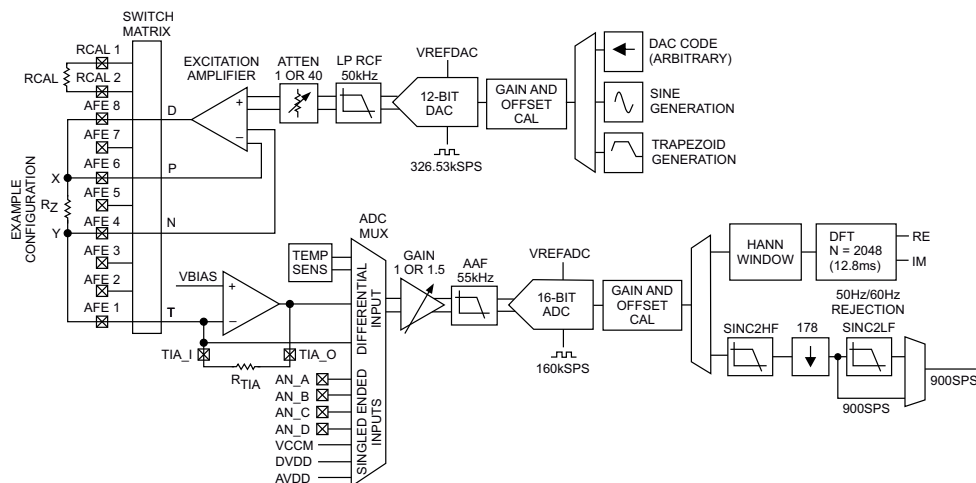
Słowo na koniec

W artykule przedstawiliśmy przegląd współczesnych mikrokontrolerów wyróżniających się na rynku wewnętrznym wyposażeniem, które pozwala nazwać je mikrokontrolerami do zadań specjalnych. W znacznym stopniu pominięte zostały układy zintegrowane z torami radiowymi (prawdopodobnie poświęcimy im osobny artykuł przeglądowy), a także rozwiązania mało perspektywiczne z punktu widzenia elektronika-praktyka, jak choćby układy DSC produkowane przez Texas Instruments czy rodziny mikrokontrolerów przeznaczone na rynek automotive SPC56/58.

Piotr Zbysiński, EP



Rysunek 19. Schemat blokowy wyspecjalizowanego mikrokontrolera ADuCM355 przeznaczonego do budowy urządzeń z sensorami chemicznymi



Rysunek 20. Schemat blokowy analogowego interfejsu wejściowego z systemem kondycjonowania sygnału stosowanego w mikrokontrolerach ADuCM350

Renesas Synergy™ Platform Delivers Security and Device-to-Cloud Connectivity



Mikrokontrolery specjalizowane od Nuvoton

Firma Nuvoton Technology Corporation jest producentem układów logicznych do komputerów oraz układów logicznych elektroniki użytkowej dużej skali integracji, wśród których są mikrokontrolery, specjalizowane układy SoC, układy audio oraz układy dźwiękowe nagrywająco/odtwarzające. Firma Marthel jest autoryzowanym dystrybutorem i posiada w swojej ofercie szeroką gamę produktów firmy Nuvoton.

Magistrala CAN

Rodzina mikrokontrolerów NuMicro typu NUC13x/23x to seria układów bazujących na rdzeniu Cortex-M0 ze zintegrowanym kontrolerem magistrali CAN (Controller Area Network). Zostały zaprojektowane dla aplikacji CAN, ale także wyposażone w szereg urządzeń peryferyjnych takich jak: LIN, USB 2.0 FS, UART, I²C i ADC, komparator analogowy, reset niskiego napięcia i detektor zaniku napięcia. Dostępne z pamięcią flash o pojemności 32...128 KB, pamięcią SRAM 4...16 KB i pamięcią ładującą (ROM) Flash 4...8 KB do programowania systemowego (ISP). Przystosowane są do pracy z taktowaniem do 50 lub 72 MHz w zakresie temperatur -40...85°C lub nawet do 105°C.

Interfejs DALI

W nowoczesnych systemach sterowania oświetleniem coraz częściej stosuje się interfejs DALI. Przy współpracy z dedykowanymi urządzeniami staje się idealny do zastosowania w instalacjach inteligentnych we wszelkiego rodzaju budynkach. Budowa, cechy i możliwości sprawiają, że jest to system uniwersalny, o bardzo dużym potencjale do rozbudowy i przeznaczony do współpracy z każdym rodzajem oświetlenia.

Nuvoton wspiera rozwój produktów z interfejsem DALI. Udostępnia bibliotekę, która została przygotowana w celu ułatwienia budowy urządzeń z tym interfejsem oraz oferuje płytkę uruchomieniową DALI 2nd Slave, przygotowaną do pracy ze standardem DALI. Płytkę bazuje na mikrokontrolerze NDA102 z rodziny NuMicro. Są to układy z rdzeniem Cortex-M0 do zastosowań przemysłowych, które wymagają wysokiej wydajności, wysokiej integracji i niskich kosztów. Działają z częstotliwością do 48 MHz i oferują 29,5 KB wbudowanej pamięci Flash (współdzielonej z Data Flash) i 4 KB SRAM. Mają wiele układów peryferyjnych, m.in. timery, UART, SPI, I²C, PWM, ADC, watchdog, komparator analogowy, dzielnik sprzętowy i wykrywacz braku zasilania. Szybka moc obliczeniowa układów pozwala na szybszą kontrolę urządzeń i budowę złożonych aplikacji.

Streaming video

Układy serii N3292x zostały zaprojektowane specjalnie do wydajnego streamingu video/audio. Wbudowane kodek wideo H.264 i akcelerator kompresji/dekompresji dźwięku pozwalają na osiągnięcie maksymalnej wydajności przy jednoczesnym obciążeniu procesora i zrównoważonym poborze energii. Przeznaczone są do zastosowania w kamerach IP, urządzeniach typu Video Baby Monitor oraz zaawansowanych interfejsach HMI (Human Machine Interface).

Układy bazują na procesorze z rdzeniem ARM926EJ-S pracującym z częstotliwością do 240 MHz. Lista bloków peryferyjnych

Więcej informacji:

PDW MARTHEL
ul. Sosnowa 24-5, Bielany Wrocławskie
55-040 Kobierzyce
tel. 71 311 07 11, 71 311 07 12, faks 71 311 07 13
marthel.pl, marthelinfo@marthel.pl



obejmuje m.in.: kodek wideo (H.264), Ethernet MAC, kodek JPEG, interfejs kamery CMOS, 32-kanalowy SPU (moduł przetwarzania dźwięku), ADC, DAC, koder TV oraz interfejs host/device USB2.0 HS. Aby zmniejszyć złożoność systemu przy jednoczesnym obniżeniu kosztów komponentów, układy produkowane są w 128 pinowej obudowie LQFP, w której, dzięki technologii MCP (Multi-Chip Package), znajduje się także pamięć DDR2 32 Mbit × 16.

Audio SoC

Rodzina układów Nuvoton Audio SoC to wysokowydajne ale zrównoważone ekonomicznie rozwiązanie, które integruje mikrokontroler z wysokiej jakości peryferiami i blokami funkcyjnymi audio w jednym układzie scalonym. Układy tej serii idealnie nadają się do realizacji takich funkcji jak nagrywanie i odtwarzanie dźwięku, rozpoznawanie głosu, wykrywanie słów kluczowych i innych wymagających aplikacji audio. Wysoce zintegrowana architektura układów eliminuje potrzebę stosowania różnych układów peryferyjnych, takich jak wzmacniacze czy kodeki audio.

Seria AUI Enabled (Audio User Interface Enabled) dzieli się na dwie grupy. Układy serii ISD91xx wyposażone są w 32-bitowy procesor Cortex-M0 oraz takie peryferia jak interfejs mikrofonu, cyfrowy interfejs audio I²S, wzmacniacz głośnikowy, regulator napięcia czy wbudowana pamięć flash. Zostały zaprojektowane do realizacji systemów audio optymalizowanych pod kątem ekonomicznym, dla zastosowań konsumenckich i przemysłowych.

Układy serii ISD94xxx zostały zaprojektowane do aplikacji wymagających dużej mocy obliczeniowej. Wyposażone są w procesor Cortex-M4 taktowany zegarem do 200 MHz ze zintegrowanym rozszerzeniem DSP i jednostką zmiennoprzecinkową FPU. Do peryferiów układu należą m.in. 12-bitowe ADC oraz interfejs USB, ponadto mogą obsługiwać wiele urządzeń peryferyjnych audio I²S, DMIC i modulator audio DPWM. Dzięki pamięci flash o pojemności do 512 KB i pamięci SRAM o pojemności do 192 KB są to idealne układy do wielu zastosowań przetwarzania dźwięku oraz interfejsów komunikacyjnych.

Bezpieczeństwo funkcjonalne: normy, analizy i sposoby jego uzyskiwania w platformach mikrokontrolerowych

W latach '80 XX wieku zaczęło rosnąć wykorzystanie elektronicznych komponentów programowalnych, takich jak mikrokontrolery (MCU) i mikroprocesory, w przemysłowych systemach sterowania. Międzynarodowa Komisja Elektrotechniczna (IEC) powołała wówczas specjalne grupy badawcze. Miały one na celu zbadanie bezpieczeństwa funkcjonalnego tych systemów i opracowania wytycznych dotyczących rozwoju bezpiecznych urządzeń je wykorzystujących. Od tego czasu zgodność z normami bezpieczeństwa funkcjonalnego stała się wymogiem dla twórców urządzeń elektronicznych m.in. w przemyśle motoryzacyjnym czy w zastosowaniach przemysłowych.

Termin „bezpieczeństwo funkcjonalne” pojawił się w dokumentach stworzonych przez Podkomitet 65A IEC w 1985 roku. Urząd Dozoru Technicznego (UDT) definiuje bezpieczeństwo funkcjonalne, jako „ogólne podejście do wszystkich działań w cyklu życia bezpieczeństwa systemów zawierających elektryczne i/lub elektroniczne i/lub programowalne elektroniczne elementy składowe”. W ogólności zasady te mówią o sposobie projektowania urządzeń elektronicznych, by były one bezpieczne dla użytkownika. Jest to szczególnie ważne np. w systemach zabezpieczających w przemyśle, motoryzacji, lotnictwie czy kolei.

Pierwszą normą przeznaczoną do stosowania w szerokiej gamie systemów przemysłowych była **IEC 61508**, która została opublikowana w 1998 roku. W 2011 roku ukazała się motoryzacyjna norma bezpieczeństwa funkcjonalnego **ISO 26262**. Celem obu norm jest ustalenie wymagań, które zmniejszają potencjalne ryzyko obrażeń fizycznych czy uszczerbku na zdrowiu ludzi z powodu awarii danego produktu. Normy te wymagają wdrożenia ścisłego procesu rozwoju produktu, przeprowadzenia wstępnej analizy zagrożeń i ryzyka oraz wdrożenia zabiegów zmniejszających ryzyko, związane z produktem. Wykorzystuje się je zarówno w projekcie sprzętu, jak i jego oprogramowania. Normy te współdzielą wiele elementów, mają bardzo zbliżony cykl rozwoju produktu i wiele takich samych rozwiązań wspierających projektowanie. Istnieje jednak wiele innych, branżowych norm bezpieczeństwa, opierających się na tych dwóch normach. Są one zazwyczaj zgodne z **IEC 61508**, uzupełniając ją o dodatkowe wymagania architektoniczne czy obecność autotestu dla systemu w konkretnych zastosowaniach. W **tabeli 1** zebrano podsumowanie różnych międzynarodowych norm bezpieczeństwa funkcjonalnego i ich związku z normą **IEC 61508**.

Aby dany projekt mógł spełniać normy bezpieczeństwa funkcjonalnego, musi istnieć ocena bezpieczeństwa funkcjonalnego urządzenia końcowego, a także wszelkich jego elementów krytycznych dla bezpieczeństwa funkcjonalnego, które są zastosowane w projekcie, a które są już zgodne z tymi normami. Końcowa ocena zgodności

jest niezależna od zespołu projektowego, a wymagany poziom bezpieczeństwa określają obowiązujące normy. Ocena taka może być przeprowadzona na przykład przez niezależną grupę roboczą w przedsiębiorstwie lub przez niezależne organizacje, takie jak Technischer Überwachungs-Verein (TÜV) czy Underwriters Laboratories (UL).

Najprostszą drogą do spełnienia norm, jest wykorzystanie sprawdzonych i zgodnych z normami podzespołów. Na rynku dostępnych jest wiele precertyfikowanych układów i elementów elektronicznych, spełniających normy bezpieczeństwa funkcjonalnego. W dalszej części artykułu przyjrzymy się, jak z zagadnieniem tym radzą sobie mikrokontrolery.

Mikrokontrolery z certyfikatem bezpieczeństwa

Normy bezpieczeństwa funkcjonalnego są zazwyczaj stosowane do elementów elektrycznych, elektronicznych i programowalnych elektronicznych systemów bezpieczeństwa w urządzeniu końcowym. Podczas certyfikacji sprawdzane są m. in. poszczególne komponenty układu na zgodność z normami. Mikrokontrolery są złożonymi i kluczowymi elementami wielu systemów bezpieczeństwa funkcjonalnego, w związku z czym są często dogłębnie analizowane w procesie certyfikacji. Często zadawane są pytania takie jak:

- Czy dany układ opracowano zgodnie ze standardem bezpieczeństwa funkcjonalnego?
- Jak mikrokontroler radzi sobie z przypadkowymi awariami?
- W jaki sposób układ radzi sobie z awariami systematycznymi?

Ponieważ normy **IEC 61508** i **ISO 26262** wymagają uwzględnienia bezpieczeństwa funkcjonalnego od samego początku fazy rozwoju produktu, bardzo ważne jest, aby wybrać mikrokontroler, który został zaprojektowany do zastosowań w zakresie tego rodzaju systemów. Takie układy mogą same posiadać odpowiednie certyfikaty np. dla konkretnego poziomu SIL. Precertyfikowany mikrokontroler, to taki układ, którego projekt został sprawdzony i nadaje się do stosowania w systemach bezpieczeństwa funkcjonalnego.

IEC 61508 przewiduje certyfikację mikrokontrolerów od SIL 1 (najniższy) do SIL 3. Tylko cały system może otrzymać najwyższy poziom – SIL 4. Analogicznie **ISO 26262** wykorzystuje wskaźnik ASIL, od A do D. ASIL A jest najniższym poziomem, a ASIL D najwyższym. Zgodnie z **ISO 26262**, mikrokontrolery mogą być certyfikowane na wszystkich czterech poziomach ASIL. Normy wymagają, aby certyfikowane układy dostarczały odpowiednią dokumentację i narzędzia, (takie jak instrukcje bezpieczeństwa, tryb awaryjny lub rozbudowana diagnostyka automatyczna). Ma to pomóc użytkownikom zrozumieć mechanizmy bezpieczeństwa układu i obliczyć odpowiednie wskaźniki.

Przykładami precertyfikowanych mikrokontrolerów są układy z rodziny Hercules firmy Texas Instruments – TMS570 oraz RMxx, które dedykowane są do stosowania wraz z **IEC 61508** do SIL3 i **ISO 26262** do ASIL D, oraz mikrokontrolery Infineon Aurix, precertyfikowane do ASIL D. Są to najwyższe poziomy nienaruszalności bezpieczeństwa w normach. Chociaż w większości norm bezpieczeństwa

Tabela 1. Branżowe normy bezpieczeństwa funkcjonalnego i ich relacje z IEC 61508

Norma	Branża	Integralność bezpieczeństwa	Metryka oceny architektury	Wymagania stawiane architekturom	Współczynnik awaryjności	Specyficzne wymagania stawiane autotestowi mikrokontrolera
IEC 61508	Systemy programowalne	SIL – 1, 2, 3, 4	SFF	HFT > 0 dla SIL 4	PFD, PFH	brak
ISO 26262	Motoryzacja	ASIL – A, B, C, D	SPFM/LFM	brak	PMHF	brak
EN 50129	Kolejowa	SIL – 1, 2, 3, 4	-	Wg IEC 61508	THR	CPU, pamięć
ISO 22201	Windy	SIL – 1, 2, 3	-	Dwukanałowe dla SIL 3	-	CPU, pamięć, przerwania, zegar, interfejsy IO i komunikacyjne
IEC 61800	Sterowniki silników elektrycznych	SIL – 1, 2, 3 (SIL 4 wg IEC 61508)	SFF	Zależnie od funkcji	PFH (brak PFD)	brak
IEC 62061	Maszyzny	SIL – 1, 2, 3 (SIL 4 wg IEC 61508)	SFF	Wsparcie dla kategorii ISO 13849	PFH _D	brak
IEC 61511	Automatyka	SIL – 1, 2, 3 (SIL 4 wg IEC 61508)	SFF	Wg IEC 61508	PFD _{sr}	brak
ISO 13849	Maszyzny	PL – a, b, c, d, e	DC _{sr}	Kategorie B, 1, 2, 3, 4	MTTF _D	brak
IEC 60730	Sprzęt domowy	Klasy A, B, C	brak	Tylko dla klasy C	brak	CPU, pamięć, przerwania, zegar, interfejsy IO i komunikacyjne

Skrót pojęcia	Objaśnienie
SIL, ASIL	Poziom integralności bezpieczeństwa; ogólny opis przynależności do jednej z dyskretnych klas bezpieczeństwa funkcjonalnego, gdzie 1 to najniższa klasa, a 4 to klasa najwyższa
SFF	Odsetek niegroźnych awarii; stosunek ilości nieszkodliwych awarii urządzenia (w tym wykrytych i zdiagnozowanych błędów) do całkowitej częstotliwości awarii systemu
SPFM	Prawdopodobieństwo wystąpienia błędu pojedynczego komponentu systemu bezpieczeństwa, który nie zostanie wykryty i w ten sposób uczyni system bezpieczeństwa funkcjonalnego niezdolnym do działania
LFM	Prawdopodobieństwo wystąpienia błędów utajonych wielu elementów, które nie są dostrzegane przez kierowcę, ani system bezpieczeństwa, a jednocześnie powodują awarię systemu bezpieczeństwa
DC _{sr}	Średnie (w czasie) pokrycie diagnostyczne systemu; stosunek wykrywanych błędów do ogólnej ilości awarii monitorowanego systemu
PFD	Prawdopodobieństwo nie zadziałania urządzenia na żądanie
PFD _{sr}	Średnie prawdopodobieństwo nie zadziałania urządzenia na żądanie
PFH _D	Prawdopodobieństwo (na godzinę) wystąpienia groźnej awarii urządzenia
PMHF	Prawdopodobieństwo wystąpienia dowolnego, losowego błędu w systemie (suma m.in. SPFM, LFM)
THR	Współczynnik tolerowalnego zagrożenia
MTTF _D	Średni czas pomiędzy niebezpiecznymi awariami
HFT	Odporność na uszkodzenia sprzętu, tj. maksymalna ilość dopuszczalnych usterek, która nie spowoduje utraty funkcji bezpieczeństwa systemu

funkcjonalnego zasadniczo nie wymaga się, aby system wykorzystywał tego rodzaju elementy, taki zabieg może znacznie zredukować wysiłek i koszty certyfikacji systemu. Komponenty takie są dostarczane wraz z niezbędnymi danymi i dokumentacją potrzebną do certyfikacji urządzenia.

Zarządzanie przypadkowymi awariami systemu

Pierwszym krokiem jest zrozumienie modów awarii i oszacowanie ich prawdopodobieństwa. Wymaga to przeprowadzenia analizy zagrożeń i oceny ryzyka zgodnie z wymogami odpowiednich norm. Deweloperzy muszą określić funkcje bezpieczeństwa i poziom redukcji ryzyka (SIL/ASIL) odpowiedni dla ich systemu. Następnie, w oparciu o niego, programiści muszą zdefiniować odpowiednią architekturę i wskaźniki. Konieczne są również odpowiednie mechanizmy

bezpieczeństwa – diagnostyka – implementowane do momentu osiągnięcia wystarczającej redukcji ryzyka.

Aby zrozumieć sposób, w jaki systemy bezpieczeństwa radzą sobie ze swoją awaryjnością, prześledźmy trzy zagadnienia z tym związane:

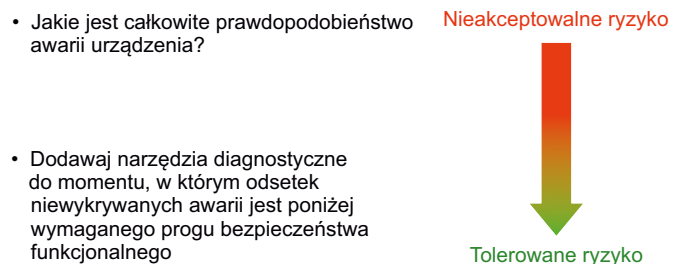
- Tryby awarii i wskaźniki awarii.
- Wskaźniki bezpieczeństwa funkcjonalnego architektury i parametry opisujące losowe awarie sprzętu.
- Mechanizmy bezpieczeństwa, pozwalające zredukować prawdopodobieństwo niebezpiecznej awarii.

Tryby awarii i wskaźniki awarii

Rozważmy nowoczesny system sterowania samochodem lub produktem przemysłowym, który posiada elektroniczne elementy programowalne i obwody wejścia/wyjścia. Zrozumienie trybów awarii tych elementów będzie pomocne w oszacowaniu całkowitego wskaźnika awaryjności produktu. Istnieją trzy główne tryby awarii wpływające na mikrokontroler w tym przypadku:

Trwała awaria obudowy układu, taka jak np. uszkodzenie jej z powodu różnej rozszerzalności cieplnej elementu i płytki drukowanej (PCB).

- Trwała awaria struktury półprzewodnikowej, taka jak rozwarcie lub zwarcie ścieżek metalizacji, wyciek ze złącza któregoś z tranzystorów, zablokowanie bramek i inne problemy wynikające ze zużycia struktury.
- Przejściowa awaria układu, taka jak na przykład wywołane promieniowaniem kosmicznym odwrócenie pojedynczego bitu w pamięci SRAM.



Rysunek 1. Ogólny mechanizm zarządzania bezpieczeństwem i awaryjnością urządzenia

Tabela 2. Wymagania poszczególnych klas ASIL dotyczące parametrów SPFM, PMHF i LFM

Klasa ASIL	SPFM	PMHF(FIT)	LFM
ASIL B	>90%	<100	>60%
ASIL C	>97%	<100	>80%
ASIL D	>99%	<10	>90%

Definiuje się tzw. wskaźnik awaryjności (FIT), gdzie FIT=1 oznacza jedną awarię na miliard godzin pracy. FIT to odwrotność średniego czasu pomiędzy awariami (MTBF), który typowo wykorzystywany jest do szacowania awaryjności w elektronice.

Norma IEC/TR 62380 zapewnia model matematyczny do oszacowania wskaźnika trwałej awaryjności obudowy układu na podstawie warunków jego użytkowania. Model pozwala również na oszacowanie prawdopodobieństwa trwałej awarii samego półprzewodnika na podstawie złożoności elementu i warunków jego eksploatacji, takich jak temperatura, czas cykli pracy i tym podobne. Ponadto większość producentów układów scalonych publikuje tego rodzaju wskaźniki trwałości dla swoich produktów, podane jako dane w FIT lub MTBF. Dane te są obliczane na podstawie badań niezawodności układu przeprowadzonych przez producenta.

Nie ma jednak znormalizowanej w branży metody szacowania wskaźnika awaryjności przejściowej. Idealnie, dane do oszacowania powinny pochodzić z rzeczywistych eksperymentów z wykorzystaniem tych układów. Texas Instruments dostarcza takie dane dla układów z rodziny Hercules – opiera się na danych zebranych z chipów testowanych w Los Alamos National Laboratory i wykorzystuje standard testowy JEDEC JESD89A. Ponieważ awaryjność przejściowa mikrokontrolerów może być od jednego do trzech rzędów wielkości wyższa niż awaryjność stała, precyzyjne dane pomagają zminimalizować ryzyko niedoszacowania ogólnego wskaźnika awaryjności. Dzięki temu, że tego rodzaju dane są dostępne, elektronicy projektujący systemy bezpieczeństwa funkcjonalnego nie muszą samodzielnie badać awaryjności mikrokontrolerów. Obniża to koszty i redukuje czas, potrzebny do stworzenia tego rodzaju systemu.

Wskaźniki bezpieczeństwa funkcjonalnego architektury i parametry opisujące losowe awarie sprzętu

Projektanci systemów muszą przeprowadzić analizę zagrożeń i ocenę ryzyka w celu oszacowania poziomu redukcji ryzyka wymaganego dla danej aplikacji. Wynik oceny jest celem bezpieczeństwa, na przykład klasyfikacja SIL 1...4, jak opisano w IEC 61508 lub poziom ASIL A...D w ISO 26262. W odniesieniu do elementów półprzewodnikowych, najwyższy możliwy do osiągnięcia poziom SIL to SIL 3; SIL 4 jest osiągalny tylko dla całego urządzenia, a nie jego składowych.

Normy określają wskaźniki, które służą do oceny mechanizmów bezpieczeństwa implementowanych tak na poziomie architektury, jak i rozwiązań zmniejszających ryzyko awarii czy prawdopodobieństwo wystąpienia błędów. W tym celu wykorzystywane są parametry takie jak SPFM (uszkodzenia jednopunktowe), LFM (uszkodzenia utajone) i SFF (odsetek niegroźnych uszkodzeń). Są to współczynniki pozwalające mierzyć architektoniczną skuteczność redukcji awaryjności. Z kolei parametry takie jak PMHF (prawdopodobieństwo wystąpienia losowych awarii) i PFH (prawdopodobieństwo awarii na godzinę) to parametry probabilistyczne pokazujące ogólny poziom ryzyka.

Przykłady zastosowania tych wskaźników przedstawiono w tabeli 2. Pokazuje ona wymagania dla poszczególnych klas ASIL (dla normy ISO 26262), dotyczące pojedynczego punktu uszkodzenia (SPFM), ukrytego wskaźnika uszkodzenia (LFM) i probabilistycznego opisu losowych awarii sprzętu (PMHF). Tabela 3 z kolei pokazuje wymagania normy IEC 61508 dla poszczególnych klas SIL (typ B) dla SFF przy tolerancji na awarię sprzętu (HFT) równej zero i dla prawdopodobieństwa awarii na godzinę (PFH).

Tabela 3. Wymagania poszczególnych klas SIL dotyczące parametrów SFF oraz PFH

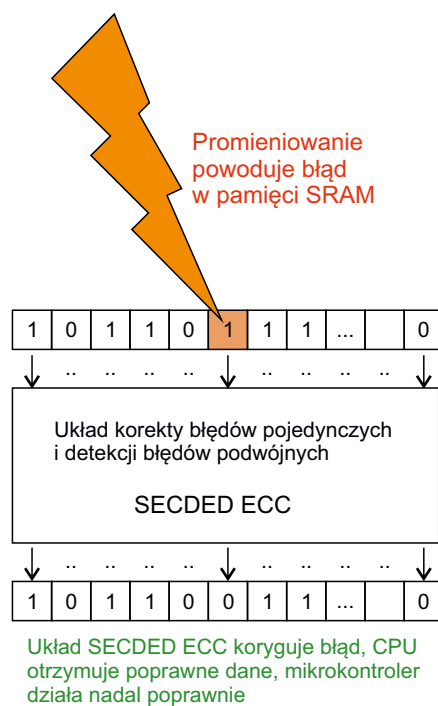
Klasa SIL (typ B)	SFF	PFH (FIT)
SIL 1	60..90%	<10000
SIL 2	90..99%	<1000
SIL 3	>99%	<100

Mechanizmy bezpieczeństwa, pozwalające zredukować prawdopodobieństwo niebezpiecznej awarii

Znając wymagania dotyczące redukcji ryzyka, twórcy systemów muszą przestrzegać różnych procedur, aby uzyskać pewność, że osiągnięto cele dotyczące wskaźników losowej awaryjności. Dlatego też stosowane są mechanizmy bezpieczeństwa, które mają pomóc w wykrywaniu awarii. Funkcje krytyczne dla bezpieczeństwa wymagają monitorowania działania mikrokontrolera w czasie rzeczywistym. Projektant powinien wykorzystywać te mechanizmy, aby skonfigurować system tak, aby po wykryciu usterki system naprawił usterkę albo układ przechodzi w stan bezpieczny.

Poniżej znajduje się kilka przykładów wykorzystywanych w mikrokontrolerach mechanizmów. Rysunek 2 prezentuje obwód korekcji błędów pamięci ECC. Obwód SECDED pozwala na korekcję pojedynczego błędu (pojedynczego bitu) – SEC – lub detekcję błędów podwójnych – DED. Błędy w pamięci SRAM powstają np. na skutek promieniowania kosmicznego, bądź innych losowych zdarzeń, które są w stanie zmienić wartość jednej z komórek pamięci operacyjnej, powodując błędy w danych. System ECC pozwala na korekcję lub wykrywanie tych awarii.

Tak jak w przypadku pamięci SRAM, np. kwant promieniowania kosmicznego, po uderzeniu w rdzeń mikrokontrolera może spowodować zmianę jakiegoś losowego bitu. Oczywiście spowoduje to niepoprawne działanie układu. Problemy te zasadniczo rozwiązuje się z pomocą redundancji – dodając drugi, równoległe pracujący układ. W ten sposób otrzymamy system, w którym uszkodzenie jednego z dwóch elementów nie powoduje jeszcze katastrofalnej awarii. Niestety w przypadku mikrokontrolerów nie jest to takie proste. Nie da się ich w prosty sposób zrównoleglić, bo nie jest możliwe wykrycie ich awarii – jeśli monitorujemy np. wyjście CPU, nie wiemy jakie ono powinno być, jeśli nie wykonamy takiej samej operacji, efektywnie



Rysunek 2. Działanie obwodu korekcji błędów SECDED ECC dla pamięci SRAM w mikrokontrolerze

potrzebując dwóch CPU. Nawet w takiej sytuacji, nie wiemy, który z równolegle pracujących mikrokontrolerów uległ awarii. Dlatego też stosuje się układy głosujące – w systemie implementuje się więcej procesorów, które „głoszą” nad poprawnym wynikiem. W większości przypadków każdy procesor wykonuje operacje tak samo, a każde odstępstwo od tego jest uznawane za awarię.

Architektury z głosowaniem i/lub redundancją określa się jako N spośród M – N out of M – NooM, co oznacza, że w systemie znajduje się ogólnie M podmodułów, a do awarii całego systemu potrzebna jest awaria co najmniej N podmodułów. W systemie takim znajduje się M głosujących, równorzędnych podmodułów. HFT takiego układu równe jest M – N.

Na **rysunku 3** zaprezentowano, wykorzystywany przez mikrokontrolery z rodziny TI Hercules układ dwóch rdzeni w architekturze nazwanej 1oo1D. Pozwala ona na eliminację wpływu błędów losowych na działanie systemu oraz bardzo ziarnistą diagnostykę procesora. Od klasycznej architektury 1oo1 układ ten odróżnia się posiadaniem dwóch równoległych rdzeni, których wyniki pracy porównywane są ze sobą przez specjalny moduł CCM, będący częścią układów diagnostycznych w mikrokontrolerze.

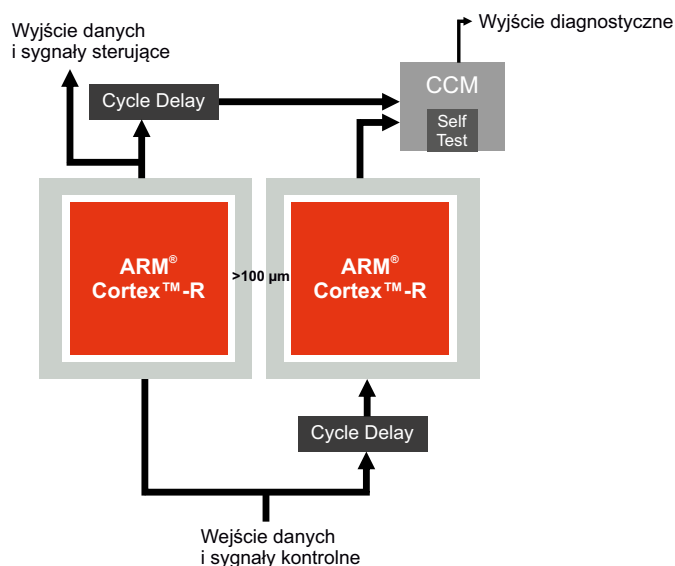
W układach z serii TI Hercules stosuje się dwa rdzenie ARM Cortex-R w celu zapewnienia podwyższonego poziomu bezpieczeństwa – odporności na losowe awarie. Do obu rdzeni trafiają te same dane i te same sygnały kontrolne. Informacje wyjściowe z układu są następnie porównywane ze sobą przez moduł CCM. Aby zredukować szansę, że dane zdarzenie losowe (np. promieniowanie elektromagnetyczne, czy kosmiczne) wywoła usterkę w obu rdzeniach w tym samym momencie, są one od siebie izolowane przestrzennie (rdzenie są odsunięte od siebie w układzie o ponad 100 mikronów, a ponadto drugi rdzeń

jest obrócony o 90° i odbity lustrzanie), elektrycznie (wokół każdego rdzenia rozciągnięty jest ochronny pierścień metalizacji) i czasowo (rdzenie działają asynchronicznie – ich działania przesunięte są względem siebie w czasie o 1,5 lub 2 cykle zegarowe, zależnie od implementacji). Mikrokontrolery Infineon Aurix wykorzystują podobną technologię – rdzenie, wykorzystywane w tych układach działają również w architekturze 1oo1D, jednakże w pojedynczym mikrokontrolerze Aurix znajdować się może do trzech rdzeni (z których dwa są podwojone i pracują jako 1oo1D). Oznacza to, że istnieje możliwość skonfigurowania ich do pracy w topologii 2oo2D lub 2oo3, zapewniając zwiększony poziom odporności na awarię i redundancji (HFT > 0).

Analiza zagrożeń i ocena ryzyka

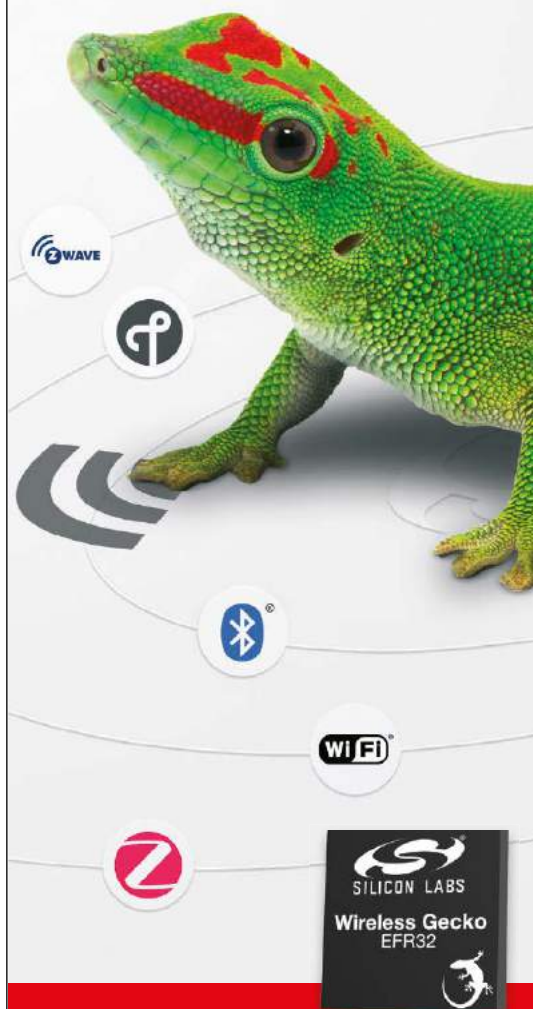
Elementy używane w produkcji końcowym mogą podlegać systematycznym lub przypadkowym awariom. Normy bezpieczeństwa funkcjonalnego zostały opracowane tak, aby powodować zmniejszenie potencjalnego ryzyka obrażeń fizycznych i szkód materialnych z powodu tego rodzaju awarii. Najpierw jednak należy zidentyfikować zagrożenia i przeanalizować ich skutki, aby można było wdrożyć odpowiednie metody minimalizacji prawdopodobieństwa zidentyfikowanych awarii.

We wcześniejszej części tego artykułu opisaliśmy awarie przypadkowe i metody zarządzania nimi. Konieczne jest teraz zbadanie zagrożeń i ocena ryzyka na poziomie gotowego urządzenia, wykorzystującego programowalne elementy, aby określić wymogi zmniejszenia ryzyka zgodnie z wybraną klasą ASIL/SIL. Gdy znane są te wymogi, programiści będą musieli wdrożyć wystarczającą liczbę mechanizmów diagnostycznych, aby wykrywać awarie i osiągnąć



Rysunek 3. Koncepcja bezpieczeństwa dwurdzeniowego 1oo1D, wykorzystana w układach z rodziny TI Hercules (źródło: <http://bit.ly/2kDTMug>)

Wirtuoz IoT



Wireless Gecko – SoC's i moduły

Najwyższa efektywność energetyczna dla ARM® Cortex®-M4 dzięki trybom Ultra Low Power:

- ▶ Bluetooth Low Energy, Bluetooth 5
- ▶ Zigbee, Thread, Bluetooth Mesh
- ▶ WiFi
- ▶ Z-Wave
- ▶ Multiprotocol Support

Po szczegóły zapraszamy na

www.glyn.pl

biuro@glyn.pl



GLYN
High-Tech Distribution

Tabela 4. Zarządzanie awariami w/g norm bezpieczeństwa w urządzeniu końcowym

Bezpieczeństwo funkcjonalne	Analiza	Przykład
Definicja systemu	Jaka jest funkcja urządzenia?	Sterownik silnika elektrycznego w pojeździe elektrycznym
Analiza zagrożeń i ocena ryzyka	Zidentyfikuj jakie są zagrożenia, skategoryzuj występujące w systemie ryzyka.	Zbyt wysoki moment obrotowy silnika, powodujący ryzyko kolizji.
Klasyfikacja wymaganego stopnia ASIL/SIL	Jakie jest maksymalne tolerowalne ryzyko w systemie?	ASIL-C
Wyznaczenie celów bezpieczeństwa	Wymagania bezpieczeństwa	Unikaj zbyt wysokiego momentu silnika
Implementacja wymaganych rozwiązań	Tryby awarii i jej prawdopodobieństwo + Implementacja diagnostyki	Implementacja diagnostyki w mikrokontrolerze do monitorowania działania generatora PWM
Obliczenie metryk bezpieczeństwa funkcjonalnego	Czy redukcja ryzyka jest dostateczna?	Wyznaczenie metryk.

wystarczającą redukcję ryzyka. Mierzy się ją wskaźnikami opisanymi w obowiązującej dany produkt normie.

W tabeli 4 zaprezentowano schematyczne przedstawienie postępowania, wraz z przykładowym zadaniem – sterownikiem silnika elektrycznego, zgodnego z normą ISO 26262. Podobne metody będą miały zastosowanie również w przypadku innych norm bezpieczeństwa funkcjonalnego. W przykładzie tym mamy do czynienia ze sterownikiem silnika w pojeździe elektrycznym. Jego głównym zadaniem jest kontrola momentu obrotowego silnika za pomocą przebiegu PWM. Moduł taki ma bardzo kompleksowe tryby awarii. Jednym z nich jest wygenerowanie zbyt dużego momentu obrotowego na silniku, co przełożyć może się na wywołanie wypadku komunikacyjnego – uderzenia w barierkę na autostradzie, zjechanie do rowu etc. Rozważmy ten przypadek dokładniej, gdyż jest on bardzo poważnym zagrożeniem dla zdrowia i życia kierowcy oraz pasażerów.

Zgodnie z normą ISO 26262, analiza zagrożeń opiera się na ocenie każdej z awarii w trzech kategoriach – poważności (S1...S3), prawdopodobieństwa niebezpiecznego wydarzenia w czasie awarii (E1...E4) oraz sterowalności sytuacją w danej awarii (C1...C3). W opisanym powyżej przypadku awaria klasyfikowana jest jako S3, ponieważ zagraża życiu; E4, ponieważ istnieje duże ryzyko, że auto w czasie awarii poruszać będzie się po autostradzie, drodze sąsiadującej z rowem melioracyjnym itp. oraz jako C2, ponieważ pomimo tego rodzaju awarii, autem można nadal kierować i hamować. Na podstawie tych klasyfikacji z normy można odczytać, iż system bezpieczeństwa funkcjonalnego musi mieć klasyfikację co najmniej ASIL-C, aby ryzyko było na tolerowanym poziomie. W tabeli 2 odczytać możemy towarzyszące tej klasie bezpieczeństwa metryki bezpieczeństwa – SPFM, PMHF (FIT) oraz LFM, które omawialiśmy we wcześniejszej części tekstu.

W tym przykładzie założymy, że momentem obrotowym silnika steruje inny komputer w pojeździe za pośrednictwem sieci CAN. Mikrokontroler steruje momentem obrotowym silnika przebiegiem PWM. Konieczne jest zatem zrozumienie w jaki sposób w układzie generowane są sygnały PWM i jakie tryby awarii mogą wpłynąć na zaangażowane peryferia. Wiedząc to można obliczyć wskaźnik awaryjności (jeśli producent dostarcza odpowiednią dokumentację) i, jeśli wskaźnik awaryjności jest zbyt wysoki w stosunku do wymagań ASIL-C, zastosować dodatkową diagnostykę w oprogramowaniu, w celu redukcji prawdopodobieństwa niewykrytej awarii systemu.

W takim systemie pozycja silnika jest określana poprzez enkoder kwadraturowy i pomiar prądu fazowego na mostku sterującym silnikiem. Procesor odczytuje swój program z pamięci Flash, a w pamięci SRAM wykonuje się algorytm FOC (*Field Oriented Control*). Algorytm ten ma na celu ustalenia odpowiednich sygnałów PWM do sterowania silnikiem z pożądanym momentem obrotowym. Wbudowany zegar PWM generuje następnie sygnały napędzające sterownik mostka. Ten prosty opis jest dostateczny, do przeprowadzenia analizy ryzyka w układzie. Oto przykładowa lista awarii mikrokontrolera, które mogą

spowodować wygenerowanie nieprawidłowego sygnału PWM. Należy pamiętać, że nie jest to w żadnym wypadku wyczerpujące i jest przeznaczone wyłącznie jako przykład w celach ilustracyjnych.

- Awaria rdzenia procesora
- Przerzucanie bitów pamięci Flash i/lub SRAM
- Brak zasilania
- Awaria zegara
- Błąd komunikacji CAN
- Awaria modułu (timera) PWM

Nieprawidłowe działanie któregośkolwiek z tych modułów, może potencjalnie spowodować wystąpienie niebezpiecznego zdarzenia. Aby wdrożyć funkcję bezpieczeństwa polegającą na „unikaniu zbyt wysokiego dodatniego momentu obrotowego silnika”, projektant musiałby zastosować diagnostykę do elementów o krytycznym znaczeniu dla bezpieczeństwa, tak aby w przypadku wykrycia awarii system mógł podjąć działania w celu złagodzenia awarii lub przejścia do zdefiniowanego, bezpiecznego stanu.

Informacje na temat dedykowanych systemów diagnostycznych, zaimplementowanych w danym mikrokontrolerze, projektanci systemu znaleźć mogą w dokumentacji, o ile dany układ dostosowany jest do „zastosowań specjalnych” w systemach bezpieczeństwa funkcjonalnego. Dokumentacja ta, wraz z narzędziami FMEDA (do systematycznej analizy trybów i efektów awarii oraz badania systemów diagnostycznych), pozwala na oszacowania częstości FIT układu, jak i wyznaczenie wskaźników bezpieczeństwa (SPFM, LFM, PMHF). Tego rodzaju dokumentacja dostępna jest tak dla normy ISO 26262, jak i dla IEC 61508 w precertyfikowanych mikrokontrolerach.

Podsumowanie

Systemy bezpieczeństwa funkcjonalnego, to układy elektroniczne, często wykorzystujące mikrokontrolery do zapewnienia bezpieczeństwa ludziom, pracującym w przemyśle, kierowcom i pasażerom samochodów, pociągów itp. Każdy z tych sektorów opisany jest szeregiem norm, definiujących tolerowane ryzyko w tych systemach, jak i sposoby jego badania i oceny.

Jednym ze sposobów na uproszczenie procesu projektowania systemu bezpieczeństwa funkcjonalnego, jest wykorzystanie precertyfikowanych układów, dedykowanych do tego rodzaju specjalnych zadań. Są to często mikrokontrolery specjalnie zoptymalizowane do takich zastosowań. Przykładami takich układów są m.in. mikrokontrolery z rodziny Hercules firmy Texas Instruments czy Aurix firmy Infineon.

Nikodem Czechowski

Źródła:

- <https://ubm.io/2kO6h6o>
- <https://ubm.io/2ISlxPW>
- <http://bit.ly/2ITfpXy>
- <http://bit.ly/2kDTMug>
- Norma PN-EN 61508
- Norma ISO 26262
- <http://bit.ly/2IUmk2G>

Faulhaber rozszerza grę silników krokowych

Ogromne przyspieszenie przy wykorzystaniu magnesu dyskowego

Więcej informacji:

Dr. Fritz Faulhaber GmbH & Co. KG
Kristina Rebmann – Marketing
Daimlerstraße 23/25, 71101 Schönaich, Niemcy
tel. +49 7031 638-148, faks +49 7031 638-8148
www.faulhaber.com, kristina.rebmann@faulhaber.de



Dzięki akwizycji szwajcarskiego producenta wysokowydajnych silników krokowych, firmy Dimatech, Faulhaber poszerzył swoją ofertę o silniki o wysokiej mocy i zwiększonej dynamice. Podstawową różnicą pomiędzy silnikami z magnesem dyskowym a konwencjonalnymi jest bardzo niska masa wirnika silnika z dyskiem magnetycznym. Składa się on z wielobiegunowego magnesu o kształcie dysku, zamontowanego na wirniku silnika. Jego mała masa minimalizuje bezwładność obrotową i pozwala na osiągnięcie wartości przyspieszenia, niedostępnych przy użyciu innych rodzajów konstrukcji o porównywalnych wymiarach. Dzięki temu silniki z magnesem dyskowym doskonale nadają się do zastosowań, gdzie wymagane są częste i szybkie zmiany prędkości lub kierunku obrotów. Konstrukcja silnika umożliwia ponadto wykonywanie bardzo małych i precyzyjnych ruchów krokowych. Dodatkowo wielkość biegunów magnesów oraz kształt pola magnetycznego zostały dobrane tak, aby zapewnić możliwość maksymalnego momentu obrotowego przy uwzględnieniu wymiarów wirnika.

Specjalna charakterystyka tych silników krokowych predestynuje je między innymi do zastosowań w automatyzacji, wymagających bardzo szybkiego poruszania niewielkich obciążeń. Takie warunki często występują u producentów półprzewodników, w przemyśle technicznym albo w technice medycznej czy robotyce. Istotną zaletą jest także niska masa silników. Ma to szczególne znaczenie w przypadku, gdy silnik porusza się razem z ładunkiem.

Przegląd zalet:

- bardzo wysokie przyspieszenie, bardzo duża zdolność do zmiany kierunku
- duża gęstość mocy
- długa żywotność
- krótkie i lekkie silniki
- wysoka prędkość wciągania
- możliwość pracy pełnokrokowej, półkrokowej i mikrokrokowej
- wysoka niezawodność



Silnik krokowy z magnesem dyskowym nadaje się do zastosowań, np. w przemyśle włókienniczym

FAULHABER Drive Systems

DNA aplikacji następnej generacji



Wizje nie znają granic

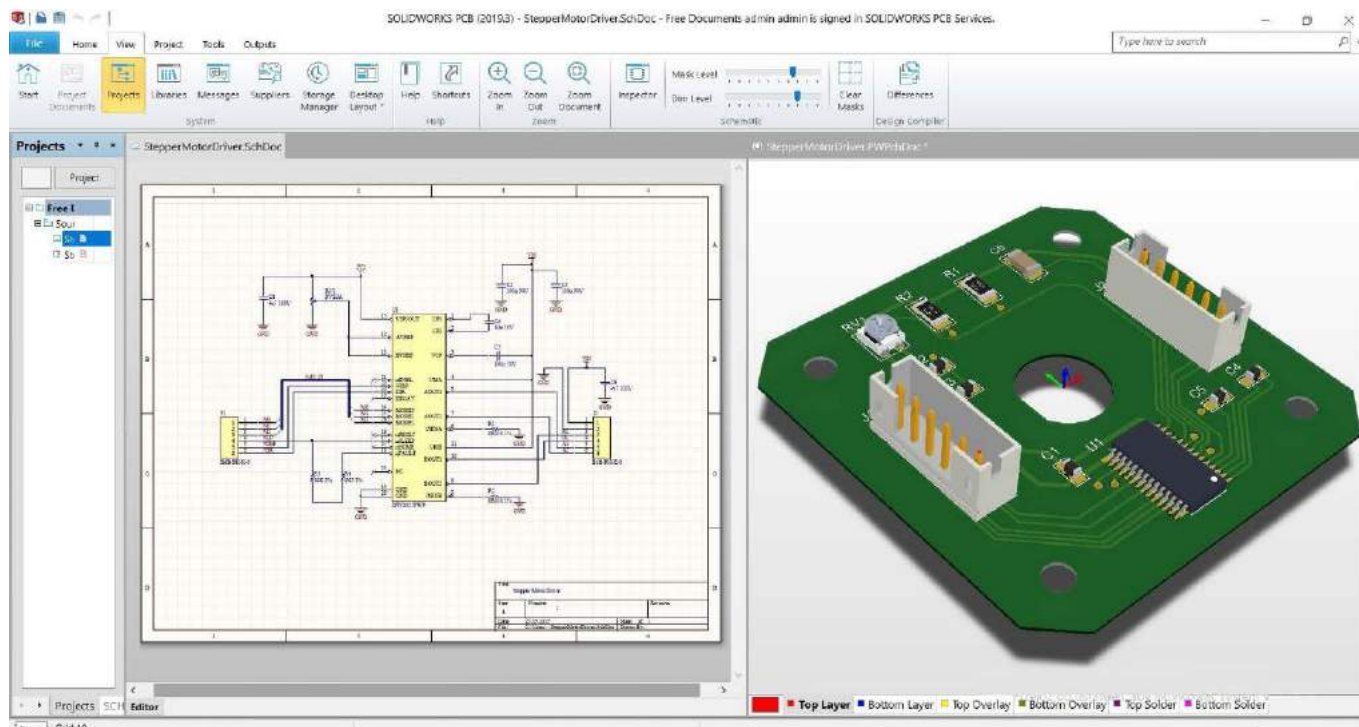
Jesteśmy twórcami kluczowych technologii w zakresie konstrukcji wydajnych miniaturowych silników elektrycznych. Produkowane dzisiaj silniki są tak małe, że mogą docierać przez naczynia krwionośne do serca i chronić życie. To nasze podzespoły dla wszystkich, których wizją jest sięgnięcie po gwiazdy.

Więcej informacji:

www.faulhaber.com
FAULHABER Polska sp. z o.o.
info@faulhaber.pl



WE CREATE MOTION



Rysunek 1. Program SOLIDWORKS PCB – widoczny jest jednocześnie schemat i trójwymiarowa płytką drukowaną

SOLIDWORKS PCB

– unikatowy mariaż narzędzi MCAD i EDA

Firma Dassault Systèmes SOLIDWORKS Corporation, we współpracy z Altium Limited przygotowała nowe narzędzie, dedykowane do projektowania obwodów elektronicznych. Umożliwia ono współpracę w zakresie projektowania mechanicznego i elektronicznego. Dzięki połączeniu tych dwóch sprawdzonych rozwiązań powstało unikatowe narzędzie integrujące w sobie projektowanie od strony mechaniki jak i elektroniki.

