Wydanie elektroniczne dostępne na www.ulubionykiosk.pl

ELEKTRONIKA PRAKTYCZNA

EP.com.pl

● Międzynarodowy magazyn elektroników konstruktorów ● styczeń ● 1/2024 ●

Tylko Prenumeratorzy

- mają dostęp do artykułów przed ich publikacją w EP na www.ep.com.pl – **EP W TOKU**
- mają dostęp do materiałów dodatkowych, takich jak pliki źródłowe projektów na naszym serwerze **FTP** www.ulubionykiosk.pl/media



ROBOTYKA W MEDYCYNIE

TEMAT NUMERU

inspirujące, użyteczne projekty

• AVTDuino UNO R4 Plus • Odbiornik nasłuchowy na pasmo KF 80 m „Kubuś” • e-Nurse – elektroniczny dyspenser leków • Uniwersalny przedwzmacniacz • Progowy licznik impulsów • IP-Coach IV. Reaktywacja • Bufor I²C Grove • Mostek master I²C/1-Wire w standardzie Grove • Mechaniczny potencjometr cyfrowy • MIDIBox. Niezwykły kontroler MIDI • Zamek do drzwi oparty na Arduino, który wykrywa rytm pukania

podzespoły, sprzęt, aplikacje

• Internet Rzeczy w pomiarach środowiskowych. Nordic Thingy:53 – platforma prototypowania IoT z uczeniem maszynowym (ML) • Efekt do gitary na bazie LPC5528 • Rewolucja w świecie audio – subminiaturowe głośniki xMEMS • Robotyka w medycynie • Zaawansowane obwody PCB do najbardziej wymagających aplikacji • QWERTY – Twój sprawdzony dostawca klawiatur membranowych i paneli dotykowych • Rozwiązania dla nowoczesnych interfejsów HMI w ofercie firmy Semicon • Klawiatury foliowe – przegląd technologii

kursy

• Kurs FPGA Lattice. Pamięć EBR

ISSN 1230-3526 Indeks 357677

18,90 zł (w tym 8% VAT) • PRICE: 8 EUR

KLAWIATURY FOLIOWE, MEMBRANOWE I NA SZKLE

eprasa.pl 1fc8f31a49



Zaprenumeruj „Elektronikę Praktyczną", a zawsze dostaniesz najnowszy numer wprost do Twojej skrzynki!

na start
do 6* wydań gratis

po 5 latach nieprzerwanej prenumeraty
do 12* wydań gratis

* Cena prenumeraty rocznej **na start** wynosi 207,90 zł. Przy zamówieniu prenumeraty dwuletniej za 340,20 zł oszczędność wynosi równowartość sześciu wydań „Elektroniki Praktycznej".

Przedłużasz prenumeratę? Aby otrzymać zniżkę lojalnościową, przedłuż prenumeratę po zalogowaniu się do swojego panelu na **www.ulubionykiosk.pl**, gdzie znajdziesz atrakcyjną ofertę prenumeraty, która uwzględnia przysługujące Ci zniżki za lojalność. Po 5 latach nieprzerwanej prenumeraty otrzymasz **rabat 50%** na prenumeratę dwuletnią.

Oferta dotyczy prenumeraty drukowanej.

Wszystkie opcje prenumeraty i e-prenumeraty znajdziesz na stronie

www.UlubionyKiosk.pl

prenumerata@avt.pl

AVT-Korporacja sp. z o.o., ul. Leszczynowa 11, 03-197 Warszawa, konto 18 1050 1012 1000 0024 3173 1013

eprasa.pl 1fc8f31a49

Nowe otwarcie



Szanowni Czytelnicy,

wraz ze styczniowym wydaniem „Elektroniki Praktycznej” obejmuję stanowisko redaktora naczelnego. Ta zaszczytna funkcja w kultowym już czasopiśmie dla elektroników jest dla mnie nie lada wyzwaniem – doskonale zdaję sobie bowiem sprawę, że nasi Czytelnicy są odbiorcami naprawdę wymagającymi, nie ma więc miejsca na jakiekolwiek potknięcie. Zadanie kierowania pracami Redakcji jest dla mnie tym donioślejsze, że sam – podobnie, jak tysiące innych elektroników w Polsce – wychowałem się na czasopismach AVT.

Będziemy oczywiście utrzymywać kierunek rozwoju, konsekwentnie realizowany przez Damiana Sosnowskiego przez ostatnich kilka lat. Aby jednak dać Czytelnikom konkretny przedsmak czekających na nich niespodzianek, uchylę rąbka tajemnicy w kwestii konkretnych planów, jakie powzięliśmy w Redakcji na najbliższe miesiące i lata. Jednym z nich jest znaczna rozbudowa zespołu stałych Autorów – część z nich jest naszym Czytelnikom doskonale znana i powróci do nas po dłuższej lub krótszej przerwie od publikowania, inni dopiero zadebiutują na łamach „Elektroniki Praktycznej”, wnosząc cenne doświadczenia i wiedzę teoretyczną. Na tym przecież polegała istota EP od początku jej istnienia i w takim właśnie kierunku będziemy dążyć – i to zarówno pod względem projektów, jak i artykułów przeglądowych czy też poradników.

Drugi spośród głównych obszarów zmian to znaczna rozbudowa działu Kursów EP – wiemy, że materiały tego typu od zawsze cieszyły się ogromnym zainteresowaniem naszych Czytelników i dlatego to właśnie na rozmaite materiały szkoleniowe położymy największy nacisk obok projektów autorskich. Już teraz trwają intensywne prace organizacyjne i koncepcyjne nad szeregiem kursów, które w większości przypadków będą pierwszymi tego typu materiałami w Polsce, a niektóre z nich – według naszej najaktualniejszej wiedzy – będą unikalne nawet w skali międzynarodowej.

Trzeci ze strategicznych obszarów zmian – którego wdrażanie zajmie pewnie najwięcej czasu – to uruchomienie nowych rubryk stałych. Podczas ich wprowadzania na łamy „Elektroniki Praktycznej” zadbamy o to, by były skorelowane z istniejącymi rubrykami – tak by ciekawe materiały dla siebie znajdowali na łamach naszego pisma zarówno stali Czytelnicy cyklów redakcyjnych, jak i osoby poszukujące nowości.

W bieżącym wydaniu publikujemy szereg nowych, interesujących projektów oraz kontynuujemy opisy urządzeń, które pojawiły się na łamach EP w listopadzie oraz grudniu 2023. Mamy m.in. gratkę dla początkujących radioamatorów – projekt odbiornika nasłuchowego „Kubuś”, stanowiący kompilację sprawdzonych rozwiązań z popularnego odbiornika „Dosia”, bazujący tym razem na pojedynczej płytce drukowanej, integrującej wszystkie najważniejsze funkcje urządzenia nasłuchowego na pasmo KF 80 m. W kolejnych częściach krótkiego cyklu, poświęconego tej ciekawej konstrukcji, przedstawimy nie tylko wszystkie zagadnienia montażowe i uruchomieniowe, ale także wyniki skrupulatnych pomiarów, wykonanych przez Autora. W numerze styczniowym znalazł się także obszerny zestaw Miniprojektów (wśród nich m.in. kilka modułów, kompatybilnych ze standardem Grove), interesujący remake popularnej płytki Arduino Uno R4 czy też projekt nakładki, pozwalającej „ożywić” leciwy już system dydaktyczny IP Coach, do dziś znajdujący się w laboratoriach niektórych uczelni wyższych i szkół.

Na styczniowej okładce zagościły natomiast dwa tematy główne – technologie klawiatur membranowych i szklanych paneli dotykowych, a także robotyka medyczna – interesujący i niezwykle złożony obszar technologii, coraz śmielej wdzierający się do codziennej praktyki klinicznej w szpitalach w kraju i za granicą.

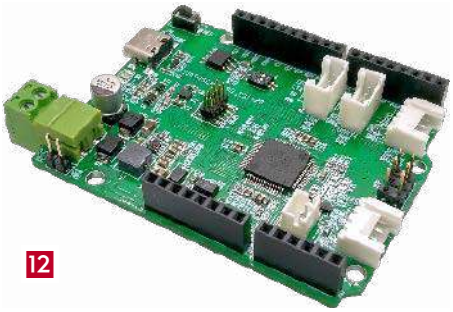
Zapraszamy do lektury, życząc wszystkim naszym Czytelnikom wszelkiej pomyślności w Nowym Roku 2024!

Przemysław Musz

eprasa.pl 1fc8f31a49

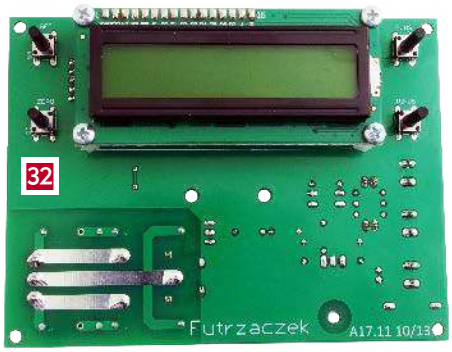
ELEKTRONIKA PRAKTYCZNA

Nr 1 (372) • Styczeń 2024

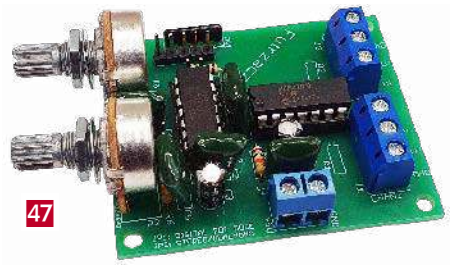

12



16

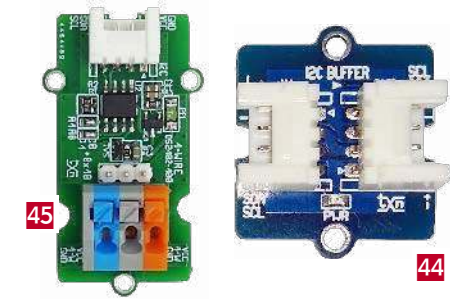


32


47



29


45

44

Nie przeocz

Nowe podzespoły 5
Dodaj do obserwowanych 10
Konkurs: Wygraj płytkę rozwojową Microchip PIC-IoT WG 37
Koktajl niusów 104

Projekty

AVTDuino UNO R4 Plus 12
Odbiornik nasłuchowy na pasmo KF 80 m „Kubuś" (1) 16
e-Nurse – elektroniczny dyspenser leków (2) 24
Uniwersalny przedwzmacniacz (3) 29
Progowy licznik impulsów 32

Projekty czytelników

IP-Coach IV. Reaktywacja 38

Miniprojekty

Bufor I²C Grove 44
Mostek master I²C/1-Wire w standardzie Grove 45
Mechaniczny potencjometr cyfrowy 47

Moduły w aplikacjach

Internet Rzeczy w pomiarach środowiskowych (1). Nordic Thingy:53 – platforma prototypowania IoT z uczeniem maszynowym (ML) 64
Efekt do gitary na bazie LPC5528 70

Projekty SOFT

MIDIBox (2). Niezwykły kontroler MIDI 84
Zamek do drzwi oparty na Arduino, który wykrywa rytm pukania 88

Temat numeru: Robotyka w medycynie

Robotyka w medycynie (1) 50

Podzespoły

Rewolucja w świecie audio – subminiaturowe głośniki xMEMS 62

Prezentacje

Zaawansowane obwody PCB do najbardziej wymagających aplikacji 59
QWERTY – Twój sprawdzony dostawca klawiatur membranowych i paneli dotykowych 74
Rozwiązania dla nowoczesnych interfejsów HMI w ofercie firmy Semicon 82

Elektronika w praktyce

Klawiatury foliowe – przegląd technologii 76

Kursy

Kurs FPGA Lattice (15). Pamięć EBR 93

Prenumerata 2
Od wydawcy 5
Hity następnego numeru 107

Dodatkowe materiały do artykułów znajdziesz na https://ulubionykiosk.pl/media

eprasa.pl 1fc8f31a49

nowe podzespoły

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



Przekaźnik dużej mocy G7EB firmy Omron o prądzie znamionowym zwiększonym do 120 A

Przekaźnik dużej mocy G7EB firmy Omron jest obecnie dostępny w wersji G7EB-1A-E o dopuszczalnym prądzie wyjściowym zwiększonym ze 100 do 120 A @ 800 VAC. Jest to przekaźnik SPST-NO polecany do zastosowań w testerach półprzewodników i systemach konwersji mocy, wyróżniający się na tle produktów konkurencyjnych małą rezystancją kontaktów (<5 mΩ) i małą emisją ciepła. Dzięki zwiększonej wartości prądu obciążenia może też znaleźć zastosowanie w jednofazowych stacjach ładowania EV Mode3 o mocy 19,2 kW oraz Mode4 o mocy 20 kW.

Dzięki małej emisji ciepła G7EB-1A-E jest przekaźnikiem o długim czasie bezawaryjnej pracy, niewymagającym chłodzenia przy użyciu radiatorów. Przy pracy z maksymalnym prądem przewodzenia jego temperatura wzrasta o około 40°C, podczas gdy w innych podobnych modelach przyrost sięga nawet 80°C. G7EB może pracować w temperaturze otoczenia od –40 do +85°C. Wytrzymuje impulsy przepięciowe do 10 kV (1,2/50 µs) pomiędzy cewką i kontaktami. Jest zamykany w obudowie do montażu THT o wymiarach 50,5×40,5×37 mm.

www.components.omron.com

Moduł nawigacyjny IoT z komunikacją naziemną LTE-M/NB-IoT i satelitarną ORBCOMM



Firma u-blox wprowadza na rynek swój pierwszy moduł nawigacyjny z obsługą komunikacji naziemnej LTE-M/NB-IoT i komunikacji satelitarnej ORBCOMM IDP, oznaczony symbolem SARA-S520M10L. Może on znaleźć zastosowanie w systemach zarządzania flotą pojazdów, śledzenia zasobów, transporcie morskim, górnictwie, rolnictwie, systemach antykradzieżowych i monitoringu oraz wszelkiego typu aplikacjach, wymagających ciągłej łączności ze względów bezpieczeństwa.

SARA-S520M10L jest najmniejszym tego typu modułem spośród dostępnych obecnie na rynku, produkowanym w obudowie LGA o wymiarach 16,0×26,0×2,2 mm. Zapewnia kompatybilność pod względem rozkładu wyprowadzeń z wcześniejszymi modułami rodziny SARA, komunikującymi się przez sieć telefonii komórkowej. Obsługuje satelity 4 konstelacji: GPS, GLONASS, BeiDou i Galileo.

SARA-S520M10L pracuje z napięciem zasilania od 3,0 do 4,5 V przy poborze prądu wynoszącym do 195 mA (komunikacja LTE Cat M1, +23 dBm). Może być przełączany w tryb oszczędnościowy, w którym pobór prądu zmniejsza się do 1 µA. Zawiera interfejsy UART, DDC/I²C i USB (diagnostyczny) oraz 6 konfigurowalnych linii I/O. Został przystosowany do pracy w temperaturze otoczenia od –40 do +85°C.

www.u-blox.com



Wodoszczelny mikrofon MEMS do pracy w ciężkich warunkach środowiskowych

Oddział audio firmy CUI Devices wprowadza do oferty kolejny miniaturowy mikrofon, zrealizowany w technologii MEMS. Model CMM-3424DT-26165-TR zawiera cyfrowe wyjście PDM. Jest zamykany w wodoszczelnej obudowie SMD (4×3×1 mm) z portem na górnej powierzchni, przystosowanej do pracy w ciężkich warunkach środowiskowych. Charakteryzuje się stopniem ochrony IPX7 i szerokim zakresem dopuszczalnej temperatury pracy od –40 do +100°C.

CMM-3424DT-26165-TR to mikrofon o dookólnej charakterystyce i paśmie od 100 Hz do 10 kHz. Charakteryzuje się czułością –26 dB (±1 dB) i stosunkiem sygnał/szum równym 65 dBA. Może być

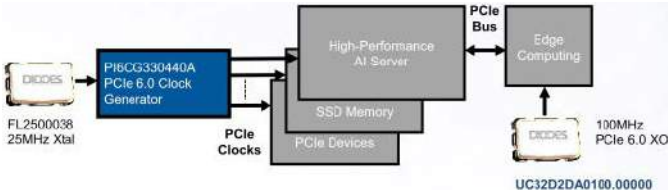
REKLAMA



eprasa.pl 1fc8f31a49

montowany w procesie przepływowym. Pracuje z napięciem zasilania od 1,6 do 3,6 V, pobierając maks. 700 µA prądu w stanie aktywnym i 50 µA w trybie standby. Jego ceny hurtowe zaczynają się od 1,82 USD przy zamówieniach 1000 sztuk.

www.cuidevices.com



Energooszczędne oscylatory kwarcowe z wyjściem HCSL do interfejsów PCIe 6.0/5.0

Energooszczędne oscylatory kwarcowe rodziny UC zostały zaprojektowane specjalnie do taktowania interfejsów PCIe 6.0/5.0 w szybkich aplikacjach sieciowych i komputerowych. Ich wyjścia HCSL (high-speed current steering logic) zapewniają kompatybilność oddolną również ze starszymi wariantami interfejsów, począwszy od PCIe 1.0 oraz z innymi szybkimi interfejsami, korzystającymi z zegarów HCSL.

	Zasilanie	Częstotliwość	Temperatura pracy	Stabilność	Obudowa
UC2543FA0100.000000	1,8 V	100 MHz	–40...+105°C	±50 ppm	2,5×2,0 mm
UC25D2EC0133.300000	2,5 V	133 MHz	–40...+85°C	±25 ppm	2,5×2,0 mm
UC25D1DB0156.250000	3,3 V	156,25 MHz	–40...+85°C	±50 ppm	2,5×2,0 mm
UC32D2DA0100.000000	2,5 V	100 MHz	–40...+85°C	±25 ppm	3,2×2,5 mm
UC32D3DA0100.000000	1,8 V	100 MHz	–40...+85°C	±25 ppm	3,2×2,5 mm
UC32D1HB0156.250000	3,3 V	156,25 MHz	–40...+125°C	±70 ppm	3,2×2,5 mm
UC50D2DA0100.000000	2,5 V	100 MHz	–40...+85°C	±25 ppm	5,0×3,2 mm
UC50D3DA0100.000000	1,8 V	100 MHz	–40...+85°C	±25 ppm	5,0×3,2 mm
UC50D1DB0156.250000	3,3 V	156,25 MHz	–40...+85°C	±25 ppm	5,0×3,2 mm

Oscylatory rodziny UC charakteryzują się małym błędem jitteru, wynoszącym w przypadku wariantów kategorii C nawet poniżej 100 fs RMS, co zapewnia zgodność ze specyfikacją PCIe 6.0. Pracują z napięciem zasilania od 1,8 do 3,3 V przy poborze prądu mniejszym o połowę od standardowych układów HCSL. Są zamykane w szczelnych, ceramicznych obudowach SMD o powierzchni od 2,5×2,0 mm do 7,0×5,0 mm. Nie wymagają współpracy z zewnętrznymi rezystorami. Są odporne na temperaturę otoczenia z zakresu od –40 do +125°C.

www.diodes.com

Polimerowe kondensatory hybrydowe odporne na impulsy prądowe do 6,1 A rms



Polimerowe kondensatory hybrydowe nowej serii ZUU charakteryzują się odpornością na duże impulsy prądowe do 6,1 A rms, bardzo małą rezystancją ESR i szerokim zakresem pojemności. W porównaniu z wcześniejszymi odpowiednikami, ich dopuszczalny prąd impulsowy zwiększono o 60% w temperaturze roboczej +135°C, a maksymalna pojemność wzrosła o 80% do 1000 µF. Jeśli chodzi o rezystancję ESR, wynosi ona w zależności od modelu od 8 mΩ.

Kondensatory ZUU wykazują żywotność 4000 h w maksymalnej temperaturze roboczej. Uzyskały kwalifikację AEC-Q200, świadczącą o dużej odporności na ciężkie warunki pracy w elektronice samochodowej. Mogą też znaleźć zastosowanie w telekomunikacji i przemyśle. Na indywidualne zamówienia mogą być dostarczane w wersjach o odporności na udary mechaniczne zwiększonej z 10 g do 30 g.

Ważniejsze parametry:

- zakres pojemności: 120...1000 µF,
- napięcia znamionowe: 25, 35, 50, 63 V,
- prąd upływu: 75,6...350 µA,
- zakres temperatury roboczej: –55...+135°C,
- wymiary obudów: od ∅ 10×12,5 mm do ∅ 10×16,8 mm.

http://industry.panasonic.eu



Seria 24-bitowych czujników ciśnienia do aparatury medycznej

W dziedzinie technologii medycznej precyzja i niezawodność są kluczowe, zwłaszcza jeśli chodzi o respiratory i urządzenia do dostarczania tlenu o dużej przepustowości. Firma Superior Sensor Technology opracowała rodzinę 7 czujników o szerokim zakresie dynamicznym, stabilnej wartości zerowej i krótkim czasie reakcji, przeznaczonych do pomiaru przepływu oraz ciśnienia wlotowego, wdechowego, wydechowego i barometrycznego w aparaturze medycznej. Pracują one z 24-bitową rozdzielczością i częstotliwością próbkowania do 12 kHz. W zależności od wersji oferują zakresy pomiarowe:

- od ±5...±25 do ±60...±130 cmH_2O (VN025CM, VN026CM, VN130CM, VN0131CM),
- ±80...±150 psi (VN150D),
- 80...150 psia (VN150A),
- 350...1100 mb (VN-BARO).

Dla ułatwienia oceny czujników serii VN firma Superior Sensor Technology dostarcza płytkę ewaluacyjną EK07-VN z interfejsem USB, zapewniającą łatwą integrację z komputerami PC za pomocą dostarczonego oprogramowania. Ułatwia ona rejestrowanie danych i zawiera gniazdo do szybkiej zmiany układów.

www.superiorsensors.com



Precyzyjne wzmacniacze operacyjne o dopuszczalnej temperaturze pracy +175°C

STMicroelectronics prezentuje dwa „samochodowe" wzmacniacze operacyjne z kwalifikacją AEC-Q100, zdolne do pracy w ekstremalnej temperaturze otoczenia z zakresu od –40 do +175°C: jednokanałowy TSZ181H1 i dwukanałowy TSZ182H1. Oba charakteryzują się bardzo małym wejściowym napięciem offsetu, wynoszącym typowo 3,5 µV @ +25°C (maks. 70 µV) i bardzo małym wejściowym prądem polaryzacji (typ. 30 pA @ +25°C, maks. 200 pA). W całym zakresie dopuszczalnej temperatury pracy maksymalne wejściowe napięcie offsetu i wejściowy prąd polaryzacji wynoszą odpowiednio 100 µV i 225 pA.

TSZ181H1 i TSZ182H1 są odporne na wyładowania ESD do 4 kV HBM. Mogą znaleźć zastosowanie w precyzyjnych układach kondycjonowania sygnałów, gdzie mogą pracować bez kalibracji. Charakteryzują się pasmem 2 MHz. Mogą być zasilane napięciem

eprasa.pl 1fc8f31a49

z zakresu 2,2...5,5 V. Pobierają typowo 1 mA prądu przy napięciu zasilania 5 V. Zawierają wejścia i wyjścia rail-to-rail.

Ceny hurtowe TSZ181H1 i TSZ182H1 zaczynają się odpowiednio od 1,58 USD i 2,66 USD.

www.st.com

Czujnik TPMS pracujący 10 lat na baterii CR2032

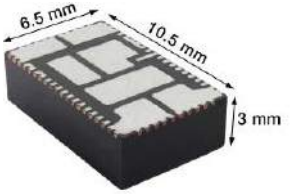


XENSIV SP49 to najnowszy czujnik do monitorowania ciśnienia w oponach (TPMS) z oferty firmy Infineon, wyprodukowany w technologii MEMS. Oprócz sensora zawiera też 32-bitowy mikrokontroler z rdzeniem ARM M0+, 19 kB pamięci flash na kod aplikacji, 1 kB pamięci RAM, 192 B pamięci RAM z retencją, interfejs sprzętowy I²C master/slave oraz symulowane programowo interfejsy UART, SPI i PWM. Pod względem rozkładu wyprowadzeń SP49 jest kompatybilny z czujnikami wcześniejszej serii SP40. Zawiera wszystkie funkcje, niezbędne do implementacji nowoczesnego modułu TPMS i wymaga podłączenia tylko kilku zewnętrznych komponentów pasywnych. Charakteryzuje się małym poborem mocy, umożliwiającym nawet 10 lat pracy przy zasilaniu z baterii CR2032.

SP49 mierzy ciśnienie w zakresie 100...920 kPa i przyspieszenie w zakresie –600...+600 g. Uzyskał kwalifikację AEC-Q100, Grade 1. Zapewnia odporność na ciśnienie do 3500 kPa, udary do 6000 g, przyspieszenie statyczne do 3500 g i wyładowania ESD do min. ±2 kV HBM. Nadaje się do pracy w temperaturze otoczenia od –40 do +125°C.

www.infineon.com

Najmniejsze w branży regulatory napięcia POL o wydajności prądowej 6, 20 i 25 A



Firma Vishay wprowadza do oferty najmniejsze na rynku regulatory napięcia POL rodziny microBRICK, dostępne w wersjach o wydajności prądowej 6 A (SiC967), 20 A (SiC931) i 25 A (SiC951). Mogą one znaleźć szeroki zakres zastosowań w układach napędowych, automatyce przemysłowej oraz do zasilania układów FPGA, SoC i ASIC w centrach danych i telekomunikacji. Są zamykane w obudowach SMD o wymiarach 10,6×6,5×3 mm. Wszystkie pracują z napięciem wejściowym z zakresu od 4,5 do 60 V. Zawierają dwa wewnętrzne tranzystory MOSFET, cewkę i kontroler, dzięki czemu wymagają jedynie minimum elementów współpracujących do konfiguracji i kompensacji pętli. Do ich zalet należy też duża gęstość mocy i duża sprawność, sięgająca 97%.

Nowe regulatory rodziny microBRICK pracują w trybie buck, umożliwiając programowanie napięcia wyjściowego w zakresie już od 0,3 V. SiC931 może być taktowany zegarem 600 kHz, 1 MHz, 1,5 MHz lub 2 MHz, natomiast w przypadku SiC967 i SiC951 częstotliwość pracy może być regulowana w zakresach odpowiednio 100 k Hz...2 MHz i 300 kHz...1,5 MHz. Wszystkie trzy modele zawierają regulowany ogranicznik prądu wyjściowego. SiC931 oferuje dodatkowo programowalny układ miękkiego startu, a SiC951 jest kompatybilny z szyną PMBus 1.3 i może być stosowany w systemach z sekwencyjnym włączaniem napięć zasilających.

SiC951 i SiC967 oferują trzy tryby pracy: *forced continuous conduction*, *ultrasonic* i *power save*, natomiast SiC931 może pracować w trybach *forced continuous conduction* i *power save*. W trybie *power save*, gdy prąd cewki przecina wartość zerową, układ sterowania wyłącza tranzystor MOSFET low side, uruchamiając tryb emulacji diody. Częstotliwość taktowania zmniejsza się proporcjonalnie do prądu obciążenia, pozwalając zapewnić możliwie największą sprawność w zakresie małych obciążeń. Architektura COT (constant on-time) zapewnia szybką odpowiedź impulsową przy minimalnej pojemności wyjściowej, małe wyjściowe napięcie tętnienia w zakresie małych obciążeń oraz stabilność pętli, niezależnie od rodzaju kondensatora wyjściowego.

Wszystkie trzy regulatory zawierają wyjście sygnalizacyjne Power Good oraz zabezpieczenie nadnapięciowe, podnapięciowe, nadprądowe, zwarciowe (z automatyczną regeneracją) i termiczne.

www.vishay.com

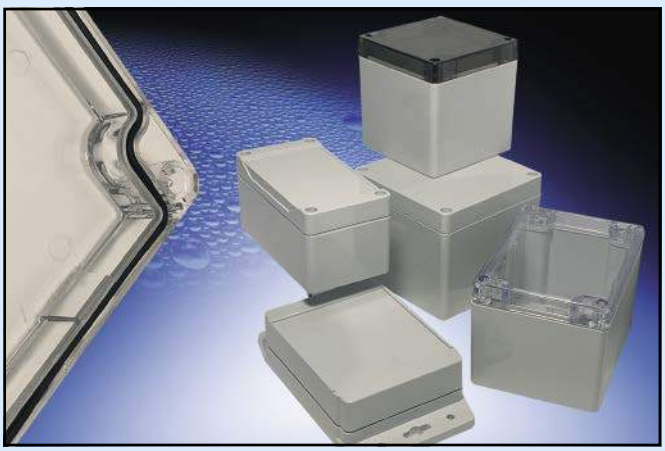


Tranzystory IGBT o napięciu przebicia 1350 V i temperaturze pracy do +175°C

STMicroelectronics wprowadza do oferty nową klasę tranzystorów IGBT, wyróżniających się napięciem przebicia zwiększonym do 1350 V i dopuszczalną temperaturą pracy +175°C. Parametry te poszerzają margines bezpieczeństwa i niezawodność w ciężkich warunkach środowiskowych, m.in. w indukcyjnych systemach grzewczych.

REKLAMA

HAMMOND®
1554
Obudowy poliwęglanu IP68
Dowiedz się więcej:
www.hammondmfg.com/1554
eusales@hammondmfg.com • + 44 1256 812812

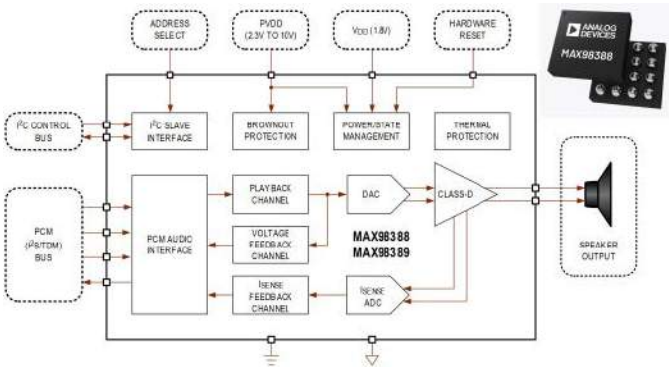


eprasa.pl 1fc8f31a49

Tranzystory serii STPOWER IH2 pozwalają zmniejszyć straty mocy podczas przewodzenia dzięki małemu napięciu nasycenia VCE(sat), wynoszącemu typowo 1,7 V przy pracy z pełnym obciążeniem. Wewnętrzna dioda zabezpieczająca charakteryzuje się małym spadkiem napięcia i małą energią wyłączania, co pozwala zwiększyć sprawność w quasi-rezonansowych konwerterach DC-DC, pracujących z częstotliwością taktowania od 16 do 60 kHz. Ogólnie nowe tranzystory STPOWER IH2 pozwalają zredukować straty mocy do 11% w aplikacjach 2 kW w porównaniu z wcześniejszymi odpowiednikami. Ich dodatni współczynnik temperaturowy napięcia $V_{CE(sat)}$ i mały rozrzut parametrów pomiędzy różnymi egzemplarzami ułatwiają równoległe łączenie kilku tranzystorów w aplikacjach dużej mocy.

Pierwsze dwa tranzystory, oferowane w ramach serii STPOWER IH2, to STGWA25IH135DF2 i STGWA35IH135DF2 o prądach drenu odpowiednio 25 A i 35 A. Oba są zamykane w standardowych obudowach TO-247. Ich ceny hurtowe wynoszą odpowiednio 1,39 USD i 1,69 USD przy zamówieniach 1000 sztuk.

www.st.com



Wzmacniacze audio klasy D z układem sprzężenia zwrotnego IV i zabezpieczeniem podnapięciowym

MAX98388 i MAX98389 to tanie, miniaturowe wzmacniacze audio klasy D z wejściem cyfrowym, zabezpieczeniem podnapięciowym i zintegrowanym układem sprzężenia zwrotnego IV (mierzącym prąd wyjściowy i napięcie wyjściowe głośnika). Zakres napięcia zasilania rozciąga się od 2,3 do 10 V, przy czym MAX98388 został zoptymalizowany do pracy z napięciem 5,5 V (z pojedynczego ogniwa), a MAX98389 z napięciem 5...10 V (z dwóch ogniw). Oba wzmacniacze zawierają układy aktywnego ograniczania emisji elektromagnetycznej (AEL) i modulacji spread-spectrum (SSM), pozwalające wyeliminować konieczność stosowania zewnętrznych filtrów, wymaganych w tradycyjnych wzmacniaczach klasy D.

MAX98388 i MAX98389 zawierają kanały do precyzyjnego pomiaru prądu i napięcia wyjściowego. Zebrane dane mogą być przesyłane do sekcji wejściowej w celu realizacji algorytmów ulepszania dźwięku, podbicia basów, ochrony głośników i funkcji haptycznych przez procesor audio DSP. Do standardowego wyposażenia należy też zabezpieczenie podnapięciowe, termiczne i nadprądowe. Wejściowy interfejs PCM obsługuje typowe formaty danych, jak I²S, left justified i TDM. Unikalny układ taktowania eliminuje potrzebę stosowania zewnętrznego zegara referencyjnego dużej częstotliwości, co pozwala zredukować powierzchnią układu i liczbę wyprowadzeń, a także zmniejsza poziom generowanych zaburzeń EMI. Wyprowadzenia wzmacniaczy zaprojektowano tak, aby uniknąć konieczności stosowania przelotek na płytkach drukowanych, co obniża koszty.

MAX98388 i MAX98389 są zamykane w obudowach WLP-16 o rozstawie wyprowadzeń 0,4 mm. Mogą pracować w temperaturze otoczenia od –40 do +85°C.

Pozostałe parametry:

- moc wyjściowa: 9,1 W (4 Ω, 10 V, THD+N≤1%),
- zakres dynamiczny: do 111 dB,
- szum wyjściowy: 10 µV RMS (zasilanie z pojedynczego ogniwa) i 14,5 µV RMS (zasilanie z dwóch ogniw),
- THD+N: < –83 dB @ 1 kHz,
- pobór mocy w stanie spoczynkowym: 22,5 mW (7,4 V, włączone sprzężenie IV),
- pobór mocy w trybie shutdown: < 5 µW,
- częstotliwość próbkowania: 8...96 kHz.

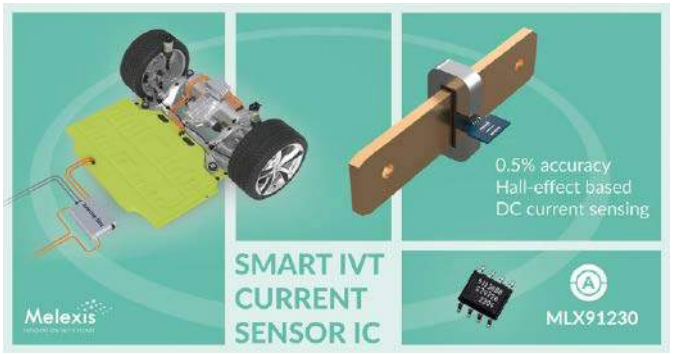
www.analog.com

Termistor NTC do pomiaru temperatury powierzchni



TDK wprowadza na rynek termistor NTC do pomiaru temperatury powierzchni, zamykany w obudowie SMD o wymiarach 10×3×3 mm. Dzięki bardzo dobremu sprzężeniu termicznemu model B57850T0103x charakteryzuje się krótkim czasem odpowiedzi. Został przystosowany do pracy w temperaturze otoczenia od –40 do +150°C. Zapewnia wytrzymałość dielektryczną do 2500 VAC @ 60 s i wodoszczelność do 500 godzin. Obecnie w ofercie dostępne są dwa modele o nominalnej rezystancji 10 kΩ: B57850T0103F000 o tolerancji 1% i B57850T0103G000 o tolerancji 2%. Zawierają elektrody z brązu fosforowego, mogące być lutowane lub spawane. Termistory B57850T0103x mogą znaleźć zastosowanie m.in. do monitorowania procesów przemysłowych oraz pomiaru temperatury radiatorów i elementów półprzewodnikowych dużej mocy.

www.tdk-electronics.tdk.com



Miniaturowy, precyzyjny czujnik prądowy Halla z kwalifikacją AEC-Q100

Melexis wprowadza na rynek pierwszy czujnik prądowy Halla 3. generacji, oznaczony symbolem MLX91230. Jest to miniaturowy układ o dużej dokładności pomiaru, zamykany w obudowie SOIC-8. Zawiera mikrokontroler, odciążający sterownik ECU od konieczności przeprowadzania operacji obliczeniowych oraz zestaw funkcji zabezpieczających.

Jednym z powtarzających się problemów w projektowaniu pojazdów elektrycznych jest to, że projektanci polegają na technice pomiaru z wykorzystaniem rezystora bocznikowego ze względu na jej prostotę, natomiast mają trudności z zapewnieniem odpowiedniego budżetu termicznego. Wraz ze zwiększaniem się natężenia prądu przepływającego przez czujnik, rezystancja powinna się zmniejszać, aby utrzymać te same straty cieplne. Z kolei wadą technologii fluxgate jest to, że wykorzystanie oferowanej przez nią izolacji galwanicznej i dużej dokładności wiąże się z wysokimi kosztami, dużym poborem mocy i sporą przestrzenią montażową. Właśnie te kwestie skłoniły firmę Melexis do opracowania modelu MLX91230.

Czujnik został zaprojektowany tak, aby mógł znaleźć zastosowanie w wielu aplikacjach. Zapewnia dużą dokładność i niezawodność,

eprasa.pl 1fc8f31a49

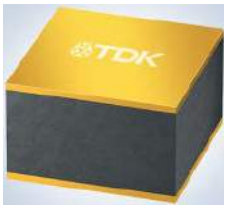
spełniając jednocześnie najwyższe wymagania odnośnie do bezpieczeństwa funkcjonalnego. Oferuje funkcje SoC (*state of charge*), SoH (*state of health*) i SoF (*state of function*) zarówno w systemach niskiego, jak i wysokiego napięcia. Nadaje się idealnie do aplikacji BMS (*battery management system*), BDU (*battery disconnect unit*) i BJB (*battery junction box*).

Dzięki cyfrowej architekturze i zaawansowanemu przetwarzaniu sygnału, MLX91230 jest w stanie zapewnić niespotykaną dotąd dokładność na poziomie 1%. W odróżnieniu od porównywalnych układów innych producentów, dokładność ta jest gwarantowana w całym zakresie temperatury pracy od –40 do +125°C i przez cały okres użytkowania z uwzględnieniem błędów nieliniowości. Mikrokontroler z wbudowaną pamięcią flash umożliwia wdrażanie niestandardowego oprogramowania i zapewnia kompensację niedoskonałości systemu pomiarowego, m.in. nasycenia ferromagnetycznego, nieliniowości i histerezy.

MLX91230 uzyskał kwalifikację AEC-Q100 i może być stosowany w systemach o poziomie nienaruszalności bezpieczeństwa do ASIL D, zgodnie z ISO 26262.

www.melexis.com

Precyzyjne termistory NTC do pomiaru temperatury diod laserowych

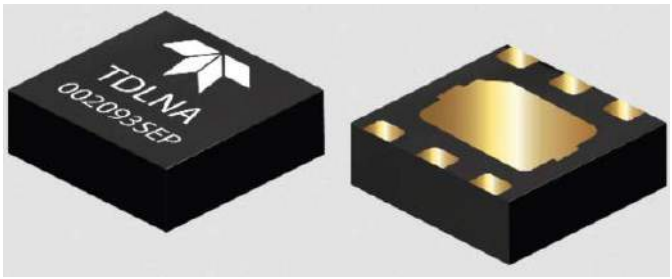


TDK wprowadza na rynek dwa precyzyjne termistory nowej serii NTCWS, przeznaczone do pomiaru temperatury diod laserowych w LiDAR-ach i transceiverach optycznych. Są one produkowane w prostopadłościennych obudowach SMD ze złotymi elementami połączeniowymi. Charakteryzują się dużą dokładnością pomiaru, wynikającą z małej tolerancji współczynnika beta i rezystancji.

	Wymiary	Rezystancja R25	B25/85
NTCWS3UF103FC1GT	0,33×0,33×0,25 mm	10 kΩ (±1%)	3930 K (±1%)
NTCWS3UF103HC1GT		10 kΩ (±3%)	

Obecnie oferta obejmuje dwa modele: NTCWS3UF103FC1GT i NTCWS3UF103HC1GT. Oba charakteryzują się rezystancją 10 kΩ i współczynnikiem beta równym 3930 K (±1%). Różnią się tolerancją rezystancji, wynoszącą odpowiednio ±1% i ±3%. Pracują w zakresie dopuszczalnej temperatury pracy otoczenia od –40 do +125°C.

www.tdk.com



Ultraniskoszumowy wzmacniacz InGaAs pHEMT 6 GHz do systemów komunikacyjnych i wojskowych

Ultraniskoszumowy wzmacniacz TDLNA002093SEP firmy Teledyne e2v HiRel Electronics został zaprojektowany do zastosowań w komunikacji satelitarnej, odbiornikach GPS oraz aplikacjach militarnych i pracujących w przestrzeni kosmicznej. Zapewnia odporność na całkowitą dawkę napromieniowania (TID) do 100 krad (SI). Oprócz bardzo dobrych właściwości szumowych, do zalet układu należy też mały pobór mocy i małe gabaryty.

TDLNA002093SEP został wyprodukowany w procesie InGaAs pHEMT. Charakteryzuje się wzmocnieniem 21 dB w zakresie częstotliwości od 1 do 6 GHz, współczynnikiem szumów poniżej 0,37 dB i mocą wyjściową 19 dB (OP1dB). Pracuje z napięciem zasilania 2,7...5,0 V, pobierając od 30 do 100 mA prądu.

TDLNA002093SEP jest zamykany w plastikowej obudowie DFN-6 o wymiarach 1,5×1,5×0,8 mm.

www.tdehirel.com



Nowe przetwornice DC/DC typu buck-boost od Texas Instruments

Firma Texas Instruments wprowadziła na rynek nowe kontrolery przetwornic DC/DC typu buck-boost. Układy TPS552892 i TPS552892-Q1 posiadają zintegrowane, poczwórne tranzystory MOSFET i mogą pracować napięciem wejściowym od 3,0 V do 36 V. Zakres napięć wyjściowych zaczyna się już na poziomie 0,8 V i kończy na 22 V, maksymalny prąd wyjściowy to 8 A. Przetwornice pracują z programowalną częstotliwością kluczowania od 200 kHz do 2,2 MHz i wspierają tryb pracy z rozproszonym spektrum w celu redukcji zakłóceń EMI. Układy TPS552892/-Q1 są dostępne w obudowach VQFN-HR o rozmiarach zaledwie 3,0×5,0 mm, wykonanych w technologii HotRod, zapewniającej doskonałe odprowadzanie ciepła.

REKLAMA



eprasa.pl 1fc8f31a49

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.



Zaawansowany generator srebra koloidalnego, sterowany przez Arduino

Generator koloidalnego srebra to urządzenie służące do produkcji zawiesiny drobnych cząstek srebra w cieczy, zazwyczaj w wodzie. Koloidalne srebro jest uważane przez niektórych za substancję mającą właściwości przeciwbakteryjne, przeciwwirusowe i przeciwgrzybicze. Podstawową zasadą działania generatora koloidalnego srebra jest elektroliza. Generator składa się z dwóch elektrod, zwykle wykonanych z czystego srebra, zanurzonych w destylowanej wodzie. Gdy przez elektrody przepływa prąd elektryczny, jony srebra uwalniane są do wody, tworząc koloidalną zawiesinę cząstek srebra.

W artykule autor opisuje, jak skonstruować zaawansowane urządzenie do wytwarzania koloidalnego srebra z zastosowaniem Arduino, które ma wszystkie funkcje takie jak typowe urządzenia komercyjne. Urządzenie to ma regulację prądu z użyciem modulacji PWM, automatyczną zmianę polaryzacji elektrod oraz timer, który wyłącza urządzenie po osiągnięciu pożądanej koncentracji jonów srebra w roztworze. W przeciwieństwie do wcześniejszego urządzenia autora, do zrobienia tego potrzebna jest już pewna wiedza z zakresu elektroniki i mikrokontrolerów.

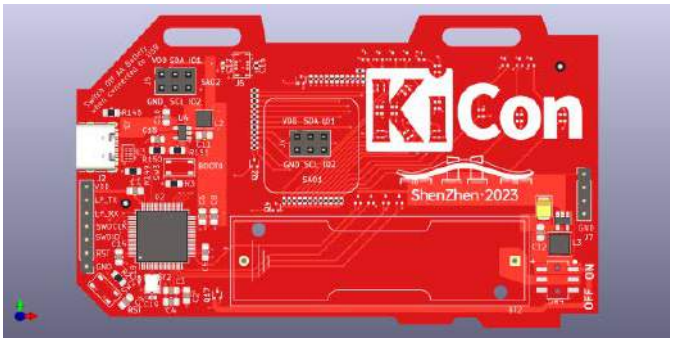
Urządzenie jest jednak stosunkowo proste w budowie i ma kilka komponentów:

- mikrokontroler Arduino Nano,
- wyświetlacz 16×2 LCD,
- układ do sterownika silnika oparty na L293D,
- stabilizator napięcia 7805,
- potencjometr,
- zasilacz 19 V, autor zastosował ładowarkę od laptopa,
- pompka powietrza – np. akwarystyczna,
- dwie srebrne elektrody.

Użytkowanie systemu zaczynamy od napełnienia szklanego pojemnika destylowaną wodą. Następnie podłączane są elektrody i urządzenie jest uruchamiane. Na wyświetlaczu LCD pojawiają się podstawowe informacje o płynącym prądzie (w miliamperach), aktualnej polaryzacji elektrod i czasie, jaki upłynął od włączenia urządzenia.

Prąd można zmieniać ciągle w zakresie od 0 do 10 mA za pomocą potencjometru umieszczonego na tyle urządzenia. Przy wyższym prądzie czas osiągnięcia określonej koncentracji jonów srebra jest krótszy, ale zaleca się stosowanie niższego prądu. Czas pomiędzy zmianami polaryzacji elektrod można ustawić w kodzie. Domyślnie ustawiony jest on na 30 minut. Maksymalny czas działania urządzenia jest również określony w kodzie i wynosi 3 godziny.

https://tiny.pl/c65nq



Plakietka na konferencję KiCon

KiCon 2023 Asia 2 odbył się w Shenzhen w Chinach 12 listopada 2023 roku. Wydarzenie może poszczycić się obszerną listą prelegentów, w tym cenionymi deweloperami KiCada, Wayne'em Stambaughiem i Sethem Hillbrandem, a także różnorodnymi prelegentami z wykładami dla słuchaczy o różnym stopniu zaawansowania w obszarze KiCada, od doświadczonych ekspertów po entuzjastycznych nowicjuszy.

Istnieje pewna tradycja – na konferencje amatorów hobbystów elektroników często wykonywane są elektroniczne plakietki, które zastępują klasyczne wejściówki i identyfikatory. Głównym mikrokontrolerem modułu jest N32L406 (ARM Cortex-M4F), który steruje kilkoma diodami RGB razem z logo KiCon. Na górze znajduje się również wyświetlacz o przekątnej 0,85 cala.

Zaprezentowany identyfikator to druga rewizja płytki. Kształt PCB odpowiada logo KiCon 2023 Asia. Centralną wyspę loga pozostawiono została złącze SAO. Znalazło się również miejsce na bardzo tani wyświetlacz z własną pamięcią, więc dodano do niego dodatkowe złącze na płytce.

Dla autora projekt ten to nie tylko ciekawe wyzwanie projektowe, ale także możliwość nauki pisania oprogramowania dla mikrokontrolerów.

https://hackaday.io/project/193278-kicon-2023-asia-badge

Moduł deweloperski dla układu CH32V003

Praca z płytkami deweloperskimi takimi jak STM32 Discovery czy Nucleo Board jest wystarczająca do szybkiego prototypowania i weryfikacji pomysłów, jednak w pewnym momencie – jak zauważa autor tej konstrukcji – warto rozważyć zaprojektowanie własnej płytki dla

eprasa.pl 1fc8f31a49



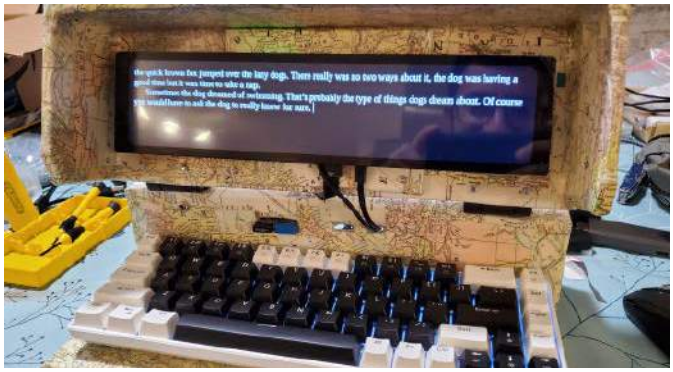
układu STM32. W zaprezentowanym projekcie prezentuje szczegóły takiej płytki wykonanej za pomocą KiCad 6, w tym: projekt obwodu, układ PCB oraz proces produkcyjny, a przy okazji daje kilka wskazówek dotyczących projektowania PCB.

Płytka deweloperska zawiera:

- dwie diody LED,
- przetwornicę DC\DC na bazie TPS5430DDA,
- przycisk użytkownika i przycisk resetu,
- dostępne wyprowadzenia GPIO, w tym piny I²C, UART i SWD do programowania i debugowania.

Jest to minimalistyczny obwód zaprojektowany za pomocą KiCada. Ma służyć jako źródło inspiracji do bardziej zaawansowanych projektów zawierających układ STM32 na własnej płytce PCB. Wszystkie pliki projektowe i produkcyjne można pobrać ze strony projektu.

https://hackaday.io/project/193238-custom-stm32-pcb-with-kicad



Cyfrowa maszyna do pisania

Autor tej konstrukcji sporo czasu spędza, pisząc różnego rodzaju treści. Pragnął mieć w pełni funkcjonalną maszynę do pisania, zdolną generować cyfrowe dokumenty, która nie rozpraszałaby uwagi. Miała być wyposażona w cichobrzmiące klawisze mechaniczne oraz cienki ekran o szerokim formacie. Zbudował więc ten nietypowy wynalazek z użyciem Raspberry Pi i oprogramowania FocusWriter oraz LibreOffice.

Ta cyfrowa maszyna do pisania miała być wykonana z litego drewna, ale drewniana obudowa nie wyszła zbyt dobrze i twórca szybko zdał sobie sprawę, że stolarka to trudne zadanie. Dlatego wszystkie nieplanowane krzywizny i przypadkowo umieszczone gwoździe ukrył, używając techniki decoupage.

Maszyna pozwala na łatwe pisanie i odczytywanie dokumentów tekstowych, a także umożliwia przenoszenie plików na komputer. Tworzenie na niej tekstu to czysta przyjemność, jak zaznacza sam autor. Urządzenie w pełni zrealizowało oczekiwania swego twórcy i zapewniło mu możliwość pracy bez zakłóceń.

https://hackaday.io/project/193218-digital-typewriter

REKLAMA

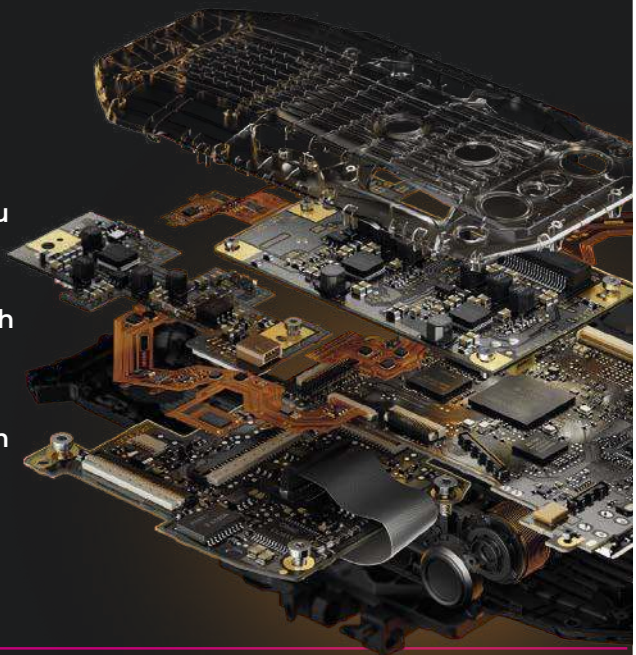
NOWOŚCI
ALTIUM DESIGNER 24

COMPUTER CONTROLS

ALTIUM DESIGNER
ALTIUM 365

- Nowe narzędzie do równoczesnej pracy wielu projektantów na jednym PCB
- Menadżer ograniczeń, pozwalający na definiowanie ograniczeń i reguł projektowych
- Narzędzie wspomagające wymianę danych z Ansys
- Prowadzenie par różnicowych pod dowolnym kątem i łukami
- Możliwość tworzenia druku bezpośrednio na trójwymiarowym podłożu

Autoryzowany dystrybutor Altium w Polsce



Computer Controls Sp. z o.o. Bielsko-Biała, ul. Bystrzańska 94 +48 33 485 94 90
info@ccontrols.pl
www.ccontrols.pl

eprasa.pl 1fc8f31a49

W ofercie AVT*
AVT6019

Podstawowe parametry:
- płytka kompatybilna z Arduino UNO R4,
- zakres napięcia zasilania to 6,5...30 V (nieco więcej niż w przypadku UNO R4),
- zasilane poprzez złącze USBC, zawiera układ idealnej diody (nie wnosi spadku napięcia),
- miejsce na dodatkową pamięć EEPROM typu 24C128,
- złącze BAT dla napięcia podtrzymania zegara RTC,
- złącza SWD, UART, CAN, I²C.

Dodatkowe materiały do pobrania ze strony www.ulubionykiosk.pl/media

AVT5850	Płytka bazowa dla Arduino Nano Every (EP 3/2021)
AVT5819	Płytka bazowa dla Arduino MKR (EP 11/2020)
AVT5777	Moduł interfejsu ethernet dla Arduino MKR Zero (EP 6/2020)
AVT5738	Wyświetlacz 7-segmentowy dla Arduino MKR Zero (EP 1/2020)
----	Sterownik silników prądu stałego dla Arduino MKR (EP 3/2019)
AVT5636	Płytka edukacyjna dla Arduino (EP 5/2018)
AVT1795	AVTduino Battery Shield (EP 3/2014)
AVT1722	AVTduino miniLCD – miniaturowy panel operatora dla Arduino (EP 1/2013)

* **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]** – płytka drukowana bez elementów i dokumentacji.

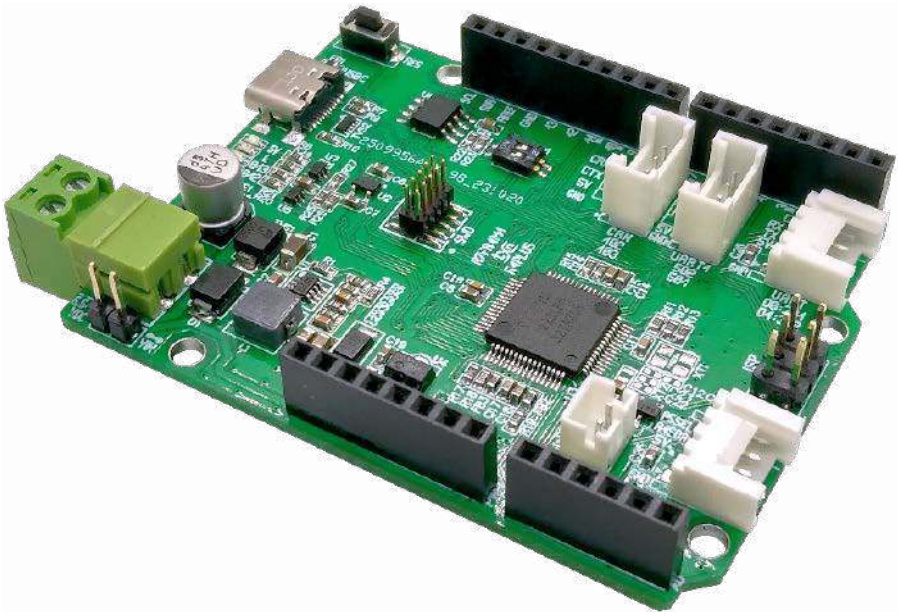
Kity, w których występuje układ scalony wymagający zaprogramowania, mają następujące dodatkowe wersje:
- **wersja [A+]** – płytka drukowana **[A] +** zaprogramowany układ **[UK]** i dokumentacja,
- **wersja [UK]** – zaprogramowany układ.

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 stronie sklepu osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: **kity@avt.pl**.

AVTDuino UNO R4 Plus

Pojawienie się na rynku Arduino UNO R4 wzbudziło spore zainteresowanie, jednak wersja bazowa płytki Minima ma kilka niedoskonałości, które zostały wyeliminowane w zaproponowanym AVTDuino UNO R4 Plus.



Identycznie jak w Arduino UNO R4, nakładka zawiera procesor R7FA4M1AB3CFM z rodziny R4M firmy Renesas, którego budowę pokazano na **rysunku 1**. Układ zawiera 32-bitowy rdzeń ARM M4 taktowany zegarem do 48 MHz, jest wyposażony w jednostkę FPU i bogate peryferia. Pozwala to na znaczące zwiększenie wydajności i możliwości aplikacyjnych w porównaniu z UNO R3.

Budowa i działanie

Schemat AVTDuino UNO R4 Plus został pokazany na **rysunku 2**. W porównaniu do wersji Minima całkowicie zmieniony został obwód zasilania. Zewnętrzne zasilanie poprzez rozłączane złącze śrubowe DCIN typu MC1.5 doprowadzone jest do przetwornicy obniżającej U1 typu MPQ4420. Układ ten wybrano ze względu na szeroki zakres napięcia zasilania, względnie prostą aplikację i przystępną cenę. Układ pracuje w fabrycznej aplikacji. Zakres napięcia zasilania, przy którym przetwornica pracuje poprawnie, to 6,5...30 V, więc nieco więcej niż w przypadku UNO R4. Dioda Schottky'ego D1 zabezpiecza układ przed odwrotnym podłączeniem zasilania, transil TVS1 przed skutkami przepięć. Kondensatory CE1, C1, C2 zapewniają filtrację zasilania.

Napięcie wejściowe ze złącza DCIN po zabezpieczeniu przez D1, TVS1 doprowadzone jest poprzez zworę VIN do złącza PWR Arduino. Zastosowanie zwory, która dla bezpieczeństwa powinna być domyślnie rozwarta, zabezpiecza nakładki, korzystające z zasilania poprzez wyprowadzenie VIN, przed uszkodzeniem w wyniku podania zbyt wysokiego napięcia VIN>12 V, do którego R4 jest przystosowane, a nakładki już niekoniecznie. Dzieje się tak pomimo nieco mylącego zapewnienia ze strony zespołu Arduino o pełnej zgodności, która, jak się okazuje, nie dotyczy napięcia zasilania.

Arduino R3 i nakładki np. Motor Shield R3 pracują poprawnie przy zasilaniu VIN

Wykaz elementów:

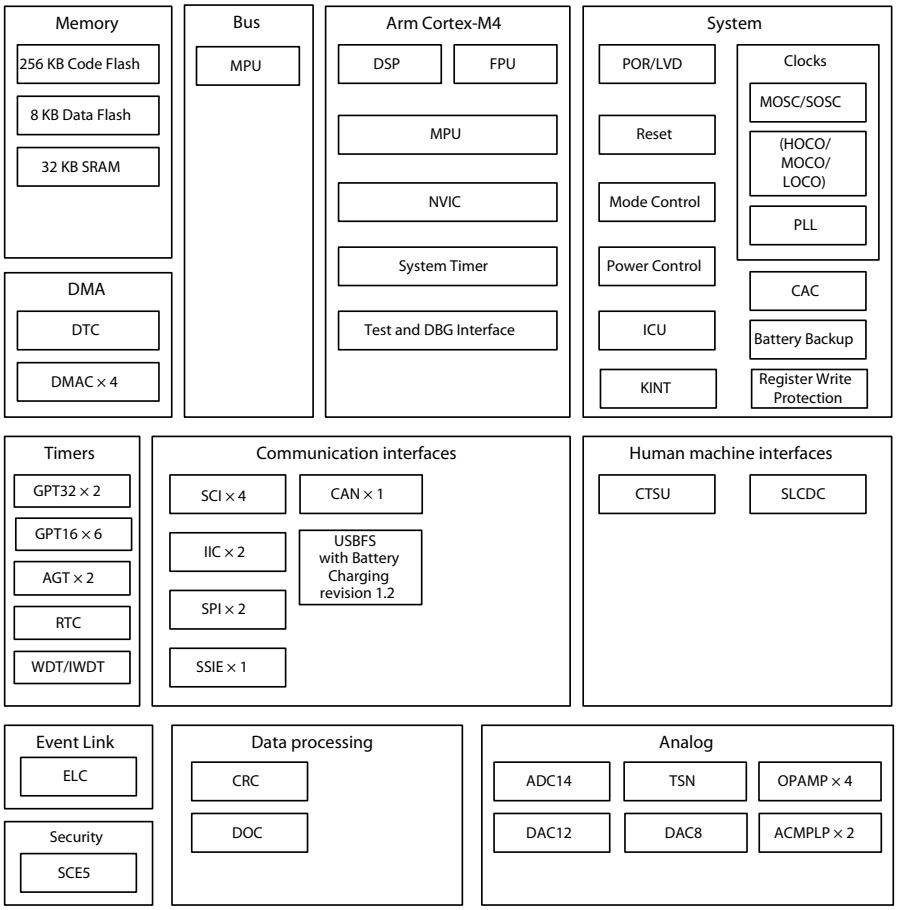
Rezystory: (SMD0603, 1%)
R1, R4, R18: 100 kΩ
R2: 41,2 kΩ
R3: 20 Ω
R5: 27 kΩ
R6: 18 kΩ
R7, R25, R26, R27: 220 kΩ
R8...R24: 5,1 kΩ

Kondensatory:
C1: 4,7 µF/50 V (SMD1206)
CE1: 47 µF/50 V (CESMD6.3)
C2, C3, C5: 0,1 µF/50 V (SMD0603)
C4A, C4B: 22 µF/10 V (SMD0805)
C6, C7, C8, C25, C26: 10 µF/10 V (SMD0603)
C9, C11, C15, C16, C17, C21, C24, : 0,1 µF/10 V (SMD0603)
C10, C14, C19, C20: 4,7 µF/10 V (SMD0603)
C12, C13, C22, C23: 12 pF COG (SMD0402) (opcja)
C18: 1 µF/10 V (SMD0603)

Półprzewodniki:
D1: dioda Schottky'ego Low Vf B250-13-F (SMB_D)
D2: dioda Schottky'ego Ultra Low Vf RB051LAM-40 (SOD128)
D3: dioda Schottky'ego podwójna wspólna katoda BAT54C (SOT-23)
LD1: dioda LED czerwona (SMD0603)
LD2, LD3: dioda LED żółta (SMD0603)
LD4: dioda LED zielona (SMD0603)
TVS1: SM6T33A (SMB_D) Transil
TVS2: ESD204 (USON10)
U1: MPQ4420GJ (TSOT23-8)
U2: LTC4411 (SOT-23-5)
U3: R7FA4M1AB3CFM (LQFP64)
U4: MCP1703T-330MB (SOT-89)
U5, U7: 74L VC2G34GW,125 (SC88)
U6: 24LC128 (SO8) (opcja)

Pozostałe:
AD: listwa męska SIP6 (Arduino)
BAT: złącze JST 2 mm 2 piny proste (opcja)
CAN, UART1: złącze Grove proste (110990030)
DCIN: złącze MC1.5 kątowe + wtyk
E2PROM: przełącznik DIP, 2 poz. Omron
FB1: dławik BLM18EG101 (SMD0603)
FB2, FB3: dławik BLM18AG601 (SMD0603)
I²C, UART: złącze Grove kątowe (110990037)
IOH: listwa męska SIP10 (Arduino)
IOL, PWR: listwa męska SIP8 (Arduino)
ISP: złącze IDC 6 (2×3 piny)
L1: 6,8 µH SRP5030T-6R8M
RES: mikroprzełącznik PB 6×3 mm
SWD: złącze szpilkowe 2×5 pinów 1,27 mm
USBC: złącze USBC USB4110GTC
VIN: listwa żeńska SIP2 + zwora
XT1: kwarc 16 MHz (3,2×2,5)
XT2: kwarc 32,768 kHz (ABS05)

eprasa.pl 1fc8f31a49



Rysunek 1. Budowa wewnętrzna procesora R7FA4M1 (za notą Renesas)

maksymalnie 12 V, podobnie nie wiadomo, jak zachowa się Ethernet Shield 2 w wersji PoE, gdy zostanie zasilony napięciem >12 V od strony wyprowadzenia VIN, nie wspominając już o zachowaniu nakładek alternatywnych zaprojektowanych poza zespołem Arduino, których zgodność z R4 należy zweryfikować we własnym zakresie.

UWAGA: proszę zwrócić szczególną uwagę na zgodność zakresu napięcia zasilania zastosowanych nakładek z napięciem zasilania Arduino R4 UNO MINIMA/Wi-Fi oraz AVTDuino UNO R4 Plus, gdyż podanie napięcia VIN większego niż 12 V, z którym każda wersja płytki R4 pracuje poprawnie, może je uszkodzić!

Dzielnik R1, R2 jest elementem zabezpieczenia podnapięciowego ULVO, z określonym progiem 4,8 V (+ spadek na D1), poniżej którego przetwornica zostanie wyłączona. Rezystory R4, R5, R6 są dzielnikiem ustalającym napięcie wyjściowe przetwornicy, które w modelu ustalone jest na ok. 5,2 V, aby częściowo skompensować spadek na kluczu zasilania (D2). Wydajność prądowa przetwornicy to 1 A, z krótkotrwałą możliwością poboru do 1,5 A.

AVTDuino UNO R4 Plus może być oczywiście zasilane poprzez złącze USBC. W wersji UNO R4 kluczowanie napięć zrealizowane jest tylko przez jedną diodę, co skutkuje sporym spadkiem napięcia 5 V, gdy zasilamy Arduino ze złącza USBC. Jest to szczególnie uciążliwe, gdy obwody zasilania we współpracującym komputerze potraktowano równie oszczędnie, wtedy napięcie 5 V potrafi obniżyć się do 4,5 V, co nie gwarantuje poprawnej pracy podłączonych rozszerzeń.

W AVTDuino UNO R4 Plus do kluczowania zasilania z portu USBC zastosowano diodę idealną U2 typu LTC4411. Zapewnia ona zabezpieczenie przeciwzwarciowe (2,6 A), zabezpiecza przed wstecznym przepływem napięcia V50P z wbudowanej przetwornicy do portu USBC oraz umożliwia współdzielenie prądu obciążenia z przetwornicą U1 zasilaną z zewnętrznego źródła. Ze względu na to, że układ U2 jest dosyć drogi, zrezygnowałem z użycia go także w obwodzie zasilania przetwornicy U1 (potencjał V50P), zastępując go diodą Schottky'ego D2 o ultraniskim spadku napięcia (katalogowo Vf=350 mV przy 1 A) typu RB051LAM i kompensując częściowo spadek napięcia podniesieniem napięcia wyjściowego przetwornicy do 5,2 V.

Dioda LD1 PWR sygnalizuje obecność zasilania 5 V. Z 5 V uzyskiwane jest dodatkowe napięcie 3,3 V o obciążalności 150 mA (chwilowo 200 mA), które doprowadzone jest do złącza PWR. Zrezygnowałem z wbudowanego w procesor stabilizatora LDO na rzecz układu U4 typu MCP1703. Rezygnacja z wbudowanego LDO zabezpiecza procesor przed uszkodzeniem w przypadku pomyłek przy uruchamianiu lub przeciążenia wbudowanego LDO, eliminując też dodatkowe nagrzewanie układu przy większych obciążeniach.

Złącze BAT, niedostępne w UNO R4 Minima, umożliwia podtrzymanie zegara RTC zewnętrznym źródłem zasilania, np. baterią 3 V, chociaż w dalszym ciągu nie rozwiązano problemu poprawnej obsługi RTC w środowisku Arduino.

Sama aplikacja procesora U3 typu R7FA4M1AB3CFM nie odbiega od wersji UNO R4. Pozostawiłem procesor w obudowie TQFP64, gdyż jest to aktualnie jedyna wersja dostępna w ilościach detalicznych. Filtry FB2, C18, FB3, C17 filtrują dodatkowo zasilanie części analogowej. Na płytce pozostawiono miejsce na zewnętrzne oscylatory procesora XT1, C12, C13 oraz zegara RTC XT2, C22, C23, które standardowo pozostają niewlutowane. Można ich użyć do własnej implementacji zmian w systemie taktowania U3.

Interfejs USBC został zabezpieczony przed skutkami przepięć przez układ TVS2 typu ESD204. Przycisk RES umożliwia manualny restart procesora U3. Wszystkie wbudowane diody sygnalizacyjne LED, tj. P012, P013, L, są odseparowane od procesora poprzez bufory U5, U7 typu LVC2G34. Pozwala to odciążyć wyprowadzenia IO oraz wyprowadzenie zasilania U3 i przeznaczyć kilka dodatkowych mA dla potrzeb użytkownika. Należy pamiętać, że procesory R7FA4M1Axx oferują tylko 8 mA obciążalności prądowej wyprowadzeń IO.

Na płytce znalazło się miejsce na opcjonalną, uzupełniającą wbudowaną w procesor pamięć DATA Flash pamięć EEPROM U6 typu 24C128 (lub dowolną inną zgodną wyprowadzeniami), której interfejs I²C może zostać odłączony przełącznikiem E2PROM.

Złącze SWD umożliwia podłączenie zgodnego programatora/debuggera. Dodatkowo na złącza Grove (standard 5 V) wyprowadzono sygnały interfejsów szeregowych UART, CAN, I²C lub odpowiadających im sygnałów cyfrowych i analogowych ułatwiające szybkie prototypowanie.

Od spodu płytki na pady wyprowadzono zasilanie 5V i masę oraz sygnały P400, P401, P402, na które można skonfigurować drugi interfejs I²C jak w UNO R4 Wi-Fi.

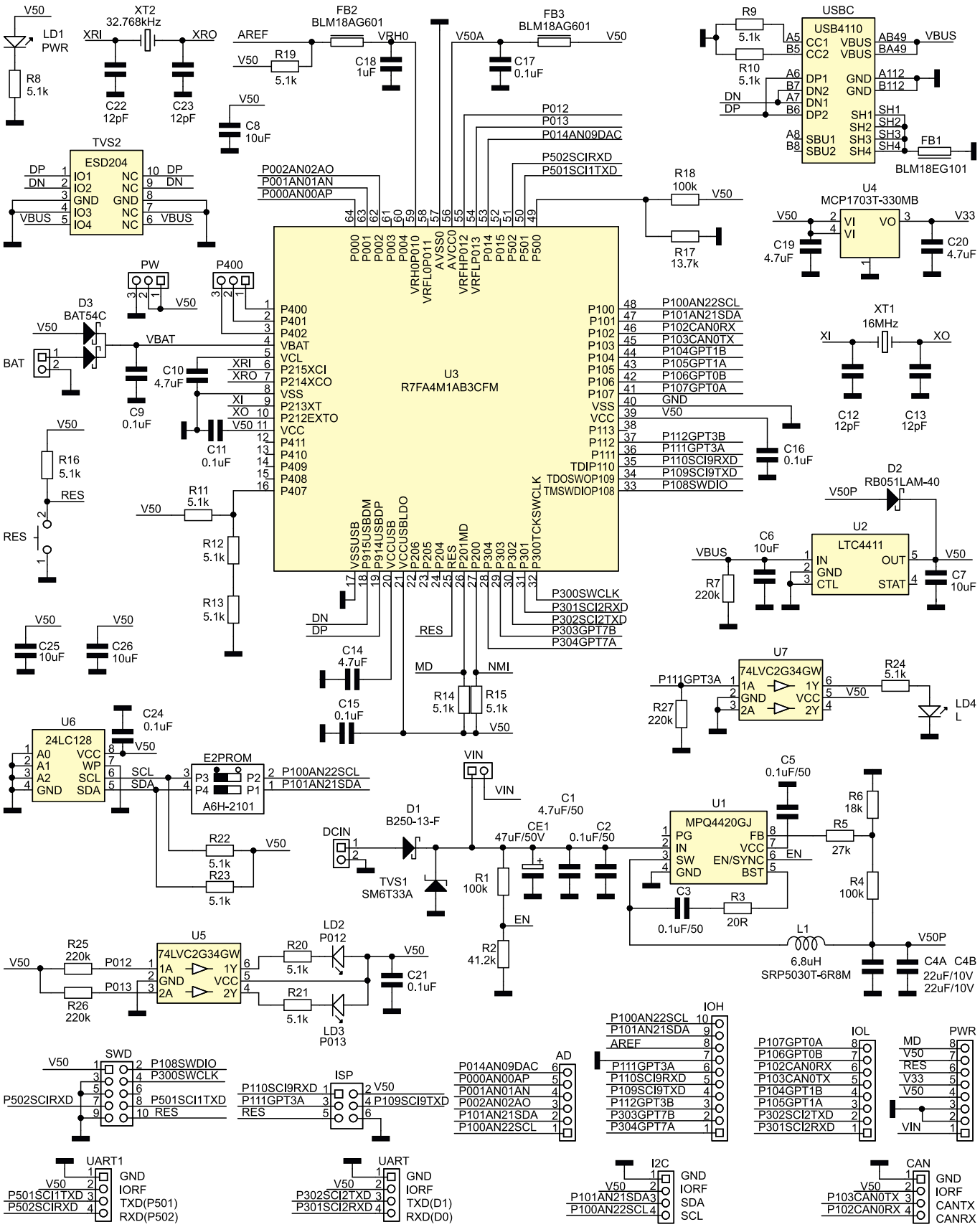
Montaż i uruchomienie

Układ zmontowany jest na dwustronnej płytce drukowanej zgodnej z Arduino (Shield) Rev3, której schemat został pokazany na **rysunku 3**. Montaż należy wykonać zgodnie z ogólnymi zasadami i nie wymaga on szczegółowego opisu. Zmontowany moduł został pokazany na **fotografii 1**.

Moduł nie wymaga uruchamiania. Po zmontowaniu przy zasilaniu z USBC

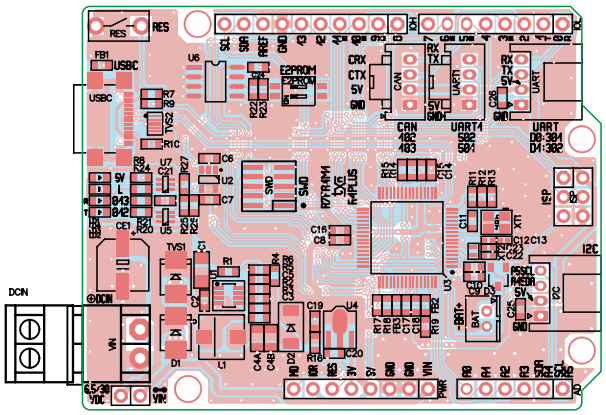
eprasa.pl 1fc8f31a49

i zasilacza zewnętrznego należy sprawdzić obecność napięć 5 V i 3,3 V. Obecność zasilania powinna być sygnalizowana świeceniem diody LD1 PWR. Jeżeli napięcia są obecne, należy zaprogramować bootloader zgodny z Arduino. W tym celu należy ze strony Renesas pobrać i zainstalować oprogramowanie RFPP Renesas_Flash_Programmer_Package_Vxx w aktualnej wersji (wymagana rejestracja). Po poprawnej instalacji do komputera z systemem Windows należy podłączyć płytkę AVTDuino UNO R4 Plus ze zwartym do masy wyprowadzeniem MD (złącze PWR), które uruchomi tryb programowania procesora. Plik *Bootloader dfu_minima.hex* dla Arduino R4 UNO znajduje się w katalogu użytkownika:

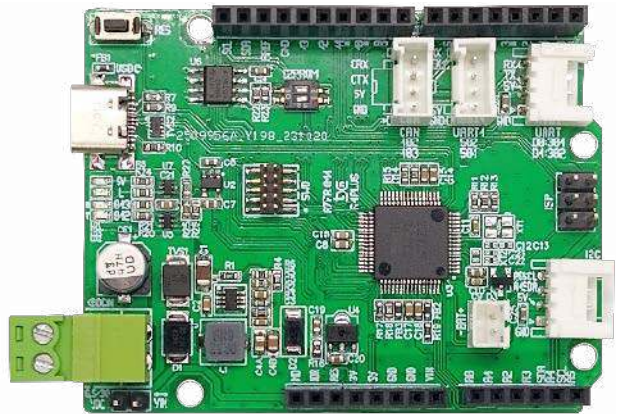


Rysunek 2. Schemat AVTDuino UNO R4 Plus

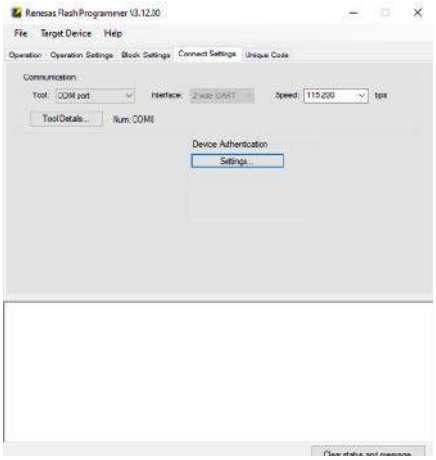
eprasa.pl 1fc8f31a49



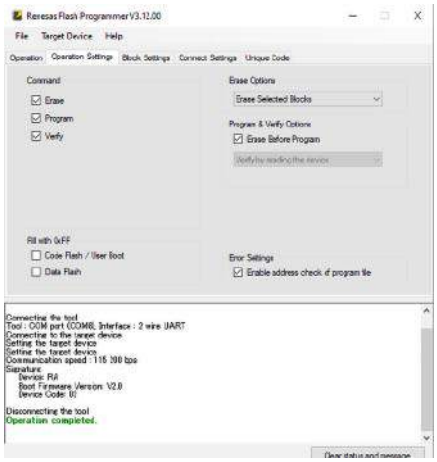
Rysunek 3. Schemat płytki PCB AVTDuino UNO R4 Plus



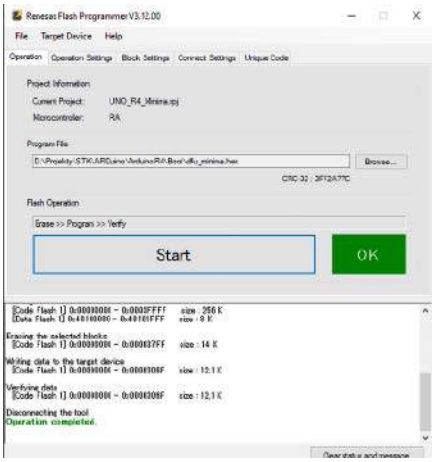
Fotografia 1. Wygląd zmontowanego modułu



Rysunek 4. Ustawienie opcji połączenia



Rysunek 5. Ustawienie opcji programowania



Rysunek 6. Programowanie bootloadera Arduino

Users\NazwaUżytkownika\AppData\Local\Arduino15\packages\arduino\hardware\renesas_uno\1.0.5\bootloaders\UNO_R4

Zgodnie z **rysunkami 4** i **5** należy ustawić opcje połączenia: *Connect Settings* oraz programowania: *Operation Settings*. Następnie w zakładce *Operation* należy wybrać aktualną ścieżkę bootloadera *dfu_minima.hex* i rozpocząć programowanie (**rysunek 6**) po przyciśnięciu przycisku RES na płytce AVTDuino UNO R4 Plus. W przypadku problemów z zaprogramowaniem należy powtórzyć całą procedurę.

Alternatywnie, jeżeli posiadamy programator J-Link, procesor można zaprogramować, używając złącza SWD. Jeżeli programowanie przebiegło poprawnie, należy płytkę odłączyć od komputera, zamknąć oprogramowanie RFPP i ponownie podłączyć AVTDuino UNO R4 Plus do komputera. W systemie Windows 10 instalacja odbędzie się automatycznie i płytka powinna pojawić się jako urządzenie portu COM.

Po uruchomieniu środowiska i ewentualnej instalacji lub aktualizacji (najnowsza wersja 1.0.5) płytek Arduino R4 UNO Boards by Arduino płytka powinna być widoczna w wyborze menedżera płytek i dostępna do programowania. W przypadku problemów z wgrywaniem aplikacji należy sprawdzić poprawność instalacji urządzenia DFU-RT Port i odpowiadającego urządzenia szeregowego USB(COMxx). W przypadku problemów należy ponownie zainstalować drivery w trybie administratora. Jeżeli wszystko działa poprawnie, można zastosować płytkę we własnych projektach.

Adam Tatuś, EP

REKLAMA





eprasa.pl 1fc8f31a49

W ofercie AVT*
AVT5990

Podstawowe parametry:
- Tor odbiorczy z bezpośrednią przemianą częstotliwości.
- Odbiór transmisji DSB lub SSB (w sprzyjających warunkach).
- Załączany za pomocą przełącznika filtr wąskopasmowy do odbioru sygnałów CW, FSK, PSK, itp.
- Pokrętła zgrubnego i precyzyjnego przestrajania.
- Wyjście kontrolne częstotliwości VFO do podłączenia częstościomierza.
- Zintegrowany wzmacniacz słuchawkowy 2×1 W/8 Ω.

Dodatkowe materiały do pobrania ze strony www.ulubionykiosk.pl/media

AVT6017	RX Ewa 40 m – odbiornik początkującego radioamatora (EP 12/2023)
AVT5900	Dwupasmowy odbiornik nasłuchowy KF RX Wiesia 80/40 m (EP 11/2021)
AVT3198	Modułowy odbiornik nasłuchowy na pasma 80 m i 40 m „Dosia” – mikroprocesorowy moduł kontrolno-sterujący (EP 2–3/2020)
AVT3230	Czteropasmowy odbiornik „Staś” – odbiornik CW/SSB początkującego nasłuchowca (80/40/30/20 m) (EdW 8/2018)
–––––	Odbiornik nasłuchowy na pasmo 80 m (EP 6/2015)
AVT2970	Odbiornik SDR na pasmo 2 m (EdW 2/2011)

* **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]** – płytka drukowana bez elementów i dokumentacji.

Kity, w których występuje układ scalony wymagający zaprogramowania, mają następujące dodatkowe wersje:
- **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 stronie sklepu osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: **kity@avt.pl**.

Odbiornik nasłuchowy na pasmo KF 80 m „Kubuś” (1)

*Odbiorniki nasłuchowe stanowią pierwszy znaczący krok do magicznego świata konstrukcji i łączności radiowych. W podanych dziedzinach pomagają one rozwijać pasje, które z czasem mogą przerodzić się w profesjonalne umiejętności techniczne i komunikacyjne (także interpersonalne). Prezentowane urządzenie powstało w odpowiedzi na znaczne zainteresowanie Czytelników jako nieco okrojona i zintegrowana na pojedynczej PCB wersja projektów z serii „*AVT-319x – Modułowy odbiornik nasłuchowy Dosia*”. W pierwszym odcinku z cyklu artykułów została przedstawiona koncepcja działania urządzenia. Projekt ten dedykuję mojemu synowi – Jakubowi.*



Radiokomunikacja amatorska w pasmie 80 m

Amatorska komunikacja radiowa w pasmie częstotliwości, w którym długość fali wynosi około 80 m, to zazwyczaj początek wspaniałej przygody z łącznościami krótkofalowymi. Przyczyną takiego stanu rzeczy jest fakt, że popularna „osiemdziesiątka” pozwala na prowadzenie lokalnych transmisji (albo choćby samych nasłuchów), w których nadawcami są polskojęzyczni operatorzy krajowi, prowadzący łączności foniczne emisją SSB (jednowstęgową). Już same nasłuchy operatorów krajowych pozwalają na stosunkowo łatwe poznawanie obyczajów komunikacyjnych, panujących w świecie radioamatorów, a także na oswojenie się z technicznymi aspektami przedmiotowych łączności.

Tabela 1. Podział pasma 80 m, obowiązujący w Polsce

Zakres częstotliwości [kHz]	Maksymalna szerokość pasma sygnału [Hz]	Rodzaje emisji oraz ich przeznaczenie
3500...3510	200	CW, pierwszeństwo dla łączności międzykontynentalnych
3510...3560	200	CW, preferencja dla zawodów
3560...3570	200	CW w trybie QRP
3570...3580	200	Wąskopasmowe emisje cyfrowe
3580...3590	500	Wąskopasmowe emisje cyfrowe
3590...3600	500	Wąskopasmowe emisje cyfrowe (stacje bezobsługowe)
3600...3620	2700	Wszystkie rodzaje emisji
3600...3650	2700	Wszystkie rodzaje emisji, preferencja dla zawodów w trybie SSB
3650...3700	2700	SSB w trybie QRP
3700...3800	2700	Wszystkie rodzaje emisji, preferencja dla zawodów w trybie SSB
3775...3800	2700	Wszystkie rodzaje emisji, preferencja dla łączności międzykontynentalnych

W tym pasmie częstotliwości radioamatorskich, w zależności od warunków propagacyjnych (atmosferycznych i jonosferycznych), w ciągu dnia zasięg łączności może wynosić do około 400 km (łączności rozumiane jako tzw. lokalne). Natomiast wieczorem i nocą maksymalny zasięg może wrosnąć nawet do kilku tysięcy kilometrów, przy czym zjawisko to występuje szczególnie zimą – z uwagi na istotnie mniejsze szumy atmosferyczne.

eprasa.pl 1fc8f31a49



Fotografia 1. Widok odbiornika w obudowie

W **tabeli 1** [1] przedstawiono podział pasma 80 m, obowiązujący w Polsce (stan aktualny na dzień 16.10.2020 r.). Jak nietrudno zauważyć, „od dołu” dostępnego pasma dominują emisje wąskopasmowe – głównie CW (Continuous Wave), czyli klasyczna telegrafia oraz wszelkiego rodzaju emisje cyfrowe). Natomiast w górnej części pasma 80 m dozwolone są tzw. emisje szerokopasmowe – głównie foniczna SSB (Single Side Band), ale także np. wolna amatorska emisja telewizyjna SSTV (Slow Scan Television).

Jak łatwo zauważyć, każdy z potencjalnych radioamatorów może więc znaleźć w pasmie 80 m coś interesującego dla siebie. Na koniec tego ustępu warto podkreślić fakt, że do wykonywania nasłuchów (tylko odbiór nadawanych sygnałów) nie jest wymagane posiadanie aktywnej licencji krótkofalarskiej, co zdecydowanie ułatwia „miękki start” w świecie radiokomunikacji amatorskiej. Dodatkowo konstruowanie urządzeń radiokomunikacyjnych, przeznaczonych na niższe pasma częstotliwości, jest potencjalnie znacząco łatwiejsze i mniej kosztowne od realizacji projektów dedykowanych wyższym pasmom HF (High Frequencies) – jak np. 40 m, 30 m, 20 m, 15 m, 10 m i jeszcze wyższym: VHF (Very High Frequencies) oraz UHF (Ultra High Frequencies).

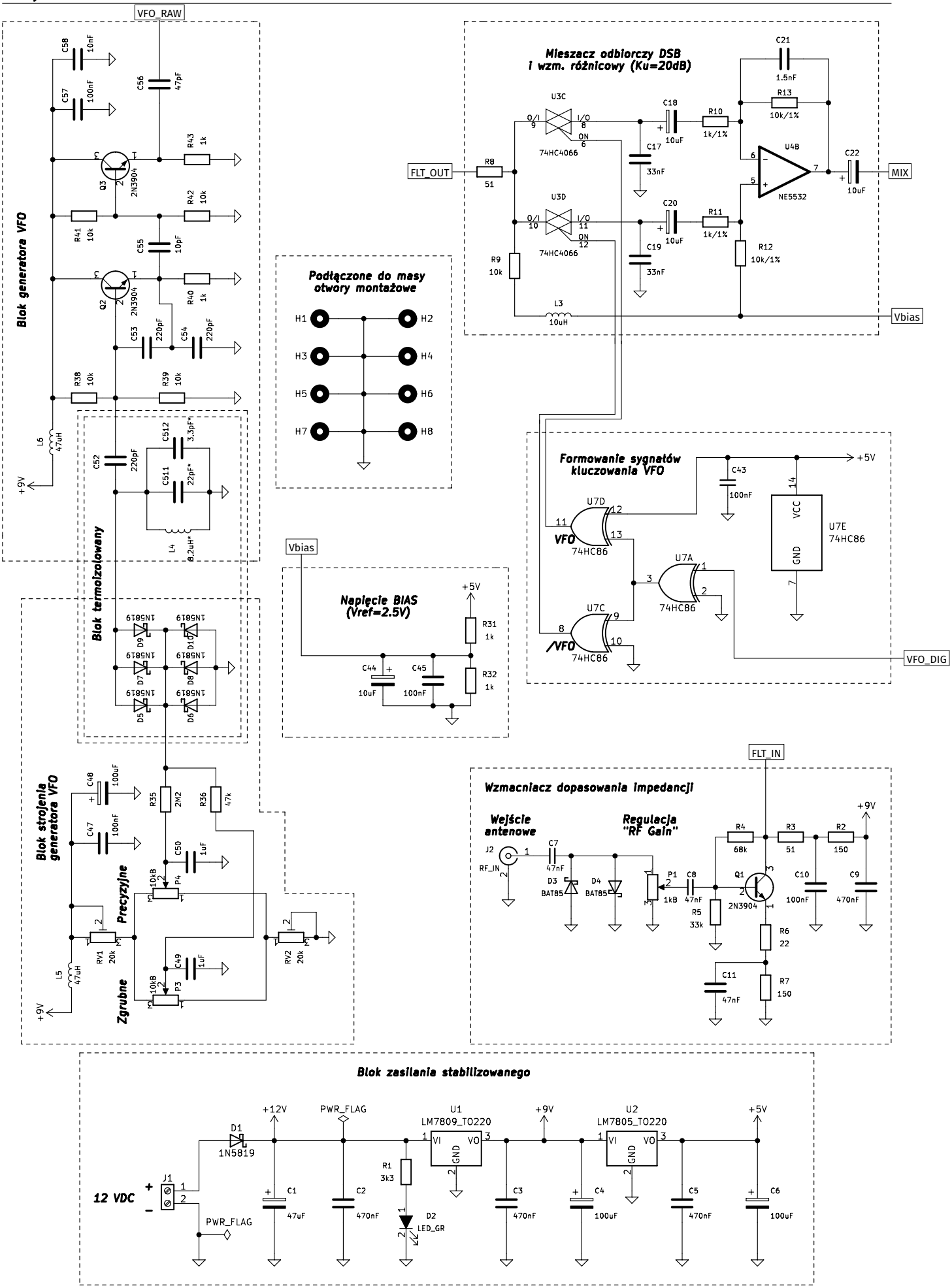
Podstawowe właściwości urządzenia

Prezentowany w tym artykule projekt dotyczy stosunkowo prostego, niemal w 100% analogowego odbiornika krótkofalowego na pasmo KF 80 m, w którym podzespoły cyfrowe stanowią jedynie niewielkie uzupełnienie – głównie w zakresie nieskomplikowanych funkcji kontrolno-sterujących. Urządzenie pracuje z tzw. bezpośrednią przemianą częstotliwości DC (Direct Conversion) i zasadniczo przeznaczone jest do odbioru fonicznej emisji dwuwstęgowej DSB (Double Sideband), jednak w sprzyjających warunkach radiowych (brak niepożądanych, zakłócających sygnałów w lustrzanej wstędze odbieranego sygnału) możliwy jest prawidłowy odbiór także jednowstęgowej emisji fonicznej SSB. Dodatkowo zastosowanie w układzie opcjonalnie załączanego filtru wąskopasmowego umożliwia także w sprzyjających warunkach odbiór emisji wąskopasmowych, jak np.: CW, FSK, PSK itp. Warto podkreślić, że w pasmie 80 m obowiązuje zasada prowadzenia łączności fonicznych SSB z wykorzystaniem dolnej wstęgi zmodulowanego sygnału LSB (Lower Sideband). Świadomość tego faktu będzie przydatna w trakcie przestrajania częstotliwości pracy odbiornika – w celu znalezienia interesującej nas emisji głosowej SSB wygodniej będzie przestrajać odbiornik od góry do dołu stosownego wycinka pasma.

Przedstawione w artykule urządzenie zasadniczo bazuje na rozwiązaniach zastosowanych w modułowym odbiorniku nasłuchowym „Dosia”, opisywanym na łamach „Elektroniki Praktycznej” w latach 2017…2019 [2], jednak dla uproszczenia konstrukcji (ukłon w stronę początkujących konstruktorów-radioamatorów) zrezygnowano z drugiego pasma KF (40 m) oraz funkcji ARW (Automatyczna Regulacja Wzmocnienia), S-metra (miernika poziomu odbieranego sygnału radiowego) a także F-metra (cyfrowego pomiaru częstotliwości odbieranego sygnału radiowego). Dodatkowo cały odbiornik umieszczono na pojedynczej, dwustronnej płytce drukowanej, dopasowanej do kompaktowej obudowy, dostępnej w handlu w umiarkowanej cenie. Opisane podejście projektowe niewątpliwie powinno ułatwić i przyspieszyć poprawny montaż oraz uruchomienie urządzenia – tym samym dając większe szanse jego potencjalnemu przyszłemu użytkownikowi na szybsze i mniej bolesne rozpoczęcie przygody z radiokomunikacją amatorską. W tym miejscu warto także podkreślić fakt, że zastosowane w projekcie rozwiązania konstrukcyjne, chociaż w XXI wieku niestanowiące „awangardy aktualnej myśli technicznej”, to jednak odbiegają dość znacząco od koncepcji stosowanych powszechnie w prostych radioodbiornikach amatorskich. Tym samym pozwalają one nie tylko krok po kroku poznać podstawy skutecznego konstruowania tego typu urządzeń, ale także, przy poprawnej realizacji projektu, wejść w posiadanie prostego urządzenia o całkiem atrakcyjnych parametrach użytkowych.

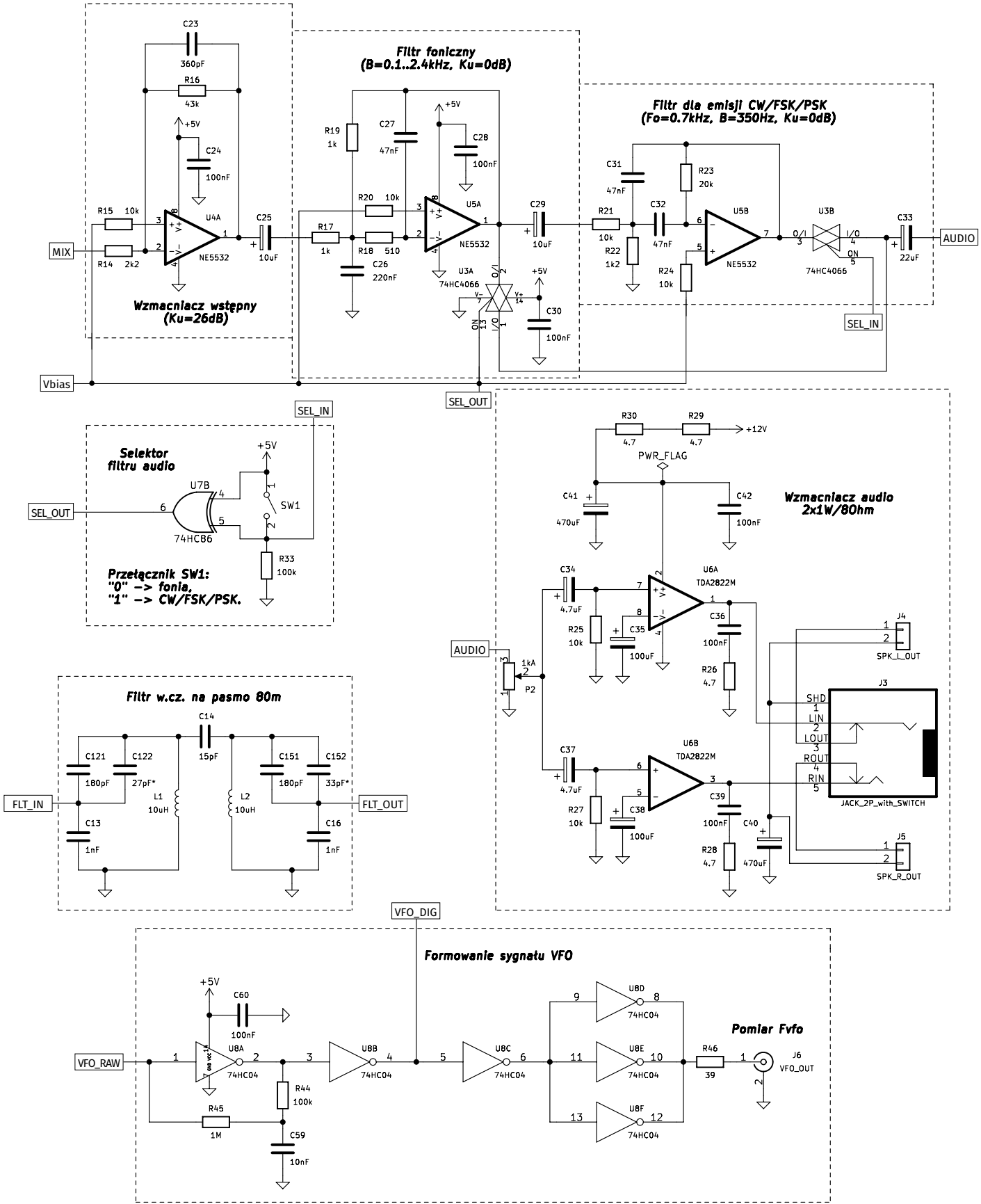
W końcowej części publikacji zostały przedstawione m.in.: metodyka oraz rezultaty podstawowych pomiarów kontrolnych, które to informacje powinny pomóc konstruktorowi urządzenia nie tylko zweryfikować jakość „produkcji” własnego odbiornika nasłuchowego, ale także wykonać pierwszy mały, lecz istotny krok w świat profesjonalnych pomiarów urządzeń radiokomunikacyjnych.

eprasa.pl 1fc8f31a49



Rysunek 1. Schemat ideowy odbiornika

eprasa.pl 1fc8f31a49



Budowa urządzenia

Szczegóły aplikacji opisanego w publikacji odbiornika nasłuchowego przedstawiono na **rysunku 1** (schemat elektryczny). Poszczególne bloki funkcjonalne układu zostały wyodrębnione przerywanymi liniami oraz oznaczone odpowiednimi etykietami.

Blok zasilania stabilizowanego stanowi swego rodzaju fundament pracy całego układu. Dostarczone na złącze J1 główne napięcie zasilania (12 V prądu stałego) wędruje dalej przez diodę Schottky'ego średniej mocy D1, której rolą jest zabezpieczenie całego urządzenia przed odwrotną (błędną) polaryzacją napięcia zasilania zewnętrznego przy możliwie jak najmniejszej stracie tegoż napięcia. Kondensatory C1 i C2 mają za zadanie filtrować główne, niestabilizowane napięcie zasilania +12 V całego układu na wejściu stabilizatora U1, przy czym to napięcie jest wykorzystywane wprost wyłącznie do zasilania

eprasa.pl 1fc8f31a49

wyjściowego wzmacniacza mocy m.cz. z układem U6. Podanie zasilania urządzenia sygnalizuje świecenie diody LED D2, której roboczy prąd pracy jest ograniczany przez rezystor R1 do wartości około 3 mA, w zupełności wystarczającej do czytelnej indykacji obecności zasilania urządzenia. Zasilany głównym napięciem +12 V stabilizator U1 wraz z pojemnościami C3 i C4 dostarcza napięcie stabilizowane +9 V. Jest ono wykorzystywane wprost przez: blok wzmacniacza wejściowego w.cz. (dopasowania impedancji), blok generatora VFO (wraz ze swoim układem strojenia) oraz przez stabilizator U2, który razem z pojemnościami C5 i C6 dostarcza w odbiorniku napięcia stabilizowanego +5 V.

Zadaniem sekcji generatora VFO (Variable Frequency Oscillator), będącego „motorem napędowym" całego odbiornika, jest dostarczanie do mieszacza sygnału o częstotliwości równej częstotliwości nośnej odbieranego sygnału. Na schemacie wyróżniono trzy bloki, należące do jego sekcji: główny blok generatora VFO, blok strojenia generatora VFO oraz blok formowania sygnału VFO. Główny blok generatora VFO pracuje w układzie w układzie Seilera, dość popularnym w klasycznych amatorskich urządzeniach radiokomunikacyjnych. Jest on zasilany napięciem stabilizowanym +9 V przez dławik L6, który razem z kondensatorami C57 i C58 separuje zasilanie generatora od zasilania reszty układu oraz chroni go przed pasożytniczymi podwzbudzeniami na wyższych niż pożądana częstotliwościach. Funkcję elementu aktywnego (generacyjnego) pełni tranzystor Q2 (popularny 2N3904) wraz z rezystorami R38…R40 (polaryzacja bazy i ujemne sprzężenie emiterowe) oraz pojemnościami C53 i C54, realizującymi pętlę sprzężenia zwrotnego. Wyjście generatora (emiter Q2) przez kondensator sprzęgający C55 o stosunkowo niewielkiej pojemności jest podawane na stopień wtórnika-bufora z tranzystorem Q3 (2N3904) i rezystorami R41…R43. O częstotliwości pracy generatora decyduje przede wszystkim równoległy obwód rezonansowy LC, dołączony do aktywnego bloku generacyjnego za pośrednictwem pojemności sprzęgającej C52. Tworzą go elementy: L4, C511 i C512, przy czym może zaistnieć potrzeba niewielkiej korekty ich wartości (względem tych podanych w artykule) dla uzyskania pożądanego, jak najszerszego zakresu przestrajania generatora (więcej informacji na ten temat podano w części publikacji dedykowanej montażowi i uruchomieniu).

Dostrajanie stałego obwodu LC do wybranej częstotliwości odbywa się za pomocą zespołu diod D5…D10 w bloku strojenia generatora VFO. W tym celu wykorzystano dwie trójki popularnych prostowniczych diod Schottky'ego małej mocy typu 1N5819. Okazały się one niezłymi zamiennikami dla coraz trudniej dostępnych w sprzedaży diod pojemnościowych (warikapów) w obudowach dostosowanych do montażu przewlekanego THT. Obie trójki równolegle połączonych diod są połączone w parach: szeregowo i przeciwsobnie dla składowej zmiennej oraz równolegle dla składowej stałej napięcia sterującego ich pojemnościami. Dla składowej stałej napięcia anody diod D6, D8 i D10 są dołączone do masy wprost, natomiast anody diod D5, D7 i D9 są dołączane do masy przez dławik L4. Natomiast dla składowej zmiennej napięcia diody D5, D7 i D9 oraz D6, D8 i D10 są połączone równolegle oraz szeregowo względem siebie (w ramach danej trójki diod). Dzięki zastosowaniu takiego rozwiązania uzyskane zostały: zwiększony zakres pojemności przestrajania Cd oraz poprawa symetrii pracy przestrajanego bloku diod pojemnościowych, skutkująca m.in. mniejszą zawartością pasożytniczych harmonicznych (głównie tych parzystych) na wyjściu generatora. Jeśli przyjąć, że pojemność każdej spośród pojedynczych diod D5…D10, zależna od jej napięcia polaryzacji wstecznej Ur, wynosi Cj(Ur), to pojemność wypadkowa całego bloku strojenia wyniesie:

$$Cd(Ur) = \frac{(3 \cdot Cj(Ur))}{2} = 1,5 \cdot Cj(Ur) \quad (1)$$

Ostatecznie uzyskiwana częstotliwość drgań zastosowanego w tym projekcie generatora Seilera będzie określona popularnym wzorem Thomsona:

$$Fvfo = \frac{1}{(2 \cdot \pi \cdot \sqrt{(L4 \cdot Cr)})} \quad (2)$$

w którym pojemność Cr jest wypadkową pojemności przestrajanej Cd(Ur) oraz wszelkich stałych pojemności, dołączonych do dławika L4. Dostrajanie zespołu diod D5…D10 do pożądanej wartości Cd(Ur) odbywa się na drodze regulacji napięcia polaryzacji wstecznej Ur, podawanego przez rezystory R35 i R36 z potencjometrów: P3 (przestrajanie zgrubne „coarse") oraz P4 (przestrajanie precyzyjne „fine"). Wartość rezystora R36 zapewnia prawidłową polaryzację wsteczną zestawu sześciu diod Schottky'ego o dość znacznych prądach wstecznych. Natomiast oporność R35, o wartości znacznie większej od R36, zapewnia możliwość podstrajania generatora VFO w niewielkim zakresie (tzw. precyzer fine). Rezystor R35 w zasadzie można dobrać wg własnych upodobań – pamiętając jednak o tym, że zgodnie z elektrotechniczną zasadą superpozycji uzyskany zakres dostrajania dokładnego będzie w przybliżeniu równy takiemu ułamkowi pełnego pasma przestrajania Bvfo, ile wynosi stosunek oporności R36/R35. W przypadku wartości zaproponowanych w projekcie będzie to 47 kΩ/2,2 MΩ, czyli około 2,14%. Przykładowo, dla całkowitego pasma przestrajania Bvfo=300 kHz, wynikowy zakres podstrajania będzie wynosił około 6,4 kHz, czyli ±3,2 kHz. Jest to o tyle korzystne, że maksymalne odstrojenie precyzerem nieco przekracza pasmo częstotliwości dla emisji fonicznej (SSB). Indukcyjność L5 separuje blok strojenia VFO od reszty zasilania +9 V w całym układzie odbiornika, natomiast pojemności C47 i C48 zabezpieczają ten blok przed pasożytniczymi zakłóceniami, wzbudzeniami i modulacjami częstotliwości generatora. Z kolei potencjometry montażowe RV1 i RV2 są przeznaczone do ustalenia zakresu przestrajania generatora VFO.

Buforowany sygnał, pochodzący z w pełni analogowej części generatora VFO, poprzez pojemność sprzęgającą C56 kierowany jest na wejście inwertera NOT (U2A) w układzie 74HC04. Bramka ta pracuje w quasi-analogowym układzie formowania przebiegu prostokątnego o poziomach logicznych CMOS 0/+5 V. W szczególności realizuje też dodatkową funkcję symetryzacji tego przebiegu – tak, aby jego współczynnik wypełnienia był jak najbardziej zbliżony do wartości D=50%. Na jej wyjściu dołączono prosty filtr dolnoprzepustowy RC z elementami R44 i C59, którego zadaniem jest ustalenie wartości średniej napięcia formowanego przebiegu prostokątnego. Wartość tego napięcia, odłożona na kondensatorze C59, jest wprost proporcjonalna właśnie do wartości współczynnika wypełnienia D i powinna być jak najbardziej zbliżona do wartości 5 V·50%=2,5 V. Rezystor R45, o wartości relatywnie dużej względem oporności R44 (zatem niezbyt mocno obciążający wyjście filtru dolnoprzepustowego z elementami R44 i C59), pracuje w pętli ujemnego sprzężenia zwrotnego i podaje wartość średnią (odfiltrowaną dolnoprzepustowo) przebiegu prostokątnego na wejście bramki U8A – jako składową stałą (offset/BIAS) dla wejściowego analogowego sygnału zmiennego z wyjścia bufora generatora Seilera. Zabieg ten ma na celu przesunięcie punktu pracy bramki U8A tak, aby uzyskać zamierzony efekt symetryzacji przebiegu prostokątnego (D=50%), albowiem bramka-inwerter NOT w środkowej części swojej charakterystyki przejściowej zachowuje się bardzo podobnie do wzmacniacza odwracającego o dość znacznym, ujemnym wzmocnieniu. Kolejne bramki NOT (U8B oraz U8C), połączone szeregowo, realizują funkcję regeneracji kształtu sygnału prostokątnego. Do wyjścia bramki U8C dołączone zostały wejścia trzech połączonych równolegle inwerterów U8D..U8F, realizujących funkcję bufora wyjściowego o łącznej wypadkowej impedancji wyjściowej równej około (45 Ω||45 Ω||45 Ω=15 Ω). Szeregowy rezystor R46 zapewnia zestandaryzowaną impedancję wyjściową modułu generatora VFO, zbliżoną do wartości 50 Ω, widzianą na zewnętrznym wyjściu J6 odbiornika. Ma on istotne znaczenie (aspekt dopasowania impedancji) przede wszystkim w przypadku, gdy sygnał prostokątny z modułu VFO będzie prowadzony

eprasa.pl 1fc8f31a49

dalej dłuższym przewodem koncentrycznym o impedancji charakterystycznej równej 505 Ω. Natomiast przy krótkim, bezpośrednim podłączeniu do wejścia kolejnego modułu (np. tak, jak to ma miejsce w odbiorniku „Dosia”) można go po prostu zmostkować – np. zastępując zworką, zrobioną z kawałka cienkiej srebrzanki. Kondensator C60 odsprzęga zasilanie omawianego bloku cyfrowego z układem U8 dla występujących w nim zakłóceń impulsowych.