Altium Limited to producent oprogramowania EDA (automatyzacja projektowania elektroniki), czyli służącego do projektowania urządzeń elektronicznych – najczęściej do rysowania schematów ideowych i projektowania płytek drukowanych. Historia firmy, znanej kiedyś jako Protel, sięga lat '80 XX wieku. Altium Designer jest obecnie wiodącym na rynku profesjonalnym narzędziem do projektowania płytek drukowanych. Nie ma się więc co dziwić, że Dassault Systèmes wybrało tę firmę do współpracy w przygotowaniu swojego najnowszego narzędzia – SOLIDWORKS PCB.

Sprawdzona technologia EDA w zupełnie nowej odsłonie

Dassault Systèmes skorzystało z ponad 25-letniego doświadczenia projektantów Altiuma do opracowania przyjaznego i intuicyjnego interfejsu użytkownika w swoim nowym pakiecie. Dzięki zastosowaniu interfejsu znanego z pakietów EDA firmy Altium, migracja do nowego oprogramowania jest niezwykle prosta, dla użytkowników, którzy kiedykolwiek mieli z nimi do czynienia.

Więcej informacji:

Dodatkowe informacje na temat oprogramowania SOLIDWORKS, w tym pakietu SOLIDWORKS PCB uzyskać można na stronie oficjalnego dystrybutora, firmy DPS Software: www.dps-software.pl

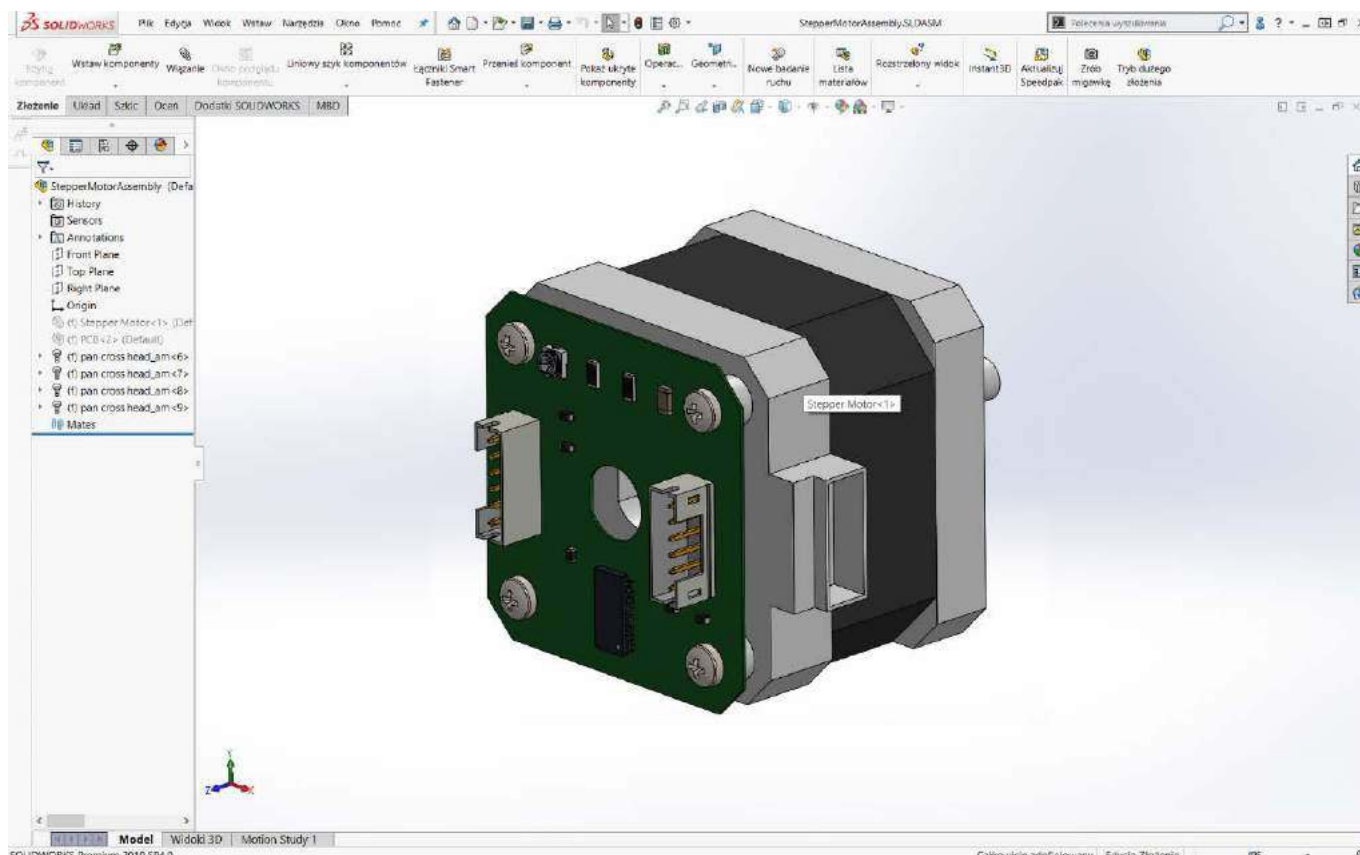


Na **rysunku 1** zaprezentowano widok interfejsu programu SOLIDWORKS PCB.

Nowoczesne środowisko edycji projektu składa się z edytora schematów i płytek drukowanych. Dodatkowo dostępne jest narzędzie do przygotowywania rozbudowanych bibliotek, jakie znamy z programów Altiuma. W systemie zaimplementowano również wiele innych rozwiązań, znanych z tego pakietu, chociażby algorytmy interaktywnego prowadzenia ścieżek czy rozbudowane, konfigurowalne reguły dla projektu płytki drukowanej.

Integracja oprogramowania ECAD i MCAD w jednym pakiecie

Oprogramowanie SOLIDWORKS PCB pozwala na współpracę projektu części elektronicznej i mechanicznej projektowanego urządzenia. Oferuje ono wyraźną korzyść, w sytuacji gdy współpraca elektronika i mechanika ma kluczowe znaczenie dla osiągnięcia założonych celów projektowych systemu. Bezpośrednia współpraca środowiska projektowania elektronicznego i mechanicznego umożliwia natywną wymianę danych projektowych za pomocą wygodnego interfejsu, zapewnia spójność projektów i dokładność odwzorowania elektroniki w projekcie mechanicznym i vice versa.



Rysunek 2. Oprogramowanie SOLIDWORKS 3D CAD z wczytaną płytką drukowaną i elementami mechanicznymi

Integracja środowisk projektowych ułatwia integrację projektów, a co za tym idzie poprawia możliwości integracji elektroniki w podzespołach mechanicznych. W dobie miniaturyzacji urządzeń elektronicznych, ścisła integracja projektów mechanicznych i elektronicznych jest kluczowa, dla uzyskania kompaktowego systemu. Jest to szczególnie istotne w przypadku urządzeń elektroniki noszonej, gdzie wykorzystanie każdego milimetra sześciennego obudowy może przesądzać o sukcesie projektu.

Zastosowanie zintegrowanych środowisk EDA i CAD zwiększa również produktywność i skraca czas pracy designerów. Jednocześnie podejście to minimalizuje liczbę popełnianych w czasie projektu błędów, które wykryte zostaną dopiero na etapie prototypowania urządzenia. Dzięki temu, że projekt mechaniczny ma bezpośredni dostęp do danych EDA i vice versa, możliwa jest szybka weryfikacja ewentualnych konfliktów poszczególnych podzespołów w całym systemie.

SOLIDWORKS PCB

W ramach nowego pakietu Dassault Systèmes opracował dwa pakiety oprogramowania. Pierwszym z nich jest SOLIDWORKS PCB. Jest to zestaw narzędzi dedykowanych do rysowania schematów ideowych i projektowania płytek drukowanych. Dzięki temu, że system oparto na sprawdzonym silniku Altiuma, jest on intuicyjny i wydajny podczas projektowaniu płytek drukowanych. Dodatkowo, dzięki wbudowanemu symulatorowi Spice można analizować działanie obwodów z poziomu edytora schematów.

Interfejs programu opracowany został tak, aby był kompatybilny wizualnie zarówno z narzędziami SOLIDWORKS, jak i z oprogramowaniem EDA obecnym na rynku. Oprócz typowych narzędzi do projektowania PCB, ma także nowoczesne rozwiązania do implementowania zaawansowanych i rozbudowanych bibliotek, dodatkowych reguł projektowych, jak i szeroko rozbudowane możliwości integracji z projektami mechanicznymi. Dzięki silnikowi projektowania, który pracuje we w pełni trójwymiarowym środowisku

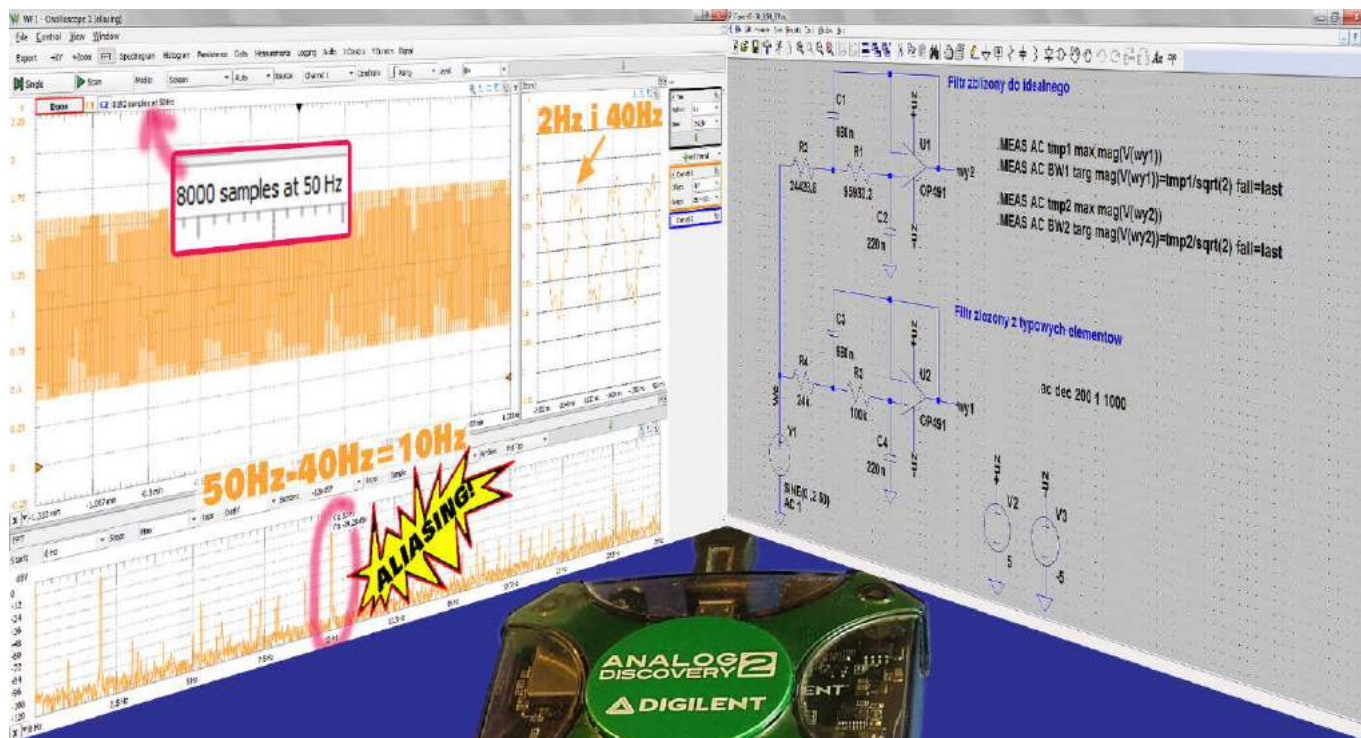
w czasie rzeczywistym, można monitorować występowanie kolizji elementów na płycie drukowanej i otaczających ją elementów obudowy.

Integracja SOLIDWORKS PCB z oprogramowaniem MCAD nie ogranicza się tylko do możliwości przetrzymywania bryły płytki drukowanej do projektu mechanicznego. Zaawansowane narzędzia do zarządzania procesem zlecania zmian konstrukcyjnych pozwalają na przenoszenie zmian konstrukcyjnych pomiędzy SOLIDWORKS PCB i SOLIDWORKS 3D CAD. Obejmuje to zmiany projektowe, które dotyczą kształtu płytki, rozmieszczenia komponentów, otworów montażowych i wycięć, co zapewnia prawidłową synchronizację projektów.

SOLIDWORKS PCB Connector

Drugim nowym narzędziem jest SOLIDWORKS PCB Connector. Jest to nowoczesne środowisko współpracy pomiędzy użytkownikami Altium Designer i SOLIDWORKS 3D CAD, eliminujące błędy wynikające z braku synchronizacji pomiędzy częścią elektryczną a mechaniczną projektu. Narzędzie to w prosty sposób łączy dane z obu środowisk, udostępniając kluczowe elementy projektowe zespołom projektantów elektroniki i mechaniki. Dzięki SOLIDWORKS PCB Connector, w trakcie całego procesu projektowania, każdy użytkownik dysponuje spójnymi informacjami, pomagającymi dotrzymać terminu wprowadzenia produktu na rynek i zredukować wysokie koszty związane z wprowadzaniem poprawek.

SOLIDWORKS PCB Connector pozwala na zintegrowanie zarządzania procesem wprowadzania zmian projektowych tak w projekcie mechanicznym jak i elektronicznym. Dzięki równoczesnemu śledzeniu zmian przez oba środowiska zespół projektowy może pracować szybciej i wydajniej. Dodatkowe narzędzia umożliwiające wprowadzanie komentarzy i zarządzanie zmianami w projekcie w zunifikowany sposób pozwalają na zachowanie porządku w repozytorium projektu.



Symulacja i pomiar czyli LTspice i Analog Discovery 2 w rękach konstruktora (3)

Filtr antyaliasingowy

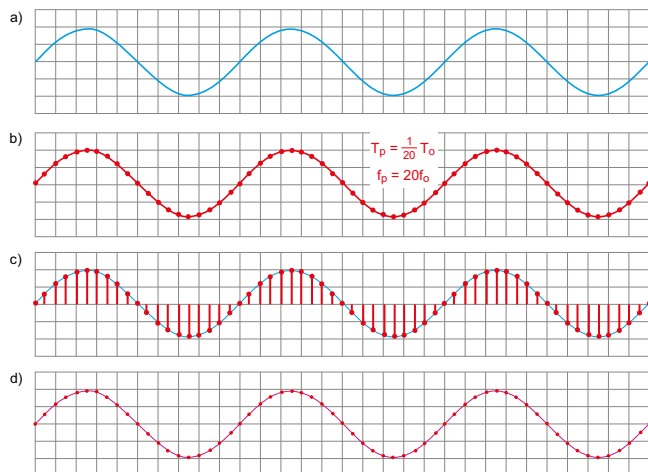
W tym odcinku zajmiemy się badaniem filtra dolnoprzepustowego, a pretekstem będą zagadnienia związane z analogowo-cyfrowym przetwarzaniem sygnałów. Filtr zaprojektujemy samodzielnie dwoma metodami, a następnie sprawdzimy obliczenia w symulatorze LTspice i przeprowadzimy pomiary układu rzeczywistego za pomocą Analog Discovery 2.

Wiele razy zastanawiałem się, co mogło być zalążkiem współczesnej techniki cyfrowej. Czy mógł nim być np. telegraf? Gdyby przyjąć takie założenie okazałoby się, że technika cyfrowa powstała przed analogową. Nie kto inny bowiem, jak znany skądinąd Charles Wheatstone skonstruował pierwszy telegraf elektryczny w roku 1837, podczas gdy pierwsze lampy analogowe powstawały dopiero na początku XX wieku – dioda w roku 1904, trioda w 1906. Na prawdziwy rozwój elektroniki trzeba było jednak jeszcze poczekać ok. 10 lat, aż do skonstruowania pompy dyfuzyjnej, która umożliwiała masową produkcję lamp o akceptowalnych parametrach.

Zasada działania telegrafu Wheatstone'a polegała na wskazywaniu kolejnych liter alfabetu przesyłanego tekstu za pomocą specjalnej wskazówki przyjmującej tylko pewne dyskretne położenia, trudno chyba jednak klasyfikować takie urządzenie jako cyfrowe. W tym samym roku Samuel Finley Breese Morse skonstruował telegraf, w którym były przesyłane tylko dwa znaki: kropka i kreska. Nie można się nie zgodzić, że to już znacznie bliżej techniki cyfrowej w dzisiejszym rozumieniu. To chyba za daleko idący wniosek, chociaż do telegrafii jeszcze zapewne wrócimy. Mimo pewnych wczesnych faktów, o których można mówić, że miały jakiś związek z techniką cyfrową, ta mogła się rozwijać dopiero po wynalezieniu tranzystora, a nawet jeszcze później po wyprodukowaniu układu scalonego.

Niemal równocześnie z rozpoczęciem produkcji mikroprocesorów (rok 1969) zapoczątkowano prace nad wykorzystywaniem ich w urządzeniach, w których występują sygnały analogowe. O ile realizacja funkcji sterujących była dla mikroprocesora naturalna, to obróbka sygnału analogowego przez mikroprocesor wymagała użycia dodatkowego elementu – przetwornika analogowo-cyfrowego lub





Rysunek 1. Ilustracja zasady przetwarzania analogowo-cyfrowego a) przebieg oryginalny, b) momenty próbkowania taktowane generatorem, c) kwantyzacja, d) odtworzenie sygnału oryginalnego na podstawie próbek metodą interpolacji liniowej

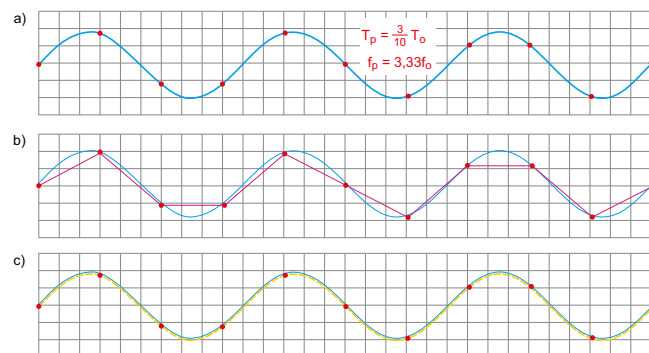
cyfrowo-analogowego. Ale jak mówi przysłowie: potrzeba jest matką wynalazków, więc stosowne elementy wkrótce opracowano i można było przystąpić do realizacji projektów.

Próbkowanie i kwantyzacja sygnału analogowego

Sprawa wydaje się dość prosta. Nasze zmysły nie są idealne i stosunkowo łatwo je oszukać. Oszukujemy tak wzrok wyświetlając statyczne klatki. Wystarczy tylko robić to dostatecznie szybko. W kamerach amatorskich było to 16 klatek na sekundę, ale mimo wrażenia płynności ruchu migotanie klatek było jednak zauważalne. W kameach profesjonalnych podwyższono więc tę szybkość do 24 klatek na sekundę i przy tej szybkości chyba już nikt nie narzeka na migotania. Można więc przypuszczać, że podobna zasada będzie obowiązywała też w ogólnym przypadku.

Weźmy więc jakiś przykładowy przebieg. Zwykle do wszelkiego rodzaju analiz używamy przebiegu sinusoidalnego. Jest on dość wygodny do opisu matematycznego, a jak wiemy z teorii opracowanej już na początku XIX wieku przez Jeana Baptiste Josepha Fouriera, z przebiegów takich można składać w zasadzie każdy dowolny przebieg o złożonym kształcie. Mamy zatem sinusoidę, którą będziemy przetwarzać w przetworniku A/C (**rysunek 1a**). Niezbędny będzie do tego jakiś generator wyznaczający momenty pobierania próbek sygnału (**rysunek 1b**). W systemie mikroprocesorowym realizacja takiego generatora jest bajecznie prosta. Intuicyjnie wyczuwany, że im częściej będziemy próbować nasz sygnał, z tym większą dokładnością będzie on odwzorowany w postaci cyfrowej. W momentach wyznaczonych przez generator konieczne jest zmierzenie wartości chwilowej sygnału, czyli kwantyzacja (**rysunek 1c**). Jak wynika z nazwy parametru, pomiar musi być dokonany w zerowym czasie tak, aby w trakcie pomiaru sygnał się nie zmienił. Jest to oczywiście wyzwanie techniczne wymagające szerszego omówienia, ale w tym artykule niestety nie ma na to miejsca.

Przetwarzanie analogowo-cyfrowe dostarczyło nam ciąg danych liczbowych reprezentujących chwilowe wartości sygnału analogowego. Przed nami otwiera się szerokie pole do wykorzystania tych danych. Załóżmy, że mamy do czynienia z sygnałem akustycznym, który chcemy cyfrowo pogłodzić lub przyciszyć. Wystarczy więc każdą próbkę pomnożyć przez odpowiedni współczynnik większy lub mniejszy od jedności, a następnie tak zmodyfikowane dane cyfrowe poddać procesowi odwrotnemu, czyli przetworzeniu cyfrowo-analogowemu (**rysunek 1d**). Ten sam generator wyznacza teraz momenty, w których sąsiednie próbki będą ze sobą łączone. Przy tak gęstym próbkowaniu jak na **rysunku 1b** nawet nie zauważymy,

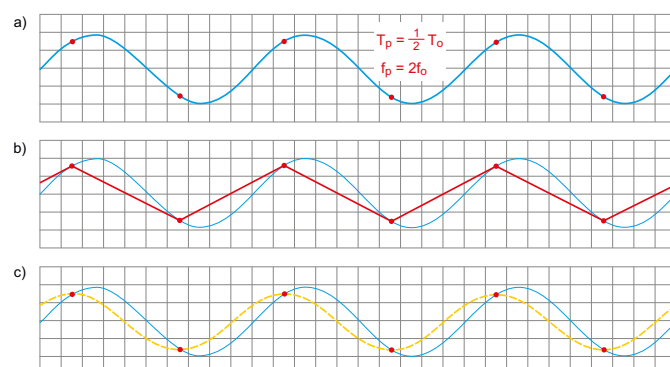


Rysunek 2. Badanie reakcji na obniżanie częstotliwości próbkowania a) ok. 3 próbek na okres, b) sygnał odtworzony metodą interpolacji liniowej, c) sygnał odtworzony metodą interpolacji nieliniowej

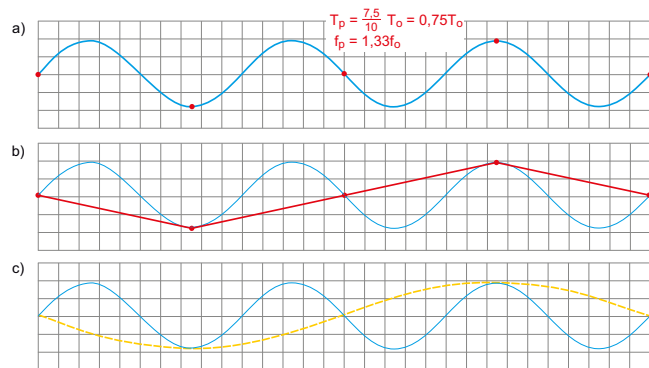
że odtworzony z przebiegu cyfrowego przebieg analogowy będzie się składał z krótkich odcinków, nie krzywizn (**rysunek 1d**).

Jeśli przyjmujemy, że wszystkimi powyższymi operacjami obarczymy mikroprocesor, to może się okazać, że nie zostanie mu za wiele czasu na inne operacje. Warto więc pomyśleć o spowolnieniu próbkowania. Zobaczmy czym to może grozić. Na **rysunku 2a** przedstawiono sygnał sinusoidalny, w którym na jeden okres przypada ok. 4 próbki. Stosując tę samą metodę odtwarzania sygnału analogowego z cyfrowego uzyskamy przebieg widocznie różniący się od oryginału, ale zachowujący jego częstotliwość podstawową. Z punktu widzenia analizy częstotliwościowej jest to ważny wniosek. Zastosowana na **rysunku 2b** metoda odtworzenia sygnału analogowego nazywa się interpolacją liniową. Polega ona na bezpośrednim łączeniu kolejnych próbek odcinkami. Jest to więc metoda najprostsza, choć jak widać nie pozbawiona wad. Stosując inne rodzaje interpolacji, np. $\sin(x)/x$ można uzyskać wynik dużo bardziej przypominający oryginał (**rysunek 2c**).

Poszło nieźle. Natychmiast rodzi się pytanie, jak daleko można posunąć się ze spowalnianiem próbkowania? Sinusoida ma dwa półokresy, zobaczmy więc co będzie się działo jeśli zastosujemy próbkowanie z szybkością dwóch próbek na okres? Przypadek ten przedstawiono na **rysunku 3a** (trzecia próbka na tym wykresie należy już do kolejnego okresu). Odtworzony z zastosowaniem interpolacji liniowej przebieg już zupełnie zmienił kształt. Teraz jest to już przebieg trójkątny, ale nadal zachowuje on oryginalną częstotliwość podstawową. Stosując jakąś interpolację nieliniową nadal można osiągnąć przebieg sinusoidalny. Warto jednak zauważyć, że amplituda przebiegu zależy od fazy próbkowania. Jeśli próbki trafiają na ekstrema sinusoidy przebieg odtwarzany będzie największy, natomiast całkowita tragedia nastąpi, gdy próbki trafią na zera sinusoidy. Wtedy zostanie odtworzona... linia ciągła na poziomie zero.



Rysunek 3. Badanie reakcji na obniżanie częstotliwości próbkowania a) 2 próbki na okres, b) sygnał odtworzony metodą interpolacji liniowej, c) sygnał odtworzony metodą interpolacji nieliniowej



Rysunek 4. Aliasing a) mniej niż 2 próbki na okres, b) sygnał odtworzony metodą interpolacji liniowej, c) sygnał odtworzony metodą interpolacji nieliniowej

No dobrze, idźmy dalej. Spowalniamy próbkowanie do szybkości mniejszej niż 2 próbki na okres (**rysunek 4a**). I w tym przypadku po odtworzeniu przebiegu analogowego metodą interpolacji liniowej uzyskujemy przebieg trójkątny, a interpolacja nieliniowa daje nawet całkiem kształtną sinusoidę, tylko... to zupełnie nowy przebieg. Częstotliwość odtworzonej sinusoidy jest zupełnie inna od oryginalnej. Gdybyśmy nie byli świadomi, że złamaliśmy jakąś zasadę moglibyśmy przypuszczać, że mamy do czynienia z zupełnie innym jakościowo sygnałem. Taki wynik jest absolutnie nie do przyjęcia.

Tak źle i tak niedobrze, więc właściwie po co walczyć z obniżeniem częstotliwości próbkowania? Powodów jest wiele. Bo im mniejsza jest częstotliwość próbkowania, tym mniej danych opisujących sygnał analogowy, a więc łatwiej będzie te dane ewentualnie przesyłać i gromadzić w pamięci. Bo uwolnimy procesor od nadmiaru operacji, i wreszcie bo łatwiej i taniej będzie wyprodukować przetworniki A/C i C/A, prawdopodobnie też będzie można uprościć połączenia na PCB. Nie lekceważyłbym też problemu emisji zakłóceń elektromagnetycznych, które rosną wraz z częstotliwością przebiegu występujących w układzie.

Telegraf a sprawa próbkowania

Na długo przed pojawieniem się urządzeń cyfrowych w laboratorium Bell'a pracował pewien Szwed – Harry Theodor Nyquist znany nam jako Nyquist, który w wieku 18 lat wyemigrował wraz z rodziną do Stanów Zjednoczonych. Może to właśnie dzięki temu zawdzięczamy jedno z fundamentalnych twierdzeń, bez którego trudno by było wyobrazić sobie współczesną elektronikę. Nyquist zajmował się telegrafami, badał m.in. przepustowość kanałów telegraficznych. W wyniku prowadzonych eksperymentów opublikował dwie prace [1] i [2] opisujące ograniczenia dotyczące przesyłania sygnałów cyfrowych w liniach transmisyjnych. Przez kilka lat współpracował w Laboratorium Bella z Claudem Elwoodem Shannonem, świetnym matematykiem i inżynierem również zajmującym się m.in. telegrafią. Dzisiaj każdy elektronik wybudzony w środku nocy potrafi z zamkniętymi oczami zacytować twierdzenie o próbkowaniu, tak jak prawo Ohma. Twierdzenie to jest wyprowadzane w różnych miejscach bardziej lub mniej zrozumiałym językiem techniczno-naukowym, ale istota jest zawsze ta sama i względnie prosta: **sygnał analogowy poddany próbkowaniu może być prawidłowo odtworzony z próbek wtedy i tylko wtedy, gdy częstotliwość próbkowania jest co najmniej 2-krotnie wyższa od najwyższej częstotliwości występującej w widmie tego sygnału**. Twierdzenie to jest znane pod kilkoma nazwami: Nyquista-Shannona, Nyquista-Kotelnikowa-Shannona, w wariantach z różną kolejnością pojawiających się nazwisk. Trudno dziś rozstrzygnąć, komu należy się palma pierwszeństwa, ale w gruncie rzeczy nie o to chodzi. O Władimirze Aleksandrowiczu Kotelnikowie warto jednak wspomnieć, gdyż i on zajmował się w swojej niezwykle bogatej działalności naukowej zagadnieniami związanymi z telegrafią.

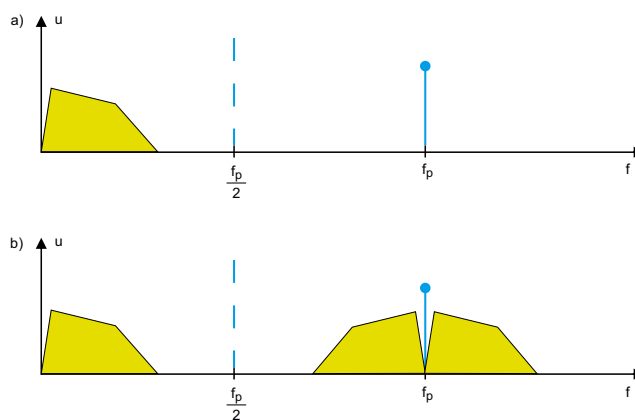


Fotografia 1. Naukowcy, którzy położyli podwaliny współczesnym technikom przetwarzania sygnałów a) Harry Theodor Nyquist, b) Claude Elwood Shannon, c) Władimir Aleksandrowicz Kotelnikow

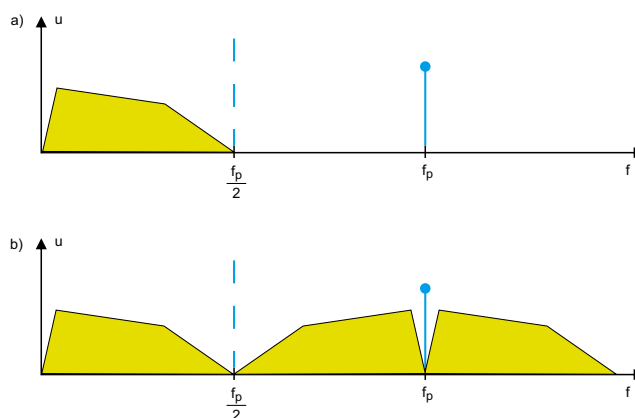
Zasługi Kotelnikowa zostały zresztą wynagrodzone szeregiem odznaczeń państwowych, w tym aż 5-krotnie Orderem Lenina. Przed przystąpieniem do eksperymentów warto spojrzeć na tych panów (**fotografia 1**). Pewnie gdyby nie oni znaleźliby się inni, ale szacunek należy się właśnie im. W czasach, w których publikowali fundamentalne prawa dotyczące próbkowania nawet nie znali jeszcze słowa „przetwornik”.

O próbkowaniu w ujęciu częstotliwościowym

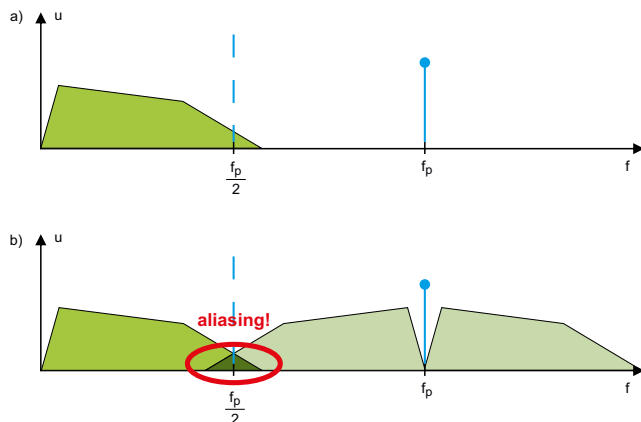
Wróćmy jeszcze na chwilę do rozważań o wyborze prawidłowej częstotliwości próbkowania. Na **rysunku 5a** przedstawiono widmo jakiegoś sygnału poddanego próbkowaniu. Jak widać pasmo tego sygnału jest mniejsze od połowy częstotliwości próbkowania, co zgodnie z twierdzeniem o próbkowaniu jest rozwiązaniem prawidłowym i nie powinno powodować żadnych problemów. W widmie sygnału próbkowanego pojawiają się (między innymi) dwa listki symetryczne względem częstotliwości próbkowania (**rysunek 5b**). Nie można o tym zapominać, ale nie powodują one żadnych zakłóceń i zafałszowań



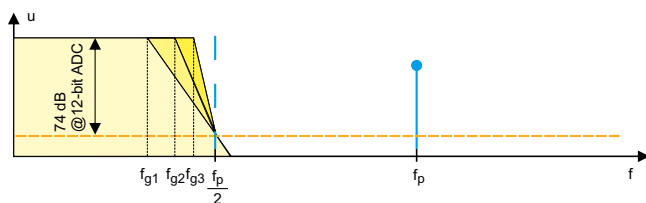
Rysunek 5. Widmo sygnału próbkowanego mieszczącego się w zakresie mniejszym niż $f_p/2$, a) widmo oryginalne, b) widmo sygnału próbkowanego



Rysunek 6. Widmo sygnału próbkowanego mieszczącego się w zakresie do $f_p/2$, a) widmo oryginalne, b) widmo sygnału próbkowanego



Rysunek 7. Widmo sygnału a) próbkowanego mieszczącego się w zakresie przekraczającym $f_p/2$, a) widmo oryginalne, b) widmo sygnału próbkowanego – widoczny aliasing

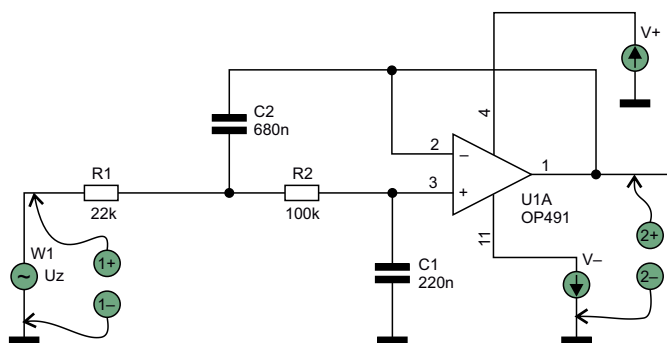


Rysunek 8. Charakterystyka częstotliwościowa przykładowych filtrów antyaliasingowych

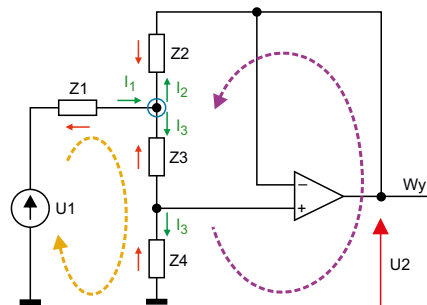
sygnału. Przypadek przedstawiony na **rysunku 6a** jest krytyczny. Pasmo sygnału próbkowanego sięga połowy częstotliwości próbkowania. Tu jeszcze nie występują żadne problemy, ale takie rozwiązanie można już nazwać jazdą po bandzie. Do katastrofy dochodzi natomiast w przypadku przedstawionym na **rysunku 7**. Pasmo sygnału przekracza połowę częstotliwości próbkowania i należy spodziewać się, że w sygnale spróbkowanym znajdą się składowe częstotliwościowe, których nie ma w oryginale. Mamy zatem do czynienia z aliasingiem. No tak, ale jak z nim walczyć?

Filtr antyaliasingowy

Można odpowiedzieć prosto: spróbować sygnał i zastosować odpowiednio dobrany filtr, żeby te składowe niepożądane usunąć. Nie! Nie tędy droga. Na filtrowanie byłoby już za późno. Przy okazji skutecznie cięlibyśmy sygnał użytkowy. Trzeba zapobiegać, nie leczyć. Owszem, trzeba zastosować filtr, lecz musimy zadbać o to, by widmo sygnału oryginalnego kończyło się przed połową częstotliwości próbkowania. Oznacza to, że filtrowanie musi być wykonane przed próbkowaniem. Zasadę pracy filtru antyaliasingowego przedstawiono na **rysunku 8**. Niestety, znowu pojawia się problem – przecież nie ma filtrów idealnych. Nie zbudujemy filtru, który przepuści wszystko do jakiejś częstotliwości i wytnie wszystko powyżej niej. Każdy rzeczywisty filtr przepuszcza wszystko do charakterystycznej dla niego częstotliwości granicznej f_g i z pewną



Rysunek 9. Filtr antyaliasingowy w konfiguracji Selleny-Key'a



Rysunek 10. Schemat układu Selleny-Key'a przerysowany do analizy transmitancji

szybkością tłumić częstotliwości powyżej niej. Szybkość tłumienia zależy od rzędu filtra. Filtr 1. rzędu tłumić z szybkością 20 dB/dekadę, filtr 2. rzędu 40 dB/dekadę itd. Im niższy jest rząd filtra, tym łatwiej można go zrealizować fizycznie, ale jak widać na rysunku 8 częstotliwość graniczna musi być bardziej oddalona od częstotliwości $f_p/2$. Czasami niestety kosztem pasma użytkowego.

Ustalenie częstotliwości granicznej filtra zależy od jego rzędu – to już wiemy, ale również od rozdzielczości przetwornika A/C, co już nie jest oczywiste. Z rozdzielczością przetwornika wiąże się zakres dynamiki, a więc odstęp sygnału od szumu. Parametr ten np. dla przetwornika 12-bitowego jest równy 74 dB (SNR = $6,02 \cdot N + 1,76$, gdzie N – rozdzielczość ADC). Teoretycznie filtr antyaliasingowy powinien stłumić sygnał co najmniej do poziomu wyznaczonego dynamiką przetwornika przy częstotliwości $f_p/2$.

Projektujemy filtr dolnoprzepustowy na 8,5 Hz

Naszym zadaniem jest zbudowanie filtra dolnoprzepustowego 2. rzędu o częstotliwości granicznej 8,5 Hz. Wybieramy jedną z najczęściej stosowanych struktur – Selleny-Keya (**rysunek 9**). Do obliczenia elementów filtra konieczna jest znajomość jego transmitancji. W literaturze można znaleźć gotowe wzory, lecz stosunkowo trudno jest znaleźć ich wyprowadzenie. Prawdopodobnie wielu Czytelników może być zainteresowanych tematem, więc spróbujmy to zrobić teraz wspólnie. W tym celu przerysowujemy nieco schemat filtra, rezygnujemy z konkretnych elementów R , C , zamiast nich wprowadzamy ogólne impedancje, które można będzie modyfikować np. wtedy gdy będziemy chcieli przerobić ten filtr na górnoprzepustowy.

Zmodyfikowany schemat przedstawiono na **rysunku 10**. Układamy równania opisujące układ. Obejmują one węzeł do którego dopływa prąd I_1 i wypływają prądy I_2 i I_3 , oczko zawierające wyjście, impedancje Z_2 , Z_3 i Z_4 oraz oczko U_1 , Z_1 , Z_3 i Z_4 .

- (1) $I_1 = I_2 + I_3$
- (2) $U_1 - I_1 Z_1 - I_3 (Z_3 + Z_4) = 0$
- (3) $U_2 + I_2 Z_2 - I_3 (Z_3 + Z_4) = 0$

Łącząc dwa pierwsze równania otrzymujemy:

$$U_1 - (I_2 + I_3) Z_1 - I_3 (Z_3 + Z_4) = 0$$

$$U_1 - I_2 Z_1 - I_3 Z_1 - I_3 Z_3 - I_3 Z_4 = 0$$

$$U_1 - I_2 Z_1 - I_3 (Z_1 + Z_3 + Z_4) = 0$$

stąd:

$$I_2 = \frac{U_1 - I_3 (Z_1 + Z_3 + Z_4)}{Z_1}$$

wstawiając powyższą zależność do równania (3):

$$U_2 + \frac{U_1 - I_3 (Z_1 + Z_3 + Z_4)}{Z_1} Z_2 - I_3 (Z_3 + Z_4) = 0$$

$$U_2 Z_1 + U_1 Z_2 - I_3 (Z_1 + Z_3 + Z_4) Z_2 - I_3 Z_1 (Z_3 + Z_4) = 0$$

stąd:

$$I_3 = \frac{U_2 Z_1 + U_1 Z_2}{Z_1 (Z_3 + Z_4) + Z_2 (Z_1 + Z_3 + Z_4)}$$

Teraz trzeba zauważyć, że wejście odwracające wzmacniacza operacyjnego jest połączone z wyjściem, ale pamiętamy, że potencjał wejścia nieodwracającego jest taki sam jak odwracającego i że do wejść wzmacniacza nie wpływają prądy. Wynika z tego, że:

$$I_3 = \frac{U_2}{Z_4}$$

więc:

$$\frac{U_2}{Z_4} = \frac{U_2 Z_1 + U_1 Z_2}{Z_1 (Z_3 + Z_4) + Z_2 (Z_1 + Z_3 + Z_4)}$$

i dalej:

$$U_2 Z_1 (Z_3 + Z_4) + U_2 Z_2 (Z_1 + Z_3 + Z_4) = U_2 Z_1 Z_4 + U_1 Z_2 Z_4$$

a z powyższego równania już prosta droga do obliczenia transmitancji:

$$K_u = \frac{U_2}{U_1} = \frac{Z_2 Z_4}{Z_1 (Z_3 + Z_4) + Z_2 (Z_1 + Z_3 + Z_4) - Z_1 Z_4} = \frac{Z_2 Z_4}{Z_1 (Z_2 + Z_3) + Z_2 (Z_3 + Z_4)}$$

Otrzymaliśmy w ten sposób równanie opisujące transmitancję układu w ogólnym przypadku. Teraz pod poszczególne impedancje występujące w tym wzorze podstawiamy elementy naszego filtru, a więc:

$$Z_1 = R_1$$

$$Z_2 = \frac{1}{sC_2}$$

$$Z_3 = R_2$$

$$Z_4 = \frac{1}{sC_1}$$

Po podstawieniu powyższych zależności uzyskujemy:

$$K_u(s) = \frac{U_2}{U_1} =$$

$$\frac{\frac{1}{sC_2} \frac{1}{sC_1}}{R_1 \left(\frac{1}{sC_2} + R_2 \right) + \frac{1}{sC_2} \left(R_2 + \frac{1}{sC_1} \right)} = \frac{1}{R_1 C_1 s (1 + R_2 C_2 s) + R_2 C_1 s + 1}$$

a po uporządkowaniu ostatecznie:

$$K_u(s) = \frac{U_2}{U_1} = \frac{1}{R_1 R_2 C_1 C_2 s^2 + C_1 (R_1 + R_2) s + 1}$$

Można powiedzieć, że to powinno wystarczyć do obliczenia wszystkich elementów, ale sprawa nie jest taka prosta. Otóż kilku mądrych ludzi (Butterworth, Bessel, Czebyszew) badało przed nami transmitancje takich i podobnych konfiguracji i okazało się, że nawet gdybyśmy projektowali filtr na określoną częstotliwość graniczną, to moglibyśmy uzyskać różne wyniki. Owszem, wszystkie filtry miałyby tę samą częstotliwość 3-decybelową i spadek charakterystyki z szybkością 40 dB/dekadę, ale np. różnie by wyglądało kolano załamania charakterystyki, nieco inaczej przebiegałaby charakterystyka fazowa, mogłyby wystąpić zafalowania charakterystyki amplitudowej itd. Decydują o tym współczynniki przy zmiennej s. Nawiasem mówiąc zmienna ta występuje w mianowniku w drugiej potęgze, mamy więc do czynienia z filtrem 2. rzędu o spadku charakterystyki w paśmie zaporowym z szybkością 40 dB/dek lub 6 dB/oktawę. Współczynniki o których mowa nazwano i zdefiniowano dla poszczególnych odmian filtrów z uwzględnieniem pulsacji granicznej, dla której występuje biegun transmitancji (zeruje się jej mianownik):

$$a_1 = C_1 (R_1 + R_2) \omega_g \quad (4)$$

$$b_1 = R_1 R_2 C_1 C_2 \omega_g^2 \quad (5)$$

Na przykład dla filtru Butterwortha są one równe:

$$a_1 = 1,4142$$

$$b_1 = 1$$

Projektowanie filtru zaczynamy od przyjęcia wartości np. kondensatora C_1 . Z zależności (5) obliczamy np. R_2 :

$$R_2 = \frac{b_1}{R_1 C_1 C_2 \omega_g^2}$$

i podstawiamy do (4):

$$a_1 = C_1 \left(\frac{b_1}{R_1 C_1 C_2 \omega_g^2} + R_1 \right) \omega$$

$$a_1 R_1 C_1 C_2 \omega_g^2 = C_1 b_1 \omega + R_1^2 C_1^2 C_2 \omega_g^3$$

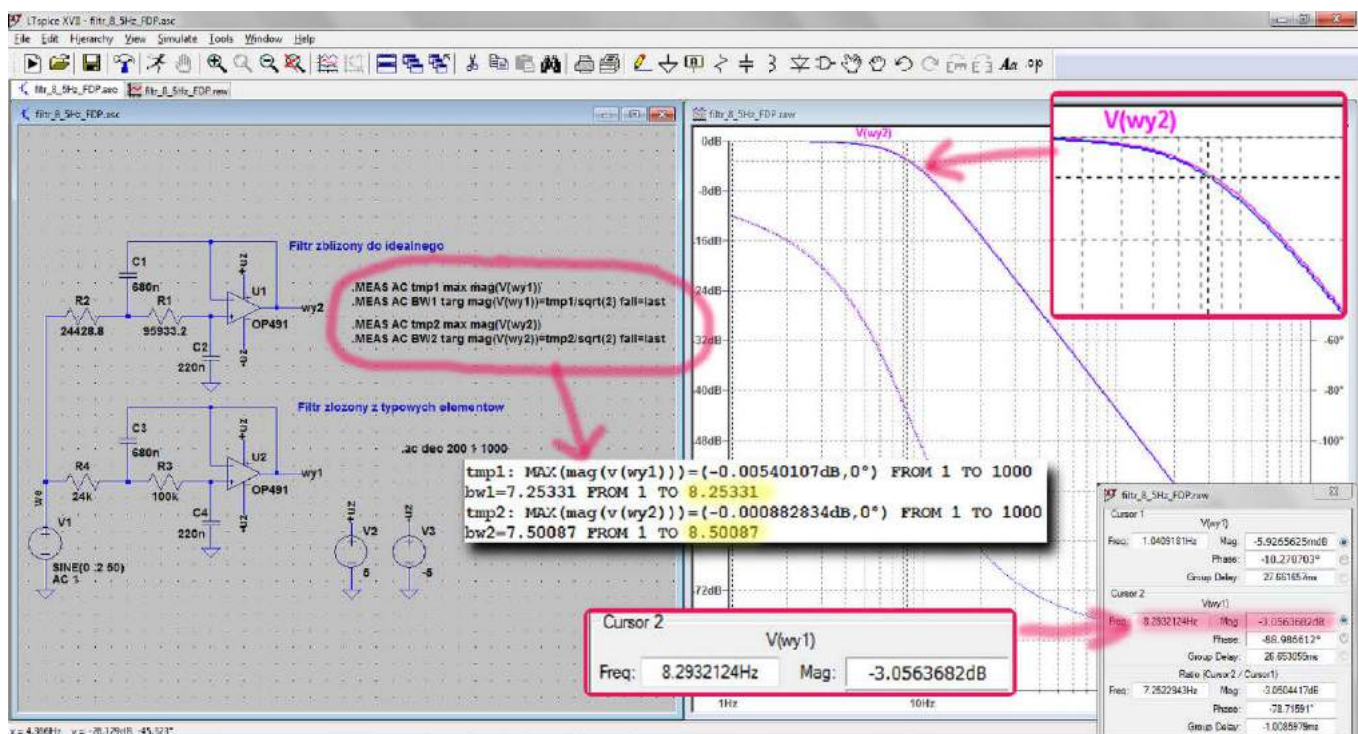
$$a_1 R_1 C_1 C_2 \omega g = C_1 b_1 + R_1^2 C_1^2 C_2 \omega_g^2$$

$$C_1^2 C_2 \omega_g^2 R_1^2 - a_1 C_1 C_2 \omega_g R_1 + C_1 b_1 = 0$$

Otrzymaliśmy równanie kwadratowe, w którym niewiadomą jest R_1 . Pozostaje więc policzenie delty i rozwiązanie równania.

$$\Delta = a_1^2 C_1^2 C_2^2 \omega_g^2 - 4 C_1^2 C_2 \omega_g^2 C_1 b_1 = \omega_g^2 (a_1^2 C_1^2 C_2^2 - 4 C_1^3 C_2 b_1)$$

czyli:



Rysunek 11. Symulacja zaprojektowanego filtru w LTspice

$$R_1 = \frac{a_1 C_1 C_2 \omega_g - \sqrt{\omega_g^2 C_1^2 (a_1^2 C_2^2 - 4 C_1 C_2 b_1)}}{2 C_1^2 C_2 \omega_g^2} =$$

$$\frac{a_1 C_1 C_2 \omega_g - \omega_g C_1 \sqrt{a_1^2 C_2^2 - 4 C_1 C_2 b_1}}{2 C_1^2 C_2 \omega_g^2}$$

$$R_1 = \frac{a_1 C_2 - \sqrt{a_1^2 C_2^2 - 4 C_1 C_2 b_1}}{2 C_1 C_2 \omega_g}$$

W powyższym wzorze występuje pojemność C_1 , którą założyliśmy wcześniej oraz C_2 , którą również założyliśmy. Widzimy jednak, że C_1 i C_2 występują pod pierwiastkiem, trzeba więc zadbać aby całe wyrażenie pod pierwiastkiem nie było ujemne.

$$a_1^2 C_2^2 - 4 C_1 C_2 b_1 \geq 0 \Rightarrow C_2 \geq \frac{4 b_1}{a_1^2} C_1 \quad (6)$$

Rezystancja R_2 będzie równa:

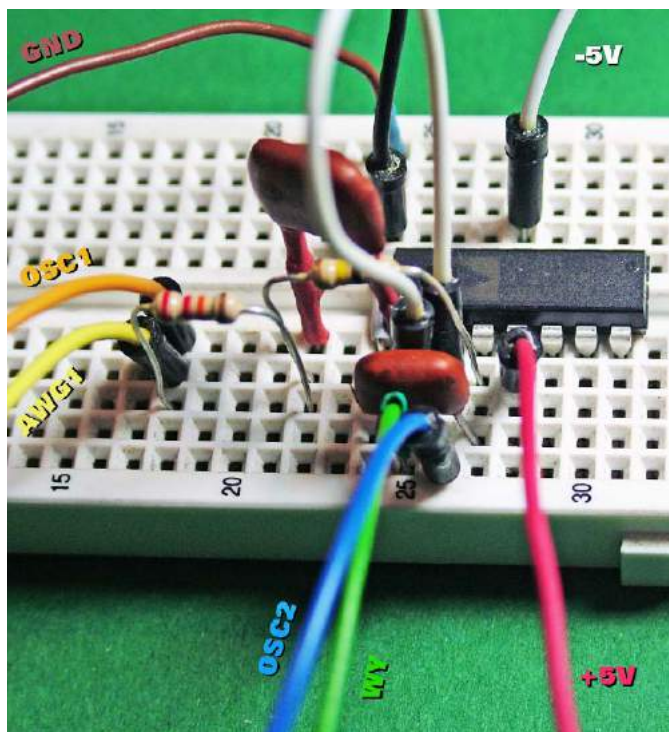
$$R_2 = \frac{a_1 C_2 + \sqrt{a_1^2 C_2^2 - 4 C_1 C_2 b_1}}{2 C_1 C_2 \omega_g}$$

To już naprawdę koniec teorii. Szybciutko obliczamy elementy filtru i przystępujemy do działań praktycznych. W materiałach uzupełniających do artykułu na CD znajdzie się arkusz excela, który ułatwi wykonanie obliczeń filtru FDP o zadanej częstotliwości 3-decybelowej wg powyższego algorytmu. Liczymy.