Przejdziemy teraz do omówienia toru przetwarzania sygnału analogowego w odbiorniku „Kubuś”. Zadaniem tego toru jest prawidłowa zamiana szerokopasmowego sygnału radiowego, doprowadzonego do portu antenowego J2 (RF_IN) na sygnał audio (czytelny dla ucha ludzkiego) i wyprowadzenie go na porty J3…J5 (gniazdo słuchawkowe „minijack” oraz opcjonalne wyprowadzenia na dwa głośniki zewnętrzne). Pierwszym blokiem toru przetwarzania sygnału analogowego jest blok wzmacniacza dopasowania impedancji, w którym sygnał wejściowy w.cz. z gniazda antenowego, podawany na port J2 odbiornika, przez kondensator szeregowy C7 wędruje dalej na równoległy ogranicznik amplitudy z diodami D3 i D4. W tej roli zastosowano popularne, małosygnałowe diody Schottky’ego. Mają one większe rezystancje dynamiczne w zakresie przewodzenia oraz bardziej łagodne przejście z obszaru zaporowego do obszaru przewodzenia od zwykłych małosygnałowych diod krzemowych. Dzięki temu ograniczanie amplitudy nadmiernie silnych sygnałów odbywa się w sposób bardziej „miękki”, powodując tym samym mniej uciążliwych dla ucha skutków ewentualnych przesterowań. Umieszczony za ogranicznikiem z diodami D3 i D4 potencjometr P1 (RF GAIN) umożliwia ewentualne osłabienie sygnału odbieranego do poziomu właściwego dla danych warunków propagacyjnych na drodze ręcznej regulacji. Dalej odebrany sygnał, przez pojemność sprzęgającą C8, trafia na stopień wzmacniacza małosygnałowego z tranzystorem Q1, pracującego dla składowej zmiennej przetwarzanego sygnału w konfiguracji zbliżonej do wspólnego emitera – jednak z dodatkowym ujemnym sprzężeniem zwrotnym w obwodach emitera (elementy: R6, R7 i C11) a także w obwodach kolektora oraz polaryzacji bazy (elementy: R2, R3 i C10 oraz R4 i R5). Zastosowane ujemne sprzężenia zwrotne nie tylko poprawiają stabilność temperaturową punktu pracy, ale także znacznie polepszają liniowość omawianego stopnia wzmacniającego, czyniąc go bardziej odpornym na przesterowania i intermodulacje. Należy w tym miejscu podkreślić fakt, że kluczowym zadaniem wzmacniacza z tranzystorem Q1 jest zapewnienie stałego dopasowania impedancyjnego na wejściu kolejnego bloku, jakim jest filtr w.cz. W szczególności, jego zastosowanie zostało podyktowane tym, że znacznie zróżnicowane impedancje potencjalnych anten odbiorczych (w praktyce: od kilku omów dla anten bardzo krótkich, do kilkuset i więcej omów dla długich anten drutowych typu LW, czyli „Long Wire”). Taka sytuacja może powodować znaczne pogorszenie dopasowania pomiędzy anteną a zestrojonym na stałe filtrem w.cz. i – w konsekwencji – popsucie poprawnie ustalonych charakterystyk przenoszenia tegoż filtru. I tak impedancję wyjściową omawianego wzmacniacza dopasowującego zbliżoną do 50 Ω zapewnia rezystor R3, który od strony zasilania jest dla składowej zmiennej zwarty do masy kondensatorem C10, natomiast elementy R2 i C9 gwarantują bardzo dobrą separację tego bloku funkcjonalnego dla szkodliwych zakłóceń, które mogłyby przenikać do kolejnych stopni przetwarzania sygnału od strony zasilania +9 V (np. z bloku generatora VFO). Omówiony wzmacniacz dopasowania impedancji ma stosunkowo niewielkie wzmocnienie napięciowe, zbliżone do 6 dB, wynikające głównie ze stosunku wartości oporności rezystorów R3 i R6.

Sygnał ze wzmacniacza dopasowania impedancji trafia następnie do bloku filtru w.cz. na pasmo 80 m. Warto tu zwrócić uwagę nieco mniej doświadczonych konstruktorów na fakt, że to szeregowe połączenie nie zawiera dedykowanej pojemności sprzęgającej (i jednocześnie separującej dla składowej stałej), ponieważ nie jest ona potrzebna – w obwodach wejściowych filtru pasmowego w.cz. nie występuje żadne połączenie do masy czy zasilania dla składowej stałej. Sam filtr pasmowy został zrealizowany w prostej, dwubiegunowej topologii – z dwoma równoległymi obwodami rezonansowymi LC (L1, C121, C122, C13 oraz L2, C151, C152, C16), sprzężonymi niewielką pojemnością C14. Filtr jest sprzężony z poprzednim oraz z kolejnym stopniem sygnałowym za pomocą dzielonych pojemności: (C121+C122) i C13 oraz (C151+C152) i C16. Takie rozwiązanie zapewnia większą odporność charakterystyk przenoszenia filtru na wpływ impedancji sąsiednich stopni przy jednoczesnej prostocie implementacji, bowiem alternatywne sprzężenie indukcyjne zapewne okazałoby się znacznie bardziej kłopotliwe w realizacji – zwłaszcza dla nieco młodszych konstruktorów. Strojenie omawianego tu filtru w.cz. w zasadzie nie jest wymagane, jednak w przypadku istotnego rozrzutu wartości jego elementów korzystne mogłoby okazać się jego podstrajanie za pomocą korekty wartości pojemności C122 oraz C152. Aspekt ten bardziej szczegółowo omówiono w dalszej części artykułu, poświęconej montażowi i uruchomieniu urządzenia. Należy

Wykaz elementów:

Rezystory: (0,25 W/5% lub jak w opisie)
R1: 3,3 kΩ
R2, R7: 150 Ω
R3, R8: 51 Ω
R4: 68 kΩ
R5: 33 kΩ
R6: 22 Ω
R9, R15, R20, R21…R24, R25, R27, R38…R39, R41, R42: 10 kΩ
R10, R11: 1 kΩ/1%
R12, R13: 10 kΩ/1%
R14: 2,2 kΩ
R16: 43 kΩ
R17, R19, R31, R32…R40, R43: 1 kΩ
R18: 510 Ω
R22: 1,2 kΩ
R23: 20 kΩ
R26, R28, R29, R30: 4,7 Ω
R33, R44: 100 kΩ
R35: 2,2 MΩ
R36: 47 kΩ
R45: 1 MΩ
R46: 39Ω
P1: 1 kΩ B (liniowy, obrotowy, wyprowadzany na tylny panel)
P2: 1 kΩ A (logarytmiczny, obrotowy, wyprowadzany na przedni panel)
P3, P4: 10 kΩ B (liniowe, obrotowe, wyprowadzane na przedni panel)
RV1, RV2: 20 kΩ B (liniowe, montażowe, w obudowach RM-065)

Kondensatory:
C1: 47 µF/16 V
C2, C3, C5, C9: 470 nF/50 V
C4, C6, C35, C38…C48: 100 µF/16 V
C7, C8, C11, C27…C31, C32: 47 nF/50 V
C10, C24, C28, C30…C36, C39, C42, C43…C45, C47, C57, C60: 100 nF/50 V
C13, C16: 1 nF/50 V
C14: 15 pF/50 V
C17, C19: 33 nF/50 V
C18, C20, C22, C25…C29, C44: 10 µF/16 V
C21: 1,5 nF/50 V
C23: 360 pF/50 V
C26: 220 nF/50 V
C33: 22 µF/16 V
C34, C37: 4,7 µF/16 V
C40, C41: 470 µF/16 V
C49, C50: 1 µF/50 V
C52, C53, C54: 220 pF/50 V
C55: 10 pF/50 V
C56: 47 pF/50 V
C58, C59: 10 nF/50 V
C121, C151: 180 pF/50 V
C122: 27 pF*/50 V
C152: 33 pF*/50 V
C511: 22 pF*/50 V
C512: 3,3 pF*/50 V

Półprzewodniki:
D1, D5, D6, D7…D8, D9, D10: 1N5819 (obudowa: DO-41)
D2: LED (np. zielona, ∅ 3,0 mm)
D3, D4: BAT85 (obudowa: DO-35)
Q1, Q2, Q3: 2N3904 (NPN, obudowa: TO-92)
U1: LM7809 (obudowa: TO-220)
U2: LM7805 (obudowa: TO-220)
U3: 74HC4066 (obudowa: DIP-14)
U4, U5: NE5532 (obudowa: DIP-8)
U6: TDA2822M (obudowa: DIP-8)
U7: 74HC86 (obudowa: DIP-14)
U8: 74HC04 (obudowa: DIP-14)

Pozostałe:
L1, L2, L3: 10 µH (osiowy)
L4: 8,2 µH* (osiowy)
L5, L6: 47 µH (osiowy)
SW1: Włącznik elektromech. bistabilny + złącze goldpin proste męskie (2 piny)
J1: Złącze śrubowe ARK-2 (raster 5 mm)
J2, J6: Gniazda BNC 50W lutowane do PCB
J3: Gniazdo minijack stereo (∅ 3,5 mm) z przełącznikiem wewn.
J4, J5: Złącze goldpin proste męskie (2 piny)
obudowa Z112A Kradex lub inna o odpowiednich wymiarach
gniazdo zasilania, np. koncentryczne 5,5/2,1 mm, montowane na panelu
włącznik zasilania 12 V/0,5 A
podstawki pod układy scalone: DIP-8 (2 szt.) oraz DIP-14 (3 szt.)
gałki na potencjometry P1..P4 (4 szt.)
śruby, nakrętki i podkładki M2,5/M3 (wg opisu w tekście)
srebrzanka ∅ 0,7 mm ok. 20 cm

* – możliwa potrzeba korekty wartości elementu w procesie optymalizacji parametrów układu

eprasa.pl 1fc8f31a49

także zwrócić uwagę na fakt, że niewielka pojemność sprzęgająca C14 w zasadzie mogłaby być jeszcze mniejsza, oferując tym samym możliwość uzyskania filtru o nieco węższym paśmie. W projekcie odbiornika „Kubuś" zrezygnowano jednak z takiego rozwiązania, ponieważ nieco mniej doświadczonym konstruktorom mogłoby ono znacznie utrudnić skuteczne, prawidłowe zestrojenie omawianych filtrów w.cz. poprzez dalsze zwiększenie wrażliwości charakterystyk przenoszenia filtru na rozrzut wartości pozostałych jego elementów.

Odebrany szerokopasmowy sygnał radiowy w.cz. (RF), odfiltrowany wstępnie dla całego pasma 80 m, kierowany jest dalej do bloku mieszacza odbiorczego DSB, powiązanego ze wzmacniaczem różnicowym o wzmocnieniu Ku równym około 20 dB. Sercem układu jest układ scalony U3 (74HC4066), który zawiera w sobie cztery przełączane poziomami logicznymi CMOS (0/+5 V) analogowe klucze półprzewodnikowe. Sygnał radiowy jest wprowadzany przez rezystor dopasowujący R8 – jednocześnie na dwa z czterech kluczy w układzie U3: U3C oraz U3D. Ich rezystancja szeregowa (przy zasilaniu układu U3 napięciem +5 V) jest zbliżona do 50 W i, wraz z rezystorem R8 oraz pojemnościami C17 i C19, stanowi filtr dolnoprzepustowy RC, który na symetrycznych wyjściach mieszacza odbiorczego z kluczami U3C i U3D odcina wysokoczęstotliwościowe produkty przemiany (mieszania). Do poprawnej pracy z analogowymi sygnałami zmiennymi klucze w układzie U3 (74HC4066) wymagają wstępnej polaryzacji napięciem stałym BIAS o wartości jak najbardziej zbliżonej do połowy ich napięcia zasilania (czyli w tym przypadku +2,5 V), co zapewnia minimalne zniekształcenia nieliniowe przenoszonych sygnałów. Takie napięcie jest podawane na oba klucze (U3C i U3D) poprzez połączone szeregowo elementy L3 i R9, zapewniające dużą impedancją dla sygnałów w.cz. (nie tłumią sygnału wejściowego w.cz., który ma zostać poddany mieszaniu odbiorczemu). Poziom stałego napięcia polaryzującego BIAS (Vref=2,5 V), wykorzystywany także do polaryzacji wejść wzmacniaczy operacyjnych w układach scalonych U4 i U5, jest ustalany przez zasilany napięciem stabilizowanym +5 V dzielnik rezystancyjny z opornikami R31 i R32 oraz kondensatorami filtrującymi C44 i C45. Z kolei klucze U3C i U3D są naprzemiennie załączane sygnałami logicznymi w przeciwnych fazach (0 i 180°), które to sygnały są wytwarzane przez bramki A, C i D układu U7 (74HC86), zawierającego łącznie cztery dwuwejściowe bramki logiczne XOR. Podstawowy cyfrowy sygnał przełączający w.cz. z generatora przestrajanego VFO (o częstotliwość Fvfo równej częstotliwości nośnej Fc odbieranego sygnału w.cz. o współczynniku wypełnienia maksymalnie zbliżonym do wartości D=50% (przebieg symetryczny) gdzie jest wprowadzany na wejście nr 1 bramki U7A. W tym miejscu należy podkreślić, że niesymetryczny przebieg przełączający klucze U3C i U3D (D≠50%) spowodowałby niezrównoważenie mieszacza odbiorczego, skutkujące generowaniem przez mieszacz nieprawidłowego sygnału m.cz. W dyskutowanym tutaj układzie formowania sygnałów kluczowania VFO bramka XOR U7A jest skonfigurowana jako prosty bufor (jedno z wejść jest na stałe podłączono do masy zasilania) dla cyfrowego sygnału, podawanego z generatora VFO, i jej zadaniem jest wyłącznie lokalna regeneracja kształtu tego sygnału. Z wyjścia tej bramki sygnał przełączający mieszacz jest kierowany na wejścia dwóch kolejnych bramek XOR (U7C i U7D), z których pierwsza jest skonfigurowana także jako bufor, a druga jako inwerter NOT sygnału (jedno z wejść podłączone na stałe do +5 V zasilania). Na wyjściach tych bramek mamy już do dyspozycji dwa sygnały przełączające dla mieszacza – ale w przeciwnych fazach. W tym miejscu należy wyjaśnić to, dlaczego w roli układu generującego sygnały w przeciwnych fazach zastosowano dwie bramki XOR, należące do tego samego układu scalonego – zamiast np. pojedynczej bramki NOT czy NAND. Otóż sygnały w przeciwnych fazach, przełączające klucze U3C i U3D w mieszaczu odbiornika, powinny mieć przesunięcie fazowe jak najbardziej zbliżone do 180°, co, jak wspomniano powyżej, jest wymogiem dla prawidłowej pracy mieszacza. Tymczasem zastosowanie w celu uzyskania sygnału w przeciwnej fazie wyłącznie pojedynczej bramki odwracającej (czyli tylko w jednej gałęzi sygnału) sprawiłoby, że jej niezerowe opóźnienie propagacji zniweczyłoby ten cel. Oczywiście, uzyskane tą drogą rezultaty byłyby tym gorsze, im większa byłaby częstotliwość przełączania kluczy oraz opóźnienie propagacji zastosowanego negatora (inwertera NOT). Przykładowo, przy częstotliwości Fvfo=Fc=10 MHz (okres sygnału równy 100 ns) różnica opóźnień propagacji między obiema gałęziami sygnałów równa 10 ns spowodowałaby różnicę faz równą aż 10 ns/100 ns·360°=36° kątowych!!! Tymczasem obie bramki XOR (U7C i U7D), skonfigurowane jako bufor oraz jako inwerter (negator), mają bardzo zbliżone (niemal identyczne) opóźnienia propagacji, co gwarantuje uzyskanie bardzo dobrej przeciwfazowości sygnałów, które przełączają klucze U3C i U3D. Wracając do układu przemiany, odfiltrowane dolnoprzepustowo sygnały z obu kluczy mieszających są podawane przez kondensatory C18 i C20 na wejście dokładnego wzmacniacza różnicowego. Wykorzystano w nim jeden z dwóch niskoszumowych wzmacniaczy operacyjnych (konkretnie U4B), dostępnych w układzie NE5532. Rezystory R10…R13 odpowiadają za wzmocnienie tego stopnia, ustalone na poziomie +20 dB. Powinny one mieć tolerancję nie gorszą niż 1% – z uwagi na konieczność zrównoważenia mieszacza, warunkowaną także przez symetryczność omawianego tutaj wzmacniacza różnicowego. Kondensator C21 wstępnie ogranicza pasmo przenoszenia tego pierwszego stopnia m.cz., natomiast przez rezystor R12 podawane jest dodatkowo napięcie ustalające punkt pracy (BIAS) wzmacniacza operacyjnego.

Zdemodulowany a także wzmocniony i odfiltrowany dolnoprzepustowo (po raz pierwszy) sygnał m.cz. jest podawany przez kondensator C22 do wzmacniacza wstępnego o wzmocnieniu równym około +26 dB, pracującego w prostej topologii wzmacniacza odwracającego. Wykorzystuje on kolejny niskoszumowy wzmacniacz operacyjny U4A oraz rezystory R14…R16 i kondensator C23, które ustalają jego wzmocnienie i pasmo przenoszenia. Natomiast kondensator C24 zwiera do masy zakłócenia w zasilaniu +5 V obu wzmacniaczy operacyjnych: U4A i U4B.

Za wzmacniaczem wstępnym odebrany sygnał audio (o częściowo już ograniczonym pasmie) jest podawany przez pojemność C25 na wejście pierwszego z dwóch właściwych filtrów akustycznych. Jest to filtr foniczny o pasmie przenoszenia dostosowanym do odbioru emisji SSB (B=0,1…2,4 kHz) i wzmocnieniu jednostkowym (Ku=0 dB). Pracuje on w dolnoprzepustowej topologii tzw. filtru Sallena-Keya (jest to tzw. filtr kratowy). Wykorzystano w nim kolejny, niskoszumowy wzmacniacz operacyjny U5A – także oparty na popularnej wciąż kostce NE5532. Jego parametry ustalają rezystory R17…R20 oraz pojemności C26 i C27, natomiast kondensator C28 filtruje zasilanie +5 V, wspólne dla obu wzmacniaczy (U5A i U5B) w zastosowanym tutaj układzie scalonym. Wyjście filtru fonicznego zostało rozgałęzione i trafia zarówno wprost na wejście kolejnego filtru, przeznaczonego dla emisji wąskopasmowych (poprzez pojemność sprzęgającą C29), jak i do klucza analogowego przełączanego cyfrowo U3A, przez który rozpoczyna się droga obejściowa (tzw. „bypass"), pozwalająca obejść filtr wąskopasmowy. Kondensator C30 zwiera do masy zakłócenia w zasilaniu +5 V wszystkich czterech kluczy U3A…U3D, które znajdują się wewnątrz wspólnego układu scalonego.

Kolejny filtr, przeznaczony dla emisji wąskopasmowych jak CW, FSK czy PSK, jest filtrem środkowoprzepustowym o jednostkowym wzmocnieniu (Ku=0 dB), środku pasma przenoszenia w okolicach Fo=700 Hz i jego szerokości równej około B=350 Hz. Został on zrealizowany w środkowoprzepustowej topologii tzw. filtru kratowego (Sallena-Keya). Elementem aktywnym (wzmacniającym) jest w nim kolejny niskoszumowy wzmacniacz operacyjny U5B,

eprasa.pl 1fc8f31a49

natomiast jego parametry robocze ustalają rezystory R21…R24 oraz pojemności C31 i C32. Odfiltrowany wąskopasmowo sygnał trafia na klucz analogowo-cyfrowy U3B.

W tym miejscu widać już jasno to, że para kluczy U3A i U3B realizuje swoisty przełącznik-selektor sygnału audio: fonicznego albo wąskopasmowego, przy czym sygnał wąskopasmowy jest i tak zawsze wstępnie filtrowany przez filtr foniczny, co tylko poprawia jego praktyczne walory użytkowe. Należy też tutaj wyjaśnić fakt, że stałonapięciowa polaryzacja obu wymienionych kluczy odbywa się wprost z wyjść wzmacniaczy operacyjnych U5A i U5B, które dla składowych stałych swoich napięć wejściowych (czyli napięć polaryzujących BIAS o Vref=2,5 V) są po prostu wtórnikami. Naprzemienne (alternatywne) załączanie kluczy U3A i U3B odbywa się za pomocą sterującego sygnału logicznego, pobieranego z bloku selektora filtru audio. Został on zrealizowany w oparciu na bramce XOR U7B, która została skonfigurowana jako inwerter (negator) poprzez podłączenie jednego z jej wejść na stałe do szyny zasilania +5 V. Dodatkowo dedykowany przełącznik elektromechaniczny SW1 wraz z rezystorem R33 zapewniają sterujące sygnały logiczne: wprost dla klucza U3B oraz odwrócony dla klucza U3A, przy czym przełącznik SW1 otwarty wybiera filtrację dla emisji fonicznych, natomiast SW1 zamknięty wybiera dodatkową filtrację wąskopasmową.

Odfiltrowany (zgodnie z wyborem nastawionym przez operatora odbiornika przełącznikiem SW1) sygnał audio poprzez pojemność C33 trafia do bloku wyjściowego wzmacniacza mocy audio – na „gorącą” końcówkę potencjometru P2. Potencjometr P2 jest potencjometrem logarytmicznym, co zapewnia odczucia słuchowe zmian głośności wprost proporcjonalne do zmian nastawień potencjometru. Wzmacniacz mocy m.cz. został oparty o popularny i niedrogi układ scalony TDA2822M (U6), który przy zasilaniu +12 V jest w stanie oddać nawet do 1 W na kanał (jest to wzmacniacz stereofoniczny) na obciążeniu 8 W przy częstotliwości f=1 kHz i całkowitych zniekształceniach nieliniowych d=10%. Większe moce wyjściowe są co prawda możliwe do uzyskania w konfiguracji mostkowej (monofonicznej) „bridge”, jednak w tym zastosowaniu nie jest to celowe – głównie z uwagi na możliwe zbyt wysokie straty termiczne i zniekształcenia nieliniowe. Wzmacniacz zaimplementowano zgodnie z notą aplikacyjną producenta – z tą jednak różnicą, że oba wejścia stereo połączono ze sobą – tak, by na obu wyjściach uzyskać takie same sygnały o jak najbardziej zbliżonych poziomach. Dla uzyskania większej mocy wyjściowej cały blok wzmacniacza audio zasilany jest bezpośrednio głównym napięciem +12 V, dostarczanym wprost do odbiornika (bez dodatkowej wewnętrznej stabilizacji). Para rezystorów R29 i R30, połączonych szeregowo dla uzyskania ich większej łącznej maksymalnej mocy strat termicznych, ma dwa zadania. Głównym z nich jest ograniczenie mocy strat termicznych, wydzielanych we wzmacniaczach U6A i U6B w przypadku zbyt silnego ich wysterowania. Może także w pewnym stopniu separować ich zasilanie od potencjalnych zakłóceń w zewnętrznym zasilaniu +12 V, np. przydźwięku sieciowego czy tętnień pochodzących od przetwornicy impulsowej. Kondensatory C41 i C42 współtworzą z rezystorami R29 i R30 filtr dolnoprzepustowy RC, blokujący ewentualne zakłócenia w napięciu zasilania wzmacniaczy U6A i U6B. Przede wszystkim jednak znacząco zwiększają chwilową wydajność prądową źródła napięcia zasilania wzmacniaczy końcowych audio w momentach szczytów poboru mocy, czym istotnie przyczyniają się do zmniejszenia zniekształceń nieliniowych w odsłuchiwanym dźwięku. Sygnał wyjściowy audio został wyprowadzony na gniazdo J3 typu minijack stereo (∅ 3,5 mm), które zawiera w sobie przełącznik sygnałów. Po odłączeniu słuchawek stereo sygnał audio będzie dostępny na portach J4 i J5, skąd można go podać na jeden lub dwa głośniki zewnętrzne.

Na schemacie odbiornika pokazano także podłączenie do masy urządzenia otworów montażowych H1…H8. Informacja ta, z pozoru trywialna, ma za zadanie podkreślić potencjalnemu konstruktorowi fakt, że otwory montażowe w płytce drukowanej odbiornika można wykorzystać bezpośrednio do połączenia masy urządzenia z metalową obudową, ale i to, że nie są one elektrycznie obojętne i w związku z tym w trakcie montażu i uruchomienia należy unikać zwierania ich z innymi obwodami elektrycznymi.

Podsumowanie

W tej części publikacji przedstawiono koncepcję działania odbiornika oraz szczegóły jego działania. Przeprowadzono także wstęp do łączności radiowych w pasmie 80 m. W kolejnych artykułach zostanie przedstawiony montaż i uruchomienie urządzenia a także pomiary kontrolne, które podpowiedzą, jak można samodzielnie konstruować urządzenia radiokomunikacyjne.

Adam Sobczyk SQ5RWQ
sq5rwq@gmail.com

Źródła:
[1] https://tiny.pl/csvcp
[2] https://tiny.pl/csvcl

REKLAMA



eprasa.pl 1fc8f31a49

W ofercie AVT*
AVT6016

Podstawowe parametry:
- liczba przegródek: 16,
- maksymalna liczba dziennych dawek: 5,
- źródło napięcia zasilającego: bateria alkaliczna typu AAA,
- prąd pobierany ze źródła zasilania (maksymalny/tryb uśpienia): 55 mA/80 µA (szczegóły w tekście artykułu).

Dodatkowe materiały do pobrania ze strony www.ulubionykiosk.pl/media
AVT5541 Dozownik detergentu, czyli czas na elektronizację WC (EP 6/2016)

* **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]** – płytka drukowana bez elementów i dokumentacji.

Kity, w których występuje układ scalony wymagający zaprogramowania, mają następujące dodatkowe wersje:
- **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 stronie sklepu osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: **kity@avt.pl**.

Autor składa podziękowania firmie Artronic za dostarczenie wyświetlacza LCD zastosowanego w projekcie urządzenia e-Nurse.

e-Nurse
– elektroniczny dyspenser leków (2)



W poprzednim numerze EP zaprezentowaliśmy konstrukcję układu elektronicznego oraz szczegóły dotyczące sterowania i zasilania adresowalnych diod LED, stanowiących prosty i intuicyjny interfejs użytkownika dyspensera e-Nurse. Tym razem zajmiemy się architekturą programu sterującego oraz opiszemy sposób montażu i obsługi urządzenia. Dla Czytelników zainteresowanych odtworzeniem urządzenia dla siebie lub bliskiej osoby udostępniamy także szczegóły dotyczące drukowalnego panelu przegródek, opracowanego przez autora na potrzeby projektu.

Program sterujący

Zaczniemy od pliku nagłówkowego do obsługi diod LED RGB upraszczającego i porządkującego późniejszy kod, który to plik pokazano na **listingu 1**.

Dalej, na **listingu 2**, pokazano ciało funkcji odpowiedzialnej za przesłanie jednego bitu danych przy udziale wspomnianego wcześniej asynchronicznego interfejsu komunikacyjnego. Warto zauważyć, że stosowną funkcję zdefiniowano, używając atrybutu *__always_inline__*, który instruuje kompilator, by ZAWSZE wstawiał rozwinięcie funkcji w miejscu wywołania, nie zaś skok do jej ciała. Jest to o tyle istotne, że operujemy na dość rygorystycznych timingach rzędu setek nanosekund, gdzie, przy częstotliwości taktowania mikrokontrolera rzędu 8 MHz (okres 125 ns), niepotrzebne (z punktu widzenia programisty, nie kompilatora) wykonanie jakiegoś rozkazu asemblera może rozłożyć całą transmisję danych, uniemożliwiając jakiekolwiek sterowanie wspomnianymi elementami.

Skoro mamy już funkcję umożliwiającą przesłanie jednego bitu informacji, to pora na funkcję (także odpowiednio zadeklarowaną) umożliwiającą przesłanie kompletnego bajta danych, przy udziale wspomnianego wcześniej interfejsu danych, której ciało pokazano na **listingu 3**.

Listing 1. Plik nagłówkowy modułu obsługi diod WS2812B-V5

```
#define LED_PORT PORTA
#define LED_DDR DDRA
#define LED_PIN_NR PA3
#define LED_SET LED_PORT |= (1<<LED_PIN_NR)
#define LED_RESET LED_PORT &= ~(1<<LED_PIN_NR)
#define LED_PIN_AS_OUTPUT LED_DDR |= (1<<LED_PIN_NR)
```

Dalej, by uprościć zapis danych do adresowalnych diod LED RGB, przewidziano funkcję pozwalającą na przesłanie kompletnej informacji o kolorze tejże diody LED, której ciało pokazano na **listingu 4**.

Zapewne pamiętacie, że asynchroniczny interfejs komunikacyjny diod LED typu WS2812B-V5 przewiduje transmisję jeszcze jednego sygnału, a mianowicie sygnału RESET. Ciało funkcji odpowiedzialnej za transmisję sygnału RESET interfejsu diod WS2812B-V5 pokazano na **listingu 5**.

Normalnie w tym miejscu zakończyłbym opis obsługi naszych ciekawych diod LED RGB, ale nie tym razem. Prawdę mówiąc, funkcje pokazane powyżej będą działać bez problemów wyłącznie na szybszych mikroprocesorach, dla których jeden cykl zegarowy jest najlepiej o rząd

Listing 2. Funkcja odpowiedzialna za przesłanie jednego bitu danych interfejsu diod WS2812B-V5

```
__inline__ void __attribute__((__always_inline__))
    ledSendBit(uint8_t Bit) {
    if(Bit) {
        LED_SET;
        _delay_us(0.65);
        LED_RESET;
        _delay_us(0.65);
    } else {
        LED_SET;
        _delay_us(0.25);
        LED_RESET;
        _delay_us(0.65);
    }
}
```


eprasa.pl 1fc8f31a49

wielkości krótszy niż timingi odpowiedzialne za wysłanie poszczególnych bitów koloru diod LED. W innym przypadku, czyli jak u nas, gdzie takt zegara trwa 125 ns (dla częstotliwości taktowania 8 MHz),

Listing 3. Funkcja odpowiedzialna za przesłanie jednego bajta danych interfejsu diod WS2812B-V5

```
__inline__ void __attribute__((__always_inline__))
    ledSendByte(uint8_t Byte){
    ledSendBit(Byte & 0x80);
    ledSendBit(Byte & 0x40);
    ledSendBit(Byte & 0x20);
    ledSendBit(Byte & 0x10);
    ledSendBit(Byte & 0x08);
    ledSendBit(Byte & 0x04);
    ledSendBit(Byte & 0x02);
    ledSendBit(Byte & 0x01);
}
```

Listing 4. Funkcja pozwalającą na przesłanie kompletnej informacji o kolorze diody LED typu WS2812B-V5

```
__inline__ void __attribute__((__always_inline__))
    ledSendColor(uint8_t R, uint8_t G, uint8_t B){
    ledSendByte(G);
    ledSendByte(R);
    ledSendByte(B);
}
```

Listing 5. Funkcja odpowiedzialna za transmisję sygnału RESET interfejsu diod WS2812B-V5

```
inline void ledReset(void){
    LED_RESET;
    _delay_us(300);
}
```

Listing 6. Uproszczone funkcje umożliwiające poprawną transmisję sygnałów sterujących interfejsu diod WS2812B-V5 dla mikrokontrolera taktowanego zegarem 8 MHz

```
#define NOP() __asm__ __volatile__ („nop”)

__inline__ void __attribute__((__always_inline__))
ledSendBit0(void) { //Dla zegara 8MHz (NOP = 125ns)
    LED_SET;
    NOP();
    LED_RESET;
    NOP();
    NOP();
    NOP();
    NOP();
    NOP();
}

__inline__ void __attribute__((__always_inline__))
ledSendBit1(void) { //Dla zegara 8MHz (NOP = 125ns)
    LED_SET;
    NOP();
    NOP();
    NOP();
    NOP();
    NOP();
    LED_RESET;
    NOP();
    NOP();
    NOP();
    NOP();
    NOP();
}

__inline__ void __attribute__((__always_inline__))
ledRed(void) {
    ledSendBit0();
    ledSendBit0();
    ledSendBit0();
    ledSendBit0();
    ledSendBit0();
    ledSendBit0();
    ledSendBit0();
    ledSendBit0();

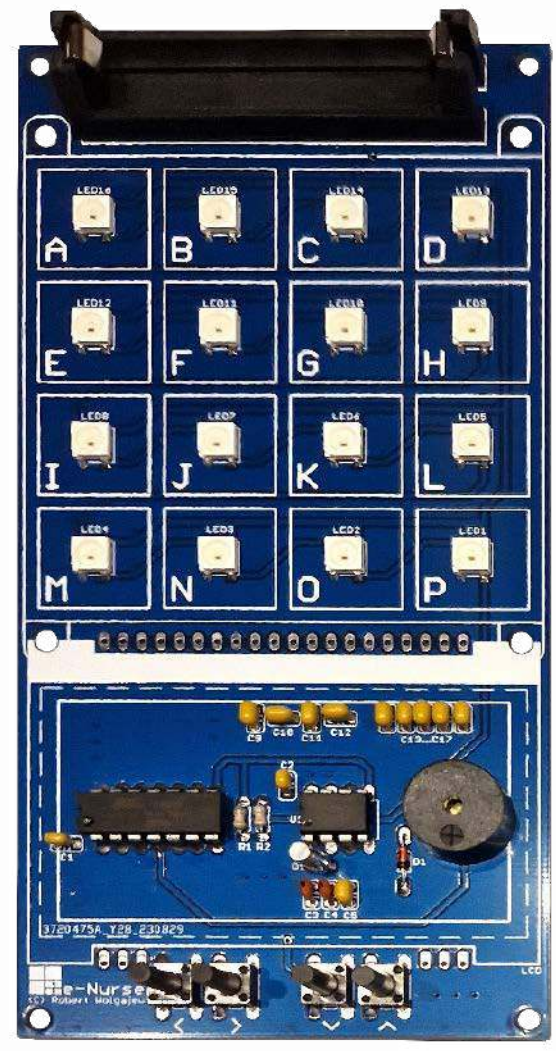
    ledSendBit1();
    ledSendBit0();
    ledSendBit0();
    ledSendBit0();
    ledSendBit0();
    ledSendBit0();
    ledSendBit0();
    ledSendBit0();

    ledSendBit0();
    ledSendBit0();
    ledSendBit0();
    ledSendBit0();
    ledSendBit0();
    ledSendBit0();
    ledSendBit0();
    ledSendBit0();
}
```

wykonanie każdego dodatkowego rozkazu (jak na przykład sprawdzenie wartości każdego bitu zmiennej przeznaczonej do wysłania) może dość skutecznie rozłożyć całą transmisję danych, gdyż niezachowane są wówczas rygorystyczne timingi kodujące bity informacji.

Jak sobie z tym poradzić? Dość prosto, choć nie nazbyt optymalnie. Ja utworzyłem osobne funkcje dla każdego rodzaju bitu (0 i 1) i każdego koloru, który zamierzałem przesyłać, a wszelkie warunki zastąpiłem „gotowym” do wykonania kodem bez zbędnych (z punktu widzenia transmisji) rozkazów. Spowodowało to powstanie wielu funkcji o bardzo prostym (i w gruncie rzeczy podobnym), acz długim kodzie, który zachował niezbędne timingi. Te uproszczone funkcje, o których mowa powyżej, pokazałem na **listingu 6**, gdzie zamieszczono odrębne funkcje dla każdego z bitów informacji (0 i 1), jak i przykładową funkcję wysyłającą dane dla koloru czerwonego. Dzięki takiemu rozwiązaniu bez problemu możemy transmitować dane do naszych diod LED, korzystając z mikrokontrolera taktowanego zegarem 8 MHz.

Oczywiście, i czego się zapewne domyślacie, w czasie takiej transmisji należy blokować możliwość obsługi przerwań systemowych. Można również poradzić sobie w prostszy sposób, zwyczajnie zwiększając prędkość zegara taktującego mikrokontroler (najlepiej co najmniej dwukrotnie), lecz ja nie chciałem iść tą drogą, gdyż wydatnie zwiększa się wtedy zapotrzebowanie systemu mikroprocesorowego na energię. Inną sprawą jest to, że w przypadku naszego mikrokontrolera nie mam już wolnych portów I/O, by podłączyć niezbędny rezonator kwarcowy, w związku z czym należałoby zastosować element o większej liczbie wyprowadzeń, co z kolei nie jest pożądane z uwagi na wymiary obwodu drukowanego. Kolejną możliwością jest użycie



Fotografia 1. Zmontowane urządzenie e-Nurse od strony warstwy TOP tuż przed przylutowaniem wyświetlacza LCD

eprasa.pl 1fc8f31a49



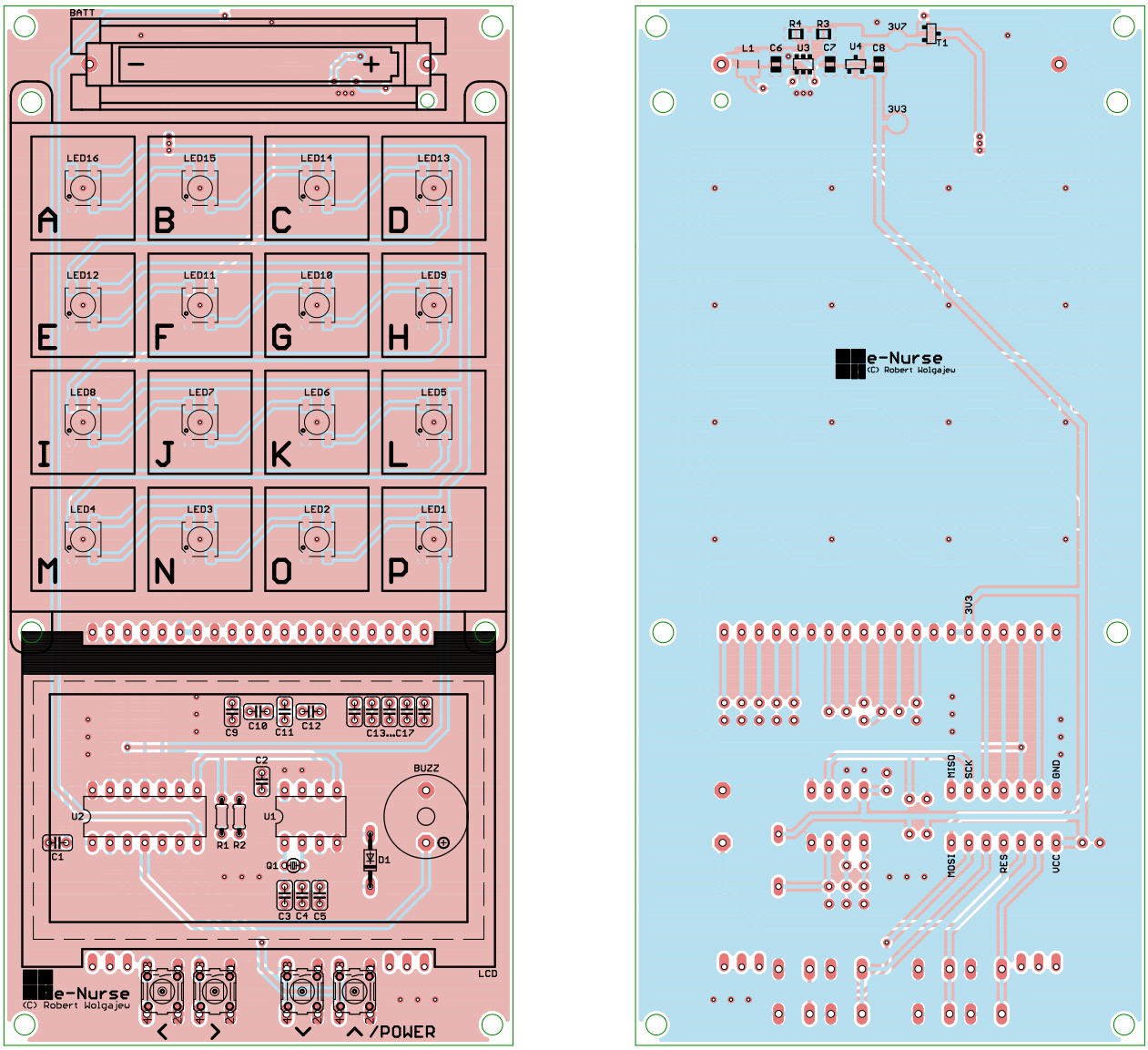
Fotografia 2. Zmontowane urządzenie e-Nurse od strony warstwy BOTTOM

jakiegoś sprzętowego, szeregowego medium transmisyjnego wbudowanego w mikrokontroler (USART lub SPI), ale i w takim przypadku nie uciekniemy od konieczności zwiększenia częstotliwości taktowania mikrokontrolera. Tyle w kwestiach implementacyjnych, w zawiązku z czym przejdźmy do zagadnień montażowych.

Montaż i uruchomienie

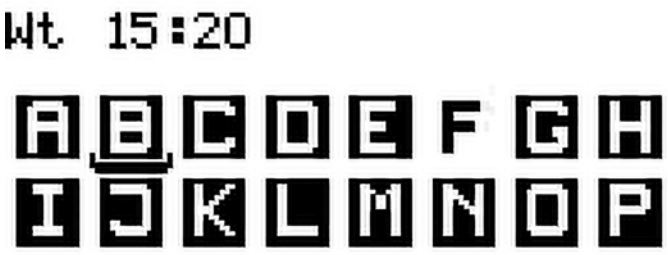
Schemat płytki PCB naszego urządzenia pokazano na **rysunku 5**. Jak widać, zaprojektowano bardzo zgrabną, dwustronną płytkę drukowaną ze zdecydowaną przewagą elementów THT montowanych po obu stronach laminatu (po stronie BOTTOM montowane są wyłącznie elementy SMD). Montaż urządzenia rozpoczynamy od warstwy BOTTOM, gdzie w pierwszej kolejności montujemy półprzewodniki, zaś w kolejnym kroku elementy bierne. W tym momencie przechodzimy na warstwę TOP, gdzie, podobnie jak poprzednio, w pierwszej kolejności montujemy diody LED RGB, następnie pozostałe elementy półprzewodnikowe (z wyłączeniem wyświetlacza), a na końcu elementy bierne (z wyłączeniem buzzera piezoelektrycznego) oraz elementy mechaniczne w postaci koszyczka baterii AAA czy przycisków funkcyjnych. Na koniec, do tak przygotowanego obwodu drukowanego, montujemy wyświetlacz LCD, lutując jego wyprowadzenia w przeznaczone do tego celu punkty lutownicze, co zapewni mu zarówno niezbędne połączenia elektryczne, jak i pożądaną stabilizację mechaniczną.

Na **fotografii 1** pokazano zmontowane urządzenie od strony warstwy TOP tuż przed przylutowaniem wyświetlacza LCD

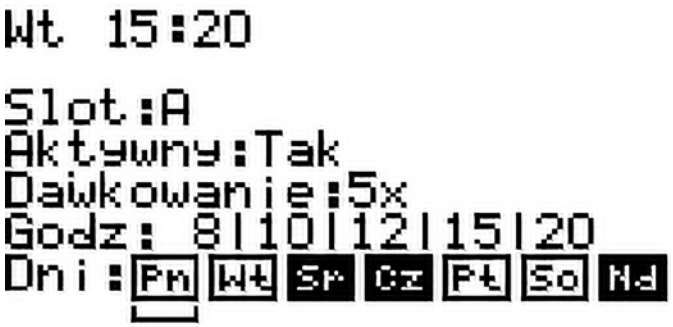


Rysunek 5. Schemat montażowy urządzenia e-Nurse

eprasa.pl 1fc8f31a49



Rysunek 6. Ekran Menu głównego urządzenia e-Nurse



Rysunek 7. Ekran Menu ustawień urządzenia e-Nurse

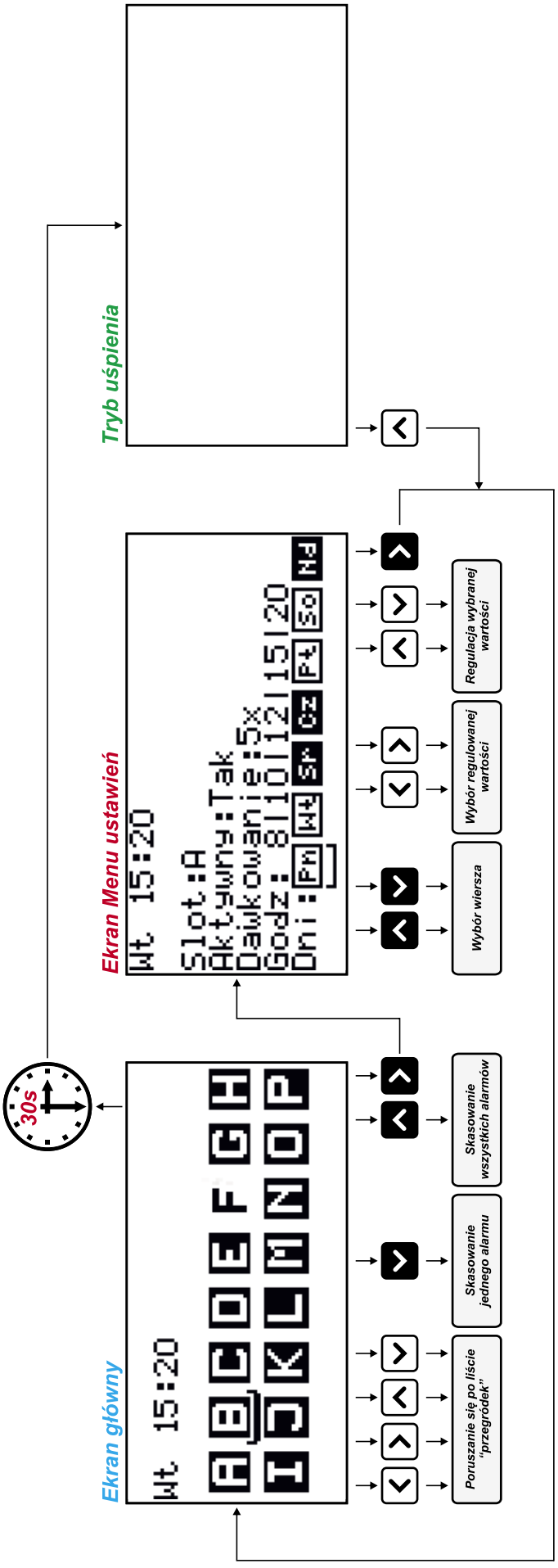
(wersja prototypowa, przed niewielkimi modyfikacjami), zaś na **fotografii 2** to samo urządzenie od strony warstwy BOTTOM (pokazano wyłącznie ten obszar obwodu drukowanego, gdzie montowane są elementy SMD i to przed niewielkimi modyfikacjami sekcji zasilania).

Obsługa

Projektując system Menu urządzenia e-Nurse oraz sposób jego obsługi, przyjąłem, że ergonomia i prostota jego użytkowania i czytelność interfejsu użytkownika powinny być najważniejszym kryterium przy konstruowaniu stosownych procedur sterujących. Zgodnie z tymi podstawowymi założeniami, na płytce sterownika przewidziano wyłącznie 4 przyciski funkcyjne pozwalające na poruszanie się po dostępnych ekranach Menu. Mówiąc zupełnie ogólnikowo, przyciski oznaczone jako NEXT, PREV (symbolizowane przez stosowne symbole strzałek) służą do zmiany typu edytowanego elementu, zaś przyciski oznaczone jako UP, DOWN (symbolizowane przez stosowne symbole strzałek) służą do zmiany wartości edytowanego elementu oraz poruszania się po kolejnych opcjach Menu urządzenia.

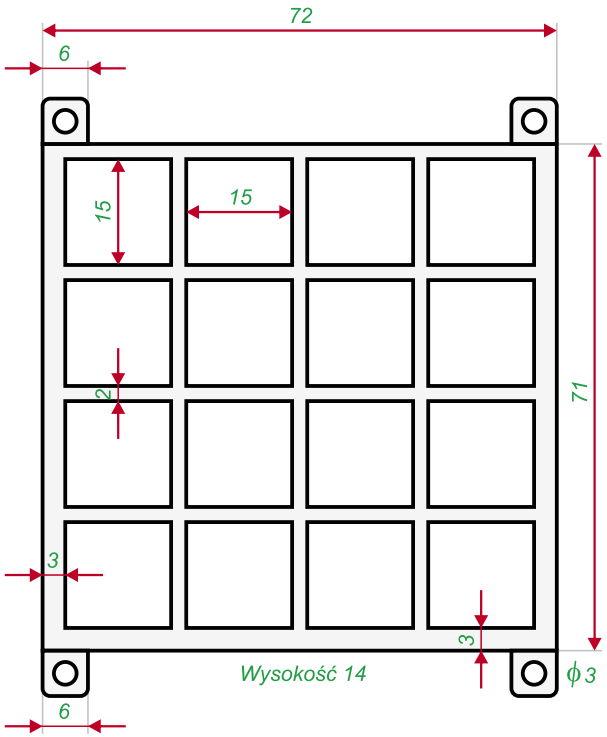
W ramach systemu Menu dostępne są 2 ekrany użytkownika: ekran Menu głównego oraz ekran Menu ustawień. Poniżej, na **rysunku 6**, pokazano ekran Menu głównego urządzenia e-Nurse, zaś na **rysunku 7** pokazano ekran Menu ustawień naszego urządzenia. Jak widać, w ramach Menu głównego dostępna jest informacja o bieżącej dacie i godzinie systemowej oraz informacja o aktywnych alarmach (aktywność alarmu pokazywana jest w postaci wypełnienia wybranego symbolu „przegródki"). Z tego poziomu dokonujemy potwierdzenia przyjęcia dawki leku, co skutkuje skasowaniem alarmu, posiłkując się przy tym przyciskami funkcyjnymi umożliwiającymi poruszanie się po liście ikonek alarmów (przesuwającymi podkreślenie konkretnego symbolu „przegródki") oraz ich kasowanie. Kasowanie może przebiegać w dwojaki sposób: długie przyciśnięcie przycisku DOWN kasuje jeden alarm dla danej „przegródki", zaś długie przyciśnięcie przycisku UP kasuje wszystkie alarmy dla tejże „przegródki". Dodatkowo, długie przyciśniecie przycisku NEXT wprowadza nas w tryb Menu ustawień.

Tryb Menu ustawień pozwala nam na dokonanie ustawień zarówno bieżącej godziny i dnia tygodnia, jak i skonfigurowanie ustawień dla każdej „przegródki" oznaczonej tu jako „Slot". Ustawienia, o których mowa powyżej, to: aktywność „przegródki" (gdyż można ją wstępnie skonfigurować, ale jeszcze nie aktywować), dzienna liczba dawek



Rysunek 8. Kompletny diagram systemu Menu urządzenia e-Nurse wraz ze sposobem jego obsługi

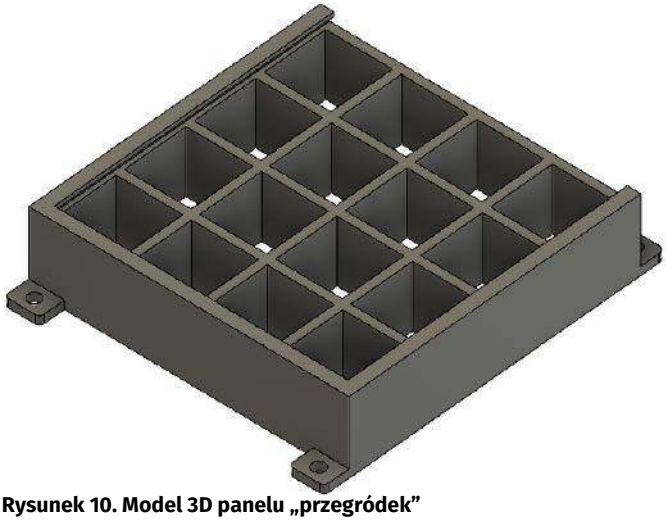
eprasa.pl 1fc8f31a49



Rysunek 9. Rysunek techniczny panelu „przegródek" (z zaznaczeniem ważniejszych wymiarów)

leku (1…5), godziny przyjmowania leku (dowolne z zakresu 0…23) oraz dni tygodnia przyjmowania leku. Poruszanie się pomiędzy elementami poddawanymi edycji, jak i ich edycję umożliwiają krótkie naciśnięcia przycisków funkcyjnych, zaś poruszanie się pomiędzy wierszami opcji Menu ustawień umożliwiają długie naciśnięcia przycisków UP i DOWN.

Wyjście z Menu ustawień i przejście do Menu głównego dokonywane jest poprzez długie naciśnięcie przycisku NEXT. Procesowi temu towarzyszy zapisanie bieżących ustawień urządzenia (ustawień „przegródek") w nieulotnej pamięci EEPROM mikrokontrolera oraz aktualizacja czasu zegara RTC. Do czasu wyjścia z Menu ustawień niemożliwe jest automatyczne przejście urządzenia do stanu uśpienia w przypadku braku aktywnych alarmów. Z kolei, proces sprawdzania alarmów i sygnalizacja ich wystąpienia (za pomocą wbudowanych



Rysunek 10. Model 3D panelu „przegródek"

diod LED RGB oraz buzzera) dokonywana jest również w czasie konfiguracji urządzenia (w Menu ustawień). Na **rysunku 8** pokazano kompletny diagram systemu Menu urządzenia e-Nurse wraz ze sposobem jego obsługi.