Obliczenia i symulacja

Zakładamy $C_1 = 220$ nF oraz przyjmujemy dane a_1 i b_1 dla filtru Butterwortha. Z równania (6) wynika, że C_2 musi być większe od 440 nF – przyjmujemy z szeregu 680 nF. Teraz można już obliczyć: $R_1 = 24428,8 \Omega$ – przyjmujemy 24 kΩ, i $R_2 = 95933,2 \Omega$ – przyjmujemy 100 kΩ.

Rysujemy w edytorze LTspice dwa schematy. Jeden będzie zawierał elementy o wartościach dokładnie obliczonych, drugi będzie miał elementy z szeregu E24, a więc takie, które posłużą następnie do budowy układu fizycznego. Schemat w LTspice wraz z wynikami pomiarów przedstawiono na **rysunku 11**. Charakterystyki częstotliwościowe obu filtrów umieszczono na jednym wykresie. Jak widać odstępstwa od wartości obliczonych nie spowodowały drastycznych różnic w charakterystykach, jednak do realizacji takich filtrów powinno się



Fotografia 2. Badany układ zmontowany na breadboardzie



Rysunek 12. Wyniki pomiarów układu rzeczywistego

używać elementów o tolerancjach nie gorszych niż 1%. Pomiar kursorowy 3-decybelowej częstotliwości granicznej układu z elementami rzeczywistymi dał wynik 8,29 Hz wobec 8,5 zakładanych. Pomiar kursorowy nie jest jednak bardzo dokładny. Kursora nie można ustawić w pozycji zadanej np. liczbowo, jest on ustawiany jedynie za pomocą myszki. Decyduje więc sprawność ręki i rozdzielczość grafiki. Dużo dokładniejszy wynik można uzyskać za pomocą komendy MEAS. Jak już wiemy definiujemy ją np. klikając na ikonkę .op. Do obliczenia częstotliwości 3-decybelowej konieczne jest przeprowadzenie analizy AC. Uruchamiamy ją komendą

.ac dec 200 2 1000

co oznacza analizowanie z rozdzielczością 200 punktów na dekadę od 1 Hz do 1000 Hz. Do obliczenia częstotliwości granicznej można posłużyć się skryptem:

.MEAS AC tmp1 max mag(V(wy1))

.MEAS AC BW1 targ mag(V(wy1))=tmp1/sqrt(2) fall=last

To jest skrypt dla filtru górnego, dla dolnego natomiast trzeba zmodyfikować zmienną tymczasową i wyjście:

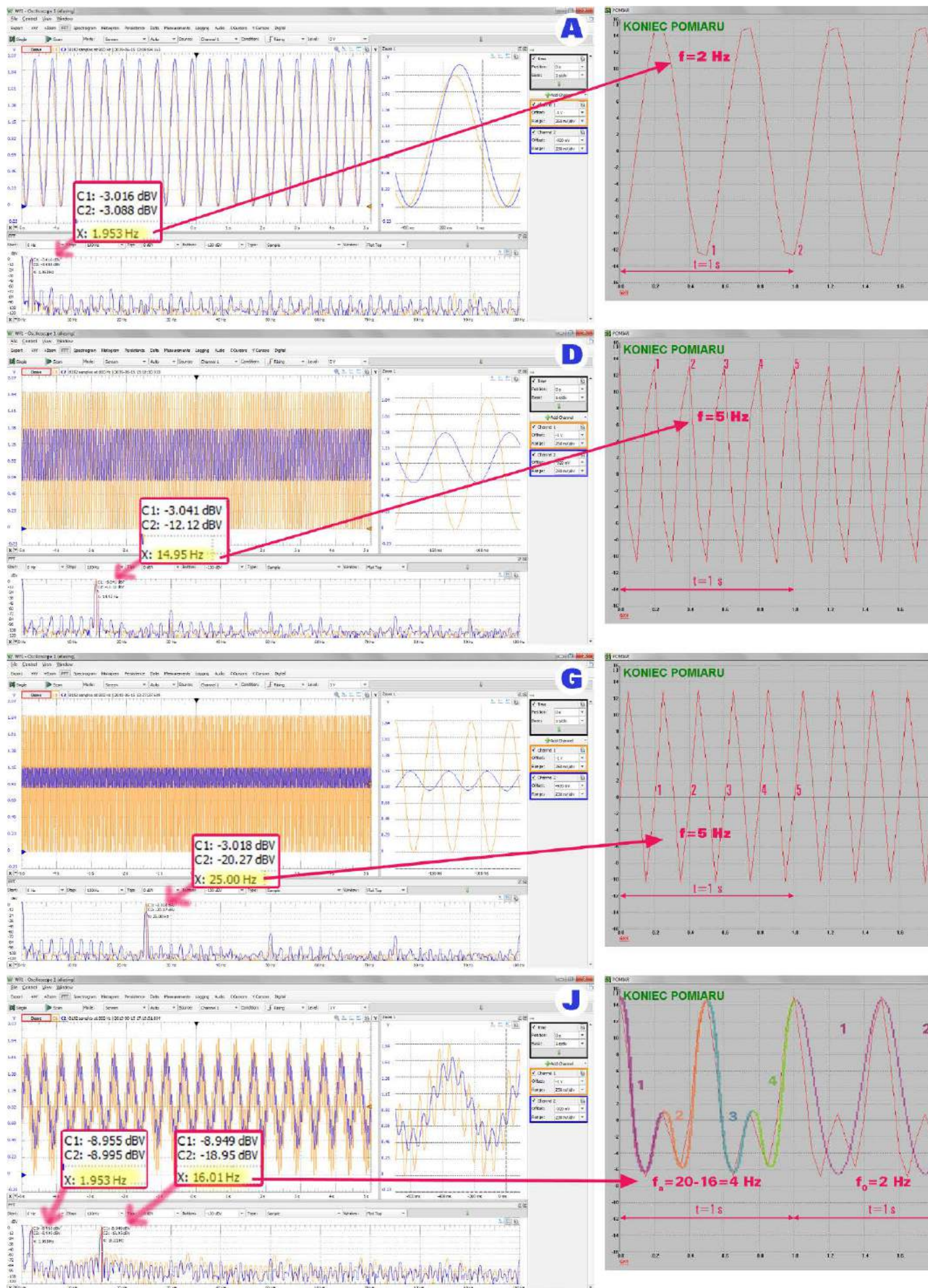
.MEAS AC tmp2 max mag(V(wy2))

.MEAS AC BW2 targ mag(V(wy2))=tmp2/sqrt(2) fall=last

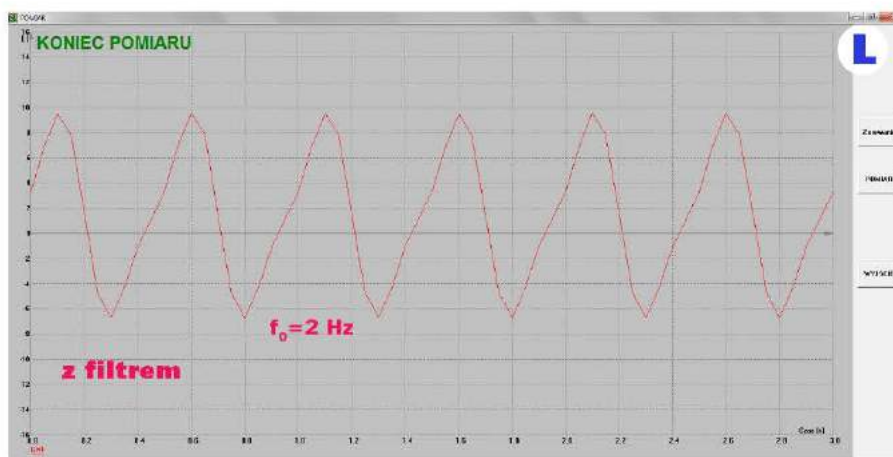
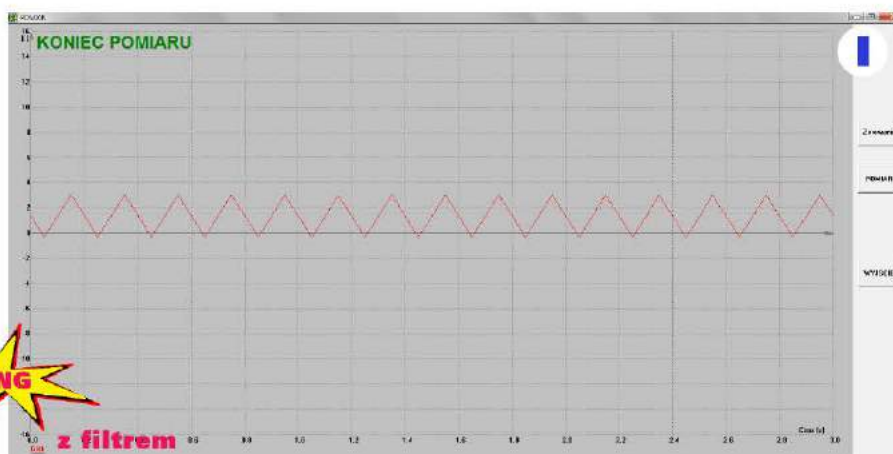
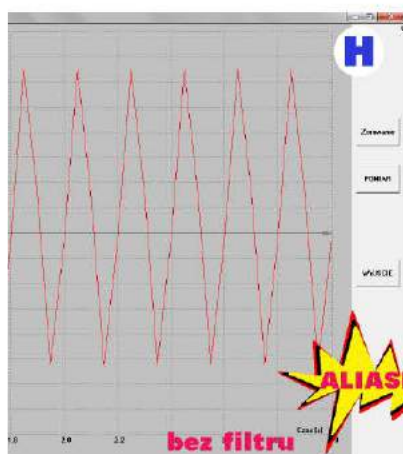
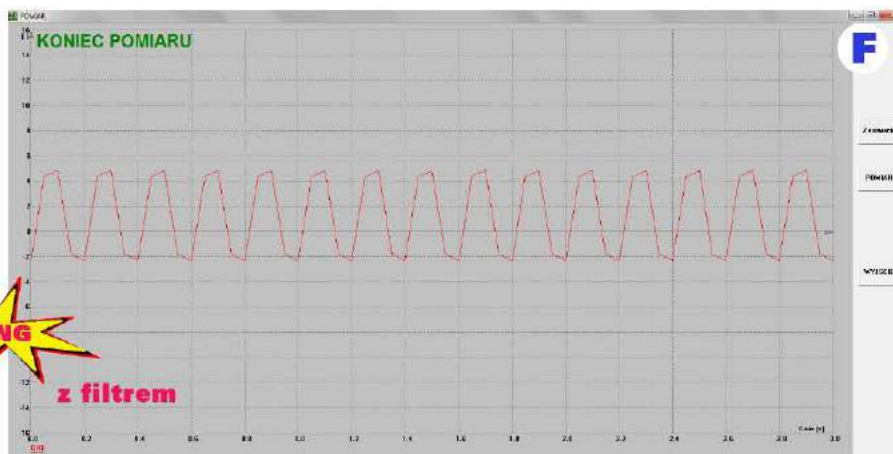
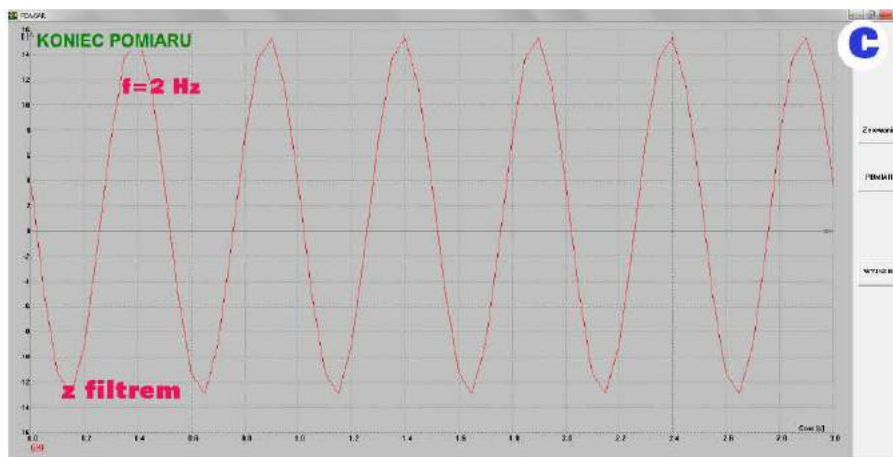
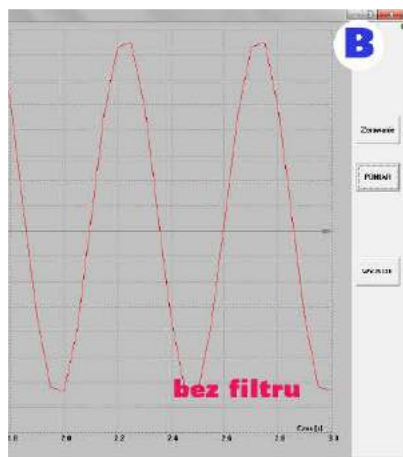
Pamiętamy, że przechodzenie do nowej linii w oknie edycji następuje po naciśnięciu kombinacji przycisków Ctrl-M, Enter natomiast kończy edycję. Skrypt powoduje utworzenie zmiennej tymczasowej tmp1 (tmp2), która zapamiętuje maksymalną wartość charakterystyki amplitudowej. Druga zmienna BW1 (BW2 dla dolnego układu) oblicza natomiast szerokość pasma. Obliczenia rozpoczynają się od częstotliwości początkowej a kończą w chwili, gdy charakterystyka amplitudowa zmniejsza się do wartości $U_{max}/\sqrt{2}$, co odpowiada spadkowi amplitudy o 3 dB. Różnica częstotliwości przy której zakończono obliczenia i częstotliwości początkowej wyznaczają pasmo (BW), a sama częstotliwość kończąca obliczenia wyznacza 3-decybelową częstotliwość graniczną filtru. Do odczytu tych zmiennych jak zwykle korzystamy z polecenia „View -> SPICE Error Log”. Wyniki pomiaru są widoczne na rysunku 11 i wynika z nich, że częstotliwości graniczne dla filtru idealnego i filtru składającego się z elementów o wartościach z szeregu E24 są równe odpowiednio: 8,50087 Hz i 8,25331 Hz. Całkiem nieźle. Zobaczmy teraz, jak zachowa się układ rzeczywisty. Montujemy go oczywiście na breadboardzie i uruchamiamy Analog Discovery 2.

Pomiary układu rzeczywistego

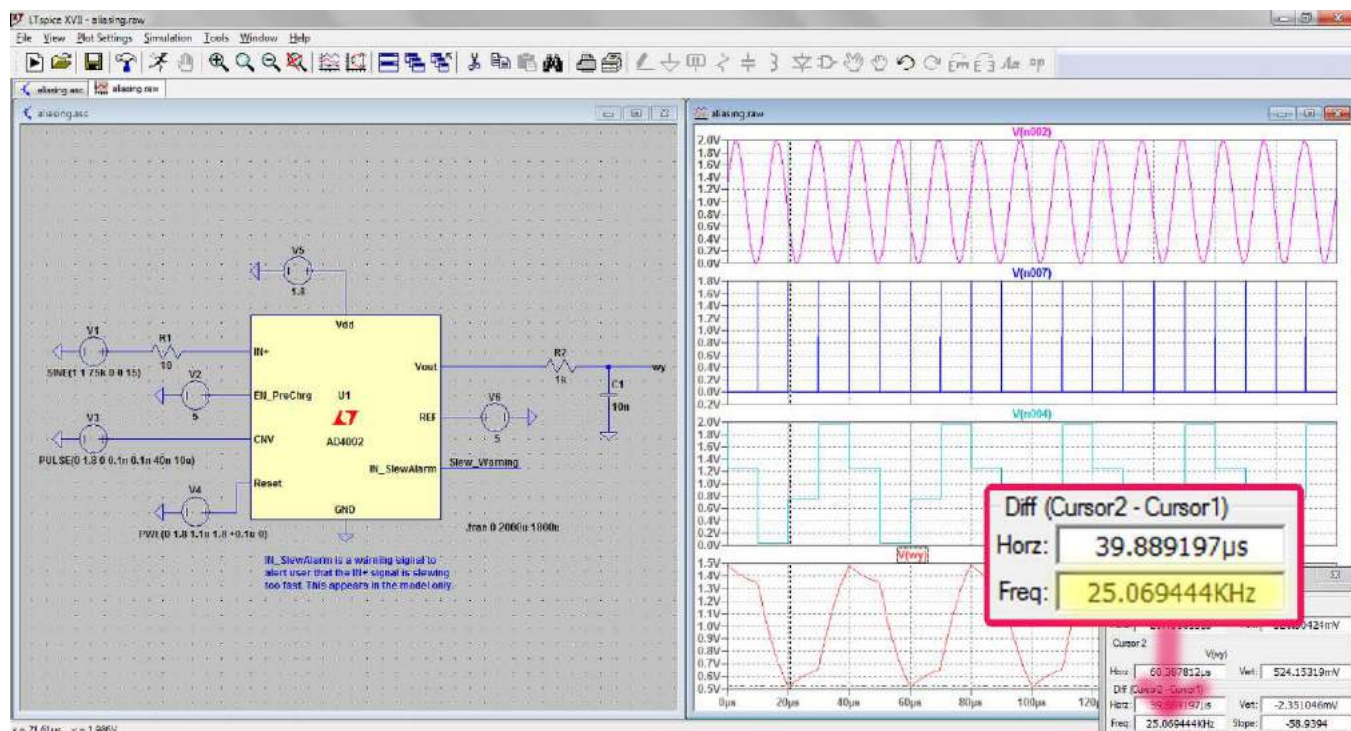
Zmontowany układ prezentuje **fotografia 2**, a dołączenie poszczególnych przyrządów pomiarowych zaznaczono na schemacie z rysunku 9. Wykorzystujemy dwa kanały oscyloskopu, kanał 1 generatora arbitralnego i obie sekcje zasilacza z napięciami +5 V i -5 V. Naszym głównym zadaniem będzie zmierzenie charakterystyki częstotliwościowej, ale wcześniej warto sprawdzić oscyloskopem czy układ w ogóle działa. Do takiego testu wybieramy sygnał



Rysunek 13. Odtworzenie sygnału próbkowanego z częstotliwością 20 Hz w programie do analizy przebiegów: a) oryginalny przebieg o częstotliwości 2 Hz, b) odtworzony przebieg 2 Hz bez filtru, c) odtworzony przebieg 2 Hz z filtrem, d) oryginalny przebieg o częstotliwości 15 Hz, e) odtworzony przebieg 15 Hz bez filtru, f) odtworzony przebieg 15 Hz z filtrem, g) oryginalny przebieg o częstotliwości 25 Hz, h) odtworzony



przebieg 25 Hz bez filtru, i) odtworzony przebieg 25 Hz z filtrem, j) oryginalny przebieg ze składowymi 2 Hz i 16 Hz, h) odtworzony przebieg ze składowymi 2 Hz i 16 Hz bez filtru, i) odtworzony przebieg ze składowymi 2 Hz i 16 Hz z filtrem



Rysunku 14. Schemat układu pomiarowego wraz z wynikami symulacji

sinusoidalny o amplitudzie 1 V i częstotliwości 1 Hz, z zerowym offsetem. Podstawę czasu oscyloskopu ustawiamy na 1 s/dz, a czułości obu kanałów na 500 mV/dz z zerowym offsetem. Powinniśmy zobaczyć dwie niemal nakładające się na siebie sinusoidy. Jeśli tak jest przechodzimy do pomiaru zasadniczego, jeśli nie, trzeba sprawdzić połączenia.

Charakterystykę częstotliwościową zdejmujemy używając narzędzia Network. I w tym przypadku ustawiamy napięcie 1 V z zerowym offsetem. Dla wykresu amplitudowego ustawiamy parametry: Top=2 dB, Bottom=-78 dB, a dla wykresu fazowego: Offset=93°, Range=180°. Dla parametrów obu kanałów pomiarowych można wybrać opcję auto. Warto jak zwykle włączyć opcję „View->Time”, żeby sprawdzić czy np. nie przesterowujemy układu badanego. Uruchamiamy pojedynczy pomiar (Single) w zakresie: Start=1 Hz, Stop=1 kHz. Wiemy już, że dla częstotliwości początkowej 1 Hz pomiar w pierwszej fazie przebiega dość wolno. Uzbijamy się więc w cierpliwość i po kilku minutach mamy już wykres częstotliwościowy rzeczywistego filtru zmontowanego na breadboardzie (rysunek 12). Wynika z niego, że częstotliwość 3-decybelowa filtru jest równa 8,98 Hz. Można powiedzieć, że zadanie zostało wykonane prawidłowo, chociaż symulacja wykazała, że przy rzeczywistych wartościach elementów częstotliwość graniczna powinna być niższa od teoretycznej, natomiast w układzie badanym jest ona wyższa. Pół herca różnicy jest jak najbardziej do zaakceptowania.

Jeszcze raz wracamy do aliasingu

Skoro już zrobiliśmy rzeczywisty filtr, warto sprawdzić jego działanie w jakimś układzie praktycznym. Mamy więc układ mikroprocesorowy z przetwornikiem analogowo-cyfrowym, dostarczający dane do programu, którego zadaniem jest analiza sygnału dołączonego do wejścia. Sprawdzimy, czy rzeczywiście przekroczenie częstotliwości Nyquista (przyp. poza połowę częstotliwości próbkowania) powoduje wystąpienie przekłamań w próbkowanym sygnale i na ile zapobiega temu filtr antyaliasingowy.

Przetwornik analogowo-cyfrowy naszego urządzenia próbkuje z częstotliwością 20 Hz, aplikacja natomiast odtwarza sygnał oryginalny z próbek z zastosowaniem interpolacji liniowej, czyli przez bezpośrednie łączenie ze sobą kolejnych próbek odcinkami (rysunek 13). Wykresy A, D, G, J na tym rysunku to oscylogramy sygnału

zmierzone oscyloskopem. Wykresy B, E, H, K to przebieg przebieg utworzony z próbek bez użycia filtru antyaliasingowego, zaś w ostatniej kolumnie mamy przypadek z użyciem filtru antyaliasingowego włączonego pomiędzy generatorem a przetwornikiem A/C.

W pierwszym pomiarze próbkowaniu był poddawany sygnał sinusoidalny o częstotliwości 1 Hz (rysunek 13a). Jak widać na rysunkach 13b i 13c został on odtworzony bardzo wiernie. W drugim pomiarze (rysunek 13d, 13e, 13f) częstotliwość mierzona 15 Hz była większa od częstotliwości Nyquista równej 10 Hz. W wersji bez filtru widoczna jest wyraźnie składowa aliasingowa 5 Hz (20 Hz minus 15 Hz) o amplitudzie prawie takiej samej, jak w prawidłowym pomiarze pierwszym.

No dobrze – chciałoby się powiedzieć – więc pomiar z filtrem powinien być już prawidłowy. Pamiętajmy jednak, że nasz filtr ma spadek charakterystyki 40 dB/dekadę i częstotliwość graniczną 8,9 Hz. Oznacza to, że dla częstotliwości 15 Hz stłumi sygnał o 12 dB, tj. ok. 4-krotnie. Sygnał oryginalny ma amplitudę peak-to-peak 15(-13)=28 umownych jednostek, zaś występujący w tym pomiarze sygnał aliasingowy ma amplitudę peak-to-peak 5-(-2)=7, a więc rzeczywiście został 4-krotnie stłumiony. Aby był on stłumiony skuteczniej należałoby zastosować filtr wyższego rzędu lub – niestety – przesunąć w dół częstotliwość graniczną.

W trzecim pomiarze (rysunek 13g, 13h, 13i) mamy przypadek, w którym częstotliwość sygnału 25 Hz jest nawet większa od częstotliwości próbkowania. Nie jest to sytuacja normalna, ale warto ją zbadać. Bez filtru odnosimy wrażenie, że mierzymy sygnał o częstotliwości 5 Hz (25-20), co oczywiście nie jest prawdą. W układzie z filtrem aliasing został zredukowany już bardzo wyraźnie. Teoretycznie filtr stłumił sygnał o ok. 21 dB, a więc 11 razy.

Na zakończenie jeszcze pomiar sygnału zawierającego dwie składowe częstotliwościowe: 2 Hz i 16 Hz (rysunek 13j, 13k, 13l). Przygotowanie takiego sygnału w generatorze arbitralnym Analog Discovery 2 jest bardzo proste, będziemy jeszcze o tym mówić w kolejnych odcinkach. W odtworzonym przebiegu bez filtru można wyraźnie doszukać się dwóch składowych: 2 Hz – składowa podstawowa i 4 Hz – składowa aliasingowa pochodząca od występującej w sygnale oryginalnym składowej 16 Hz (20-16=4). Po zastosowaniu filtru aliasing jest prawie niewidoczny, wyraźnie natomiast rozpoznajemy składową podstawową 2 Hz.

Należy się jeszcze wytłumaczenie, dlaczego były mierzone tak małe częstotliwości. Otóż w chwili pisania artykułu dysponowałem urządzeniem przeznaczonym do badania przebiegów o względnie niskich częstotliwościach, które posłużyło jako interfejs między generatorem a aplikacją użytkową. Urządzenie to samo w sobie ma już własny filtr antyaliasingowy, żeby więc nie wejść w zakres jego działania konieczne było zejście do tak niskich częstotliwości. Wszystkie prawa oczywiście nadal obowiązują.

Ostrzeżenie

Można powiedzieć, że aliasing atakuje z każdej, najmniej oczekiwanej strony. Przykładem niech będą choćby pomiary oscyloskopowe. Przy złe dobranej podstawie czasu możemy zobaczyć na ekranie to, czego nie ma w mierzonym sygnale. Przykład pokazano na fotografii tytułowej. Oscyloskopy na ogół nie posiadają filtrów antyaliasingowych.

Aliasing w symulacji

Dla niedowiarków, którzy nie dysponują urządzeniem mogącym posłużyć do zaobserwowania aliasingu w rzeczywistości z pomocą przychodzi oczywiście symulator LTspice. W standardowej bibliotece tego programu znajduje się m.in. przetwornik AD4002, który wykorzystamy do eksperymentu. Schemat układu pomiarowego wraz z wynikami symulacji przedstawiono na **rysunku 14**. Przetwornik jest zasilany napięciem 1,8 V, do pracy wymaga jeszcze dołączenia napięć V_{REF} i $V_{EN_PreChrg}$, które są równe 5 V. Do wejścia IN+ doprowadzamy sygnał sinusoidalny o amplitudzie 1 V, offsecie 1 V i częstotliwości 75 kHz. Do wejścia CNV jest podawany

przebieg taktujący próbkowaniem. Są to impulsy o szerokości 40 ns mające czas narastania i opadania równy 100 ps. Impulsy mają częstotliwość 100 kHz, tyle więc wynosi częstotliwość próbkowania. Jak widać w układzie nie zastosowano filtra antyaliasingowego, a częstotliwość sygnału przekracza częstotliwość Nyquista (w tym przypadku 50 kHz). Możemy więc spodziewać się aliasingu. Rzeczywiście, na wyjściu widzimy sygnał o częstotliwości 25 kHz ($100 - 75 = 25$). Częstotliwość mierzymy za pomocą kursorów bezpośrednio z wykresu. Takiego sygnału na pewno nie podawaliśmy na wejście przetwornika. Nie chcę nawet myśleć co by się działo, gdybyśmy taki wynik zamieścili na przykład w jakiejś pracy naukowej. Byłaby to kosmiczna kompromitacja.

To był długi odcinek, ale opisywał niezwykle istotne zagadnienie. Następny będzie krótszy – spróbujemy rozstrzygnąć czy mostkiem Wheastone'a można mierzyć pojemność?

Jarosław Doliński, EP

Źródła:

Fotografie: Shannon – Photo by Keystone/Getty Images, Kotielnikow – Wikipedia, <http://bit.ly/2mt98Cf>

1. Nyquist, Harry. „Certain factors affecting telegraph speed”. Bell System Technical Journal, 3, 324–346, 1924.
2. H. Nyquist, „Regeneration theory”, Bell System Technical Journal, vol. 11, pp. 126-147, 1932.
3. Bruce Carter, Ron Marcini, „Wzmocniacze operacyjne teoria i praktyka”, Wydawnictwo BTC, Wydanie I

REKLAMA

www.facebook.com/ElektronikaPraktyczna



UWAGA KONKURS

Elektroniku mały i duży

KITyAVT ogłasza konkurs wiedzy o zestawach AVT.

Do zdobycia atrakcyjne nagrody i gadżety. Szczegóły i regulamin na profilu Facebook KITyAVT

nie przegap
śledź profil już
od dziś...



SKLEP.AVT.PL

Śledź nowości, bierz udział w konkursach, zgarnij odłotowe gadżety i nagrody do Kity AVT.

Nie przegap!

interesujących materiałów w siostrzanym czasopiśmie



W październikowym wydaniu Elektroniki dla Wszystkich między innymi:

Elektroniczna gra – łamigłówka.
Nauka podstaw elektroniki
Znakomity sposób by zainteresować elektroniką młodsze pokolenie! Łamigłówka polegająca na zbudowaniu z dostarczonych podzespołów działającego układu elektronicznego.

Izolowana galwanicznie sonda oscyloskopowa 2,5kV
Interesujący przykład jak można samodzielnie wykonać wysokonapięciową sondę oscyloskopową, i to izolowaną galwanicznie od masy oscyloskopu i przewodu uziemiającego.

Kuchnia Konstruktora. Taki zwyczajny zasilacz...
W drugim odcinku cyklu decydujemy się na konkretne rozwiązania układowe i materiałowe, tworzymy schemat oraz budujemy próbną wersję podstawową.

Przetwornice indukcyjne
Po omówieniu różnych wersji przetwornicy Røysera wracamy do klasycznych przetwornic z dwukierunkowym magnesowaniem rdzenia. Zaczynamy rozwiązywać zagadkę, jaką drogą płynie prąd magnesujący w przerwach między impulsami roboczymi.

Kurs Arduino – matrycowe wyświetlacze LED
W dziewiętnastym odcinku kursu omawiamy różne sposoby sterowania wyświetlaczami LED. Zapoznajemy się z kostką MAX7219 i jak to zwykle jest z Arduino... napotykamy poważne problemy.

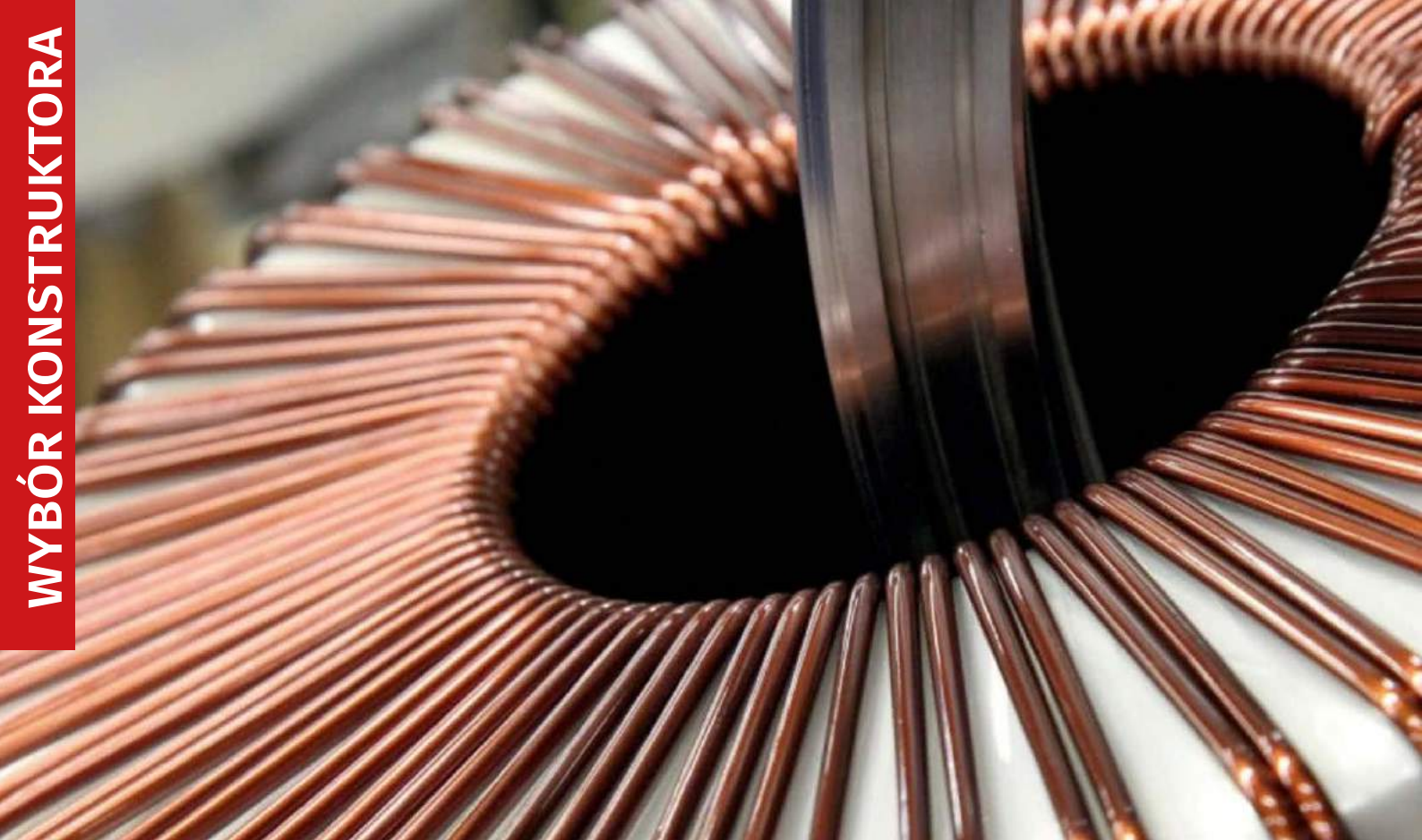
Ponadto w numerze:

- Miernik lamp elektronowych, część 2
- Infinity – system automatyki domowej.
- Współpraca z czujnikiem tlenu węgla
- Wtórnik audio z zasilaczem
- Wielofunkcyjne obciążenie aktywne, czyli... układ do marmowania energii
- Automatyka – elektronika w wentylacji i rekuperacji
- Szkoła Konstruktorów – Przedstaw swój wcześniejszy układ, lub taki, który chciałbyś zrealizować
- Szkoła Konstruktorów – Zaproponuj elektroniczny układ dowolnej przystawki
- Baterie odbiorniki lampowe, część 1

www.elportal.pl

A może masz pomysł na ciekawy artykuł lub projekt?
Skonstruowałeś urządzenie,
które jest godne zaprezentowania szerszej publiczności?
Możesz napisać artykuł edukacyjny?
Chcesz podzielić się doświadczeniem?
W takim razie zapraszamy do współpracy na łamach Elektroniki dla Wszystkich.
Kontakt: edw@elportal.pl

EdW możesz zamówić na www.ulubionykiosk.pl
Do kupienia również w Empikach i wszystkich większych kioskach z prasą.



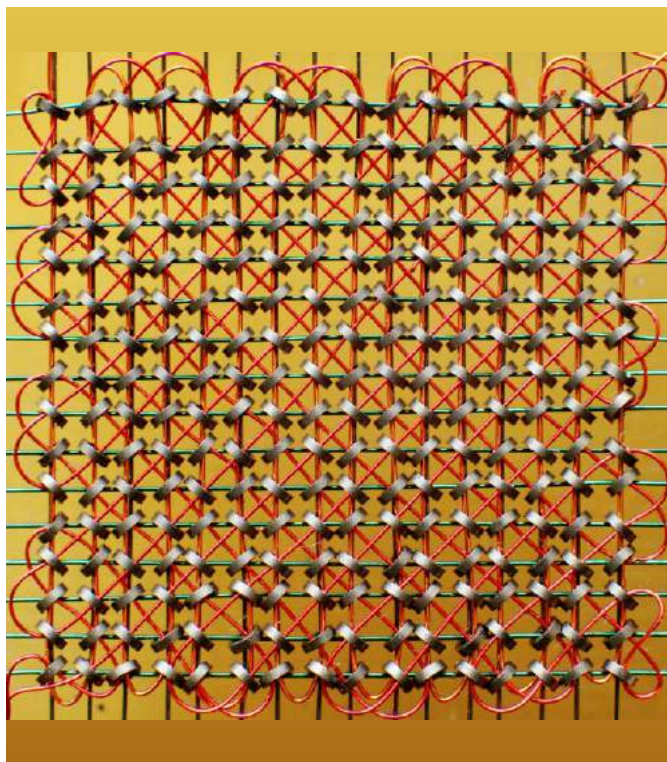
Elementy indukcyjne i transformatory

*Elementy indukcyjne, a w szczególności cewki, zawsze były zmartwieniem elektroników. Były i są utrapieniem dla większości elektroników, ale niewielki wpływ miały na to wykonywane ręcznie pamięci ferrytowe, zawierające setki i tysiące małych ferrytowych rdzeni toroidalnych, jak pokazuje przykład na **fotografii 1**. Cieszymy się, że dziś misternie tkane magnetyczne pamięci ferrytowe są jedynie mało znaną ciekawostką, ale z elementami magnetycznymi nadal jest kłopot.*

Zrozumienie zasad działania innych elementów elektronicznych nie stanowiło większego problemu.

Natomiast wielu elektronikom trudno było przyswoić zagadnienia związane z indukcyjnością. I niewiele pomagało prawdziwe w sumie stwierdzenie, że „cewki działają tak samo jak kondensatory, tylko odwrotnie”. Niektórzy do dziś pamiętają jedynie łatwo trafiające do przekonania powiedzenie: *prąd w cewce opóźnia się względem napięcia, bo... zaplątuje się w zwojach*.

Działanie kondensatorów łatwo poczuć intuicyjnie, ponieważ ich zachowanie stanowi analogię zachowania obiektów z naszego życia codziennego, w najprostszym przypadku choćby wiadra z wodą. Natomiast początkującym trudno wyobrazić i przyswoić sobie „odwrotne” działanie cewek. Nawet gdy ktoś już zaakceptował fakt, że napięcie samoindukcji może stać się dowolnie duże i zrozumiał podstawową



źródło: Wikipedia, CC BY-SA 3.0

Fotografia 1. Wykonana ręcznie pamięć ferrytowa

zależność $u = -L(\Delta i / \Delta t)$, nie stawał się ekspertem. Kolejnym problemem było zrozumienie ograniczeń, związanych z wykorzystaniem rdzeni ferromagnetycznych, które z jednej strony są błogosławieństwem, z drugiej – podstawowym źródłem kłopotów.

Gdy elektronik przebrnął także i przez te zagadnienia, trafiał na kolejny orzech do zgryzienia: często z wykorzystaniem niedoskonałego rdzenia musiał zaprojektować cewkę lub transformator o dokładnie określonych właściwościach. Te parametry to nie tylko indukcyjność i prąd maksymalny, ale też dobroć, stabilność cieplna oraz skłonność do wytwarzania zakłóceń i wrażliwość na zewnętrzne zakłócenia.

Dawniej wielu elektroników, nie tylko hobbystów, zmuszonych było nie tylko zaprojektować, ale i samodzielnie wykonać cewkę czy transformator. A nie jest łatwe porządne nawinięcie uzwojenia z zastosowaniem sekcjonowania i przekładek, zawierającego kilkanaście tysięcy zwojów drucikiem przypominającym włos, o średnicy 0,07 czy nawet 0,06 mm. Do tego dochodzi jeszcze porządne złożenie rdzenia z blaszanych kształtek. A takie dławiki stosowano w urządzeniach lampowych

Szczęśliwie rozwój techniki zmniejszył zapotrzebowanie na rozmaite elementy indukcyjne, zwłaszcza te precyzyjne, stosowane w filtrach, rozdzielających sygnały o różnych częstotliwościach. Dziś takie filtry realizowane są zupełnie inaczej, najczęściej cyfrowo. Obecnie mówiąc o indukcyjnościach do filtrów, mamy na myśli filtry przeciwzakłóceniu.

Elementy indukcyjne nadal stanowią bardzo ważny składnik urządzeń elektronicznych, ale dziś sytuacja jest łatwiejsza pod wieloma względami. Niezbyt często element indukcyjny trzeba projektować dokładnie pod specyficzne potrzeby. W ogromnej większości przypadków można wykorzystać gotowe elementy z szerokiej oferty producentów i dystrybutorów. Niemniej nadal trzeba, a przynajmniej wypadałoby rozumieć parametry, a w szczególności ograniczenia występujące w cewkach i transformatorkach.

Zakup elementów indukcyjnych u ich producenta w naturalny sposób związany jest z konsultacjami. Bo przecież producent to nie tylko kierownictwo firmy, maszyny i robotnicy, ale też ekipa fachowców w biurze konstrukcyjnym.

Gorzej jest (może być) przy zakupie elementów u dystrybutora. Pamiętajmy, że elementy indukcyjne zawsze były i nadal są, zmorą większości elektroników z uwagi na specyficzne parametry i ograniczenia. Zapewne nie są jednak zmorą większości handlowców, którzy po prostu nie czują szczegółów technicznych. Dlatego przy zakupie elementów indukcyjnych u dystrybutora jakiegoś dużego zagranicznego koncernu trzeba zadać sobie pytanie: czy jesteśmy pewni, co dokładnie chcemy kupić?

Jeżeli jesteśmy pewni, to nie ma problemu. Ale jeśli w grę wchodzi konsultacja w sprawie doboru elementu o optymalnych właściwościach i optymalnej cenie, to otwarte pozostaje pytanie, czy pracownicy dystrybutora mają odpowiednią wiedzę? Czy może lepiej konsultować swoje potrzeby z pracującymi dla producenta fachowcami, którzy według potrzeb klientów projektują elementy indukcyjne z wykorzystaniem współcześnie dostępnych materiałów ferromagnetycznych?

Jak to w życiu bywa, jednoznacznej odpowiedzi nie ma. Aby wybrać optymalne rozwiązanie, zawsze trzeba rozważyć sytuację i wziąć pod uwagę szereg czynników, nie tylko ściśle technicznych.

Autor w czasie pisania tego materiału stoi przed dylematem wyboru źródła zasilania, a konkretnie zasilacza buforowego dla zastawu akumulatorów rezerwowych (SLA 360 Ah) do inteligentnego domu, raczej domku. W grę wchodzi albo zastosowanie transformatora toroidalnego, albo zasilacza impulsowego, prawdopodobnie z przetwornicą wstępną PFC. Każde z rozwiązań ma różne zalety i wady.

Jednym z aspektów jest to, co powiedział inżynier Mamoiń w kultowym filmie Rejs. Zapewne inżynier Mamoiń elektronikiem nie był, niemniej wyraził skłonność wielu elektroników konstruktorów i decydentów mówiąc:

Proszę pana, ja jestem umysł ścisły. Mnie się podobają melodie, które już raz słyszałem. Po prostu. No... To... Poprzez... No reminiscencję. No jakże może podobać mi się piosenka, którą pierwszy raz słyszę.

Tak, lubimy to, co znamy. Często trzymamy się utartych ścieżek i szablonów, które nie wiadomo jak i kiedy uformowały się, a następnie utrwaliły w naszych głowach. Z jednej strony to dobrze, bo wcześniejsze doświadczenia procentują w kolejnych rozwiązaniach. Ale może jednak warto otworzyć się na nieszablonowe, nowocześniejsze rozwiązanie, które okaże się znacząco lepsze? Dotyczy to także wykorzystania elementów indukcyjnych. Często decyzja nie jest łatwa, bo w grę wchodzi szereg czynników. W każdym razie warto dbać, by stale mieć otwarty umysł, by nadążać nie tylko za zmianami sytuacji ekonomicznej, ale też nadążać za postępem.

Wprawdzie postęp w dziedzinie samych elementów indukcyjnych jest nieporównanie mniejszy i powolniejszy, niż na przykład w dziedzinie półprzewodników, niemniej i tu daje się wyraźnie zauważyć.

Trzeba też rozumieć, na czym ten postęp polega. Otóż także w dziedzinie elementów indukcyjnych podstawowe zasady nie mogą się zmienić, a zasadniczo postęp polega na stosowaniu materiałów ferromagnetycznych o coraz lepszych właściwościach. Lepszych, ale można powiedzieć: tylko trochę lepszych. Ogromny postęp w zakresie miniaturyzacji przetwornic i zasilaczy impulsowych wcale nie wynika ze stosowania elementów indukcyjnych o parametrach wielokrotnie lepszych niż dawniej.

Nie! Ich parametry są „tylko trochę lepsze”, a postęp wynika głównie z ogromnego wzrostu częstotliwości pracy, co jest zasługą przede wszystkim półprzewodnikowych kluczy – przełączników (krzemowych tranzystorów MOSFET, a ostatnio coraz częściej tranzystorów z węgla krzemu SiC i azotku galu GaN), ich sterowników oraz coraz lepszych rozwiązań układowych (zwłaszcza przetwornic rezonansowych).

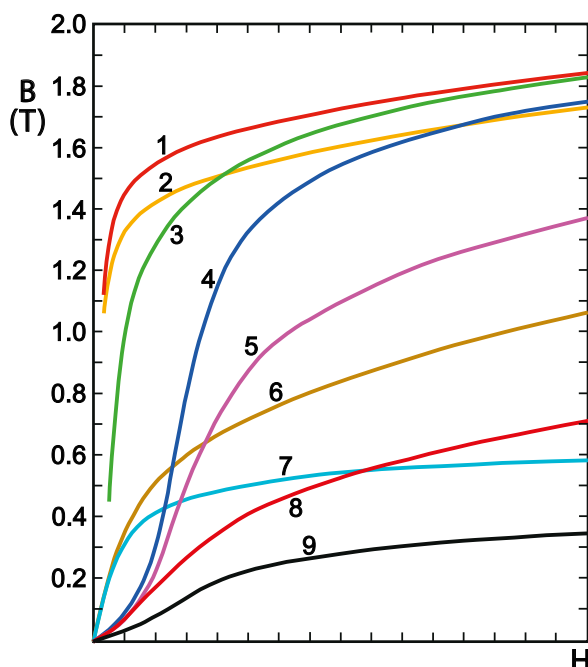
Nie zawsze sytuacja jest jasna i oczywista. Zwiększanie częstotliwości pracy znakomicie zwiększa gęstość mocy (moc przypadająca na jednostkę masy lub objętości przetwornicy), ale jednocześnie zwiększa problemy z kompatybilnością elektromagnetyczną EMC.

W sprzęcie powszechnego użytku oprócz cech użytkowych i atrakcyjności dla klienta, podstawową sprawą jest obniżanie kosztów, co na przykład dziś oznacza odejście od klasycznych transformatorów na rzecz zasilaczy impulsowych – przetwornic. Klasyczne, ciężkie transformatory raz na zawsze znikły ze sprzętu powszechnego użytku.

Słabo zorientowanym wydaje się, że przetwornice impulsowe, które dziś już są tańsze od zasilaczy z klasycznymi transformatorkami, mają same zalety. Tak nie jest i to z kilku powodów. Spektakularnym przykładem są najtańsze chińskie moduły: przetwornice, impulsowe zasilacze sieciowe, a także impulsowe wzmacniacze audio (różne odmiany klasy D). Ich najważniejszą, czasem jedyną zaletą jest niska cena, natomiast często nie spełniają one żadnych wymagań dotyczących kompatybilności elektromagnetycznej EMC oraz bezpieczeństwa (nie mają też znaku CE).

Celem tego artykułu nie jest wyjaśnianie szerokiego i złożonego problemu EMC oraz kwestii oddzielenia galwanicznego i bezpieczeństwa, ani ze strony przepisów i norm, ani ze strony realnych kłopotów z tym związanych. Celem artykułu nie jest też tłumaczenie dwuznaczności powiedzenia: *lepsze jest wrogiem dobrego*. W każdym razie różne skądinąd nowoczesne układy impulsowe mogą sprawiać wiele kłopotów, dlatego w zastosowaniach innych niż urządzenia masowej produkcji jak najbardziej nadal jest miejsce na klasyczne transformatory sieciowe. Nadal jest mnóstwo zastosowań, gdzie redukcja masy i objętości nie ma większego (lub żadnego) znaczenia, a liczy się niezawodność. W szeregu zastosowań, jak na przykład w urządzeniach medycznych, aparaturze pomiarowej, bardzo ważna jest też minimalizacja poziomu zakłóceń. Wtedy zastosowanie starych dobrych klasycznych transformatorów sieciowych jest niemal koniecznością.

Ponieważ Czytelnikami tego materiału są osoby o różnym poziomie wiedzy na tematy związane z elementami indukcyjnymi,



Rysunek 2. Krzywe magnesowania pierwotnego różnych materiałów: kilku odmian stali (1, 2, 3, 4, 5), lanego żelaza (6), niklu (7), kobaltu (8) i magnezytu (9)

a "cewki są zmorą elektroników", więc należałoby przypomnieć podstawowe informacje w dużym skrócie i w sposób, który może zaskoczy fachowców.

Niewielki tyk historii i podstaw

Cewka lub uzwojenie transformatora to najczęściej pewna liczba zwojów drutu. Najmniej zorientowanym potrzebna jest odpowiedź na podstawowe pytanie: po co w elementach indukcyjnych stosować rdzenie, które sprawiają tyle kłopotów?

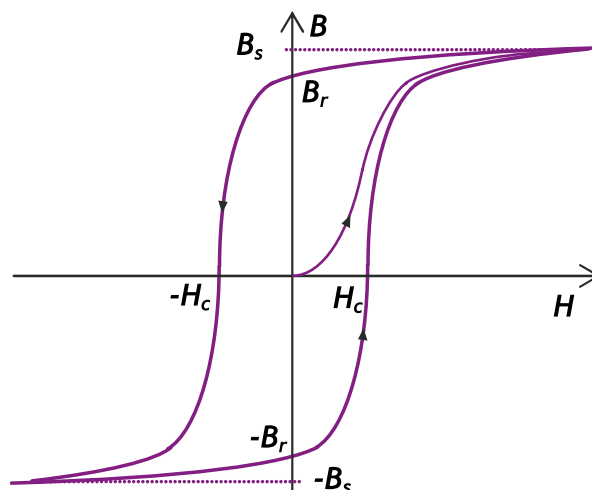
Bodaj najprostsza odpowiedź brzmi: w sumie chodzi przede wszystkim o zmniejszenie strat i częściowo o możliwość miniaturyzacji.

Otóż dodanie rdzenia ferromagnetycznego pozwala uzyskać dużą indukcyjność przy małej liczbie zwojów. Gdybyśmy chcieli zbudować cewkę powietrzną o znaczącej indukcyjności, bez rdzenia, to musiałaby ona mieć mnóstwo, powiedzmy tysiące zwojów. Gdyby ją nawinąć cieniutkim drucikiem, by cewka miała niedużą masę, ale miałaby nieakceptowalnie wielką rezystancję. A to przy przepływie prądu powodowałoby nieakceptowalnie duże straty ($P=I^2R$). Gdyby z kolei taka cewka miała mieć niedużą rezystancję, musiałaby być nawinięta bardzo grubym drutem, co z kolei dałoby nieakceptowalne rozmiary, masę i cenę.