Obudowa

I na sam koniec obiecany „ekstras". Gotowy panel obudowy „przegródek" wykonany w aplikacji do projektowania 3D przez mojego, zawsze niezawodnego, Kolegę **Bartłomieja Wawrzyszko**. Element ten zaprojektowany został w taki sposób, by po jego przykręceniu do obwodu drukowanego (4 śruby ∅ 3 mm maks.) stanowił kompletne rozwiązanie dozownika leków zbliżone swoim wyglądem do prostych kasetek na leki dostępnych w aptekach czy drogeriach. Co więcej, w bocznych ściankach elementu (trzech z czterech) przygotowano specjalne wcięcia do wsunięcia w nie przezroczystej pleksi o grubości do 1,5 mm i wymiarach 68×80 mm tak, by stanowiła ona pokrywę dozownika uniemożliwiającą wypadnięcie leków ze swoich „przegródek". Prawda, że fajne?

Na **rysunku 9** pokazano rysunek techniczny panelu „przegródek" (z zaznaczeniem ważniejszych wymiarów), zaś na **rysunku 10** pokazano z kolei model 3D tegoż elementu. Plik produkcyjny do wykonania wyżej wymienionej obudowy udostępniony jest w materiałach dodatkowych do artykułu.

Robert Wołgajew, EP

REKLAMA



eprasa.pl 1fc8f31a49

Podstawowe parametry:
- regulator poziomu zrealizowany na znanym i od wielu lat produkowanym układzie PGA2320 firmy Texas Instruments,
- aktywny układ regulacji tonów niskich i tonów wysokich z możliwością jego całkowitego ominięcia (bypass),
- selektor wejść z 3 wejściami stereofonicznymi przełączanymi miniaturowymi przekaźnikami sygnałowymi,
- zbudowany na bazie wzmacniaczy operacyjnych zoptymalizowanych do zastosowań audio,
- zawiera przetwornik cyfrowo-analogowy na bazie stosunkowo taniego i bardzo dobrego układu PCM1794A.

Dodatkowe materiały do pobrania ze strony www.ulubionykiosk.pl/media

AVT6005-2	toneCtrl – regulator barwy dźwięku (2) (EP 10/2023)
AVT6005-1	toneCtrl – regulator barwy dźwięku (1) (EP 9/2023)
AVT5975	Regulator barwy dźwięku, głośności i balansu (EP 3/2023)
AVT5873	Stereofoniczny aktywny regulator głośności (EP 8/2021)
AVT5851	7-pasmowy korektor graficzny (EP 4/2021)
AVT5816	Regulator balansu tonów (EP 10/2020)
AVT5637	Wielokanałowy regulator głośności VCA (EP 8/2018)
AVT5629	Cyfrowy regulator głośności z układem PT2257 (EP 6/2018)
AVT3222	Sterowany dowolnym pilotem potencjometr audio z przekaźnikiem (EdW 5/2018)

* **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]** – płytka drukowana bez elementów i dokumentacji.

Kity, w których występuje układ scalony wymagający zaprogramowania, mają następujące dodatkowe wersje:
- **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 stronie sklepu osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: **kity@avt.pl**.

Uniwersalny przedwzmacniacz (3)

Prezentujemy ostatnią część opisu przedwzmacniacza. Artykuły w EP 11/23 i EP 12/23 dotyczyły przede wszystkim części sprzętowej i wszelkich zagadnień konstrukcyjnych. Tym razem opiszemy obsługę urządzenia i związane z nią rozwiązania programowe.



Obsługa i program sterujący

Obsługa przedwzmacniacza jest bardzo prosta. Wszystkie niezbędne informacje są wyświetlane na ekranie sterownika – **fotografia tytułowa**. Ekran został podzielony na 2 obszary: pierwszy większy zawiera informacje o tłumieniu/wzmocnieniu sygnału – Volume (w dB), przez układ PGA2320. Drugi obszar wyświetla informacje o stanie układu pomijania filtra barwy tonu – Bypass (włączony/wyłączony), oraz o aktywnym wejściu analogowym InputA.

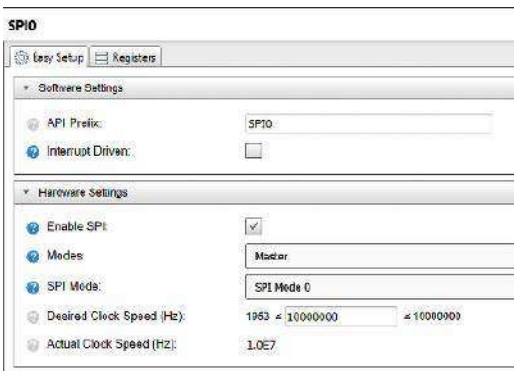
Kręcenie osią enkodera w prawo powoduje zmniejszanie tłumienia, a po przejściu przez zero – zwiększanie wzmocnienia. Kręcenie w lewo powoduje zwiększanie tłumienia lub zmniejszanie wzmocnienia. Krótkie przyciśnięcie i puszczenie osi enkodera sekwencyjnie zmienia aktywne wejście selektora IN1 –> IN2 –> DAC. Dłuższe przyciśnięcie i puszczenie osi enkodera (kilka sekund) sekwencyjnie włącza i wyłącza układ Bypass. Kiedy Bypass jest w pozycji ON, to filtr barwy tonu jest wyłączony z toru sygnałowego.

W obecnej wersji programu nie ma zaimplementowanej obsługi zdalnego sterowania. Jednak układ jest na to przygotowany sprzętowo i w planach jest wykonanie takiej wersji.

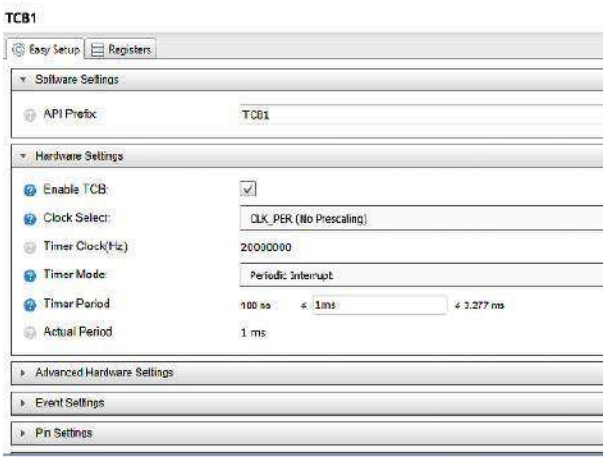
Program sterujący

Program sterujący został napisany w języku C w środowisku MPLAB X IDE. Użyłem bezpłatnej wersji kompilatora MPLAB XC8 przeznaczonego dla 8-bitowych mikrokontrolerów firmy Microchip rodzin PIC16, PIC18 i AT. Bezpłatna wersja ma tylko ograniczenie optymalizacji rozmiaru kodu. Poza tym jest to w pełni funkcjonalny kompilator. W naszym przypadku nie ma to znaczenia, bo w użytym mikrokontrolerze pozostaje i tak prawie połowa dostępnej pamięci.

Do konfiguracji tych interfejsów, ale też portów i systemu taktowania użyłem wtyczki – konfiguratora do programu MPLAB X IDE o nawie MCC. Program używa dwu interfejsów sprzętowych: modułu transmisji szeregowej SPI (**rysunek 17**) do komunikacji z wyświetlaczem i modułu timera TCB1



Rysunek 17. Konfiguracja interfejsu SPI



Rysunek 18. Konfiguracja timera TCB1

eprasa.pl 1fc8f31a49

Wykaz elementów:

Blok przedwzmacniacza, DAC i SPDIF

Rezystory: (SMD1206)
R1, R3, R25, R27: 3,3 kΩ
R2, R6, R7, R30, R31, R39, R42, R94: 47 kΩ
R4, R10, R11, R16, R20, R21, R24, R28, R33, R35, R41, R44, R54, R55, R56, R57, R58, R59, R69, R74, R75, R76, R77, R78, R83, R84, R86, R90: 1 kΩ
R5, R14, R29, R37: 6,8 kΩ
R8, R12, R36: 910 Ω
R9: 1,5 kΩ
R13, R17, R22, R23, R48, R63, R66, R82, R87: 620 Ω
R15, R38: 560 Ω
R19, R40: 12 kΩ
R26: 100 kΩ
R34: 1,5 kΩ
R43, R61, R73, R80, R92: 330 Ω
R45, R50: 91 Ω
R46, R52, R53: 150 Ω
R47, R49, R60, R67, R79, R89: 820 Ω
R51: 60 Ω
R62, R72, R81, R88, R91: 680 Ω
R64, R85: 100 Ω
R65, R68, R70, R71: 27 Ω
R93: 75 Ω
R95: 2,2 kΩ

Kondensatory:
C1, C2, C18, C19, C23, C24, C27, C31: 100 µF/16 V tantalowy SMD
C3, C7, C14, C16: 1000 µF/35 V THT
C4, C5, C9, C20, C21, C25, C26, C28, C32, C37, C40, C41, C48, C51, C52, C53, C60, C61, C65, C66, C67, C72, C77, C79, C80, C82, C86, C88, C90, C94, C98, C103: 100 nF ceramiczny (SMD1206)
C6, C38, C39, C42, C43, C44, C45, C46, C47, C102, C104: 10 µF/16 V tantalowy SMD
C8, C11, C29, C33: 10 nF
C10, C30, C100: 47 nF
C12, C17, C34, C36: 470 nF
C13, C22: 100 pF ceramiczny
C15, C35: 330 nF
C49, C50: 2,2 µF
C54, C55, C56, C62, C63, C68, C74, C78, C81, C83, C87, C89, C92, C99: 10 µF/16 V elektrolityczny SMD
C57, C58, C59, C69, C70: 47 µF/16 V elektrolityczny SMD
C64, C76, C84, C96: 2,2 nF ceramiczny (SMD1206)
C71, C75, C85, C95: 2,7 nF ceramiczny (SMD1206)
C73, C91: 8,2 nF ceramiczny (SMD1206)
C93: 4,7 nF ceramiczny (SMD1206)
C97: 68 nF ceramiczny (SMD1206)
C101: 10 nF ceramiczny (SMD1206)

Półprzewodniki:
D1, D2, D3, D4, D6, D7, D8, D9, D11, D12, D14, D15, D16, D17, D19, D20: 1N4007
D5, D10, D13, D18, D21: 1N4148
Q1, Q2, Q3, Q4, Q5, Q6: BC237
U1, U2, U3, U12, U14, U16, U17: OPA1612
U4: PGA2320IDW
U5, U6, U7: LM317
U8: LM337
U9, U10, U11: TL431 (SOT23)
U13: PCM1794A (SMD)
U15: DIR9001PW (SMD)
U18: stabilizator 3,3 V (SO8)
U19: SN74HCU04D (SO14)

Pozostałe:
F1, F2: bezpiecznik kubełkowy
J2: goldpin kątowy 1×18
P5, P6: potencjometr 2×10 kΩ
P7, P8, P11, P12: goldpin
ZZAC: ARK2/500
K1, K2, K3, K4, K5: TE V23079 cewka 5 V
IN1R, IN2R, IN1L, IN2L, OUTL, OUTR, SPDIF: gniazdo cinch kątowe do PCB
T1: transformator 2×12 V TSZZ 12/008M
T2: transformator 2×6 V TSZZ 7/004M

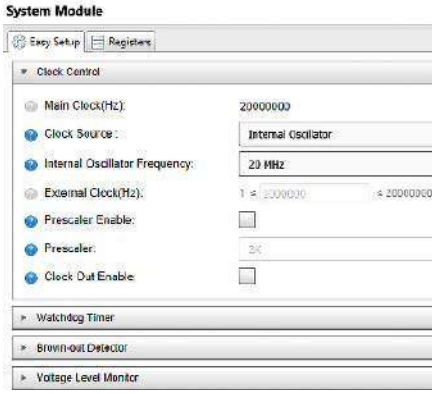
Blok sterownika mikroprocesorowego

Rezystory: (SMD1206)
R1, R2, R4, R6, R7, R8, R9, R13: 10 kΩ
R5: 150 Ω
R10, R11, R12, R14: 150 Ω

Kondensatory:
C1, C9, C13: 100 nF (SMD0805)
C3, C4, C7, C11, C12, C14, C15, C16: 10 µF/16 V tantalowy SMD
C8, C10, C17: 100 nF (SMD1206)

Półprzewodniki:
U1: ATmega4808 (TSSOP28)
U2: SPX1117-3.3 (TO252)
U3: odbiornik IR
U4: TLP281-4 (SO16)
U5: 7805

Pozostałe:
J1: wyświetlacz 1,8 cala, rozdzielczość 160×128 pikseli, sterownik ST7735
IMP1: impulsator z przyciskiem
J5: goldpin 1×2

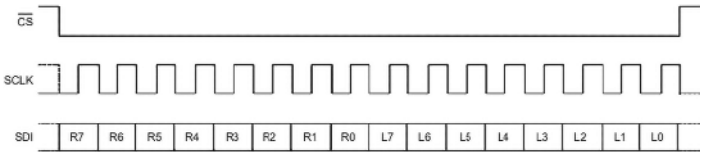


Rysunek 19. Taktowanie mikrokontrolera

(**rysunek 18**). TCB1 zgłasza przerwanie co 1 ms. W procedurze przerwania od tego timera jest uruchamiany pracujący w tle program obsługujący odczytywanie obrotu enkodera.

Mikrokontroler jest taktowany częstotliwością 20 MHz (**rysunek 19**), pomimo że przy zasilaniu napięciem +3,3 V według specyfikacji producenta powinno to być maksymalnie 15 MHz. Przeprowadzone testy wykazały, że w temperaturze pokojowej mikrokontroler pracuje stabilnie przy 20 MHz. Pozwala to na płynniejsze wyświetlanie informacji na ekranie. Na pozostałe funkcje programu nie ma to wpływu.

Komunikacja z układem PGA2320 jest realizowana przez interfejs SPI z 16-bitowym słowem danych. Działanie tego interfejsu jest emulowane programowo. Maksymalna katalogowa częstotliwość zegara CLK może sięgać nieco ponad 6 MHz. Jednak pamiętamy, że w celu wyeliminowania zakłóceń generowanych przez mikrokontroler i wyświetlacz została zastosowana izolacja galwaniczna mas wykonana na transoptorach. Takie rozwiązanie znacząco ogranicza częstotliwości prawidłowo przenoszonych sygnałów cyfrowych. W praktyce zegar nie może mieć częstotliwości większej niż kilka kHz. Ponieważ ilość przesyłanych danych w czasie jest bardzo mała, to nie jest to istotne ograniczenie. W trakcie pojedynczego przestawienia tłumienia PGA2320 jest przesyłanych tylko 16 bitów – **rysunek 20** (R0...R7 ustawienie kanału prawego, L0...L7 ustawienie kanału lewego)



Rysunek 20. Protokół wymiany informacji z PGA2320

Na **listingu 1** pokazano procedurę wysyłania 16-bitowego słowa do PGA2320, a na **listingu 2** przykład programowego sterowania linią SDO dla mikrokontrolera ATmega. Sterowanie pozostałymi liniami CS i SCK przebiega analogicznie.

Na **rysunku 21** pokazano oscylogramy rzeczywistych przebiegów na liniach – kolejno od góry: CS, SDO i SCK. Cykl przesyłania danych zaczyna się od aktywowania interfejsu przez wystawienie stanu niskiego na linii CS. W czasie, kiedy na linii SCK jest stan niski, na linii SDO jest ustawiany bit danych. Ten bit jest przesyłany do PGA2320 przy narastającym zboczu sygnału SCK.

Tłumienie/wzmocnienie układu PGA2320 jest określane według zależności:

$$Gain(dB) = 3{,}15 - [0{,}5 \cdot (255 - N)]$$

gdzie N jest 8-bitową wartością przesyłaną do układu dla każdego z kanałów (zostało to pokazane na rysunku 20). Program operuje na wartości *N* z powyższej zależności, ale do wyświetlania wartości w decybelach jest wyliczana wartość *Gain(dB)* zapisywana w zmiennej typu *float*. Potem ta wartość jest konwertowana na ciąg znaków przez standardową funkcję *sprintf* – **listing 3**.

Listing 1. Programowa realizacja sterowania układem PGA2320

```
void PGA_Ctrl (uint8_t volL, uint8_t volR) {
    uint8_t i;
    uint16_t volu
    //16 bit do przesłania
    volu = volR;
    volu = (volu << 8);
    volu = volu |volR;

    PGA_Ctrl_SCK (LO);          //SCK = 0;
    PGA_Ctrl_CS (LO);           //CS = 0
    for(i= 0; i< 16; i++) {
        PGA_Ctrl_SCK (LO);      //SCK =0

        //wysłanie bitu
        if((volu & 0x8000) == 0)
            PGA_Ctrl_SDO (LO);
        else
            PGA_Ctrl_SDO (HI); //SCK = 1
        //kolejny bit
        volu = (volu << 1);
        PGA_Ctrl_SCK (HI);
    }
    PGA_Ctrl_CS (HI);           //CS=1
    PGA_Ctrl_SCK (HI);          //SCK=0
    PGA_Ctrl_SDO (HI);          //SDO=0
}
```

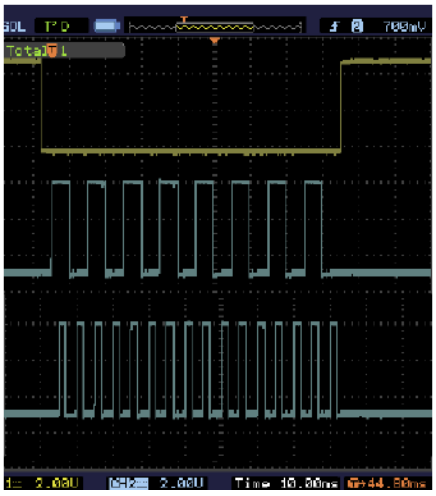
Listing 2. Sterowanie linią SDO

```
/*sterowanie linia danych magistrali PGA*/
void PGA_Ctrl_SDO(uint8_t cmd) {
    if(cmd == LO) {
        PORTD.OUTCLR =(1 << SDOP);
    }
    else
        if(cmd == HI)
            PORTD.OUTSET =(1 << SDOP);
    _delay_ms(1);//opóźnienie – izolacja transoptora
}
```

Listing 3. Przygotowanie do wyświetlania wartości tłumienia w dB

```
// wartość N zapisana w zmiennej vol
char txt[5];
float gain;
gain = 31.5 -(0.5*(255-vol));
sprintf(txt,"%2.1f",gain);
```


eprasa.pl 1fc8f31a49



Rysunek 21. Oscylogramy przesyłania danych do PGA2320

Program sterujący w obecnej wersji wpisuje w trakcie regulacji do obu kanałów PGA2320 takie same wartości. Nie ma zatem możliwości realizacji funkcji balansu. W poprzednich moich przedwzmacniaczach sterowanych cyfrowo funkcja balansu był wbudowana, ale nigdy jej nie używałem. Dlatego teraz z niej zrezygnowałem. W pętli regulacji jest inkrementowana lub dekrementowana 8-bitowa zmienna odpowiadająca tłumieniu dla obu kanałów. Nowo ustawiona wartość jest zapisywana w pamięci eeprom, wyświetlana i wysyłana do układu PGA2320 – **listing 4**.

W obecnej wersji oprogramowania jest używany sygnał wyciszenia MUTE – **listing 5**. Tak jak linie interfejsu SPI również MUTE jest izolowany galwanicznie. Sterowanie wyciszaniem jest używane do eliminacji zakłóceń (stuków) w trakcie działania selektora wejść i przełączania układu BYPASS – **listing 6**.

Podsumowanie

Zaprezentowany przedwzmacniacz jest z założenia układem, który ma spełniać funkcję poprawnie skonstruowanego toru audio. Nie ma tu jakichś specjalnych układów: wymyślnego zasilania, lampowego konwertera w układzie przetwornika, drabinkowych „potencjometrów" czy buforów na tranzystorach FET. Za to szczególną uwagę zwrócono na poprawne dopasowanie impedancji pomiędzy poszczególnymi stopniami.

Prawidłowe działanie toru regulacji barwy wymaga, by impedancja źródła sygnału na wejściu miała niską wartość, a jego wyjście było obciążone stopniem o wysokiej impedancji wejściowej. Również układ PGA2320 wymaga, by jego wejścia były sterowane źródłem sygnału o niskiej impedancji wyjściowej. Redukuje to szumy układu i zniekształcenia nieliniowe.

Wszystkie te wymogi najłatwiej jest spełnić, korzystając z dobrych wzmacniaczy operacyjnych. W prototypie zastosowałem podwójne wzmacniacze OPA1612 zoptymalizowane do pracy w wysokiej klasy torach audio. Również w filtrze regulacji barwy pracuje układ OPA1612. W sumie w torze audio pracują trzy podwójne wzmacniacze operacyjne. Żeby wydajnie poprawić uniwersalność przedwzmacniacza, zdecydowałem się umieścić w nim kompletny przetwornik analogowo-cyfrowy z bardzo dobrym układem PCM1794A z konwerterem prąd/napięcie i filtrem analogowym, również zbudowanym na wzmacniaczach operacyjnych OPA1611 i OPA1611.

Źródłem danych przetwornika może być sygnał S/PDIF z odtwarzacza CD lub dane przesyłane po USB z komputera. Wymaga to zamontowania dodatkowego modułu USB/I²S z wyjściami zgodnymi ze standardem Amanero. Opcjonalnie można tu zamontować moduł Bluetooth 5.x audio z wyjściem I²S i używać smartfonu jako źródło sygnału. Smartfony używające kodeków Aptx_HD, LDAC itp. w połączeniu z platformami oferującymi niską kompresję danych audio potrafią być źródłem dźwięku zaskakująco dobrej jakości. Przetwornik charakteryzuje się bardzo dobrą jakością dźwięku.

Tor audio przy włączonym układzie Bypass jest raczej neutralny. Dotyczy to zarówno wejść analogowych, jak i wyjścia przetwornika cyfrowo-analogowego. Sytuacja zmienia się po wyłączeniu funkcji Bypass i tym samym włączeniu w tor regulatorów barwy. Jak wspominałem przy opisie układu filtrów, układ regulacji barwy charakteryzuje się świetnymi parametrami elektrycznymi, szczególnie jeżeli chodzi o charakterystykę fazową. Układ działa subtelnie, ale w krańcowych położeniach potencjometrów jest wyraźnie słyszalny. Jednak, jeżeli ktoś oczekuje efektów rodem ze wzmacniaczy starej PRL-owskiej Unitry, w której oprócz agresywnie działających filtrów barwy jest wbudowany niewyłączalny filtr Kontur, to być może będzie zawiedziony.

Zastosowany regulator barwy jest bardzo dobrym wyborem przy cichym słuchaniu materiału muzycznego bardzo dobrej jakości. Przyznam, że użytkuję ten przedwzmacniacz głównie z wyłączona funkcją Bypass. W połączeniu z lampowym wzmacniaczem mocy Sansuix daje to rewelacyjny efekt przy cichym słuchaniu. Audiofile od wielu lat raczej nie używają regulatorów barwy, mimo że w złotych czasach audio (lata 70. i 80. XX wieku) każdy wzmacniacz taki regulator miał. Tu mamy wybór. Można układ filtrów całkowicie ominąć poprzez styki przekaźnika i być w zgodzie z audiofilskim sumieniem. Jeżeli ktoś jednak nie traktuje słuchania muzyki ortodoksyjnie, to być może doceni nowe możliwości. W końcu chodzi o to, aby przy tym czerpać jak najwięcej przyjemności.

Tomasz Jabłoński, EP

Listing 4. Pętla regulacji tłumienia

```
//volume DOWN zwiększ tłumienie
if(key == KOD_IMP_DWN) {
    ++vol;
    if(vol == 255)
        vol = 254;
    //wyświetl wartość tłumienia w dB
    HMI_DispVol( 5,25,vol);

    //utwórz 16 bitowa wartość z dwu takich samych bajtów
    volume = vol;
    volume = ((vol <<8)|vol);
    //zapamiętaj wartość tłumienia w eeprom
    HMI_PutVolume(volume);
    PGA_Ctrl (vol,vol);//wyslij do PGA
}
//volume UP zmniejsz tłumienie
if(key == KOD_IMP_UP) {
    if(vol != 0) {//korekta dla wartości maksymalnej
        --vol ;
        if(vol == 0)
            vol = 1;
    }
    HMI_DispVol( 5,25,vol);
    volume = vol;
    volume = ((vol <<8)|vol);
    HMI_PutVolume(volume);
    PGA_Ctrl (vol,vol);//wyslij do PGA
}
```

Listing 5. Przełączanie wejść z funkcją MUTE

```
void HMI_Set_Audio_Input(uint8_t input) {
    PGA_Ctrl_MUTE(1);//mute on
    _delay_ms(5);

    if(input == 0 ) {// wybranie wejścia IN1
        PORTD.OUT |= (1 << SEL1); //aktywny
        PORTD.OUT &= ~(1 << SEL2);
        PORTD.OUT &= ~(1 << SEL3);
    }
    if(input == 1 ) {// wybranie wejścia IN2
        PORTD.OUT |= (1 << SEL2); //aktywny
        PORTD.OUT &= ~(1 << SEL1);
        PORTD.OUT &= ~(1 << SEL3);
    }
    if(input == 2 ) {//wybranie wyjścia DAC
        PORTD.OUT |= (1 << SEL3); //aktywny
        PORTD.OUT &= ~(1 << SEL1);
        PORTD.OUT &= ~(1 << SEL2);
    }
    PGA_Ctrl_MUTE(0);//mute off
    _delay_ms(50);
}
```

Listing 6. Sterowanie funkcją BYPASS

```
void HMI_Set_Audio_Bypass(uint8_t bypass) {
    PGA_Ctrl_MUTE(1);//mute on
    _delay_ms(5);

    if(bypass == 0)
        //włącz BYPASS
        PORTA.OUT |= (1 << BYPAS);
    if(bypass == 1)
        //wyłącz BYPASS
        PORTA.OUT &= ~(1 << BYPAS);

    PGA_Ctrl_MUTE(0);//mute off
    _delay_ms(50);
}
```


eprasa.pl 1fc8f31a49

W ofercie AVT*
AVT6020

Podstawowe parametry:
- zliczanie impulsów wejściowych w zakresie 0...999999,
- zadawanie wartości progowej, po której następuje załączenie przekaźnika,
- maksymalna częstotliwość impulsów: około 20 kHz,
- wejście impulsów z możliwością dołączenia rezystora typu pull-up lub pull-down,
- możliwość ustawienia reakcji na zbocze opadające, narastające lub oba,
- nastawy przechowywane w nieulotnej pamięci EEPROM,
- zewnętrzne wejście zerujące + przycisk monostabilny na płytce,
- dwa wyjścia (NO i NC) przekaźnika o obciążalności do 8 A,
- alfanumeryczny wyświetlacz LCD o organizacji 2 wiersze po 16 kolumn,
- zasilanie napięciem stałym 12 V (lub 24 V po modyfikacji).

Dodatkowe materiały do pobrania ze strony www.ulubionykiosk.pl/media

AVT5983	Zaawansowany licznik impulsów (EP 5/2023)
AVT5945	sCounter (EP 8/2022)
AVT1824	Programowany licznik zdarzeń (EP 8/2014)
AVT1750	Licznik impulsów (EP 8/2013)
AVT3188	Licznik impulsów (zdarzeń)
AVT1810	Uniwersalny licznik z LCD

* **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]** – płytka drukowana bez elementów i dokumentacji.

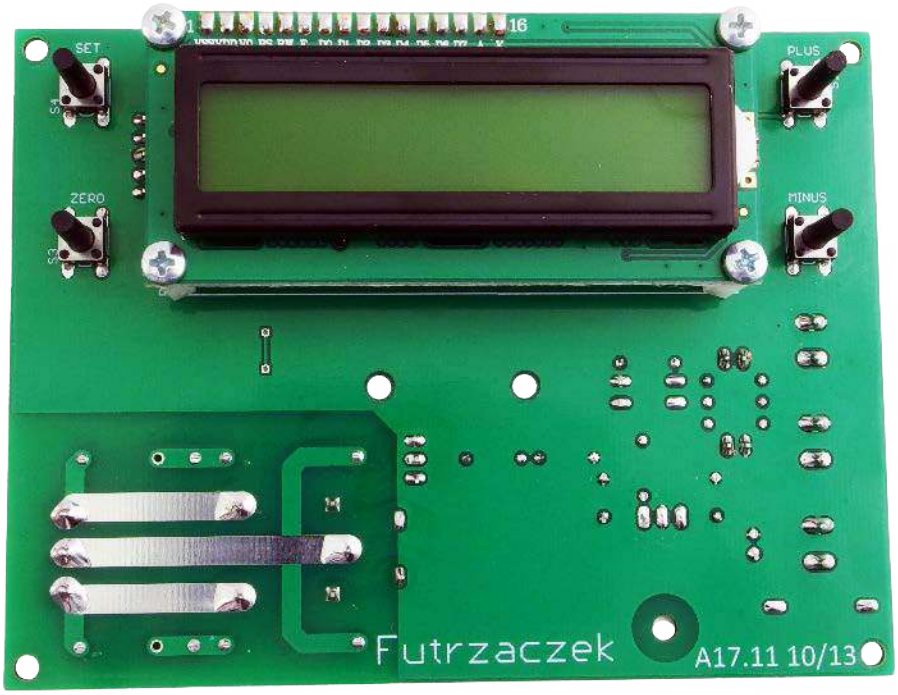
Kity, w których występuje układ scalony wymagający zaprogramowania, mają następujące dodatkowe wersje:
- **wersja [A+]** – płytka drukowana **[A]** + zaprogramowany układ **[UK]** i dokumentacja,
- **wersja [UK]** – zaprogramowany układ.

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 stronie sklepu osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: **kity@avt.pl**.

Progowy licznik impulsów

Układów zliczających impulsy jest w handlu całe mnóstwo. Z różnymi wyświetlaczami, z rozmaitymi funkcjami, cechujące się różną pojemnością. Lecz niewiele z nich potrafi zasygnalizować osiągnięcie uprzednio zadanej liczby impulsów. Taka funkcja może się bardzo przydać choćby przy automatyzacji pakowania drobnych przedmiotów.



Niektóre produkty można zliczać poprzez ich zważenie i podzielenie uzyskanego wyniku przez średnią masę jednej sztuki, lecz muszą one być jednakowe. W innym przypadku trzeba policzyć każdą sztukę z osobna. A gdyby tak układ elektroniczny mógł sygnalizować operatorowi, że odpowiednia liczba sztuk już wpadła do opakowania, byłoby cudownie. Taką możliwość daje zaprezentowany układ. Nie musi on służyć jedynie do określania liczby przedmiotów, gdyż po sprzęgnięciu z enkoderem inkrementalnym może wspomagać odmierzanie odcinków o określonej długości. Obsługa tego urządzenia jest banalnie prosta, a jego możliwości są naprawdę spore!

Budowa i działanie

Schemat ideowy omawianego układu znajduje się na **rysunku 1**. Do zliczania impulsów i obsługi wyświetlacza służy mikrokontroler US1 typu ATmega8A. Nie jest to najnowszy układ, lecz pozostaje popularny i niezawodny. Nie realizuje w tym urządzeniu zadań krytycznych pod względem czasowym, więc taktowanie jego rdzenia zegarem o częstotliwości 8 MHz, którego źródłem jest wbudowany oscylator RC, całkowicie wystarcza do poprawnego działania niniejszego układu. Nieopodal jego wyprowadzeń zasilania znajdują się kondensatory C1...C3, które filtrują napięcie zasilające z tętnień. Ich źródłem mogą być zarówno tranzystory wchodzące w skład struktury US1, jak i okoliczne elementy, na przykład diody zabezpieczające wejścia.

Aby wygodnie zadawać żądane wartości, w układzie znalazły się cztery przyciski monostabilne S1...S4. Wewnętrzne rezystory podciągające, które udostępnia mikrokontroler, zostały wsparte dodatkowymi, zewnętrznymi, co zwiększa odporność na zakłócenia. Zwiększeniu wygody – tym razem nie codziennej obsługi, lecz programowania pamięci Flash mikrokontrolera US1 – służy z kolei złącze J1, na które zostały wyprowadzone sygnały interfejsu ISP, popularnego w układach z rodziny AVR. Rezystory z drabinki RN1 ustalają ich potencjał względem masy, zapewniając tym samym możliwość rozładowania wyprowadzeń złącza J1 z nagromadzonych ładunków elektrostatycznych.

Impulsy wejściowe należy podawać na zaciski złącza J2. Sterują one bezpośrednio wejściem mikrokontrolera, więc dla poprawnego działania wymaga się, by ich amplituda była nie mniejsza niż 3,5 V. Jeżeli byłaby wyższa, diody D1 i D2 ograniczą ją do akceptowalnej wartości. Rezystor R5 ogranicza prąd tych diod, zaś R6 prąd diod zabezpieczających wbudowanych w mikrokontroler – takie dwustopniowe zabezpieczenie bardzo dobrze zabezpiecza nawet przed krótkotrwałymi wyładowaniami elektrostatycznymi. Do dyspozycji użytkownika są również rezystory podciągające: jeden do dodatniej linii zasilającej (R3), zaś drugi do masy (R4). Można je załączyć przez nałożenie zworki

eprasa.pl 1fc8f31a49

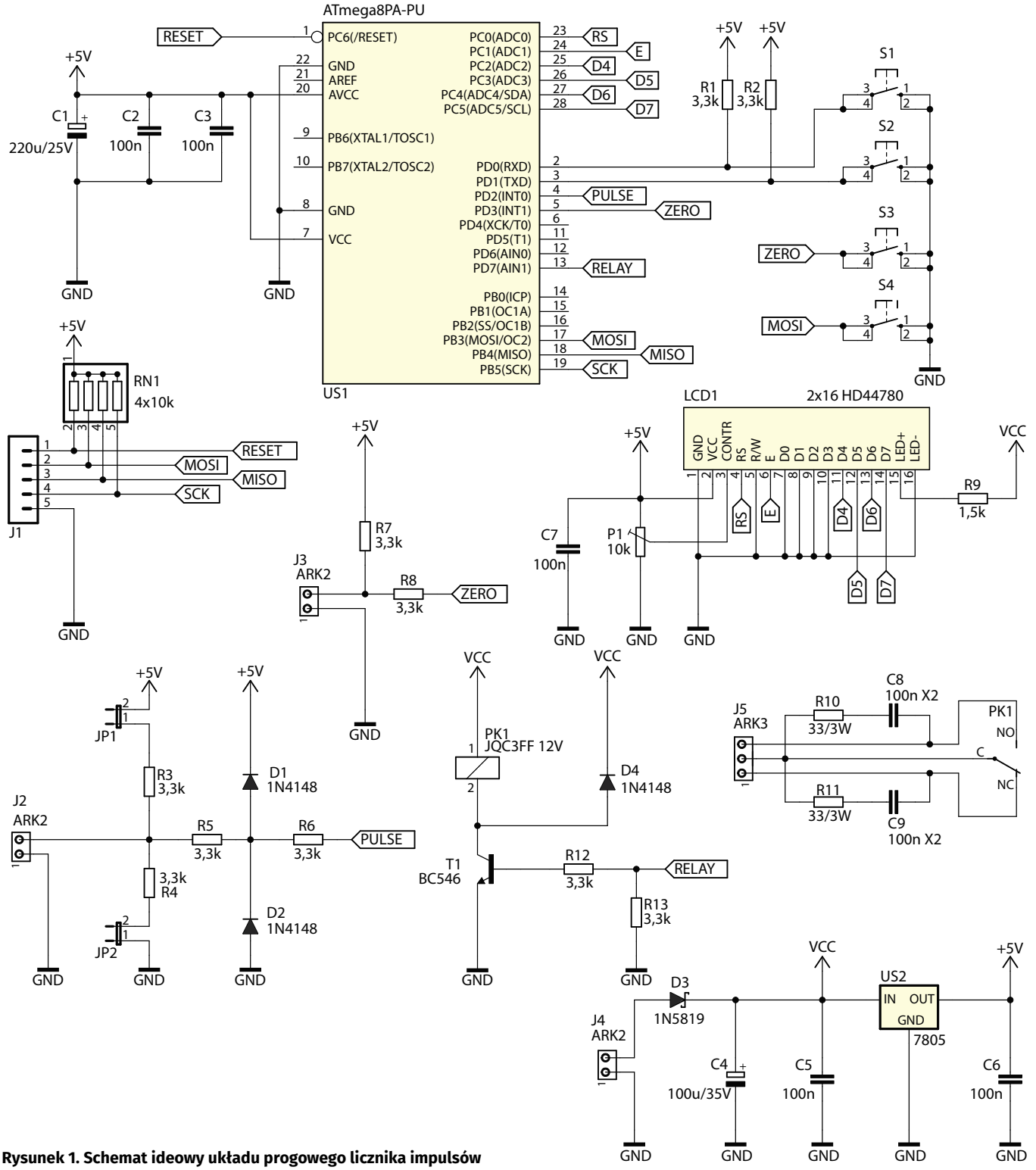
na jedno ze złączy, odpowiednio JP1 lub JP2, lecz można też ich nie załączać, wówczas wejście impulsów przedstawia sobą bardzo wysoką impedancję.

W prostszy sposób zostało zabezpieczone wejście zerujące, którego zaciski znajdują się w złączu J3. Rezystor R8 zapewnia ograniczenie prądu diod zabezpieczających wejście mikrokontrolera, ponieważ założono, że między zaciskami tego złącza a zewnętrznym przyciskiem monostabilnym będzie krótki odcinek przewodu. Tę samą funkcję spełnia również przycisk S3, znajdujący się na płytce, nie trzeba koniecznie montować zewnętrznego.

Napięcie zasilające układ podłącza się do zacisków złącza J4. Dioda D3 odcina zasilanie w przypadku pomylenia polaryzacji tego napięcia. Stabilizator liniowy typu 7805 dostarcza napięcia 5 V dla układów cyfrowych. Moc strat na nim jest na tyle niska, że nie ma potrzeby stosowania przetwornicy impulsowej – jego metalowa wkładka wystarcza do odprowadzania ciepła.

Użytkownik widzi informacje podawane przez układ na alfanumerycznym wyświetlaczu LCD1, który zawiera sterownik typu HD44780 lub zgodny z nim. Ma dwa wiersze po szesnaście znaków w każdym. Potencjometrem P1 ustawia się kontrast owego wyświetlacza. Rezystor R9 pozwala na zasilanie jego podświetlenia z niestabilizowanego napięcia wejściowego, nie obciążając tym samym stabilizatora US2.

Zasilanie dla sterowanego przez układ urządzenia powinno być poprowadzone przez zaciski złącza J5, do którego są doprowadzone wyprowadzenia styków normalnie otwartych (NO) oraz normalnie zamkniętych



Rysunek 1. Schemat ideowy układu progowego licznika impulsów

eprasa.pl 1fc8f31a49

Wykaz elementów:

Rezystory: (THT o mocy 0,25 W jeżeli nie napisano inaczej)
R1...R8, R12, R13: 3,3 kΩ
R9: 1,5 kΩ (opis w tekście)
R10, R11: 33 Ω 3 W
RN1: 4 × 10 kΩ SIL5
P1: 10 kΩ montażowy leżący jednoobrotowy

Kondensatory:
C1: 220 µF 25 V raster 2,5 mm
C2, C3, C5...C7: 100 nF raster 5 mm MKT
C4: 100 µF 35 V raster2,5 mm
C8, C9: 00 nF MKP X2 305 V AC raster 15 mm

Półprzewodniki:
D1, D2, D4: 1N4148
D3: 1N5819
LCD1: 2×16, zgodny z HD44780 np. LCD2X16 15
T1: BC546
US1: ATmega8A-PU DIP28
US2: 7805 TO220

Pozostałe:
J1: goldpin 5 pin męski 2,54 mm THT
J2...J4: ARK2/500
J5: ARK3/500
JP1, JP2: goldpin 2 pin męski 2,54 mm THT + zworka
PK1: JQC3FF/121ZS (opis w tekście)
S1...S4: microswich 6×6 13,5 mm
Jedna podstawka DIP28 wąska
Złącze męskie i żeńskie goldpin 1×16 pin 2,54 mm
Cztery tuleje dystansowe gwint wewnętrzny M3
12 mm poliamid
Osiem śrub M3 6 mm

(NC) przekaźnika PK1. Rezystor (33 Ω) i kondensator (100 nF z dielektrykiem typu X2) tworzą prosty układ pochłaniający energię wydzielaną w momencie łączenia i rozłączania styków przekaźnika, kiedy dochodzi do ich iskrzenia. Zmniejsza to ryzyko wystąpienia błędu w funkcjonowaniu układu, ponieważ redukcji ulega poziom emitowanych zakłóceń elektromagnetycznych. Zabezpieczone są w ten sposób zarówno styki NO, jak i NC.

Montaż i uruchomienie

Układ został zmontowany na dwustronnej płytce drukowanej o wymiarach 120×90 mm, której schemat pokazano na **rysunku 2**. W odległości 3 mm od krawędzi płytki znalazły się cztery otwory montażowe, każdy o średnicy 3,2 mm. Dodatkowo, aby płytka nie uginała się podczas naciskania przycisków, w pobliżu środka płytki dodano dodatkowe otwory. **Rysunek 3** pokazuje ich lokalizację na płytce.

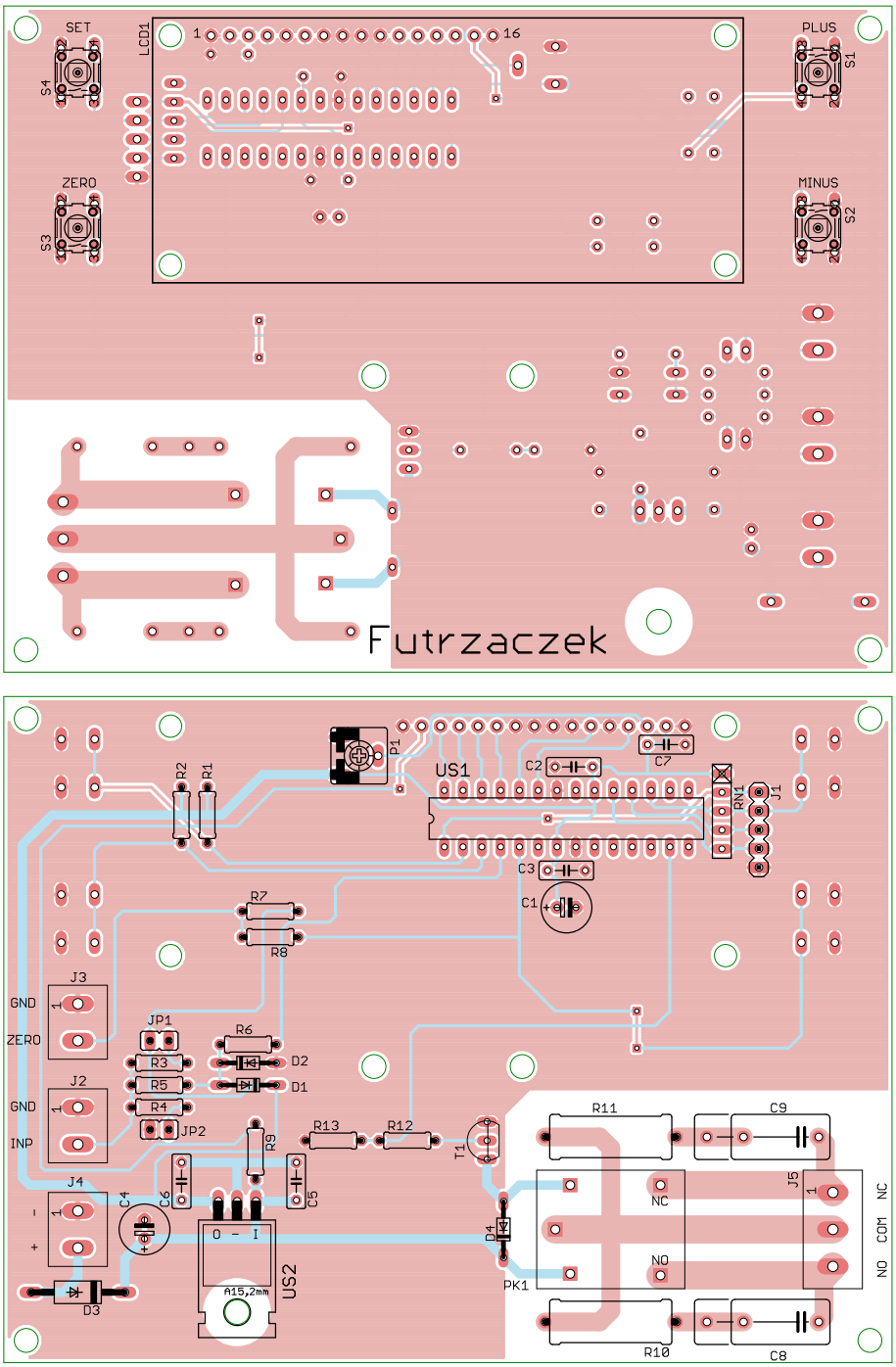
Montaż proponuję rozpocząć od elementów o najmniejszej wysokości obudowy, czyli rezystorów i diod. Pod mikrokontroler warto zastosować podstawkę. Stabilizator US1 można dogiąć tak, aby położyć go na powierzchni laminatu lub też pozostawić w pionie, aby wygodniej móc przykręcić do niego niewielki radiator. Ten etap montażu można zobaczyć na **fotografii 1**.

Na drugiej stronie płytki jest miejsce dla wyświetlacza LCD1 oraz czterech przycisków S1...S4. Ich lokalizację po przylutowaniu przedstawia fotografia tytułowa. Na samym końcu, po wlutowaniu wszystkich przycisków, należy przykręcić wyświetlacz LCD za pomocą czterech tulei dystansowych oraz przylutować jego złącza. Warto lutowanie zostawić na sam koniec, po dokładnym przykręceniu, aby ewentualny naddatek wysokości tulei mógł być skompensowany przez lekkie wystawienie złączy z powierzchni płytki, lecz tak, aby część męska i żeńska wchodziły w siebie całkowicie. Rozmieszczenie przycisków i wyświetlacza na powierzchni płytki znajduje się na **rysunku 4**, dzięki niemu łatwiej będzie wykonać obudowę dla tego urządzenia.

Na etapie uruchamiania konieczne jest zaprogramowanie pamięci Flash mikrokontrolera dostarczonym wsadem oraz zmiana jego bitów zabezpieczających na takie wartości:

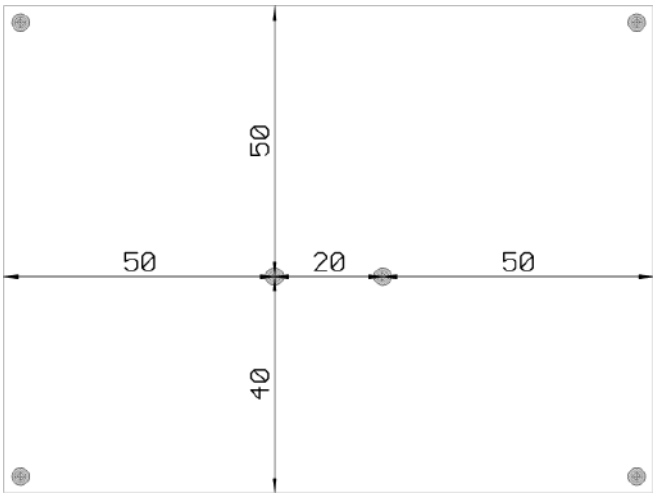
```
Low Fuse = 0x24
High Fuse = 0xD9
```

Szczegóły są widoczne na **rysunku 5**, który zawiera widok okna konfiguracji tychże bitów z programu BitBurner. W ten sposób zostanie uruchomiony wewnętrzny generator o częstotliwości wyjściowej 8 MHz oraz Brown-Out Detector, który wprowadzi mikrokontroler w stan zerowania, jeżeli jego napięcie zasilające spadnie poniżej 4 V. To znacznie zmniejsza ryzyko zawieszenia się układu podczas uruchamiania

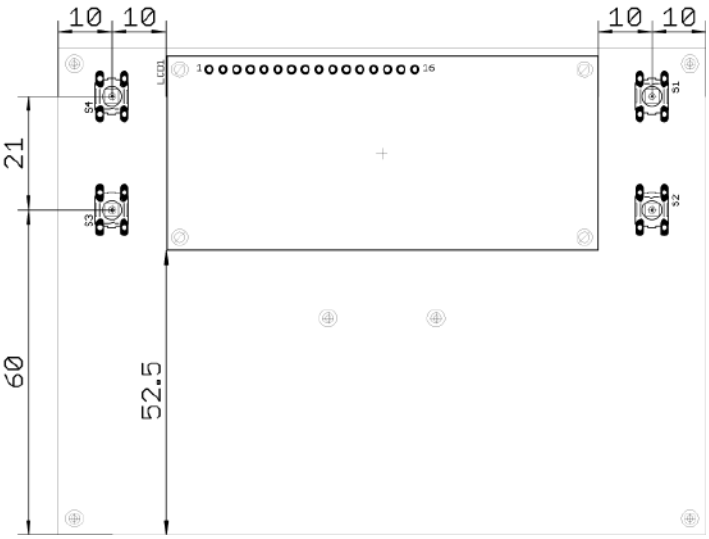


Rysunek 2. Schemat montażowy i wzór ścieżek płytki

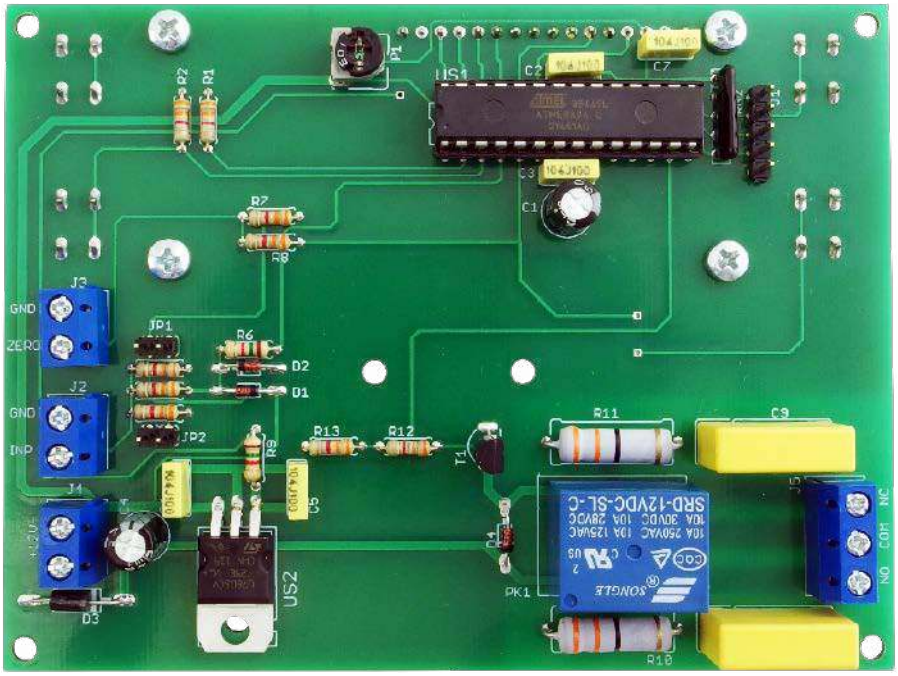
eprasa.pl 1fc8f31a49



Rysunek 3. Położenie otworów montażowych na środku płytki drukowanej



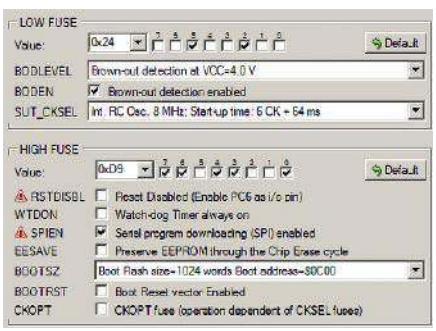
Rysunek 4. Rozmieszczenie przycisków i wyświetlacza LCD na płytce drukowanej



Fotografia 1. Widok zmontowanego układu od strony spodniej (Bottom)

– zwłaszcza wtedy, gdy napięcie zasilające narasta powoli.

Poprawnie zaprogramowany układ jest gotowy do działania po ustawieniu kontrastu wyświetlacza potencjometrem P1. Do zasilania powinno służyć napięcie stałe o wartości około 12 V, a dokładniej z przedziału 10…16 V. Takie granice obejmują zakres prawidłowej pracy przekaźnika (z uwzględnieniem spadku napięcia na przewodzącej diodzie D3 i nasyconym tranzystorze T1) w temperaturze 20°C. Warto, aby owe napięcie było dobrze filtrowane, a jeszcze lepiej stabilizowane. Pobór prądu ze źródła napięcia o wydajności 12 V wynosi około 10 mA przy wyłączonym przekaźniku oraz około 40 mA, kiedy cewka zostanie załączona.



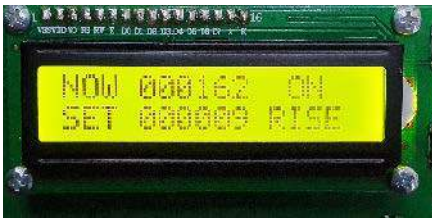
Rysunek 5. Szczegóły ustawienia bitów zabezpieczających

Układ można łatwo przystosować do zasilania napięciem 24 V. Aby to zrobić, w układzie wystarczy dokonać kilku niewielkich modyfikacji. Po pierwsze, przekaźnik PK1 powinien mieć cewkę przystosowaną do napięcia 24 V. Najlepiej pasujący podzespół ma oznaczenie JQC3FF/241ZS z oferty sklepu AVT. Po drugie rezystor R9, który ogranicza prąd pobierany przez diody LED wyświetlacza LCD, powinien mieć rezystancję 2,7 kΩ, aby zachować podobną jasność podświetlenia. Po trzecie, jeżeli obudowa urządzenia miałaby słabą wentylację, polecam do stabilizatora US2 przykręcić radiator, który rozproszy wydzielane w nim ciepło. Wystarczy niewielki radiator przeznaczony do chłodzenia elementów w obudowach typu TO220, na przykład D02A z asortymentu wspomnianego już sklepu AVT.

Z uwagi na szerokość ścieżek łączących wyprowadzenia przekaźnika ze złączem J5, proponuję przyjąć maksymalny prąd płynący przez to złącze na poziomie 8 A. Ścieżki te zostały odsłonięte spod maski lutowniczej, co ułatwia ich pogrubienie, na przykład spoiwem lutowniczym.

Eksploatacja

Widok ekranu działającego urządzenia jest widoczny na **fotografii 2**. W górnym wierszu mamy informację o liczbie zliczonych impulsów, począwszy od wyzerowania lub uruchomienia układu – wartość ta nie jest zapamiętywana po zaniku zasilania. W dolnym wierszu jest wyświetlana progowa liczba impulsów, zadana wcześniej przez użytkownika. Jeżeli liczba NOW jest większa lub równa liczbie SET, to przekaźnik załącza się (w prawym górnym rogu pojawia się ON). W przeciwnym razie pozostanie on wyłączony, co sygnalizuje napis OFF. Wyzerowanie układu wyłącza przekaźnik, gdyż warunek NOW >= SET przestaje

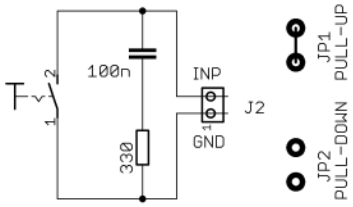


Fotografia 2. Widok wyświetlacza w trakcie normalnej pracy układu

eprasa.pl 1fc8f31a49



Fotografia 3. Sygnalizacja przepełnienia licznika



Rysunek 6. Przykład podłączenia elementu stykowego do wejścia układu

być spełniony. W dolnym lewym rogu mamy informację o aktywnym zboczu sygnału wejściowego. RISE oznacza zliczanie na zboczu narastającym, FALL na opadającym, zaś BOTH na obu zboczach.

W jaki sposób ustawia się te parametry? Wciskając przycisk S4 (SET), zmieniamy aktywne zbocze. Natomiast trzymając S1 (PLUS) lub S2 (MINUS) i wciskając dodatkowo S4 (SET), można zwiększać lub zmniejszać zadaną liczbę impulsów, czyli wartość SET. Im dłużej te przyciski są trzymane, tym szybciej odbywa się przewijanie, dlatego przemiatanie pełnego zakresu może trwać zaledwie kilkanaście sekund. Po upływie około 10 s od ostatniej zmiany wartość ta jest zapamiętywana w nieulotnej pamięci EEPROM, podobnie jak rodzaj aktywnego zbocza.

Pojemność licznika impulsów wynosi 999999. Jeżeli wartość ta zostanie przekroczona, pokaże się napis OVF! (**fotografia 3**) – i układ przestanie liczyć dalej, lecz przekaźnik nie zmieni swojego stanu. Dopiero wyłączenie zasilania lub wyzerowanie układu, czyli wciśnięcie przycisku S3 (ZERO) lub zwarcie zacisków złącza J3, przywraca możliwość liczenia.

Maksymalna częstotliwość zliczania to około 20 kHz, powyżej tej wartości układowi zdarza się gubić impulsy. Należy też pamiętać, że sam przekaźnik ma pewien czas reakcji, rzędu kilku milisekund, które trzeba uwzględnić w działaniu całego systemu. Układ ten powstał z myślą o znacznie wolniejszych przebiegach do zliczania, o częstotliwości rzędu kilku czy kilkunastu herców, gdyż wtedy opóźnienia wynikające z obsługi przekaźnika czy wyświetlacza nie są istotne.

Impulsy wejściowe powinny mieć parametry odpowiadające sygnałom TTL, czyli stan niski w pobliżu potencjału 0 V, zaś stan wysoki o wartości około 5 V. Ten zakres może być szerszy, ewentualny naddatek napięcia zostanie ograniczony przez diody zabezpieczające wejście znajdujące się na zaciskach złącza J2. Z uwagi na moc traconą w rezystorach zaleca się, by wartości szczytowe sygnału wejściowego nie przekraczały ±25 V.

Impedancja wejściowa układu dla sygnałów o parametrach TTL jest bardzo wysoka, rzędu wielu megaomów, poza tym zakresem wynosi około 3,3 kΩ. Można ustalić rezystancję wejściową układu poprzez zastosowanie rezystora pull-up lub pull-down. Ten pierwszy załącza się poprzez nałożenie zworki JP1, zaś drugi zworką JP2. Mogą one być przydatne szczególnie wtedy, gdy chcemy wejście układu sprzęgnąć z elementem elektromechanicznym, na przykład przełącznikiem krańcowym lub ręcznym przyciskiem monostabilnym. Prosty schemat z **rysunku 6** ilustruje konfigurację zworek (JP1 zwarta, JP2 rozwarta) oraz elementy, które warto dodać, by zredukować wpływ iskrzenia styków na zliczanie. Bez tego obwodu RC, który w tym przykładzie składa się z kondensatora 100 nF i rezystora 330 Ω, pojedynczy ruch styków może być odebrany jako kilkanaście lub nawet kilkadziesiąt impulsów zliczonych przez układ.

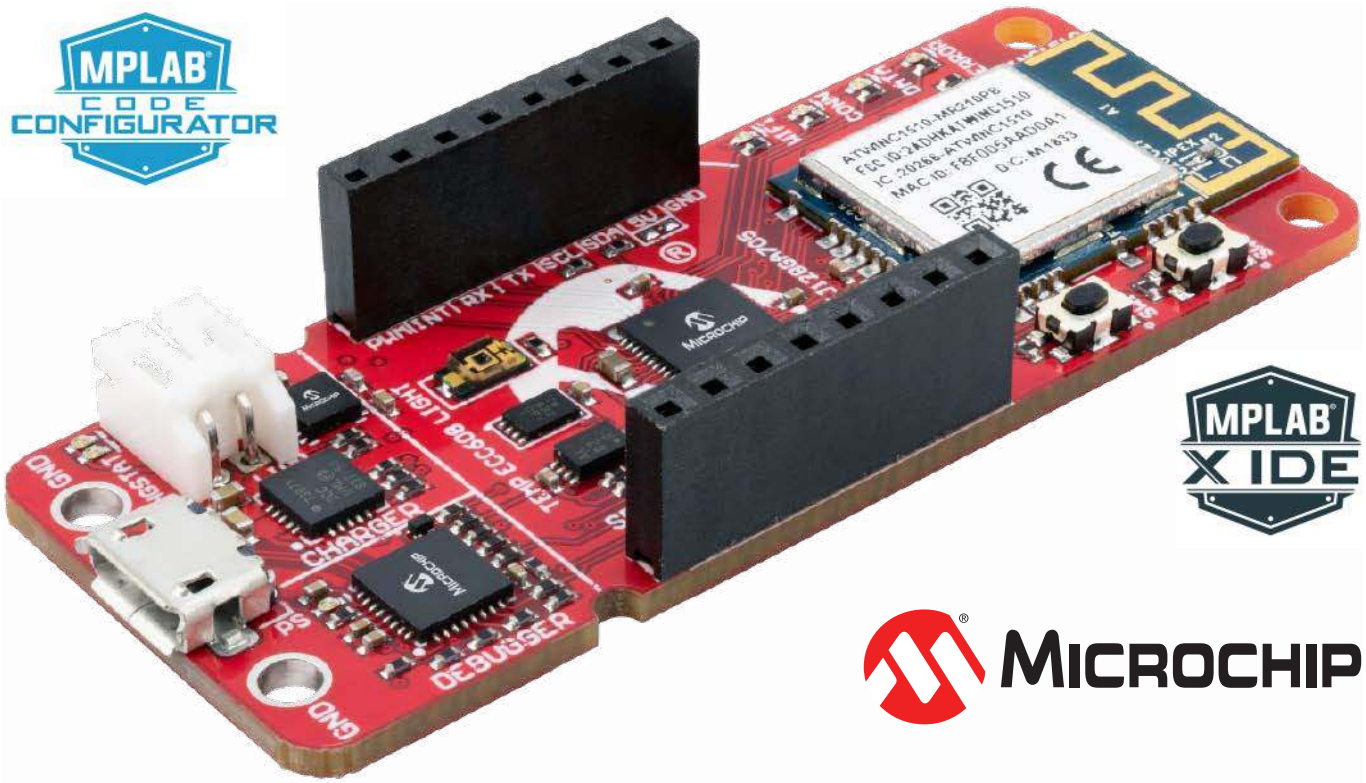
Michał Kurzela, EP

REKLAMA



O projektach, miniprojektach, projektach soft i na wiele innych tematów dyskutuj na forum.ep.com.pl

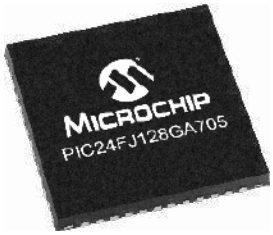
eprasa.pl 1fc8f31a49

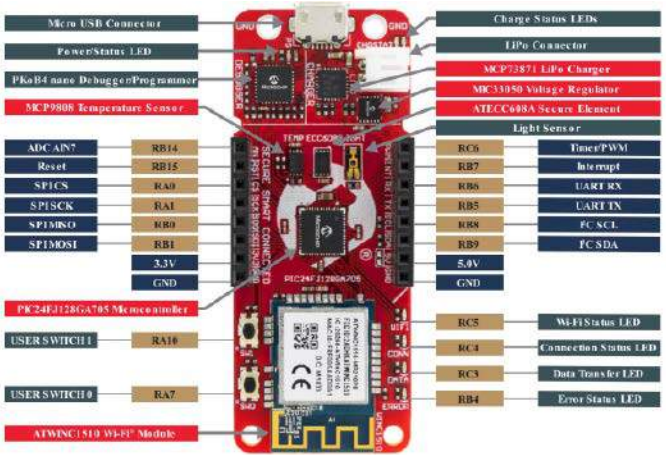


Wygraj płytkę rozwojową Microchip PIC-IoT WG

Płytka rozwojowa PIC-IoT WG łączy w sobie wydajny MCU PIC24FJ128GA705, układ gwarantujący bezpieczeństwo aplikacji ATECC608A CryptoAuthentication oraz w pełni certyfikowany kontroler sieci Wi-Fi ATWINC1510. Taki zestaw zapewnia najprostszy i najskuteczniejszy sposób połączenia wbudowanej aplikacji z Google Cloud IoT.

16-bitowy mikrokontroler PIC24FJ128GA705 oferuje do 128 kB pamięci ECC Flash, 16 kB pamięci RAM i zapewnia wyjątkowo niski pobór mocy. Idealnie nadaje się do zastosowań ogólnych. W pamięci MCU zamontowanego na płytce rozwojowej jest fabrycznie załadowane oprogramowanie sprzętowe, które umożliwia szybkie łączenie się i wysyłanie danych do platformy Google Cloud Platform. Ta przykładowa aplikacja odczytuje dane z wbudowanych czujników: temperatury i światła. Zbudowanie własnego, niestandardowego projektu jest niezwykle proste – można łatwo wygenerować kod, korzystając z bezpłatnych bibliotek oprogramowania w MPLAB Code Configurator (MCC). Płytka zawiera również wbudowany debugger i nie wymaga zewnętrznego sprzętu do programowania i debugowania MCU. Interfejs umożliwia pełną obsługę programowania i debugowania poprzez środowisko MPLAB X IDE. Debugger PKOB nano na komputerze PC jest wykrywany jako urządzenie pamięci masowej umożliwiające





Rysunek 1. Najważniejsze komponenty i układ wyprowadzeń płytki rozwojowej Microchip PIC-IoT WG

łatwe programowanie oraz konfigurację Wi-Fi na zasadzie przeciągnij i upuść, a dodatkowo zapewnia dostęp do interfejsu portu szeregowego – realizuje mostek szeregowy na USB.

Aby mieć szansę na wygranie płytki rozwojowej PIC-IoT WG lub otrzymanie kuponu z 15% rabatem i bezpłatną wysyłką, należy wypełnić formularz zgłoszeniowy na stronie: https://page.microchip.com/E-Prak-WG.html

Więcej informacji na temat płytki Microchip PIC-IoT WG wraz z przykładami projektów można znaleźć na stronie: https://www.microchip.com/en-us/development-tool/ac164164.

eprasa.pl 1fc8f31a49

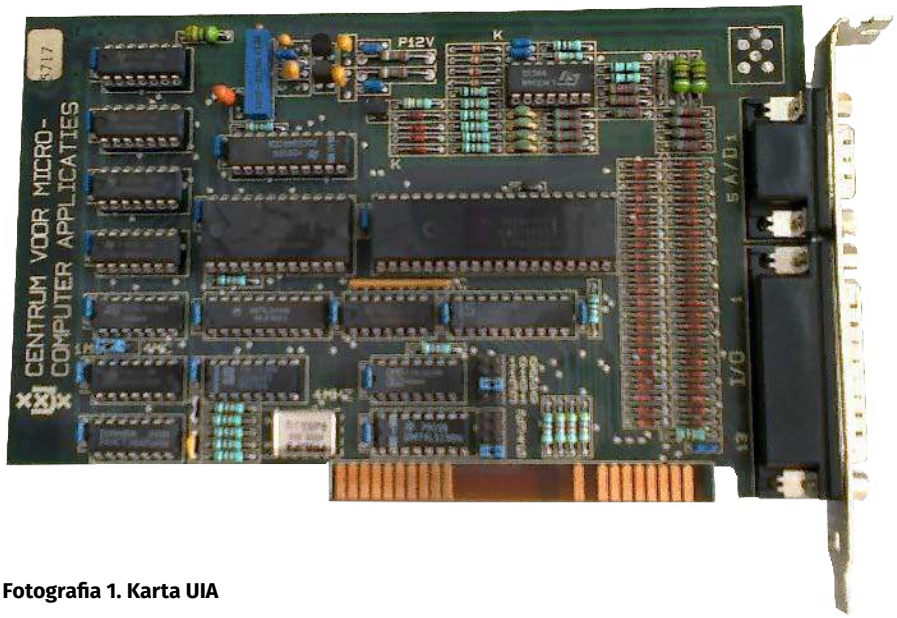
Dział „Projekty Czytelników”. zawiera opisy projektów nadesłanych do redakcji EP przez Czytelników. Redakcja nie bierze odpowiedzialności za prawidłowe działanie opisywanych układów, gdyż nie testujemy ich laboratoryjnie, chociaż sprawdzamy poprawność konstrukcji.
Prosimy o nadsyłanie własnych projektów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

PROJEKT NR 259

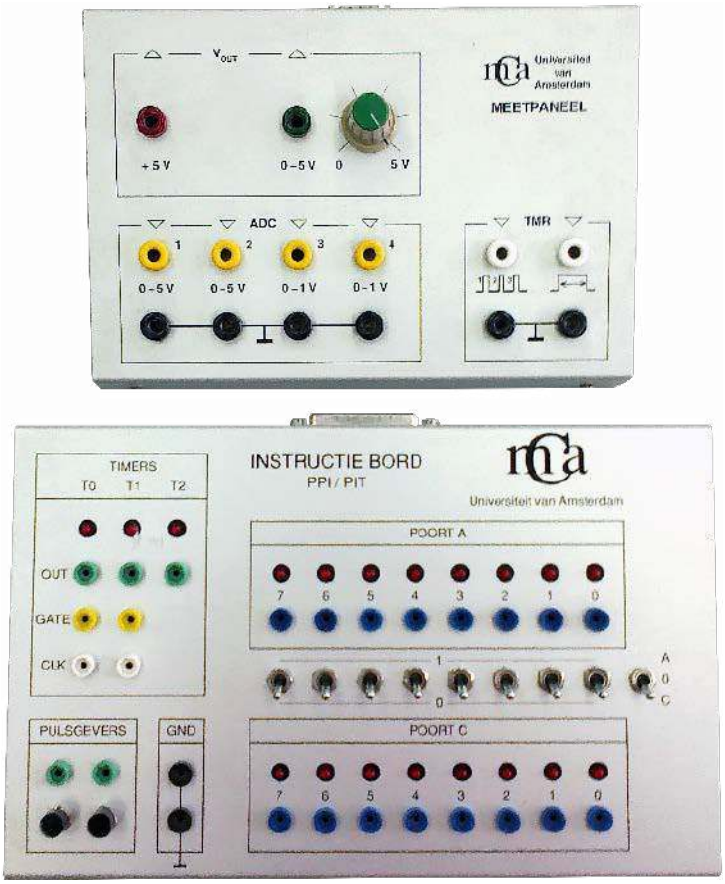
IP-Coach IV. Reaktywacja

Nauczanie w myśl zyskującej coraz większą popularność metodologii STEM wymaga stosowania nowoczesnych narzędzi sprzętowych, dających uczniom i studentom lepszy wgląd w naturę badanych obiektów i zjawisk. Zaprezentowany tutaj projekt stanowi doskonały przykład tego, jak w stosunkowo prosty sposób tchnąć nowe życie w osprzęt dydaktyczny starej daty.

Podstawowe parametry:
- Projekt w formie nakładki typu Arduino Shield, umożliwiający korzystanie z osprzętu IP-Coach bez oryginalnego oprogramowania dla MS-DOS.
- Interfejsy dla czujników analogowych i konsolek systemu dydaktycznego IP Coach2lib.
- Wbudowany przekaźnik do sterowania większych obciążeń.
- Napięcia odniesienia konfigurowane za pomocą zworek.



Fotografia 1. Karta UIA



Fotografia 2. Konsolki pomiarowe IP-Coach

Fundacja CMA (ang. Centre for Microcomputer Applications) (1), założona w 1987 na uniwersytecie w Amsterdamie, zajmuje się opracowywaniem pomocy naukowych, wspomagających zastosowanie mikrokomputerów do akwizycji danych i sterowania eksperymentami (w szczególności z zakresu fizyki, chemii i biologii, zarówno na poziomie szkolnym, jak i uniwersyteckim). W ofercie znajduje się zarówno specjalne oprogramowanie, czujniki i elementy wykonawcze, jak również interfejsy pomiarowe.

Jednym z pierwszych produktów CMA, dostępnych w Polsce na początku lat 90., była seria IP-Coach. System IP-Coach składa się interfejsu pomiarowego UIA bądź UIB, konsolek pomiarowych, czujników oraz wspomnianego wcześniej oprogramowania IPCoach.

Interfejsy UIA/UIB w postaci kart rozszerzeń (**fotografia 1**) montowane były w gniazdach ISA komputerów typu IBM PC/XT/AT. Do gniazd interfejsu podłączane były konsolki pomiarowe, do których następnie wpinało się czujniki pomiarowe. Na **fotografii 2** można zobaczyć konsolkę analogową

eprasa.pl 1fc8f31a49

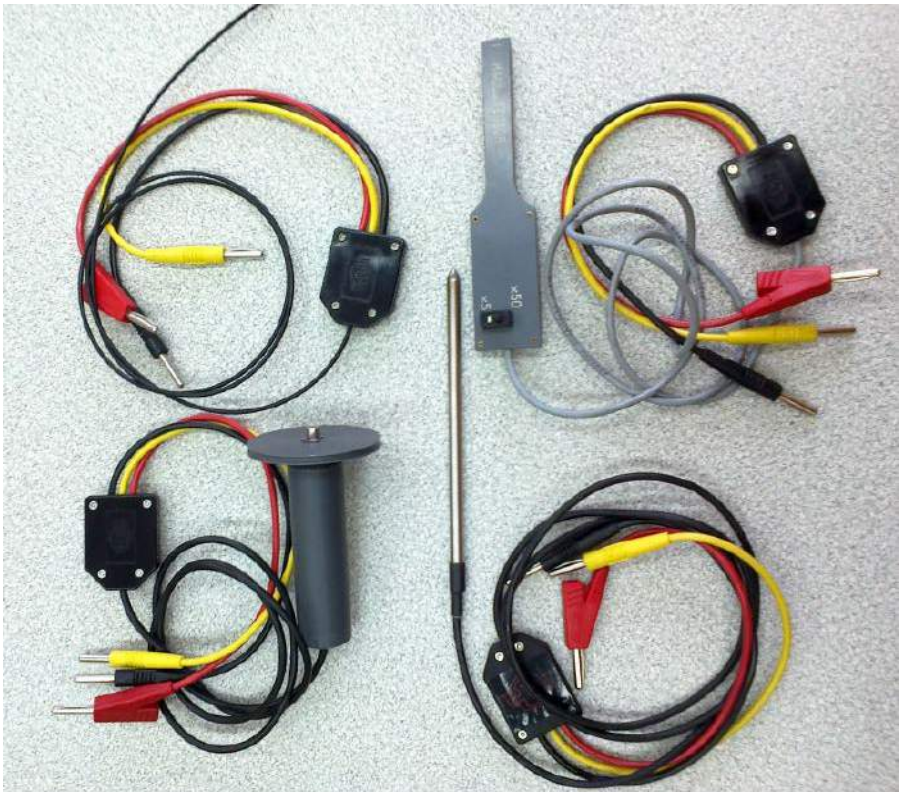
z 4 wejściami kanałów przetwornika ADC oraz konsolkę cyfrową, umożliwiającą obsługę prostych sygnałów binarnych.

Na **fotografii 3** pokazano przykładowe czujniki analogowe, w kolejności od góry: światłowodowy czujnik oświetlenia (w zakresie VIS), czujnik pola magnetycznego (hallotron), czujnik położenia kątowego i sondę temperatury. Czujniki analogowe standardowo wyposażone są we wtyki bananowe w rozmiarze 4 mm. Do zasilania czujników służy napięcie +5 V, podawane na wtyki czerwony i czarny z karty UIA/UIB, zaś sygnał wyjściowy dostępny jest na wtyku żółtym. Czujnik analogowy podłącza się do jednego z wejść ADC konsolki analogowej.

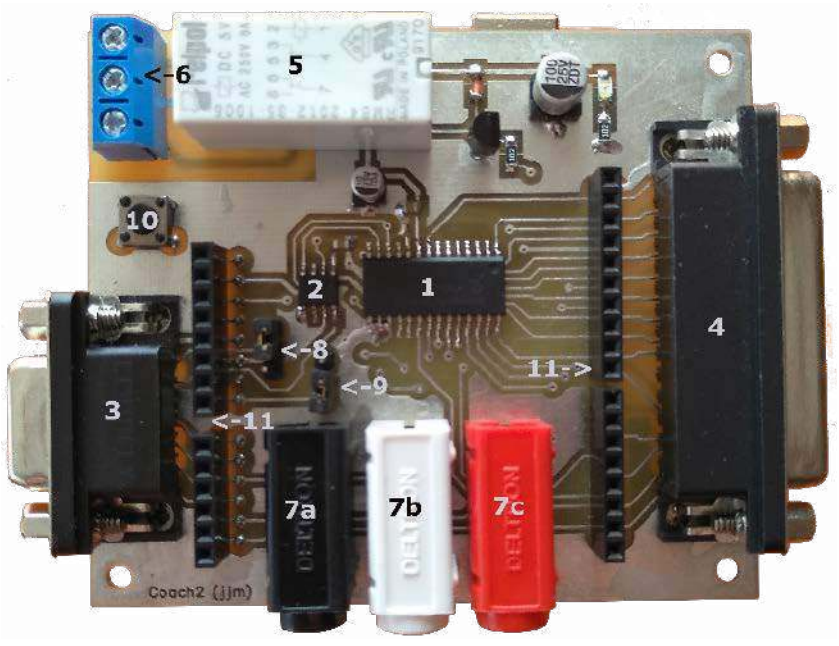
Program do obsługi systemu zawiera moduły, umożliwiające m.in. kalibrację czujników, akwizycję danych pomiarowych i ich wizualizację w postaci wykresów x-y. Dodatkowo pakiet IP-Coach ma wbudowaną możliwość modelowania zjawisk w oparciu o, zapisany za pomocą języka skryptowego, model numeryczny.

Oryginalne oprogramowanie IP-Coach (do wersji 4.0) przeznaczone było do pracy w systemie MS-DOS. Z biegiem lat szyna ISA komputera PC została zastąpiona nowocześniejszymi rozwiązaniami, a interfejsy UIA/UIB oraz dostępne dodatkowe składniki stały się praktycznie bezużyteczne. Podobnie stało się w przypadku oprogramowania IP-Coach – to zostało zastąpione nowocześniejszymi wersjami. W chwili obecnej obowiązującą wersją jest licencjonowane oprogramowanie Coach7, a jako interfejsy pomiarowe dostępne są: Coach Lab++, WiLab, CLAB i VinciLab 2.

Systemy IP-Coach były chętnie używane w fizycznych pracowniach dydaktycznych wielu wydziałów fizyki, w tym w Instytucie Fizyki Doświadczalnej Uniwersytetu Gdańskiego, gdzie – oprócz kart UIA/UIB – ciągle dostępnych jest kilkanaście rodzajów czujników oraz zestawy konsolek pomiarowych.



Fotografia 3. Czujniki analogowe IP-Coach z wtykami 4 mm



Fotografia 4. Nakładka Coach2

IP-Coach i Arduino

W celu umożliwienia ponownego użycia czujników i konsolek ze starego systemu IP-Coach zaprojektowano specjalną nakładkę (shield) do popularnych płytek, zgodnych z ekosystemem Arduino (2) – w szczególności z podstawowym modelem Arduino UNO. Do komunikacji z nakładką i czujnikami przygotowano bibliotekę programistyczną (Coach2lib), która instalowana jest w środowisku Arduino IDE. Widok nakładki zaprezentowano na **fotografii 4,** a jej schemat elektryczny można zobaczyć na **rysunku 1**.

Podstawowym zastosowaniem utworzonego w ten sposób zestawu mogą być samodzielnie wykonywane ćwiczenia, między innymi w pracowni elektronicznej i informatycznej. Co bardzo ważne – nie jest też wymagane użycie licencjonowanego oprogramowania Coach7.

Konstrukcja interfejsu Coach2

Na interfejs nakładki Coach2 składają się dwa gniazda DB9 (J1, numer #3 na fotografii 1) i DB25 (J4, #4) – analogiczne do tych, w które wyposażony jest oryginalny interfejs UIA/UIB – a także gniazda bananowe 4 mm (J5, J6, J7, #7) do czujników analogowych oraz złącze śrubowe ARK (J11, #6) do miniaturowego przekaźnika RM84 (U3, #5).

Do gniazda DB25 doprowadzone są sygnały z portów cyfrowych (PORTA i PORTC) i wielofunkcyjnego portu PORTX. W systemie IP-Coach i na karcie UIA/UIB sygnały cyfrowe sterowane były przez układ I8255, który umożliwiał jedynie sterowanie portami PORTA jako całością i PORTC jako dwiema niezależnymi połówkami. W przypadku płytki Coach2 sygnały cyfrowe sterowane są już przez 16-bitowy ekspander MCP23S17 (U1, #1). Możliwe jest więc niezależne sterowanie każdym bitem portów cyfrowych oraz generowanie przerwań przy zmianie stanu poszczególnych bitów PORTA i PORTC. Układ MCP23S17 komunikuje się z modułem głównym Arduino za pomocą interfejsu SPI.

eprasa.pl 1fc8f31a49

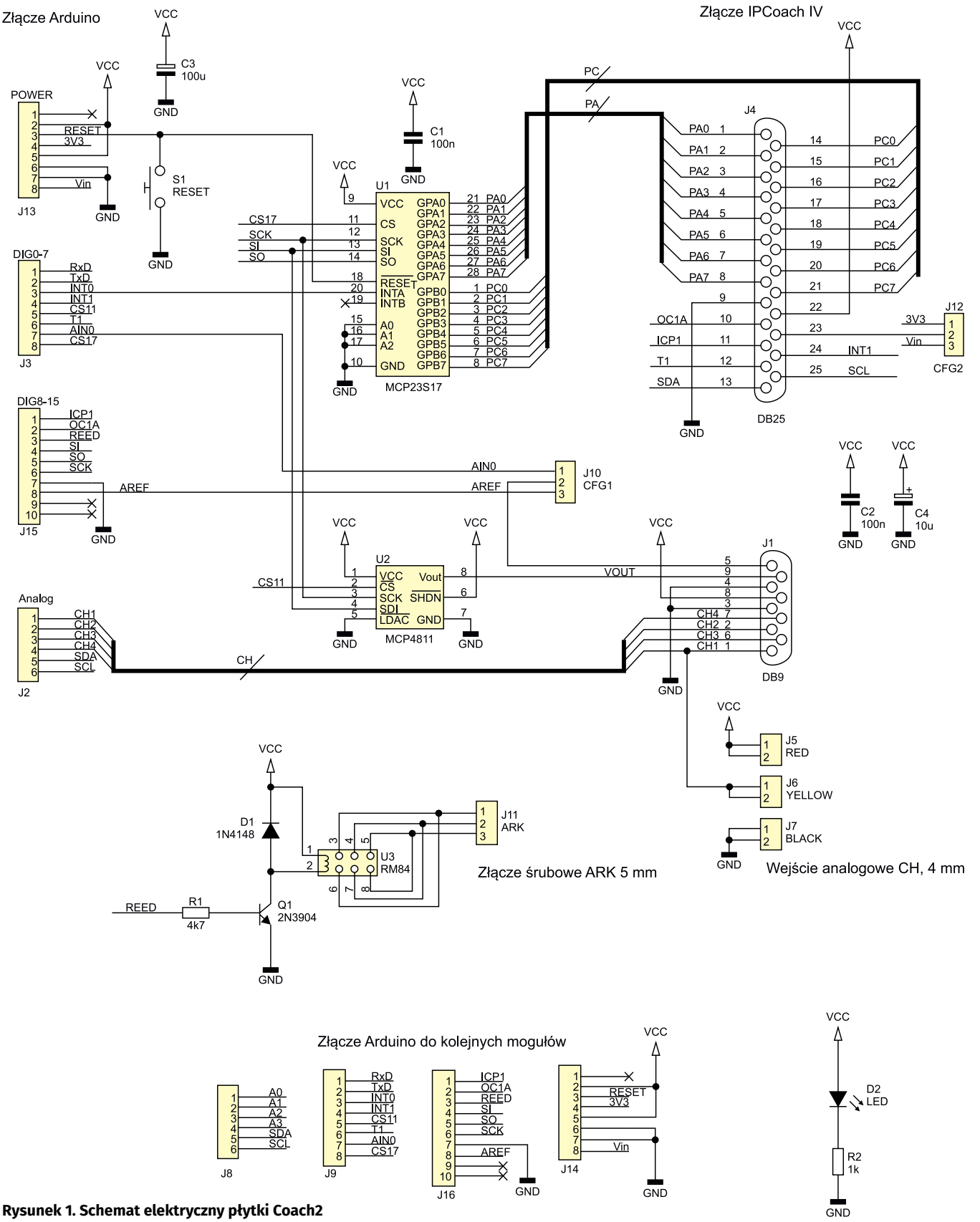
Wykaz elementów:

Półprzewodniki:
D1: 1N4148
Q1: BC847B
U1: MCP23S17
U2: MCP4811

Kondensatory:
C1, C2: 100 nF
C3: 100 µF
C4: 10 µF

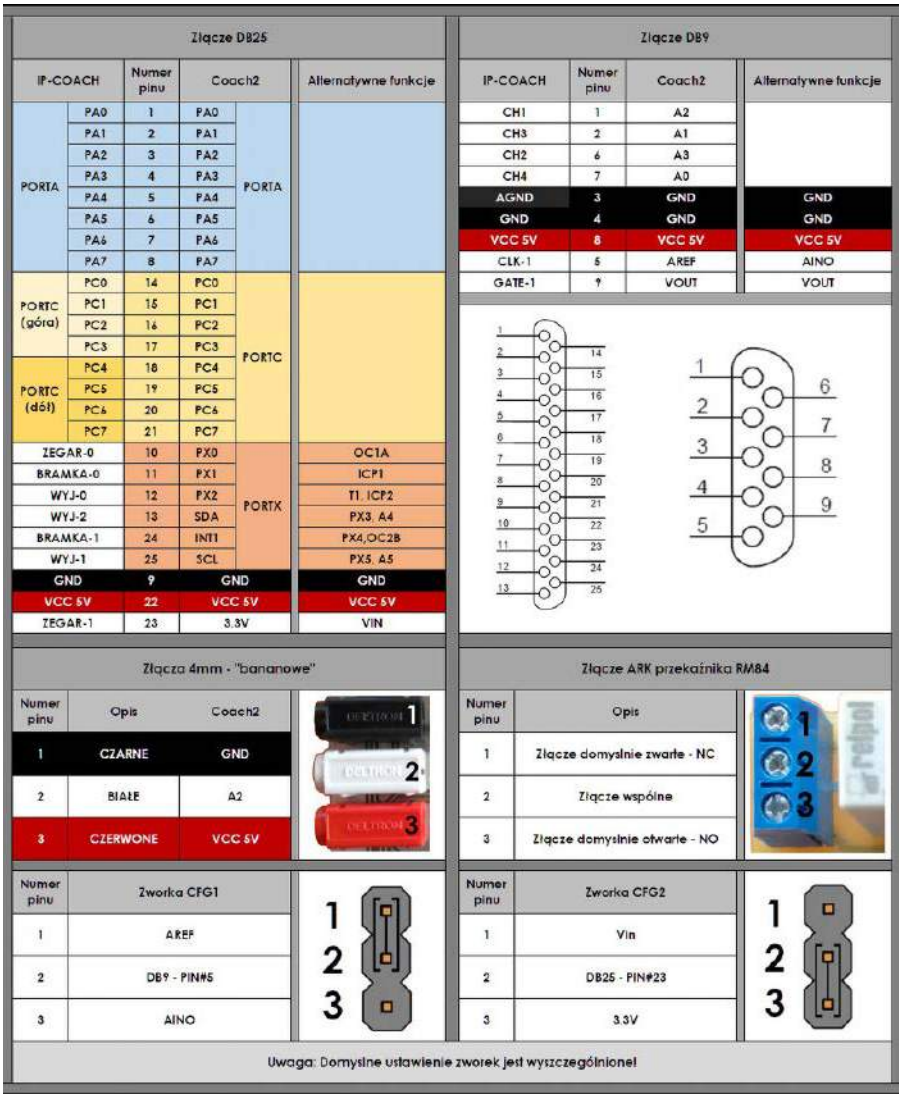
Rezystory:
R1: 4,7 kΩ
R2: 1 kΩ

Pozostałe:
S1: microswitch THT 6x6 mm
U3: przekaźnik RM84
J1: gniazdo DB-9 do druku
J2: gniazdo goldpin 6 pin
J3, J13: gniazdo goldpin 8 pin
J4: gniazdo DB25A do druku
J5: gniazdo bananowe 4 mm (czerwone) DELTRON
J6: gniazdo bananowe 4 mm (żółte lub białe) DELTRON
J7: gniazdo bananowe 4 mm (czarne) DELTRON
J8: wtyk goldpin 6 pin
J9, J14: wtyk goldpin 8 pin
J10, J12: wtyk goldpin 3 pin
J11: gniazdo ARK 5 mm 3-torowe
J15: wtyk goldpin 10 pin
J16: gniazdo goldpin 10 pin



Rysunek 1. Schemat elektryczny płytki Coach2

eprasa.pl 1fc8f31a49



Złącze DB25

IP-COACH		Numer pinu	Coach2		Alternatywne funkcje
PORTA	PA0	1	PA0	PORTA	
	PA1	2	PA1		
	PA2	3	PA2		
	PA3	4	PA3		
	PA4	5	PA4		
	PA5	6	PA5		
	PA6	7	PA6		
	PA7	8	PA7		
PORTC (góra)	PC0	14	PC0	PORTC	
	PC1	15	PC1		
	PC2	16	PC2		
	PC3	17	PC3		
PORTC (dół)	PC4	18	PC4		
	PC5	19	PC5		
	PC6	20	PC6		
	PC7	21	PC7		
ZEGAR-0		10	PX0	PORTX	OC1A
BRAMKA-0		11	PX1		ICP1
WYJ-0		12	PX2		T1, ICP2
WYJ-2		13	SDA		PX3, A4
BRAMKA-1		24	INT1		PX4,OC2B
WYJ-1		25	SCL		PX5, A5
GND		9	GND		GND
VCC 5V		22	VCC 5V		VCC 5V
ZEGAR-1		23	3.3V		VIN

Złącze DB9

IP-COACH	Numer pinu	Coach2	Alternatywne funkcje
CH1	1	A2	
CH3	2	A1	
CH2	6	A3	
CH4	7	A0	
AGND	3	GND	GND
GND	4	GND	GND
VCC 5V	8	VCC 5V	VCC 5V
CLK-1	5	AREF	AIN0
GATE-1	9	VOUT	VOUT

Złącza 4mm - "bananowe"

Numer pinu	Opis	Coach2
1	CZARNE	GND
2	BIAŁE	A2
3	CZERWONE	VCC 5V

Złącze ARK przekaźnika RM84

Numer pinu	Opis
1	Złącze domyślnie zwarte - NC
2	Złącze wspólne
3	Złącze domyślnie otwarte - NO

Numer pinu	Zworka CFG1
1	AREF
2	DB9 - PIN#5
3	AIN0

Numer pinu	Zworka CFG2
1	Vin
2	DB25 - PIN#23
3	3.3V

Uwaga: Domyślne ustawienie zworek jest wyszczególnione!

Rysunek 2. Opis wyprowadzeń z nakładki Coach2

Listing 1. Sterowanie sygnałem PA0 za pomocą programu Arduino

```
#include "Coach2lib.h"
COACH coach;

void setup()
{
        coach.begin();
coach.pinMode(PA0,OUTPUT);
}

void loop()
{
coach.digitalWrite(PA0,HIGH);
delay(100);
coach.digitalWrite(PA0,LOW);
delay(100);
}
```

Listing 2. Sterowanie linią PA0 za pomocą rozkazów SCPI

```
IDN?
CONF:DIG:DIR:PIN PA0,OUTPUT
OUT:DIG:DATA:PIN PA0,HIGH
OUT:DIG:DATA:PIN PA0,LOW
```

Montaż i uruchomienie

Układ należy zmontować na płytce drukowanej, przedstawionej na **rysunku 3**. Prawidłowo złożone urządzenie działa od razu po zakończeniu montażu i nie wymaga specjalnych zabiegów w celu uruchomienia – należy jedynie pamiętać o odpowiednim ustawieniu zworek.

Obsługa interfejsu

W oprogramowaniu sterującym płytką Coach2 przewidziano dwa zestawy procedur sterujących. Pierwszy z nich opiera się na podejściu charakterystycznym dla środowiska Arduino, drugi zaś umożliwia sterowanie płytką zgodnie ze specyfikacją SCPI (3) (ang. Standard Commands for Programmable Instruments).

Ardunio IDE

Klasyczne (dla środowiska Arduino IDE) programowanie wymaga wstępnej instalacji biblioteki do obsługi interfejsu. Bibliotekę stanowią dwa pliki: źródłowy Coach2lib.cpp i nagłówkowy Coach2lib.h. Instalację

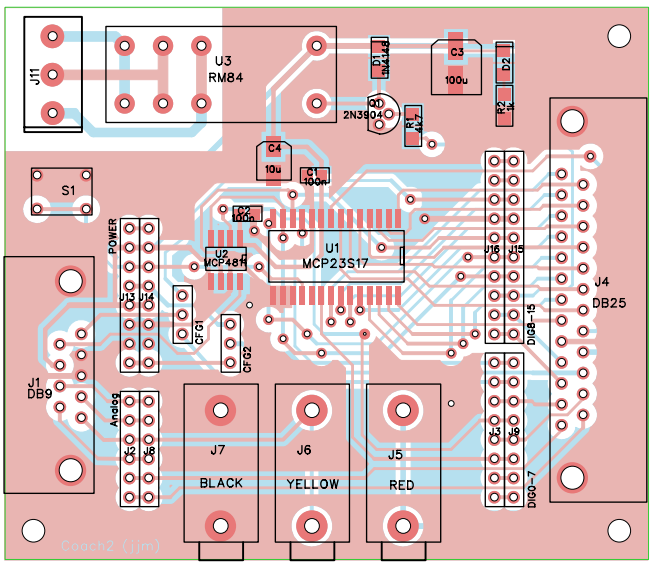
Największe zmiany w stosunku do oryginalnych kart UIA/UIB wprowadzono na wyprowadzeniach #10 – #13 i #23 – #25 gniazda DB25. W IP-Coach sygnały te służyły jedynie jako wejścia/wyjścia liczników opartych na układzie I8254. W Coach2 (z wyjątkiem wyprowadzenia #23) sygnały te podłączone są bezpośrednio do płytki Arduino UNO i stanowią dodatkowy, wielofunkcyjny PORTX. Poszczególne piny mogą służyć jako wejścia/wyjścia (tak jak w przypadku PORTA i PORTC), pozwalają na generowanie sygnału PWM i obsługę bloku TIMER1 procesora płytki Arduino. Po dołączeniu biblioteki Wire wyprowadzenia #13 i #25 mogą służyć do sterowania interfejsem I²C.

Przekaźnik RM84 pozwala na włączanie/wyłączanie urządzeń zasilanych bezpośrednio z sieci energetycznej 230 V.

Do gniazda DB9 doprowadzone są wejścia czterech kanałów (CH1-CH4) 10-bitowego przetwornika ADC procesora Arduino (na kartach UIA/UIB był to 8-bitowy przetwornik ADC084), przy czym zakres mierzonych napięć w każdym kanale zmieniono na 0..5 V. Sygnał z kanału CH1 dołączony jest do dodatkowego gniazda 4 mm, umożliwiając tym samym wpinanie czujników analogowych do nakładki z pominięciem dodatkowych konsolek. Na płytce Coach2 znajduje się również 10-bitowy przetwornik cyfrowo-analogowy MCP4811, (U2, #2), którego wyjście dołączone zostało do pinu #9 złącza DB9. Przetwornik obsługiwany jest przez interfejs SPI i może generować napięcie w zakresie 0…5 V.

Opis dostępnych sygnałów wejścia/wyjścia przedstawiony jest na **rysunku 2**.

Zworki (J10, #8 i J12, #9) służą do konfiguracji napięcia referencyjnego przetwornika analogowo-cyfrowego procesora ATmega. Na płytce znajdują się oczywiście także gniazda szpilkowe wg standardu Arduino (J8, J9, J14, J16) oraz przycisk reset (S1, #10).



Rysunek 3. Widok płytki drukowanej nakładki

eprasa.pl 1fc8f31a49

Listing 3. Plik nagłówkowy Coachlib.h

```
/**
*********************************************************************
*  \file Coach2lib.h
*
* funkcjonalność  powiązana z układem MCP23S17 oparta jest na:
* https://github.com/MajenkoLibraries/MCP23S17
*
* funkcjonalność powiązana z układem MCP4921 oparta jest na:
* https://github.com/michd/Arduino-MCP492X
*********************************************************************/
#ifndef Coach2lib_h
#define Coach2lib_h

#include <Arduino.h>
#include <SPI.h>

//piny Arduino Uno <--> Coach2:
#define INT0   2                                                    //przerwanie z MCP23S17
#define INT1   3                                                    //przerwanie z DB25, pin#24
#define CS11   4                                                    //CS do MCP4921
#define T1     5                                                    //wejście zegarowe do TIMER1 z DB25, pin#12
#define AINO   6                                                    //wejście do Analog Comparator zDB9, pin#5
#define CS17   7                                                    //CS do MCP23S17

#define ICP1   8                                                    //Input capture do TIMER1 z DB25, pin#11
#define OC1A   9                                                    //wyjście z TIMER1 do DB25, pin#10
#define REED  10                                                    //wyjście do przekaźnika
#define AREF  15                                                    //wejście AREF z DB9, pin#5

//Przyporządkowanie we/wy MCP23S17
#define PA0    0                                                    //DB25, pin#1
#define PA1    1                                                    //DB25, pin#2
#define PA2    2                                                    //DB25, pin#3
#define PA3    3                                                    //DB25, pin#4
#define PA4    4                                                    //DB25, pin#5
#define PA5    5                                                    //DB25, pin#6
#define PA6    6                                                    //DB25, pin#7
#define PA7    7                                                    //DB25, pin#8

#define PC0    8                                                    //DB25, pin#14
#define PC1    9                                                    //DB25, pin#15
#define PC2   10                                                    //DB25, pin#16
#define PC3   11                                                    //DB25, pin#17
#define PC4   12                                                    //DB25, pin#18
#define PC5   13                                                    //DB25, pin#19
#define PC6   14                                                    //DB25, pin#20
#define PC7   15                                                    //DB25, pin#21

//Przyporządkowanie pinów cyfrowych PORTX do Arduino UNO
#define PX0    9                                                    //DB25, pin#10 PWM0 OC1A
#define PX1    8                                                    //DB25, pin#11 ICP1
#define PX2    5                                                    //DB25, pin#3  PWM1 T1 ICP2
#define PX3   18                                                    //DB25, pin#4  SDA A4 I²C
#define PX4    3                                                    //DB25, pin#5  INT1 PWM0 OC2B
#define PX5   19                                                    //DB25, pin#6  SCL A5 I²C

//Przyporządkowanie pinów PWM PORTX do Arduino UNO
#define PWM0   9                                                    //DB25, pin#10 PWM0 OC1A
#define PWM1   5                                                    //DB25, pin#3  PWM1 T1 ICP2
#define PWM2   3                                                    //DB25, pin#5  INT1 PWM0 OC2B

//Przyporządkowanie pinów TONE PORTX do Arduino UNO
#define TONE0  9                                                    //DB25, pin#10 PWM0 OC1A
#define TONE1  5                                                    //DB25, pin#3  PWM1 T1 ICP2
#define TONE2  3                                                    //DB25, pin#5  INT1 PWM0 OC2B

//Przyporządkowanie portów
#define PORTA  0
#define PORTC  1

//Przyporządkowanie stanów przekaźnika
#define OFF    0
#define ON     1

class COACH {
  private:
    uint8_t _addr=0;                                                // 3-bit chip address
    uint8_t _reg[22];                                               // kopia 22 rejestrów MCP23S17

    enum {
      IODIRA,  IODIRB,  IPOLA,  IPOLB,
      GPINTENA, GPINTENB, DEFVALA, DEFVALB,
      INTCONA, INTCONB, IOCONA, IOCONB,
      GPPUA, GPPUB, INTFA,  INTFB,
      INTCAPA, INTCAPB,
      GPIOA, GPIOB,
      OLATA, OLATB
    };

    SPISettings _spiSettings;

    uint8_t readRegister(uint8_t addr);
    void writeRegister(uint8_t addr, uint8_t value);
    void readAll(void);
    void writeAll(void);

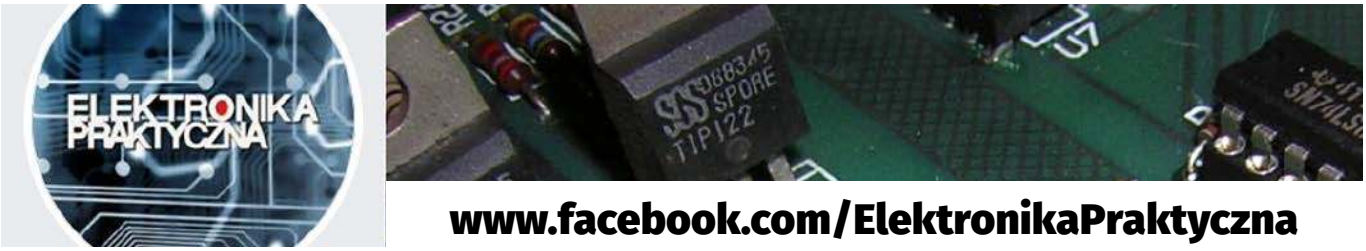
  public:
    COACH(void);

    void begin(void);                                               //CONFIG
    void IDN(void);                                                 //CONFIG
    void analogReference(uint8_t type);                             //CONFIG

    void test();                                                    //DEBUG

    void pinMode(uint8_t pin, uint8_t mode);                        //PORTA, PORTC
    void digitalWrite(uint8_t pin, uint8_t state);                  //PORTA, PORTC
    uint8_t digitalRead(uint8_t pin);                               //PORTA, PORTC
    void portMode(uint8_t port, uint8_t value);                     //PORTA, PORTC
```


eprasa.pl 1fc8f31a49

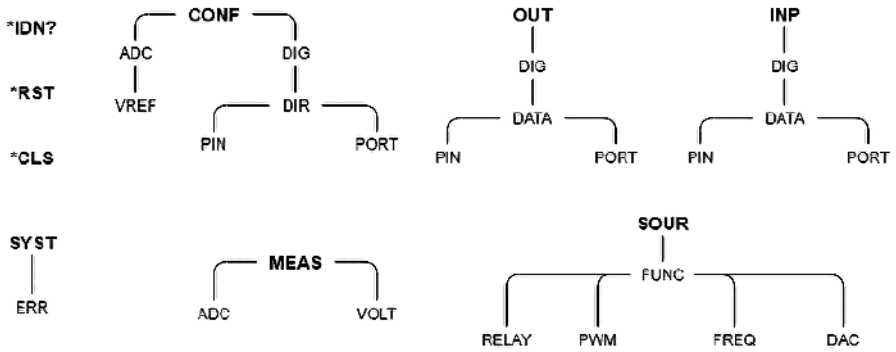


Listing 3. Plik nagłówkowy Coachlib.h – cd.

```
    void portMode(uint16_t value);                                          //PORTA, PORTC
    void writePort(uint8_t port, uint8_t value);                            //PORTA, PORTC
    void writePort(uint16_t value);                                         //PORTA, PORTC
    uint8_t readPort(uint8_t port);                                         //PORTA, PORTC
    uint16_t readPort();                                                    //PORTA, PORTC

    void pinModeEx(uint8_t pin, uint8_t mode);                              //PORTX
    void digitalWriteEx(uint8_t pin, uint8_t state);                        //PORTX
    uint8_t digitalReadEx(uint8_t pin);                                     //PORTX
    void PWM(uint8_t pin, uint16_t value);                                  //PORTX
    void tone(uint8_t pin, uint16_t frequency, uint16_t duration=0);        //PORTX
    void INT(void (*function)(void), uint8_t type);                         //PORTX INTERRUPT – INT1
    void noINT(void);                                                       //PORTX INTERRUPT – INT1

    uint16_t analogRead(uint8_t pin);                                       //DB9, PORTX
    float analogReadVolt(uint8_t pin);                                      //DB9, PORTX
    void DAC(uint16_t value);                                               //DB9
    void relay(uint8_t state);                                              //DB9
};
#endif
```



Rysunek 4. Drzewo rozkazów SCPI

biblioteki przeprowadza się bezpośrednio ze środowiska Aduino IDE.

W tworzonym programie należy dołączyć plik nagłówkowy i zdefiniować obiekt COACH. Po inicjacji obiektu (w funkcji setup()), w głównej pętli programu (w funkcji loop()), za pomocą wybranych metod obiektu COACH, steruje się poszczególnymi sygnałami interfejsu. Komunikacja z komputerem sterującym odbywa się za pomocą portu USB płytki Arduino. Port ten emuluje port szeregowy z parametrami domyślnie ustawionymi na 19200, 8, NP, 1.

Przykładowy program (**listing 1**) inicjuje nakładkę, ustawia pin PA0 jako wyjście, a następnie w pętli zmienia jego stan z HIGH na LOW.

SCPI

Alternatywnym sposobem wykorzystania interfejsu Coach2 jest użycie standardu SCPI (zgodnie ze specyfikacją IEEE 488.2). W przeciwieństwie do programowania w środowisku Arduino IDE, od użytkownika nie jest wymagane samodzielne pisanie programu sterującego. Komunikację z płytką można zrealizować przy użyciu dowolnego programu do obsługi terminalu szeregowego, np. Hyperterminal, PuTTY czy choćby monitora portu szeregowego, dostępnego w środowisku Arduino IDE.

Obsługa odbywa się poprzez interakcyjne przesyłanie komunikatów, zapisanych za pomocą ciągów znaków ASCII. Przykładowa sekwencja rozkazów SCPI sprawdza podłączenie nakładki, ustawia pin PA0 jako wyjście a następnie zmienia jego stan na HIGH, po czym na LOW (**listing 2**).

Przed użyciem standardu SCPI w płytkę należy jednorazowo wgrać szkic SCPICoach2.ino. Na **rysunku 4** można zobaczyć drzewo dostępnych rozkazów SCPI. Obsługę komunikacji i parsowanie rozkazów zapewnia ogólnodostępna biblioteka Vrekrer_scpi_parser(4).

dr inż. Janusz J. Młodzianowski
Wydział Matematyki, Fizyki i Informatyki,
Uniwersytet Gdański

Oprogramowanie płytki Coach2 zostało opracowane w ramach pracy licencjackiej studenta kierunku fizyka, pana Michała Mazurka, na Wydziale Matematyki, Fizyki i Informatyki Uniwersytetu Gdańskiego.

Literatura

1. Fundacja CMA: https://cma-science.nl
2. Arduino: https:/www.arduino.cc
3. Standard SCPI: https://tiny.pl/c6zpv
4. Parser SCPI: https://tiny.pl/c6zlh

REKLAMA



eprasa.pl 1fc8f31a49

Podstawowe parametry:
- moduł zawiera scalony bufor I²C/SMBus typu TCA9511A, który jest „przezroczysty" dla transmisji danych,
- zapewnia poprawę parametrów elektrycznych sygnału i oddzielenie segmentów magistrali,
- wymaga zasilania napięciem 2,7...5,5 V.

Dodatkowe materiały do pobrania ze strony www.ulubionykiosk.pl/media

AVT5912	Expander wyjść z interfejsem I²C (EP 1/2022)
AVT5901	Moduł z zegarem RTC i pamięcią FRAM po I²C (EP 11/2021)
AVT5894	Ośmiokanałowy multiplekser magistrali I²C (EP 10/2021)
AVT5884	Expander portu szeregowego UART dla magistrali I²C (EP 9/2021)
AVT5881	Multiplekser analogowy sterowany z magistrali I²C (EP 8/2021)
AVT5854	Przedłużacz magistrali I²C (EP 4/2021)

* **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]** – płytka drukowana bez elementów i dokumentacji.