W przypadku transformatorów można podać dodatkowe następujące wyjaśnienie, przedstawiające sprawę z innego punktu widzenia: w uzwojeniu wtórnym indukuje się napięcie (i może płynąć prąd) zgodnie z prawem Faradaya, które mówi, iż indukowane napięcie (siła elektromotoryczna indukcji) jest wprost proporcjonalne do *szybkości zmian* strumienia indukcji pola magnetycznego, co można wyrazić wzorem:

$$\mathcal{E} = - \frac{d\Phi_B}{dt}$$

Najprościej biorąc, aby uzyskać duże *zmiany* strumienia, pożądana jest też jak największa *wartość* tego strumienia magnetycznego (Φ) obejmującego uzwojenie wtórne. Tymczasem ten strumień magnetyczny jest wytwarzany przez prąd (magnesujący) płynący w uzwojeniu pierwotnym. Dodanie rdzenia ferromagnetycznego powoduje, że przy tym samym prądzie pierwotnym (magnesującym) uzyskujemy, i to niejako za darmo, dużo silniejszy strumień magnetyczny. Można powiedzieć, że obecność rdzenia za darmo zwiększa strumień



Rysunek 3. Wykres magnesowania materiału ferromagnetycznego

magnetyczny (i indukcję B, czyli gęstość tego strumienia). Najogólniej biorąc, stopień tego zwiększania strumienia określa parametr zwany przenikalnością magnetyczną (μ). Najczęściej mówimy o *przenikalności względnej*, pokazującej ile razy przenikalność danego materiału jest większa od przenikalności magnetycznej próżni.

Nie ulega wątpliwości, że pożądaną są rdzenie o jak największej przenikalności magnetycznej. Problem w tym, że niewiele substancji (materiałów) ma właściwości ferromagnetyczne, wzmacniające pole magnetyczne. Podstawowe to popularne pierwiastki: żelazo, nikiel, a także kobalt, metale ziem rzadkich oraz liczne związki tych i innych pierwiastków (znane są też nieliczne ferromagnetyki organiczne).

Co ciekawe, bardzo dużą przenikalność magnetyczną, rzędu kilku do kilkunastu tysięcy ma... najzwyklejsze, popularne i bardzo tanie żelazo. Czyste żelazo – pierwiastek chemiczny o symbolu Fe, a także jego związki i stopy z innymi pierwiastkami. Dlatego też najpierw, w XIX wieku żelazo, a potem blachy stalowe były pierwszymi materiałami, pełniącymi rolę rdzeni.

Oprócz przenikalności, drugą ważną kwestią jest problem nasycenia rdzenia. Pole magnetyczne często wyobrażamy sobie i przedstawiamy sobie graficznie jako mniej lub bardziej liczne zamknięte linie sił pola. Mówimy że strumień magnetyczny Φ to liczba tych linii, a ich gęstość w materiale rdzenia to indukcja magnetyczna B. Problem w tym, że materiały ferromagnetyczne spełniają swoją rolę *wzmacniania pola magnetycznego*, czyli zachowują dużą wartość przenikalności μ tylko do określonej gęstości strumienia, czyli do jakiejś określonej maksymalnej wartości indukcji B (B_{SAT}). Ta maksymalna wartość indukcji B, zwana *indukcją nasycenia*, jest różna dla poszczególnych materiałów magnetycznych. Aby pracować z silnym polem magnetycznym (o dużej liczbie linii sił pola), konieczne jest zastosowanie rdzenia o na tyle dużym przekroju, żeby gęstość strumienia (indukcja B) nie była większa od indukcji nasycenia dopuszczalnej dla danego materiału rdzenia. Dla danego strumienia, rdzeń o większej indukcji nasycenia może być po prostu mniejszy.

Co ciekawe, indukcja nasycenia jest duża, jedna z największych, dla popularnego żelaza, ponad 1,5 tesli, natomiast dla rdzeni ferrytowych jest dużo mniejsza, rzędu 0,2...0,4 tesli. **Rysunek 2** pokazuje krzywe magnesowania pierwotnego różnych materiałów, w tym kilku odmian stali (1, 2, 3, 4, 5), lanego żelaza (6) a także niklu (7), kobaltu (8) i magnezytu (9). Niezaznaczone tu krzywe magnesowania ferrytów przebiegałyby podobnie, jak trzy najniższe krzywe (7...9).

Dlaczego więc rdzenie elementów magnetycznych nie są wykonywane z czystego żelaza i stali? Przecież żelazo ma i bardzo dużą przenikalność, i dużą indukcję nasycenia!

Przyczyną są straty, polegające na wydzielaniu się ciepła w rdzeniu. Straty są tym większe, im większa jest indukcja, większa częstotliwość pracy, a ponadto trochę rosną z upływem czasu (starzenie).

Chodzi między innymi o straty w rdzeniu, wywołane przepływem tak zwanych prądów wirowych (*eddy currents*). Problem w tym,

że obecność zmiennego pola magnetycznego powoduje indukowanie napięć nie tylko w uzwojeniach, ale we wszystkich materiałach przewodzących, także w żelaznym rdzeniu, co oznacza właśnie przepływ w rdzeniu prądów wirowych. Te prądy płynąc przez rezystancję materiału rdzenia powodują straty w postaci wydzielanego ciepła ($P=I^2R$).

Najprościej biorąc, dodatek do żelaza krzemu (blachy żelazokrzemowe) zwiększa rezystywność materiału rdzenia, co zmniejsza prądy wirowe i straty. W praktyce stosowane są blachy żelazokrzemowe o zawartości krzemu 1,5...3%. Większa zawartość Si, rzędu 7%, zmniejsza straty jeszcze bardziej, ale materiał staje się bardzo kruchy i praktycznie nie nadaje się do obróbki mechanicznej.

Dodatkowo prądy wirowe są mniejsze, jeśli rdzeń ma postać jak najcieńszych blaszek izolowanych nieprzewodzącym lakierem – wtedy prądy wirowe zostają ograniczone do małych pętli, zawierających się w poszczególnych blaszkach.

Niestety dodanie do żelaza krzemu i podzielenie rdzenia na najcieńsze nawet blaszki nie rozwiązuje problemu strat do końca. Trzeba pamiętać o drugim głównym rodzaju strat. Otóż podczas pracy, w rdzeniach występuje zmienne pole magnetyczne, a to oznacza większe i mniejsze chwilowe namagnesowanie, a nawet przemagnesowanie rdzenia w przeciwnym kierunku. Wszystkie materiały ferromagnetyczne mają pewną cechę, która w większości przypadków jest ułomnością i przyczyną problemu, a pożądana jest tylko w magnesach trwałych. Chodzi o histerezę, czyli swego rodzaju pamięć namagnesowania, przez co graficznym przedstawieniem wykresu namagnesowania materiału ferromagnetycznego nie jest ani linia prosta, ani linia łamana (w dwóch miejscach). Wykresem tym jest pętla, określona przez indukcję nasycenia oraz tak zwaną pozostałość magnetyczną i koercję. Przykład na **rysunku 3**.

Poszczególne materiały ferromagnetyczne mają bardzo różne pętle histerezy.

Podczas przemagnesowywania rdzenia histereza magnetyczna jest przyczyną tak zwanych *strat histerezowych*. Najprościej biorąc, czym szersza pętla histerezy i czym większa częstotliwość, tym większe straty histerezowe. I tu wracamy do praktyki. Nie zapominajmy, że kluczową kwestią jest wielkość strat, zarówno w uzwojeniu, co nazywa się *stratami w miedzi*, jak i w rdzeniu, co do dziś niezbyt trafnie nazywa się *stratami w żelazie*. Od dawna prostą, jak najbardziej sensowną regułą było takie projektowanie elementów magnetycznych, żeby straty ogólne były akceptowalne dla danego zastosowania i żeby *straty w miedzi* były w przybliżeniu równe *stratom w żelazie*.

W świadomości wielu elektroników zakorzenione jest (nieprawdziwe dziś) wyobrażenie, iż mamy dwie główne grupy materiałów magnetycznych. Jedna grupa to rozmaite blachy, w których podstawowymi składnikami jest żelazo i nikiel. Druga grupa to ferryty, które najprościej biorąc są ceramicznymi spiekami różnych tlenków metali, przede wszystkim żelaza i niklu oraz dodatków innych tlenków, m.in. manganu, boru, cynku, itru, ...

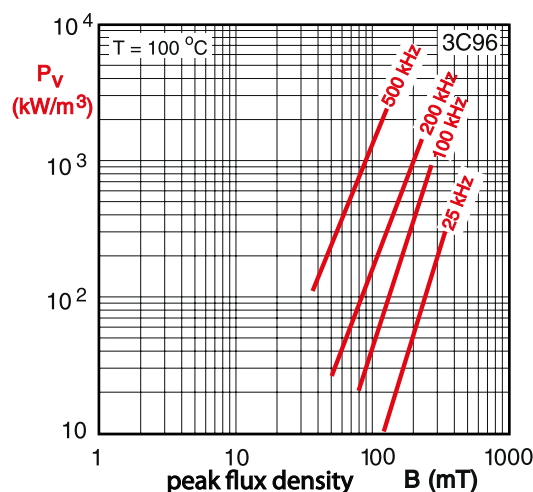
Rdzenie z blach stosowane były od ponad stu lat w elementach indukcyjnych, pracujących przy częstotliwościach niskich, głównie częstotliwości sieci energetycznej 50 Hz (60 Hz). Przy wyższych częstotliwościach, już od jednego kiloherca wzwyż, stosowano ferryty.

A czy transformatora sieciowego 50 Hz nie można zrealizować na rdzeniu ferrytowym?

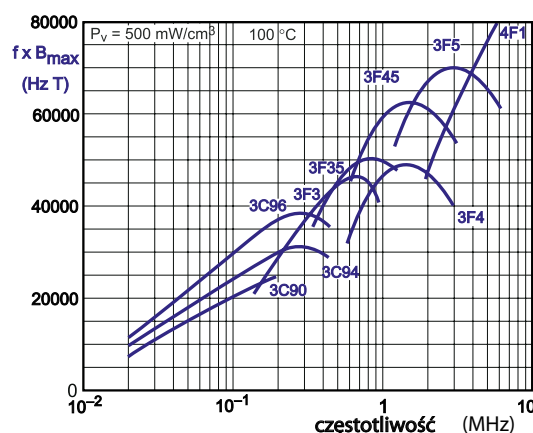
Otóż jak najbardziej można! Tylko najprościej mówiąc, z uwagi na niedużą przenikalność i niedużą indukcję nasycenia, taki ferrytowy transformator 50 Hz musiałby mieć o wiele więcej zwojów, co niekorzystnie zwiększyłoby straty w miedzi. Lepszy i tańszy jest transformator 50 Hz z rdzeniem z tanich blach stalowych – żelazokrzemowych.

A czy można zbudować dławik w.c.z. albo transformator do zasilacza impulsowego z rdzeniem z blach żelazokrzemowych?

Teoretycznie tak. Ale z tym byłby większy kłopot. Wprawdzie indukcja nasycenia żelaza jest duża, ale nie da się jej wykorzystać, bowiem ze wzrostem częstotliwości i wraz ze wzrostem indukcji



Rysunek 4. Wykres zależności strat od częstotliwości i szczytowego namagnesowania (indukcji) dla pewnego ferrytu

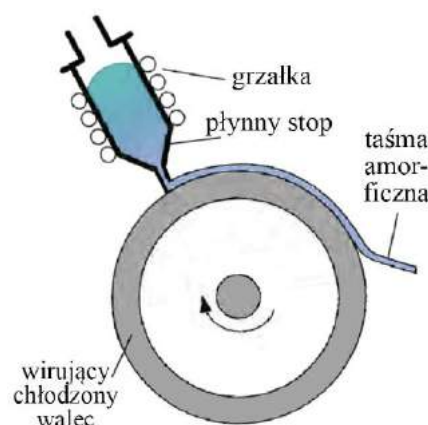


Rysunek 5. Przebieg współczynnika jakości (performance factor) w funkcji częstotliwości dla różnych materiałów

ogromnie rosną straty w rdzeniu. Blachy żelazokrzemowe po prostu nie nadają się do pracy przy wysokich częstotliwościach.

Podobnie jest z różnymi rodzajami ferrytów i innych materiałów: **straty zawsze rosną ze wzrostem częstotliwości**. Każdy materiał ferromagnetyczny ma określone straty przy danej częstotliwości i określonej indukcji. **Rysunek 4** pokazuje zależność strat od częstotliwości i szczytowego namagnesowania (indukcji) dla pewnego ferrytu.

Z kolei **rysunek 5** pokazuje przebieg tak zwanego współczynnika jakości (performance factor) w funkcji częstotliwości dla różnych materiałów. Materiały, które przy wyższych częstotliwościach mają mniejsze straty, mają też gorszą, mniejszą przenikalność i zwykle



Rysunek 6. Uproszczony rysunek procesu otrzymywania metalicznego szkła

też mniejszą indukcję nasycenia. Jednak stosujemy je przy wysokich częstotliwościach właśnie z uwagi na mniejsze straty w rdzeniu. I właśnie wypadkowa wielkość strat jest czynnikiem decydującym o wyborze materiału rdzenia do danego zastosowania.

Dziś często materiał magnetyczny pracuje przy tak wysokich częstotliwościach, że właśnie z uwagi na rosnące straty dobieramy indukcję w rdzeniu znacznie mniejszą, niż indukcja nasycenia. Praca przy maksymalnej dopuszczalnej indukcji doprowadziłaby do przegrzania rdzenia i uzwojeń, a nawet osiągnięcia przez rdzeń tak zwanego punktu Curie czyli temperatury, w której straci on właściwości ferromagnetyczne (temperatury Neela w ferrytach).

Te rozważania dotyczą przede wszystkim transformatorów i dławików przekazujących i magazynujących energię.

Jeszcze inne aspekty trzeba uwzględnić przy projektowaniu elementów magnetycznych o innym przeznaczeniu. Dziś coraz rzadziej wykorzystujemy cewki w obwodach rezonansowych i filtrujących, gdzie kluczowymi czynnikami są stabilność i liniowość. Coraz rzadziej materiały magnetyczne pełnią rolę ekranów ochronnych przed przenikaniem pola magnetycznego. Ale coraz częściej wykorzystujemy, wręcz musimy wykorzystywać elementy indukcyjne służące do szerokopasmowej filtracji zakłóceń. Wtedy kwestia stratności nabiera innego znaczenia. Nikną jedne problemy, a nabierają znaczenia inne, na przykład sprawa nieuniknionych pojemności i wynikającego stąd realnego zakresu częstotliwości pracy. Dlatego do poszczególnych zastosowań wykorzystujemy rdzenie z innych materiałów.

Fascynująca historia ferromagnetyków

Jak wspomnieliśmy, wielu elektroników ma zakorzenione wyobrażenie, że pierwsze historycznie rodzaje materiałów magnetycznych to po pierwsze żelazo i stal żelazokrzemowa, a po drugie ferryty, natomiast inne materiały, jak rdzenie amorficzne, proszkowe i nanokrystaliczne to wynalazek ostatnich lat. Po części jest to prawda, ale błędne wyobrażenia dotyczą ferrytów i rdzeni proszkowych.

Systematyczne badania zjawisk i materiałów magnetycznych podjęto głównie we Francji i Anglii w latach 20. XIX wieku, czyli niedługo po wojnach napoleońskich i pokoju wiedeńskim.

Wskaźnikiem pola magnetycznego i przepływu prądu (stałego) była wtedy igła kompasu magnetycznego. Szybko zauważono, iż zwyczajne żelazo silnie wzmacnia, ale też ekranuje pole magnetyczne. Później przy eksperymentach z prądami zmiennymi okazało się, że w silnym zmiennym polu magnetycznym żelazo mocno się grzeje i nie spełnia oczekiwanej roli. Sporo ponad sto lat temu wynaleziono blachę żelazokrzemową, gdzie dodatek krzemu i podział na elektrycznie izolowane listki pozwoliły znacząco zmniejszyć straty.

Jeszcze w XIX wieku zauważono, że właściwości magnetyczne materiałów zależą też od obróbki mechanicznej (walcowanie) oraz termicznej (wyżarzanie). Powstały liczne odmiany blach transformatorowych (żelazo i do kilku procent krzemu), różniące się nie tylko składem chemicznym, ale też sposobem obróbki.

Odrębna sprawa to materiały magnetyczne o innym składzie i przeznaczeniu. Bodaj najbardziej znany jest tak zwany permalój (Permalloy), zawierający głównie nikiel i żelazo. Nie jest to materiał nowy – opatentowano go sto lat temu!

Permalój to nie jeden konkretny materiał – tym mianem określa się różne stopy, których kluczowym składnikiem jest nikiel. Ponad sto lat temu badano magnetyczne właściwości różnych materiałów, w tym stopów niklu i niklu z żelazem (także żelaza i kobaltu). Efektem są między innymi różne stopy niklu i żelaza o dużej zawartości niklu 35% do 95%, wszystkie znane jako permalój. Są to materiały magnetycznie najbardziej miękkie, o najmniejszej koercji, mające ogromną przenikalność, nawet rzędu miliona (supermalój).

Permaloje są dużo droższe od blachy żelazokrzemowej i nie są stosowane na rdzenie transformatorów sieciowych, ale znajdują zastosowanie m.in. do realizacji wysokiej klasy transformatorów audio,

gdzie wymagane jest szerokie pasmo i jak najmniejsze zniekształcenia. Zastosowanie rdzenia o dużej przenikalności pozwala uzyskać potrzebną dużą indukcyjność przy niewielkiej liczbie zwojów, co oznacza mniejsze szkodliwe pojemności i wyższe częstotliwości rezonansów własnych. Kwestia pojemności i użytecznego zakresu częstotliwości pracy jest też istotna w filtrach przeciwzakłóceniovych. To są jednak odrębne, obszerne zagadnienia.

Inną nazwą materiału magnetycznego, znaną prawie każdemu elektronikowi jest mumetal (MuMetal, μ metal). To też nie jest jeden jedyny stop o ściśle określonym składzie, tylko stopy o zawartości niklu rzędu 75...80%, a reszta to żelazo, około 15%, molibden i ewentualnie miedź. Mumetal jest więc też odmianą permaloju. Mumetal jest stosowany głównie jako skuteczny materiał ekranujący, znacząco ograniczający wpływ pól magnetycznych, także tych 50 Hz. Wydawałoby się, że następne historycznie były ferryty.

Nie!

Ferryty, a konkretnie tak zwane ferryty miękkie, w Europie pojawiły się po zakończeniu II wojny światowej, czyli po roku 1945! Tu część elektroników może zaprotestować, że przecież rdzenie ferrytowe stosowano w przedwojennych radiach, co można łatwo sprawdzić...

Rzeczywiście, w przedwojennych radioodbiornikach stosowano rdzenie niezłazne, ale nie były to rdzenie ferrytowe. Były to rdzenie, które należałoby nazwać proszkowymi. To właśnie może być zaskoczeniem dla wielu elektroników, którzy rdzenie proszkowe kojarzą z nowoczesnymi cewkami i dławikami.

Problem w tym, że powszechnie dziś znane ferryty są tak zwanymi ferrimagnetykami (nie ferromagnetykami). Pierwsze ferryty opracowano w Japonii w roku 1930 (Yogoro Kato i Takeshi Takei na politechnice tokijskiej), a potem od roku 1935 ich produkcją zajął się znany do dziś koncern TDK. Natomiast w Europie nad ferrytami pracował holenderski Philips i to właśnie pracujący dla Philipsa J. L. Snoek opracował pierwsze komercyjne ferryty miękkie dopiero w roku 1945. Kilka lat później teoretyczne wyjaśnienie właściwości ferrimagnetyków (i antyferromagnetyków) przedstawił L. Neel.

Ferryty to ceramiczne spieki tlenków. Ogólny wzór chemiczny ferrytów to $M(Fe_xO_y)$, gdzie M to dodatkowy pierwiastek wyznaczający właściwości, taki jak na przykład magnez, mangan, nikiel, cynk, miedź, kobalt, aluminium, bar... Najpopularniejsze są ferryty manganowo-cynkowe o wzorze ogólnym $Mn_{(1-x)}Zn_xFe_2O_4$ oraz mające mniejsze straty przy wysokich częstotliwościach ferryty niklowo-cynkowe $Ni_{(1-x)}Zn_xFe_2O_4$. Opracowano szereg ferrytów o różnym składzie i przeznaczeniu, w tym magnesy trwałe oraz ferryty miękkie na rdzenie, dla różnych zakresów częstotliwości pracy, co sygnalizuje rysunek 5.

No tak, ale jeśli ferryty pojawiły się po II wojnie światowej, to jakie rdzenie stosowano w sprzęcie przedwojennym?

Otóż w radioodbiornikach w obwodach niskiej częstotliwości stosowano transformatory i dławiki z rdzeniami z blach żelazokrzemowych, a w obwodach w.c.z. stosowano rdzenie, które były spiekami różnych materiałów, głównie żelaza. Warto też pamiętać, że na początku ery radia wykorzystywano tylko fale długie i średnie, ponieżej 1 megaherca, a upowszechnienie stacji na falach krótkich wiązało się też z wytworzeniem elementów mogących pracować przy częstotliwościach kilku, potem kilkunastu megaherców. Przy wyższych częstotliwościach kluczowym problemem materiałów magnetycznych jest dobra przewodność elektryczna żelaza i prądy wirowe. Aby zminimalizować problem, wymyślono i opracowano różne odmiany tak zwanych rdzeni proszkowych. Rdzenie proszkowe upowszechniły się od lat 20., czyli od początku rozwoju radiofonii.

Nie będziemy tu opisywać, czym jest żelazo karbonyłowe i jaki ma związek z tlenkiem węgla. Najprościej biorąc, w omawianych rdzeniach stosowano żelazny proszek, a więc drobinki żelaza, sklejone odpowiednim lepiszczem (pierwotnie szelakiem), które izoluje te drobinki od siebie, zmniejszając problem prądów wirowych,

wprowadzając tak zwaną szczelinę rozproszoną i zwiększając zdolność magazynowania energii. W ten prosty i sprytny sposób wykorzystuje się pozytywne właściwości żelaza, w tym dużą indukcję nasycenia i niedużą histerezę. W pierwszej połowie XX wieku takie rdzenie proszkowe poszczególni wytwórcy często opracowywali i realizowali metodami doświadczałnymi (prób i błędów). Stosowali różne materiały podstawowe i różne dodatki, z reguły nie opierając się na opracowaniach naukowych, tylko na eksperymentach i wcześniejszych doświadczeniach. Dziś popularne są różne odmiany rdzeni proszkowych o składzie i procesie technologicznym starannie zbadanym naukowo.

W każdym razie rdzenie proszkowe pojawiły się ponad 20 lat wcześniej, niż ferrytowe. A wynalazek ferrytów znakomicie rozszerzył zakres częstotliwości, w których mogły pracować materiały magnetyczne. Jednak ferrytom pod wieloma względami też daleko do ideału.

Innym zaskakującym wynalazkiem są tak zwane *szkła metaliczne*. Generalnie materiały magnetyczne (metale) mają strukturę krystaliczną i właśnie rodzaj i uporządkowanie struktury krystalicznej decyduje o właściwościach ferromagnetycznych. Na początku lat 50. odkryto, że zaskakująco dobre właściwości ferromagnetyczne mogą mieć materiały, które nie mają struktury krystalicznej, tylko są bezpostaciowe – *amorficzne*. Okazało się między innymi, iż niektóre materiały ferromagnetyczne zawierające metale, po stopieniu w odpowiednio wysokiej temperaturze i po błyskawicznym schłodzeniu w określonych specyficznych warunkach, nie odzyskują struktury krystalicznej po powrocie do stałej postaci. Można powiedzieć, że powstały stop metali przestaje być krystaliczny, ma budowę amorficzną, bezpostaciową i jest jakby zastygłą cieczą, podobnie jak szkło. Co ciekawe, dobierając zarówno skład, jak i warunki procesu takiej zamiany metali w szkło można uzyskać materiał o bardzo dobrych właściwościach magnetycznych w tym stosunkowo dużej indukcji nasycenia i małych stratach.

Jednym ze sposobów otrzymywania takich metalicznych szkieł jest natryskiwanie cieniutkiej warstewki ciekłego stopu na wirujący chłodzony bęben, często w atmosferze jakiegoś obojętnego gazu (rysunek 6).

Na powierzchni bębna cieniutka warstewka metalu szybko tężeje, ale nie nabiera struktury krystalicznej. Otrzymanym produktem jest cieniutka taśma szkła metalicznego, która może być zmielona na proszek o różnej wielkości ziaren, co też wpływa na właściwości magnetyczne. Z takiego metalo-szklanego proszku formowane są rdzenie o potrzebnych kształtach, często w kształcie torusa.

Kolejną historycznie propozycją są materiały *nanokrystaliczne*, o znikomych rozmiarach kryształów do 100 nm. Pomysł na ich realizację pojawił się pod koniec lat 50. Ale zaczęto je opracowywać i wprowadzać na rynek dopiero pod koniec lat 80. XX wieku. Dziś znane są różne sposoby uzyskiwania materiałów

nanokrystalicznych. Jednym z nich jest poddanie amorficznych szkieł metalicznych kontrolowanej obróbce termomagnetycznej. Odpowiednie podgrzewanie powoduje zmianę struktury z amorficznej właśnie na nanokrystaliczną. Co ciekawe, jeśli odbywa się to w obecności odpowiednio dobranego pola magnetycznego, z tego samego wyjściowego materiału amorficznego można uzyskać różne właściwości, w tym różne kształty charakterystyki magnesowania powstałego materiału nanokrystalicznego.

Rdzenie proszkowe, amorficzne i nanokrystaliczne mogą pracować przy stosunkowo wysokich częstotliwościach, ale ze względu na rosnące straty, przy częstotliwościach rzędu megaherców nadal stosuje się odpowiednie rdzenie ferrytowe.

Zagadnienia związane z uzyskiwaniem materiałów magnetycznych: blach, tasm, włókien, rdzeni proszkowych, amorficznych metalicznych szkieł (*metal glass*) oraz materiałów nanokrystalicznych są fascynujące. W literaturze można znaleźć szereg interesujących informacji, niemniej często szczegóły nie są publikowane. Problem jest też z rdzeniami proszkowymi. Określenie rdzeń proszkowy, pierwotnie dotyczące sproszkowanego czystego żelaza, jak najbardziej może dziś dotyczyć krystalicznych proszków z różnych stopów żelaza, ale też sproszkowanego metalicznego szkła i proszków nanokrystalicznych. Nazwy i określenia handlowe oraz materiały reklamowe niewiele mówią, a wręcz mogą mylić. Często nie znamy szczegółów składu, struktury i sposobu wytwarzania.

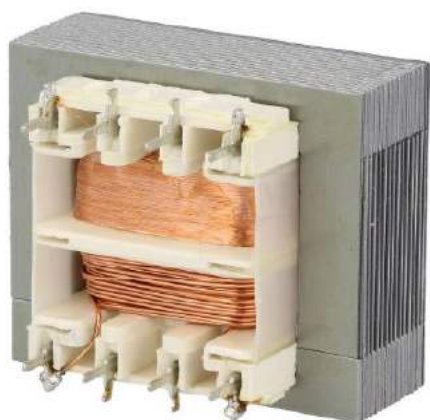
Oczywiście z punktu widzenia konstruktora kluczowa jest znajomość parametrów – karty katalogowe z reguły są dostępne. Jednak **dane z kart katalogowych trzeba prawidłowo zinterpretować**. I nie tylko zinterpretować, ale też porównać z innymi materiałami. A dla poszczególnych zastosowań inne są najistotniejsze parametry. Praktyka pokazuje, iż użycie takich czy innych materiałów w danym zastosowaniu i przy danych założeniach i warunkach pracy warto skonsultować z doświadczonym praktykiem. Podobnie użycie gotowych elementów z różnymi dostępnymi rdzeniami.

W każdym razie mamy na rynku coraz więcej nowoczesnych materiałów. Nie znaczy to oczywiście, że nowoczesne rdzenie proszkowe i ze szkieł metalicznych są bliskie ideału i że nie mają wad wcześniejszych materiałów. Niestety, one też mają swoje wady i ograniczenia, niemniej są coraz częściej stosowane z uwagi na znacząco lepsze parametry i coraz niższe ceny.

I to właśnie obniżka cen decyduje o wzroście popularności takich nowych, ulepszonych materiałów.

Postęp w dziedzinie materiałów magnetycznych na pewno nie jest tak spektakularny, jak w dziedzinie układów scalonych gdzie wciąż obowiązuje wykładnicze prawo Moore'a, ale wciąż wprowadzane są magnetyczne nowości i rozwiązania niemożliwe do zrealizowania jeszcze kilkanaście lat temu. Rozwiązania niekoniecznie wynikające z parametrów materiałów magnetycznych, tylko z ogólnego rozwoju elektroniki. Charakterystycznym przykładem są cewki i transformatory *planarne*, które pomału stają się coraz popularniejsze, ale wokół których narosło już sporo niejasności.

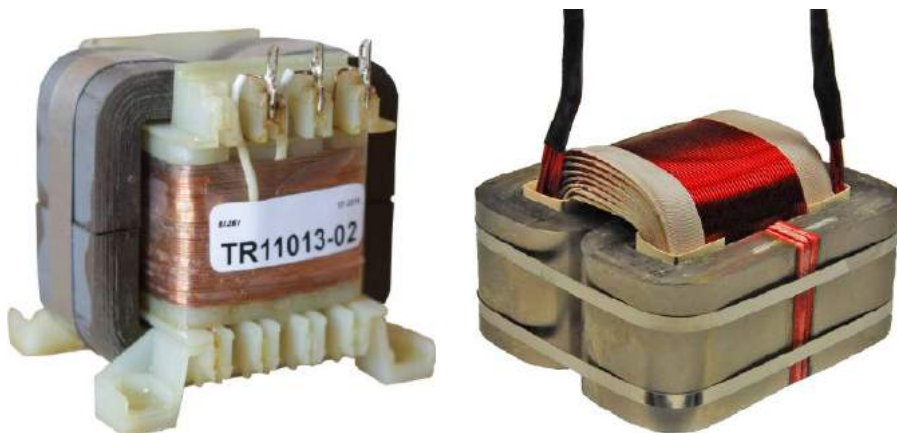
Zanim jednak przejdziemy do transformatorów planarnych przypomnijmy, że dawniej do realizacji transformatorów i dławików m.c.z. wykorzystywano *rdzenie kształtkowe* (fotografia 7a). Najpierw bez kłopotu nawijano na karkasie uzwojenie(-a), potem pracowicie (i nie bez kłopotów) składano rdzeń z pojedynczych blaszanych kształtek EI lub M. Takie rozwiązanie było stosunkowo nieskomplikowane technologicznie, jednak miało też istotne wady. Transformatory o takiej konstrukcji są stale dostępne, zwłaszcza w wersjach zalewanych tworzywem sztucznym (fotografia 7b).



Fotografia 7a. Transformator na rdzeniu kształtkowym



Fotografia 7b. Transformatory zalewane



Fotografia 8. Transformatory na rdzeniach zwijanych ciętych

Lepsze, bliższe ideału są rozwiązania, gdzie rdzeń ma kształt toroidu, a najlepiej torusa. Jednym z kluczowych problemów staje się wtedy nawijanie uzwojeń. Kilka czy kilkanaście zwojów można na torusie/toroidzie nawinąć bez większych kłopotów, gorzej, gdy zwojów ma być wiele.

Rdzenie w kształcie pierścionków dość szybko pojawiły się w aparaturze w.cz. Także do produkcji transformatorów sieciowych zaczęto stosować rdzenie fabrycznie zwijane z taśm. Najpierw były to rdzenie po zwinięciu przecinane na dwie połówki, które potem po zmontowaniu na karkasie z uzwojeniami były ściskane metalowymi obejmami. Chodzi o tak zwane *rdzenie zwijane cięte* (RZC, CP, SU) – przykład na fotografii 8.

Przecinanie rdzenia jest niekorzystne, dlatego kolejnym krokiem było zastosowanie *rdzeni zwijanych z taśm*, rdzeni które nie były przecinane, tylko miały kształt toroidalny o przekroju prostokątnym (fotografia 9a, b). Poważnym problemem było to, że na takim gotowym toroidalnym rdzeniu bez jego przecinania trzeba było nawinąć wiele zwojów drutu. Dopiero gdy opracowano maszyny, które sprytnie potrafiły nawijać takie uzwojenia, sieciowe transformatory toroidalne, potocznie nazywane toroidami, zaczęły wypierać z rynku wcześniejsze rozwiązania.

Z pewnych względów korzystniej jest realizować transformatory toroidalne większej mocy, natomiast najmniejsze transformatory

sieciowe nadal często zawierają klasyczne rdzenie z kształtkami EI. Generalnie nowocześniejsze transformatory toroidalne projektowane są powiedzmy: oszczędnie według zasady *mało żelaza, mało miedzi*, co pod pewnymi względami oznacza pracę na granicy dopuszczalnych parametrów i odbija się na praktycznych właściwościach: prawie każdy wie, że większe transformatory toroidalne w chwili dołączania do sieci mogą powodować wybite domowych bezpieczników. Związane jest to z nasyceniem rdzenia i niekorzystnym zbiegiem okoliczności (pozostałość magnetyczna w rdzeniu i dołączenie do sieci blisko momentu przejścia przez zero). To jest wada, powodująca

konieczność stosowania tzw. *softstartów* przy włączaniu takich większych transformatorów. Sieciowe transformatory toroidalne mają niewielką rezystancję uzwojeń, czyli jak mówimy są bardzo sztywne i mogą dostarczać przez niedługi okres moc dużo większą, niż katalogowa moc nominalna.

To wiedzą prawie wszyscy. Mniej powszechna jest wiedza o najmniejszych transformatorach, zawierających klasyczne rdzenie EI. Z uwagi na mały przekrój rdzenia muszą one zawierać wiele zwojów, co z jednej strony jest niekorzystne, bo zwiększa rezystancję, zwłaszcza uzwojenia pierwotnego, przez co napięcie wyjściowe silnie zależy od prądu obciążenia (mówimy, że takie transformatory są miękkie). Jednak w pewnych zastosowaniach takie właściwości są zaletą i celowo projektowane są transformatory bardzo miękkie, o tak wielkiej rezystancji uzwojeń, że nawet przy awarii i zwarcu uzwojenia wtórne, prąd zwarcia jest na tyle nieduży, że transformatorowi nie grozi przegrzanie. Zastosowanie takich zwarciobezpiecznych transformatorów może być w niektórych zastosowaniach godnym polecenia rozwiązaniem, jednak takie właściwości generalnie mają tylko niektóre transformatory o mocach kilku watów.

Dziś coraz powszechniejsze są sieciowe przetwornice/zasilacze impulsowe (SMPS), gdzie ze względów bezpieczeństwa też muszą być stosowane transformatory, zapewniające odpowiednie oddzielenie galwaniczne od sieci energetycznej. Przez lata takie sieciowe transformatory impulsowe wykonywane były z wykorzystaniem rdzeni ferrytowych i proszkowych. Przykład na fotografii 10 pokazuje transformator impulsowy na rdzeniu ferrytowym EI.

Postęp związany ze zwiększaniem częstotliwości pracy pozwolił na zmniejszenie rozmiarów oraz zmniejszenie indukcyjności i liczby zwojów takich transformatorów. Coraz częściej zaczęto mówić o *transformatorach planarnych*.

Cewki i transformatory planarne

Niejasności związane są z pytaniem: jak rozumieć słowo *planarny*?

Niestety, nie ma ścisłej definicji, a określenie *transformator planarny* nie tylko w języku polskim może być i jest aktualnie używane w odniesieniu do różnych rodzajów transformatorów. Zgodnie ze znaczeniem angielskiego słowa *planar*,



Fotografia 9a. Transformator toroidalny



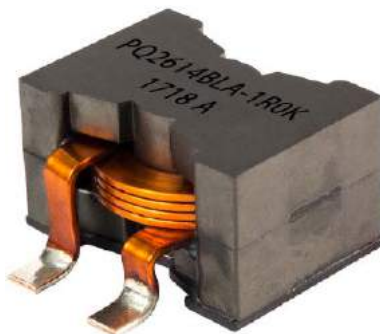
Fotografia 9b. Transformator toroidalny zalewany



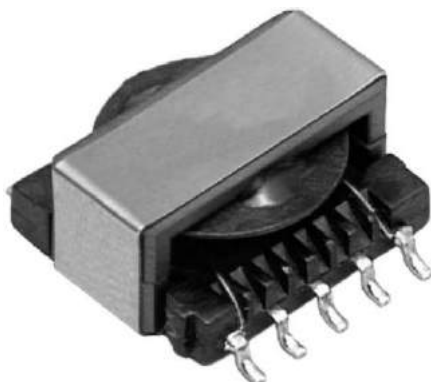
Fotografia 10. Transformator impulsowy na rdzeniu ferrytowym



Fotografia 11a. Transformator z uzwojeniem płaskim



Fotografia 11b. Cewka SMD z uzwojeniem płaskim



Fotografia 12. Transformator o płaskiej konstrukcji z klasycznym uzwojeniem

na pewno chodzi o transformator *płaski*. Ale w języku angielskim jest też słowo *flat*, też używane w odniesieniu do transformatorów i cewek. Określenia *planar transformer* używają inżynierowie i handlowcy na całym świecie w odniesieniu do różnych elementów. Określenie *płaski* może dotyczyć

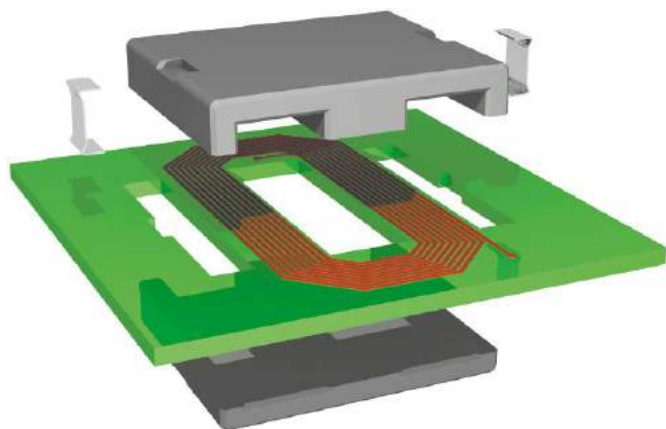
kształtu samego transformatora oraz formy jego uzwojeń.

Fotografia 11a pokazuje transformator, natomiast fotografia 11b cewkę, oba elementy zostały wykonane z uzwojeniami płaskimi.

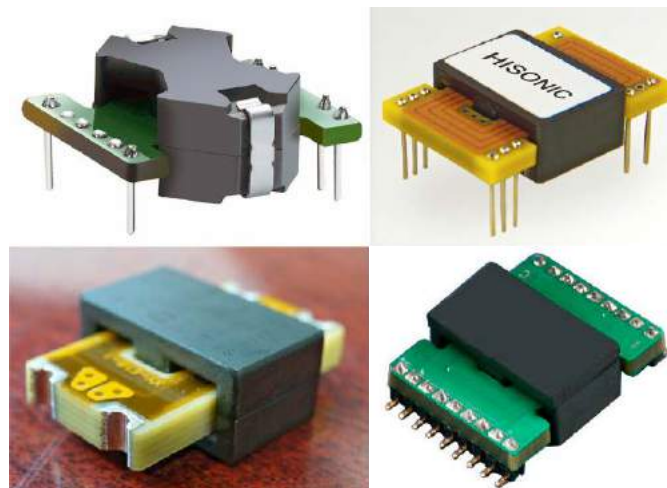
W ofertach niektórych znanych dostawców *transformatory planarne* oznaczają klasyczne transformatory, z klasycznymi uzwojeniami z drutu (rysunek 12). Tutaj słowo planarny niewątpliwie odnosi się do płaskiego kształtu transformatora.

Wykorzystywanie płaskiego drutu (przewodnika) jest jak najbardziej uzasadnione i celowe: przy wysokich częstotliwościach daje o sobie znać zjawisko naskórkowości, polegające na tym, że prądy w.c.z. płyną tylko przy powierzchni przewodnika, a nie wnikają w głąb.

Jednak coraz częściej, dziś już najczęściej, określenie transformator planarny oznacza transformator, którego uzwojenia wykonane są w postaci ścieżek płytki drukowanej. Rysunek 13 pochodzi z niemieckojęzycznej Wikipedii: <http://bit.ly/2IU2Viu>. Uzwojenie jest płaskie, użyte kształtki ferrytowe także, więc takie rozwiązanie



Rysunek 13. Budowa transformatora, którego uzwojenia wykonane są w postaci ścieżek płytki drukowanej



Rysunek 14. Przykłady transformatorów planarnych

w pełni zasługuje na miano *transformatora planarnego*.

Takie transformatory planarne są stosowane coraz częściej. Wzrost szybkości tranzystorów przełączających i ich sterowników pozwala na pracę przy częstotliwościach ponad 1 MHz. Wtedy wymagane indukcyjności są co najwyżej rzędu pojedynczych mikrohenerów, a nawet poniżej jednego mikrohenera, co pozwala zmniejszyć liczbę zwojów do kilku, a to z kolei daje możliwość uzyskania małej rezystancji uzwojeń – ścieżek, a tym samym małych strat i dużej sprawności.

Kilka płaskich, cienkich, niewykazujących problemu naskórkowości zwojów można wytworzyć na wielowarstwowej płytce drukowanej. Grubość warstwy miedzi zwykle jest wtedy znacznie większa, niż standardowe 35 mikrometrów (1oz). A jako rdzenie coraz częściej wykorzystywane są nowe kształtki ferrytowe, opracowane specjalnie właśnie do transformatorów i cewek planarnych.

Proces produkcji takiego transformatora planarnego (cewki planarnej) polega więc w istocie jedynie na założeniu dwóch części ferrytowego rdzenia na płytkę drukowaną, posiadającą drukowane uzwojenia i wycięcia na rdzeń. Taka realizacja wymaga jedynie zakupu dwuczęściowych rdzeni (i ewentualnie łączących je zapinek). Niemniej liczni wytwórcy oferują gotowe transformatory planarne do zamontowania na płytkach. Przykłady na fotografii 14 pokazują różne wykonania.

Transformatory planarne mogą przenosić zaskakująco dużą moc przy niewielkich rozmiarach przede wszystkim dlatego, że pracują przy wysokiej częstotliwości co najmniej kilkuset kiloherców.

Podsumowanie

Tematyka różnorodnych elementów indukcyjnych i ich rdzeni jest ogromnie szeroka. Wprawdzie postęp w dziedzinie materiałów magnetycznych nie jest oszałamiający, niemniej stopniowo dostępne stają się nowe rozwiązania oraz materiały, wprawdzie znane od lat, ale dopiero teraz na tyle tanie, żeby zdobywać szerszy udział w rynku. Przedstawione tu informacje, przypomniane z konieczności pobieżnie i niezbyt precyzyjnie wskazują, że kwestia optymalnego doboru elementów indukcyjnych nie jest prosta. Nie należy bazować na być może już niezbyt świeżych informacjach zdobytych w szkole i na studiach. Jeszcze większe znaczenie ma fakt, że obecnie dominują rozwiązania układowe, zupełnie nieznane kilkanaście lat temu. Choćby tylko z tych względów warto być otwartym na nowe rozwiązania i nowe materiały.

Piotr Górecki

Dylemat konstruktora

Specjalizacją naszej firmy są systemy zasilania awaryjnego. Nasze produkty muszą spełniać najwyższe standardy bezpieczeństwa i niezawodności. Jednocześnie stawiane są wymagania funkcjonalne, które wymuszają stosowanie nietypowych rozwiązań. Poszukujemy producenta transformatorów i dławików, który zapewni bogatą ofertę produktów oraz pełne wsparcie techniczne.

Polecane rozwiązanie

Produkty firmy Indel są projektowane i wykonywane według najwyższych światowych standardów i przy użyciu najlepszych komponentów. Dzięki temu spełniają nawet najwyższe wymagania stawiane przez urządzenia specjalne (np. wojskowe, medyczne, górnicze, kolejowe). Poddawane są rygorystycznej kontroli i dlatego są niezawodne oraz spełniają najwyższe wymagania jakościowe. Produkty Indel, stosowane są w wielu różnych branżach takich jak elektronika, elektrotechnika, energetyka, automatyka, oświetlenie, alarmy i zabezpieczenia zasilania, medycyna, w sprzęcie profesjonalnym i powszechnego użytku. Oferta firmy obejmuje transformatory na rdzeniach kształtkowych i toroidalnych, zalewane, bezpieczeństwa, separacyjne, montowane na PCB i na szynie DIN, do oświetlenia, głośnikowe oraz do wzmacniaczy lampowych a także różne elementy indukcyjne takie jak autotransformatory, cewki, dławiki, statczniki elektroniczne, elektroniczne transformatory oświetleniowe itp.



... Więcej informacji

Firma Indel jest jednym z liderów w Polsce i Europie środkowej w produkcji profesjonalnych transformatorów i elementów indukcyjnych, początki firmy sięgają 1977 r. Firma produkuje i dostarcza bardzo szeroką gamę nowoczesnych i innowacyjnych transformatorów o mocach od 0,5 VA do 600 kVA mających zastosowanie w wielu różnych branżach. Zaopatruje wielu producentów urządzeń elektronicznych i elektrotechnicznych,

liczne hurtownie i firmy instalacyjne. Produkty posiadają certyfikaty CE, GOST i spełniają wymagania dyrektywy ROHS. Firma Indel wdrożyła system ISO 9001:2000 w zakresie dystrybucji materiałów elektronicznych oraz elektrotechnicznych. Pod względem jakości, produkty dorównują komponentom znanych firm zagranicznych a jednocześnie są konkurencyjne cenowo.

Publikacje:

www.elektronikab2b.pl

- Indel – czołowy dostawca podzespołów indukcyjnych w Polsce,
- Szeroki asortyment produktów o wysokiej jakości

Kontakt:

INDEL Sp. z o.o.

01-924 Warszawa,

ul. Rodziny Połanieckich 34A

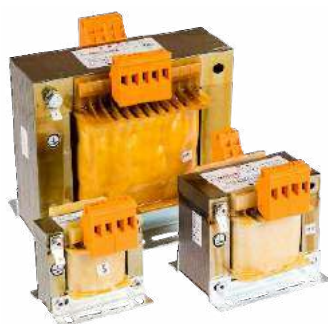
tel./faks 22 669 99 37

www.indel.pl handlowy@indel.pl

Dylemat konstruktora

Projektujemy i wykonujemy aparaturę pomiarową i urządzenia stosowane w medycynie. Naszym produktom stawiane są wygórowane wymagania funkcjonalne a jednocześnie muszą one spełniać najwyższe standardy bezpieczeństwa i jakości. Interesuje nas oferta transformatorów separacyjnych i bezpieczeństwa zgodnych z aktualnymi normami.

Polecane rozwiązanie



Transformatory bezpieczeństwa (obniżające napięcie) zasilają urządzenia elektryczne napięciem bezpiecznym, czyli napięciem, w stanie jałowym nieprzekraczającym 50 V AC. Z kolei transformatorów separacyjnych używamy, gdy nie-

zbędna jest izolacja galwaniczna pomiędzy uzwojeniami PRI/SEC. W tej konstrukcji napięcie wejściowe równe jest napięciu wyjściowemu. Transformatory separacyjne są stosowane w pomieszczeniach o dużej wilgotności, gdzie z różnych przyczyn nie można zastosować transformatora bezpieczeństwa.

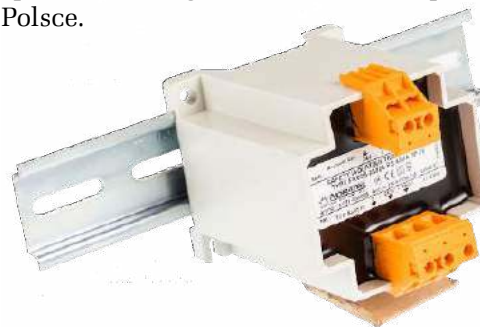
Firma Noratel oferuje szeroki asortyment rozwiązań z zakresu transformatorów separacyjnych i bezpieczeństwa, zgodnych z normą EN61558-2-6. Dostępne są wersje z zabezpieczeniami przed zwarcie i przeciążeniem a także z ogranicznikiem prądów rozruchowych. Wszystkie produkowane wyroby posiadają znak CE, a znaczna część spełnia wymagania DNV, Lloyds Register of Shipping i Bureau Veritas.



... Więcej informacji



Grupa Noratel to producent transformatorów, ze 100-letnim doświadczeniem w branży. Firma swoją działalność rozpoczęła jako lokalny producent sprzętu radiowego, dzisiaj tworzy międzynarodową grupę zrzeszającą 16 spółek w 13 krajach, zatrudnia łącznie niemal 3 tys. pracowników, a powierzchnia hal produkcyjnych zajmuje ok. 10 tys. m². Noratel stanowi najbardziej uniwersalną fabrykę urządzeń indukcyjnych w Europie. W Polsce firma jest jednym z najlepiej rozwijających się przedsiębiorstw w branży. 28 maja 2015 roku otrzymała nagrodę ministra gospodarki za długoletnią działalność produkcyjną w Polsce.



Publikacje:

www.elektronikab2b.pl

- Transformatory Noratel – szerokie portfolio, wiele branż i rynków
- Transformatory, jakich nie znacie
- Niezawodne zasilacze firmy Noratel

Kontakt:

Noratel Sp. z o.o.

72-003 Dobra Szczecińska,

ul. Szczecińska 1K

Sekretariat – tel. 91 311 30 41

Dział sprzedaży – tel. 91 425 06 74/75

www.noratel.com, sprzedaz@noratel.pl


WYBÓR
KONSTRUKTORA

Dylemat konstruktora

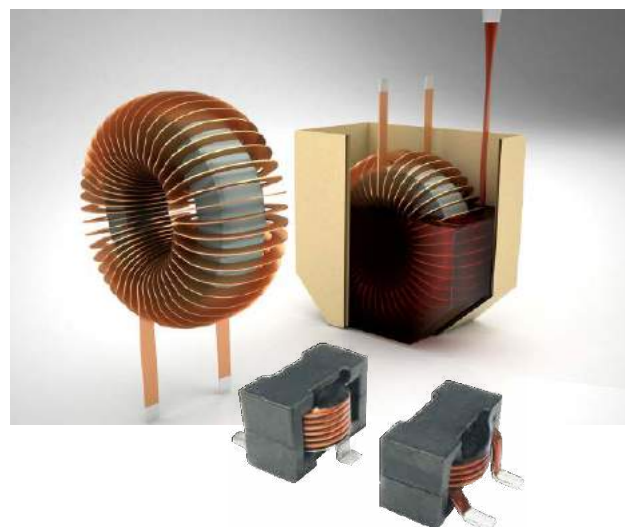
Nasza firma jest pionierem w produkcji systemów multimedialnych przeznaczonych do montażu w autokarach i innych środkach transportu publicznego. Nasze urządzenia odpowiadają m.in. za nagłośnienie, monitory, dostęp do internetu. Aby umożliwić działanie tych systemów w docelowym zastosowaniu, musieliśmy opracować wiele nietypowych, autorskich rozwiązań. Poszukujemy producenta dławików dużej mocy wykonywanych z użyciem najnowszych technologii.

Polecane rozwiązanie

Firma AET to polski producent elementów indukcyjnych, dysponujący nowoczesnym parkiem maszynowym, który specjalizuje się w wykonywaniu induktorów wg projektu klienta jak i zaprojektowanych przez własne biuro konstrukcyjne. Wśród dławików dużej mocy dostępna jest seria LP EDGE, czyli niskoprofilowe wykonane na rdzeniach magnetycznych o niskich stratach mocy. Uzwojenia dławików wykonano w technologii nawijania krawędziowego, która zapewnia bardzo dobre wypełnienie przestrzeni nawojowej drutem o przekroju prostokątnym. Cechą charakterystyczną dławików LP EDGE jest zwarta, kompaktowa konstrukcja, duża obciążalność prądowa oraz możliwość pracy w wysokich częstotliwościach.

Podobnymi cechami charakteryzują się dławiki serii DTHF EDGE. Jest to nowa seria toroidalnych dławików indukcyjnych, w których uzwojenia nawinięto drutem profilowym o przekroju prostokątnym. Rdzeń magnetyczny wykonany z materiału o dużej gęstości strumienia magnetycznego umożliwia dużą obciążalność prądową

przy zachowaniu względnie liniowej charakterystyki indukcyjności w szerokim zakresie temperatur. Ta innowacyjna technologia zapewnia wysoką wydajność prądową dławika przy zachowaniu jego niewielkich gabarytów. Dławiki mogą być umieszczone w kubkach i zalane żywicą o dużym współczynniku przewodności cieplnej.



... Więcej informacji

Firma AET to wiodący polski producent elementów indukcyjnych, który od ponad 25 lat działa w branży elektronicznej. W 2016 roku przeprowadzono inwestycje dofinansowane ze środków Unii Europejskiej, pozwalające na wielkoseryjną produkcję transformatorów, dławików indukcyjnych oraz cewek powietrznych. Spełnienie najwyższych norm jakościowych zapewnia doświadczony ponad 50-osobowy zespół

wykwalfikowanych pracowników i innowacyjne zarządzanie. Elementy indukcyjne AET znajdują zastosowanie w technice oświetleniowej, w układach zasilania awaryjnego, w ładowarkach do pojazdów elektrycznych, w przetwornicach do elektrowozów szynowych i wielu innych aplikacjach. Dzisiaj firma wytwarza blisko 200 tysięcy sztuk wyrobów miesięcznie, a możliwości produkcyjne są nieustannie zwiększane.

Publikacje:

www.elektronikab2b.pl

- AET uruchamia wyposażenie linii do produkcji elementów indukcyjnych nowej generacji
- AET świętuje 25-lecie i inwestuje IRE 2019 str 133

Kontakt:

AET Sp. z o.o. Sp.k.
63-400 Ostrów Wielkopolski
ul. Wysocka 48
www.aet.com.pl, biuro@aet.com.pl

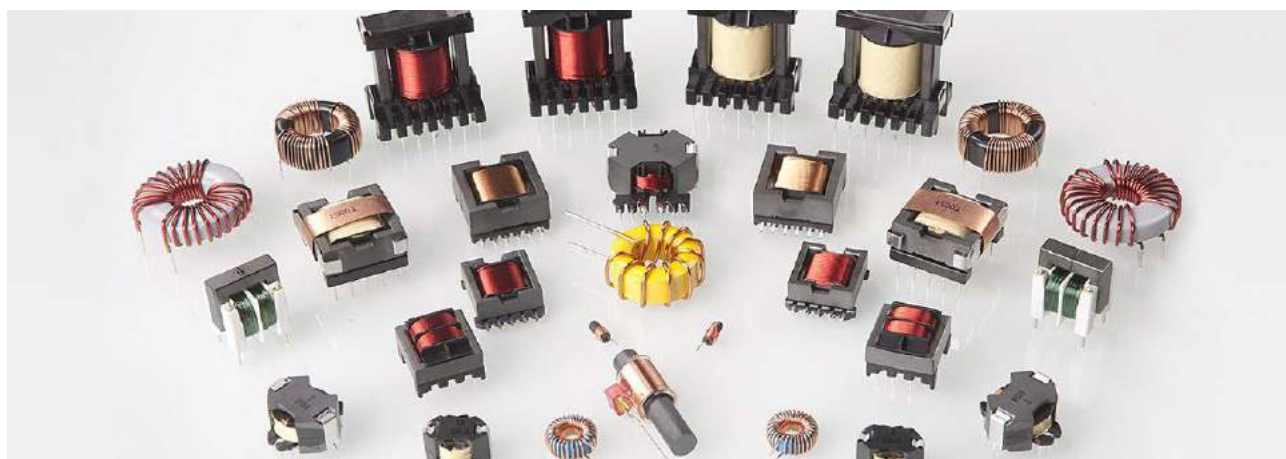
Dylemat konstruktora

Specjalizacją naszej firmy są systemy automatyki budynkowej. Stale udoskonalamy nasze produkty, staramy się by spełniały najwyższe normy i oferowały najnowszą funkcjonalność. W odpowiedzi na oczekiwania klientów, najnowsza seria paneli sterujących i modułów wykonawczych będzie cechowała się niezwykle niskim poborem energii. Interesuje nas współpraca z dostawcą dysponującym szeroką ofertą transformatorów do zasilaczy impulsowych, który udzieli wsparcia technicznego i pomoże w opracowaniu optymalnych rozwiązań.

Polecane rozwiązanie

Producentem, który posiada bogatą ofertę transformatorów do zasilaczy impulsowych jest firma Telzam. Oferuje transformatory do zasilaczy o małych i średnich mocach wyjściowych. Różnorodność rdzeni, z których są wykonywane pozwala na dobranie komponentów odpowiednich do przenoszonych mocy i optymalne wykorzystanie miejsca

na płytkach PCB. Użycie wysokiej jakości materiałów umożliwia pracę tych przetworników przy częstotliwościach do kilku MHz. Firma Telzam od wielu lat dokłada starań aby sprostać oczekiwaniom klientów. Produkowane wyroby zawsze są wynikiem ścisłej współpracy z konstruktorami odbiorcy, dlatego są dopracowane do konkretnej aplikacji.



... Więcej informacji

Firma Telzam zajmuje się produkcją elementów indukcyjnych od 1993 roku. Głównym celem firmy jest produkowanie wyrobów jak najwyższej jakości, oferowanie coraz szerszego asortymentu oraz zapewnienie pełnego zakresu doradztwa technicznego. Atutem firmy jest produkcja według indywidualnych wymagań klienta, również w małych seriach i specyficznych

wykonaniach oraz dostarczanie tych wyrobów w krótkim czasie. Nasze elementy indukcyjne projektowane i wykonywane są przy użyciu dokładnie przetestowanych komponentów, dzięki czemu są niezawodne oraz spełniają najwyższe wymagania jakościowe, co potwierdza certyfikat systemu zarządzania jakością ISO 9001:2009.

Publikacje:

www.elektronikab2b.pl

- Telzam – producent elementów indukcyjnych

Kontakt:

Telzam Sp. z o.o.

18-300 Zambrów, al. Wojska Polskiego 33

tel. 86 271 43 65

www.telzam.pl, telzam@telzam.pl

Oferta prenumeraty Elektroniki Praktycznej

**Prenumerata
wersji
drukowanej**

roczna
176,00 zł
(rabat 8%)

dwuletnia
256,00 zł
(rabat 33%)

**DO 50%
ZNIŻKI NA
PRENUMERATĘ!**

**Prenumerata ze zniżką lojalnościową
za nieprzerwany staż prenumeraty**
(opcja dostępna po zalogowaniu na www.avt.pl):
roczna (staż 1 rok) – 160,00 zł (rabat 16%)
roczna (staż 2 lata) – 144,00 zł (rabat 25%)
roczna (staż 3 lata) – 128,00 zł (rabat 33%)
dwuletnia (staż 3 lata) – 224,00 zł (rabat 41%)
dwuletnia (staż 5 lat) – 192,00 zł (rabat 50%)

Prenumerata e-wydań

- roczna – 91,80 zł (rabat 15%)
- dwuletnia – 151,20 zł (rabat 30%)

Prenumeratorzy wersji drukowanej za równoległe e-wydania płacą 20% ceny: opłata za e-prenumeratę równoległą wynosi 21,60 zł/rok i 43,20 zł/2 lata.

Każdorazowo opłacenie prenumeraty jest premiowane prezentem.

W tym numerze są to do wyboru:

- koszulka z logo EP (rozm. L, XL)
- płyta Poli Rise „Anywhere But Here”

Zamów swój prezent mailowo – prenumerata@avt.pl.



Przywileje Prenumeratorów

- masz prawo do rabatu od 30 do 50% na www.UlubionyKiosk.pl (wystarczy podczas zamówienia podać numer swojej prenumeraty)
- masz prawo do rabatu na www.sklep.avt.pl
- z każdym numerem swojej prenumeraty otrzymasz DVD Niezbędnik Elektronika

Zamów prenumeratę EP w dogodny dla siebie sposób:

- mailowo – prenumerata@avt.pl
- przelewem na konto: AVT-Korporacja sp. z o.o., ul. Leszczyńska 11, 03-197 Warszawa, ING Bank Śląski 18 1050 1012 1000 0024 3173 1013
- a najlepiej na **WWW.AVT.PL** – jeśli zamawiasz po raz pierwszy na www.avt.pl lub przedłużysz ją po zalogowaniu do swojego Panelu Prenumeratora, otrzymasz dodatkowo kody rabatowe na bezpłatne pobranie dowolnych czasopism w wersji elektronicznej ze strony www.UlubionyKiosk.pl

Szanowny Kliencie, od 25 maja 2018 roku w krajach Unii Europejskiej obowiązuje Ogólne rozporządzenie o ochronie danych osobowych (RODO). Zachęcamy do zapoznania się z poniższą klauzulą informacyjną.

Administratorem Twoich danych jest AVT-Korporacja sp. z o.o. z siedzibą ul. Leszczyńska 11, 03-197 Warszawa, e-mail: prenumerata@avt.pl. Chodzi o dane osobowe, które zbieramy, aby móc wysłać Ci nasze czasopisma w formie drukowanej lub elektronicznej oraz inne towary (np. prezenty), a także w innych prawnie usprawiedliwionych celach, w tym marketingu bezpośredniego naszych produktów i usług (tzw. uzasadniony interes administratora). Podanie danych jest dobrowolne, ale niezbędne do zrealizowania zamówienia na prenumeratę. Twoje dane osobowe mogą być przekazane Poczcie Polskiej, która będzie dostarczać do Ciebie przesyłki. Bez Twojej zgody nie prześlemy i nie będziemy dokonywać obrotu (nie użyczymy, nie sprzedamy) Twoich danych osobowych innym osobom lub instytucjom. Twoje dane osobowe możemy przekazać jedynie podmiotom uprawnionym do ich uzyskania na podstawie obowiązującego prawa (np. sądy lub organy ścigania) – ale tylko na ich żądanie w oparciu o stosowną podstawę prawną. Będziemy przetwarzać Twoje dane osobowe przez 5 lat od zakończenia roku obrotowego, w którym wystąpiła ostatnia płatność. Dane osobowe do celów marketingowych będziemy przetwarzać do czasu wycofania przez Ciebie zgody na przetwarzanie lub do czasu usunięcia danych. Informujemy, że masz prawo do żądania od administratora dostępu do Twoich danych, ich sprostowania, usunięcia, ograniczenia ich przetwarzania, wniesienia sprzeciwu wobec przetwarzania Twoich danych lub ich przenoszenia. W każdej chwili możesz odwołać zgodę na przetwarzanie Twoich danych osobowych oraz możesz zażądać, by Twoje wszystkie dane zostały przez nas usunięte.

ISIXRTOS v3 mini system operacyjny dla mikrokontrolerów rodziny M0/M3/M4/M7 (7)

Obsługa wyświetlacza graficznego

Powszechne użycie oraz dostępność smartfonów powodują, że od współczesnych urządzeń wbudowanych, opartych o mikrokontrolery, oczekujemy interfejsu bazującego na dotykowym wyświetlaczu graficznym. Jeszcze nie tak dawno najpopularniejszym interfejsem spotykanym w tego typu urządzeniach był wyświetlacz alfanumeryczny 2×16 ze sterownikiem HD44780 wraz z kilkoma przyciskami mechanicznymi. Tego typu rozwiązania powoli odchodzą do lamusa coraz chętniej zastępowane przez kolorowe wyświetlacze graficzne z ekranem dotykowym. Niestety zastosowanie interfejsu graficznego wiąże się z koniecznością użycia większych zasobów obliczeniowych i pamięciowych, które wielokrotnie przewyższają wymagania głównego algorytmu realizowanego przez urządzenie. Stawia to również dodatkowe wyzwania przed konstruktorem, ponieważ konieczne jest zaprojektowanie interfejsu graficznego, który jest ergonomiczny oraz funkcjonalny, a jak wiadomo nie każdy konstruktor elektronik posiada odpowiedni zmysł artystyczny. Z pomocą przychodzą nam różne biblioteki graficzne, które znacząco upraszczają tworzenie własnych aplikacji.

System ISIX-RTOS wyposażony został w bibliotekę graficzną **libgfx**, która umożliwia tworzenie prostych interfejsów graficznych dla ekranów kolorowych zarówno z klasycznym interfejsem, jak i dotykowym. Biblioteka została napisana w nowoczesnym dialekcie C++17 i nastawiona jest na minimalizację zużycia pamięci. Dysponując wyświetlaczem TFT posiadającym kontroler ze zintegrowaną pamięcią obrazu, biblioteka do działania potrzebuje jedynie 16 kB pamięci RAM. W przypadku, gdy mamy wyświetlacz bez kontrolera, wymagania pamięciowe znacząco rosną, ponieważ potrzebna jest pamięć obrazu, z której generowany jest obraz.

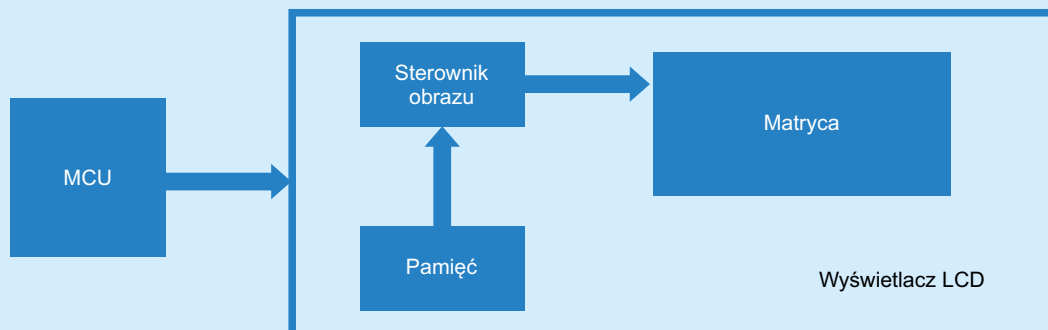
W poprzednim odcinku uruchomiliśmy zewnętrzną pamięć SDRAM, która jest niezbędna do uruchomienia wyświetlacza LCD dołączonego do zestawu STM32F469I-DISCO. Kontynuując, w bieżącym odcinku pokażemy w jaki sposób uruchomić wyświetlacz graficzny znajdujący się w zestawie STM32F469I-DISCO, tak aby współpracował z biblioteką graficzną **libgfx** systemu ISIX.

komendy graficzne (**rysunek 1**). W tym przypadku mikrokontroler komunikuje się ze sterownikiem pamięci obrazu za pomocą magistrali szeregowej np. SPI lub równoległej np. I8080, przysyłając zestaw komend oraz dane, które mają zostać wyświetlone. Zaletą tego typu rozwiązania jest to, iż nie potrzebujemy dużej ilości pamięci wewnętrznej, ponieważ za generowanie obrazu odpowiedzialny jest sam wyświetlacz. Wadą tego typu rozwiązania jest natomiast cena wyświetlacza oraz stosunkowo ubogi zestaw komend wspomagających generowanie obrazu.

W przypadku większych wyświetlaczy dużo bardziej popularne jest podejście z **rysunku 2**, gdzie kontroler obrazu zintegrowany jest z układem SOC (mikrokontrolerem). Zintegrowanie kontrolera z układem SOC znacząco obniża koszt samego wyświetlacza oraz umożliwia realizację zaawansowanych kontrolerów wspomagających np. grafikę 2D czy 3D. Niestety wadą tego rozwiązania jest konieczność użycia zewnętrznej pamięci, która dodatkowo musi

Schemat i architektura wewnętrzna

Jeśli planujemy dołączyć do mikrokontrolera wyświetlacz graficzny o niewielkich rozmiarach (2...4"), w większości przypadków będzie posiadał zintegrowany kontroler, wraz z wewnętrzną pamięcią RAM, realizujący proste

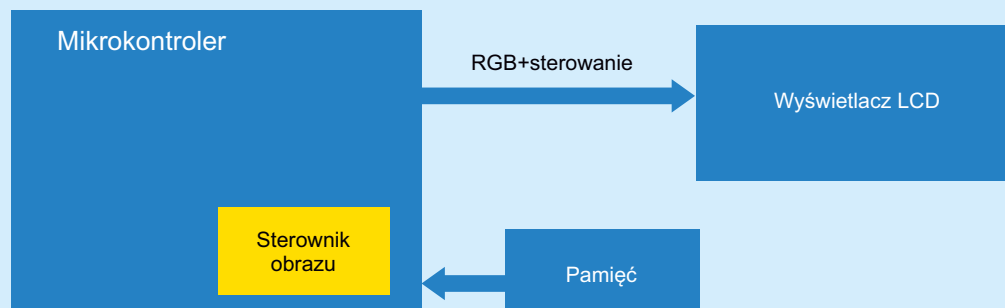


Rysunek 1. Schemat blokowy systemu z wyświetlaczem ze zintegrowanym sterownikiem

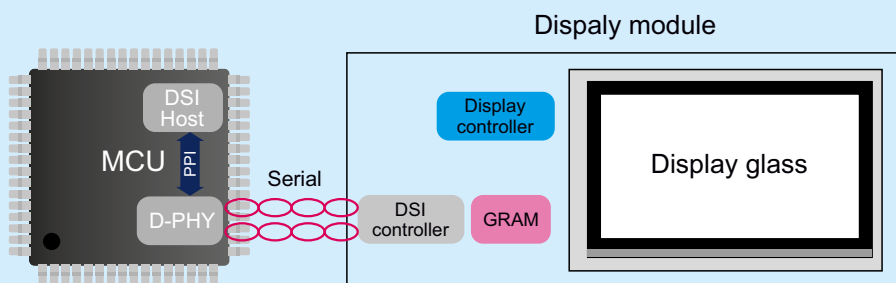
być szybka, oraz odpowiednio duża. W takim przypadku najczęściej konieczne będzie dołączenie pamięci zewnętrznej za pomocą szerokiej magistrali 32 bitowej, co powoduje, iż zaletą użycia mikrokontrolera jednokładowego ulega rozmyciu.

Innym problemem jest stosunkowo duża ilość linii, jakich musimy użyć, aby połączyć wyświetlacz z kontrolerem. Producenci wyświetlaczy i układów SOC szybko dostrzegli ten problem i w celu połączenia wyświetlacza z układem zewnętrznym zaproponowali interfejs MIPI-DSI, który używając szybkiej transmisji z linkiem różnicowym umożliwia przesłanie sygnałów kontrolnych oraz danych obrazu bezpośrednio do wyświetlacza za pomocą dwóch par różnicowych (rysunek 3). Podobnie jak na poprzednim rysunku mikrokontroler zawiera sterownik obrazu, jednak sygnały równoległe zamiast być wyprowadzone na linie GPIO mikrokontrolera, kierowane są do bloku DSI, który konwertuje sygnały równoległe na szeregowy, a następnie przesyła je do wyświetlacza. Zaletą tego typu rozwiązania jest standaryzacja oraz możliwość dołączenia wyświetlacza za pomocą kilku przewodów.

Taki rodzaj wyświetlacza mamy w zestawie STM32F469I-DISCO (rysunek 4). Wyświetlacz zintegrowany z zestawem posiada rozdzielczość 480×800 i wyposażony jest w układ OTM8009A oraz pojemnościowy panel dotykowy, który będzie tematem odrębnego artykułu. Różnicowe linie danych D0, D1, CLK interfejsu MIPI zostały dołączone do odpowiadających linii w STM32F469. Z uwagi na rygorystyczne wymagania czasowe mikrokontroler posiada dedykowane wyprowadzenia nie współdzielone z liniami GPIO, a więc nie ma potrzeby wykonywania konfiguracji, jak ma to zazwyczaj



Rysunek 2. Schemat blokowy systemu z mikrokontrolerem ze zintegrowanym sterownikiem wyświetlacza



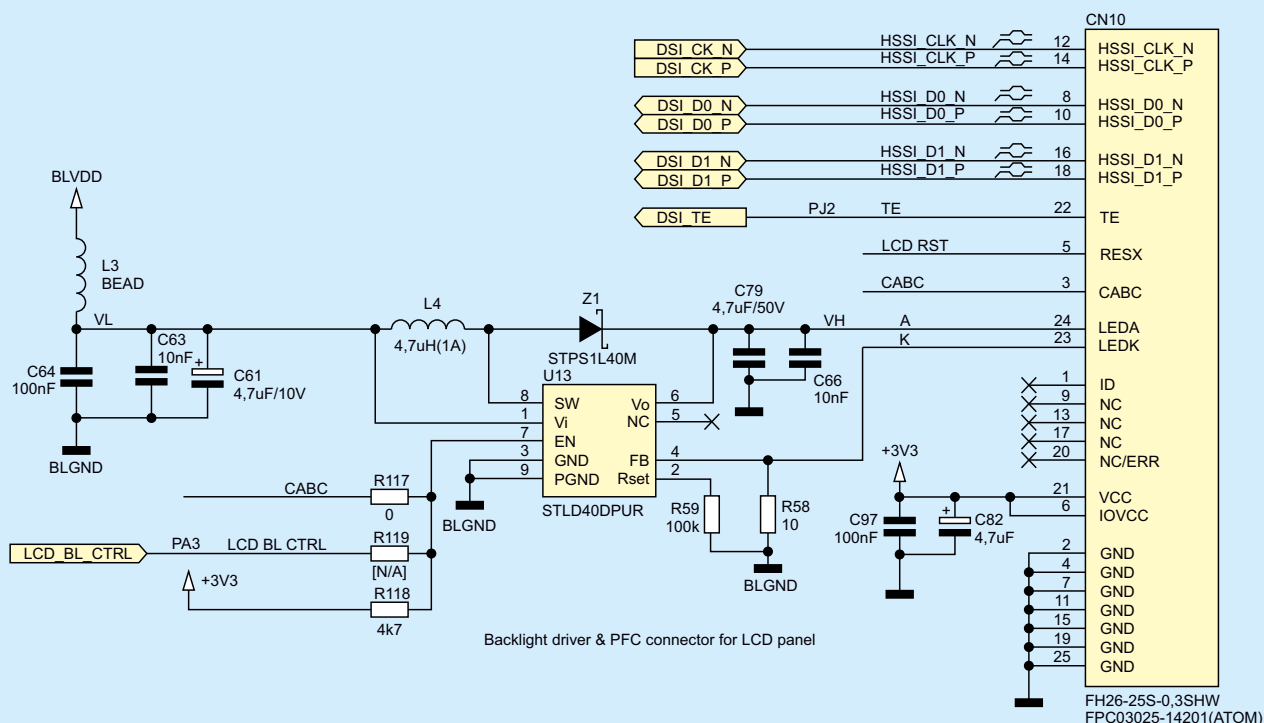
Rysunek 3. Schemat obrazujący połączenie wyświetlacza poprzez interfejs MIPI-DSI

miejsce w przypadku innych funkcji alternatywnych. Dodatkowo sterownie podświetlaniem zostało zrealizowane z wykorzystaniem układu przetwornicy impulsowej U13.

Obsługa sprzętowa wyświetlacza LCD w mikrokontrolerach STM32 jest zrealizowana z użyciem trzech bloków:

- Sterownik kontrolera wyświetlacza **LTDC** (LCD Display Kontroler).
- **DMA2D**, zaawansowany kanał DMA graficznego kontrolera pamięci umożliwiający akcelerację operacji graficznych 2D.
- Kontroler **DSI** umożliwiający konwersję sygnału równoległego RGB na interfejs szeregowy zgodny ze standardem MIPI-DSI.

Sterownik **LTDC** jest odpowiedzialny za generowanie obrazu na podstawie bufora pamięci i umożliwia konwersję bufora obrazu z różnych formatów do formatu natywnego wyświetlacza,



Rysunek 4. Schemat połączenia wyświetlacza zestawu STM32F469I-DISCO

zapewniając możliwość redukcji wykorzystanej pamięci. Dodatkowo obsługuje dwie warstwy obrazu oraz format RGBA, umożliwiając miksowanie dwu warstw na wyjściu wyświetlacza.

Sterownik **DMA2D** może być wykorzystany do wspomagania generowania obrazu 2D i umożliwia realizację 4 podstawowych zadań:

- Wypełnienie obszaru prostokątnego zadaniem kolorem.
- Skopiowanie jednego fragmentu obrazu do innego.
- Automatyczną konwersję w locie między formatami pikseli podczas kopiowania.
- Miksowanie dwóch obrazów o różnych rozmiarach i formatach obrazu do jednego formatu wynikowego.

Jak już wcześniej wspomniano kontroler wyświetlacza LTDC posiada wyjście z interfejsem równoległym, które może być wyprowadzone bezpośrednio na linie GPIO, jeśli dysponujemy tego typu wyświetlaczem, lub może być podany do kontrolera **DSI**, którego zadaniem jest konwersja tego formatu na format szeregowy DSI.

Oprogramowanie

Aby na wyświetlaczu uzyskać podstawowy obraz trzeba wykonać szereg czynności, co jest stosunkowo skomplikowane, ponieważ wymaga konfiguracji kilku odrębnych układów peryferyjnych mikrokontrolera. Należy:

- skonfigurować zewnętrzną pamięć SDRAM, aby było miejsce na przechowywanie ramek obrazu (zagadnienie opisane w poprzednim odcinku),
- uruchomić kontroler LTDC, tak aby generował obraz zgodny z formatem w jakim pracuje wyświetlacz,
- skonfigurować interfejs DSI, zgodnie z parametrami dostarczonymi przez producenta wyświetlacza,
- przesłać komendy kontrolno-sterujące do wyświetlacza za pomocą interfejsu DSI.

Dopiero po wykonaniu powyższych czynności, możemy uzyskać obraz na wyświetlaczu zestawu ewaluacyjnego.

Wróćmy teraz na chwilę do biblioteki graficznej **libgfx**, aby biblioteka miała możliwość wyświetlania obrazu konieczne jest napisanie sterownika wyświetlacza, którego klasą bazową powinna stanowić klasa **gfx::drv::disp_base**. Interfejs tej klasy stanowią następujące metody:

```
virtual void set_pixel( coord_t x, coord_t y, color_t color ) = 0;
```

Metoda odpowiedzialna za zapalenie pojedynczego piksela o współrzędnych przekazanych za pomocą argumentów x,y i kolorze przekazanym jako parametr color.

```
virtual color_t get_pixel( coord_t x, coord_t y ) = 0;
```

Metoda pobierająca kolor danego piksela o współrzędnych przekazanych w parametrach x,y

```
virtual void clear( color_t color ) = 0;
```

Metoda wypełniająca cały ekran kolorem przekazanym jako parametr color

```
virtual void blit( coord_t x, coord_t y, coord_t cx, coord_t cy, coord_t src_y, const color_t *buf ) = 0
```

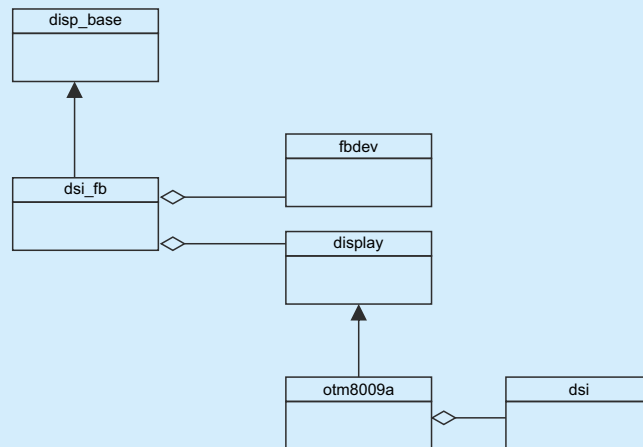
Metoda wyświetlająca fragment pamięci przekazany jako argument buf, na wyświetlaczu rozpoczynając od współrzędnych x,y

```
virtual void vert_scroll( coord_t x, coord_t y, coord_t cx, coord_t cy, int lines, color_t bgcolor ) = 0
```

Metoda wykonująca scroll fragment pamięci obrazu o współrzędnych x,y i rozmiarze c,x cy o ilość linii przekazanych jako argument lines.

```
virtual bool power_ctl( power_ctl_t mode )
```

Metoda, która umożliwia włączenie lub wyłączenie wyświetlacza lub przejście wyświetlacza w stan uśpienia



Rysunek 5. Hierarchia klas biblioteki graficznej

```
virtual void backlight( int /*percent*/ );
virtual int backlight();
```

Metody umożliwiające ustawienie jasności podświetlania wyświetlacza lub pobranie aktualnej wartości jasności wyświetlacza.

Główną klasę sterownika wyświetlacza dla biblioteki graficznej zestawu STM32F469I-DISCO zrealizowano poprzez kod klasy **dsi_fb**, której klasą bazową stanowi wspomniana wcześniej klasa **disp_base**. Ponieważ starano się, aby sterownik mógł być wykorzystany wielokrotnie z różnymi typami wyświetlaczy hierarchia klas jest stosunkowo nieskomplikowana (**rysunek 5**). Klasa używa do współpracy obiektu **fbdev**, który stanowi kod sterownika graficznego LTDC mikrokontrolera STM32. Współpracuje również z klasą bazową **display**, która jest odpowiedzialna za sterowanie fizycznym wyświetlaczem umożliwiając: włączenie lub wyłączenie wyświetlacza, zmianę

Listing 1. Metody klasy fbdev

```
int fbdev::open()
{
    int ret {};
    do {
        //Get periph clock config
        dt::clk_periph pclock;
        ret = dt::get_periph_clock(io(), pclock);
        if(ret) break;
        //Enable device clock
        ret = clock::device_enable(pclock);
        if(ret) break;
        //Configure controller
        for(auto i=0U;i<m_fbinfo->n_layers;++i) {
            hardware_setup(m_fbinfo->layers[i],i);
        } while(0);
        return ret;
    }
}
```

Listing 2. Metody write() klasy dsi, odpowiedzialne za przesyłanie komend do wyświetlacza

```
int open() noexcept override;

/** Close already opened DSI device */
int close() noexcept override;

/** Write operation
int write(int addr, const uint8_t* buf, size_t siz) noexcept override;

/** DSI short command operator */
int raw_write(vch_t vchid, data_t data_type, param_t data0, param_t data1) noexcept;

/** DSI long command operator */
int raw_write(vch_t vchid, data_t data_type, const data_t* param, size_t nparams) noexcept;
```

Listing 3. Klasa otm8009

```
int open() noexcept override;

/** Close device
int close() noexcept override;

/** Setup backlight mode
void backlight( int percent ) noexcept override;

/** Set orientation
int orientation(orientation_t orient) noexcept override;
```

formatu kolorów, oraz zmianę orientacji obrazu. Jest ona równocześnie klasą bazową dla klasy **otm8009a**, która implementuje obsługę wyświetlacza znajdującego się w zestawie ewaluacyjnym. Klasa **otm8009** korzysta z klasy **dsi**, odpowiedzialnej za obsługę magistrali DSI, potrzebnej do przesyłania danych oraz komend sterujących. Skupimy się teraz na najbardziej istotnych fragmentach kodu omówionych obiektów.

Klasa **fbdev** zawiera dwie metody: *open()*, która jest odpowiedzialna za inicjalizację kontrolera LTDC, zgodnie z parametrami konfiguracyjnymi odczytanymi ze struktury DTS, oraz metodą *close()*, która odpowiedzialna jest wyłączenie kontrolera LTDC (**listing 1**). W pierwszym kroku ze struktury DTS odczytujemy konfigurację sygnałów zegarowych, czyli identyfikator magistrali oraz numer bitu odpowiedzialny za włączenie sygnałów zegarowych. Następnie za pomocą funkcji *device_enable()* włączamy sygnały zegarowe odpowiedzialne za taktowanie kontrolera LTDC. W kolejnym kroku ze struktury **fb_info** odczytujemy parametry konfiguracyjne poszczególnych warstw graficznych, oraz za pomocą metody *hardware_setup()* dokonujemy konfiguracji kontrolera LTDC poprzez wpisy do odpowiednich rejestrów. Po wykonaniu powyższych czynności kontroler LTDC rozpoczyna pracę.

Klasa **dsi** implementuje obsługę układu DSI i oprócz samej konfiguracji związanej z parametrami wyświetlacza oraz zawiera kilka metod *write()* odpowiedzialnych za przesyłanie komend do wyświetlacza kanałem sterującym (**listing 2**). Z uwagi na to, że wszystkie wspomniane metody sprowadzają się do wpisów do rejestrów DSI oraz z uwagi na ograniczone łamy artykułu omówienie ich implementacji zostanie pominięte.

Klasa **otm8009** zawiera kod odpowiedzialny za obsługę wyświetlacza LCD umieszczonego w zestawie i implementuje wspomniany wcześniej generyczny **idisplay** (**listing 3**). Podobnie jak poprzednio mamy tutaj metody *open()* i *close()* odpowiedzialne za inicjalizację oraz wyłączenie wyświetlacza. Metoda *backlight()* odpowiada za sterowanie jasnością podświetlania. Metoda *orientation()*, umożliwia konfigurację orientacji obrazu. Wszystkie wspomniane metody sprowadzają się do przesłania za pomocą magistrali DSI komunikatów do wyświetlacza. Dla przykładu metoda zmiany orientacji wyświetlacza została zaimplementowana w sposób jak na **listingu 4**. Jak możemy zauważyć implementacja jest bardzo prosta i polega na przesłaniu odpowiednich sekwencji poleceń do kontrolera wyświetlacza, za pomocą metody *write_seq()*, która z kolei wywołuje metodę *write()* z klasy **dsi**.

Gdy mamy już przygotowane klasy odpowiedzialne za obsługę kontrolera LTDC, DSI, oraz wyświetlacza, klasa **display_fd** może zaimplementować niskopoziomowe API odpowiedzialne za rysowanie. Za włączenie systemu graficznego odpowiada metoda *power_ctl()* (**listing 5**). Jeśli jako argument został przekazany rozkaz włączenia wyświetlacza metoda najpierw wywołuje metody *open()* dla kontrolera LTDC a następnie wywołuje metodę *open()* dla klasy wyświetlacza, w efekcie czego następuje inicjalizacja układów peryferyjnych odpowiedzialnych za generowanie obrazu. Podobnie, gdy jako argument przekazano żądanie wyłączenia wyświetlacza wywoływane są analogiczne metody *close()* odpowiedzialne za wyłączenie poszczególnych układów peryferyjnych.

Gdy już mamy zainicjalizowane niezbędne peryferia, wpis odpowiednich wartości do bufora

obrazu zgodnie z formatem RGB888 spowoduje wyświetlenie odpowiednich punktów na ekranie. W przypadku najprostszej implementacji programowej bez wspomagania DMA2D, rysowanie po ekranie sprowadza się do wpisywania odpowiednich wartości do pamięci. Dla przykładu zaświecenie pojedynczego piksela na pozycji x,y o kolorze **color** pokazuje kod na **listingu 6**.

W pierwszej kolejności wyznaczamy adres w pamięci, odpowiadający pikselowi na pozycji x,y W tym celu współrzędną y mnożymy szerokość wyświetlacza oraz ilość pamięci zajmowanej przez jeden piksel a następnie dodajemy pozycję x pomnożoną przez ilość pamięci zajmowanej przez pojedynczy piksel. Na zakończenie pod odpowiednie adresy pamięci wpisujemy poszczególne składowe kolorów. Pozostałe omówione wcześniej metody możemy zrealizować w podobny sposób poprzez wpisy pod odpowiednie adresy w pamięci obrazu.

Aby zaprezentować możliwości biblioteki graficznej napisano proste demo, które prezentuje wygląd kilku widżetów biblioteki na ekranie. Kod realizujący to zadanie znajduje się w klasie **tft_livedemo**. Aby biblioteka graficzna działała poprawnie należy utworzyć obiekt klasy **gfx::gui::frame**, i przekazać mu referencję do obiektu sterownika wyświetlacza (**listing 7**).

Na początku tworzymy instancję obiektu sterownika magistrali **dsi**, w kolejnym kroku tworzymy nową instancję klasy obsługującej kontroler **LTDC**. W kolejnych krokach tworzymy instancję obiektu **otm8009** odpowiedzialnego za obsługę wyświetlacza przekazując mu referencję do obiektu obsługującego magistralę DSI. W kolejnym kroku tworzymy obiekt sterownika biblioteki **libgfx** przekazując mu referencję do obiektu wyświetlacza oraz obiektu odpowiedzialnego

Listing 4. Przykładowa metoda zmiany orientacji wyświetlacza

```

//!! Change display orientation
int otm8009a::orientation(orientation_t orient) noexcept
{
    using namespace detail;
    if(orient==orientation_t::landscape) {
        auto ret = write_seq(OTM_CMD_MADCTL, (MADCTL_MV | MADCTL_MX));
        if(ret) return ret;
        ret = write_seq(OTM_CMD_CASET, 0x00, 0x00, 0x03, 0x1F);
        if(ret) return ret;
        ret = write_seq(OTM_CMD_PASET, 0x00, 0x00, 0x01, 0xDF);
        if(ret) return ret;
    } else if(orient==orientation_t::portrait) {
        auto ret = write_seq(OTM_CMD_MADCTL, 0);
        if(ret) return ret;
        ret = write_seq(OTM_CMD_CASET, 0x00, 0x00, 0x01, 0xDF);
        if(ret) return ret;
        ret = write_seq(OTM_CMD_PASET, 0x00, 0x00, 0x03, 0x1F);
        if(ret) return ret;
    }
    return error::success;
}

```

Listing 5. Metoda power_ctl() odpowiedzialna za włączenie systemu graficznego

```

bool dsi_fb::power_ctl(power_ctl_t mode)
{
    int ret {};
    do {
        if(mode==power_ctl_t::on) {
            // Open ltdc controller
            ret = m_fb.open();
            if(ret) break;
            //!! Fixme this rename as a separate function
            ret = m_ddsp.open();
            if(ret) break;
        } else if(mode==power_ctl_t::off) {
            ret = m_ddsp.close();
            auto ret2 = m_fb.close();
            if(ret2) ret = ret2;
        }
    } while(0);
    return ret;
}

```

Listing 6. Kod powodujący zaświecenie pojedynczego piksela na pozycji x,y o kolorze color

```

// * Set PIXEL */
void dsi_fb::set_pixel(coord_t x, coord_t y, color_t color)
{
    const auto p =
        reinterpret_cast<volatile uint8_t*>(m_fb.fbmem()
        + get_width() * y * 3 + x * 3);

    *p = color_t_B(color);
    *(p+1) = color_t_G(color);
    *(p+2) = color_t_R(color);
}

```

za sterownik LTDC, z którego będzie mógł odczytać adres bazowy bufora obrazu. W ostatnim kroku mając utworzoną instancję sterownika biblioteki **libgfx**, tworzymy klasę ramki biblioteki graficznej, przekazując jej referencję obiektu sterownika obrazu. Od tego momentu korzystając z instancji klasy **frame**, możemy tworzyć różne widżety z biblioteki **libgfx**.

Do tej pory nie zapoznaliśmy się jeszcze z biblioteką **libgfx** (co uczynimy w kolejnych odcinkach), niemniej jednak, aby coś wyświetlić na ekranie utworzymy kilka komponentów graficznych w celach demonstracyjnych. Główny wątek odpowiedzialny za rysowanie grafiki realizowany jest przez metodę `windows_test`, która jest wywołana w oddzielnym wątku systemu operacyjnego. Najistotniejsze fragmenty programu pokazuje **listing 8**. Na początku tworzymy okno główne programu które będzie zajmować całą dostępną powierzchnię wyświetlacza. W kolejnych krokach w odpowiednich miejscach okna tworzymy odpowiednio: przycisk, menu wyboru, suwak, oraz okno edycyjne. Następnie w parametrach konfiguracyjnych komponentów wpisujemy parametry konfiguracyjne. Na zakończenie ustawiamy widoczność okna na ekranie co spowoduje jego wyświetlenie. W ostatnim kroku wywoływana jest metoda `execute` obiektu **gfx::frame** w wyniku czego uruchomiona zostanie obsługa pętli głównej biblioteki reagującej na zdarzenia.

Uruchomienie

Aby uruchomić przykład należy do komputera podłączyć zestaw **STM32F469I-DISCO** za pomocą złącza mini USB. W kolejnym kroku pobieramy kod źródłowy programu wpisując polecenie:

```
git clone -b pub/ep1019 --recursive https://www.bofff.pl/cgit/public/isixsamples
```

Następnie należy skonfigurować, skompilować program oraz zaprogramować płytkę ewaluacyjną za pomocą następujących poleceń:

```
waf configure --crystal-hz=8000000 --debug waf
waf program
```

Po wykonaniu powyższych czynności na wyświetlaczu powinny ukazać się elementy graficzne jak na **fotografii 6**. Łatwo możemy stwierdzić, że widżety nie działają na dotyk. Dzieje się tak ponieważ brakuje nam sterownika odpowiedzialnego za obsługę dotykowego ekranu pojemnościowego, czym zajmiemy się w kolejnym odcinku.

Lucjan Bryndza, EP
lucjan.bryndza@bofff.pl

Listing 7. Utworzenie obiektu klasy **gfx::gui::frame**, i przekazanie referencji do obiektu sterownika wyświetlacza

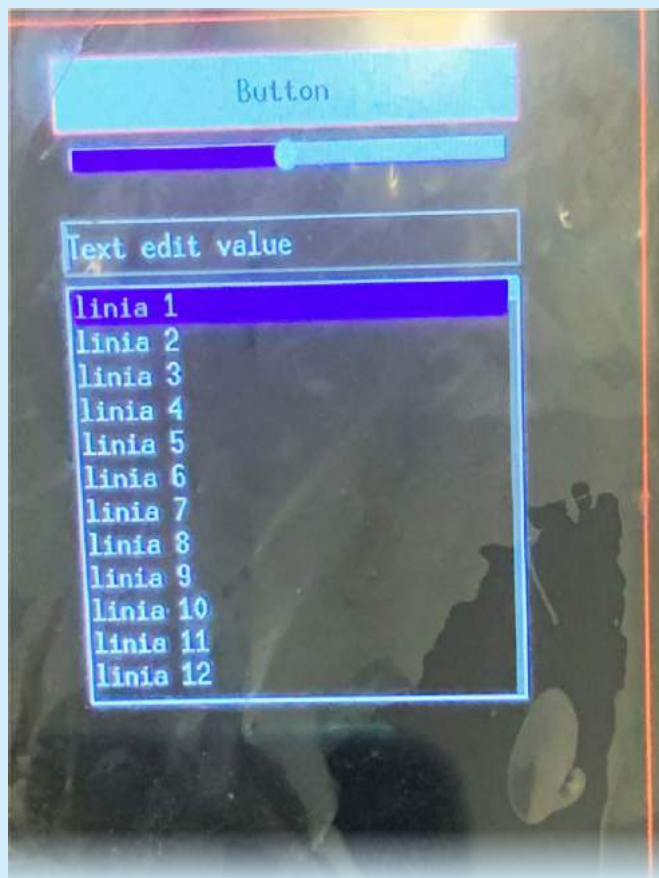
```
periph::display::bus::dsi m_dsi { „dsi” };
periph::display::fbdev m_fb { „ltdc” };
periph::display::otm8009a disp1l { m_dsi, „display” };
gfx::drv::dsi_fb m_disp { m_fb, disp1l };
gfx::gui::frame frame { m_disp };
```

Listing 8. Kod programu demonstracyjnego

```
m_disp.power_ctl( power_ctl_t::on );
window win( rectangle( 10, 10, m_disp.get_width()-20, m_disp.get_height()-20 ),
            frame, window::flags::border | window::flags::selectborder );

button btn( rectangle(80, 20, 300, 40), layout(), win );
choice_menu choice1( rectangle(80, 147, 300, 250), layout(), win, choice_menu::style::normal );
seekbar seek( rectangle(80, 65, 300, 20), layout(), win );
editbox edit( rectangle(80, 110, 300, 30), layout(), win );
edit.readonly(true);
seek.value( 50 );
btn.caption(„Button”);
edit.value(„Text edit value”);
choice1.items( menu1 );

frame.set_focus( &win );
frame.execute();
```



Fotografia 6. Efekt działania programu demonstracyjnego

REKLAMA

**ELEKTRONIKA
PRAKTYCZNA**

**MATERIAŁY
DODATKOWE**



MEDIA

Aby skorzystać z materiałów dodatkowych dołączonych do numeru, należy:

1. Wejść na stronę **www.media.avt.pl**
2. Zarejestrować się lub zalogować
3. Wybrać wydanie „Elektroniki Praktycznej”, które ma trafić do biblioteki osobistej
4. Odpowiedzieć na proste pytanie dotyczące bieżącego numeru
5. Pobrać pliki

RISC-V – budujemy własny mikrokontroler (2)

Otwarte oprogramowanie jest spotykane w wielu dziedzinach od kilkudziesięciu lat. W wielu przypadkach wypierało swoich płatnych konkurentów. Zadaniem, jakie przed sobą stawiają członkowie projektu RISC-V, jest przeprowadzenie podobnej rewolucji w dziedzinie procesorów.



Dostęp do pamięci

W poprzedniej części omówiliśmy instrukcje operujące na rejestrach i stałych oraz skoki i skoki warunkowe. Zatem możemy wykonywać działania i podejmować decyzje. Ostatnim brakującym elementem jest komunikacja ze światem zewnętrznym, którym dla rdzenia jest przestrzeń adresowa. Poza pamięcią RAM, znajdują się w niej, także rejestry obsługujące peryferia. One same nie są częścią rdzenia i nie ma ich opisu w ISA. Każdy twórca mikrokontrolera może stworzyć więc własne. ISA dostarcza nam natomiast zbiór instrukcji pozwalających na odwołanie się do przestrzeni adresowej. **Rysunek 9** przedstawia 5 instrukcji pozwalających na załadowanie wartości z komórki pamięci

Przykładowy program z **listingu 5** (*code/load.S*), wczytuje kolejne znaki c-stringu do rejestru x2. Dla przypomnienia c-string to ciąg znaków w stylu języka C, czyli tablica bajtów zakończona zerowym bajtem. W drugiej linii do rejestru x1 wczytujemy adres początku tablicy z ciągiem znaków. Główna pętla zaczyna się od etykiety *loop*. Najpierw za pomocą instrukcji *lbu* wczytujemy pojedynczy bajt spod adresu *x1+0* do rejestru x2. Następnie zwiększamy wartość x1, aby wskazywała na następny element w tablicy. W linii 7 sprawdzamy, czy wczytaliśmy znak zerowy, oznaczający koniec ciągu znaków. Jeżeli jeszcze nie, to wykonujemy skok do etykiety *loop*. Jeżeli jednak dotarliśmy do końca to przechodzimy do linii 9 i wykonujemy skok na początek programu. W linii 11 za pomocą dyrektywy *.data*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00		
imm[11:0]												RS1			func3			RD			OPCODE			zachowanie									
imm[11:0]												RS1			000			RD			0000011			LB	R[rd]=M[rs+imm]								
imm[11:0]												RS1			001			RD			0000011			LH	R[rd]=M[rs+imm]								
imm[11:0]												RS1			010			RD			0000011			LW	R[rd]=M[rs+imm]								
imm[11:0]												RS1			100			RD			0000011			LBU	R[rd]=M[rs+imm]								
imm[11:0]												RS1			101			RD			0000011			LBH	R[rd]=M[rs+imm]								

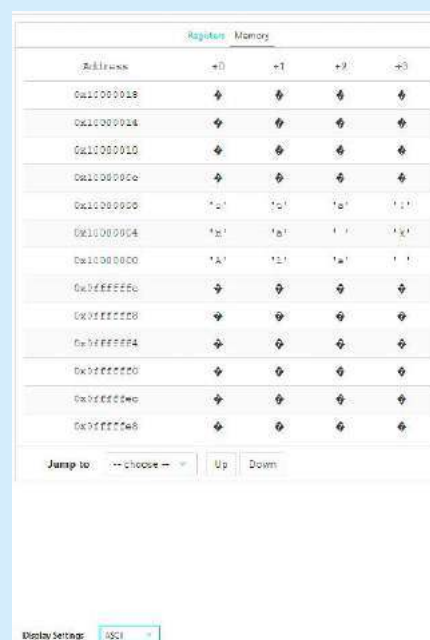
Rysunek 9. Instrukcje wczytujące zawartość komórki pamięci do rejestru

do rejestru rdzenia. Adres w pamięci zawsze podawany jest w bajtach, niezależnie od wczytywanego rozmiaru. Jest on obliczany jako suma wartości rejestru *RS1* oraz stałej bezpośredniej *imm*. Możemy wczytać do pamięci bajt, 2 bajty (pół słowa), albo 4 bajty (słowo) korzystając odpowiednio z instrukcji *LB*, *LH* i *LW*. W przypadku wczytywania liczb ze znakiem o długości krótszej niż długość rejestru niewykorzystane bity muszą zostać wypełnione znakiem. Jest to domyślne działanie rozkazów. Jeżeli wczytywana wartość jest bez znaku musimy skorzystać z rozkazów *LBU* i *LBH*, które zawsze wypełnią niewykorzystane bity zerami. Problem ten nie występuje przy ładowaniu słowa, ponieważ wypełnia ono cały rejestr *rd*.

Listing 5. Wczytywanie c-stringu do rejestru

```
01 start:
02 la x1, array
03
04 loop:
05 lbu x2, 0(x1)
06 addi x1, x1, 1
07 bnez x2, loop
08
09 j start
10
11 .data
12 array:
13 .asciiiz „Ala ma kota!”
```

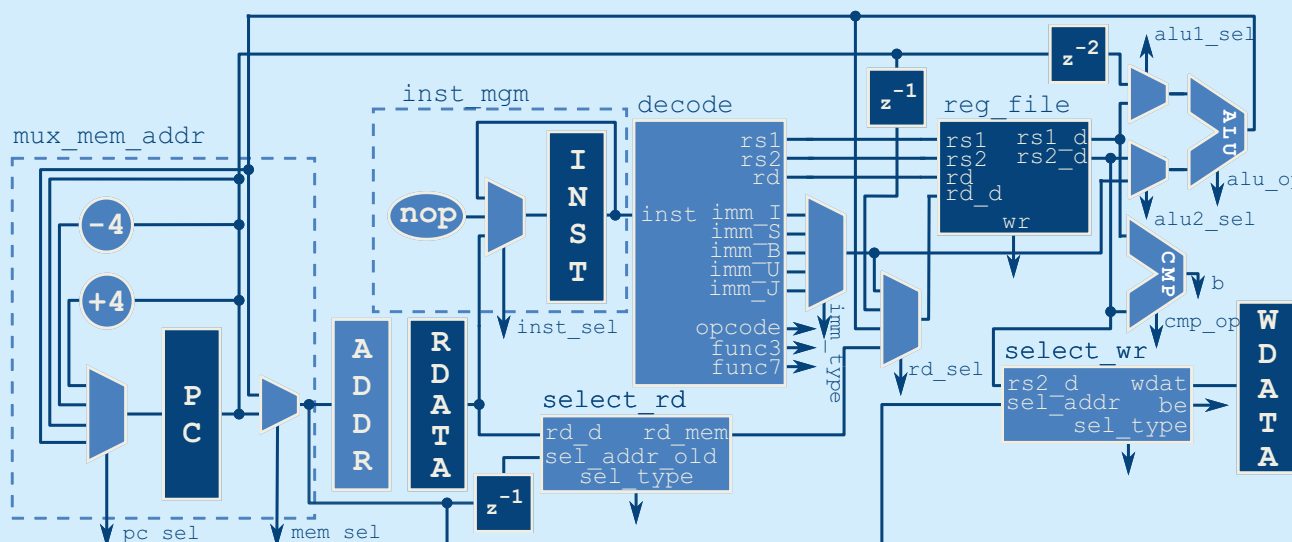
informujemy assembler, że następne dane chcemy umieścić w sekcji danych. W kolejnej linii znajduje się etykieta, którą wykorzystaliśmy do wczytania adresu pod którym znajduje się c-string, który jest zdefiniowany w linii 13 za pomocą kolejnej dyrektywy *.asciiiz*. Aby zobaczyć wygenerowaną za wartość pamięci po uruchomieniu programu w symulatorze w prawym panelu wybieramy



Rysunek 10. Zawartość pamięci danych

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00			
imm[11:5]						RS2						RS1				func3				imm[4:0]				OPCODE				zachowanie						
imm[11:5]						RS2						RS1				000				imm[4:0]				0100011				SB	M[rs1+imm]=R[rs2]					
imm[11:5]						RS2						RS1				001				imm[4:0]				0100011				SH	M[rs1+imm]=R[rs2]					
imm[11:5]						RS2						RS1				010				imm[4:0]				0100011				SW	M[rs1+imm]=R[rs2]					

Rysunek 11. Instrukcje zapisujące zawartość rejestru do komórki pamięci



Rysunek 12. Schemat ścieżki danych implementowanego mikrokontrolera

zakładkę *Memory* (pamięć), a następnie w polu *Jump to* (skocz do) wybieramy *Data* (dane). Dla czytelności w polu *Display Settings* (ustawienia wyświetlania) możemy wybrać *ASCII*. Zobaczymy wtedy wartości w kolejnych rejestrach pamięci podobnie jak na **rysunku 10**. Gdy wrócimy do zakładki *Registers* (rejstry) i zaczniemy wykonywać kroki programu, zobaczymy kolejne litery wczytywane do rejestru x2.