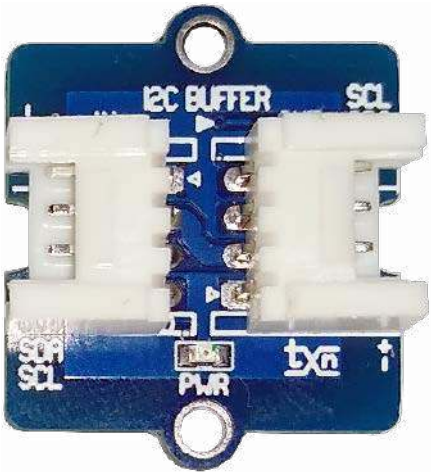
Kity, w których występuje układ scalony wymagający zaprogramowania, mają następujące dodatkowe wersje:
- **wersja [A+]** – płytka drukowana **[A]** + zaprogramowany układ **[UK]** i dokumentacja,
- **wersja [UK]** – zaprogramowany układ.

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 stronie sklepu osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: kity@avt.pl.

Bufor I²C Grove

Gdy długość magistrali I²C zwiększa się, mogą pojawić się problemy z niezawodnością transmisji. Zwiększone pojemności obciążają linie zegara i danych, zwiększając czasy narastania sygnału, co może powodować niepoprawne wykrywanie stanu wysokiego. Aby zwiększyć zasięg magistrali, można użyć bufora I²C na bazie układu TCA9511 A.



Budowa i działanie

Schemat modułu został pokazany na **rysunku 1**. Układ zawiera scalony bufor I²C/SMBus typu TCA9511A, który jest „przezroczysty" dla transmisji danych, ale zapewnia poprawę parametrów elektrycznych sygnału i oddzielenie segmentów magistrali. Moduł jest typową aplikacją TCA9511A. I²C wraz z zasilaniem 2,7...5,5 V doprowadzone jest do gniazda I²CI zgodnego z Grove, buforowany sygnał dostępny jest na gnieździe I²CO. Dioda LD sygnalizuje obecność zasilania, rezystory R2...R5 polaryzują segmenty magistrali.

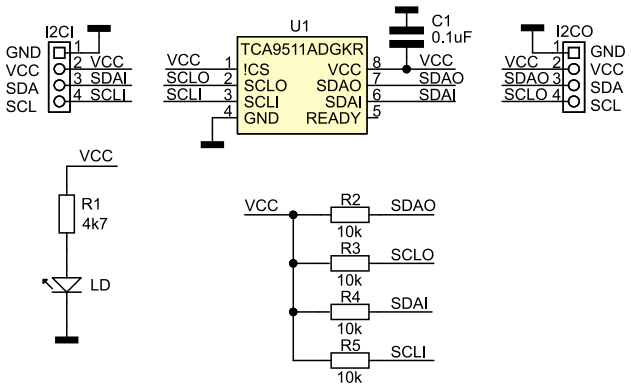
Moduł instalowany jest na magistrali w miejscu, gdzie pojemności zagrażają poprawnej jej pracy. Magistrala zostaje podzielona na segmenty, a dodatkowo każde z wyprowadzeń I²C U1 ma akceleratory poprawiające szybkość zboczy narastających zapewniające niezawodność przy zwiększonym zasięgu transmisji lub większej liczbie układów na magistrali.

Dodatkowym zastosowaniem bufora może być tworzenie magistral typu gwiazda, co czasem upraszcza okablowanie. W tym przypadku wejścia kilku buforów połączone są z masterem magistrali, a wyjścia buforów stają się osobnymi segmentami I²C. Ze względu na kumulujący się offset oraz opóźnienia propagacji nie jest zalecane łączenie więcej niż dwóch buforów szeregowo.

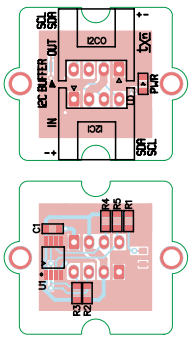
Montaż i uruchomienie

Moduł zmontowany jest na miniaturowej dwustronnej płytce drukowanej mechanicznie zgodnej z Grove. Schemat płytki został pokazany na **rysunku 2**. Montaż jest typowy i nie wymaga opisu, a sam układ nie wymaga uruchamiania, pozostaje tylko zastosowanie bufora we własnej aplikacji.

Adam Tatuś, EP



Rysunek 1. Schemat ideowy układu translatora



Rysunek 2. Schemat płytki PCB

Wykaz elementów:

Rezystory: (SMD0603, 1%)
R1: 4,7 kΩ
R2, R3, R4, R5: 10 kΩ

Kondensatory:
C1: 0,1 µF, 10 V (SMD0603)

Półprzewodniki:
LD: Led czerwona (SMD0603)
U1: TCA9511ADGKR (VSSOP8_065)

Pozostałe:
I²CI, I²CO: złącze Grove kątowe (110990037)

REKLAMA

EP.com.pl
Strona z mnóstwem doskonałych projektów

eprasa.pl 1fc8f31a49

W ofercie AVT*
AVT6021

Podstawowe parametry:
- bazuje na układzie DS2482-100,
- zasilanie magistrali zabezpieczone bezpiecznikiem polimerowym,
- wyposażony w opcjonalny obwód aktywnego podciągania magistrali 1-Wire,
- pracuje poprawnie przy zasilaniu 3,3 lub 5 V.

Dodatkowe materiały do pobrania ze strony www.ulubionykiosk.pl/media
- ---- Uniwersalny translator poziomów I²C (EP 7/2023)
- ---- Graficzny wyświetlacz OLED 2,3" z interfejsem I²C (EP 5/2023)
- ---- Wyświetlacz segmentowy LCD z interfejsem I²C na bazie układu PCF8577C (EP 4/2023)
- ---- Dwukanałowy czujnik temperatury z interfejsem I²C (EP 2/2023)
- ---- Uniwersalny adapter I²C (EP 1/2023)
- ---- Licznik impulsów z podtrzymaniem bateryjnym (EP 12/2022)
- ---- Ekspander DAC z układem LTC1663 (EP 10/2022)
- ---- Przedłużacz magistrali I²C (EP 9/2022)

* **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]** – płytka drukowana bez elementów i dokumentacji.

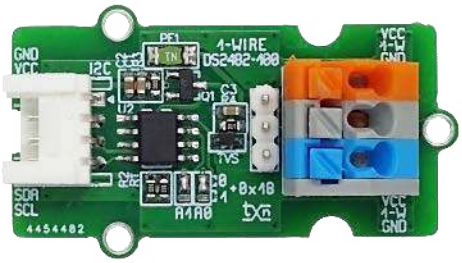
Kity, w których występuje układ scalony wymagający zaprogramowania, mają następujące dodatkowe wersje:
- **wersja [A+]** – płytka drukowana **[A]** + zaprogramowany układ **[UK]** i dokumentacja,
- **wersja [UK]** – zaprogramowany układ.

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 stronie sklepu osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: **kity@avt.pl**.

Mostek master I²C/1-Wire w standardzie Grove

Magistrala 1-Wire, dzięki termometrom DS18B20 i pastylkom identyfikacyjnym, utrzymuje popularność do dziś. zaprezentowany mostek umożliwia podłączenie magistrali 1-Wire do urządzenia wyposażonego w interfejs I²C. Korzystając z OWFS (One Wire File System) w prosty sposób można używać komponentów 1-Wire dołączonych do komputerków SBC, w tym oczywiście Raspberry Pi.



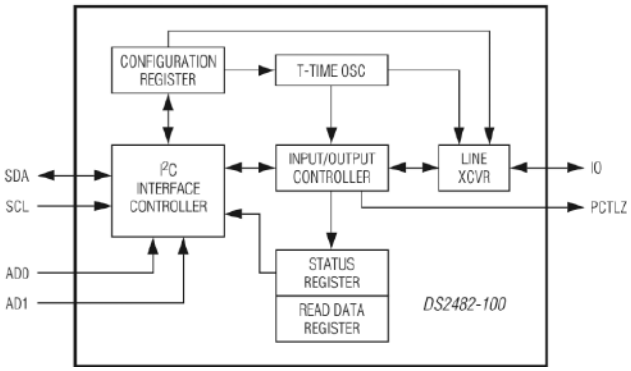
Mostek bazuje na układzie DS2482-100 firmy Analog Devices (dawniej Maxim, a jeszcze dawniej Dallas), którego budowę pokazano na **rysunku 1**. Układ zawiera kontroler Master 1-Wire współpracujący z magistralą I²C oraz interfejs linii 1W z możliwością programowego aktywnego sterowania podciąganiem magistrali.

Budowa i działanie

Schemat mostka został pokazany na **rysunku 2**. Specjalizowany układ U1 wymaga jedynie odsprzęgania zasilania i elementów zabezpieczających linie 1-Wire. Pracuje poprawnie przy zasilaniu 3,3 lub 5 V.

Magistrala I²C oraz zasilanie z systemu SBC doprowadzone są do złącza I²C, rezystory R1, R2 podciągają szyny magistrali. 1-Wire wyprowadzono poprzez rezystor szeregowy R4 do złącza szpilkowego W1 oraz sprężynowego W2. Dioda TVS, zabezpiecza magistralę przed skutkami przepięć. Zasilanie magistrali jest zabezpieczone bezpiecznikiem polimerowym PF1.

Układ U1 ma możliwość adresacji przy pomocy zwór A0...A1. Domyślnie wszystkie zwory połączone są z masą, co ustawia adres bazowy 0x18. Układ wyposażony jest w opcjonalny obwód aktywnego podciągania z tranzystorem Q1 i rezystorem R3, przydatnym przy współpracy modułu z układami wymagającymi chwilowo większych prądów zasilania (np. EEPROM).

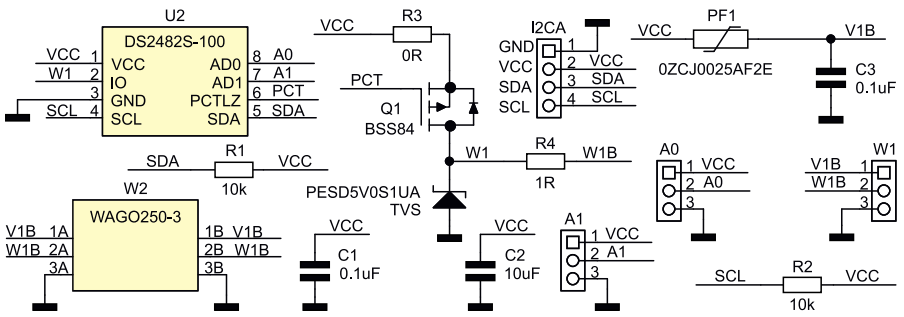


Rysunek 1. Budowa wewnętrzna układu DS2482-100 (za notą AD)

Montaż i uruchomienie

Moduł zmontowany jest na miniaturowej dwustronnej płytce drukowanej mechanicznie zgodnej z Grove, której schemat pokazano na **rysunku 3**. Montaż jest typowy i nie wymaga opisu, a sam układ nie wymaga uruchamiania.

Szybkie sprawdzenie można wykonać przy pomocy Raspberry Pi. Po podłączeniu modułu do magistrali I²C i zasileniu napięciem 3,3 V, używając pakietu i2c-tools, poleceniem `i2cdetect -y 1` sprawdzamy obecność układu na magistrali, powinien być obecny



Rysunek 2. Schemat ideowy mostka I²C/1-Wire

Wykaz elementów:

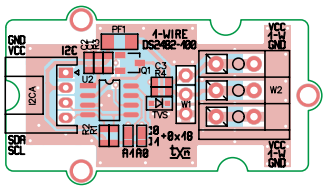
Rezystory: (SMD0603, 5%)
R1, R2: 10 kΩ
R3: 0 Ω (opcja) dobrać dla ograniczenia prądu
R4: 1 Ω

Kondensatory: (SMD0603, 10 V)
C1, C3: 0,1 µF
C2: 10 µF

Półprzewodniki:
Q1: BSS84 (opcja) tranzystor MOSFET
(SOT-23)
U2: DS2482S-100 (SO8)

Pozostałe:
I²CA: złącze grove kątowe (110990037)
PF1: bezpiecznik polimerowy 24 V; 0,5 A; 0,55Ω 0ZCJ0025 AF2E (SMD1206)
TVS: dioda zabezpieczająca PESD5,0S1UAV (SOD323F)
W1: złącze szpilkowe SIP3, raster 2,54 mm
W2: złącze sprężynowe WAGO250-3

eprasa.pl 1fc8f31a49



Rysunek 3. Schemat płytki PCB

pod ustawionym domyślnie adresem 0x18, jak pokazano na **rysunku 4**.

Jeżeli układ jest widoczny, można przejść do instalacji pakietu OWFS (*One Wire File System*, https://www.owfs.org/), który mapuje nam urządzenia 1-Wire na system plików, analogicznie jak podczas mapowania dysków. Przy pomocy OWFS z poziomu systemu operacyjnego mamy dostęp praktycznie do wszystkich parametrów jakie udostępniają układy 1-Wire.

Przed instalacją OWFS warto zaktualizować system:

```
sudo apt-get update
sudo apt-get install owfs
python-ow ow-shell
```

Po instalacji konieczna jest edycja pliku:

```
sudo nano /etc/owfs.conf
```

Następnie należy dodać znak komentarza przy linii:

```
# server: FAKE = DS18S20,DS2405
```

oraz pod sekcją:

```
# USB device: DS9490
#server: usb = all
```

Teraz dodajemy następujący wpis:

```
server: device = /dev/i2c-1
```

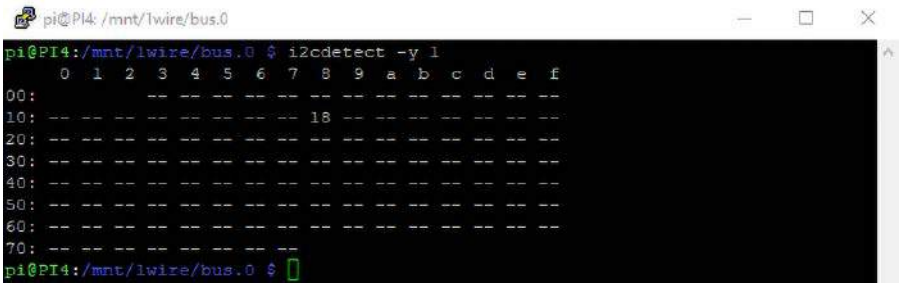
oraz usuwamy komentarz z linii:

```
mountpoint = /mnt/1wire
allow_other
```

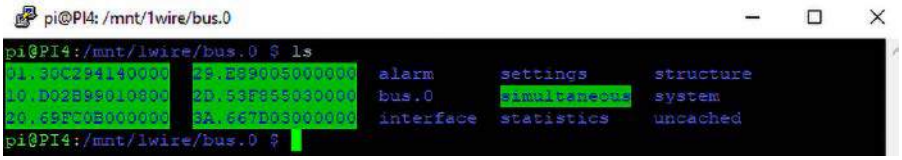
Po zapisaniu zmian, tworzymy katalog w którym montowane będą urządzenie 1-Wire:

```
sudo mkdir /mnt/1wire
```

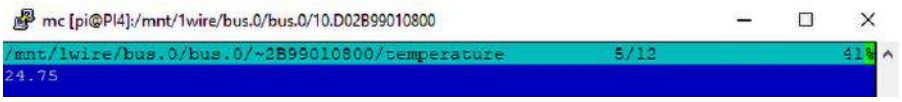
a następnie restartujemy SBC:



Rysunek 4. Detekcja układu DS1482-800 na magistrali I²C



Rysunek 5. Poprawnie zamontowany DS2482-100



Rysunek 6. Przykładowy odczyt temperatury DS1820

```
sudo reboot now
```

Po uruchomieniu aktywujemy serwer:

```
sudo systemctl enable
owserver.service
```

i montujemy urządzenia 1-Wire:

```
sudo owfs -C -uall -m /mnt/1wire
--allow_other
```

Uwaga, czasem urządzenia montowane są dwukrotnie, aby usunąć ten problem edytujemy plik:

```
sudo nano /lib/systemd/system/
owfs.service
```

i zastępujemy wpis:

```
ExecStart=/usr/bin/owfs -c /etc/
owfs.conf --allow_other %t/owfs
```

zmodyfikowanym:

```
ExecStart=/usr/bin/owfs --allow_
other %t/owfs
```

Po ponownym restarcie możemy sprawdzić urządzenia zamontowane w katalogu */mnt/1wire*, a w przypadku mostka DS2482-100 w podkatalogu *bus.0* (**rysunek 5).**

W podkatalogu *bus.0* możemy sprawdzić podłączone do DS2482-100 układy 1-Wire. W podkatalogach zaznaczonych na zielono o nazwie identycznej z unikalnym numerem seryjnym ID układu, zapisane są udostępniane parametry. Przykładowo dla DS1820 o ID 10D02B9901080018 z pliku możemy odczytać zmierzoną temperaturę, co widać na **rysunku 6**. Jeżeli wszystko działa, można zastosować moduł we własnej aplikacji.

Adam Tatuś, EP

REKLAMA

Świat projektantów i programistów
dla elektroniki w nowej odsłonie.
Odwiedź wiecznie młody

ELPORTAL.pl

eprasa.pl 1fc8f31a49

W ofercie AVT*
AVT6022

Podstawowe parametry:
- dwa ośmiobitowe potencjometry cyfrowe o rezystancji 10 kΩ (lub innej, po zmianie układu scalonego),
- dwa jednoobrotowe potencjometry mechaniczne do zadawania położenia ślizgacza w układach cyfrowych,
- regulacja obu kanałów oddzielna lub współbieżna (wybierana zworką)
- dwa tryby działania potencjometru cyfrowego: najniższa pozycja może być odłączana od zacisku wejściowego lub nie, wybierane zworką,
- możliwość dodania rezystorów dla regulacji pseudo-logarytmicznej,
- zasilanie napięciem stałym 3...5 V,
- pobór prądu około 6 mA przy zasilaniu 5 V.

Dodatkowe materiały do pobrania ze strony www.ulubionykiosk.pl/media

----	toneCtrl – regulator barwy dźwięku (EP 9...10/2023)
AVT5975	Regulator barwy dźwięku, głośności i balansu (EP 3/2023)
AVT5873	Stereofoniczny aktywny regulator głośności (EP 8/2021)
AVT5851	7-pasmowy korektor graficzny (EP 4/2021)
AVT5816	Regulator balansu tonów (EP 10/2020)
AVT5637	Wielokanałowy regulator głośności VCA (EP 8/2018)
AVT5629	Cyfrowy regulator głośności z układem PT2257 (EP 6/2018)
AVT3222	Sterowany dowolnym pilotem potencjometr audio z przekaźnikiem (EdW 5/2018)
AVT1979	Korektor barwy dźwięku (EP 11/2017)
AVT1971	Stereofoniczny regulator barwy tonu zasilany z baterii (EP 9/2017)
AVT1972	Potencjometr „Panorama" audio (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]** – płytka drukowana bez elementów i dokumentacji.

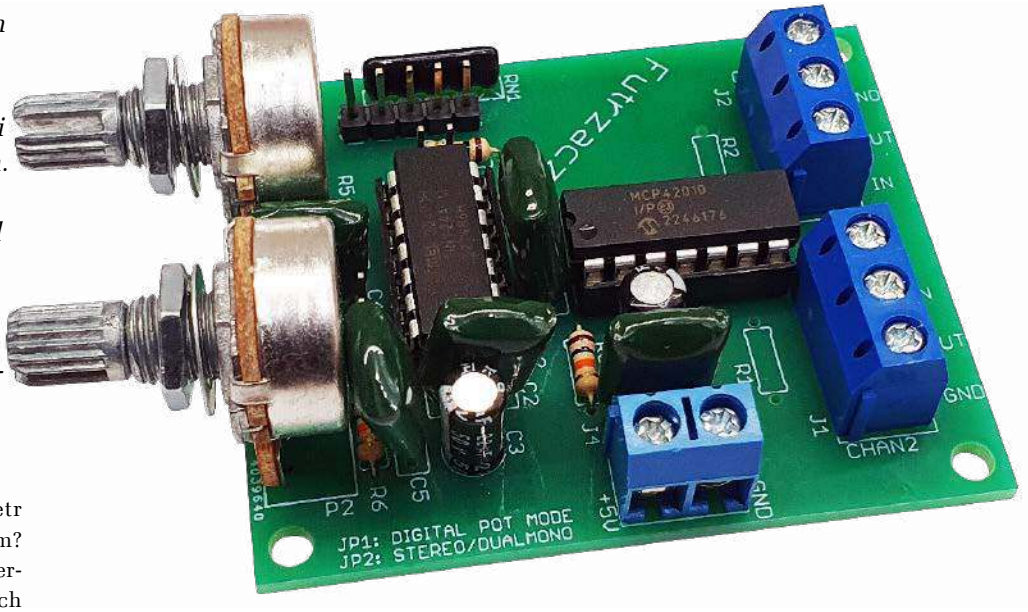
Kity, w których występuje układ scalony wymagający zaprogramowania, mają następujące dodatkowe wersje:
- **wersja [A+]** – płytka drukowana **[A]** + zaprogramowany układ **[UK]** i dokumentacja,
- **wersja [UK]** – zaprogramowany układ.

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 stronie sklepu osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: **kity@avt.pl**.

Mechaniczny potencjometr cyfrowy

Potencjometry zawarte w układach scalonych są bez wątpienia trwałe – nie zużywają się, nie trzeszczą, nie ma ryzyka zabrudzenia ścieżki oporowej kurzem i starym smarem. Z kolei nie każdy lubi regulację enkoderem lub przyciskami, nadal liczne jest grono zwolenników tradycyjnej „gałki" do regulacji głośności lub innych parametrów. Jak pogodzić te dwa światy? Zaprezentowany układ jest odpowiedzią.



W jaki sposób uczynić potencjometr regulacji głośności (niemal) wiecznym? Audiofilskie regulatory wykonane na hermetycznych przekaźnikach sygnałowych są z pewnością trwałe, ale drogie i potrafią mieć niemałe gabaryty. Z kolei zwykłe, jednoobrotowe potencjometry mechaniczne ze ścieżką węglową z czasem się zużywają, choćby były nie wiadomo jak dobrze wykonane. Z kolei cena tych bardzo dobrych jakościowo potrafi powalić na kolana.

Czy można uczynić potencjometr za kilkadziesiąt groszy tak trwałym, jak najdroższe regulatory, a jednocześnie całkiem niedrogim? Owszem! Wystarczy sprzęgnąć go z potencjometrem cyfrowym, który będzie pobierał z tego pierwszego informację o położeniu ślizgacza. Przy okazji, rozrzut między kanałami będzie zdecydowanie mniejszy niż ma to miejsce w typowych potencjometrach stereofonicznych.

Budowa i działanie

Schemat ideowy układu znajduje się na **rysunku 1**. Układ US1 typu MCP42010 to półprzewodnikowy, dwukanałowy, ośmiobitowy potencjometr liniowy o rezystancji ścieżek oporowych wynoszącej 10 kΩ. Steruje się nim poprzez interfejs SPI, którym z kolei zarządza mikrokontroler US2 typu ATtiny24A. Rezystor R1 blokuje wejście sygnału zegarowego magistrali SPI w sytuacji, kiedy mikrokontroler nie nadaje. Rezystory R1 i R2 pozwalają na uzyskanie pseudo-logarytmicznej charakterystyki regulacji zamiast liniowej, (co wyjaśnię w dalszej części artykułu). Złącza J1 i J2 są podłączone do wyprowadzeń potencjometrów i opisane tak, jak gdyby sterowany głośnością:
- IN (wejście) – górny kraniec ścieżki oporowej, dołączany do OUT przy maksymalnym ustawieniu,
- GND (masa) – dolny kraniec ścieżki oporowej, dołączany do OUT przy minimalnym ustawieniu,
- OUT (wyjście) – ślizgacz potencjometru, „przesuwający" się pomiędzy IN a GND.

Wspomniany już mikrokontroler US2 można programować poprzez złącze J3. Wyprowadzono na nie linie interfejsu ISP, przeznaczonego do programowania mikrokontrolerów z rodziny AVR. Rezystory drabinki RN1 podciągają te wyprowadzenia

Wykaz elementów:

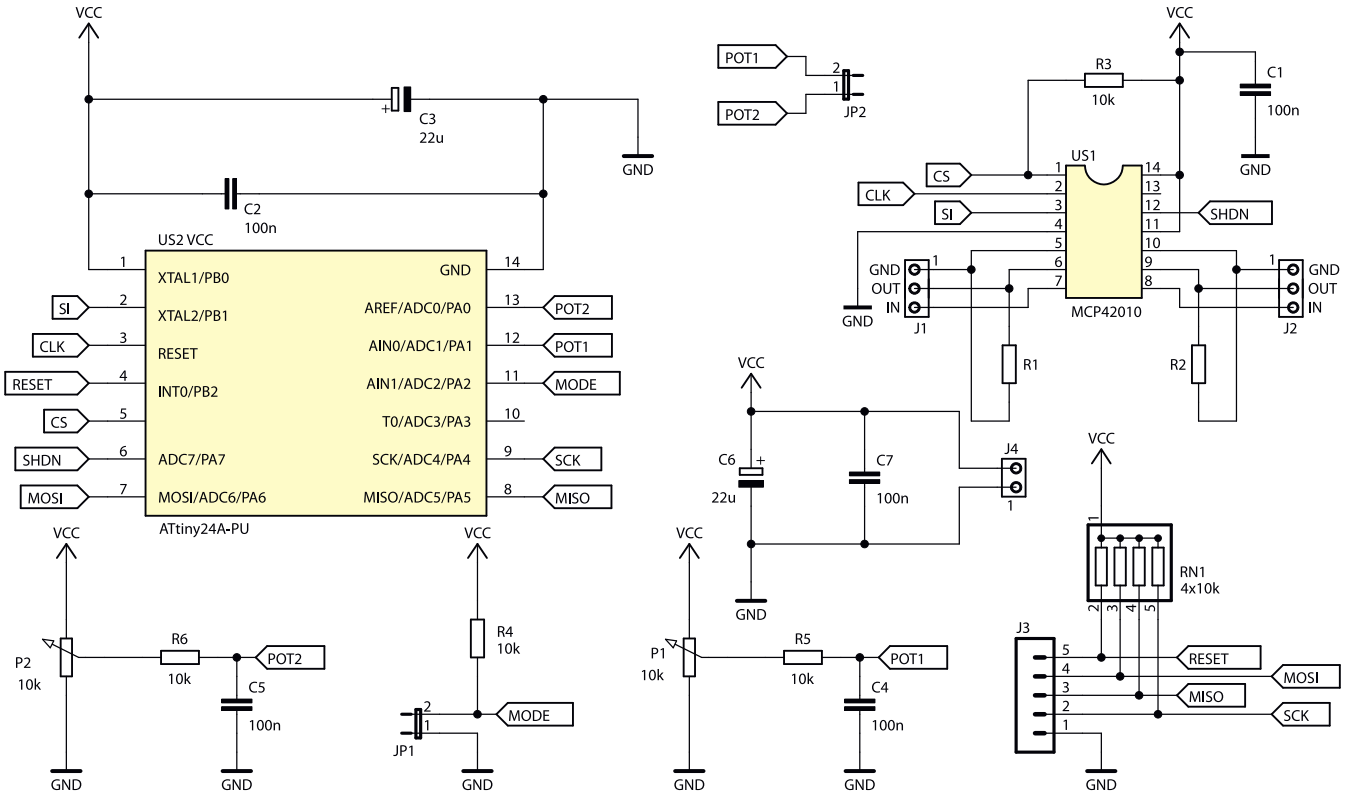
Rezystory: (THT o mocy 0,25 W)
R1, R2: opis w tekście
R3...R6: 10 kΩ
RN1: 4×10 kΩ SIL5
P1, P2: 10 kΩ liniowy, jednoobrotowy do ścianki

Kondensatory:
C1, C2, C4, C5, C7: 100 nF raster 5 mm MKT
C3, C6: 22 µF 25 V raster 2,5 mm

Półprzewodniki:
US1: MCP42010 (DIP14)
US2: ATtiny24A-PU (DIP14)

Pozostałe:
J1, J2: ARK3/500
J3: goldpin 5 pin męski 2,54 mm THT
J4: ARK2/500
JP1, JP2: goldpin 2 pin męski 2,54 mm THT + zworka
Dwie podstawki DIP14

eprasa.pl 1fc8f31a49



Rysunek 1. Schemat ideowy mechanicznego potencjometru cyfrowego

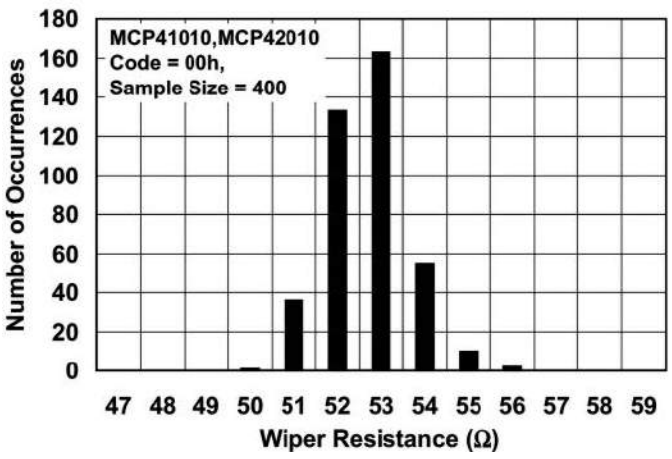
do dodatniego potencjału zasilania, kiedy nie są używane. Kondensatory C2 i C3, ulokowane blisko wyprowadzeń zasilania US2, odsprzęgają jego zasilanie.

Nastawy potencjometru cyfrowego są pobierane z przetwornika analogowo-cyfrowego, który został wbudowany w mikrokontroler. Potencjometry mechaniczne P1 i P2 zostały włączone jako dzielniki napięcia zasilającego, więc na ich ślizgaczach pojawia się (względem masy) napięcie wprost proporcjonalne do kąta skręcenia osi. Obwody RC filtrują to napięcie z zakłóceń, w tym również z trzasków, jakie mogłyby wprowadzać potencjometry mechaniczne po dłuższej eksploatacji. Zawężają również pasmo szumowe, więc na wyjściu przetwornika pojawia się dziesięciobitowa liczba odpowiadająca położeniu ślizgacza każdego z potencjometrów. Ucinając dwa najmłodszebitymożnasterować wprost potencjometrem cyfrowym, bez dodatkowego skalowania. Zworką JP2 można sprawić, by obie sekcje potencjometru cyfrowego regulowały się współbieżnie, więc można nie montować jednegoz potencjometrów mechanicznych i łatwo uzyskać regulacją stereofoniczną.

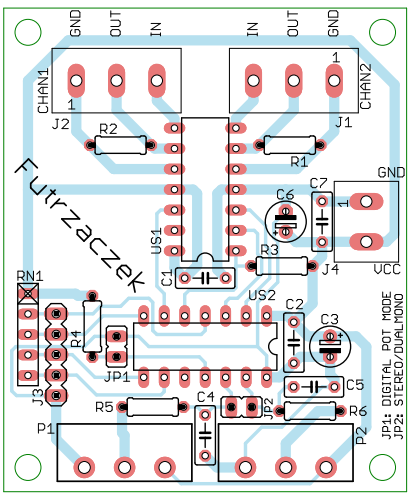
Zworka JP1 służy do ustawiania trybu pracy potencjometru cyfrowego. Jeżeli jest rozwarta, wówczas linia MODE przyjmuje wysoki stan logiczny, co mikrokontroler interpretuje jako konieczność sterowania potencjometrem cyfrowym używając jedynie jego rejestrów: wartość 0 (0x00) odpowiada najniższej pozycji, zaś 255 (0xFF) najwyższej. Jeżeli świat byłby idealny, wówczas ta zworka nie byłaby potrzebna, lecz histogram znajdujący się na **rysunku 2** świadczy o tym, że niestety tak nie jest. Na tymże histogramie przedstawiono liczbę układów spośród badanej grupy 400 sztuk, które miały określoną rezystancję ślizgacza. Widać wyraźnie, że przeważająca większość badanych sztuk znalazła się w przedziale 51…54 Ω przy zasilaniu napięciem 5 V.

Co to oznacza? Po ustawieniu takiego potencjometru cyfrowego na minimum, będzie on nadal „przepuszczał” sygnał. Dokładniej, wprowadzi tłumienie wynoszące około 46 dB – zakładając, że „górna” rezystancja dzielnika napięciowego wyniesie 10 kΩ, zaś „dolna” 53 Ω. To nie jest wiele, bowiem w układach audio spotyka się regulację głośności o dynamice 60 dB lub więcej, chcielibyśmy też całkowitego wyciszenia po ustawieniu gałki regulatora w skrajnie lewej pozycji. Cóż więc można zrobić?

Najprostszym rozwiązaniem, które zostało w tym projekcie zostało zaimplementowane, jest wyłączenie potencjometru cyfrowego. Służy do tego linia SHDN, która w trakcie normalnej pracy znajduje się w wysokim stanie logicznym. Jeżeli jednak zworka JP1 będzie zwarta, wówczas mikrokontroler przełączy SHDN w stan niski po skręceniu obu potencjometrów P1 i P2 na minimum. Wtedy zacisk



Rysunek 2. Częstość występowania układów o określonej rezystancji ślizgacza



Rysunek 3. Schemat płytki PCB

eprasa.pl 1fc8f31a49



Rysunek 4. Szczegóły ustawienia bitów zabezpieczających

IN jest odłączany od ścieżki rezystancyjnej, natomiast OUT i GND są ze sobą zwierane, rezystancją rzędu 50 Ω. Umożliwia to całkowite wyciszenie sygnału audio po skręceniu obu potencjometrów na minimum.

Montaż i uruchomienie

Układ został zmontowany na jednostronnej płytce drukowanej o wymiarach 50×60 mm, której schemat został pokazany na **rysunku 3**. W odległości 3 mm od krawędzi płytki znalazły się cztery otwory montażowe, każdy o średnicy 3,2 mm.

Montaż proponuję rozpocząć od elementów o najmniejszej wysokości obudowy, czyli rezystorów. Pod układy scalone proponuję zastosować podstawki. Na sam koniec polecam wlutować potencjometry P1 i P2 lub złącza śrubowe (np. ARK3/500 z oferty sklepu AVT) którymi można podłączyć te potencjometry zamontowane w innym miejscu, na przykład na panelu czołowym obudowy. Dystans między środkami osi potencjometrów wynosi 30 mm. Gotowy układ można zobaczyć ze szczegółami na fotografii tytułowej.

Na etapie uruchamiania jest konieczne zaprogramowanie pamięci Flash mikrokontrolera dostarczonym wsadem oraz zmiana jego bitów zabezpieczających. Oto ich nowe wartości:

```
Low Fuse = 0xE2
High Fuse = 0xDD
```

Szczegóły są widoczne na **rysunku 4**, który zawiera widok okna konfiguracji tychże bitów z programu BitBurner. W ten sposób zostanie uruchomiony wewnętrzny generator RC o częstotliwości oscylacji 8 MHz oraz Brown-Out Detector, który wprowadzi mikrokontroler w stan zerowania, jeżeli jego napięcie zasilające spadnie poniżej 2,7 V. To znacznie zmniejsza ryzyko zawieszenia się mikrokontrolera podczas uruchamiania.

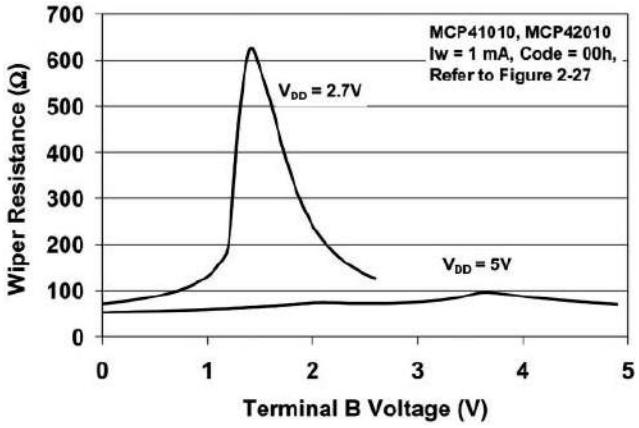
Poprawnie zaprogramowany układ jest od razu gotowy do działania. Zasilać go można napięciem z przedziału 3…5 V, choć polecam wartość wyższą, o czym dalej. Warto, aby owe napięcie było dobrze filtrowane, a najlepiej stabilizowane. Pobór prądu przy 5 V wynosi około 6 mA.

Układ MCP42010 można zasilać napięciem wynoszącym już 2,7 V, lecz nie jest to korzystne z dwóch powodów.

Tabela 1. Rozrzuty rezystancji względem nominalnej oraz między sekcjami

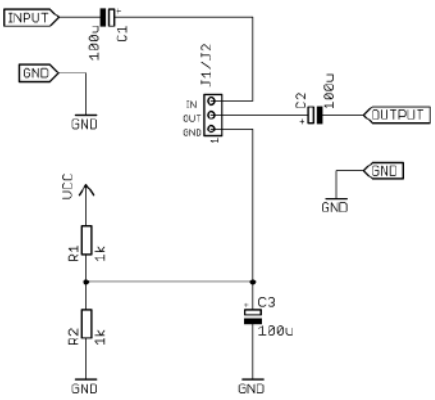
Parameters	Sym	Min	Typ	Max	Units
Rheostat Mode					
Nominal Resistance	R	8	10	12	kΩ
Rheostat Differential Non Linearity	R-DNL	–1	±1/4	+1	LSB
Rheostat Integral Non Linearity	R-INL	–1	±1/4	+1	LSB
Rheostat Tempco	ΔRAB/ΔT	–	800	–	ppm/°C
Wiper Resistance	RW	–	52	100	Ω
	RW	–	73	125	Ω
Wiper Current	IW	–1	–	+1	mA
Nominal Resistance Match	ΔR/R	–	0,2	1	%

Po pierwsze, maksymalne napięcie przyłożone do ślizgacza nie może przekraczać napięcia zasilającego, co ogranicza wartość międzyszczytową sygnału, który chcemy tłumić. Po drugie, rezystancja ślizgacza jest wówczas znacznie wyższa i ulega silnym zmianom w zależności od składowej stałej sygnału – **rysunek 5**. Dlatego, jeżeli tylko zaistnieje taka możliwość, lepiej byłoby użyć napięcia 5 V do zasilania niniejszego układu.



Rysunek 5. Rezystancja ślizgacza w funkcji wartości składowej stałej i napięcia zasilającego

Z potencjometrami cyfrowymi wiążą się dwa istotne ograniczenia. Pierwsze jest takie, że potencjał któregokolwiek zacisku potencjometru nie może zejść poniżej potencjału masy, nie może też wzrosnąć powyżej napięcia zasilającego. Jeżeli chcemy regulować amplitudę przemiennego sygnału analogowego, trzeba mu nadać składową stałą, najlepiej wynoszącą połowę napięcia zasilającego. Przykład takiego podłączenia jest na **rysunku 6**. To tylko jedno z wielu możliwych rozwiązań.



Rysunek 6. Przykład dodania składowej stałej do sygnału analogowego

Drugą wadą, która w wielu przypadkach nie jest bardzo dotkliwa, jest wysoki rozrzut rezystancji ścieżki oporowej (R). Faktyczna rezystancja może ulegać rozrzutom aż o 20% względem nominalnej – **tabela 1**. Natomiast zaletą jest bardzo wysoka współbieżność regulacji, rozrzut między obiema sekcjami nie przekracza 1% (ΔR/R). To bardzo dobry wynik, rzadko spotykany wśród potencjometrów z węglową ścieżką oporową.

Oprócz układów MCP42010, w sprzedaży są dostępne również MCP42050 oraz MCP42100. Mają one inne rezystancje ścieżek oporowych, odpowiednio 50 kΩ i 100 kΩ. Ich zastosowanie nie wymaga jakichkolwiek modyfikacji w układzie.

Układ umożliwia liniową regulację tłumienia przez potencjometr cyfrowy, lecz w niektórych zastosowaniach wymagana jest inna charakterystyka, chociażby logarytmiczna – na przykład w układach audio. Można z dobrym przybliżeniem zrealizować taką charakterystykę poprzez włączenie między zaciski OUT i GND w potencjometrze cyfrowym rezystora stałego. Zostały na to przewidziane rezystory R1 i R2 na schemacie ideowym. Ich wartość można ustalić doświadczalnie, ale moje testy wykazały, że jego rezystancja powinna wynosić 20…25% całkowitej rezystancji ścieżki potencjometru liniowego. Oznacza to, że układ MCP42010 (10 kΩ) wymagałby rezystorów o wartości z przedziału 2…2,5 kΩ. Najlepiej, aby były to niskoszumne rezystory metalizowane, na przykład o rezystancji 2,2 kΩ.

Michał Kurzela, EP

eprasa.pl 1fc8f31a49



Robotyka w medycynie (1)

Roboty chirurgiczne stały się wręcz symbolem współczesnej medycyny. To, co jeszcze nie tak dawno należało raczej do kategorii science fiction – ultranowoczesny manipulator, zdolny do wykonywania najbardziej złożonych operacji z precyzją nieosiągalną dla większości chirurgów – dziś stanowi narzędzie, które można znaleźć w niemal każdym większym szpitalu. Robotyka medyczna to jednak nie tylko chirurgia – istnieje kilka innych obszarów współczesnej terapii, coraz intensywniej korzystających z osiągnięć mechatroniki XXI wieku. W tym numerze „Elektroniki Praktycznej" przyjrzymy się wybranym technologiom, zmieniającym oblicze współczesnej medycyny.

Prawdziwy automat, czy tylko „pomocna dłoń"?

Podstawowa definicja robota zakłada posiadanie przez niego pewnej autonomii, umożliwiającej mu wykonywanie określonych zadań w sposób automatyczny. Przykładowo – słownik internetowy PWN określa robota jako „urządzenie zastępujące człowieka przy wykonywaniu niektórych czynności", a słowo klucz w tym przypadku to „zastępowanie". Od robota medycznego powinniśmy więc oczekiwać samoczynnej realizacji, jeśli nie całego zabiegu, to przynajmniej pewnego zakresu podzadań, składających się na operację. A jednak w praktyce istnieje niewiele przykładów tego typu rozwiązań, które (teoretycznie) byłyby w stanie nie tylko z sukcesem przejść przez niebotycznie wymagający proces certyfikacji i badań klinicznych, ale także… zyskać zaufanie lekarzy, odpowiedzialnych za zdrowie i życie pacjentów.

Okazuje się, że w rzeczywistych warunkach klinicznych taka autonomia wcale nie jest najważniejsza – o wiele istotniejsze jest bowiem odpowiednie wsparcie chirurga, którego doświadczeniu, wiedzy i umiejętnościom jeszcze przez wiele lat nie dorówna żadne urządzenie techniczne. W efekcie opowieści o inteligentnych maszynach, zdolnych do prowadzenia zabiegów operacyjnych bez udziału człowieka (lub tylko pod jego nadzorem), można jak na razie z pobłażliwym uśmiechem włożyć między bajki.

Mało tego – życie zweryfikowało także drugie, bardzo ważne założenie, które przyświecało twórcom pierwszych telemanipulatorów chirurgicznych, stosowanych do wykonywania operacji na odległość (tzw. telechirurgia). Tak naprawdę bowiem – pomimo że współczesne roboty chirurgiczne faktycznie mogą być sterowane za pośrednictwem łączy teleinformatycznych nawet na ogromnym dystansie – w niemal wszystkich aplikacjach konsola sterująca, obsługiwana przez chirurga, znajduje się… co najwyżej kilka metrów od stołu operacyjnego. Dlaczego zatem użycie robotów chirurgicznych nadal ma w ogóle jakikolwiek sens praktyczny? Z jakimi ograniczeniami zmagają się inżynierowie, pracujący nad nowymi modelami robotów medycznych? Na te i inne pytania postaramy się znaleźć odpowiedzi w niniejszym materiale.

Roboty laparoskopowe

U zdecydowanej większości osób pierwsze skojarzenie z hasłem „robot medyczny" odnosi się do systemów przeznaczonych do prowadzenia operacji laparoskopowych. Technika laparoskopowa – dzięki minimalnej inwazyjności, znacząco redukującej zarówno czas hospitalizacji, jak i rekonwalescencji pacjentów – została dość wcześnie zaadaptowana do świata najnowszych technologii medycznych, czego efektem jest powszechność robotów chirurgicznych na całym świecie. Dziś, po 13 latach od instalacji pierwszego w Polsce systemu da Vinci marki Intuitive Surgical (miało to miejsce we wrocławskim Wojewódzkim Szpitalu Specjalistycznym w 2010 roku), polscy pacjenci mogą liczyć na refundację NFZ na niektóre zabiegi w asyście robota (taka możliwość pojawiła się dopiero 1 sierpnia 2023 r.), zaś za granicą operacje robotowe są już w kilku obszarach chirurgii

eprasa.pl 1fc8f31a49



Fotografia 1. Profesor Wojciech Witkiewicz przy jednej z dwóch konsol sterujących robota chirurgicznego 3. generacji – da Vinci Si (https://t.ly/HP_lf)



Fotografia 2. Podstawowy zestaw do chirurgii robotowej, od lewej: wózek z komputerem sterującym i systemem wizyjnym; wózek z ramionami robota; konsola sterująca (https://t.ly/cyb6M)



Fotografia 3. Wózek z ramionami robota – system da Vinci Xi (https://t.ly/7Xo-b)

standardem. Nawiasem mówiąc, wrocławski robot wykonał swoją ostatnią operację 8 grudnia 2023 r. – zaledwie kilka miesięcy po wprowadzeniu refundacji maszyna musiała planowo przerwać służbę z uwagi na zakończenie okresu technicznej przydatności do użytku (**fotografia 1**).

Każdy zestaw do operacji robotowych składa się z trzech głównych elementów: wózka mobilnego z ramionami narzędziowymi, konsoli sterującej (stanowiącej interfejs HMI, obsługiwany przez siedzącego przy nim chirurga) oraz szafy z głównym komputerem, systemem wizyjnym, sterownikiem oświetlacza oraz urządzeniem do elektrokoagulacji (**fotografia 2**). W niektórych przypadkach możliwa jest praca w trybie dwukonsolowym, co niebywale zwiększa możliwości w zakresie treningu, w tym nawet śródoperacyjnego (jedna z konsol pracuje niejako w trybie wirtualnym, podczas gdy druga pełni funkcję nadrzędnego kontrolera, obsługiwanego przez głównego operatora).

Wspomniany już system da Vinci jest pierwszym robotem chirurgicznym, który w takim stopniu rozpowszechnił się w szpitalach na całym świecie – do tej pory za pomocą ponad 7,5 tysiąca maszyn wykonano przeszło 12 milionów operacji w 70 krajach, a materiały Intuitive Surgical wskazują, że średnio co 16,8 sekundy gdzieś na świecie rozpoczyna się kolejny zabieg z użyciem tego ultranowoczesnego sprzętu. Dziś najbardziej zaawansowanym modelem oferowanym przez amerykańskiego producenta jest czteroramienny robot da Vinci Xi (**fotografia 3**), kosztujący w chwili pisania niniejszego artykułu około 13 milionów złotych. Swoją topową pozycję w portfolio Intuitive Surgical da Vinci Xi zawdzięcza m.in. poprawionej ergonomii, bardziej smukłemu designowi ramion oraz szeregowi dodatkowych funkcjonalności (zintegrowane sterowanie stołem operacyjnym, zestawy automatycznych ustawień czy też rozbudowane opcje pozycjonowania wózka mobilnego, zawierającego mechatroniczne ramiona). Tańsza wersja systemu – da Vinci X (**fotografia 4**) – pod wieloma względami nie ustępuje jednak młodszemu bratu, wciąż jest bowiem jednym z najbardziej zaawansowanych robotów medycznych na świecie.

Bardzo interesującym elementem ekosystemu da Vinci pozostaje model da Vinci SP (**fotografia 5**). Robot – w odróżnieniu od opisanych wcześniej modeli X oraz Xi – zawiera tylko jedno ramię, przeznaczone jednak do obsługi aż czterech narzędzi, umieszczonych



Fotografia 4. Robot da Vinci X (https://t.ly/STKkz)

eprasa.pl 1fc8f31a49



Fotografia 5. Robot da Vinci SP (https://t.ly/ogn9r)



Fotografia 6. Głowica narzędziowa robota da Vinci SP – widoczna stereoskopowa kamera z oświetlaczem (na górze) oraz elastyczne ramiona z instrumentami chirurgicznymi (https://t.ly/o0JLk)

wokół wspólnej tulei, wprowadzanej do ciała pacjenta. Laparoskopia z dostępu jednoportowego (określana mianem operacji „przez dziurkę od klucza") znacząco redukuje liczbę blizn na ciele pacjenta (ze standardowych 4 lub 5 do zaledwie jednej), umożliwia ponadto bardziej efektywne operowanie w trudno dostępnych miejscach lub obszarach o niewielkiej objętości (np. w chirurgii szyi). Wymaga to jednak rozwiązania szeregu problemów technicznych. Zamiast czterech osobnych narzędzi, umieszczonych na sztywnych tulejach, potrzebujemy bowiem konstrukcji, z których każda ma dodatkowe stopnie swobody – wszystkie narzędzia, złożone w jedną, równoległą „wiązkę", są bowiem wsuwane do organizmu pacjenta przez wspólny port, a następnie rozkładane tak, by uzyskać standardowe położenia poszczególnych przyrządów (główne narzędzia – prawe i lewe, narzędzie pomocnicze oraz tor wizyjny z miniaturową kamerą i silnym oświetlaczem). Patrząc na fotografie końcówek robota da Vinci SP (**fotografia 6**), trudno nie odnieść wrażenia, że stanowi on spełnienie futurystycznych wizji sprzed lat, kiedy o tego typu operacjach można było czytać jedynie na stronach czasopism popularnonaukowych bądź powieści science fiction.

Diametralnie inne rozwiązanie w zakresie pozycjonowania kamery stereoskopowej zaproponowała firma Vicarious Surgical. W tym przypadku nie mamy już bowiem do czynienia z elastycznym endoskopem, zdolnym do wyginania się w kilku miejscach (niczym wąż) – zamiast tak rozbudowanej konstrukcji, projektanci zatrudnieni w start-upie zdecydowali się pójść po linii najmniejszego oporu, instalując dwie kamery wraz z oświetlaczami LED w specjalnej kapsule (**fotografia 7**), która – po wprowadzeniu do ciała pacjenta przez port laparoskopowy – obraca się o 90°, umożliwiając obrazowanie operowanych



Fotografia 7. Narzędzia robota chirurgicznego marki Vicarious Surgical w stanie umożliwiającym wsunięcie ich do portu laparoskopowego (https://t.ly/uMlSW)

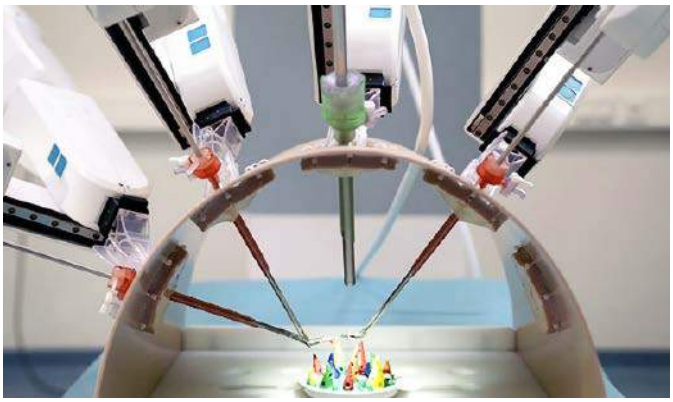


Fotografia 8. Narzędzia robota chirurgicznego marki Vicarious Surgical w stanie gotowości do wykonywania zabiegu(https://t.ly/2N5-W)

tkanek (**fotografia 8**). W przypadku samych ramion zastosowanie konstrukcji wieloprzegubowej było już oczywiście nieodzowne, stąd nie powinno dziwić pewne podobieństwo do opisanego wcześniej systemu da Vinci SP.

Rzecz jasna, na świecie wciąż powstają konkurencyjne roboty chirurgiczne, często bardzo zbliżone pod względem ogólnej koncepcji konstrukcyjnej do rozwiązań, opracowanych przez Intuitive Surgical. Niemiecki system o nazwie avatera (**fotografia 9**) odróżnia się od produktów amerykańskiego potentata przede wszystkim jednorazowymi narzędziami laparoskopowymi, z których wszystkie (oprócz 10-milimetrowej kamery endoskopowej) mają średnicę tulei równą 5 mm. Co ważne, konstruktorzy robota avatera dostrzegli problemy z komunikacją głosową pomiędzy chirurgiem a resztą personelu znajdującego się w sali operacyjnej – zamiast obszernej, zabudowanej konsoli, zasłaniającej uszy oraz usta chirurga, niemieccy inżynierowie opracowali zatem wyjątkowo lekką, kompaktową konstrukcję, dającą pełną swobodę podczas porozumiewania się z innymi członkami zespołu operacyjnego (**fotografia 10**).

Rynek robotyki chirurgicznej stopniowo podbijają nie tylko systemy mniej lub bardziej zbliżone do amerykańskiego da Vinci, ale także projekty odróżniające się od konkurencji pod względem



Fotografia 9. Ramiona robota avatera podczas pracy na fantomie laparoskopowym (https://t.ly/5Nmm6)

eprasa.pl 1fc8f31a49



Fotografia 10. Smukła konstrukcja konsoli operatorskiej robota avatera umożliwia bezproblemową komunikację głosową chirurga z personelem, znajdującym się w sali operacyjnej (https://t.ly/5Nmm6)

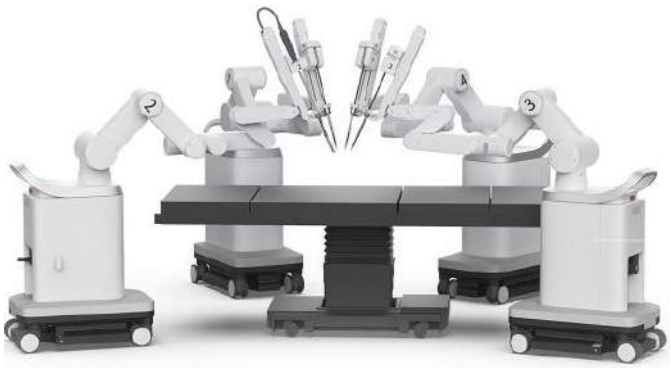
konkretnych parametrów lub rozwiązań – warto wspomnieć chociażby o robocie Senhance marki Asensus Surgical, która obok narzędzi 5-milimetrowych ma w swojej ofercie także końcówki 3-milimetrowe (**fotografia 11**). Całe narzędzie ma zatem średnicę zewnętrzną na poziomie… igły do biopsji gruboigłowej! Co ciekawe, nadmiarowa



Fotografia 11. Rodzina narzędzi laparoskopowych o średnicy 3 mm i 5 mm marki Asensus Surgical (https://t.ly/_caaB)



Fotografia 12. Robotyczny system laparoskopowy Versius marki CMR Surgical (https://t.ly/Zo2mO)



Fotografia 13. System Carina marki Ronovo Surgical (https://t.ly/EeEGp)

konstrukcja łańcucha kinematycznego daje możliwość ustalenia „cyfrowego punktu podparcia" ramion, co minimalizuje traumatyzację tkanek otaczających porty laparoskopowe.