Instrukcje pozwalające na zapis do pamięci są tylko 3, ponieważ tym razem nie ma różnicy, czy zapisujemy wartości z, czy bez znaku. Są pokazane na **rysunku 11**. Adresowanie jest analogiczne jak przy wczytywaniu: sumujemy wartość z rejestru rs1 i stałą imm. Do pamięci zapisywane są młodsze bity rejestru rs2: 8 dla rejestru sb, 16 sh oraz cały rejestr dla rozkazu sw. Na **listingu 6** przedstawiono program, który zapisuje tablicę 3 liczb do pamięci. Tym razem przyjęto, że każda z nich ma rozmiar 4 bajtów. Na początku zapisujemy rozmiar tablicy w rejestrze x1 i adres jej początku w x2. Zapis odbywa się w pętli *loop*. Ponieważ tablica o rozmiarze 3 posiada elementy o adresach od 0 do 2, więc na początku pętli zmniejszamy x2 o 1. Następnie tworzymy adres. Ponieważ rozmiar każdego elementu do 4 bajty, musimy pomnożyć adres elementu razy 4. Realizujemy to w linii 6 za pomocą przesunięcia bitowego. W kolejnej linii do adresu elementu dodajemy adres bazowy. Tak otrzymany adres następnego elementu zapisujemy w rejestrze x3. Sam zapis odbywa się w za pomocą przed chwilą poznanego rozkazu sw. Na końcu sprawdzamy czy dotarliśmy do końca tablicy. Program kończy się etykietą *array*, pod którą umieścimy naszą tablicę.

Podobnie jak w poprzednim przykładzie w symulatorze wybieramy zakładkę *Memory* i w polu *Jump* przechodzimy do *Text*. Aby wyświetlić wyższe adresy możemy skorzystać z przycisku *Up* (góra). Gdy będziemy wykonywać kolejne kroki programu zobaczymy najpierw jak pod adresem 0x2c wpisania zostanie liczba 2, następnie pod 0x28 liczba 1 i na końcu pod adres 0x24 liczba 0. Ta ostatnia już tam była, ale podczas normalnej pracy programu, nikt nie obiecuje, że fragment pamięci, który dostaliśmy został wcześniej wyzerowany.

Poznaliśmy 37 z 40 instrukcji wchodzących w skład ISA RV32I. Pozostałe 3 pominę w mojej implementacji rdzenia. Pierwszą brakującą instrukcją jest *fence*, która odpowiada za kolejowanie operacji na pamięci. Standard pozwala ją jednak traktować jako instrukcję *nop*. Ostatnie dwie to *ecall* i *ebreak*, które mogą być jednak zastąpione jedną instrukcją *system*. Służy ona do wywołania przerwania, pozwalającego na przykład przekazać kontrolę nad maszyną do niższej warstwy, na przykład systemu operacyjnego (tak zwane *trap* – pułapka). Więcej szczegółów na jej temat można znaleźć także w dokumencie *Privileged ISA Specification*.

Budowa rdzenia

Wiemy już jakie efekty powinno spowodować wykonanie przez rdzeń poszczególnych instrukcji. Teraz potrzebujemy stworzyć odpowiedni sprzęt, który będzie pracował zgodnie z naszymi oczekiwaniami. Rdzeń możemy podzielić na dwa podstawowe elementy: ścieżkę danych (*data path*) i ścieżkę sterowania (*control path*). Pierwsza z nich zawiera elementy przetwarzające, a druga generuje sygnały, które odpowiednio sterują częścią pierwszą.

Rysunek 12 przedstawia schemat ścieżki danych. Dla czytelności pominięte zostały ścieżki zegara oraz resetu. Nawet pomimo to na pierwszy rzut oka może wydać się on dość skomplikowany. Dlatego po kolei przyjrzymy się każdemu z jego elementów. Kolorem ciemnoniebieskim zaznaczono rejestry, a jasnoniebieskim logikę kombinacyjną. Ścieżki zakończone strzałkami są połączone do części kontrolnej. Prezentowany mikrokontroler jest zbudowany w architekturze von Neumanna (czasami nazywanej architekturą Princeton). Oznacza to, że zarówno dane jak i program znajdują się w jednej pamięci. Adres w pamięci, do którego mikrokontroler chce uzyskać

```

Listing 6. Zapis do pamięci
01 start:
02 li x1 3
03 la x2 array
04
05 loop:
06 addi x1 x1 -1
07 slli x3 x1 2
08 add x3 x2 x3
09 sw x1 0(x3)
10 bnez x1 loop
11
12 j start
13 array:

```



Rysunek 13. Symbol multipleksera



Rysunek 14. Symbol jednostki obliczeniowej

Listing 7. Implementacja multipleksera w języku SystemVerilog

```

12 package pcPkg;
20   typedef enum bit {
21       MEM_PC,
22       MEM_ALU
23   } mem_sel;
24 endpackage : pcPkg

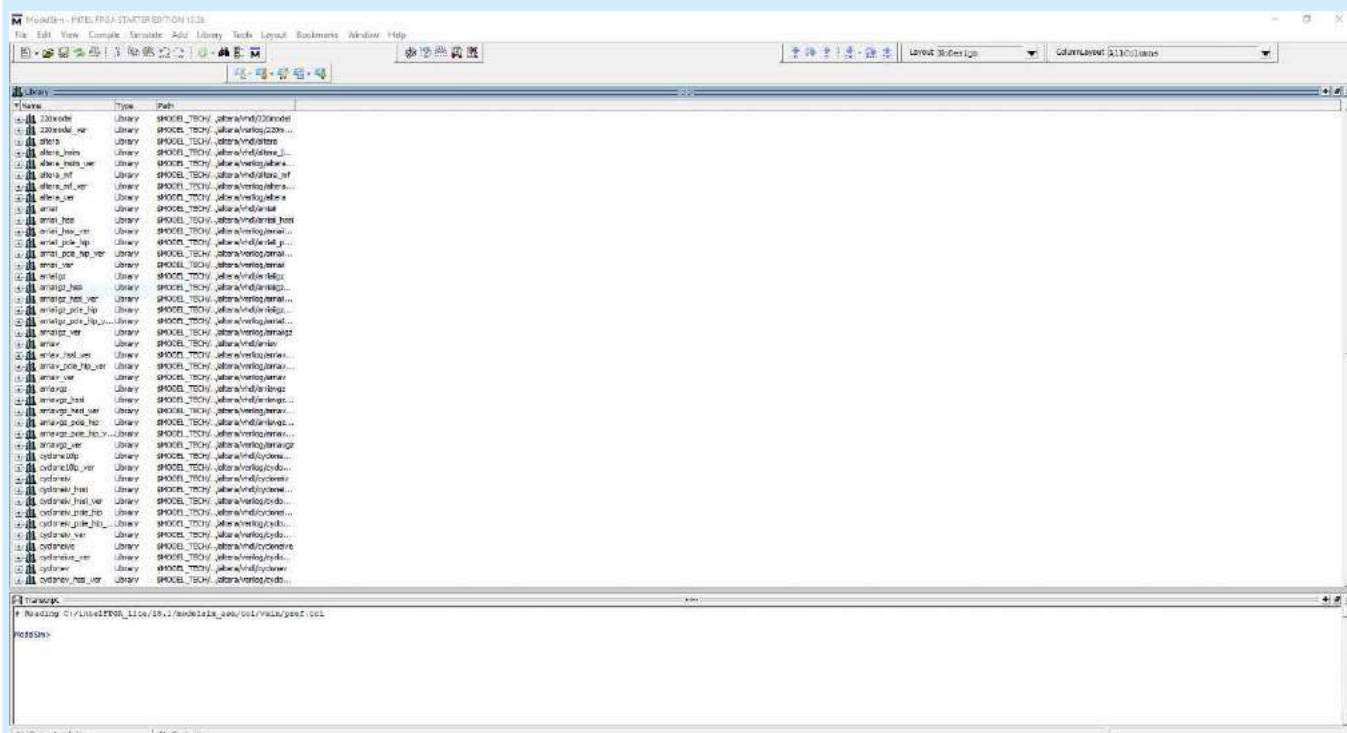
25
51   always_comb
52   case (mem_sel)
53       pcPkg::MEM_PC: addr = pc_reg;
54       pcPkg::MEM_ALU: addr = alu_out;
55   endcase

```

dostęp jest ustawiany na ścieżce ADDR. Odczytana wartość pojawia się w rejestrze RDATA na kolejnym cyklu zegara. Natomiast zapis do pamięci ma miejsce poprzez ścieżkę WDATA.

Najczęściej występującym elementem jest przedstawiony na **rysunku 13** multiplekser. Posiada on kilka wejść danych i jedno wyjście. Każde z nich może być wektorem o długości 1 lub wielu bitów. Posiada także jedno wejście sterujące decydujące o tym, które z wejść będzie połączone z wyjściem.

Na **listingu 7** przedstawiono przykładową implementację multipleksera w języku SystemVerilog. Są to fragmenty z pliku *core/mem_sel.sv*. Na początku w *package* (pakiet) zdefiniowano typ wyliczeniowy *mem_sel*, który zostanie wykorzystany do sterowania multiplekserem. Następnie wewnątrz modułu, w bloku *always_comb* znalazła się sama logika multipleksera zrealizowana za pomocą polecenia *case*. Pozostałe bloki są bardziej wyspecjalizowane. Symbole podobne do tego z **rysunku 14** reprezentują jednostki obliczeniowe. Jego wejścia to dwa wektory, na których przeprowadzana jest operacja, oraz wybór działania. Na wyjściu natomiast dostępny jest wynik.



Rysunek 15. Okno startowe programu ModelSim

Listing 8. Test bench dla jednostki arytmetyczno logicznej (*core/alu_tb.sv*)

```

09 `timescale 100ns / 10ns
10
11 module alu_tb;
12     logic [31:0] in1;
13     logic [31:0] in2;
14     aluPkg::alu_op op;
15
16     alu dut (
17         .alu_in1(in1),
18         .alu_in2(in2),
19         .alu_op(op),
20         .alu_out()
21     );
22
23     int i;
24     logic [31:0] in1_a[N-1:0] = {32'd10, 32'd3, -32'd4, 32'd4, -32'd16};
25     logic [31:0] in2_a[N-1:0] = {32'd2, 32'd10, 32'd4, -32'd4, 32'd2};
26     int n = ($size(in1_a) > $size(in2_a)) ? $size(in2_a) : $size(in1_a);
27
28     initial begin
29         for (i=0; i<n; i++) begin
30             in1 = in1_a[i];
31             in2 = in2_a[i];
32             op = op.first;
33             do
34                 #5 op = op.next;
35             while (op != op.first);
36         end
37     end
38 endmodule

```

W rdzeniu wykorzystane są dwie. Pierwsza z nich oznaczona jako ALU (*core/alu.sv*) odpowiada za operacje arytmetyczno-logiczne takie jak dodawanie czy odejmowanie. Używana jest ona przez operacje na rejestrach oraz do wykonywania skoków. Druga CMP (*core/cmp.sv*) realizuje sprawdzania, czy skoki warunkowe mają zostać wykonane.

Przed złożeniem całego rdzenia warto zasymulować działanie poszczególnych elementów. W tym celu przygotowane są tak zwane *test bench* (ang. stanowisko testowe). Znajdują się one w plikach z dopiskiem *_tb* w nazwie. Na **listingu 8** przedstawiono *test bench* dla jednostki arytmetyczno logicznej (*core/alu_tb.sv*). W linii 9 wybieramy 100 ns jako podstawową jednostkę czasu. Sam test znajduje się w module *alu_tb*, który nie ma żadnych wejść ani wyjść. W liniach 12-14 zdefiniowane są zmienne

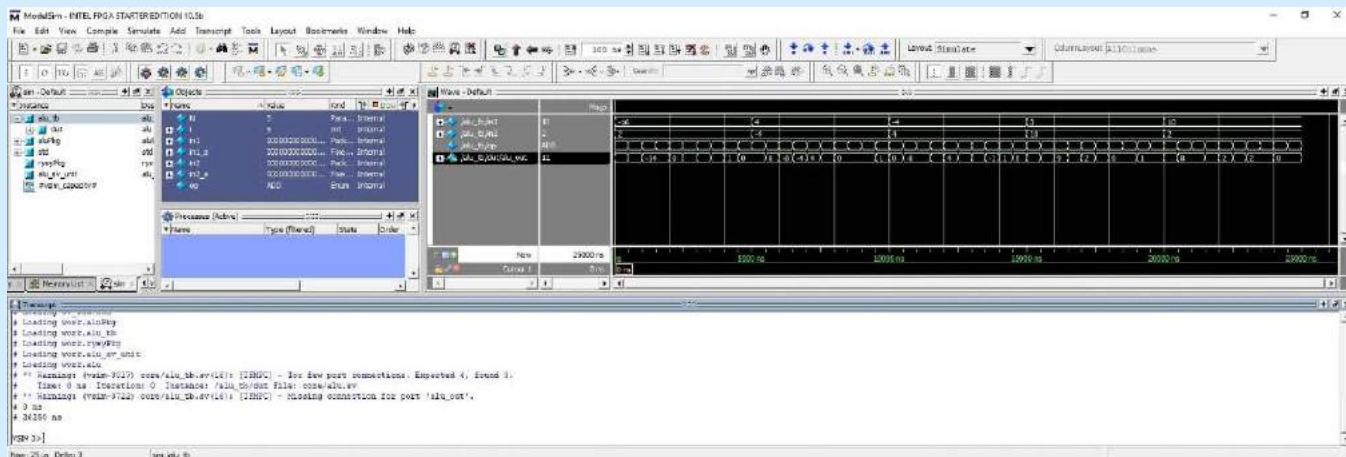
Listing 9. Skrypt uruchamiający symulację ALU

```
08 vsim -novopt work.alu_tb
09
10 add wave -position end -dec sim:/alu_tb/in1
11 add wave -position end -dec sim:/alu_tb/in2
12 add wave -position end -dec sim:/alu_tb/op
13 add wave -position end -dec sim:/alu_tb/dut/alu_out
14
15 run -all
16 wave zoom full
```

pomocnicze, które będą służyły do wygenerowania wejść dla ALU. Sam moduł jest umieszczony w liniach 16-21. Dalej znajdują się zmienne, które posłużą do generowania wymuszeń. Tablice *in1_a* i *in2_a* zawierają listy kolejnych operandów. Liczba testów jest równa długości krótszej tablicy. Wartość ta jest obliczana w linii 26 i zapisywana w zmiennej *n*.

simulate_alu.do). W ósmej linii następuje uruchomienie symulatora i wczytanie naszego modułu. Kolejne cztery linie dodają sygnały, które chcemy obserwować. Po otwarciu symulacji moglibyśmy je przeciągnąć za pomocą myszy, lecz skrypt pozwala nam zautomatyzować tę czynność. Następnie uruchamiana jest symulacja. Skończy się po przesymulowaniu wszystkich przypadków, czyli po 25 μ s. Skrypt możemy uruchomić za pomocą polecenia:
do scripts/simulate_alu.do

Po chwili na ekranie powinniśmy zobaczyć efekt podobny do tego z **rysunku 16**. Aby zobaczyć dokładną wartość pola musimy je przybliżyć za pomocą narzędzia lupa. Powiększony fragment pokazuje **rysunek 17**. Widzimy na nim zmianę wartości w funkcji czasu. W pierwszych dwóch wierszach znajdują się liczby, na których



Rysunek 16. Program ModelSim z otwartym oknem symulacji

Signal	Value	Operation	Result
/alu_tb/in1	-16		
/alu_tb/in2	2		
/alu_tb/op	ADD		
/alu_tb/dut/alu_out	-14		

Rysunek 17. Wynik symulacji działania ALU dla różnych danych wejściowych i działań

W bloku *initial* (linia 28) znajduje się generowanie sygnałów testowych. Za pomocą pętli *for* do wejść przypisywane są kolejne elementy tablicy. Dla każdej pary liczb wewnętrzna pętla *while* przechodzi przez wszystkie elementy typu wyliczeniowego. Korzystamy tu z dwóch funkcji języka: *first* zwraca pierwszy, a *next* następny element typu wyliczeniowego. Po ostatnim elemencie zwracany jest z powrotem pierwszy, dzięki czemu pętla się zakończy. #5 z linii 34 wprowadza opóźnienie o 5 jednostek symulacji, czyli 0,5 μ s.

Aby przeprowadzić symulację włączamy program ModelSim. Można go zainstalować razem ze środowiskiem Quartus 2 od firmy Intel, który można pobrać spod [10]. Jego okno startowe przedstawia **rysunek 15**. Na dole widzimy okno linii poleceń. Znakiem zachęty jest w nim *ModelSim>*. Przechodzimy w nim do folderu, do którego sklonowaliśmy repozytorium z przykładami. W moim przypadku jest to:

```
cd c:/riscv/rysy_core
```

Możemy sprawdzić czy się powiodło za pomocą polecenia:

```
pwd
```

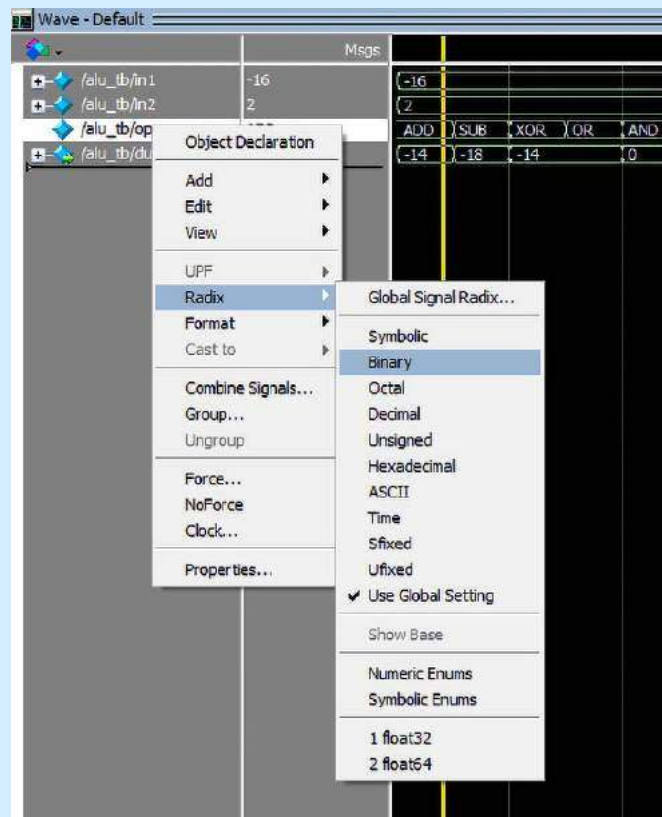
```
# C:/riscv/rysy_core
```

Wyświetla ono aktualny katalog.

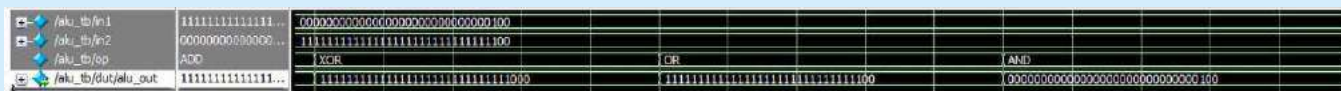
Teraz możemy skompilować projekt za pomocą przygotowanego skryptu:

```
do scripts/compile.do
```

Jeżeli kompilacja zakończy się powodzeniem możemy przejść dalej. Warto przeglądać logi z kompilacji. Są one długie, lecz w przypadku błędów często zawierają wskazówkę, gdzie można szukać przyczyny. Kolejnym krokiem jest uruchomienie symulacji. Tu także możemy użyć skryptu jak na **listingu 9** (znajdziemy go także w pliku *scripts/*



Rysunek 18. Wybór sposobu wyświetlania liczb



Rysunek 19. Symulacja wykonywania działań XOR, OR i AND wyświetlona w systemie binarnym

```

24 logic [REG_LEN-1:0]x[REG_NUM-1:0];
25
26 assign rs1_d = (rs1 == 0) ? '0 : x[rs1];
27 assign rs2_d = (rs2 == 0) ? '0 : x[rs2];
28
29 always_ff @(posedge clk)
30     if (reg_wr)
31         x[rd1] <= rd_d;

```

wykonujemy działania. Trzecia linia to operacja. Zamiast liczb widzimy tutaj nazwy poszczególnych elementów typu wyliczeniowego. W ostatnim wierszu znajduje się wynik. Liczby są wyświetlane w formacie dziesiętnym, co ułatwia analizę działań arytmetycznych. Weryfikacja operacji logicznych byłaby łatwiejsza w trybie binarnym. Aby zmienić sposób wyświetlania (**rysunek 18**) klikamy na wiersz prawym przyciskiem myszy i wybieramy interesujący nas format z zakładki *Radix* (podstawa systemu liczbowego). W tym przypadku będzie to *Binary* (binarny). Na **Rysunku 19** przedstawiono wyniki symulacji operacji logicznych *xor*, *or* i *and* (albo, lub i i).

Wróćmy z powrotem do naszego rdzenia. Zaczynając od lewej strony mamy blok `mux_mem_addr`, który pozwala wybrać jaką wartość zostanie wpisana podczas następnego narastającego zbocza zegara do rejestru PC. Najczęściej będzie to adres kolejnej instrukcji, czyli PC+4. Jednak w przypadku skoków wybrana zostanie wartość z jednostki arytmetyczno logicznej. W niektórych przypadkach możliwość podtrzymania poprzedniego adresu, lub cofnięcia się do poprzedniego także okażą się przydatne. Wyboru dokona część sterująca za pomocą ścieżki `pc_sel`.

Kolejny multiplexer sterowany ścieżką *mem_sel* ustala czy pamięć ma być zaadresowana wartością z rejestru PC, czy z wyjścia ALU. Pierwsza możliwość pozwala wczytać kolejną instrukcję, a druga jest wykorzystywana przy pobieraniu (albo zapisywaniu danych).

W kolejnym cyklu zegara wartość spod podanego adresu zostanie pobrana do RDATA i przez to równocześnie wejdzie na wejścia dwóch bloczków: *inst_mgm* i *select_rd*. Drugi z nich odpowiada za pobieranie danych z pamięci. Dokładniej przyjrzymy mu się później. Najpierw zajmiemy się dekodowaniem instrukcji. Za pomocą ścieżki *inst_sel* dokonywany jest wybór, czy następna instrukcja ma pochodzić z pamięci, czy ma być to instrukcja pusta *nop*, czy chcemy podtrzymać poprzednią wartość. Z rejestru INST rozkaz trafia na bloczek *decode*. Zajmuje się on dzieleniem kodu rozkazu na części. Jedyłą, która występuje zawsze jest *opcode*. Jednak ponieważ poszczególne elementy, takie jak adresy rejestrów źródłowego i docelowego, różne typy stałych czy kody funkcji znajdują się zawsze w tych samych miejscach możemy zawsze przygotować odpowiednie sygnały. Jego działanie polega więc na przegrupowaniu bitów instrukcji tak aby stworzyły nowe wektory, które ewentualnie będą wykorzystane. Zawarta w nich informacja nie zawsze będzie poprawna, ale nad jej użyciem czuwać będzie część sterująca.

Kolejnym ważnym elementem są rejestry (*reg_file*). Błoczek ten, poza zegarem ma pięć wejść: *rs1*, *rs2* i *rd* pozwalają na wybór wektorów do odczytu i do zapisu. *rd_d* doprowadza dane, a sam zapis jest włączany przez wejście *wr*. Wyjścia są tylko dwa: *rs1_d* i *rs2_d* zawierają dane z wybranych rejestrów. Początkowo planowałem, żeby rejestry były zerowane przy resecie rdzenia. Wtedy jednak w całości został on zsyntezowany z elementów logicznych (LE – logic elements). Pomińnięcie resetu, który nie jest niezbędny powoduje, że jako rejestry może zostać wykorzystana blokowa pamięć RAM. Zmiana ta spowodowała zmniejszenie wykorzystanych LE z około 3700 do 1600 (przy 8000 dostępnych w wybranym układzie) przy

równoczesnym wykorzystaniu 2 kilo bitów pamięci (z 378 kb dostępnych). Logikę jego działania przedstawia **listing 10**. W linii 24 zdefiniowano tablicę wektorów *x* w której przechowywane są dane. Odczyt danych następuje w liniach 26 i 27. Ponieważ wartość zerowego rejestru jest zawsze równa 0 dodano sprawdzenie za pomocą znanego także z języka C operatora `?:`. Samo działanie rejestru jest modelowane w bloku *always_ff*. Na każdym narastającym zboczu zegara nastąpi sprawdzenie, czy *reg_wr* jest ustawione. Jeżeli tak wartość z wejścia zostaje wpisana w żądane miejsce.

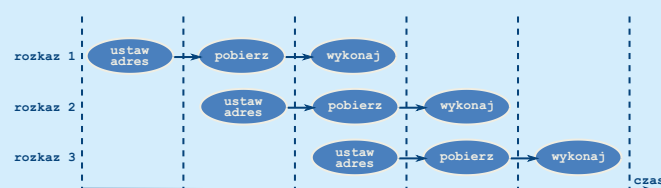
Bloki *select_wr* i *select_rd* pozwalają na wybór bajtów, które mają być zapisane albo odczytane z pamięci. Przyjrzymy się im dokładniej przy omawianiu instrukcji STORE i LOAD. Pozostaje jeszcze tajemniczy moduł oznaczony jako z^2 . Oznacza on opóźnienie sygnału o dwa takt zegara.

Sterowanie rdzeniem

Dostęp do pamięci, a następnie wczytanie danych do rejestru INST zajmują po jednym takcie zegara. Oznacza to, że pobranie i wykonanie każdej instrukcji zajmuje łącznie trzy takty zegara. Schematycznie przedstawia to **rysunek 20**. Jednak zadania te mogą być wykonywane dla kolejnych rozkazów równolegle. Takie podejście nosi nazwę przetwarzania potokowego. Idee obrazuje **rysunek 21**. Przedstawia on w jakich momentach czasu następuje przetwarzanie kolejnych kroków trzech kolejnych instrukcji. Podczas pierwszego taktu zegara następuje ustawienie adresu pierwszego rozkazu. W czasie drugiego taktu pierwszy rozkaz zostaje zapisany do rejestru INST, a ustawiony zostaje adres drugiego. W czasie trzeciego taktu zegara pierwsza instrukcja jest wykonana, druga jest wczytana do rejestru INST, a ustawiony jest adres trzeciej. W kolejnym kroku obsługiwane są już tylko dwie instrukcje, a w piąty wykonana zostaje ostatnia. Dzięki przetwarzaniu potokowemu wykonanie trzech rozkazów zamiast 9 zajęło 5 taktów zegara. W ogólnym przypadku wykonanie n rozkazów podzielonych na p kroków zajmie $n+p-1$ taktów zegara (zamiast $n \cdot p$). Jak zobaczymy później sytuacja nie jest zawsze taka piękna. Na przykład w przypadku skoków adres kolejnej instrukcji poznajemy dopiero na ostatnim etapie, czyli podczas wykonania instrukcji. Oznacza to, że rdzeń będzie musiał wyczyścić potok i poświęcić dwa dodatkowe takty na jego ponowne napełnienie. W dużych procesorach, których potok jest zdecydowanie dłuższy koszt jego wyczyszczenia jest nieporównywalnie większy. Z tego względu stosuje się *branch prediction* (przewidywanie rozgałęzienia (skoku)). Procesor próbuje zgadywać jaki rozkaz powinien być ładowany jako następny. Dopiero jeśli zgadnie źle konieczne jest wyczyszczenie potoku. W założeniu nie powinno być to widoczne dla użytkownika, jednak błędy związane z tym mechanizmem są podstawą dość znanych ataków *meltdown* i *spectre* [11].



Rysunek 20. Kolejne etapy wykonywania instrukcji



Rysunek 21. Przetwarzanie potokowe instrukcji

ADDR	zawartość
0x00	addi x1, x0, 10
0x04	addi x2, x0, 5
0x08	add x3, x2, x1
0x0c	nop
0x10	nop
	nop

Rysunek 22. Program demonstrujący działanie na rejestrach

Operacje na rejestrach

Na początek przyjrzymy się jak działają instrukcje operujące na rejestrach. Zawartość pamięci z programem demonstracyjnym prezentuje **rysunek 22**. Pierwszy rozkaz powoduje, że do rejestru *x1* wczytana jest liczba 10. Precyzyjniej mówiąc wykonana jest operacja $x1 = x0 + 10$, ale jak wiemy rejestr *x0* ma zawsze wartość równą 0. Kolejna operacja ładuje do rejestru *x2* liczbę 5. Ostatnia instrukcja realizuje operację $x3 = x1 + x2$. Na potrzeby naszych rozważań zakładamy, że reszta pamięci zawiera rozkaz *nop* (brak operacji).

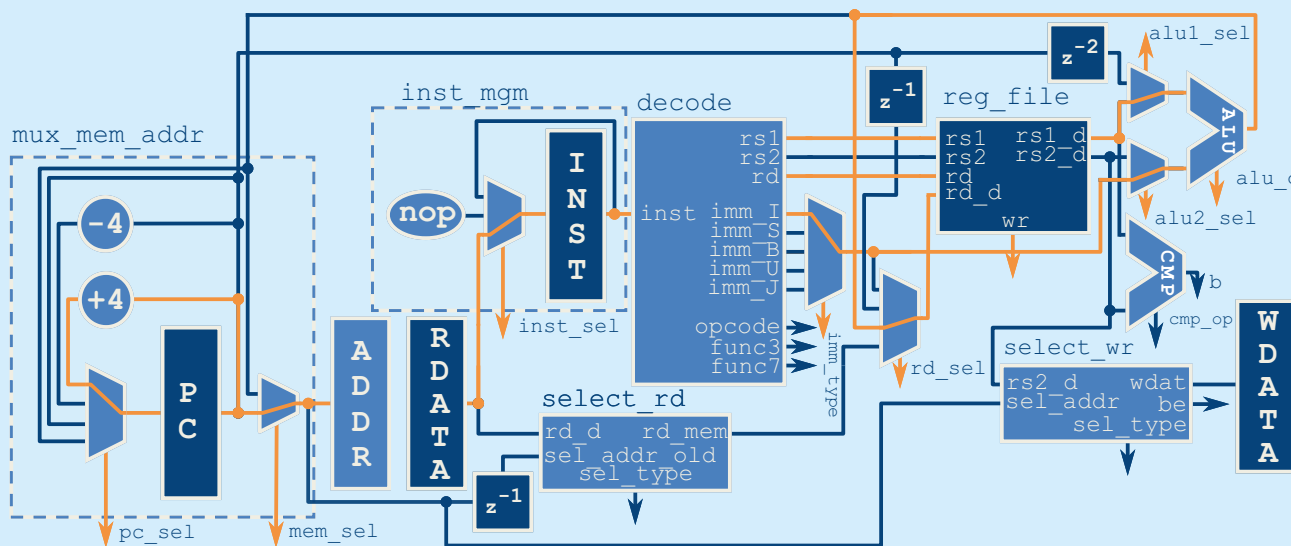
Wykonywanie programu rozpoczyna się od naciśnięcia przez użytkownika przycisku reset, który powoduje ustawienie PC na 0. Wiemy już jednak, że rozkaz z pamięci odczytany spod tego adresu pojawi się w rejestrze INST dopiero za dwa cykle. Tutaj z pomocą przychodzi nam bloczek *inst_mgmt*, który pozwala wybrać jako wartość rozkaz *nop*, czyli brak operacji. Jest to rozkaz, który obiecuje nam, że stan rdzenia (w tym przypadku ustalony przez reset) nie zostanie zmieniony. Blok sterowania musi więc zapewnić, że ta opcja zostanie wybrana przez dwa pierwsze takty.

Zanim przyjrzymy się więc jak będzie wyglądał potok instrukcji wykonywanych przez rdzeń podczas wykonywania naszego programu zobaczymy w jaki sposób część sterująca zapewni przepływ danych, gdy instrukcja dojdzie do fazy wykonania. Część sterująca zapewnia wtedy odpowiedni przepływ danych w rdzeniu. Dokonuje tego poprzez wybór odpowiednich sygnałów sterujących. Ścieżka oznaczona na **rysunku 23** kolorem pomarańczowym obrazuje przepływ danych dla operacji z rodziny OP-IMM. Zaczniemy analizę od rejestru licznika programu. Ta grupa instrukcji nie zmienia kolejności wykonywania programu, dlatego wybiera, że rejestr PC zostanie zwiększony o 4, dzięki czemu będzie wskazywał następną instrukcję. Multiplexer *mem_sel* wybiera, że pamięć zostanie zaadresowana wartością z rejestru PC. Kolejnym elementem jest wybór następnej instrukcji. Ponieważ te rozkazy nie zaburzają przepływu programu

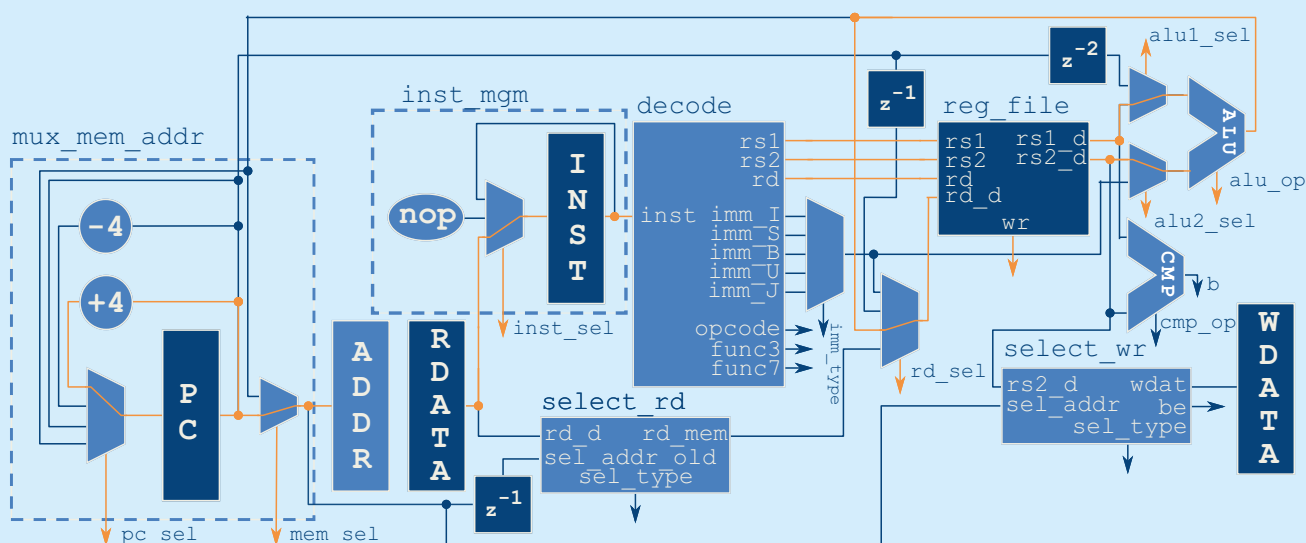
inst_sel przepuszcza wartość odczytaną z pamięci. W chwili, którą analizujemy rejestr INST zawiera rozkaz *addi*. Trafia on na dekodery, który dzieli go na poszczególne fragmenty takie jak *opcode*, numery rejestrów czy wartości różnego typu stałych. Za każdym razem generowane są wszystkie możliwości, mimo że większość z nich nie ma sensu. O tym, które zostaną wykorzystane zadecyduje blok sterujący za pomocą multiplexerów. W naszym przypadku interesujące informacje są zawarte w polach *rs1*, *rd* i *imm_I*. Pole *rs2* jest kodowane przez te same bity instrukcji co *imm_I*. Będzie więc zawierało wartość zależną od podanej stałej, lecz z punktu widzenia funkcjonalności będą to śmieci. Pomimo to zostanie ona także podana na wejście bloku rejestrów, co spowoduje, że jego zawartość pojawi się na wyjściu *rs2_d*. Nie będzie ona jednak wykorzystana, ponieważ *alu2_sel* wybierze zamiast niej stałą bezpośrednią. Operacja, którą ma wykonać ALU zostanie wybrana poprzez ścieżkę *alu_op*. Zostanie ona ustawiona przez blok sterujący na podstawie wartości pola *func3*. Wynik działania zostanie skierowany przez multiplexer *rd_sel* z powrotem do bloku rejestrów. Stan wysoki na wejściu *wr* spowoduje, że podczas kolejnego zbocza zegara zostanie on zapisany w rejestrze docelowym.

Sytuacja dla instrukcji z rodziny OP jest bardzo podobna (**rysunek 24**). Jediną różnicą jest inne ustawienie multiplexera *alu2_sel*. Tym razem zamiast stałej bezpośredniej wybrana jest zawartość rejestru *rs2*. Powoduje to, że wartość wybrana przez *imm_type* jest nieistotna. Nie ma znaczenia, którą wartość wybierze, ponieważ i tak nie zostanie ona nigdzie zapisana. A każda z możliwości zawiera śmieci.

Zobaczmy teraz jak będzie wyglądał potok instrukcji. Jest on przedstawiony na **rysunku 25**. Wiemy już, że wykonanie 3 instrukcji zajmie łącznie 5 cykli zegara. W pierwszym takcie zegara adres wskazuje na pierwszą instrukcję. Nie wiemy jednak jaki adres był wskazywany wcześniej, więc do rejestru RDATA została wczytana przypadkowa wartość. W fazie wykonaj znajduje się instrukcja *nop* umieszczona tam podczas resetu. W kolejnym takcie adres został ustawiony na instrukcję numer dwa, natomiast instrukcja numer jeden została już zapisana w rejestrze INST. W fazie wykonaj znalazłyby się wczytane poprzednio śmieci, lecz część kontrolna będzie musiała zadbać, żeby zostały one nadpisane przez drugą instrukcję *nop*. Dopiero w trzecim takcie zostanie wykonana pierwsza z naszych instrukcji. Jak łatwo zauważyć w tym momencie rdzeń zajmuje się całym naszym krótkim kodem. W czwartym takcie wykonana zostanie druga komenda, natomiast adres wskazuje na pierwszą z instrukcji *nop*, którymi na potrzeby tego eksperymentu wypełniliśmy niewykorzystaną część pamięci. W piątym takcie zostanie wykonana



Rysunek 23. Przepływ danych dla operacji na rejestrach i stałej bezpośredniej



Rysunek 24. Przepływ danych dla operacji na rejestrach

ostatnia instrukcja naszego kodu. Strzałki pokazują „drogę” którą przebyły kolejne rozkazy.

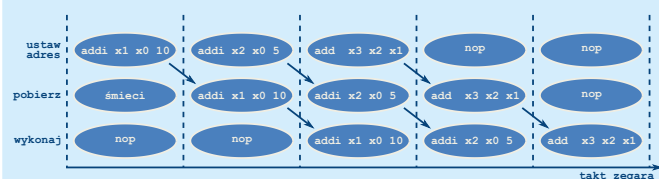
Aby dokładniej przyjrzeć się co dzieje się ze stanem poszczególnych rejestrów rdzenia uruchomimy nasz program w symulatorze ModelSim. Najpierw musimy zasemblować przykładowy program do postaci binarnej. Można zrobić to ręcznie, ale wygodniej skorzystać z symulatora Venus [8]. Gotowy program znajduje się w pliku *code/regs.mem* oraz na **listingu 11**. W pierwszej kolumnie znajduje się 32 bitowa liczba zapisana w podstawie szesnastkowej. Jest to zakodowany rozkaz procesora. W drugiej kolumnie, za znakiem komentarza znajduje się odpowiadająca jej instrukcja Assemblera. Ostatnia kolumna zawiera pseudoinstrukcje. Wygenerowane wartości liczbowe zostaną wykorzystane do zainicjowania pamięci symulowanego procesora. Tym razem do symulacji użyjemy model z pliku *rysy_tb.sv*, który zawiera już działający mikrokontroler. Jego budowę zajmiemy się później, gdy już zapoznamy się z elementami składowymi. Na razie do uruchomienia posłuży nam skrypt *scripts/simulate_regs.do*. Uruchamiamy go podobnie jak poprzednio. Włączamy ModelSim i w konsoli przechodzimy do folderu zawierającego repozytorium z przykładami. W moim przypadku jest to:

```
cd c:/riscv/rysy_core
```

Następnie uruchamiamy:

```
do scripts/simulate_regs.do
```

Po chwili zobaczymy efekt podobny do tego z **rysunku 26**. W pierwszej linii widzimy sygnał opisany jako *clk*. Jest to zegar. Wszystkie



Rysunek 25. Fazy, w których znajdują się poszczególne instrukcje podczas kolejnych cykli zegara



Rysunek 26. Symulacja wykonania kodu z rysunku 22

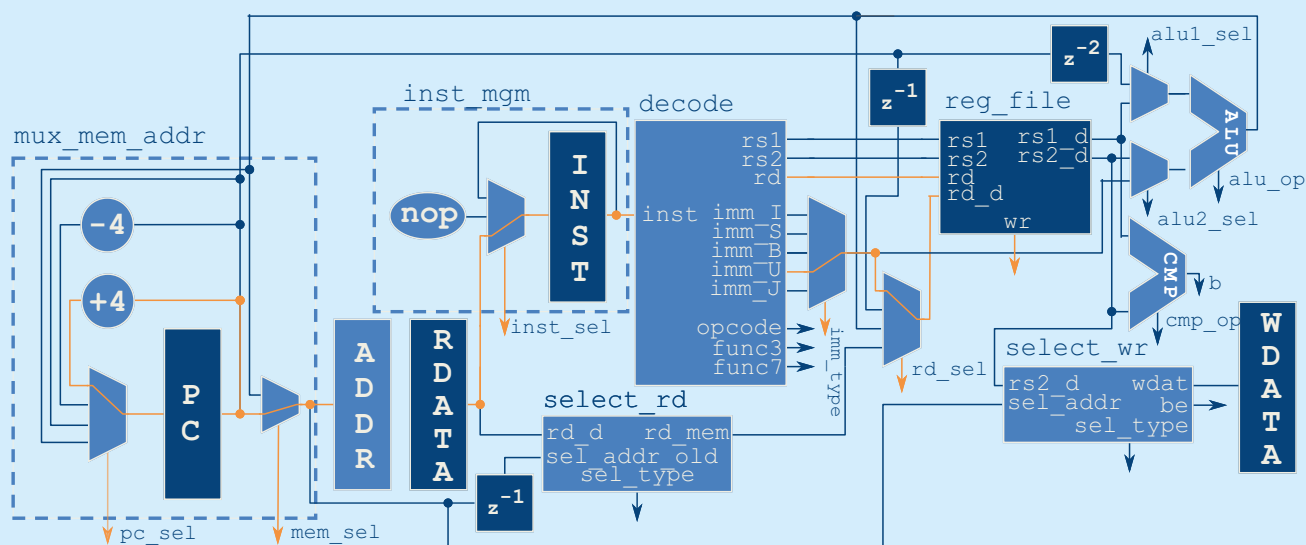
zdarzenia w naszym rdzeniu są wywoływane jego narastającym zboczem. W drugim wierszu przedstawiono *rst*, odpowiadający za reset mikrokontrolera. Jest on aktywowany stanem wysokim. Jego zdjęcie następuje w 2,5 μ s symulacji. Praca procesora rozpoczyna się od następnego zbocza narastającego, czyli w 3 μ s. Kolejna linia zawiera wartość licznika programu PC. Sygnał *q* odpowiada wartości odczytanej z pamięci. Piątym jest *inst*, czyli rejestr przechowujący kod aktualnie interpretowanej instrukcji. Ostatnie trzy wiersze zawierają wartości rejestrów *x1*, *x2* i *x3*. Zielonym kolorem zaznaczone są sygnały ustalone, a czerwonym takie, które nie zostały zainicjowane. Ponieważ reset nie blokuje odczytu z pamięci, kod pierwszej instrukcji znalazł się w rejestrze *q* już po 2 μ s. Jednakże dopiero między 2 i 3 μ s pierwszy rozkaz wchodzi w fazę ustaw adres. Pomiędzy 3 i 4 μ s rejestr PC wskazuje już adres następnej instrukcji, a rejestr *q* zawiera kod początkowego rozkazu. Wykonanie pierwszej instrukcji następuje pomiędzy 4 i 5 μ s, a jej wynik jest zapisywany na następnym narastającym zboczach zegara (5 μ s). Widzimy, że w tym momencie rejestr *x1* przyjmuje wartość 10. W analogiczny sposób przez potok przepływają pozostałe dwie instrukcje. Na piątym zboczach zegara (7 μ s) następuje zapis efektu ostatniej instrukcji.

Są jeszcze dwie instrukcje pozwalające na wczytanie danych do rejestrów. Pierwsza z nich *lui* pozwala wczytać 20 starszych bitów rejestru. Jej przepływ danych przedstawia **rysunek 27**. Jest ona znacznie prostsza, gdyż następuje jedynie wybór stałej oraz rejestru docelowego.

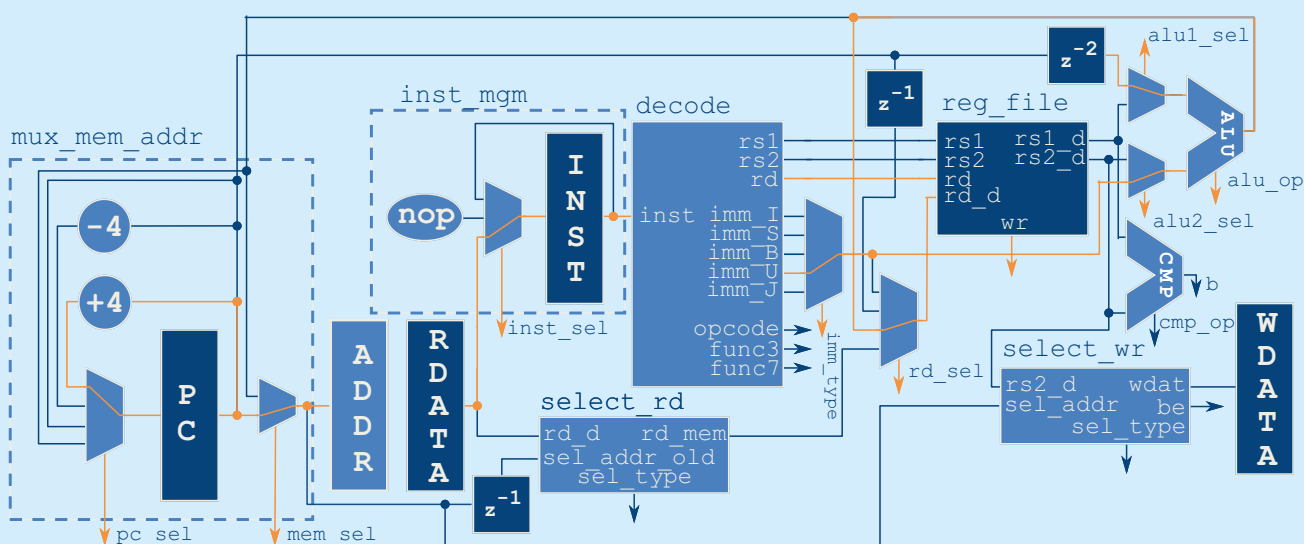
Druga z nich to *auipc*, pokazana na **rysunku 28**. Zapisuje ona do rejestru sumę adresu instrukcji oraz stałej bezpośredniej. I tu widzimy po co niezbędny jest blok opóźniający o dwa takty zegara z^{-2} . Rejestr

Listing 11. Program z rysunku 22 w postaci binarnej

```
01 00a00093 // addi x1 x0 10    addi x1, x0, 10
02 00500113 // addi x2 x0 5     addi x2, x0, 5
03 001101b3 // add x3 x2 x1     add x3, x2, x1
04 00000013 // addi x0 x0 0      nop
05 00000013 // addi x0 x0 0      nop
```



Rysunek 27. Przepływ danych dla instrukcji lui



Rysunek 28. Przepływ danych dla instrukcji auipc

PC wskazuje adres instrukcji *auipc* dwa takty przed fazą wykonania, w której nastąpi jej wykonanie. Dodane opóźnienie powoduje, że odpowiednia wartość płynie przez potok razem z kodem rozkazu. Zachęcam czytelnika do samodzielnego rozpisania stanu procesora dla kolejnych taktów zegara, a następnie przygotowania podobnej symulacji jak dla wcześniejszych rozkazów i porównania wyników.

Podsumowanie

Mam nadzieję, że udało mi się zainteresować czytelników tematem tworzenia własnych mikrokontrolerów. W kolejnym artykule dokładnie przeanalizujemy działanie naszego rdzenia dla operacji skoków oraz zapisu/odczytu do/z pamięci i w końcu poruszymy tematykę sprzętu.

Rafał Kozik
rafkozik@gmail.com

Bibliografia

- [1] <http://bit.ly/2MFFapx>, dostęp na dzień 29.04.2019
- [2] <http://bit.ly/2YP0EGY>, dostęp na dzień 29.04.2019
- [3] Zeloof S., First IC : <http://bit.ly/33dwwEx> dostęp na dzień 29.04.2019

- [4] <http://bit.ly/2MHjQQL>
- [5] Computer Structures, MIT 6.004, Spring 2019, <http://bit.ly/33dx4dL>
- [6] MIT 6.004 RISC-V ISA Reference Card, <http://bit.ly/2TcOS43>
- [7] The RISC-V Instruction Set Manual, Volume I: User-Level ISA, Document Version 2.2, Editors Andrew Waerman and Krste Asanovic, RISC-V Foundation, May 2017
- [8] <http://bit.ly/31jY2OV>
- [9] <http://bit.ly/2yDgVjP>
- [10] Quartus Prime Lite Edition, <https://intel.ly/2MC6TYp>
- [11] Ataki na procesory – PortSmash, TLBleed, Foreshadow, <http://bit.ly/2M307eL>
- [12] Patterson D., Hennessy J., Computer Organization and Design: The Hardware Software Interface (RISC-V Edition), Morgan Kaufmann, 1st ed, 2017,
- [13] MAXimator Altera MAX10 FPGA Evaluation Board, <http://bit.ly/2YJneNp>
- [14] Great Ideas in Computer Architecture (Machine Structures), CS61C Berkeley with Nicholas Weaver, Spring 2019
- [15] Moduł dla Visual Studio Code <http://bit.ly/2YP0Jug>

AudioDSP zestaw z procesorem Sigma DSP ADAU1701 (7)



Konfiguracja DSP procesorem zewnętrznym

To już ostatnia część kursu AudioDSP... Cieszę się że spotkał się on ze sporym odzewem, jednocześnie gorąco zachęcam do eksperymentów z AudioDSP we własnym zakresie.

Siódma część kursu poświęcona jest sposobom konfiguracji DSP przy zastosowaniu zewnętrznego procesora. Zestaw AudioDSP z procesorem ADAU1701 umożliwia pracę samodzielną bez zewnętrznego kontrolera, odbywać może się to na dwa sposoby:

1. Pamięć programu i parametry aplikacji ładowane są bezpośrednio poprzez programator USBi, a interfejs użytkownika zrealizowany przez GUI Sigma Studio. Metoda skuteczna, ale dosyć niepraktyczna przy aplikacjach DSP w samodzielnym urządzeniach, gdyż każdorazowo trzeba ładować program korzystając z IDE i trzeba posiadać programator USBi, dedykowany w zasadzie dla każdego obsługiwanego urządzenia. Przykładem tak zrealizowanego urządzenia może być prototypowy moduł dla ramy API500, którego konfiguracja jest zmieniana w zależności od wymogów obróbki sygnału w studiu nagraniowym. Moduł posiada jedno wejście XLR i jedno wyjście XLR oraz wbudowany programator USBi. Realizowana funkcja zmieniana jest poprzez załadowanie przygotowanej wcześniej aplikacji i parametryzację działania w czasie rzeczywistym z SigmaStudio.
2. Pamięć programu i parametry odczytywane są z pamięci E2PROM w trybie Selfboot, a nastawy parametrów realizowane są poprzez obsługę manipulatorów podłączonych do GPIO. Ułatwia to konstruowanie urządzeń pracujących samodzielnie. Pamięć programu zapisywana jest tylko raz przy użyciu programatora USBi, a wszelkie zmiany parametrów dzięki mechanizmowi WriteBack zapisywane są w pamięci E2PROM w trakcie detekcji zaniku zasilania DSP. Ta metoda także posiada ograniczenia np.: w postaci skończonej liczby wyprowadzeń GPIO procesora ADAU1701, braku przystosowania niektórych bloków do sterowania sygnałem GPIO, oraz niemożność użycia cyfrowych interfejsów szeregowych audio

blokowanych przez funkcje GPIO (lub odwrotnie). W ten sposób zrealizowane były dotychczasowe projekty z zastosowaniem procesorów Sigma DSP, publikowane w EP.

Gdy powyższe ograniczenia stają się uciążliwe ADAU1701 oferuje możliwość kontroli przez zewnętrzny procesor sterujący, nie jest to już tak łatwe jak współpraca z SigmaStudio, ale jest oczywiście całkowicie wykonalne, gdyż oprogramowanie generuje pliki niezbędne do szybkiej aplikacji DSP, bez względu na zastosowany procesor konfigurujący (wspierany jest język C). Tutaj, także możliwe są dwa sposoby kontroli używające wbudowanych w DSP interfejsów I²C lub SPI. AudioDSP bazuje na interfejsie I²C, używany jest do konfiguracji i współpracy z programatorem USBi poprzez złącze USBi (IDC10), magistrala I²C dostępna jest też na dedykowanym złączu I²C (KK4) oraz na złączu zewnętrznego modułu sterującego uPC. Obsługa przez interfejs SPI przebiega podobnie.

UWAGA: interfejs I²C w module AudioDSP pracuje w standardzie 3,3 V, doprowadzenie wyższego napięcia nieodwracalnie uszkodzi DSP i współpracujące układy.

Pierwszym z możliwych sposobów konfiguracji zewnętrznej jest przełączenie ADAU1701 w tryb I²C (*Selfboot* = 0) i konfiguracja ADAU1701 DSP z zewnętrznego procesora. W trybie tym pamięć E2PROM jest nieużywana, wybór adresu bazowego ADAU1701 określony jest stanem wyprowadzeń ADDR0/1 w AudioDSP. Jeśli stan na wyprowadzeniach wynosi „0”, to adres jest równy 0x34. DSP jest urządzeniem *slave* magistrali I²C. Zawartość pamięci RAM aplikacji, parametrów i rejestrów konfiguracji sprzętowej ładowana jest poprzez I²C bezpośrednio do DSP z pamięci mikrokontrolera sterującego. Należy mieć świadomość rozmiaru i ilości danych wymaganych do przesłania z procesora do DSP, która dla ADAU1701 wynosi odpowiednio:

Capture						
Mode	Time	Parameter Name	Address	V.	Data	Bytes
Block Write	20:55:55 - 540ms	IC 1.CoreRegister	0x081C		0x00, 0x00	2
Block Write	20:55:55 - 550ms	Program Data	0x0400		0x00, 0x19, 0x1F, 0x20, 0x01,	5120
Block Write	20:55:55 - 550ms	Param	0x0000		0x00, 0x00, 0x00, 0xFF,	4096
Block Write	20:55:55 - 560ms	IC 1.HWConFfiguration	0x081C		0x00, 0x00, 0x08, 0x08,	24
Block Write	20:55:55 - 560ms	IC 1.CoreRegister	0x081C		0x00, 0x04	2

Rysunek 1. Podgląd wartości ładowanych do DSP po kompilacji

S	CHIP ADDRESS, R/W = 0	AS	SUBADDRESS HIGH	AS	SUBADDRESS LOW	AS	DATA BYTE 1	AS	DATA BYTE 2	...	AS	DATA BYTE N	P
---	--------------------------	----	--------------------	----	-------------------	----	-------------	----	-------------	-----	----	-------------	---

Rysunek 2. Zapis w trybie single write

S	CHIP ADDRESS, R/W = 0	AS	SUBADDRESS HIGH	AS	SUBADDRESS LOW	AS	DATA- WORD 1, BYTE 1	AS	DATA- WORD 1, BYTE 2	AS	DATA- WORD 2, BYTE 1	AS	DATA- WORD 2, BYTE 2	AS	...	P
---	--------------------------	----	--------------------	----	-------------------	----	----------------------------	----	----------------------------	----	----------------------------	----	----------------------------	----	-----	---

Rysunek 3. Zapis w trybie Burst

S	CHIP ADDRESS, R/W = 0	AS	SUBADDRESS HIGH	AS	SUBADDRESS LOW	AS	S	CHIP ADDRESS, R/W = 1	AS	DATA BYTE 1	AM	DATA BYTE 2	...	AM	DATA BYTE N	P
---	--------------------------	----	--------------------	----	-------------------	----	---	--------------------------	----	----------------	----	----------------	-----	----	----------------	---

Rysunek 4. Odczyt w trybie Single Read

S	CHIP ADDRESS, R/W = 0	AS	SUBADDRESS HIGH	AS	SUBADDRESS LOW	AS	S	CHIP ADDRESS, R/W = 1	AS	DATA- WORD 1, BYTE 1	AM	DATA- WORD 1, BYTE 2	AM	...	P
---	--------------------------	----	--------------------	----	-------------------	----	---	--------------------------	----	----------------------------	----	----------------------------	----	-----	---

Rysunek 5. Odczyt w trybie Burst

- 2 bajty dla rejestrów rdzenia *CoreRegister*, na czas ładowania parametrów następuje wyłączenie rdzenia,
- 4096 bajtów dla parametrów *Param*, 1024×32 bity, od adresu 0x0000 do 0x03FF,
- 5120 bajtów dla pamięci programu *Program Data*, 1024×40 bitów (5-bajtów, aby można było skonfigurować filtry, pozostałe bloki wymagają mniejszej liczby bajtów do konfiguracji), od adresu 0x0400 do 0x07FF,
- 24 bajty konfiguracji sprzętowej *HWConFiguration*, od adresu 0x0800 do 0x0827, w tym rejestry *Safeload* i rejestry zarezerwowane,
- 2 bajty, wyłączenie wyciszania rdzenia po zaprogramowaniu pamięci.

Ilość danych i ich wartości przesyłane do DSP można podglądać w okienku Capture po kompilacji i załadowaniu aplikacji, co pokazano na **rysunku 1**. Warto zwrócić uwagę na szybkość interfejsu magistrali I²C wbudowanego w procesor sterujący, ADAU1701 obsługuje maksymalnie 400 kHz, załadowanie całej pamięci zajmuje więc niezerowy czas...

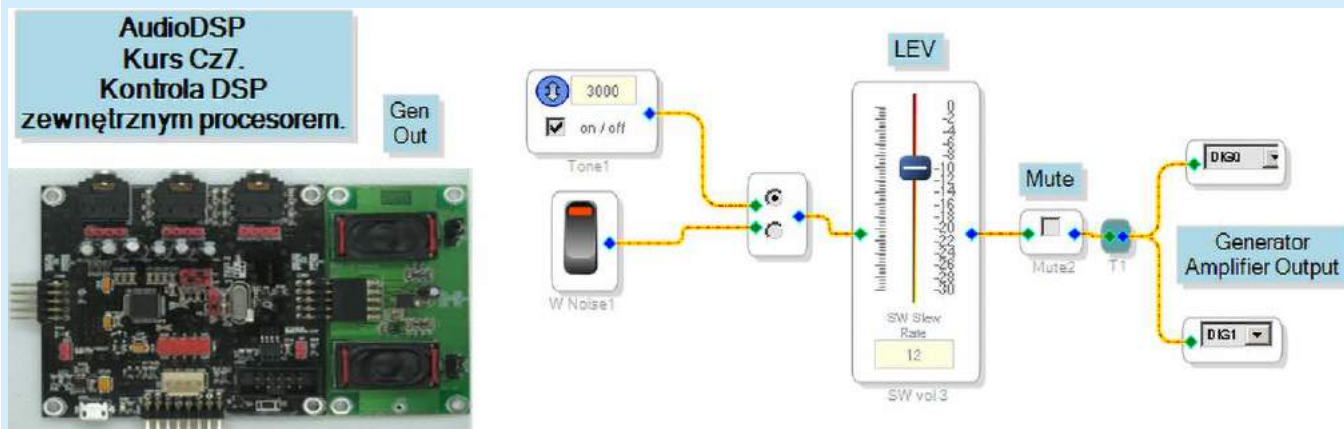
Drugi sposób to przechowanie aplikacji i parametrów w pamięci E2PROM w trybie *Selfboot* (*Selfboot* = 1) i zmiana tylko wybranych parametrów aplikacji przez magistralę I²C. Znacząco ograniczona zostaje ilość przesyłanych danych, co może być przydatne, gdy do dyspozycji mamy mikrokontroler z ograniczonymi zasobami. W trybie tym szczególną uwagę należy zwrócić na obsługę magistrali I²C, gdyż podczas ładowania danych z pamięci oraz podczas cyklu Writeback, DSP jest urządzeniem master magistrali I²C, co może powodować konflikty, gdy jednocześnie aktywny będzie drugi master magistrali.

ADAU1701 umożliwia bezpośredni zapis rejestrów, gdy dane wpisywane są w trakcie działania DSP. Pomimo prostoty rozwiązanie takie nie jest polecane, ze względu na możliwe pojawienie

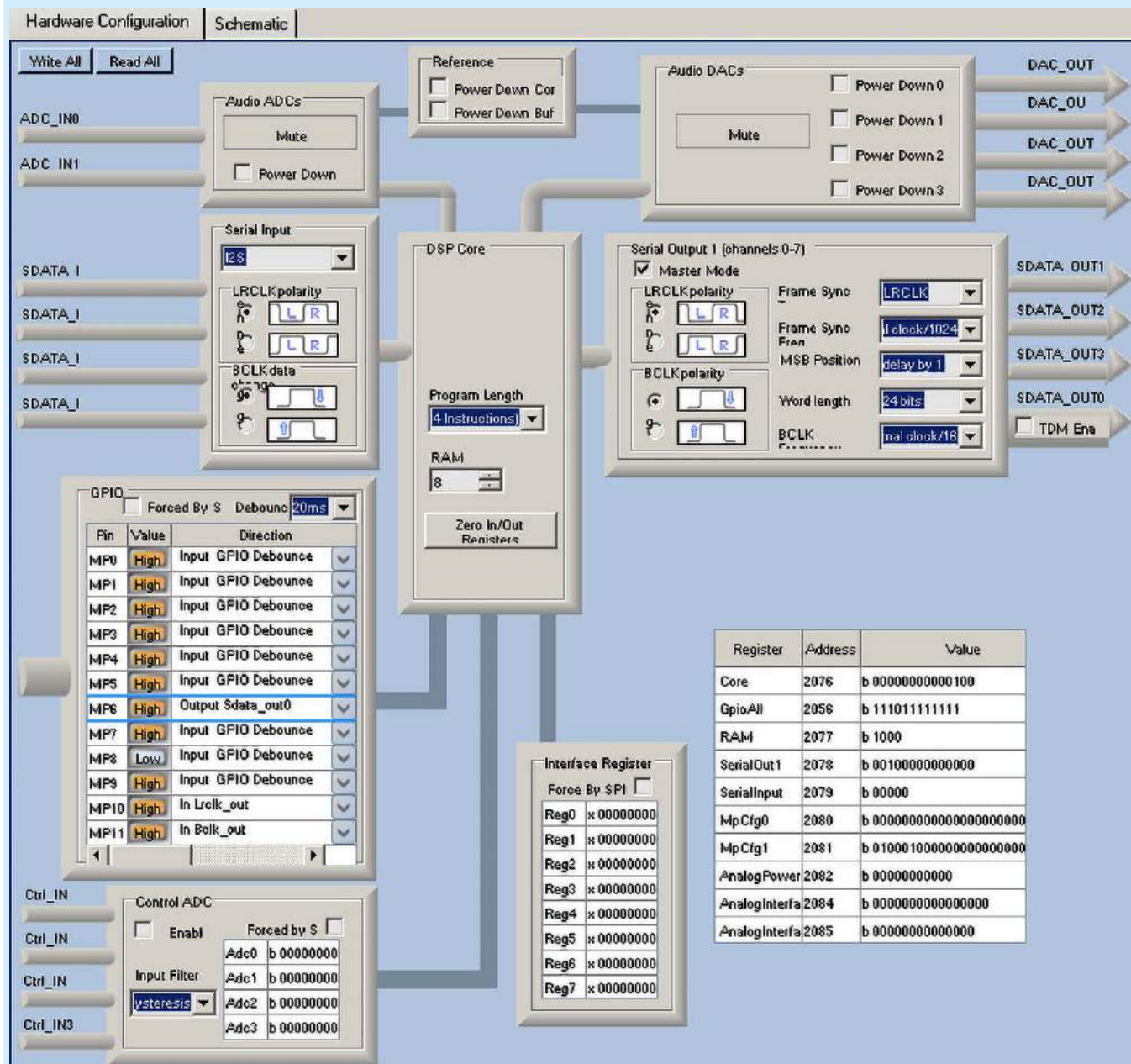
się artefaktów w obrabianym sygnale audio. Szczególnie istotne jest to, gdy do bloku funkcjonalnego musi być wpisana większa ilość danych np. przy zmianie parametrów filtrów, ustawień kompresorów itp. Łatwo sobie wyobrazić co stanie się z sygnałem obrabianym, gdy podczas zapisu tylko część współczynników filtra będzie zmieniona na nowe... Aby zapobiec artefaktom ADAU1701 został wyposażony w mechanizm *Safeload*. Jest to specjalny sposób zapisu parametrów do pracującego DSP. Dla jego realizacji DSP posiada dedykowane pośrednie rejestry *Safeload Data Register* (adresy 0x0810 do 0x0814) umożliwiające zapisanie pięciu 28-bitowych danych (w rzeczywistości zapisywane są 32-bity ze względu na konieczność przesłania całego bajtu na magistrali I²C). Adres, pod który zostanie zapisany 5 bajtowy ciąg wartości określają rejestry *Safeload Address Register* (adresy 0x0815 do 0x0819). Mapowanie adresów rejestrów jest liniowe, rejestr danych 0x0810 zapisany zostanie pod adres wskazany w rejestrze 0x0815 itd, rejestr ten zaś musi być skojarzony ze blokiem którego parametry są modyfikowane.

ADAU1701 obsługuje dwa tryby magistrali I²C, pierwszy klasyczny z zapisem jednokrotnym (*Single Write*), którego ramkę pokazuje **rysunek 2**. Drugi tryb to tryb z automatycznym zwiększaniem adresu (*Burst Mode*), którego ramkę przedstawiono na **rysunku 3**. W trybie tym DSP po odczytaniu podadresu, automatycznie zwiększa licznik adresowy i kolejne słowa zapisuje pod następny adres, zmniejsza to ilość przesyłanych danych i przyspiesza zapis dużej ilości danych do DSP. W przypadku odczytu sytuacja jest podobna, ramkę w trybie odczytu pojedynczego (*Single Read*) przedstawiono na **rysunku 4**, w trybie z automatycznym zwiększaniem adresu na **rysunku 5**.

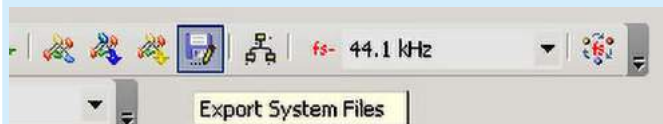
Niezależnie od wybranego sposobu konfiguracji zewnętrznej DSP przed rozpoczęciem tworzenia oprogramowania dla procesora sterującego, niezbędne jest stworzenie i przetestowanie aplikacji



Rysunek 6. Aplikacja generatora Generator_I2C.dspproj



Rysunek 7. Konfiguracja DSP



Rysunek 8. Eksport plików systemowych

dla DSP. Dla przykładu niech będzie to prosty generator przebiegu sinusoidalnego lub szumu, kształt przebiegu wybierany jest multiplekserem, poziom sygnału regulowany jest blokiem regulatora poziomu, istnieje możliwość wyciszenia wyjścia, sygnał generatora dostępny jest na wyjściu DIG0/1 i doprowadzony jest do modułu DSP1701A_DAMP2518S_K1425_HW. Schemat aplikacji Generator_I2C.dspproj pokazano na **rysunku 6**, a konfigurację procesora DSP na **rysunku 7**. Po wykonaniu kompilacji i ładowania projektu opcją *Link Compile Download* – klawisz F7, aktywna staje się ikona *Export System Files*, dzięki której możemy wyeksportować pliki konfiguracyjne niezbędne do stworzenia aplikacji sterującej DSP, **rysunek 8**. Po wyborze katalogu zapisane zostaną w nim pliki z wyeksportowanymi danymi konfiguracyjnymi, co przedstawiono na **rysunku 9**. Pliki zapisane są w formacie tekstowym i można otworzyć je w dowolnym edytorze

tekstowym, co jest niezbędne do znalezienia parametrów i adresów koniecznych do zintegrowania DSP z kodem mikrokontrolera sterującego. Pliki te zawierają nagłówki zgodne z C/C++ określające parametry projektu i używane rejestry DSP.

Podczas eksportu zostanie wygenerowanych kilka typów plików:

- *.params – zawiera wszystkie nazwy parametrów aplikacji DSP, adresy i wartości do zapisania w pamięci,

- *.hex – zawiera wartości bajtów pamięci parametrów w formacie szesnastkowym, dane te są również odzwierciedlone w polu *Parameter Data* pliku *.params.

Nazwa	Roz.	Wielkość	Czas	Atryb
...		<DIR>	2019-09-04 21:34	----
TxBuf IC_1	dat	59 780	2019-09-04 21:34	-a--
NumBytes IC_1	dat	27	2019-09-04 21:34	-a--
Generator_I2C_NetList	xml	2 413	2019-09-04 21:34	-a--
Generator_I2C_IC_1_REG	h	29 380	2019-09-04 21:34	-a--
Generator_I2C_IC_1_PARAM	h	4 782	2019-09-04 21:34	-a--
Generator_I2C_IC_1	h	62 136	2019-09-04 21:34	-a--
Generator_I2C	xml	62 391	2019-09-04 21:34	-a--
Generator_I2C	params	75 652	2019-09-04 21:34	-a--
Generator_I2C	hex	30 722	2019-09-04 21:34	-a--
defines	h	910	2019-09-04 21:34	-a--

Rysunek 9. Zawartość katalogu z plikami systemowymi

Mode	Time	Cell Name	Parameter Name	Address	Value	Data	Bytes
Safeload Write	22:6:52 - 482ms	SW vol 3	SWGain1940DB...	0x0006	0.28183...	0x00, 0x24, 0x13, 0x47	4
Safeload Write	22:6:52 - 482ms	SW vol 3	SWGain1940DB...	0x0007	0.00024...	0x00, 0x00, 0x08, 0x00	4
Safeload Write	22:6:52 - 862ms	SW vol 3	SWGain1940DB...	0x0006	0.25118...	0x00, 0x20, 0x26, 0xF3	4
Safeload Write	22:6:52 - 862ms	SW vol 3	SWGain1940DB...	0x0007	0.00024...	0x00, 0x00, 0x08, 0x00	4
Safeload Write	22:6:52 - 992ms	SW vol 3	SWGain1940DB...	0x0006	0.22387...	0x00, 0x1C, 0xA7, 0xD7	4
Safeload Write	22:6:52 - 992ms	SW vol 3	SWGain1940DB...	0x0007	0.00024...	0x00, 0x00, 0x08, 0x00	4
Safeload Write	22:6:53 - 72ms	SW vol 3	SWGain1940DB...	0x0006	0.19952...	0x00, 0x19, 0x8A, 0x13	4
Safeload Write	22:6:53 - 72ms	SW vol 3	SWGain1940DB...	0x0007	0.00024...	0x00, 0x00, 0x08, 0x00	4

Rysunek 10. Przykładowe parametry bloku SW vol3 – zmiana głośności

Mode	Time	Cell Name	Parameter Name	IC
Safeload Write	22:18:8 - 534ms	Tone1	sin_lookupAlg1...	IC 1
Safeload Write	22:18:8 - 534ms	Tone1	sin_lookupAlg1...	IC 1
Safeload Write	22:18:8 - 534ms	Tone1	sin_lookupAlg1...	IC 1
Safeload Write	22:18:11 - 944ms	Tone1	sin_lookupAlg1...	IC 1
Safeload Write	22:18:11 - 944ms	Tone1	sin_lookupAlg1...	IC 1
Safeload Write	22:18:11 - 944ms	Tone1	sin_lookupAlg1...	IC 1
Block Write	22:18:15 - 704ms	monoMuxSwSle...	IC 1	

Rysunek 11. Eksport sekwencji do pliku

Mode	Time	Cell Name	Parameter Name	IC
Safeload Write	22:18:8 - 534ms	Tone1	sin_lookupAlg1...	IC 1
Safeload Write	22:18:8 - 534ms	Tone1	sin_lookupAlg1...	IC 1
Safeload Write	22:18:8 - 534ms	Tone1	sin_lookupAlg1...	IC 1
Safeload Write	22:18:11 - 944ms	Tone1	sin_lookupAlg1...	IC 1
Safeload Write	22:18:11 - 944ms	Tone1	sin_lookupAlg1...	IC 1
Safeload Write	22:18:11 - 944ms	Tone1	sin_lookupAlg1...	IC 1
Block Write	22:18:15 - 704ms	monoMuxSwSle...	IC 1	

Rysunek 12. Zapis sekwencji do DSP

*.h – które definiują nazwy, adresy i wartości wszystkich parametrów aplikacji DSP,

*_REG.h – plik nagłówkowy zawiera definicje wszystkich rejestrów sprzętowych wraz z wartościami.

Oczywiście po każdej zmianie algorytmu, konieczna jest ponowna generacja kompletu plików konfiguracyjnych. Aby przyspieszyć tworzenie aplikacji, gdy konfiguracja DSP jest już określona i sprawdzona, a zależy nam tylko na szybkim wyznaczeniu zmienianych parametrów bloków funkcjonalnych, należy wykorzystać okienko *Capture* z rysunku 1. Każdorazowo po zmianie parametru, Sigma Studio zapisuje rejestry DSP odpowiednio przeliczonymi wartościami. Dla przykładu zmiana głośności bloku SW vol3, która zapisana jest w pliku *.params przedstawia się następująco:

Cell Name = SW vol 3
 Parameter Name = SWGain1940DBAlg3target
 Parameter Address = 6
 Parameter Value = 0.316227793693542
 Parameter Data : 0x00 , 0x28 , 0x7A , 0x27

Cell Name = SW vol 3
 Parameter Name = SWGain1940DBAlg3step
 Parameter Address 7
 Parameter Value = 0.000244140625
 Parameter Data : 0x00 , 0x00 , 0x08 , 0x00

Przykładową zmianę poziomu sygnału i odpowiadające jej zapisywane wartości możliwe do sprawdzenia w okienku *Capture* (tryb *Safeload*) pokazano na **rysunku 10**.

Funkcje pomocne w przygotowaniu wartości parametrów konfiguracyjnych są dostępne pod prawym klawiszem myszy. Oprócz eksportu i kopiowania danych w różnych formatach, istotną funkcją jest *Add to sequence*, umożliwiającą przygotowanie sekwencji zmienianych parametrów. Dla przykładu niech będzie to zmiana częstotliwości generatora sinusoidy z 250 Hz na 4 kHz, a następnie przełączenie się na generator szumu. Oczywiście do sekwencji można dopisać pozostałe zmiany parametrów bloków. Możliwe

jest stworzenie kilkunastu sekwencji oraz zapisanie ich plików konfiguracyjnych na dysk ikoną *Export Sequence Data*, co pokazano na **rysunku 11** lub zapisanie ich do pamięci DSP wg **rysunku 12**. Podobnie jak w przypadku eksportu konfiguracji, podczas eksportu sekwencji tworzone są pliki *.c, *.h, *.xml zawierające definicje i wartości parametrów. Przykładową zapisaną sekwencję *Mode 0* przedstawia **listing 1**. Po przygotowaniu wszystkich sekwencji wykorzystywanych do parametryzacji aplikacji DSP, można rozpocząć proces żmudnego tworzenia oprogramowania dla procesora sterującego. Przykładowy listing realizujący sekwencję *Mode 0* dla Arduino MKR (I²C – 3,3 V!), w trybie *Selfboot* ADAU1701 pokazano na **listingu 2**.

Podobny zestaw instrukcji można zrealizować np. dla Raspberry Pi korzystając z poleceń systemowych *i2cset*, a po zapisaniu ich w pliku wykonywalnego skryptu, można parametryzować DSP z linii poleceń. W przypadku eksperymentów z Pi warto zaprogramować EEPROM na płycie AudioDSP aplikacją Generator_I2C, tak jak dla pracy samodzielnej. W pierwszej kolejności po podłączeniu AudioDSP do Pi sprawdzamy czy zostało prawidłowo rozpoznane na magistrali I²C poleceniem *i2cdetect -y 1*. Spotka nas niespodzianka, ADAU1701 nie jest wykrywany, pod adresem 0x50 znaleziona jest pamięć 24LC256 z AudioDSP, adres 0x51 to pamięć EEPROM programatora USBi, pod który to adres pod żadnym pozorem nie należy zapisywać, gdyż nieodwracalnie uszkodzimy w ten sposób firmware programatora. Adresy

Listing 1. Przykładowa sekwencja Mode 0

```

Safeload Write
Time: 22:18:8 - 534ms
IC: IC 1
Cell Name: Tone1
Param Name: sin_lookupAlg19401mask
Param Address: 0x0000
Param Value: 255
Bytes: 4
Param Data:      0x00,      0x00,      0x00,      0xFF

Safeload Write
Time: 22:18:8 - 534ms
IC: IC 1
Cell Name: Tone1
Param Name: sin_lookupAlg19401increment
Param Address: 0x0001
Param Value: 0.0113378763198853
Bytes: 4
Param Data:      0x00,      0x01,      0x73,      0x85

Safeload Write
Time: 22:18:8 - 534ms
IC: IC 1
Cell Name: Tone1
Param Name: sin_lookupAlg19401ison
Param Address: 0x0002
Param Value: 1
Bytes: 4
Param Data:      0x00,      0x80,      0x00,      0x00

Safeload Write
Time: 22:18:11 - 944ms
IC: IC 1
Cell Name: Tone1
Param Name: sin_lookupAlg19401mask
Param Address: 0x0000
Param Value: 255
Bytes: 4
Param Data:      0x00,      0x00,      0x00,      0xFF

Safeload Write
Time: 22:18:11 - 944ms
IC: IC 1
Cell Name: Tone1
Param Name: sin_lookupAlg19401increment
Param Address: 0x0001
Param Value: 0.181405901908875
Bytes: 4
Param Data:      0x00,      0x17,      0x38,      0x4F

Safeload Write
Time: 22:18:11 - 944ms
IC: IC 1
Cell Name: Tone1
Param Name: sin_lookupAlg19401ison
Param Address: 0x0002
Param Value: 1
Bytes: 4
Param Data:      0x00,      0x80,      0x00,      0x00

Block Write
Time: 22:18:15 - 704ms
IC: IC 1
Param Name: monoMuxSwSlew1coeffname
Param Address: 0x0005
Param Value: 1
Bytes: 4
Param Data:      0x00,      0x00,      0x00,      0x01

```

Listing 2. Listing realizujący sekwencję Mode 0 dla Arduino MKR

```

#include <Wire.h>
//ADAU1701 sequence Mode0

void setup()
{
    Wire.begin(); // join i2c bus

    // -24dB volume SW vol3 safeload 4 bytes
    Wire.beginTransmission(0x34); // ADAU1701 I2C A0,A1=0
    Wire.write(0x00); // high address
    Wire.write(0x06); // low address
    Wire.write(0x00);
    Wire.write(0x08);
    Wire.write(0x13);
    Wire.write(0x85);
    Wire.endTransmission();
    delay(1000);

    Wire.beginTransmission(0x34);
    Wire.write(0x00);
    Wire.write(0x07);
    Wire.write(0x00);
    Wire.write(0x00);
    Wire.write(0x08);
    Wire.write(0x00);
    Wire.write(0x00);
    Wire.endTransmission();
    delay(100);

    // mute off Mute2 Block write
    Wire.beginTransmission(0x34);
    Wire.write(0x00);
    Wire.write(0x08);
    Wire.write(0x00);
    Wire.write(0x80);
    Wire.write(0x00);
}

```

Listing 2. cd.

```

Wire.write(0x00);
Wire.endTransmission();
delay(100);
}

void loop()
{
    // mux noise Block write
    Wire.beginTransmission(0x34);
    Wire.write(0x00);
    Wire.write(0x05);
    Wire.write(0x00);
    Wire.write(0x00);
    Wire.write(0x00);
    Wire.write(0x00);
    Wire.write(0x01);
    Wire.endTransmission();
    delay(1000);

    // 250Hz sin safeload 4 bytes x 3
    Wire.beginTransmission(0x34);
    Wire.write(0x00); // mask
    Wire.write(0x00);
    Wire.write(0x00);
    Wire.write(0x00);
    Wire.write(0x00);
    Wire.write(0x00);
    Wire.write(0xFF);
    Wire.endTransmission();

    Wire.beginTransmission(0x34);
    Wire.write(0x00); // increment
    Wire.write(0x01);
    Wire.write(0x00);
    Wire.write(0x01);
    Wire.write(0x00);
    Wire.write(0x73);
    Wire.write(0x85);
    Wire.endTransmission();

    Wire.beginTransmission(0x34);
    Wire.write(0x00); // on
    Wire.write(0x02);
    Wire.write(0x00);
    Wire.write(0x80);
    Wire.write(0x00);
    Wire.write(0x00);
    Wire.write(0x00);
    Wire.endTransmission();
    delay(1000);

    // mux sin
    Wire.beginTransmission(0x34);
    Wire.write(0x00);
    Wire.write(0x05);
    Wire.write(0x00);
    Wire.write(0x00);
    Wire.write(0x00);
    Wire.write(0x00);
    Wire.write(0x00);
    Wire.endTransmission();
    delay(1000);

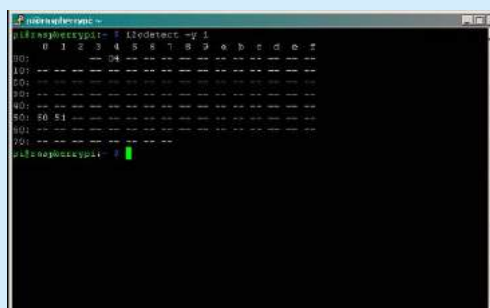
    // 4kHz safeload 4 bytes x 3
    Wire.beginTransmission(0x34);
    Wire.write(0x00); // mask
    Wire.write(0x00);
    Wire.write(0x00);
    Wire.write(0x00);
    Wire.write(0x00);
    Wire.write(0x00);
    Wire.write(0xFF);
    Wire.endTransmission();

    Wire.beginTransmission(0x34);
    Wire.write(0x00); // increment
    Wire.write(0x01);
    Wire.write(0x00);
    Wire.write(0x00);
    Wire.write(0x17);
    Wire.write(0x38);
    Wire.write(0x4F);
    Wire.endTransmission();

    Wire.beginTransmission(0x34);
    Wire.write(0x00); // on
    Wire.write(0x02);
    Wire.write(0x00);
    Wire.write(0x80);
    Wire.write(0x00);
    Wire.write(0x00);
    Wire.write(0x00);
    Wire.endTransmission();

    Wire.endTransmission();
    delay(1000);
}

```

**Rysunek 13. Detekcja układów I²C**

```

Listing 3. Format komendy i2cset
mute off
i2cset -y 1 0x34 0x00 0x08 0x00 0x80 0x00 0x00 i

mute on
i2cset -y 1 0x34 0x00 0x08 0x00 0x00 0x00 0x00 i

mux noise
i2cset -y 1 0x34 0x00 0x05 0x00 0x00 0x00 0x01 i

mux sin
i2cset -y 1 0x34 0x00 0x05 0x00 0x00 0x00 0x00 i

vol -24db
i2cset -y 1 0x34 0x00 0x06 0x00 0x08 0x13 0x85 i
i2cset -y 1 0x34 0x00 0x07 0x00 0x00 0x08 0x00 i

sin 4kHz
i2cset -y 1 0x34 0x00 0x00 0x00 0x00 0x00 0xFF i
i2cset -y 1 0x34 0x00 0x01 0x00 0x17 0x38 0x4F i
i2cset -y 1 0x34 0x00 0x02 0x00 0x80 0x00 0x00 i

sin 250Hz
i2cset -y 1 0x34 0x00 0x00 0x00 0x00 0x00 0xFF i
i2cset -y 1 0x34 0x00 0x01 0x00 0x01 0x73 0x85 i
i2cset -y 1 0x34 0x00 0x02 0x00 0x80 0x00 0x00 i

```

po detekcji pokazano na **rysunku 13**. Dla bezpośredniego zapisu do DSP można użyć polecenia systemowego *i2cset*, dla magistrali pierwszej skonfigurowanego dla zapisu adresu 16 bitowego oraz ciągu danych według plików generowanych przez Sigma Studio. Format komendy *i2cset* dla poleceń wygląda jak na **listingu 3**. Polecenia można dla usprawnienia wpisać w skrypt testowy, który np. wykona zmianę 5 krotnie i zakończy pracę, listing przykładowego skryptu *ADAU1701.sh* pokazuje na **listing 4**.

Na zakończenie warto wspomnieć o dostępnych w sieci bibliotekach dla Arduino, np. <http://bit.ly/2kUDgpx>, które mogą zostać wykorzystane lub stanowić wzór dla własnych implementacji sterowania ADAU1701.

```

Listing 4. Kod skrypu ADAu1701.sh
#!/bin/bash
echo 'ADAu1701 Mode 0'
echo 'vol -24db'
i2cset -y 1 0x34 0x00 0x06 0x00 0x08 0x13 0x85 i
i2cset -y 1 0x34 0x00 0x07 0x00 0x00 0x08 0x00 i
echo 'mute off'
i2cset -y 1 0x34 0x00 0x08 0x00 0x80 0x00 0x00 i
echo 'generator i2c'

x=1;

while [ $x -le 5 ] ; do

# mux noise
i2cset -y 1 0x34 0x00 0x05 0x00 0x00 0x00 0x01 i
sleep 1.0

# mux sin
i2cset -y 1 0x34 0x00 0x05 0x00 0x00 0x00 0x00 i

# sin 250Hz
i2cset -y 1 0x34 0x00 0x00 0x00 0x00 0x00 0xFF i
i2cset -y 1 0x34 0x00 0x01 0x00 0x01 0x73 0x85 i
i2cset -y 1 0x34 0x00 0x02 0x00 0x80 0x00 0x00 i
sleep 1.0

#sin 4kHz
i2cset -y 1 0x34 0x00 0x00 0x00 0x00 0x00 0xFF i
i2cset -y 1 0x34 0x00 0x01 0x00 0x17 0x38 0x4F i
i2cset -y 1 0x34 0x00 0x02 0x00 0x80 0x00 0x00 i
sleep 1.0

#x=[x + 1]
done

i2cset -y 1 0x34 0x00 0x08 0x00 0x00 0x00 0x00 i

```

Na tym kończę kurs AudioDSP...

Dziękuję za uwagę i odzew, jednocześnie gorąco zachęcam do eksperymentów z AudioDSP we własnym zakresie, aktualne materiały pomocnicze oczywiście zostaną zamieszczone na ftp – ***media.avt.pl***.

Adam Tatuś, EP

100% ELEKTRONIKI NA AVT.PL/PRENUMERATA

[illegible]

Stożka Kształtów - Elektroniczny układ doświetlonej przystawki

Elektronika

dla wszystkich

10 DROGI PAŹDZIERNIK • CENA 2,00 zł brutto • MAGAZYNOWY

www.elektro.pl

Elektroniczna gra Łamigłówka

Izolowana galwanicznie sonda oscyloskopowa 2,5kV

- ▶ Testy
- ▶ Wyświetlacz z czułością tensu napięcia
- ▶ Minimalna długość elementów
- ▶ Wzrostki zasilające zasilaczem
- ▶ Wzrostki zasilające sterownikiem
- ▶ Łączność z sondą oscyloskopową
- ▶ Zasilacz
- ▶ Automatyzacja
- ▶ Elektroniczne wzmocnienie i rekompensacja
- ▶ Transmisja danych w kodzie binarnym
- ▶ Wzrostki zasilające
- ▶ Przetwarzanie danych
- ▶ Mapa danych - Wyświetlacz danych
- ▶ Rozbudowa sterownika
- ▶ Elektroniczne sterowanie

Przebieg testowania
Automatyzacja D20.pl
Elektronika D20.pl

Miejsca dla
specjalistów

Drukarki 3D
Drukarki 3D

Przebieg testowania
Automatyzacja D20.pl
Elektronika D20.pl

10/2019 październik
WYDAWCA: Wydawnictwo PR

Wydawca: Zarządca Miernikarski,
przez zarządcę Zarządca Chwał. Uniwersytet, str. 26

WYDAWCA: Wydawnictwo PR
WYDAWCA: Wydawnictwo PR

Elektronik

MAGAZYN ELEKTRONIKI PROFESJONALNEJ

Narzędzia i usługi do prototypowania elektronicznego

Pomiar natężenia prądu za pomocą czujników Halla

Nastawianie prądu elektrycznego jest jedną z kluczowych wyzwań dla inżynierów, którzy nie mają czasu na wybudowanie i pomiar natężenia prądu w wielu rodzajach systemów, np. w branży przemyślnych czy motoryzacyjnej. Istnieje wiele metod pomiaru prądu, np. w oparciu o: woltometry, czujniki Halla, termopary, termistory, itp. W tym artykule omówimy sposoby na pomiar natężenia prądu za pomocą czujników Halla, które są to czujniki, które mierzą natężenie prądu w przewodniku, który jest podłączony do czujnika.

Praca nr 68

W numerze

- Pobudki prototypowania elektronicznego za pomocą „Jaguar”... str. 10
- Barierne dla prototypu dla... str. 11
- Wzrost i możliwości analizy... str. 12
- Czynności utrzymujące... str. 14

ISSN 1429-3600

Narzędzia do statycznej analizy kodu - str. 58

Istnieje wiele metod pomiaru prądu, np. w oparciu o: woltometry, czujniki Halla, termopary, termistory, itp. W tym artykule omówimy sposoby na pomiar natężenia prądu za pomocą czujników Halla, które są to czujniki, które mierzą natężenie prądu w przewodniku, który jest podłączony do czujnika.

Praca nr 68

IoT - od aplikacji do realizacji

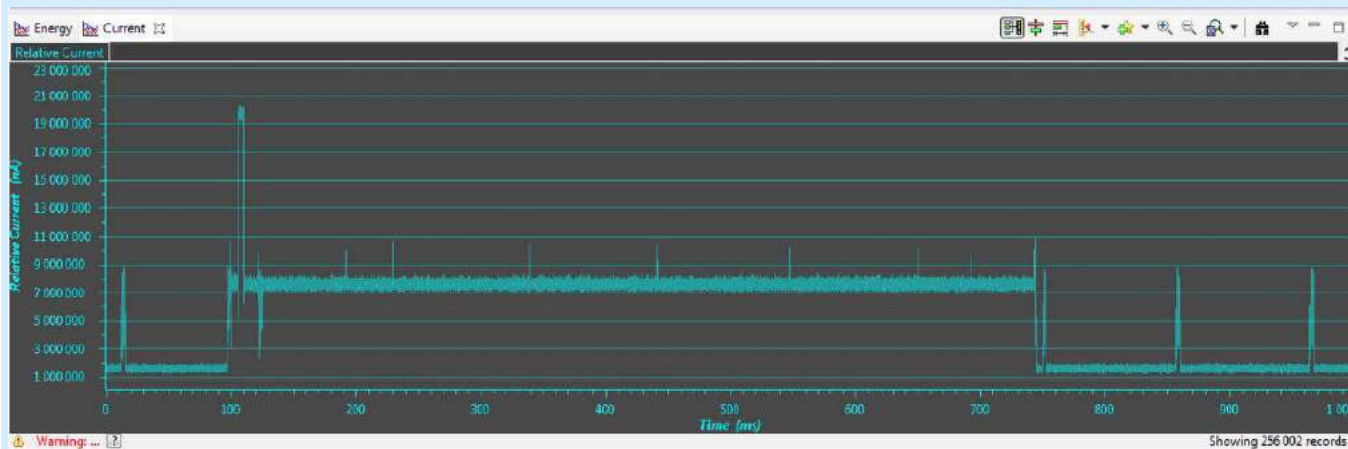
Internet Rzeczy (IoT) ma szansę wywrzeć ogromny wpływ na świat rzeczy podłączonych do Internetu. Oczekiwane definicje nie odzwierciedlają jego ogromnego potencjału, który będzie stał się węzłem sieci i może być dowolnie duży, wyrażony w milionach komunikacji. Słusznie jest, że w 2020 roku liczba rzeczy, które są połączone w przeliczeniu 20-30 mld sztuk, a nawet może być przekroczona.

Praca nr 68

Narzędzia do statycznej analizy kodu - str. 58

DIGIKEY.PL

**Prenumerujesz Elektronikę Praktyczną + Elektronikę dla Wszystkich?
Skorzystaj z promocji 1+1=3 i zamów bezpłatną prenumeratę Elektronika**



Systemy dla Internetu Rzeczy (30)

Praca z jednoczesną komunikacją radiową BLE 5 oraz IEEE 802.15.4



Szybki rozwój układów scalonych SoC (System on Chip) z procesorami komunikacyjnymi pozwala na realizację jednoczesnej transmisji wykonywanej w dwóch różnych pasmach radiowych. Nowa wersja produkcyjna (Rev. E) układu SimpleLink CC1352R1 firmy Texas Instruments umożliwia jednoczesną pracę w pasmie 2,4 GHz z obsługą protokołu Bluetooth Low Energy v5 oraz na obsługę komunikacji w pasmie poniżej 1 GHz (Sub-1GHz) w standardzie IEEE 802.15.4.

Podziękowania dla pana Marcina Paszkiewicza, szefa oddziału polskiego firmy Texas Instruments za udostępnienie wersji produkcyjnych modułów sprzętowych CC1352R1 LaunchPad.

W artykule zostaną omówione praktyczne eksperymenty z nową wersją produkcyjną (Rev. E, PG 2.1) procesora SimpleLink CC1352R1 firmy Texas Instruments [1]. Obsługuje on pracę z wieloma protokołami oraz w dwóch pasmach: 2,4 GHz oraz w pasmie poniżej 1 GHz (Sub-1GHz). Technologia *Dynamic Multi-protocol Manager (DMM)* udostępniana jako część pakietu programowego CC13x2 26x2 SDK [4] umożliwia jednoczesną transmisję w dwóch sieciach. W pasmie 2,4 GHz obsługiwany jest protokół Bluetooth 5 a w pasmie poniżej 1 GHz protokół IEEE 802.15.4 oraz autorski protokół EasyLink. Omówienie układów dwupasmowych zostało zamieszczone w poprzednim artykule tej serii [S17]. Omówienie układów scalonych z obsługą Bluetooth 5 zostało zamieszczone w poprzednim artykule [2].

Dokumentacja

Odszukanie dokumentacji dotyczącej użycia technologii DMM z układami scalonymi rodziny CC13x2 nie jest łatwe. Dla rodziny CC13x2 opis jest zamieszczony w portalu TIREX [7]. Dla procesora CC1352R1F3 opis DMM znajduje się w pakiecie *SimpleLink CC13x2 and CC26x2 software development kit (SDK)* [4]. Trzeba nawigować do ścieżki *Documents → Documentation Overview*. Następnie w sekcji *TI DMM Documentation* znajdują się odnośniki do czterech

dokumentów: *TI DMM Release Notes*, *TI DMM User's Guide* [8], *TI DMM Examples User's Guide* oraz *TI DMM Runtime APIs*.

Zestaw startowy SimpleLink CC1352R1 LaunchPad

Do eksperymentów z technologią DMM potrzebna jest nowa wersja zestawu CC1352R1 LaunchPad Rev.B firmy Texas Instruments [3]. Zestaw startowy jest wyposażony w nową wersję produkcyjną (Rev. E, PG2.1) procesora SimpleLink CC1352R1F3 [1]. Zestaw jest przeznaczony do pracy w pasmach 868 MHz/915 MHz oraz 2,4 GHz. Cała elektronika zestawu CC1352R1 LaunchPad jest umieszczona na jednej wielowarstwowej płytce drukowanej. Płytkę jest podzielona na dwie części: u góry płytki jest debugger sprzętowy standardu XDS110 oraz układ monitora zasilania a na dole układ scalony CC1352R1F3. Opis budowy poprzedniej wersji zestawu CC1352R1 LaunchPad został zamieszczony w poprzednim artykule tej serii EP 8/2018 [S18]. W nowej wersji Rev. B zestawu został tylko zmieniony układ monitora zasilania oraz wersja procesora.

Oprogramowanie zestawów startowych LaunchPad

Praca z DMM wymaga użycia środowiska *Code Composer Studio (CCS)* w wersji co najmniej 9.1 oraz procesora CC1352R1F3 w wersji Rev. E (PG2.1) [4]. Opis oprogramowania narzędziowego dla układów CC26xx i CC13xx platformy SimpleLink został zamieszczony w EP 11/2017 [S12]. Opis oprogramowania dedykowanego dla

procesora CC1352R1F3 został zamieszczony w EP5/2018 [S15].

W pakiecie CC13x2 26x2 SDK v3.20 [4] udostępnionych jest kilka przykładowych projektów w ścieżce *Examples* → *Development Tools* → *CC1352R LaunchPad* → *dmm*. Najciekawszy jest projekt DMM 15.4 Sensor + BLE Remote Display (*dmm_w154sensor_remote_display_app*). Umożliwia zastosowanie technologii DMM do jednoczesnej pracy z protokołem BLE5 oraz protokołem IEEE 802.15.4g. Projekt zawiera plik README w którym jest zamieszczony opis aplikacji projektu [9]. Projekt ten został zastosowany do eksperymentów praktycznych opisanych dalej.

Bardzo przydatne są trzy warsztaty dostępne w ramach „SimpleLink Academy 3.20.01 for SimpleLink CC13x2/CC26x2 SDK 3.20”:

- DMM Fundamentals – Umożliwia pracę dwuzakresową z protokołem BLE oraz autorskim protokołem EasyLink [13].
- Customizing Use Case – Umożliwia pracę dwuzakresową z protokołem BLE oraz protokołem IEEE 802.15.4g [14].
- DMM Integration – Umożliwia budowanie aplikacji dwuzakresowej od podstaw [15].

Ponadto przydatny jest warsztat „Sensor and Collector – TI 15.4-Stack Project Zero” [16] w którym jest opisana praca z aplikacją *Collector* z protokołem IEEE 802.15.4g [10].

Do eksperymentów praktycznych zostało zastosowane środowisko programowe w wersji CCS v9.1. W przypadku gdy eksperymenty z pracą dwupasmową odbywają w większej grupie uczestników to trzeba zapewnić pracę sieci IEEE 802.15.4g osobno dla każdego uczestnika. Potrzebne jest wtedy przydzielenie każdemu osobnego kanału i ewentualnie osobne PAN ID. Sposób postępowania jest dokładnie omówiony w ćwiczeniu „Sensor and Collector – TI 15.4-Stack Project Zero” [16]. Polega to na takich samych modyfikacjach w pliku *config.h* w projekcie Sensor Remote Display oraz projekcie Collector. Kolejny problem może dotyczyć rozpoznania w aplikacji mobilnej „własnego” zestawu Sensor Remote Display wysyłającego pakiety rozgłaszania. Najprostszym sposobem jest wybieranie nadajnika o największym poziomie sygnału RSSI. Pewniejszym sposobem jest łączenie się z urządzeniem według adresu MAC.

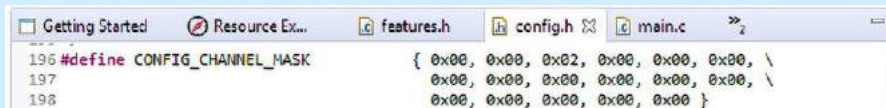
Aplikacje mobilne

Do wykonania połączenia standardu BLE pomiędzy zestawem startowym CC1352R1 LaunchPad pracującym z aplikacją *Sensor Remote Display* oraz urządzeniem mobilnym, musi być używana aplikacja *LightBlue Explorer* [5, 6] lub BLE Scanner. Można też zastosować aplikację BTool dostarczaną w ramach pakietów programowych SDK dla układów rodziny CC12x2R/CC26x2R platformy SimpleLink [S19]. Wymagany jest wtedy dodatkowy zestaw startowy LaunchPad z obsługą BLE.

Zadanie 1. Przygotowanie zestawów uruchomieniowych LaunchPad do pracy

Do eksperymentów praktycznych potrzebny jest podstawowy zestaw startowy CC1352R1 LaunchPad Rev. B oraz dodatkowy zestaw startowy LaunchPad z obsługą transmisji z protokołem IEEE 802.15.4g w paśmie poniżej 1 GHz. Może to być zestaw z procesorem CC1310, CC1350, CC1312 lub CC1352. Przed rozpoczęciem pracy z każdym zestawem startowym LaunchPad należy koniecznie wykonać kasowanie całej pamięci Flash, wykonując następujące kroki:

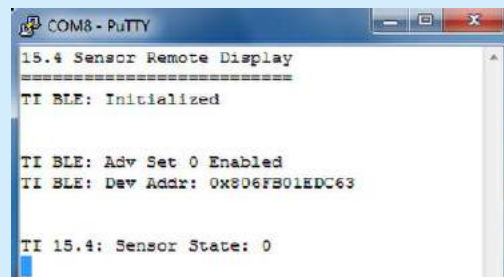
1. Dołącz zestaw startowy CC1352R1 LaunchPad do komputera PC używając kabla USB.
2. Wystartuj aplikację *UniFlash 5.1*. Zostanie automatycznie wyszukany moduł LaunchPad dołączony do komputera PC.
3. Kliknij *OK* lub *Start*.
4. Wybierz zakładkę *Settings & Utilities* i przewiń prawy panel do pozycji *Manual Erase*.



Rysunek 1. Modyfikacja wyboru kanału w aplikacji Collector



Rysunek 2. Modyfikacja odstępu wystania danych



Rysunek 3. Informacje aplikacji Sensor Remote Display po rozpoczęciu pracy

5. Kliknij na przycisk *Erase Entire Flash*.
6. Jeśli zostanie wyświetlony komunikat o konieczności aktualizowania oprogramowania firmowego (dla XDS110) to wykonaj aktualizację.
7. Zamknij aplikację *UniFlash 5.1* i odłącz zestaw od komputera.

Zadanie 2. Zaprogramowanie aplikacji Collector

Opis dotyczy postępowania przy programowaniu zestawu startowego CC1352R LaunchPad Rev B. Takie samo postępowanie jest konieczne w przypadku użycia innych zestawów startowych. Należy postępować według poniższego skróconego opisu.