Diametralnie inne podejście do konstrukcji mechatronicznych systemów chirurgicznych prezentuje robot Versius brytyjskiej marki CMR Surgical. W tym przypadku nie mamy już do czynienia z pojedynczą platformą mobilną, na której z jednej, wspólnej kolumny wyprowadzone są wszystkie ramiona robota. Zamiast niej otrzymujemy bowiem osobne wózki (**fotografia 12**), które mogą być dowolnie rozmieszczane w zależności od potrzeb: rodzaju zabiegu, wybranego przez chirurgów dostępu operacyjnego, a nawet składu i liczebności zespołu. Rzecz jasna, takie rozwiązanie stwarza kolejne problemy techniczne do obejścia – konieczne jest bowiem dokładne przemyślenie sposobu unikania kolizji pomiędzy poruszającymi się ramionami, co w przypadku swobodnie rozstawianych manipulatorów jest znacznie trudniejsze niż w systemach o „sztywnym" układzie odniesienia. Versius to zresztą niejedyny system chirurgiczny, oparty na niezależnych ramionach – podobne rozwiązanie zaproponowała chińska firma Ronovo Surgical, oferująca system Carina (**fotografia 13**), choć w tym przypadku znacznie większe podstawy jezdne – w porównaniu do maszyny CMR Surgical – mogą okazać się nieco mniej wygodne w użyciu, zwłaszcza na mniejszych blokach operacyjnych. Co ciekawe, marka chwali się na swojej stronie oficjalnej, że stworzenie kompletnej koncepcji projektu zajęło jej jedynie 3 miesiące, a już po kolejnych 7 miesiącach firma dysponowała prototypem funkcjonalnym i rozpoczynała testy na zwierzętach...

Jeszcze inną koncepcję zastosowano w jednym z pierwszych robotów używanych w warunkach klinicznych – system ZEUS amerykańskiej firmy Computer Motion (wchłoniętej później przez Intuitive Surgical) miał ramiona montowane bezpośrednio do specjalnego stołu operacyjnego (**fotografia 14**), co z jednej strony dawało sporą oszczędność miejsca wokół stołu, z drugiej zaś z pewnością stanowiło pewne utrudnienie podczas wstępnego pozycjonowania pacjenta. Ten sposób montażu znacznie lepiej sprawdza się natomiast w przypadku pojedynczych ramion, stosowanych do sterowania kamerą endoskopową – ta właśnie część robota ZEUS była zresztą tożsama



Fotografia 14. System ZEUS – jeden z pierwszych robotów chirurgicznych, które weszły do praktyki klinicznej (https://t.ly/QwfkI)

eprasa.pl 1fc8f31a49



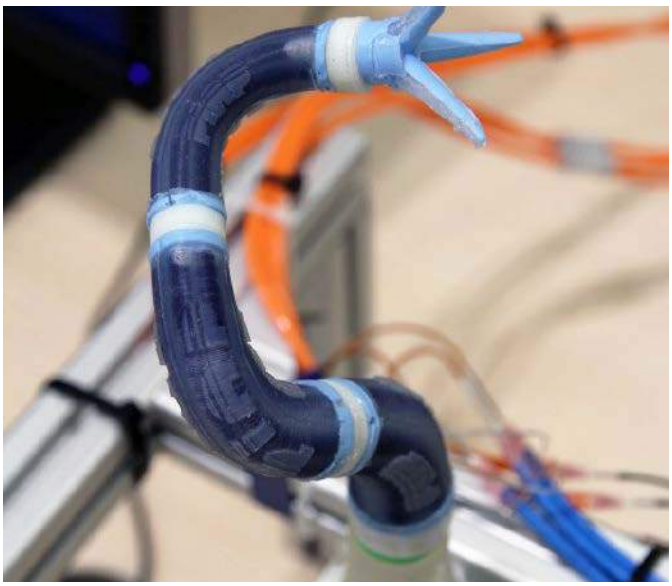
Fotografia 15. Robot Robin Heart 2 podczas testów na fantomie szkieletu ludzkiego (https://t.ly/WL8fy)

z jego poprzednikiem, manipulatorem AESOP, opracowanym na początku lat 90. XX wieku. Co ciekawe, podobne założenie projektowe zastosowali polscy konstruktorzy w projekcie robota Robin Heart 2 (**fotografia 15**).

Na marginesie warto dodać, że dwie instytucje z naszego kraju (Przemysłowy Instytut Automatyki i Pomiarów oraz Fundacja Rozwoju Kardiochirurgii im. prof. Zbigniewa Religi) brały także udział w dużym, międzynarodowym projekcie STIFF-FLOP, mającym na celu rozwój systemów ramion o kontrolowanej elastyczności dla potrzeb operacji, prowadzonych w najtrudniej dostępnych obszarach ciała ludzkiego (**fotografia 16**).

Roboty do operacji ortopedycznych

Drugą – po robotach laparoskopowych – kategorię robotów chirurgicznych stanowią maszyny przeznaczone do operacji ortopedycznych. W tym przypadku założenia projektowe są już diametralnie inne niż we wszystkich omówionych wcześniej rodzajach robotów laparoskopowych, a różnice te wynikają wprost z odmiennych potrzeb chirurgów. Podczas gdy roboty laparoskopowe są nawigowane wewnątrz ciała pacjenta na drodze sprzężenia wizyjnego (chirurg widzi trójwymiarowy obraz tkanek i na tej podstawie pozycjonuje narzędzia), maszyny do aplikacji ortopedycznych bazują na ściśle określonych wytycznych, przygotowanych uprzednio na etapie planowania operacji. Planowanie to opiera się na cyfrowych, trójwymiarowych modelach kości pacjenta, utworzonych z wyników badań obrazowych (zwykle tomografii komputerowej) – po ustaleniu trajektorii wprowadzania narzędzia czy też implantu robot pomaga w pozycjonowaniu instrumentu, np. wiertarki, służącej do wykonywania otworów pod śruby kostne. Niektóre konstrukcje wyposaża się w zintegrowane głowice narzędziowe, zatem oprócz pozycjonowania przejmują one od chirurga także zadanie właściwej obróbki kości.



Fotografia 16. Prototyp robota opracowanego w ramach projektu STIFF-FLOP (https://t.ly/CwQYs)

Przykładem opisywanego systemu może być robot Mako (**fotografia 17**), stosowany do operacji endoprotezoplastyki całkowitej stawu biodrowego (nazywanej potocznie „wymianą stawu biodrowego") oraz kolanowego. Na podstawie obrazów tomograficznych (**rysunek 1**) chirurg planuje dokładne miejsce, do którego zostanie następnie wprowadzona głowica specjalnej frezarki, przygotowującej miejsce pod umieszczenie sztucznej panewki stawu. Dokładne ustawienie narzędzi w czasie operacji ma kluczowe znaczenie dla późniejszej rekonwalescencji, a przede wszystkim – dalekosiężnych wyników leczenia, gdyż jakiekolwiek błędy mogą prowadzić do patologicznego przerostu tkanki kostnej, obluzowania sztucznego stawu, jego przemieszczenia czy nawet groźnych w skutkach złamań bądź uszkodzeń nerwów obwodowych.



Fotografia 17. Robot Mako do wspomagania operacji endoprotezoplastyki całkowitej stawu biodrowego (https://t.ly/3Zh4i)



Rysunek 1. Widok oprogramowania do planowania operacji z użyciem robota Mako (https://t.ly/3Zh4i)



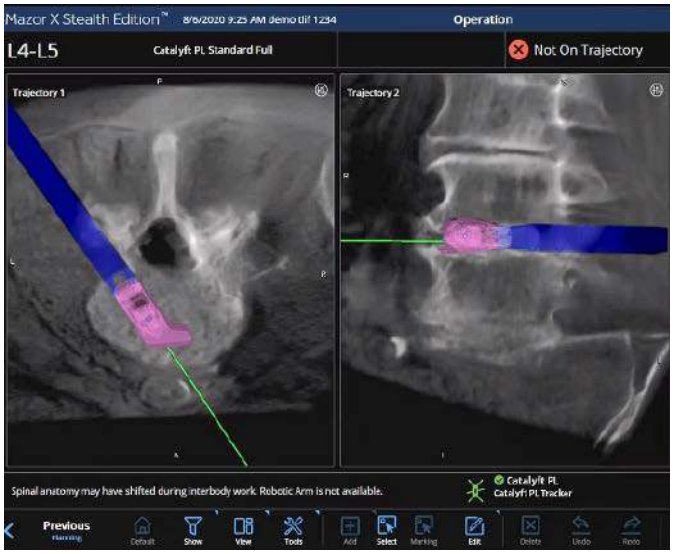
Fotografia 18. System Mazor marki Medtronic, przeznaczony do wspomagania operacji kręgosłupa (https://t.ly/sMPM1)

eprasa.pl 1fc8f31a49

Jeszcze większe wymagania stawia ortopedom chirurgia kręgosłupa, w szczególności procedury wprowadzania śrub w poszczególne kręgi. Nietrudno domyślić się, jak krytyczne znaczenie ma całkowite bezpieczeństwo pacjenta oraz niezawodność wyposażenia, np. podczas wprowadzania wiertła w kość, tuż obok przebiegającego w kanale kręgowym rdzenia. Jakiekolwiek niepożądane przesunięcie narzędzia mogłoby wszak zakończyć się dla pacjenta tragicznie. Dlatego też do powszechnej praktyki klinicznej weszły bardzo zaawansowane systemy robotyczne, które oprócz stabilnego (pod względem mechaniki oraz sterowania) manipulatora zawierają także rozbudowany system wizyjny, pozwalający na niezwykle precyzyjne pozycjonowanie zarówno ciała pacjenta, jak i ramienia robotycznego w przestrzeni trójwymiarowej. W tym celu stosuje się dokładne skanery 3D oraz specjalne znaczniki optyczne – przykład takiego rozwiązania, system Mazor marki Medtronic, można zobaczyć na **fotografii 18**. Dokładne ustawienie tulei prowadzącej umożliwia pewne wprowadzenie narzędzia (np. wiertarki – **fotografia 19**). Dodatkową zaletą stosowania zaawansowanych systemów pozycjonowania wizyjnego jest także możliwość wspomagania tych części zabiegu, które są wykonywane w całości ręcznie przez chirurga – znaczniki optyczne zamontowane na narzędziu (**fotografia 20**) pozwalają na śledzenie orientacji końcówki roboczej w w czasie rzeczywistym, w sposób bodaj najbardziej intuicyjny, bo bazujący na nałożeniu odpowiedniej wizualizacji na obraz tomograficzny (**rysunek 2**). Taka technologia bywa określana mianem chirurgii wspomaganej obrazowo (ang. Image Guided Surgery). Warto dodać, że w nurcie rozwoju robotów pozycjonujących (na potrzeby neurochirurgii i ortopedii) powstała swego czasu jedna z bodaj najciekawszych koncepcji – miniaturowy robot, mieszczący się na dłoni i mocowany bezpośrednio do struktur kostnych pacjenta. Co ciekawe, robot MARS (**fotografia 21**) został zaprojektowany tak, by… w całości mógł być poddawany sterylizacji (!) To technologiczne „cacko" powstało przeszło 20 lat temu i do dziś pozostaje jednym z najbardziej kompaktowych robotów chirurgicznych, jakie kiedykolwiek zbudowano.



Fotografia 19. Wiertarka kostna, wprowadzana do tulei prowadzącej robota Mazor (https://t.ly/8t9I0)



Rysunek 2. Wizualizacja orientacji narzędzia w czasie rzeczywistym – system Mazor (https://t.ly/opwvx)



Fotografia 21. Mikrorobot chirurgiczny MARS (https://t.ly/7Nucd)



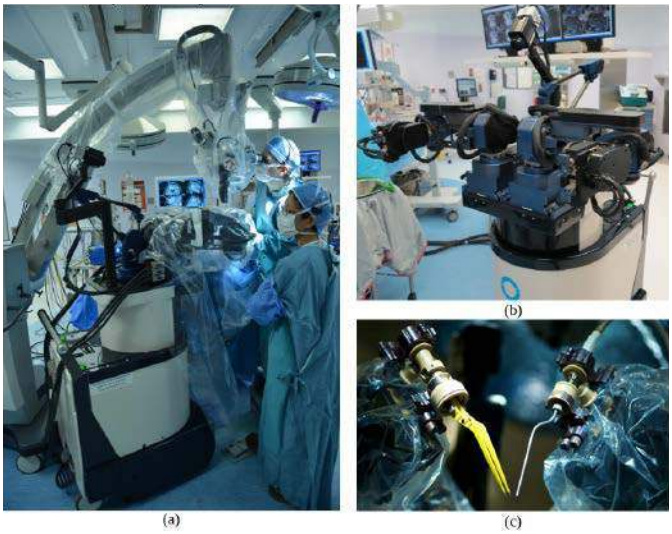
Fotografia 20. Zastosowanie systemu pozycjonowania wizyjnego Mazor do pracy z narzędziem ręcznym (https://t.ly/ULA6v)

Roboty neurochirurgiczne

Sens stosowania robotów w neurochirurgii oraz ortopedii zdaje się w dużym stopniu zazębiać. Choć niewątpliwie mikrochirurgia ośrodkowego układu nerwowego (przede wszystkim mózgowia) wymaga diametralnie innych narzędzi niż „twarda" chirurgia ortopedyczna, to w obydwu przypadkach kluczowe znaczenie ma niezwykle dokładne pozycjonowanie narzędzi, rzecz jasna w odniesieniu do danych obrazowych. W przypadku neurochirurgii są to najczęściej dane z rezonansu magnetycznego (MRI), a w najbardziej zaawansowanych systemach możliwe jest zastosowanie robota wspieranego obrazowaniem śródoperacyjnym – chirurg otrzymuje zatem aktualne informacje o położeniu narzędzi i zmianach dokonywanych w anatomii operowanych tkanek niemal w czasie rzeczywistym (co rzecz jasna – ze względów ochrony radiologicznej – nie byłoby możliwe do wykonania przy zastosowaniu tomografii komputerowej, choć i ta bywa używana w salach hybrydowych, oczywiście w znacznie węższym zakresie). W gantrze skanera MR możliwe jest jednak ustawienie tylko i wyłącznie takiego robota, który został od początku do końca zaprojektowany do pracy w silnym polu magnetycznym – nietrudno domyślić się, jak wielkie jest to utrudnienie zarówno dla elektroników, jak i mechaników, opracowujących konstrukcję maszyny.

Robot neuroArm to doskonały przykład mechatronicznego systemu neurochirurgicznego, stosowanego do rozmaitych operacji mózgu. Urządzenie składa się z wózka z dwoma manipulatorami (**fotografia 22**) oraz systemem wizyjnym (kamerą mikroskopową),

eprasa.pl 1fc8f31a49



Fotografia 22. Robot neurochirurgiczny neuroArm: (a) – ramię robota czasie operacji, (b) – manipulatory robota (przed uzbrojeniem w narzędzia), (c) – końcówki robocze umieszczone na ramionach robota: po lewej – szczypce do koagulacji bipolarnej, po prawej – ssak (https://t.ly/1gn6G)



Fotografia 23. Robot neurochirurgiczny ROSA, przeznaczony głównie do biopsji mózgu, neuroendoskopii oraz wspomagania implantacji elektrod do głębokiej stymulacji mózgu (https://t.ly/dZFIu)

a także konsoli sterującej z manetkami, nieco zbliżonymi konstrukcyjnie do tych, które zaprezentowaliśmy już przy okazji robotów laparoskopowych. Co ciekawe, robot zawiera czujniki siły, dające chirurgowi bezcenne sprzężenie haptyczne – szczególnie istotne podczas operacji na tkankach miękkich, w dodatku należących do tak złożonego i krytycznie ważnego organu, jakim jest mózg.

System ROSA (**fotografia 23**) powstał w celu wspomagania implantacji elektrod do głębokiej stymulacji mózgu (DBS), a także rozmaitych innych procedur na tym narządzie. Zastosowanie stereotaktycznego pozycjonowania narzędzia pozwala na znaczne skrócenie czasu trwania zabiegu (w niektórych przypadkach nawet o 80%!), co rzecz jasna zmniejsza obciążenie anestezjologiczne pacjenta. Mało tego – w przypadku operacji pediatrycznych dodatkową zaletą jest brak konieczności pełnego golenia skóry głowy dziecka, gdyż cały zabieg odbywa się w sposób minimalnie inwazyjny. System może być używany także do tak złożonych procedur, jak biopsja mózgu, neuroendoskopia przeznosowa czy też endoskopowe zabiegi, prowadzone na trzeciej komorze mózgu.

Roboty do zabiegów endowaskularnych

Postęp w dziedzinie robotyki medycznej nie ominął także małoinwazyjnych procedur przeznaczyniowych. Choć w tym przypadku zróżnicowanie wykonywanych przez operatora ruchów jest mocno ograniczone względem operacji laparoskopowych czy też



Fotografia 24. Robot Magellan marki Hansen Medical, przeznaczony do wspomagania zabiegów kardiologii inwazyjnej i angiologii (https://t.ly/XC1nX)

neurochirurgicznych (są to głównie obroty oraz przesuw cewnika/prowadnika wzdłuż światła naczynia), to także w tym zakresie istnieje dla robotów medycznych spore pole do popisu. O ile pacjenci mogą dzięki mechatronicznym urządzeniom zyskać lepszą jakość prowadzenia zabiegu (dokładność i powtarzalność ruchów zmniejsza ryzyko powikłań), o tyle bodaj najistotniejszy jest tutaj aspekt… bezpieczeństwa operatora. Jak powszechnie wiadomo pozycjonowanie narzędzi endowaskularnych (wewnątrznaczyniowych) odbywa się z użyciem fluoroskopii, czyli obrazowania rentgenowskiego w czasie rzeczywistym. To zaś oznacza, że operator cały czas musi znajdować się w pobliżu lampy rentgenowskiej (będącej częścią składową tzw. ramienia C), co w oczywisty sposób naraża go na przyjmowanie pewnej dawki promieniowania jonizującego. Choć fluoroskopia korzysta ze stosunkowo niskich dawek, to problem ekspozycji i tak nasila się proporcjonalnie do czasu i liczby wykonywanych procedur.

System Magellan marki Hansen Medical (**fotografia 24**) powstał właśnie po to, by zredukować powyższy problem do minimum. Głowica robota, w której mocuje się cewnik oraz niezbędne do przeprowadzenia zabiegu akcesoria, może być sterowania zdalnie, z konsoli umieszczonej za szybą ze szkła ołowiowego, skutecznie zatrzymującego większość rozproszonego promieniowania X. W ten sposób ekspozycja na promieniowanie jonizujące spada nawet o 95% względem klasycznych technik wykonywania zabiegów endowaskularnych.

Roboty do zabiegów mikrochirurgii otwartej

Robotyka dotarła także do świata mikrochirurgii otwartej – czyli wszelkiego rodzaju zabiegów, w których chirurg operuje na strukturach o rozmiarach rzędu ułamków milimetra. Do tej grupy zalicza się wiele różnych rodzajów operacji, w tym wszelkie zabiegi rekonstrukcyjne, transplantacyjne i replantacyjne kończyn, czy też niektóre operacje z zakresu angiologii bądź chirurgii plastycznej. Choć najlepsi chirurdzy z powodzeniem radzą sobie z zadaniem zszywania cienkich nerwów bądź naczyń o średnicy zewnętrznej na poziomie 0.5 mm (!), to tak wyszkolonych operatorów jest w skali całego globu stosunkowo



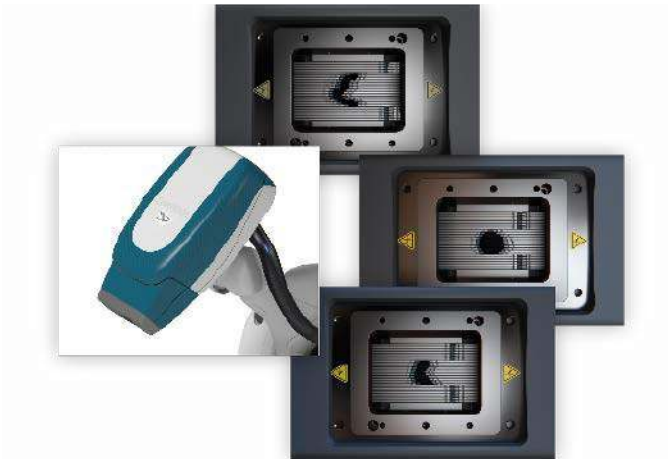
Fotografia 25. System MUSA-3 marki Microsure (https://t.ly/HDESO)

eprasa.pl 1fc8f31a49

niewielu. Zastosowanie mechatronicznego asystenta, który – podobnie jak w przypadku laparoskopii – automatycznie odfiltruje niepożądane drżenie rąk (tremor), pozwala więc na przesunięcie granic wymiarowych preparowanych tkanek jeszcze niżej. Dobrym przykładem może być tutaj robot MUSA-3 marki Microsure (**fotografia 25**) czy też system MIRA marki Virtual Incision (**fotografia 26**). Warto dodać, że o ile drugie z wymienionych urządzeń bazuje na specjalnych końcówkach narzędziowych, o tyle MUSA-3 może współpracować z konwencjonalnymi, manualnymi narzędziami mikrochirurgicznymi, co dodatkowo skraca i ułatwia proces treningu operatora, dając mu możliwość korzystania z ulubionych, sprawdzonych instrumentów.



Fotografia 26. Robot mikrochirurgiczny MIRA marki Virtual Incision (https://t.ly/ioX9E)



Fotografia 28. Kolimator wielolistkowy robota CyberKnife S7 (https://t.ly/eNp6p)

Roboty radioterapeutyczne

Radioterapia to kolejny obszar współczesnej medycyny, który coraz intensywniej korzysta z robotów. System CyberKnife S7 (**fotografia 27**) to ultranowoczesna aparatura do naświetlań z użyciem metody stereotaktycznego pozycjonowania pacjenta w celu dostarczenia silnej wiązki promieniowania jonizującego tylko do tego obszaru, w którym jest on niezbędny. Tak selektywne działanie okazuje się możliwe poprzez przemieszczanie ramienia wokół obszaru anatomicznego poddawanego terapii – jeżeli bowiem ten sam guz (nawet naprawdę niewielki) zostanie poddany działaniu promieniowania z wielu różnych kierunków (**rysunek 3**), to efekt terapeutyczny skumuluje się dokładnie w punkcie przecięcia wszystkich wiązek, zaś otaczające tkanki „dostaną" tylko niewielką część promieniowania, propagującego na wylot przez ciało pacjenta. Głowicę robota wyposażono w pełnoprawny kolimator wiązki, stosowany w standardowych aparatach do radioterapii (**fotografia 28**), a jakby tego było mało – urządzenie jest w stanie samodzielnie wymieniać kolimatory w razie potrzeby, co umożliwia jeszcze lepsze dopasowanie warunków zabiegu do potrzeb danego pacjenta. Ponadto system CyberKnife S7 potrafi automatycznie kompensować ruchomość oddechową, co daje szansę na znacznie precyzyjniejsze dozowanie dawki do narządów w rejonie klatki piersiowej i jamy brzusznej, które w wyniku akcji oddechowej mogą przemieszczać się przy każdym wdechu oraz wydechu.



Fotografia 27. System CyberKnife S7 (https://t.ly/XKsTY)

Roboty rehabilitacyjne

Rehabilitacja jest wręcz wymarzonym obszarem aplikacyjnym dla robotów – wymaga bowiem wielokrotnego, żmudnego powtarzania tych samych lub delikatnie modyfikowanych (w miarę postępów terapii) ruchów na danej kończynie lub tułowiu pacjenta. Nic więc dziwnego, że roboty rehabilitacyjne powoli zdobywają popularność w tej dziedzinie medycyny, czego doskonałym przykładem może być manipulator LBR Med marki KUKA – potentata na rynku precyzyjnych robotów przemysłowych. Urządzenie należałoby raczej zakwalifikować do grupy cobotów, czyli robotów współpracujących



Rysunek 3. Wizualizacja torów propagacji wiązek promieniowania jonizującego podczas radioterapii, prowadzonej z użyciem systemu CyberKnife S7 (https://t.ly/42yHu)



Fotografia 29. Ramię robota KUKA LBR Med, zastosowane do prowadzenia rehabilitacji (https://t.ly/7fxLS)

eprasa.pl 1fc8f31a49

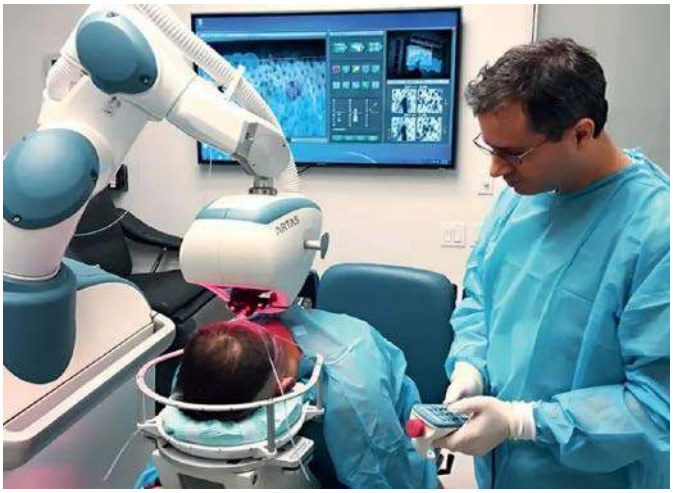
– LBR Med został bowiem specjalnie zaprojektowany do celów bezpośredniej kooperacji z człowiekiem (operatorem). Dołożenie efektora w postaci odpowiedniego uchwytu dla rehabilitowanej kończyny (**fotografia 29**) pozwala na jej bezpieczne, powtarzalne pozycjonowanie, bez konieczności wykonywania tych zadań przez fizjoterapeutę, zaś proces programowania trajektorii polega na „nauczeniu" robota odpowiednich ruchów poprzez intuicyjne, manualne prowadzenie mechatronicznego ramienia.

Roboty kosmetologiczne

Na koniec niniejszego materiału pozostawiliśmy interesujące rozwiązanie, które wprawdzie nie dotyczy krytycznych obszarów medycyny, ale dla tysięcy osób stanowi nadzieję na poprawę jakości życia. Mowa o robocie Artas, przeznaczonym do… prowadzenia zabiegów przeszczepu włosów. Tego typu procedury są niezwykle żmudne, wymagają bowiem wykonywania powtarzalnych, bardzo precyzyjnych ruchów w liczbie od kilkuset do nawet kilku tysięcy. Robot Artas (**fotografia 30**) znacząco ułatwia realizację tego typu transplantacji i wyraźnie skraca czas, dzielący pobranie włosów od wszczepienia ich na docelowe miejsce na skórze głowy pacjenta. Co ważne, zastosowanie precyzyjnie sterowanego manipulatora o wielu stopniach swobody umożliwia uzyskanie odpowiednich kątów pobierania oraz implantacji, co pozytywnie wpływa na powodzenie całego przedsięwzięcia.

Podsumowanie

W artykule przedstawiliśmy przekrój współczesnego rynku robotów medycznych. Ta niezwykle interesująca dziedzina techniki, rozwijająca się od kilkudziesięciu lat na styku medycyny zabiegowej oraz mechatroniki, imponuje dziś możliwościami, które dalece wykraczają poza fizyczne możliwości nawet najlepszych chirurgów w zakresie powtarzalności i precyzji ruchów. A jednak nic nie wskazuje na to, by w najbliższych latach jakikolwiek robot był w stanie samodzielnie wykonać nawet najprostszą operację. Zamiast



Fotografia 30. Zastosowanie robota Artas do przeszczepu włosów (https://t.ly/QMH12)

tego inżynierowie w wielu krajach świata w pocie czoła pracują nad chociażby częściową automatyzacją niektórych elementów zabiegu – np. nad implementacją półautomatycznych lub w pełni automatycznych gestów szycia. Wymaga to jednak ścisłej współpracy mechatronicznych manipulatorów z systemami wizyjnymi, a nade wszystko – zaawansowanymi algorytmami sztucznej inteligencji, zdolnymi do poradzenia sobie z odtworzeniem sekwencji ruchów (związanych z wkłuwaniem, przesuwaniem oraz wyciąganiem igły z nicią chirurgiczną), która dla nawet najbardziej początkującego adepta chirurgii jest rutynową czynnością, przy odrobinie praktyki niewymagającą nawet głębszego zastanowienia. Po raz kolejny okazuje się zatem, że nawet najbardziej wyrafinowane systemy automatyczne są wciąż tylko narzędziem w rękach ludzi. Może to i lepiej?

inż. Przemysław Musz, EP

REKLAMA



eprasa.pl 1fc8f31a49

Zaawansowane obwody PCB do najbardziej wymagających aplikacji

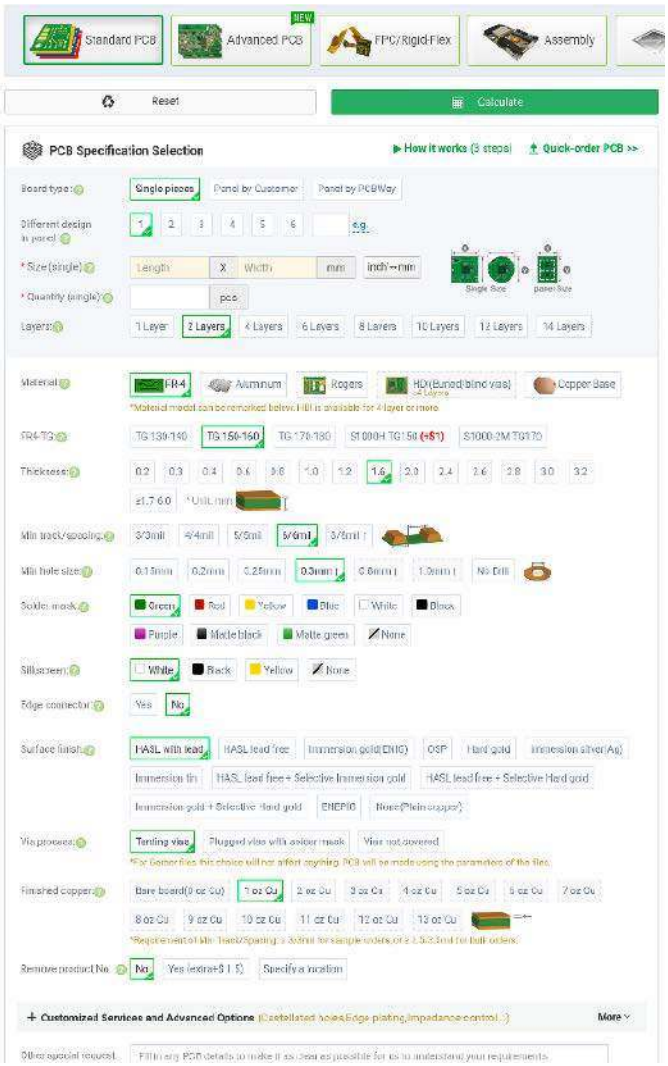
Liczba producentów obwodów drukowanych na światowym rynku PCB sukcesywnie rośnie, ale wciąż tylko nieliczni są w stanie sprostać najbardziej wygórowanym wymaganiom, które stawiają im na co dzień odbiorcy płytek. A jednak tylko nieliczni potrafią wyróżnić się na tle konkurencji. Firma PCBWay należy do ścisłej czołówki i to nie tylko pod względem oferowanych możliwości technologicznych, ale także doskonałego zarządzania całym cyklem realizacji zamówień.



Fotografia 1. Szeroka gama kolorów soldermaski, dostępnych w PCBWay

Ekspresowe zamówienia PCB – prostsze niż kiedykolwiek (i to nawet dla 14 warstw!)

W rynkowej praktyce zamawianie produkcji obwodów drukowanych – w szczególności prototypów oraz krótkich serii produkcyjnych – odbywa się z reguły za pośrednictwem formularza on-line. Zazwyczaj tego typu narzędzia obsługują jednak tylko najprostsze zlecenia, dotyczące płytek dwu-, ewentualnie czterowarstwowych i mają mocno ograniczone możliwości konfiguracji parametrów PCB. W przypadku systemu opracowanego przez PCBWay formularz PCB Instant Quote (**rysunek 1**) pozwala na wybór wszystkich najistotniejszych cech obwodów i to w imponująco szerokim zakresie. Oprócz parametrów takich, jak liczba warstw, wymiary zewnętrzne płytki, finalna grubość po laminacji, kolory soldermaski (**fotografia 1**) oraz warstwy opisowej czy też grubość miedzi, użytkownik ma możliwość wyboru jednego z kilkunastu (!) rodzajów wykończenia powierzchni, minimalnych szerokości ścieżek i odstępów izolacyjnych, minimalnych średnic otworów, a nawet… rodzaju laminatu. Co ciekawe, nawet w zakładce przeznaczonej dla standardowych obwodów do wyboru jest nie tylko kilka rodzajów FR-4 (zróżnicowanych pod względem temperatury zeszklenia), ale nawet podłoża aluminiowe i miedziane oraz laminaty mikrofalowe marki Rogers. Mało tego – można także ustalić sposób zatykania przelotek (standardowy tenting lub zatykanie soldermaską) – a należy pamiętać, że wciąż mówimy tutaj tylko o standardowych obwodach drukowanych, w liczbie warstw od 1 do – bagatela – 14 (!). Duże ułatwienie stanowi możliwość zamówienia – w ramach tego samego formularza – nie tylko produkcji PCB, ale także przeznaczonego dla nich szablonu SMT, a nawet… usługi montażu kontraktowego. W większości przypadków podsumowanie kosztów jest dokonywane automatycznie przez system informatyczny (i to od razu dla kilku różnych terminów realizacji), a dla obwodów na podłożach metalowych oraz specjalnych (Rogers) dokładną wycenę przygotowuje dział handlowy PCBWay.

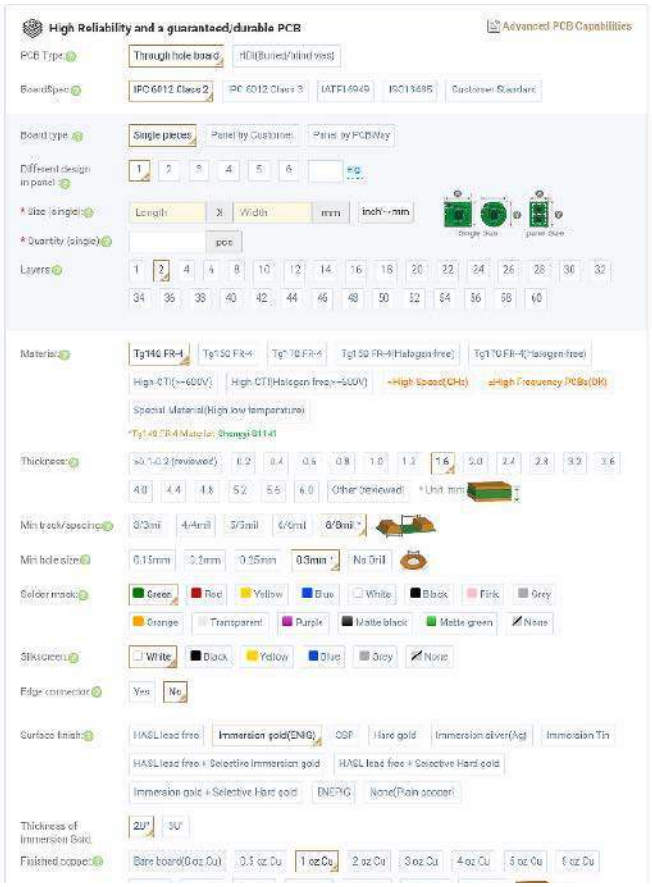


Rysunek 1. Formularz szybkich zamówień PCB w procedurze produkcji standardowej

Zaawansowane technologie na wyciągnięcie ręki

Opisane powyżej funkcje konfiguracji zamówień PCB to zaledwie wierzchołek przysłowiowej góry lodowej, gdyż o prawdziwych możliwościach technologicznych parku maszynowego PCBWay można przekonać się, przechodząc do zakładki Advanced PCB (**rysunek 2**). O zawrót głowy przyprawia już sama tylko maksymalna liczba warstw (równa 60) – a to wciąż dopiero początek rozbudowanego formularza. Użytkownik może wybrać, wg jakich wymagań normalizacyjnych

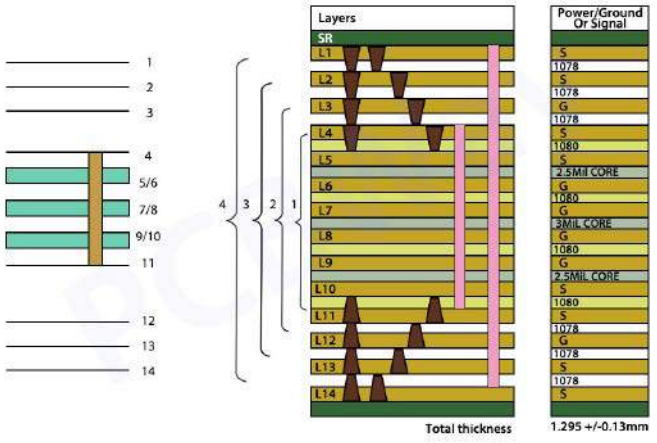
eprasa.pl 1fc8f31a49



Rysunek 2. Formularz zamówień zaawansowanych obwodów drukowanych

ma być wykonana produkcja – firma PCBWay zapewnia zgodność z IPC 6012 (klasy 2 i 3), a także IATF16949 (system zarządzania jakością dla branży motoryzacyjnej) oraz ISO 13485 (system zarządzania jakością dla wyrobów medycznych – przykład obwodów do zastosowań w ochronie zdrowia można zobaczyć na **fotografii 2**). Obwody mogą bazować na laminatach mikrofalowych o paśmie roboczym nawet powyżej 25 GHz, a całkowitą grubość PCB po laminacji można wybrać w zakresie od 0,2 mm do nawet 6,0 mm. PCBWay oferuje produkcję płytek w technologii heavy copper (grubość zewnętrznych warstw miedzi do 13 oz!), a do ustalenia przez użytkownika jest nawet… grubość metalizacji otworów (i to w zakresie do 70 µm). Rzecz jasna, dla PCBWay nie stanowią problemu nawet najbardziej rozbudowane stosy obwodów HDI ze wszystkimi możliwymi typami przelotek (**rysunek 3**).

Ostatnia część formularza – na pierwszy rzut oka niepozorna lista dodatkowych opcji – daje pełną kontrolę nad szeregiem zaawansowanych



Rysunek 3. Przykładowy stos płyty HDI, wyprodukowanej w procesie laminacji sekwencyjnej

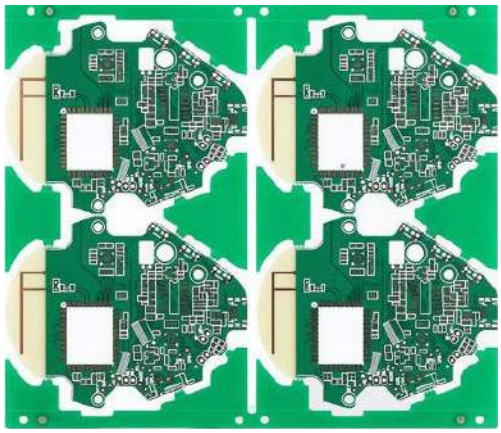
technologii, których próżno szukać u zdecydowanej większości konkurencyjnych producentów. PCBWay umożliwia m.in.:

- tworzenie padów typu castellated (stosowanych w modułach lutowanych powierzchniowo na płytach bazowych),
- metalizację krawędzi,
- kontrolę impedancji,
- nadruk ścieżek za pomocą pasty grafitowej,
- kompletne wypełnianie przelotek żywicą epoksydową,
- realizację przelotek w padach SMD (via-in-pad),
- wykończenie otworów montażowych z podfrezowaniem lub fazą (counterbore/countersink),
- frezowanie wnęk na głębokość (w kierunku osi Z),
- przygotowanie padów przelotowych pod komponenty wtykane na wcisk,
- wypełnianie przelotek pastą na bazie miedzi (w celu zwiększenia przewodności cieplnej),
- aplikację wewnętrznych bloków miedzi (tzw. copper coins),
- rozwiercanie metalizacji otworów na kontrolowaną głębokość (back drilling),
- produkcję obwodów hybrydowych oraz sztywno-giętkich.

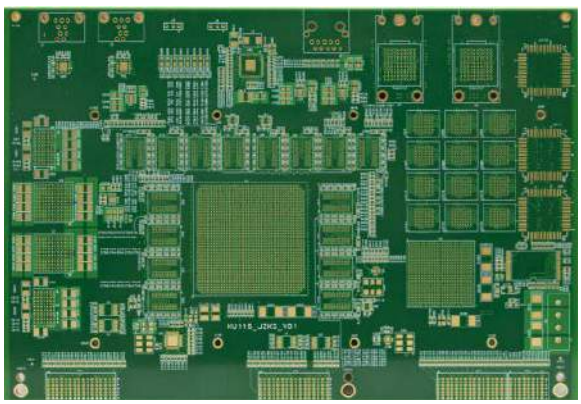
Co więcej, do wyboru jest także szereg opcjonalnych testów i raportów, m.in. badania mikroprzekrojów, testy lutowalności i termicznego czy też wykonywane na miejscu w fabryce badania impedancji dla obwodów RF, mających znaczenie krytyczne dla funkcjonalności danego urządzenia. Nic więc dziwnego, że wśród naszych klientów znajdują się m.in. firmy z najbardziej wymagających branż – medycznej, motoryzacyjnej czy nawet kosmicznej (**fotografia 3**).

Zamówienie szablonów w ośmiu kliknięciach

Proces zamawiania produkcji szablonów do nakładania pasty lutowniczej – w przypadku systemu online oferowanego przez PCBWay – okazuje się równie łatwy i przyjemny, jak analogiczna procedura dla samych płytek, a całość sprowadza się do wspomnianych ośmiu



Fotografia 2. Przykładowe obwody drukowane do aplikacji medycznej



Fotografia 3. Zaawansowana płyta HDI do zastosowań kosmicznych

eprasa.pl 1fc8f31a49

kliknięć (**rysunek 4**). W pierwszych trzech krokach należy wybrać rodzaj szablonu (z ramą lub bez ramy), jego rozmiar (z obszernej listy predefiniowanych możliwości), a także tryb produkcji (szablon zwykły lub o stopniowanej grubości). Następnie należy wybrać, które strony płytki mają być wzięte pod uwagę (warstwy top, bottom lub obydwie), przy czym możliwe jest wykonanie obydwu szablonów na tym samym lub na dwóch osobnych arkuszach blachy nierdzewnej. Potem wystarczy jeszcze tylko ustalić liczbę egzemplarzy, grubość całkowitą oraz sposób traktowania powierzchni, a także zaznaczyć, czy producent ma wykonywać znaczniki optyczne (fiducials). I to wszystko – w zaledwie kilku kliknięciach możemy zamówić szablon ściśle dopasowany do naszych potrzeb i możliwości technologicznych (np. rozmiaru oraz rodzaju posiadanej drukarki szablonowej).



Rysunek 4. Sekcja formularza, przeznaczona do konfiguracji szablonu pasty lutowniczej

Logistyka i śledzenie zamówień na światowym poziomie

Godnym uwagi aspektem pozostaje sposób obsługi zamówień, który także zasługuje na miano ścisłej czołówki w branży PCB. Ze wsparcia logistycznego można skorzystać przez całą dobę za pośrednictwem czatu online – to bardzo ważne zwłaszcza z uwagi na sporą różnicę czasową (szczególnie w przypadku odbiorców z Europy). Wszystkie zagadnienia, związane tak z kwestiami finansowymi oraz spedycyjnymi, jak i przebiegiem procesu technologicznego, są dostępne w rozbudowanym panelu użytkownika. Zamawiający może samodzielnie monitorować postępy prac w czasie rzeczywistym, a dokładność wskazań jest oszałamiająca – w dowolnym momencie wiadomo bowiem, na którym etapie procesu technologicznego (a zatem – obrazowo rzecz ujmując – „w której maszynie") znajdują się obecnie produkowane obwody. Co ważne, informacja zwrotna w sprawie kompletności i poprawności dokumentacji (z produkcyjnego punktu widzenia) jest dostępna nawet w ciągu kilku minut od przesłania plików Gerber oraz owiertów – zespół wykwalifikowanych inżynierów błyskawicznie weryfikuje przesłane dokumenty i w razie wszelkich niejasności lub braków instruuje użytkownika, co należy dosłać lub poprawić, by zamawiana płytka była możliwa do prawidłowego wyprodukowania.

Podsumowanie

W artykule przedstawiliśmy szereg zagadnień, niezwykle istotnych z punktu widzenia tak firm, jak i osób prywatnych, zlecających produkcję obwodów drukowanych w PCBWay. Zarówno proces zamawiania, jak i śledzenia postępów produkcji są silnie wspierane tak przez doskonale przemyślany system online, jak i przez sztab wykwalifikowanych inżynierów oraz konsultantów – wszystko to pozwala nam świadczyć usługi produkcji PCB, szablonów SMT oraz montażu kontraktowego na najwyższym, światowym poziomie.

www.pcbway.com

REKLAMA



eprasa.pl 1fc8f31a49



źródło: https://tinyurl.com/26cw3bzr

Rewolucja w świecie audio – subminiaturowe głośniki xMEMS

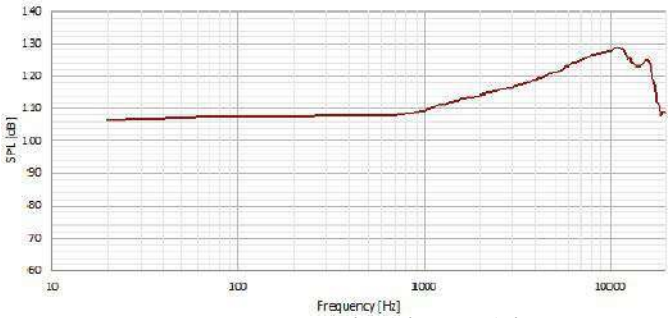
Klasyczne membrany głośnikowe są zwykle produkowane w oparciu na stosunkowo wąskiej grupie materiałów, których obróbka – w przypadku konstrukcji elektrodynamicznych – nie stanowi raczej większego wyzwania technologicznego. A co, gdyby celulozę bądź polipropylen zastąpić materiałem pozornie całkowicie niezdolnym do pracy w roli membrany – krzemem? Takie wyzwanie, z doskonałym zresztą skutkiem, podjęła firma xMEMS, wprowadzając na rynek serię miniaturowych głośników, opartych na technologii MEMS.

Kalifornijska firma xMEMS jest światowym pionierem w dziedzinie projektowania i produkcji monolitycznych (tj. całkowicie krzemowych) mikrogłośników, bazujących na nowoczesnej technologii MEMS, zaś o skali przedsięwzięcia niech świadczy fakt, iż stosunkowo młody start-up ma ponad 130 patentów oraz ponad 100 kolejnych zgłoszeń patentowych w tym zakresie. Poniższy krótki artykuł przedstawia fundamentalne zagadnienia konstrukcyjne i przybliża podstawy działania głośników MEMS, a także ich różnice w stosunku do konwencjonalnych technologii.

Krótki rys historyczny

Tradycyjne głośniki już od ponad 100 lat generują dźwięk przy użyciu kombinacji ruchomej cewki, połączonej z membraną i współpracującej z nieruchomym magnesem. Na przestrzeni dekad rozwoju techniki audio powstały tysiące odmian głośników, różniących się budową, zastosowanymi materiałami, jakością wykonania – a co za tym idzie – wynikową jakością odbioru dźwięku przez słuchacza. Na odpowiedź częstotliwościową głośnika oraz jego osiągi w zakresie mocy maksymalnej wpływają praktycznie wszystkie aspekty konstrukcyjne i materiałowe, a to wiąże się z kolosalnym rozrzutem cen, widocznych dla odbiorcy końcowego. Są na rynku głośniki (i – co ciekawe – znajdują one nabywców) o wartości dobrej klasy samochodu.

Z drugiej strony, od kilkudziesięciu lat są dostępne na rynku stosunkowo tanie głośniki piezoelektryczne, zbudowane z różnych materiałów krystalicznych. Podzespoły te mają z reguły bardzo prostą budowę w porównaniu z tradycyjnymi głośnikami dynamicznymi, niemniej jednak jakość odtwarzania oraz ich trwałość praktycznie ograniczyły zastosowanie tych głośników do aplikacji sygnalizacyjnych (buzzery).

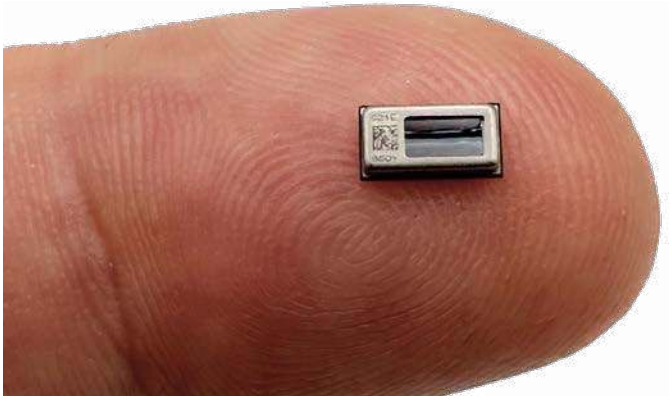


Rysunek 1. Charakterystyka częstotliwościowa głośnika xMEMS

eprasa.pl 1fc8f31a49

Tabela 1.

Materiał membrany	Moduł Younga (sztywność) [GPa]	Gęstość [kg/m³]	Prędkość względna w ośrodku NA
Papier	2	800	0,2
Polipropylen	1,68	900	0,2
Aluminium	68	2700	0,6
Tytan	116	4500	0,6
Krzem	160	2330	1
Diament	1220	3520	2,25



Fotografia 1. Głośnik xMEMS typu Cowell (https://tinyurl.com/2n9575pt)



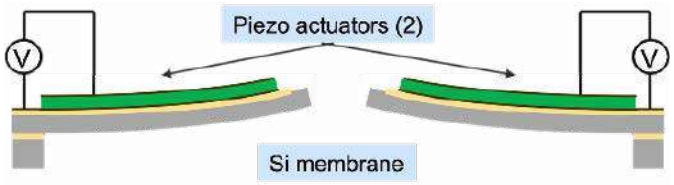
Fotografia 2. Głośnik xMEMS typu Montara Plus (https://tinyurl.com/4hmy94u6)

xMEMS – nowe otwarcie w zakresie przetworników akustycznych

Firma xMEMS wychodzi z propozycją głośnika miniaturowego (obudowa SMD), całkowicie krzemowego, o stosunkowo wysokich (jak na tak niewielkie wymiary) osiągach w zakresie SPL. Mało tego – przetwornik oferuje bardzo wysoką odporność na warunki środowiskowe, zaś z uwagi na zastosowaną technologię MEMS produkcja odbywa się z dużą precyzją i powtarzalnością, charakterystyczną wszak dla zakładów wytwarzających komponenty półprzewodnikowe.

W tym miejscu ważna uwaga, dotycząca parametru SPL głośnika (ang. Sound Pressure Level) – jest to mierzona w decybelach wartość poziomu ciśnienia akustycznego, jakie jest w stanie wytworzyć określony model przetwornika, przy dostarczeniu mu sygnału elektrycznego o danej amplitudzie i częstotliwości – z tego też względu ten sam parametr jest niekiedy nazywany po prostu czułością (ang. sensitivity).

Warto wiedzieć, że najważniejszą zaletą zastosowania krzemu do wytworzenia membrany głośnika jest jego sztywność, umożliwiająca uzyskanie dźwięku wysokiej jakości i to już od samego początku



Rysunek 2. Uproszczony przekrój konstrukcji przetwornika xMEMS



Rysunek 3. Budowa słuchawki dousznej na bazie głośnika xMEMS (https://tinyurl.com/2ucsx8s8)

pasma słyszalności (około 20 Hz). Jak wiadomo, niewielkie głośniczki, stosowane m.in. w słuchawkach dousznych, „kuleją" zwykle w dolnych obszarach spektrum, podbijając jednocześnie wysokie tony – w tym przypadku charakterystyka jest perfekcyjnie płaska w zakresie od 20 Hz do niemal 1 kHz, a dalszy wzrost SPL w funkcji częstotliwości przebiega w dość „przewidywalny", wręcz liniowy sposób (i to aż do 10 kHz – patrz **rysunek 1**).

Najwyższej klasy głośniki średnio-wysokotonowe bazują czasem na „egzotycznych", bardzo sztywnych materiałach (np. tytanie czy nawet diamencie), aby uzyskać możliwie klarowny dźwięk w szerokim paśmie. Jednak większość dzisiejszych dousznych głośników używa membran plastikowych, których sztywność plasuje się na poziomie około 1/100 wartości tego parametru dla krzemu (patrz **tabela 1**). Krzem jest 95 razy sztywniejszy od tworzyw sztucznych, ponad 2 razy sztywniejszy od aluminium i o 30% sztywniejszy od tytanu. Zapewnia to bardziej jednolite pasmo przenoszenia, eliminując typowe niedoskonałości widmowe tradycyjnych głośników.

Poniżej przedstawiamy najważniejsze parametry głównych członków rodziny mikrogłośników marki xMEMS.