1. Dołącz zestaw startowy do komputera używając kabla USB.
2. Wystartuj środowisko CCSv9.1.
3. Kliknij na przycisk *Browse Examples*.
4. W oknie *Resource Explorer* rozwiń ścieżkę: *SimpleLink CC13x2 26x2 SDK – v.3.20.00.68* → *Examples* → *Development Tools* → *CC1352R LaunchPad* → *TI 15.4 Stack* → *collector* → *TI-RTOS* → *CCS Compiler* → *collector*
5. W prawym oknie kliknij na przycisk *Import to IDE*.
6. W oknie *Project Explorer* rozwiń drzewo projektu *collector_cc1352R1lp* oraz gałąź *subg*.
7. Otwórz plik *config.h*.
8. W linii 88 zamień *APIMAC_STD_US_915_PHY_1* na *APIMAC_STD_ETS_863_PHY_3*.
9. Zmień zawartość linii 196...198. Jest to maska bitowa przydziału kanałów transmisji protokołu IEEE 802.15.4. Dla pasma 868 MHz (Europa) dostępne są kanały 0 do 33. Opis sposobu przypisania numeru kanału do bitu pola jest opisany powyżej linii 196.
10. Dla kanału 17 trzeba ustawić zawartość według **rysunku 1**.
11. W linii 117 należy zmienić wartość 90000 na 30000. Ustawi to odstęp wysłania danych w sieci IEEE 802.15.4 na 30 sekund (**rysunek 2**).
12. Bardzo przydatne jest też dodanie do projektu Collector wyświetlania odebranych danych (temperatura w °C).
13. Otwórz plik *csf.c* i w linii 625 wstaw następujący kod:
`LCD_WRITE_STRING_VALUE(„Temperature=”, pMsg->tempSensor.objectTemp, 10, 6);`
14. Z menu wybierz *Project* → *Clean* i kliknij *Clean*.
15. Kliknij na przycisk *Debug*.
16. Jeśli zostanie wyświetlony komunikat o konieczności aktualizowania oprogramowania firmowego (dla XDS110) to wykonaj aktualizację.

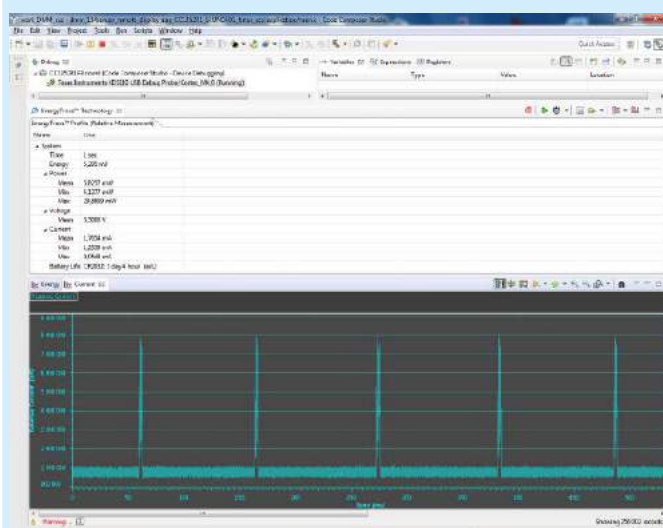
17. W oknie *Menadżer Urządzeń* sprawdź i zanotuj numer portu „COMx1” dla kanału *Application/User UART*.
18. Zamknij CCS.
19. Oznacz pierwszy zestaw startowy jako „Collector”.

Zadanie 3. Zaprogramowanie aplikacji *Sensor Remote Display*

1. Dołącz drugi zestaw startowy CC1352R1 LaunchPad Rev B do komputera, używając kabla USB.
2. Uruchom środowisko CCSv9.1. Zastosuj inną nazwę foldera roboczego niż poprzednio.
3. Kliknij na przycisk *Browse Examples*.
4. W oknie *Resource Explorer* rozwiń ścieżkę: SimpleLink CC13x2 26x2 SDK – v.3.20.00.68 → Examples → Development Tools → CC1352R LaunchPad → dmm → dmm_154sensor_remote_display_app → TI-RTOS → CCS Compiler → dmm_154sensor_remote_display_app
5. W prawym oknie kliknij na przycisk *Import to IDE*.
6. W oknie *Project Explorer* rozwiń drzewo projektu dmm_154sensor_remote_display_app oraz gałąź *sensor* → *subg*.
7. Otwórz plik *config.h*.
8. W linii 93 zamień *APIMAC_STD_US_915_PHY_1* na *APIMAC_STD_ETS1_863_PHY_3*.
9. Ustaw ten sam numer kanału jak dla aplikacji *Collector*. Zmień zawartość linii 155...157 na taką samą jak poprzednio (zadanie 2, punkt 10).
10. Z menu wybierz *Project* → *Clean* i kliknij *Clean*.
11. Kliknij na przycisk *Debug*.
12. Jeśli zostanie wyświetlony komunikat o konieczności aktualizowania oprogramowania firmowego (dla XDS110) to wykonaj aktualizację.
13. W oknie *Menadżer Urządzeń* sprawdź i zanotuj numer portu „COMx2” dla kanału *Application/User UART*.
14. Wystartuj aplikację terminala *PuTTY*.
15. Kliknij przycisk *Serial*, w polu *Serial line* wpisz numer portu szeregowego „COMx2”, w polu *Speed* (szybkość transmisji) wpisz wartość 115200.
16. W oknie *Category* kliknij na *Colours*, zaznacz *Use system colours* i kliknij *OK*.
17. W perspektywie *CCS Debug* aplikacji CCS kliknij na przycisk *Resume*.
18. W oknie *PuTTY* zostanie wyświetlona informacja (rysunek 3), do której wrócimy później.

Zadanie 4. Praca z *EnergyTrace* w trybie samodzielnym

Technologia *EnergyTrace* jest opracowana przez firmę Texas Instruments i udostępniana jako część środowiska programistycznego *Code Composer Studio* [11]. Technologia *EnergyTrace* jest analizatorem mocy zasilania,



Rysunek 4. Okna *EnergyTrace* z pomiarem prądu pracy Sensora w trybie rozgłaszania BLE

który pozwala na pomiar prądu zasilania układów scalonych rodziny SimpleLink CC13x2/CC26x2. Dane udostępniane są w postaci wykresów oraz zestawienia tabelarycznego. Opis pracy z technologią *EnergyTrace* został dokładnie opisany w poprzednim artykule tej serii EP 6/2018 [S16]. Opis pracy z układem CC1352R1F3 został zamieszczony w dokumencie „Measuring CC13xx and CC26xx current consumption” [12].

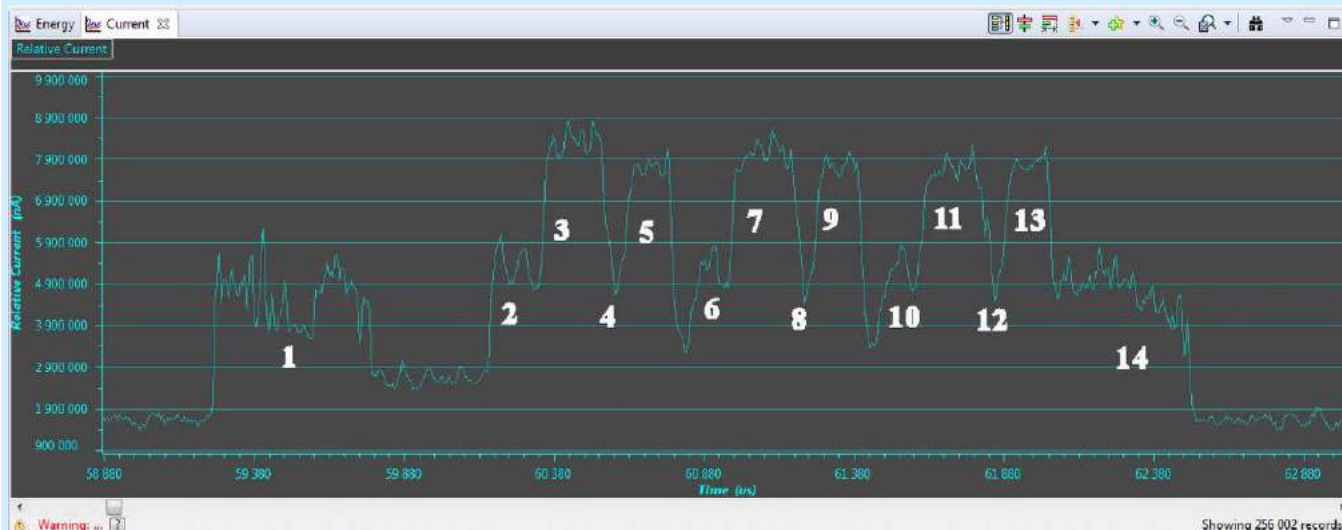
1. W perspektywie *CCS Debug* kliknij na pasku menu na przycisk *EnergyTrace Technology*.
2. Kliknij na przycisk *Proceed*.
3. Zostanie otworzone okno *EnergyTrace* z zakładkami wykresów *Power* i *Energy*. Podczas pierwszego używania technologii *EnergyTrace* w środowisku CCS należy poutawiać różne parametry.
4. Kliknij na przycisk *Advanced Menu*.
5. W podręcznym menu wybierz *Preferences*, zostanie otworzone okno z opisem technologii *EnergyTrace*.
6. W polu *Target Connection* ustaw *Connection* na XDS110 oraz *Voltage* na 3300,0 mV.
7. Wybierz wyższy zakres pomiaru prądu – zaznacz *High current...*
8. Kliknij *OK*, ustawienia zostaną zapisane.
9. Kliknij na przycisk *Select Measurement Duration*. Wybierz długość czasu pobierania danych (1 sec) z układu scalonego.
10. Kliknij na przycisk *Start trace collection*.
11. Rozpocznie się zbieranie danych. Przycisk zmieni się na *Stops trace collection* umożliwiając zatrzymanie akwizycji.
12. Na pasku stanu (pod paskiem menu) wyświetlane jest odliczanie czasu: „Timer ... 1” (aż do zera).
13. Po zakończeniu zbierania danych w oknie zakładki *EnergyTrace Profile* są pokazywane obliczone dane. Wykres jest pokazywany na bieżąco w zakładkach *Energy* oraz *Current*.
14. Jeśli nie jest widoczna zakładka *Current* to wykonaj następujące polecenia:
15. Kliknij na przycisk *Advanced Menu*.
16. W podręcznym menu wybierz *Show graphs*. Pojawia się teraz nowa zakładka *Current*.
17. Wybierz zakładkę *Current*.

Okna zakładek można przeciągać do innych miejsc okna perspektywy w celu ich jednoczesnego oglądania (rysunek 4). Wykres prądu jest skalowany w nA. W oknie *EnergyTrace Technology* jest pokazywana zmierzona wartość minimalna i maksymalna prądu oraz wynik pomiaru napięcia zasilania.

Tryby pracy aplikacji *Sensor*

Aplikacja *Sensor* pracuje w czterech stanach (tabela 1). Gdy stos BLE pracuje w stanie rozgłaszania to ma on niższy priorytet niż stos TI 15.4. Ale gdy stos BLE pracuje w stanie połączenia to ma priorytet wyższy. Taka organizacja pracy wynika z synchronicznej natury transmisji protokołu BLE. Stos TI 15.4 pracuje z transmisją asynchroniczną, więc

Tabela 1. Tryby pracy aplikacji <i>Sensor</i>				
Stan	Stos BLE	Stos TI 15.4	Priorytet	Opis
1	Advertising	-	BLE	Tryb rozgłaszania BLE. Brak aktywności w sieci IEEE 802.15.4.
2	Connected	-	BLE	Tryb połączenia BLE. Brak aktywności w sieci IEEE 802.15.4.
3	Advertising	Connected	TI 15.4	Tryb rozgłaszania BLE z transmisją IEEE 802.15.4.
4	Connected	Connected	BLE	Tryb połączenia BLE z transmisją IEEE 802.15.4.



Rysunek 5. Pobór prądu w trakcie wysłania pakietu rozgłaszania BLE

Tabela 2. Stany podczas wysyłania pakietu rozgłaszania

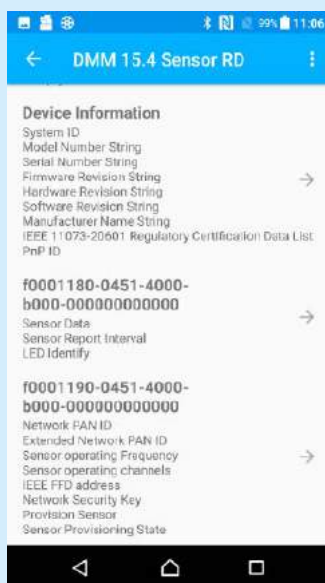
Nr	Stan	Opis
1	Przetwarzanie wstępne	Wybudzenie systemu RTOS, czas ustabilizowania pracy oscylatora XTAL
2	Przygotowanie modułu radiowego	Uruchomienie modułu radiowego i włączenie nadajnika
3	Praca nadajnika	Moduł radiowy transmituje pakiet rozgłaszania z trzema bajtami danych w kanale CH37
4	Przełączenie z pracy nadajnika na pracę odbiornika	Przełączenie z nadawania na odbiór
5	Praca odbiornika	Odbiór
6	Przełączenie z pracy odbiornika na pracę nadajnika	Przełączenie z odbioru na nadawanie
7	Praca nadajnika	Nadawanie
8	Przełączenie z pracy nadajnika na pracę odbiornika	Przełączenie z nadawania na odbiór
9	Praca odbiornika	Odbiór
10	Przełączenie z pracy odbiornika na pracę nadajnika	Przełączenie z odbioru na nadawanie
11	Praca nadajnika	Nadawanie
12	Przełączenie z pracy nadajnika na pracę odbiornika	Przełączenie z odbioru na nadawanie
13	Praca odbiornika	Odbiór
14	Przetwarzanie końcowe i przejście do stanu Standby	Stos BLE przetwarza odebrane pakiety danych, ustawia timer uśpienia w przygotowaniu do następnego eventu i następnie przechodzi do stanu Standby

możliwe jest powtórzenie transmisji pakietu bez utraty danych lub połączenia sieci.

Zadanie 5. Praca Sensora w trybie rozgłaszania BLE

Po uruchomieniu aplikacji *Sensor Remote Display*, pracuje aktywny stos sieci Bluetooth 5 (BLE). Można w łatwy sposób powrócić do tego stanu. W tym celu należy przycisnąć przycisk Reset na górze płytki zestawu startowego CC1352R1 LaunchPad. Na **rysunku 3** pokazane są informacje wyświetlone w oknie terminala po rozpoczęciu pracy aplikacji Sensor Remote Display:

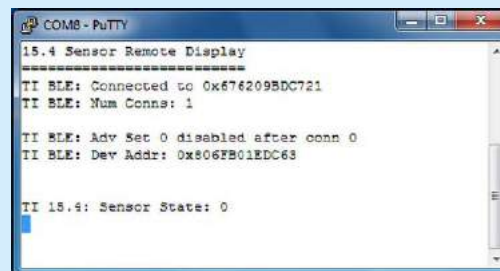
- W pierwszej linii TI BLE pokazywany jest stan pracy stosu BLE. W stanie *Initialized* oraz *Disconnected* realizuje on rozgłaszanie.



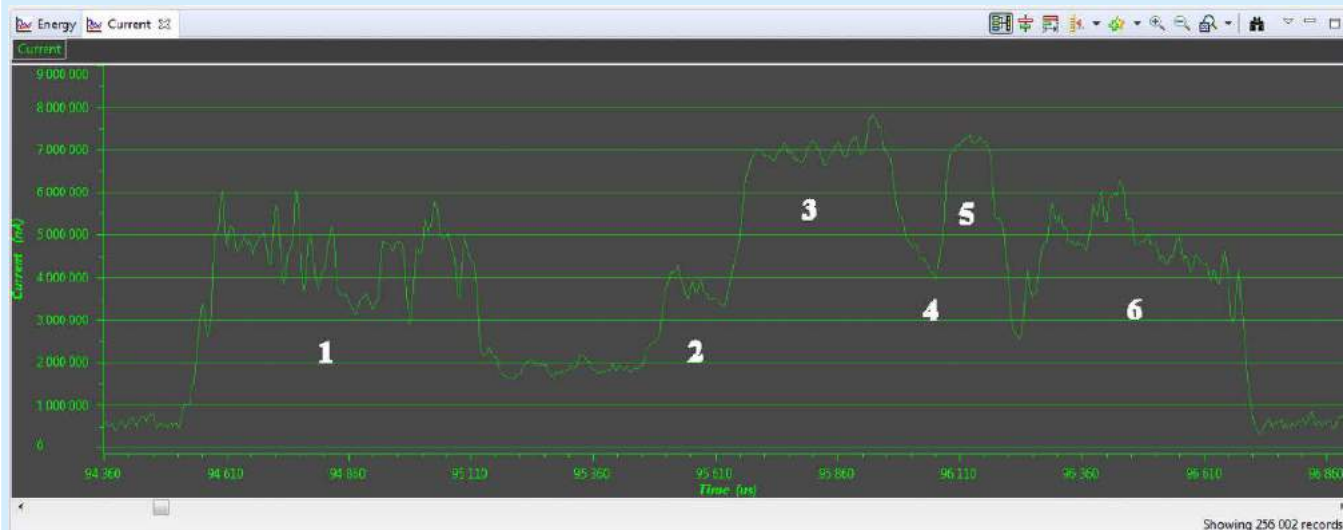
Rysunek 6. Obsługiwane charakterystyki sieci DMM w aplikacji LightBlue Explorer

- W linii *Adv Set* jest pokazany stan włączenia rozgłaszania BLE.
- W linii *Dev Addr* jest pokazywany MAC adres zestawu startowego CC1352R1 LaunchPad, na którym pracuje aplikacja Sensor Remote Display.
- W linii *TI 15.4: Sensor State* jest pokazywany stan pracy stosu TI 15.4.

Wykonany w zadaniu 4 pomiar poboru prądu jest pokazany na **rysunku 4**. Pakiety rozgłaszania BLE są wysyłane w ostępach ok. 100 ms. Maksymalna wartość prądu wynosi 9,0548 mA. Jednak duży pobór prądu trwa bardzo krótko. Dlatego średni prąd zasilania



Rysunek 7. Informacje aplikacji Sensor Remote Display w trybie komunikacji BLE



Rysunek 8. Pobór prądu w trakcie wysyłania pakietu komunikacji BLE

Tabela 3. Stany podczas wyłania pakietu komunikacji

Nr	Stan	Opis
1	Przetwarzanie wstępne	Wybudzenie systemu RTOS, czas ustabilizowania pracy oscylatora XTAL
2	Przygotowanie modułu radiowego	Uruchomienie modułu radiowego i włączenie nadajnika
3	Praca odbiornika	Odbiór
4	Przełączenie z pracy odbiornika na pracę nadajnika	Przełączenie z odbioru na nadawanie
5	Praca nadajnika	Nadawanie
6	Przetwarzanie końcowe i przejście do stanu Standby	Stos BLE przetwarza odebrane pakiety danych, ustawia timer uśpienia w przygotowaniu do następnego eventu i następnie przechodzi do stanu Standby

wynoszący 1,7654 mA jest tylko trochę wyższy niż pobór minimalny 1,2539 mA.

Dokładny przebieg czasowy poboru prądu w trakcie wysyłania pakietu rozgłaszania pokazuje **rysunek 5**. Został zaprezentowany typowy pakiet rozgłaszania z nadawaniem i odbieraniem danych we wszystkich trzech kanałach rozgłaszania BLE.

Dane pakietu rozgłaszania zawierają ton pilotujący, 1 bajt preambuły, 4 bajty adresu, 3 bajty danych (PDU), 3 bajty CRC oraz 6 bajtów AdvA. Podstawowy czas transmisji takiego pakietu wynosi 144 μ s. Każdy dodatkowy bajt dodaje 1 μ s do czasu pracy nadajnika. Na rysunku zostały ponumerowane kolejne stany występujące w trakcie wysyłania pakietu rozgłaszania. Są one szczegółowo przedstawione w **tabeli 2**.

Zadanie 6. Praca Sensora w trybie komunikacji BLE

Do wykonania połączenia standardu BLE została zastosowana aplikacja LightBlue Explorer pracująca pod systemem Android [5, 6].

1. Uruchom aplikację LightBlue Explorer na urządzeniu mobilnym.
2. Wykonaj połączenie. Kliknij na linię DMM 15.4 Sensor RD. Poniżej pokazywany jest adres węzła, co ułatwia jego wybór w bardziej rozbudowanej sieci.

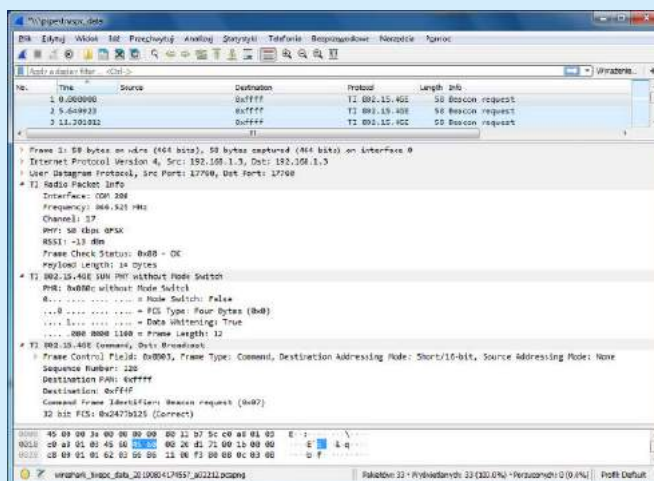
Teraz na ekranie pokazywane są różne dane.

Po przewinięciu ekranu w dół pokazywane są informacje oraz dwa serwisy (**rysunek 6**). Dla każdego serwisu jest pokazana lista obsługiwanych charakterystyk. Informacje są dostępne od razu. Aktualne dane pierwszego serwisu są dostępne dopiero po nawiązaniu przez węzeł Sensor komunikacji w sieci TI 15.4. Dostęp do danych drugiego serwisu wymaga hasła dostępu ponieważ jest stosowane połączenie z szyfrowaniem. Informacje aplikacji Sensor po połączeniu ze smartfonem poprzez łącze BLE są pokazane na **rysunku 7**. W pierwszej linii TI BLE pokazywany jest adres BLE dołączonego urządzenia mobilnego.

Dokładny przebieg czasowy poboru prądu w trakcie wysyłania pakietu komunikacji BLE jest pokazany na **rysunku 8**. Na rysunku

zostały ponumerowane kolejne stany występujące w trakcie wysyłania pakietu. Są one szczegółowo przedstawione w **tabeli 3**.

Zadanie 7. Praca Sensora w trybie rozgłaszania BLE z komunikacją IEEE 802.15.4



Rysunek 9. Podgląd pakietów rozgłaszania sieci TI 15.4



Rysunek 10. Informacje aplikacji Collector z komunikacją IEEE 802.15.4

Stos TI 15.4 w czasie pracy przechodzi pomiędzy różnymi stanami:

0. Oczekiwanie na inicjalizację. Uruchomiony ale nie startuje i czeka na wystartowanie przez użytkownika.
1. Dołączanie do sieci. Startuje skanowanie i wybór sieci do dołączenia.
2. Odtwarzanie połączenia sieciowego. Znajdowanie informacji o sieci, ponowne łączenie się do niej.
3. Dołączony do sieci. Normalna praca w sieci.
4. Ponownie dołączony do sieci. Normalna praca w sieci po powrocie.
5. Po utracie połączenia z siecią. Węzeł jest „osierocony” i nie krywa sieci.
6. Węzeł nie został dołączony do sieci. Odmowa dostępu.

Po uruchomieniu aplikacji *Sensor Remote Display* stos TI 15.4 jest uruchomiony (stan 0) ale nie powoduje to żadnej aktywności modułu radiowego. W celu rozpoczęcia pracy stosu TI 15.4 należy przycisnąć przycisk BTN-2 (prawy) na płycie zestawu startowego CC1352R1 LaunchPad oznaczonej jako „Sensor”. Spowoduje to przejście do skanowania i szukania sieci (stan 1). Jeśli sieć zostanie znaleziona i zgadzają się parametry to zostanie zestawione połączenie (stan 3). Czynności do wykonania są następujące:

1. Na urządzeniu mobilnym wykonaj rozłączenie komunikacji BLE.
2. Dołącz zestaw startowy oznaczony jako „Collector” do komputera używając kabla USB.
3. Wystartuj aplikację terminala PuTTY.
4. Kliknij przycisk *Serial*. W polu *Serial line* wpisz numer portu szeregowego „COMx1”. W polu *Speed* (szybkość transmisji) wpisz wartość 115200.
5. W oknie *Category* kliknij na *Colours*. Zaznacz *Use system colours*. Kliknij OK.
6. Przyciśnij przycisk Reset (na górze) zestawu startowego CC1352R1 LaunchPad oznaczonego jako Collector.
7. W oknie PuTTY aplikacji Collector zostanie pokazana informacja jak na **rysunku 11** (pierwsze dwie linie).
8. W oknie PuTTY aplikacji Sensor Remote Display zostanie pokazana informacja: „TI BLE: Device Disconnected”.
9. Przyciśnij przycisk BTN-2 (prawy) na płycie zestawu startowego CC1352R1 LaunchPad oznaczonej jako „Sensor”.



Rysunek 11. Informacje aplikacji Sensor Remote Display w trybie komunikacji BLE z komunikacją IEEE 802.15.4

W oknie PuTTY aplikacji Sensor Remote Display zostanie pokazana informacja o przejściu stosu TI 15.4 ze stanu 0 do stanu 1. Zostanie wysłana seria pakietów rozgłaszania (rysunek 9). Przechwytywanie pakietów komunikacji z zastosowaniem aplikacji Wireshark daje bardzo dobry wgląd w stan i parametry komunikacji. Sposób używania aplikacji Wireshark do monitorowania transmisji z komunikacją IEEE 802.15.4 został dokładnie opisany w poprzednim artykule EP 2/2018 [S14].

Na rysunku tytułowym pokazany jest zapis poboru prądu zestawu startowego CC1352R1 LaunchPad z aplikacją Sensor pracującą w trybie rozgłaszania BLE z komunikacją IEEE 802.15.4. Na początku zapisu jest widoczny wykres poboru prądu w trakcie wysłania pakietu rozgłaszania BLE, zgodny z rysunkiem 4. Następnie jest aktywność związana z obsługą transmisji z protokołem IEEE 802.15.4 w paśmie 868 MHz. Wysyłanie pakietów rozgłaszania BLE zostało zablokowane. Na koniec kontynuowane jest wysyłanie pakietów rozgłaszania BLE z typowym odstępem czasu 100 ms.

10. Przyciśnij przycisk BTN-2 (prawy) na płycie zestawu startowego CC1352R1 LaunchPad oznaczonej jako „Collector”.

Rozpocznie mrugać czerwona dioda co sygnalizuje wystartowanie i otworzenie sieci do dołączania węzłów. W oknie PuTTY aplikacji Collector zostanie pokazana informacja PermitJin-ON a następnie Joined (rysunek 11). Następuje automatyczne dołączenie węzła z aplikacją DMM do sieci IEEE 802.15.4 tworzonej przez aplikację Collector w paśmie Sub-1 GHz.

W oknie PuTTY aplikacji Sensor Remote Display zostanie pokazana informacja o przejściu stosu TI 15.4 ze stanu 1 do stanu 3 – Normalna praca w sieci. Jest to jedyna zmiana w stosunku do sytuacji z rysunku 7.

Wybrane pozostałe artykuły kursu Systemy dla Internetu Rzeczy

- [S12] Oprogramowanie narzędziowe dla układów CC26xx i CC13xx platformy SimpleLink, EP12/2017
- [S14] Podglądanie ruchu w sieci radiowej z protokołem IEEE 802.15.4, EP 2/2018
- [S15] Zestaw CC1352R1 LaunchPad, EP 5/2018
- [S16] Dynamiczne monitorowanie prądu zasilania układu SoC, EP 6/2018

Literatura

- [1] CC1352R SimpleLink multi-band wireless MCU, Product Page, Texas Instruments <http://bit.ly/2vJkmqm>
- [2] Układy scalone z obsługą Bluetooth 5, Henryk A. Kowalski, „Elektronika Praktyczna”, 5/2018
- [3] SimpleLink Multi-Band CC1352R Wireless MCU LaunchPad Development Kit, LAUNCHXL-CC1352R1, Texas Instruments <http://bit.ly/2l0oP99>
- [4] SimpleLink CC13x2 and CC26x2 software development kit (SDK), 16 Jul 2019, Texas Instruments <http://bit.ly/2kUKgmr>
- [5] LightBlue Explorer, Punch Through Design <https://apple.co/2kvlMyU>
- [6] LightBlue Explorer, Version 1.3.2, 24 lipca 2019, Punch Through Design <http://bit.ly/2w6w108>
- [7] TI Resource Explorer (TIREX), Texas Instruments <http://bit.ly/2HHDqqo>
- [8] SimpleLink™ CC13x2 / CC26x2 SDK Dynamic Multi-protocol User's Guide, 1.10.00, Texas Instruments <http://bit.ly/2mn9xWF>

- [S17] Jednoczesna komunikacja radiowa z użyciem dwóch protokołów i w dwóch pasmach, EP 8/2018
- [S18] Praca z jednoczesną komunikacją radiową z użyciem dwóch protokołów i w dwóch pasmach, EP 8/2018
- [S19] Bezpieczeństwo transmisji z protokołem Bluetooth Low Energy, EP 10/2018

- [9] DMM 15.4 Sensor + BLE Remote Display (README.html), Texas Instruments <http://bit.ly/2m6wkWx>
- [10] TI 15.4-Stack Collector Example (README.md), Texas Instruments <http://bit.ly/2lZxs9W>
- [11] EnergyTrace User Guide, Texas Instruments <http://bit.ly/2mocJRW>
- [12] Measuring CC13xx and CC26xx current consumption, SWRA478D, January 2019, Texas Instruments <http://bit.ly/2mockW0>
- [13] Dynamic Multi-protocol Manager Fundamentals, Texas Instruments <http://bit.ly/2mtpOcr>
- [14] Dynamic Multi-protocol Manager Custom Use Case, Texas Instruments <http://bit.ly/2mOV7vk>
- [15] Dynamic Multi-protocol Manager Integration From Scratch, Texas Instruments <http://bit.ly/2kMR4CL>
- [16] Sensor and Collector – TI 15.4-Stack Project Zero, Texas Instruments <http://bit.ly/2kteQn3>

Po pomyślnym wysłaniu parametrów konfiguracyjnych przez Collector do węzła Sensor w oknie PuTTY aplikacji Sensor Remote Display zostanie pokazana informacja *ConfigRsp* (rysunek 10). Dane z węzła Sensor są wysyłane, po ustawiony czasie odstępu, do węzła Collector. W oknie PuTTY aplikacji Collector pokazywana jest informacja o numerze węzła Sensor od którego przyszedł pakiet transmisji oraz wartość przysłanych danych (rysunek 11, ostatnie dwie linie).

Zadanie 8: Praca Sensora w trybie komunikacji BLE z komunikacją IEEE 802.15.4

Praca ze współbieżną transmisją posiada ograniczenia. Przy startowaniu sieci TI 15.4 występuje czasowe zatrzymanie pracy stosu BLE. Jeśli stos BLE był w stanie połączenia to zostanie ono zerwane i stos BLE powróci do stanu rozgłaszania. Występuje wtedy konieczność ponownego zestawienia połączenia.

- Wykonaj połączenie BLE aplikacji LightBlue Explorer aplikacją Sensor. Teraz jednocześnie współpracują ze sobą trzy urządzenia przy użyciu dwóch pasm i dwóch protokołów komunikacyjnych.
 - Przewiń ekran w dół.
 - Wejdź do drugiego serwisu UUID: 1190 (zobacz rysunek 6).
 - Wejdź do obsługi charakterystyki Network PAN ID (UUID: 1191).
 - Kliknij na przycisk *READ AGAIN*.
 - Po krótkiej chwili aplikacja *LightBlue Explorer* zasygnalizuje konieczność uwierzytelniania. W oknie aplikacji Sensor jest pokazywane wymagane hasło w linii TI BLE: Passcode (rysunek 11).
 - Wpisz odczytane hasło w aplikacji *LightBlue Explorer*. Aplikacja Sensor Remote Display zawiera menadżera wiązania (bonding), który zapisuje klucz w pamięci nieulotnej, w celu zastosowania po ponownym wznowieniu połączenia. Nie jest wtedy wymagane ponowne przeprowadzenie parowania z uwierzytelnianiem. W oknie aplikacji Sensor zostanie wyświetlona dodatkowa linia z informacją *Encryption success* (rysunek 12). Dokładny opis mechanizmów bezpieczeństwa transmisji BLE został zamieszczony w poprzednim odcinku kursu EP 10/2018 [S19].
 - Odczytana wartość Network PAN ID wynosi 0x0001.
- Jeśli sieć TI 15.4 nie jest w stanie połączenia to dane odczytywane w obu serwisach nie są poprawne, typowo są to same zera lub wartość F.
- Wejdź do pierwszego serwisu UUID: 1180 (zobacz rysunek 6).
 - Wejdź do obsługi charakterystyki Sensor Report Interval (UUID: 1182).
 - Kliknij na przycisk *READ AGAIN*. Odczytana wartość pokazuje odstęp powtarzania komunikacji w milisekundach (zapis hex).
 - Kliknij na pole *Data to write* na lewo od przycisku *WRITE*.
 - Wpisz nową wartość, dla 5 sek. jest to 00 00 13 88.



Rysunek 12. Informacje aplikacji Sensor po zmianie czasu odstępu transmisji

- Kliknij na przycisk *WRITE*.

W oknie aplikacji Sensor zostanie wyświetlona dodatkowa linia z nową wartością odstępu powtarzania komunikacji (rysunek 12).

Podsumowanie

Obecna wersja oprogramowania firmowego (CC13x2 26x2 SDK v3.20) udostępnia pracę z pełnym stosem IEEE 15.4 oraz BLE 5. W pokazanym powyżej przykładzie zostały zmienione parametry pracy sieci IEEE 802.15.4 pracującej w paśmie 868 MHz poprzez łącze z komunikacją standardu BLE z urządzenia mobilnego. W taki sam sposób można podejrzeć aktualne dane w tej sieci TI 15.4 przekazywane z Sensora do Collectora.

Jednoczesna praca dwuzakresowa daje bardzo duże możliwości aplikacyjne, np. przy budowie inteligentnych zamków drzwiowych do zastosowań hotelowych. Gość hotelowy obsługuje w sposób bezpieczny (szyfrowanie) drzwi swojego pokoju ze swojego urządzenia mobilnego na bliską odległość. A jednocześnie załoga hotelowa ma scentralizowaną obsługę i bezpieczny nadzór nad wszystkimi drzwiami hotelu z transmisją dłuższego zasięgu.

Czy może być coś jeszcze lepszego? Może, oczekiwane jest rozszerzenie protokołu Thread na pasmo Sub-1GHz. Thread pracuje z wykorzystaniem protokołu komunikacyjnego IEEE 802.15.4. W kolejnej warstwie została zastosowana sieć 6LoWPAN (Low-Power Wireless Personal Area Networks), która umożliwia pracę sieci w konfiguracji „mesh” oraz adresowanie IPv6. Pozwoli to na drastyczne zwiększenie obsługiwanych funkcjonalności przy zastosowaniu dotychczasowego układu scalonego.

Henryk A. Kowalski
Instytut Informatyki
Politechnika Warszawska

Miesięcznik „Elektronika Praktyczna” (12 numerów w roku) jest wydawany przez AVT-Korporacja Sp. z o.o. we współpracy z wieloma redakcjami zagranicznymi.

Wydawca:



AVT-Korporacja Sp. z o.o.
03-197 Warszawa, ul. Leszczyńska 11
tel.: 22 257 84 99, faks: 22 257 84 00

Adres redakcji:

03-197 Warszawa, ul. Leszczyńska 11
tel.: 22 257 84 60
faks: 22 257 84 00
e-mail: redakcja@ep.com.pl
www.ep.com.pl

Redaktor Naczelny:

Wiesław Marciniak

Redaktor Programowy,

Przewodniczący Rady Programowej:
Piotr Zbysiński

Zastępca Redaktora Naczelnego,

Redaktor Prowadzący:
Damian Sosnowski

Zastępca Redaktora Naczelnego,

Menedżer Magazynu
Marcin Karbowniczek

Szef Pracowni Konstrukcyjnej:

Grzegorz Becker, tel.: 22 257 84 58

Redaktor strony internetowej www.ep.com.pl

Dariusz Welik

Zespół marketingu i reklamy:

Katarzyna Gugała, tel.: 22 257 84 64
Adam Kęska, tel.: 22 257 84 63
Bożena Krzykawska, tel.: 22 257 84 42
Grzegorz Krzykowski, tel.: 22 257 84 60
Maksymilian Hoser, tel.: 22 257 84 65

Sekretarz Redakcji:

Grzegorz Krzykowski, tel.: 22 257 84 60

DTP i okładka:

MAD Sp. z o.o.

Stali Współpracownicy:

Jacek Bogusz, Lucjan Bryndza, Jarosław Doliński,
Andrzej Gawryluk, Krzysztof Górski, Tomasz Jabłoński,
Michał Kurzela, Szymon Panecki, Sławomir Skrzyński,
Ryszard Szymaniak, Adam Tatuś, Robert Wołgajew

Uwaga!

Kontakt z wymienionymi osobami jest możliwy via e-mail, według schematu: imię.nazwisko@ep.com.pl

Prenumerata w Wydawnictwie AVT

www.avt.pl/prenumerata

lub tel.: 22 257 84 22

e-mail: prenumerata@avt.pl

www.sklep.avt.pl, tel.: 22 257 84 66

Prenumerata w RUCH S.A.

www.prenumerata.ruch.com.pl

lub tel.: 801 800 803, 22 717 59 59

e-mail: prenumerata@ruch.com.pl



Wydawnictwo
AVT-Korporacja Sp. z o.o.
należy do Izby Wydawców Prasy

Copyright AVT-Korporacja Sp. z o.o.

03-197 Warszawa, ul. Leszczyńska 11

Projekty publikowane w „Elektronice Praktycznej” mogą być wykorzystywane wyłącznie do własnych potrzeb. Korzystanie z tych projektów do innych celów, zwłaszcza do działalności zarobkowej, wymaga zgody redakcji „Elektroniki Praktycznej”. Przedruk oraz umieszczenie na stronach internetowych całości lub fragmentów publikacji zamieszczonych w „Elektronice Praktycznej” jest dozwolone wyłącznie po uzyskaniu zgody redakcji. Redakcja nie odpowiada za treść reklam i ogłoszeń zamieszczanych w „Elektronice Praktycznej”.



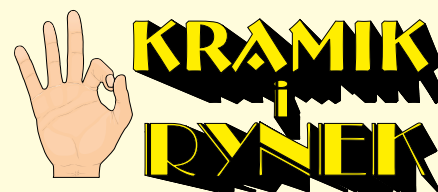
Ogłoszenia i reklamy hurtowni, sklepów, importerów, producentów, dealerów, itp. są płatne. Cena podstawowego modułu (35x20 mm) wynosi 66 zł + VAT. Koszt minimalnej ramki dla ogłoszeń o wielkości 3 modułów wynosi 198 zł + VAT. Moduły można łączyć zarówno w pionie jak i w poziomie. Maksymalna szerokość ogłoszenia to 5 modułów, wysokość 12 modułów. Rabaty stosujemy wyłącznie dla reklam powyżej 8 modułów: 4-6 emisji 10%, 7-11 emisji 15% i od 12 emisji 25%.

Oferta specjalna:

- publikacja fragmentów cennika w ramce o wielkości: 8 modułów w pionie cena 264 zł + VAT, 9 modułów w poziomie 305 zł + VAT
- rabat specjalny dla firm poszukujących pracowników wynosi 25% (wyłącznie dla dużych reklam).

Wszelkich informacji udziela **Grzegorz Krzykawski, tel. 22 257 84 60**, e-mail: grzegorz.krzykawski@ep.com.pl.

Reklamy do tej rubryki mogą być przygotowane przez Zamawiającego w CMYK, w formacie CDR (wersja do 2018), AI, EPS (tekst zmieniony na krzywe), PSD, PDF (programy Adobe w wersji do CC 2019 włącznie PC/Mac) z próbnym wydrukiem. Małe reklamy mogą być przygotowane w redakcji (gratis) na podstawie odręcznego szkicu. Opracowania te nie będą jednak wówczas uzgadniane z Zamawiającym przed oddaniem do druku.



Redakcja nie odpowiada za treść reklam i ogłoszeń zamieszczonych w Elektronice Praktycznej

Hurtownia elementów elektronicznych "AKSOTRONIK" zaprasza do swojego sklepu internetowego. Zaloguj się i kupuj ON-LINE na naszej stronie:

WWW.AKSOTRONIK.COM.PL

Przetwornica Step-Down LM2596 3,40 PLN
Moduł WiFi do Arduino 16,60 PLN
Elektromagnes 6V oraz 12V Cena od 15,90 PLN
Woltomierze LED AC oraz DC Cena od 7,50 PLN
Kable do płyt stykowych Cena od 1,30 PLN
Zarówki LED do samochodu Cena od 1,60 PLN
Moduły z przekątnikiem Cena od 3,60 PLN
Moduł bluetooth HC05 do Arduino Cena 20 PLN
Czujniki indukcyjne Cena od 7 PLN
Anteny WiFi oraz GSM Cena od 2,80 PLN
Silniki krokowe 200 oraz 400 kroków Cena od 30 PLN

Aksotronik
ELEMENTY ELEKTRONICZNE
WWW.AKSOTRONIK.COM.PL

Uwaga!!! Powyższe ceny dotyczą zakupów minimalnych ilości hurtowych, poprzez nasz sklep internetowy. W swojej ofercie posiadamy m.in.: półprzewodniki (diody, układy scalone, tranzystory, triaki, elementy optoelektroniczne), elementy dystansowe, złączki, przełączniki, elementy akustyczne, rezystory, kondensatory, kwarce, podstawki, moduły Arduino.

Zapraszamy do kontaktu: INFO@aksotronik.com.pl, tel: (22) 783-20-51

RACK i Eurocarta 19" Wyposażenie szaf 19"
www.obudowa.pl
Producent obudów dla elektroniki tel. 032-230-2301

KITY AVT

<http://bit.ly/2FNvrIN>

www.ep.com.pl

www.delta.poznan.pl

Wielkość matrycy: 2,1 Mpx (Sony IMX290)
Obiektów: 2,8 - 12 mm (kąt widzenia 85° - 23°)
Rozdzielczość: 1920 x 1080 (1080p)
Zasięg oświetlacza IR: 60 m
Protokoły sieciowe: TCP/IP, NTP, HTTP, ONVIF 2.4
PoE 802.3af, Klasa szczelności IP66
3D-DNR, WDR, detekcja ruchu, H.265 / H265+
Aplikacje dla: Android, iOS
Bardzo dobra jakość w jeszcze lepszej cenie

▶ **APTI-250C61-2812WP**

Wielkość matrycy: 2,1 Mpx (przetwornik 1/2.9")
Obiektów: 3,6 mm (kąt widzenia 69°)
Rozdzielczość: 1920 x 1080 (1080p)
Przepływność (bitrate): 64 ... 12000 kbit/s
Protokoły sieciowe: TCP/IP, NTP, HTTP, ONVIF 2.4
PoE 802.3af, Klasa szczelności IP66
3D-DNR, WDR, detekcja ruchu, H.265 / H265+
Aplikacje dla: Android, iOS
Bardzo dobry stosunek jakości do ceny

▶ **APTI-201D2-36WP**

Wielkość matrycy: 4,0 Mpx (przetwornik 1/2.5")
Obiektów: 2,8 - 12 mm (kąt widzenia 81° - 23°)
Rozdzielczość: 2560 x 1440
Obsługa audio - zewnętrzny mikrofon
Protokoły sieciowe: TCP/IP, NTP, HTTP, ONVIF 2.4
PoE 802.3af, Klasa szczelności IP66, H.265
3D-DNR, F-DNR (Defog), WDR, detekcja ruchu, Aplikacje dla: Android, iOS
Wysoka rozdzielczość dostępna w atrakcyjnej cenie

▶ **APTI-41VA3-2812WP**

Największy wybór w Polsce

ELMAX 1988
OBWODY DRUKOWANE
Produkcja, Projektowanie, Montaż

Certyfikat Underwriters Laboratories
UL 94V-0 E480148 TYPE 1

Zakład produkcyjny:
05-260 Marki
ul. Duża 1
tel. 22 781 63 95
22 761 95 80
fax. 22 781 63 95 w 23
www.elmax.waw.pl
elmax@elmax.waw.pl

Płytki jednostronne
Płytki dwustronne
Płytki na podłożu aluminium
Płyty czolowe FR4

Serie dowolne
Prototypy
Maksymalny wymiar płytek 1w 630 mm

Dokumentacja technologiczna
Dokumentacja konstrukcyjna
Trawione szablony SMD

Aktywny kalkulator prototypów na stronie internetowej

Montaż elektroniki
Krótkie terminy
Wykonania super expresowe

Pokrycie Sn lub SnPb inne na życzenie
Maski, opisy montażowe w różnych kolorach



Wykaz firm ogłaszających się w tym numerze „Elektroniki Praktycznej”

Reklamy stron internetowych na str. 10

AET	108
AKSOTRONIK.....	137
ARMEL	137
BORNICO.....	8
COMPUTER CONTROLS.....	11
DELTA-OPTI.....	137
DPS SOFTWARE.....	86
ELMARK AUTOMATYKA	12
ELMAX.....	137
FALDRUK	25
FARNELL ELEMENT14.....	140
FAULHABER	85
GAMMA	10
GLYN	83
HAMMOND	9
INDEL	106
MARTHEL	79
MICROCHIP	7
NORATEL	107
PIEKARZ	10
RELPOL	13
RENEX	10
RS COMPONENTS.....	2, 63
RUTRONIK.....	75
ST MICROELECTRONICS.....	
.....	3, 35, 43, 45, 47
TELZAM	109



Bezprzewodowa EKO stacja do pomiaru jakości powietrza

Stacja pomiarowa zasilana energią słoneczną na bazie bezprzewodowego, laserowego sensora pyłów zawieszonych BBair z Bluetooth LE. Wyniki pomiarów prezentowane w atrakcyjnej formie na graficznym wyświetlaczu LCD, który można powiesić na ścianie lub postawić na biurku. Wystarczy jedno spojrzenie, aby wiedzieć co dzieje się na zewnątrz: jaki jest poziom stężenia pyłów zawieszonych, czy jest ciepło czy może zimno i czy powietrze jest przynikliwe wilgotne, czy może przyjemnie suche. Równie łatwo można wszystko to sprawdzić przy pomocy telefonu z aplikacją BBair.

Przenośny rejestrator temperatury

Urządzenie przeznaczone do długotrwałych pomiarów i rejestracji temperatur. Do przechowywania danych używa wymiennego, łatwo dostępnego nośnika pamięci masowej. Wyniki są zapisywane w formie pliku tekstowego. Specjalnie napisany program komputerowy ułatwia ustawianie parametrów, kontrolę przebiegu cyklu pomiarowego i wstępną wizualizację zarejestrowanych pomiarów.

Sztuczne obciążenie zmiennoprądowe w.cz. o impedancji 50 Ω

Urządzenie przeznaczone przede wszystkim do pracy z wysokimi częstotliwościami radiowymi (w.cz.) z przedziału od 100 kHz do 30 MHz. Przyrząd absolutnie niezbędny w warsztacie każdego radioamatora – konstruktora. Głównie przeznaczenie, to wzorcowe obciążenie wyjścia wzmacniacza mocy w.cz. o dopasowanej rezystancji $R_{load}=50\ \Omega$. Jest niezwykle przydatne w trakcie wszelkich prób technicznych oraz napraw, regulacji a także strojenia wzmacniaczy.

Minimalistyczna płytką do prototypowania z FPGA

Układy FPGA z rodziny Intel MAX10 są wyjątkowo ciekawe i warte zastosowania we własnych projektach. Spowodowane jest to kilkoma cechami rzadko występującymi w innych podzespołach tego typu. Pierwszą zaletą jest zintegrowana nieulotna pamięć Flash, która pozwala nie tylko na przechowywanie konfiguracji, ale może być także używana przez logikę w czasie pracy. Drugą cechą ułatwiającą projektowanie jest możliwość zasilania za pomocą tylko jednego napięcia 3,3 V.

Półprzewodnikowa cewka Tesli o podwójnym rezonansie

W półprzewodnikowej cewce Tesli o podwójnym rezonansie (Dual Resonant Solid State Tesla Coil) iskiernik został zastąpiony obwodem elektronicznym, dzięki czemu do pracy nie jest wymagane wysokie napięcie. Dzięki zastosowaniu falownika oraz specjalnego sterownika możliwa jest regulacja i kontrola parametrów pracy takich jak: wartości prądu i napięcia w obwodzie, ilość pobieranej mocy czy częstotliwość i wypełnienie sygnału.

a ponadto tematy wiodące EP11:

SERWONAPĘDY I SILNIKI KROKOWE

oraz

SYSTEMY EMBEDDED



Politechnika
Śląska

SIEMENS

Ingenuity for life

KAMAMI

Instytut Elektroniki Politechniki Śląskiej w Gliwicach wraz z firmami SIEMENS oraz KAMAMI ogłasza konkurs dla uczniów szkół ponadgimnazjalnych na najciekawszy projekt z dziedziny elektroniki, informatyki lub automatyki pod hasłem:

elektronika

- by żyło się łatwiej

EDYCJA 2020

Do konkursu można zgłaszać indywidualne, jak i zespołowo (maksymalnie 2 os.) realizowane projekty takie jak: układy elektroniczne, konstrukcje mechatroniczne, roboty, modele pojazdów, konstrukcje latające, aplikacje sterowników PLC, układy automatyki, aplikacje na smartfony i tablety, programy komputerowe, pomoce dydaktyczne i inne.

Autorzy najciekawszych rozwiązań zostaną zaproszeni do zaprezentowania swoich prac na Politechnice Śląskiej w Gliwicach

Na laureatów czekają cenne nagrody!

Więcej na: konkurs.iele.polsl.pl

Konkurs objęty patronatem
Ministerstwa Nauki i Szkolnictwa Wyższego
oraz Honorowym Patronatem Prezydenta Miasta Żory

PATRONAT:



Ministerstwo Nauki
i Szkolnictwa Wyższego



PATRONAT MEDIALNY:

ELEKTRONIKA
PRAKTYCZNA

MIKRO
KONTROLER

Elektronika
dla wszystkich



RADIO 107.9 FM

Do konkursu można zgłaszać projekty oryginalne, jak również projekty już prezentowane w innych konkursach. Przyjmowane będą zarówno projekty indywidualne, jak i tworzone w zespołach dwuosobowych – np. będące efektem działalności kół zainteresowań.

Ważne daty: 15 grudnia 2019 – termin rejestracji uczestników • 4 stycznia 2020 – termin nadsyłania projektów konkursowych • 6 marca 2020 – uroczysty finał konkursu

Sprawdź nasz największy asortyment od wiodących dostawców

- Produkty w magazynie, gotowe do wysyłki
- Oferta produktowa od etapu projektowania, po utrzymanie ruchu
- Doskonała, lokalna obsługa klienta
- Kwotacje na produkty spoza katalogu i niestandardowe produkty dostosowane do indywidualnych potrzeb klienta



Honeywell

OMRON

NXP

ON Semiconductor **ON**

ADI maxim integrated.

Panasonic

KEYSIGHT
TECHNOLOGIES

multicomp PRO



Life Is On | **Schneider**
Electric

ST *life.augmented*

VISHAY

RENESAS

MICROCHIP

ROHDE & SCHWARZ

FLUKE

Tektronix



molex

Weller

pl.farnell.com

 **Farnell**
AN AVNET COMPANY