Cowell (**fotografia 1**) to najmniejszy na rynku głośnik typu MEMS o wymiarach 3,2×6,0×1,15 mm i polu powierzchni membrany równym 22 mm³. Uzyskiwany za jego pomocą poziom SPL to 116 dB (@ 2 kHz/30 Vpp).

Montara Plus (**fotografia 2**) to miniaturowy głośnik MEMS o stosunkowo dużej mocy – jego wymiary to 5,15×10,8×1,15 mm, a powierzchnia membrany jest niemal 3× większa niż w przypadku modelu Cowell (64 mm³). Przetwornik osiąga wartość poziomu ciśnienia akustycznego równą 127 dB przy pobudzeniu sygnałem 2 kHz/30 Vpp.

Przykładowe zastosowania głośników xMEMS:

- IEM (In Ear Monitor – **rysunek 3**), czasem zwane jako monitory douszne (stosowane przez profesjonalistów z branży muzycznej),
- Aplikacje medyczne (np. słuchawki zdolne do pracy w silnym polu magnetycznym),
- Słuchawki pozbawione problemu generowania alarmu przez wykrywacze metali na lotniskach,
- Słuchawki pracujące w trudnych warunkach pogodowych, w środowisku przemysłowym,
- Bezprzewodowe słuchawki douszne dla konsumentów,
- Aparaty słuchowe.

Bartosz Olesiński

http://www.ep.com.pl/EPwtoku

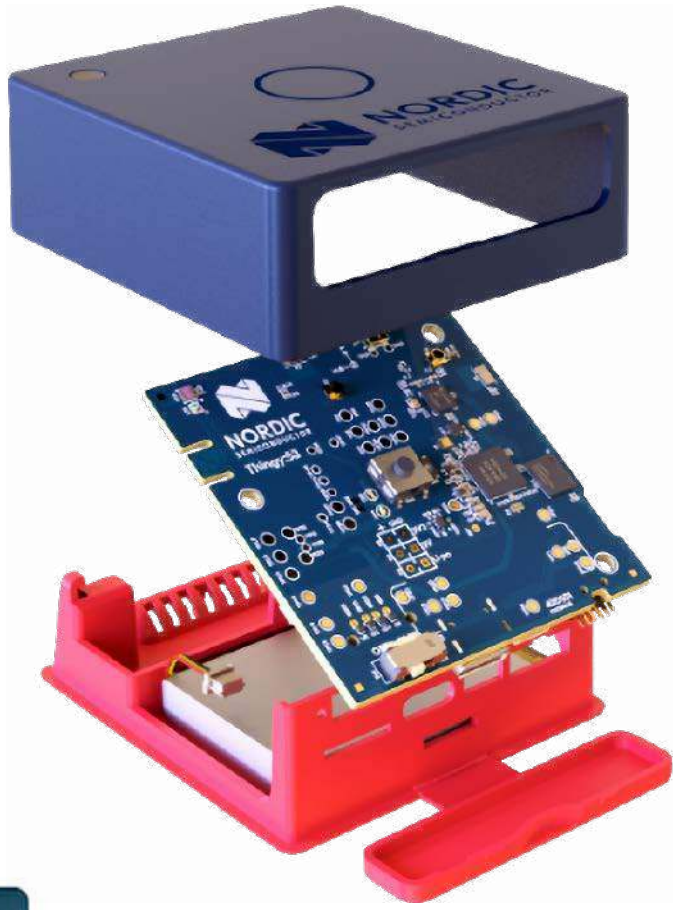
eprasa.pl 1fc8f31a49

Internet Rzeczy w pomiarach środowiskowych (1)

Nordic Thingy:53 – platforma prototypowania IoT z uczeniem maszynowym (ML)

Nordic Thingy:53 firmy Nordic Semiconductor to łatwa w użyciu platforma do eksperymentów w zakresie Internetu Rzeczy. Umożliwia tworzenie prototypów systemów IoT bez konieczności budowania niestandardowego sprzętu. Moduł Thingy:53 jest zbudowany z zastosowaniem układu scalonego nRF5340 – dwurdzeniowego procesora komunikacyjnego SoC. Moc obliczeniowa i rozmiar pamięci dwóch rdzeni Arm Cortex-M33 umożliwiają uruchamianie wbudowanych modeli uczenia maszynowego (ML) bezpośrednio na urządzeniu.

Autor składa podziękowania Panu Maciejowi Michnie z Centrum Badań i Rozwoju Nordic Semiconductor w Krakowie za udostępnienie modułów sprzętowych Nordic Thingy:53.



Po wprowadzeniu pakietów deweloperskich Thingy:52 oraz Thingy:91 IoT odpowiednio w latach 2017 i 2019, norweska firma wypuściła w roku 2022 nową platformę Thingy:53, opartą na dwurdzeniowym SoC. Nordic Thingy:53 zawiera czujniki, które zbierają dane o jego własnych ruchach i otaczającym środowisku. Informacje dotyczące temperatury, wilgotności, jakości i ciśnienia powietrza, a także dźwięku oraz koloru i natężenia światła można łatwo wyodrębnić do analizy lokalnej lub zdalnej. Nordic Thingy:53 ma także dwa przyciski, diodę LED RGB oraz brzęczyk.

Pudełko zestawu Nordic Thingy:53 zawiera kilka elementów (**rysunek 1**) [1]:

- płytkę drukowaną (PCA20053) w plastikowej, dwuczęściowej (czarna i czerwona) obudowie z akumulatorkiem i dołączoną anteną NFC (PCA64110) (**rysunek tytułowy** [1]),
- ulotkę informacyjną,
- kartę rozszerzeń (PCA64142) do pomiaru prądu i debugowania.

Nordic Thingy:53 bazuje na trzech głównych układach scalonych firmy Nordic Semiconductor: nRF5340 (dwurdzeniowy procesor



Rysunek 1. Elementy zestawu Nordic Thingy:53 [2]

eprasa.pl 1fc8f31a49

komunikacyjny SoC), nRF21540 (RF FEM ze wzmacniaczem mocy RF (PA) i wzmacniaczem niskoszumowym (LNA)) oraz nPM1100 (układ scalony zarządzania zasilaniem, w skrócie PMIC).

Kluczowe cechy Nordic Thingy:53 [2]

- Układ scalony nRF5340, który obsługuje Bluetooth Low Energy, 802.15.4 i komunikację Near Field Communication (NFC) jako znacznik pasywny.
- Scalony front-end RF nRF21540 (FEM) dla zwiększenia zasięgu,
- Układ scalony zarządzania zasilaniem (PMIC) typu nPM1100 do zarządzania energią i ładowania akumulatora litowo-polimerowego (Li-Poly).
- Dwie anteny 2,4 GHz i jedna pasywna antena NFC (elastyczna płytka drukowana).
- Wkładana karta rozszerzeń do pomiaru prądu i debugowania, ułatwiająca profilowanie mocy i pracę nad oprogramowaniem.
- Złącze kompatybilne z Qwiic i STEMMA QT do podłączania modułów zewnętrznych.
- Przyciski programowalne przez użytkownika i dioda LED RGB.
- Czujnik środowiskowy temperatury, wilgotności, jakości i ciśnienia powietrza.
- Czujnik koloru i natężenia światła.
- Akcelerometr o niskim poborze mocy.
- 6-osiowy IMU z żyroskopem i akcelerometrem.
- Magnetometr.
- Mikrofon z modulacją gęstości impulsów (PDM) z funkcją budzenia dźwiękiem.
- Brzęczyk.
- Pamięć zewnętrzna 64 Mb z interfejsem Quad Serial Peripheral Interface (QSPI).
- Akumulator Li-Poly o pojemności 1350 mAh.
- Gniazdko USB-C do podłączenia do komputera i ładowania akumulatora.
- Normalny zakres temperatur pracy 0…35°C.

Układ scalony procesora komunikacyjnego nRF5340 [7]

Rdzeń aplikacyjny:

- 128/64 MHz Arm Cortex-M33 z technologią TrustZone,
- 1 MB pamięci Flash i 512 kB pamięci RAM o niskim poborze prądu,
- Podsystem bezpieczeństwa ARM TrustZone CryptoCell-312,
- Układ peryferyjny QSPI do komunikacji z zewnętrzną pamięcią Flash (tryb Execute-In-Place (XIP)) z opcjonalnym szyfrowaniem i deszyfrowaniem w locie,
- Znacznik NFC-A z polem wybudzania i funkcją „dotknij, aby sparować”,
- Do pięciu urządzeń master/slave SPI z EasyDMA,
- Do czterech I²C z EasyDMA (kompatybilne z układami pracującymi w trybach master/slave),
- Do czterech bloków UART (obsługujących linie CTS/RTS) z EasyDMA,
- Urządzenia peryferyjne audio: I²C, interfejs mikrofonu cyfrowego (PDM),
- Do czterech bloków PWM z EasyDMA,
- Przetwornik ADC, 12-bitowy, 200 kSps z EasyDMA, osiem konfigurowalnych kanałów z programowalnym wzmocnieniem,
- USB Full-speed (12 Mb/s).

Rdzeń komunikacyjny („radiowy”):

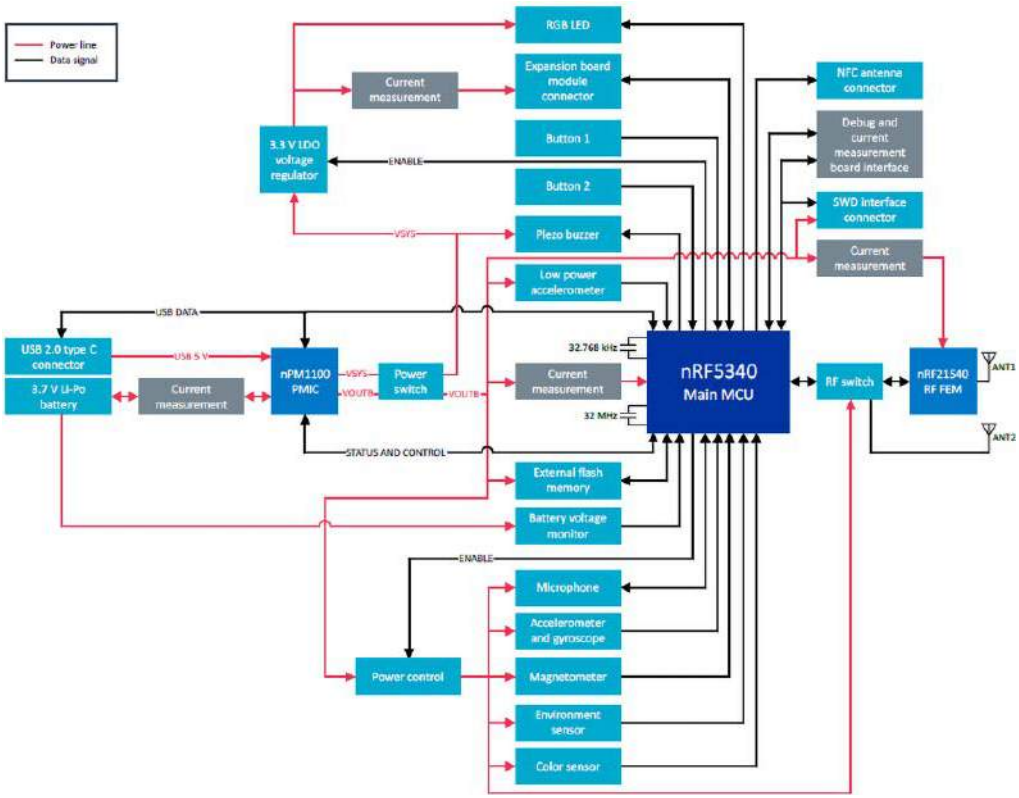
- 64 MHz ARM Cortex-M33,
- 256 kB pamięci Flash i 64 kB pamięci RAM o niskim poborze prądu,
- Transceiver obsługujący Bluetooth 5.2, IEEE 802.15.4-2006, 2,4 GHz,
- Master/slave SPI z EasyDMA,
- I²C z EasyDMA kompatybilny z układami pracującymi w trybach master/slave,
- UART (CTS/RTS) z EasyDMA.

Układ scalony rozszerzenia zasięgu nRF21540 [8]

- FEM (Front-End Module) ze wzmacniaczem mocy RF (PA) i wzmacniaczem niskoszumnym (LNA),
- Obsługuje Bluetooth Low Energy, IEEE 802.15.4 i aplikacje firmowe,
- Moc wyjściowa 20 dBm przy 3,0 V (+22 dBm przy 3,6 V),
- Regulowana moc wyjściowa z dokładnością do ±1 dB w zakresie od 5 do 21 dBm,
- Wzmocnienie odbioru +13 dB,
- Interfejs sterowania poprzez I/O, SPI lub ich kombinację.

Układ scalony zarządzania zasilaniem nPM1100 [9]

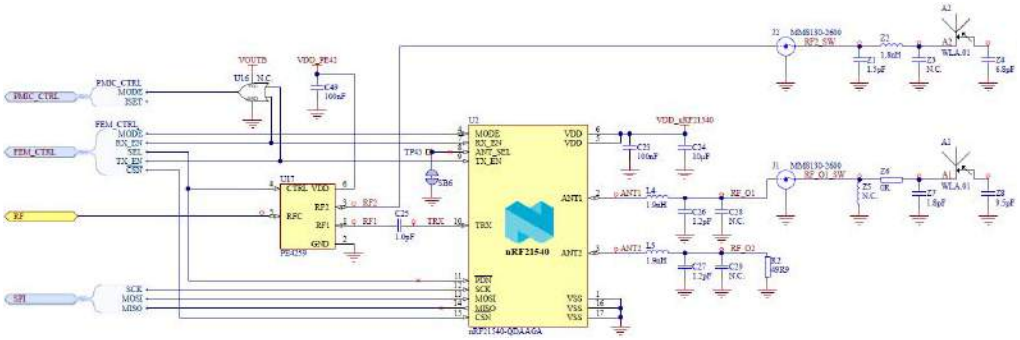
- Liniowa ładowarka akumulatorów Li-Ion/Li-Poly 400 mA z zabezpieczeniem termicznym:
  - Automatyczne ładowanie podtrzymujące, ładowanie stałym prądem i stałym napięciem,
  - Zabezpieczenie termiczne akumulatora,
  - Ograniczenie prądu rozładowania.
- Regulator obniżający napięcie 1,8…3,0 V, 150 mA:
  - Automatyczne przejście pomiędzy trybem histerezy i PWM,
  - Wymuszony tryb PWM, zapewniający zasilanie z niskim poziomem zakłóceń.



Rysunek 2. Schemat blokowy Nordic Thingy:53 [2]

eprasa.pl 1fc8f31a49

- Regulator wejściowy z ograniczeniem prądu, zgodnym z USB 100 mA i 500 mA:
  - Zakres napięcia wejściowego 4,1…6,7 V dla normalnej pracy.
- Sterowniki LED do sygnalizacji stanu ładowarki.
- Zakres napięć wejściowych dla zasilania akumulatorowego 2,3…4,35 V.



Rysunek 3. Układ scalony interfejsu radiowego nRF21540 (RF FEM) [2]

Na schemacie blokowym (**rysunek 2**) można zobaczyć połączenia pomiędzy komponentami sprzętowymi płytki Nordic Thingy:53. Cztery szare prostokąty reprezentują układy kluczy, umożliwiających pomiar prądu.

Nordic Thingy:53 wykorzystuje układ scalony interfejsu radiowego nRF21540 (U2), aby poprawić zasięg i niezawodność połączenia. nRF21540 umożliwia wzmocnienie RX do +13 dB, niski poziom szumów 2,5 dB i moc wyjściową TX do +21 dBm (**rysunek 3**). Wyjście anteny (ANT) układu nRF5340 SoC jest podłączone do przełącznika radiowego PE4259 firmy Murata (U17), który umożliwia wybór tylko jednej z dwóch anten w danym momencie: A1 (dołączonej do nRF21540, TX +20 dBm max.) lub A2 (dołączonej do nRF5340, TX +5 dBm max.). Anteny są dołączone poprzez gniazda radiowe z wyłącznikiem (SWF firmy Murata), co pozwala na wpięcie zewnętrznej anteny lub miernika mocy RF.

Złącze USB-C

Złącze USB-C (J6) jest podłączone do interfejsu USB w nRF5340 SoC i nPM1100 PMIC. Umożliwia komunikację Nordic Thingy:53 z komputerem oraz pozwala na ładowanie akumulatora.

Czujniki ruchu

Nordic Thingy:53 zawiera 3-osiowy akcelerometr małej mocy ADXL362 (U8), 6-osiowy IMU BMI270 (U7), integrujący w sobie akcelerometr i żyroskop, a także 3-osiowy magnetometr BMM150 (U12).

ADXL362 firmy Analog Devices to 3-osiowy akcelerometr MEMS o bardzo niskim poborze mocy, który pobiera mniej niż 2 µA przy wyjściowej szybkości transmisji danych równej 100 Hz i 270 nA w trybie uśpienia z budzeniem, wyzwalanym przez ruch. Oprócz wyjątkowo niskiego zużycia energii, ADXL362 ma także wiele funkcji, umożliwiających dalszą redukcję mocy na poziomie systemowym. Zawiera wyjście FIFO, wbudowany czujnik temperatury z mikrozasilaniem i kilka trybów wykrywania aktywności, w tym regulowany próg uśpienia i budzenia, który może działać z zasilaniem na poziomie 270 nA przy (przybliżonej) częstotliwości pomiaru 6 Hz.

Gdy Nordic Thingy:53 znajduje się w trybie uśpienia o niskim poborze mocy, akcelerometr ADXL362 wykrywa wszelkie interakcje użytkownika. Sensor możne być użyty do wygenerowania przerwania (GPIO), które obudzi urządzenie, zawiera ponadto szeregowy interfejs peryferyjny (SPI) i może wykrywać ruch w trzech osiach.

6-osiowy IMU BMI270 firmy Bosch to inteligentny moduł IMU o bardzo niskim poborze mocy, łączący w sobie 16-bitowy, trójosiowy żyroskop i 16-bitowy, trójosiowy akcelerometr. Czujnik integruje precyzyjny pomiar przyspieszenia i prędkości kątowej z inteligentnymi funkcjami przerwań wyzwalanych ruchem. BMI270 umożliwia intuicyjne rozpoznawanie gestów, kontekstu i aktywności, a to za sprawą zintegrowanemgo licznika kroków, działającego na zasadzie „plug-and-play".

Magnetometr BMM150 firmy Bosch to 3-osiowy cyfrowy czujnik geomagnetyczny. BMM150 zapewnia absolutną orientację przestrzenną i wektory ruchu z dużą dokładnością i dynamiką. Można go używać jako samodzielne urządzenie lub w połączeniu z 6-osiowym IMU BMI270. Magnetometr można podłączyć do pomocniczej magistrali I²C 6-osiowego IMU, aby zapewnić synchronizację próbkowania danych. Oznacza to, że dane magnetometru są próbkowane w tym samym czasie, co dane akcelerometru i żyroskopu. Może to być ważne szczególnie w przypadku zastosowań, związanych z fuzją danych z czujników.

Czujniki środowiskowe

Scalony układ czujnika otoczenia BME688 firmy Bosch (U9) z interfejsem I²C to pierwszy czujnik jakości powietrza ze sztuczną inteligencją (AI) i zintegrowanymi czujnikami ciśnienia, wilgotności i temperatury o wysokiej liniowości i doskonałej dokładności. Sensor został dokładnie opisany w artykule „Czujnik gazu Bosch BME688 ze Sztuczną Inteligencją" [18].

Czujnik koloru BH1749 firmy Rohm (U10) wykrywa światło czerwone, zielone, niebieskie (RGB) oraz podczerwień i konwertuje wyniki pomiarów na wartości cyfrowe. Wysoka czułość, szeroki zakres dynamiki i doskonała charakterystyka filtru blokującego podczerwień umożliwiają wykonywanie za pomocą omawianego układu dokładnych pomiarów oświetlenia i temperatury barwowej światła otoczenia. Czujnik znajduje się pod światłowodem dla LED RGB, który kieruje światło na czujnik.

Mikrofon i brzęczyk

Mikrofon PDM VM3011 firmy Vesper (U13) zbiera sygnał wejściowy audio. Układ ma również funkcję Wake-on-Sound, której można użyć do wybudzenia SoC nRF5340, gdy zdarzenie audio przekroczy ustawiony próg głośności.

W roli wyjściowego urządzenia audio moduł Nordic Thingy:53 ma brzęczyk piezoelektryczny (BZ1). Przetwornik jest sterowany przez tranzystor z wykorzystaniem wyjścia PWM.

Pamięć zewnętrzna

Nordic Thingy:53 ma wbudowaną zewnętrzną pamięć Flash o pojemności 64 Mb (U15). Układ MX25R6435F firmy Macronix jest pamięcią typu multi-I/O, obsługującą zwykły interfejs SPI oraz QSPI.

Diody i przyciski

Interfejs użytkownika Nordic Thingy:53 składa się z diody LED RGB (sterowanej tranzystorami) i dwóch przycisków. Duży przycisk (SW3), umieszczony w środku czarnej pokrywki, jest przeznaczony do dowolnego zaprogramowania przez użytkownika. Mały przycisk (SW2), znajdujący się tuż obok, umożliwia wprowadzenie procesora w tryb bootowania.

Zasilanie

Nordic Thingy:53 ma wbudowany układ scalony nPM1100 (PMIC) (U3) do zarządzania energią i ładowania akumulatora. Został on dokładnie opisany w artykule „Układ scalony zarządzania zasilaniem nPM1100" [17]. Układ nPM1100 PMIC oferuje dwa wyjścia napięcia (**rysunek 4**) [17]:

- Nieregulowane wyjście VSYS (3,2–4,2 V),
- Wyjście VOUTB (3,0 V).

Nieregulowane wyjście VSYS służy do zasilania zewnętrznego regulatora 3V3 i brzęczyka (BZ1). Stabilizator LDO 3V3 zasila

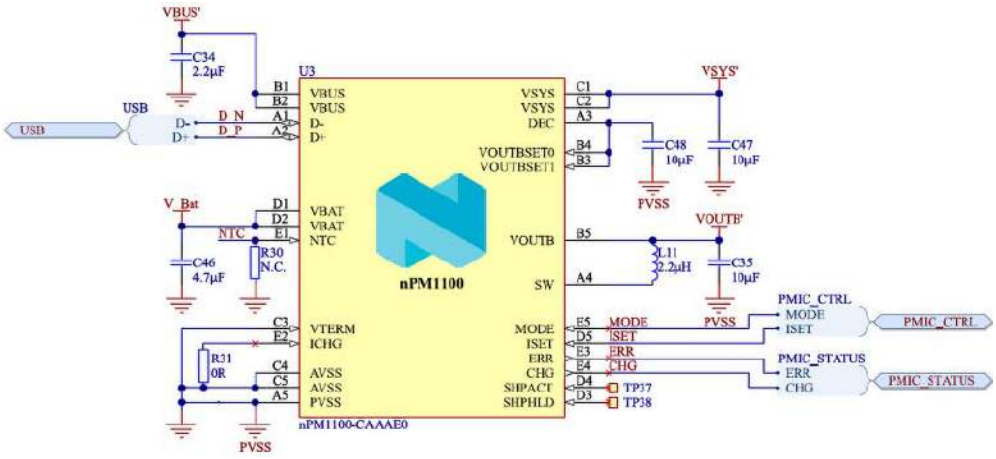
eprasa.pl 1fc8f31a49

diodę LED RGB (LD1) i wszelkie karty zewnętrzne, podłączone do złącza karty rozszerzeń (P1).

Domena VOUTB jest skonfigurowana na napięcie 3,0 V i służy do zasilania układów nRF5340, nRF21540 i wszystkich czujników.

Głównym źródłem zasilania Nordic Thingy:53 jest akumulator litowo-polimerowy o pojemności 1350 mAh. Można go ładować poprzez USB. Akumulator Li-Poly jest podłączony bezpośrednio do układu scalonego zarządzania zasilaniem nPM1100 (PMIC) (U3), który zapewnia ładowanie niezależnie od położenia wyłącznika zasilania (SW1). Limit prądu ładowarki akumulatora nPM1100 PMIC jest ustawiony na 400 mA, ale rzeczywisty prąd ładowania zależy od limitu prądu, pobieranego z łącza USB poprzez VBUS. nPM1100 PMIC automatycznie wykrywa, czy jest podłączony do standardowego portu pobierania danych (SDP), tzw. dedykowanego portu ładowania (DCP) lub portu pobierania danych i ładowania (CDP) uniwersalnej magistrali szeregowej (USB) i ustawia limit prądu VBUS zgodnie z typem portu USB.

W pozycji ON wyłącznika zasilania (SW1) układ nPM1100 PMIC dostarcza zasilanie do pozostałej części płytki poprzez linie VSYS i VOUTB. Gdy wyłącznik zasilania znajduje się w pozycji OFF, obie domeny napięcia nPM1100 PMIC są odłączone. Wtedy obwód odprowadza ładunek z kondensatorów, podłączonych do szyny zasilania VOUTB.



Rysunek 4. Układ scalony nPM1100 (PMIC) zarządzania energią i ładowania akumulatora [2]

Monitor napięcia akumulatora

Nordic Thingy:53 zawiera obwód monitorowania akumulatora, który umożliwia próbkowanie napięcia akumulatora przez przetwornik ADC układu nRF5340. Tranzystory Q7A i Q7B pozwalają na włączenie wykrywania napięcia akumulatora poprzez dzielnik rezystancyjny (R11 i R15).

Regulator 3,3 V

Nordic Thingy:53 może zasilać urządzenia zewnętrzne prądem do 300 mA poprzez stabilizator 3,3 V (U6), podłączony do wyjścia VSYS układu nPM1100.

Sterowanie zasilaniem czujników

Zasilanie czujników w Nordic Thingy:53 można wyłączyć za pomocą przełącznika TCK106 (U19), który jest sterowany z wyprowadzenia układu nRF5340.

Pomiar prądu

Do czteropunktowego pomiaru prądów w Nordic Thingy:53 można użyć znajdującej się w zestawie karty rozszerzeń PCA64142. Przełączniki obciążenia U4, U5, U11 i U14 typu TCK106 na płytce Nordic Thingy:53 są odłączone, gdy płytka pomiaru prądu i debugowania Nordic Thingy:53 jest włożona do gniazdka kart rozszerzeń (P9). Zamiast nich używane są wtedy złącza P1-P4, znajdujące się na płytce pomiaru prądu i umożliwiające próbkowanie następujących parametrów:

- Prądu (V_bat), płynącego do układu nPM1100 z akumulatora (VLi_Ion) przez złącze P1,
- Prądu (VDD_nRF53), płynącego do układu nRF5340 z szyny VOUTB przez złącze P2,
- Prądu (VDD_nRF21540), płynącego do układu nRF21540 z szyny VOUTB przez złącze P3,
- Prądu (VDD_EXP_BRD), płynącego do urządzeń zewnętrznych z regulatora 3V3 poprzez złącze P4.



Rysunek 5. Rozmieszczenie złącz Nordic Thingy:53 [3]

Interfejs programowania i debugowania

Nordic Thingy:53 zawiera złącze (10x, 1 mm) interfejsu programowania i debugowania (P8). Interfejs ten jest dostępny także na złączu interfejsu kart rozszerzeń (P9).

Interfejs kart rozszerzeń (pomiaru prądu i debugowania)

Zestaw Nordic Thingy:53 jest wyposażony w kartę do pomiaru prądu i debugowania (PCA64142), którą można włożyć do złącza krawędziowego kart rozszerzeń P9 płytki Nordic Thingy:53 (**rysunek 5** – szczelina pod wyłącznikiem zasilania). Płytka pomiaru prądu i debugowania upraszcza zadanie monitorowania poboru prądu w czterech różnych szynach zasilania za pomocą narzędzia do profilowania mocy, takiego jak zestaw Power Profiler Kit II. Interfejs umożliwia także podłączenie zewnętrznych obwodów lub płytek do Nordic Thingy:53 poprzez użycie wyprowadzeń TRACE jako zwykłych wejść/wyjść ogólnego przeznaczenia (GPIO). Piny są domyślnie skonfigurowane jako GPIO. Zewnętrzne obciążenia można podłączyć do napięć VOUTB lub 3V3. Maksymalne obciążenie VOUTB wynosi 150 mA, a najwyższe dozwolone obciążenie przez obwody zewnętrzne na szynie VOUTB zależy od całkowitego wewnętrznego poboru prądu. Maksymalne obciążenie szyny 3V3 wynosi 300 mA.

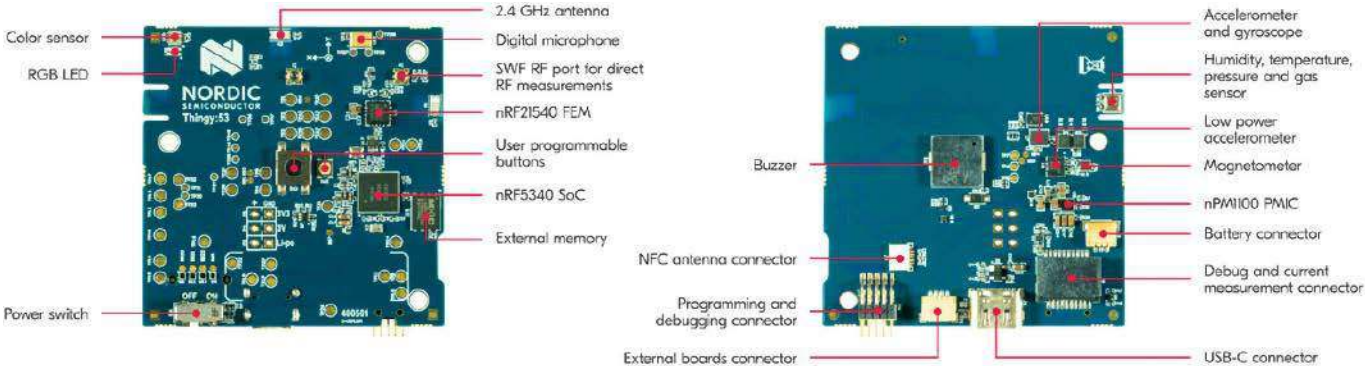
Interfejs modułów zewnętrznych

Złącze P1 na płytce Nordic Thingy:53 może być używane do podłączania modułów zewnętrznych i jest kompatybilne z Qwiic oraz STEMMA QT (raster 1 mm). Złącze P1 jest kompatybilne również z modułami prototypowymi STEMMA, Grove i DFRobot Gravity, ale wymagają one kabla do konwersji ze złącza o rozstawie 1 mm na złącze o rozstawie 2 mm.

Złącza, punkty testowe i mostki lutownicze

Nordic Thingy:53 ma szereg złączy z wyprowadzonymi szynami zasilania, kilka punktów testowych oraz szereg mostków lutowniczych, umożliwiających włączanie i wyłączanie poszczególnych funkcjonalności. Zmiany w nich nie są potrzebne do normalnego użytkowania Nordic Thingy:53.

eprasa.pl 1fc8f31a49



Rysunek 6. Rozmieszczenie elementów na płytce Nordic Thingy:53 (widok z góry i z dołu) [3]

Konstrukcja

Na czerwonej klapce z boku obudowy znajdują się ikonki złączy, umieszczonych pod nią, od lewej: wyłącznik zasilania (pod nim znajduje się szczelina interfejsu kart rozszerzeń), złącze USB-C, złącze interfejsu modułów zewnętrznych oraz złącze interfejsu kart rozszerzeń (pomiaru prądu i debugowania). Na środku czarnej obudowy (góra) producent umieścił przycisk użytkownika (rysunek 5).

Ułożenie elementów na płytce Nordic Thingy:53 w widokach z góry i z dołu pokazano na **rysunku 6**.

Dokumentacja

Dokumenty udostępnione przez Nordic Semiconductor w postaci plików PDF to: „Nordic Thingy:53 Hardware User Guide" [2] oraz „Nordic Thingy:53 Product Brief" [3]. Pozostałe materiały są dostępne w postaci stron internetowych. Najważniejsze informacje zawiera strona produktu „Nordic Thingy:53 IoT prototyping platform" [1], zawierająca linki do kolejnych stron dokumentacji: „Nordic Thingy:53 Hardware description" [4], „Getting started with Thingy:53" [5] oraz „Developing with Thingy:53" [6]. Ponadto do dyspozycji użytkowników producent oddał też portal wsparcia technicznego „DevZone" [10].

Na zakładce *Downloads* strony produktu [1] jest dostępny plik z dokumentacją techniczną, w tym schematy sprzętowe i obrazy warstw płytki drukowanej wraz z oryginalnym projektem dla środowiska Altium Designer. W tym samym miejscu można również znaleźć plik z prekompilowaną najnowszą wersją oprogramowania firmowego zestawu.

Rozpoczęcie korzystania z Thingy:53 zostało dobrze opisane w raporcie z praktycznych eksperymentów [12], a praca z modelami ML oraz zastosowanie aplikacji komputerowych i mobilnych opisano w następnych eksperymentach [13, 14]. Bardzo dobrym wstępem do zapoznania się z zestawem jest zapis wideo webinariów firmowych [15]. Doskonałe wprowadzenie do uczenia maszynowego stanowi strona „nRF Machine Learning" [11].

Bardzo ciekawą inicjatywą jest także „Nordic Developer Academy" – interaktywna platforma edukacyjna online, która zapewnia programistom informacje techniczne i know-how, dotyczące tworzenia produktów bezprzewodowych przy użyciu rozwiązań Nordic Semiconductor [19]. W oparciu o Thingy:53 można realizować program kursu „nRF Connect SDK Fundamentals" – praktycznego, samodzielnego szkolenia online, skupiającego się na nauce podstaw tworzenia oprogramowania sprzętowego.

Aplikacja mobilna nRF Edge Impulse

Każdy Nordic Thingy:53 ma fabrycznie zainstalowany program „Edge Impulse firmware", zawierający domyślny model rozpoznawania ruchu [16]. Program jest kompatybilny z aplikacją mobilną nRF Edge Impulse oraz środowiskiem programowym Edge Impulse Studio.

Aplikacja mobilna nRF Edge Impulse została opracowana we współpracy z firmą Edge Impulse w celu szkolenia i wdrażania wbudowanych modeli uczenia maszynowego na Nordic Thingy:53 przy użyciu Edge Impulse Studio. Aplikacja umożliwia przesyłanie nieprzetworzonych danych z czujnika za pośrednictwem urządzenia mobilnego do opartego na chmurze Edge Impulse Studio i wdrażanie w pełni wytrenowanych modeli ML w Nordic Thingy:53 przez Bluetooth Low Energy (BLE). Aplikacja pozwala na wybór pomiędzy istniejącymi projektami w danych Edge Impulse Studio lub utworzeniem nowego projektu. Umożliwia też zbieranie danych z czujników. Zgromadzone bazy dane można oznaczyć jako treningowe lub testowe i korzystać z nich bezpośrednio w aplikacji za pomocą modelu uczenia maszynowego, wpisanego do Nordic Thingy:53.

Pierwsze kroki

Na zakładce „Get Started" strony produktu [1] opisanych zostało osiem kroków, niezbędnych do rozpoczęcia pracy z oprogramowaniem firmowym zestawu Nordic Thingy:53. Opis ten rozmija się z aktualną wersją aplikacji. Na szczęście istnieje też dokumentacja, udostępniona przez firmę Edge Impulse na stronie „Nordic Semi Thingy:53" [16].

Rozpoczęcie korzystania z Thingy:53 jest dość łatwe. Można wypróbować wbudowany model Edge Impulse, który jest domyślnie zainstalowany fabrycznie we wszystkich Thingy:53. Zasadniczo jest to model wykrywania ruchu, który informuje o stanie bezczynności, ruchu węża, ruchu w górę i w dół oraz ruchu falowego [12].

Aby samodzielnie przetestować działanie omawianego ekosystemu, należy wykonać następujące kroki.

1. Pobierz i zainstaluj na smartfonie aplikację „nRF Edge Impulse" ze sklepu Play lub App Store.
2. Załóż bezpłatne konto w aplikacji lub na stronie Edge Impulse.
3. Otwórz klapkę i przesuń wyłącznik zasilania do pozycji „I". Zobaczysz migające niebieskie i zielone światło.
4. Włącz obsługę transmisji Bluetooth na smartfonie.
5. Uruchom aplikację.
6. Zaloguj się do swojego konta.
7. Utwórz nowy projekt –> Wprowadź nazwę swojego projektu (**rysunek 7a**).
8. W dodanych urządzeniach zobaczysz nazwę „Edge Impulse" (rysunek 7b). Kliknij na pole nazwy, a urządzenie zostanie połączone. Po podłączeniu Thingy:53 będzie świecić stale czerwonym światłem.

Uwaga: Przy następnym uruchomieniu aplikacji pokazywane są (na niebiesko) aktywne zestawy Nordic Thingy:53 z programem Edge Impulse. W tym momencie widoczna będzie nazwa Edge Impulse, a także adres sieci Bluetooth. Kliknij na pole nazwy zestawu i w następnym oknie wybierz przycisk *Connect*.

Testowanie modelu domyślnego

Na dole okna aplikacji znajduje się menu, udostępniające cztery sposoby pracy (rysunek 7c):

- Zakładka *Devices* (wybrana) pokazuje ustawienia zestawu i wszystkich czujników,
- Zakładka *Data* umożliwia zbieranie danych jako treningowe lub testowe,

eprasa.pl 1fc8f31a49

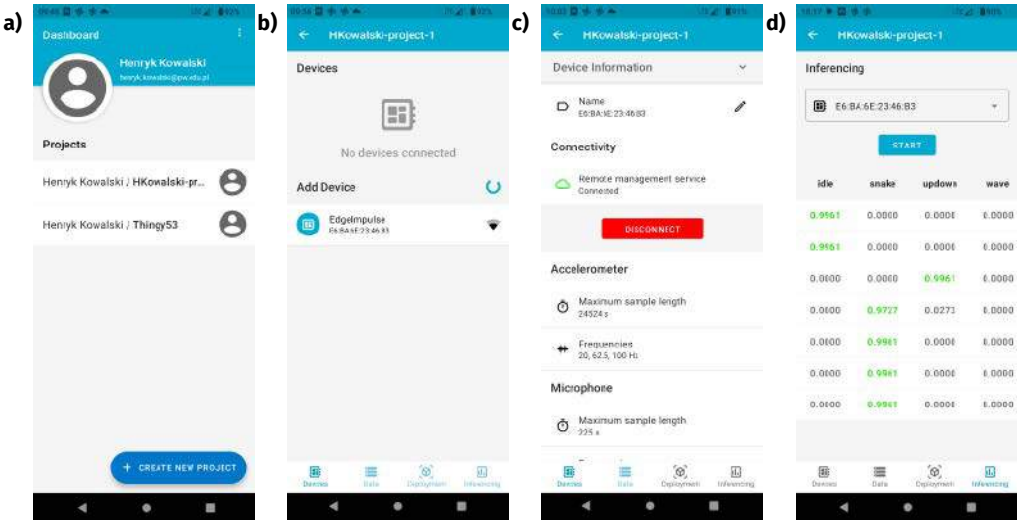
- Zakładka *Deployment* umożliwia prace, z modelami ML,
- Zakładka *Inferencing* (wnioskowanie) pozwala na zbieranie danych i wnioskowanie.

9. Wybierz zakładkę *Inferencing* i kliknij na przycisk START.

Jeśli Twój Thingy:53 będzie leżał stabilnie, to zostanie wyświetlony rezultat wnioskowania „bezczynność", wykonany przez Thingy:53 (rysunek 7d).

Niebieska dioda LED będzie migać za każdym razem, gdy układ wyśle odpowiedź do aplikacji Edge Impulse.

10. Poruszaj zestawem (dosyć energicznie) do góry i w dół, a następnie na boki.
11. Zmiany wartości zobaczysz w aplikacji. W oknie aplikacji wyświetlane są rezultaty wnioskowania dla kolejnych porcji danych (rysunek 7d).
12. Po zakończeniu testowania kliknij przycisk STOP.



Rysunek 7. Okna aplikacji nRF Edge Impulse podczas pracy

Opis aktualnej wersji aplikacji jest dostępny na stronie „Nordic Semi Thingy:53" [16]. Udokumentowana została też aktualizacja oprogramowania firmowego Thingy:53 w sposób bezpośredni (USB) i z zastosowaniem aplikacji mobilnej nRF Programmer (BLE).

Podsumowanie

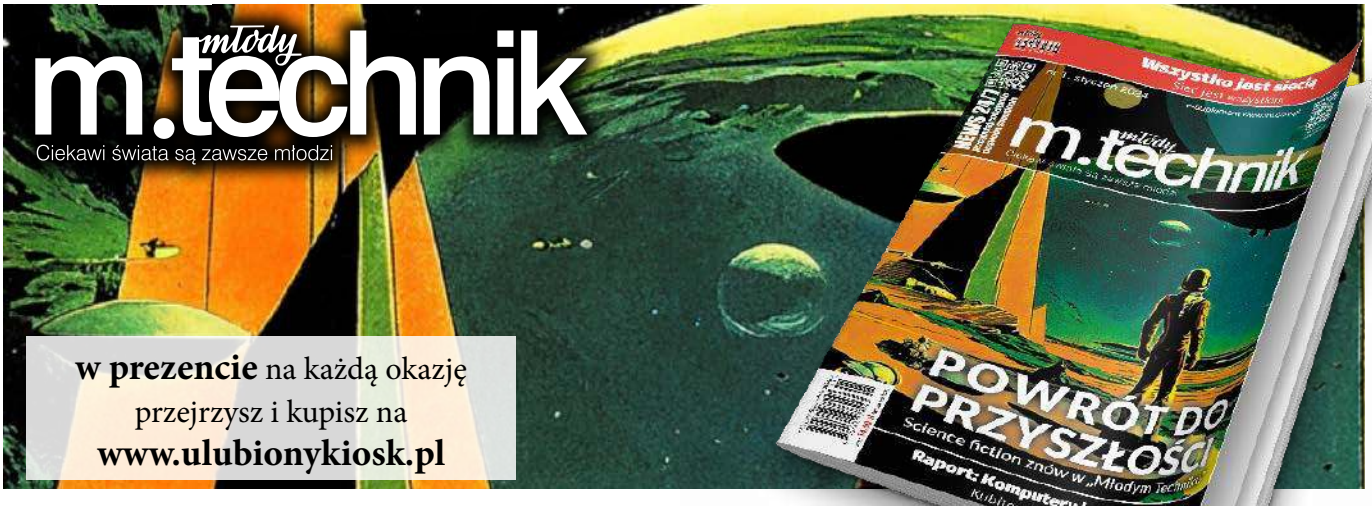
Nordic Thingy:53 firmy Nordic Semiconductor udostępnia łatwy sposób rozpoczęcia prawdziwej pracy z zastosowaniem uczenia maszynowego, wykonywanego bezpośrednio na urządzeniu IoT. Zainstalowany fabrycznie program z modelem rozpoznawania gestów pracuje „prosto z pudełka". Zastosowane w urządzeniu najnowsze czujniki pozwalają na rejestrowanie danych wysokiej jakości. Zagadnienia dotyczące programowego zastosowania Thingy:53 są na tyle obszerne, że zostaną omówione osobno.

Henryk A. Kowalski
Instytut Informatyki
Politechnika Warszawska

Literatura:

[1] Nordic Thingy:53 IoT prototyping platform, Nordic Semiconductor, https://t.ly/kgyJQ
[2] Nordic Thingy:53 Hardware, User Guide, 2023-04-12, https://t.ly/Jgtkg
[3] Nordic Thingy:53 Product Brief Version 1.1, https://t.ly/sY5iN
[4] Nordic Thingy:53 Hardware description, 2023-04-12 https://t.ly/Nmrpv
[5] Getting started with Thingy:53, Nov 09, 2023, https://t.ly/_5yEI
[6] Developing with Thingy:53, Nordic Semiconductor, Nov 09, 2023, https://tiny.pl/cspvf
[7] nRF5340 System-on-Chip, https://tiny.pl/cspvd
[8] nRF21540 RF FEM, https://tiny.pl/cspv5
[9] nPM1100 Power Management IC, https://tiny.pl/cspv1
[10] DevZone, https://devzone.nordicsemi.com/
[11] nRF Machine Learning, nRF Connect SDK 2.5.99, Nov 09, 2023, https://tiny.pl/cspvj
[12] Roadtest Product Review of the Nordic Thingy:53 with Edge Impulse, 27 Feb 2023, https://tiny.pl/cspvp
[13] Nordic Thingy:53 review – Basics simple projects, https://tiny.pl/cspvl
[14] ROADTEST REVIEW: Nordic Semiconductor Thingy:53 development platform, 26 Mar 2023, https://tiny.pl/cspv4
[15] Say hello to Thingy:53 – The prototyping platform for embedded ML, Nordic Semiconductor, July 08, 2022, https://tiny.pl/cspvn
[16] Nordic Semi Thingy:53, 2023, https://tiny.pl/cspvk
[17] *Układ scalony zarządzania zasilaniem nPM1100*, EP 3/2022, https://tiny.pl/cspv2
[18] Czujnik gazu Bosch BME688 ze Sztuczną Inteligencją, EP11/2021, https://tiny.pl/cspv8
[19] Nordic Developer Academy, Nordic Semiconductor, https://tiny.pl/cspvs

REKLAMA



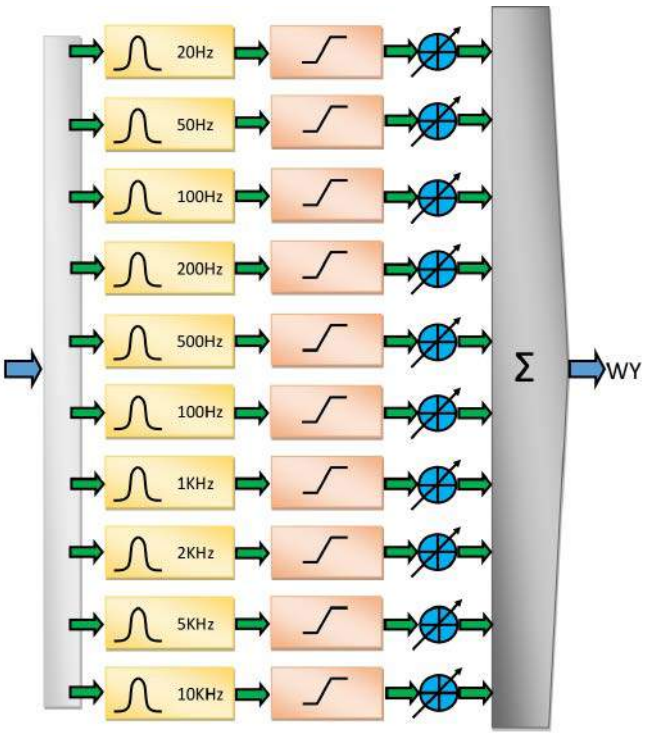
eprasa.pl 1fc8f31a49



Efekt do gitary na bazie LPC5528

W artykule zaprezentuję kolejne zastosowanie płytki ewaluacyjnej firmy NXP demonstrujące działanie mikrokontrolera serii LPC55xx. Jest to najbardziej popularny efekt do gitary elektrycznej o nazwie Fuzz Graficzny. Płytka ewaluacyjna wyjątkowo dobrze nadaje się do tego zastosowania, ponieważ jest wyposażona w 16-bitowy kodek WM8904 oraz procesor LPC5528 zawierający odpowiednie interfejsy do komunikacji z nim, a dołączenie graficznego wyświetlacza powoduje, że używanie urządzenia jest bardzo wygodne.

Efekt do gitary elektrycznej o nazwie Fuzz Graficzny jest prawie zawsze używany przy muzyce rockowej, jego brzmienie jest łagodniejsze od klasycznego efektu „distortion" i zazwyczaj stosowany jest do gitary prowadzącej, może być jednak używany przez gitary solowe. Realizacja efektu polega na tym, że sygnał rozdzielony jest na wiele pasm częstotliwości, a w każdym z nich znajduje się zwykły „fuzz" klasyczny. Sposób działania tego efektu pokazano na **rysunku 1**.

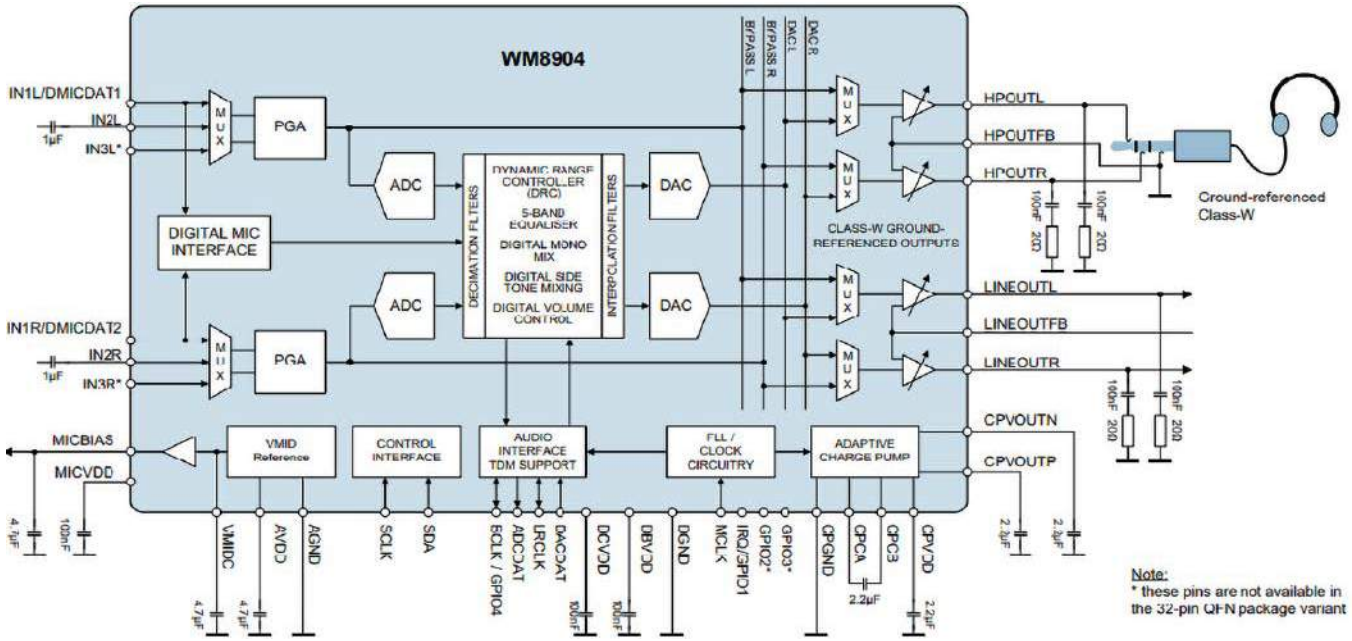


Rysunek 1. Schemat przedstawiający działanie efektu Fuzz Graficzny

Działanie urządzenia

Pierwszy etap polega na przetworzeniu sygnału analogowego na cyfrowy za pomocą przetwornika znajdującego się w układzie kodeka. Następnie sygnał jest przetwarzany przez zespół

eprasa.pl 1fc8f31a49



Rysunek 2. Schemat blokowy kodeka

filtrów cyfrowych, i w każdym kanale wytwarzane są zniekształcenia. Tworzenie zniekształceń polega na obcięciu chwilowej amplitudy sygnału. Następnie kanały są sumowane i sygnał cyfrowy wysyłany jest do przetwornika analogowo-cyfrowego zawartego w układzie kodeka.

Opis programu komputerowego

Cały program składa się z dwóch części: procedur przetwarzania sygnału, w skład których wchodzi również komunikacja z układem kodeka oraz z interfejsu użytkownika, który jest odpowiedzialny za ustawianie parametrów urządzenia i obsługuje przyciski sprzętowe, enkoder oraz steruje wyświetlaczem graficznym.

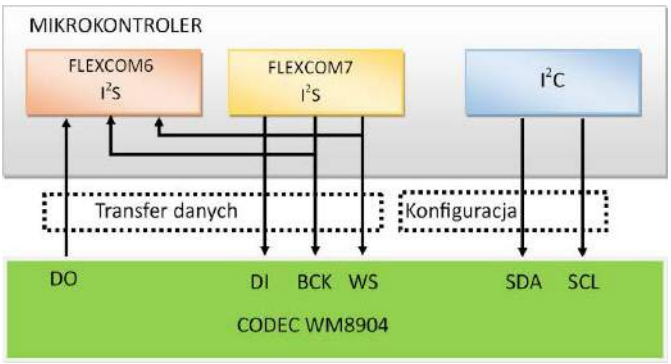
Część DSP – sterowanie kodekiem

Sterowanie układem WM8904 odbywa się z użyciem dwóch interfejsów: I²C oraz I²S. Magistrala I²C służy do konfiguracji urządzenia poprzez programowanie wewnętrznych rejestrów procesora dźwięku. Można w ten sposób zaprogramować: wzmocnienie wewnętrznego wzmacniacza PGA, źródła sygnału wejściowego i wyjściowego, częstotliwość próbkowania itp. Interfejs I²S służy do przesyłania próbek dźwiękowych.

W naszym programie używamy gotowej biblioteki służącej do konfiguracji kodeka. Obsługa transmisji danych odbywa się poprzez dwa interfejsy I²S. Zastosowany mikrokontroler dostarcza system dziewięciu uniwersalnych bloków komunikacji szeregowej, nazwanych FLEXCOM, każdy z nich można skonfigurować jako jeden z interfejsów do transmisji danych (UART, USART, SPI, I²C, I²S). W naszym układzie korzystamy z FLEXCOM6 i FLEXCOM7, gdyż są one elektrycznie podłączone do układu WM8904. Interfejs I²S zaimplementowany na module FLEXCOM6 jest używany jako wejście sygnału akustycznego, natomiast transmisja odbywająca się poprzez FLEXCOM7 wysyła dane do przetwornika DAC, czyli pełni funkcję wyjścia. Ponieważ sygnały sterujące transmisją (BCK i WS) są takie same dla obydwu interfejsów, należy dokonać ich połączenia. Służy do tego moduł SYSCTL.

Program ma funkcję regulacji sygnału wyjściowego oraz ustawienia typu wejścia – wejście mikrofonowe lub wejście liniowe. Rodzaj wejścia wybiera się poprzez zmianę wzmocnienia wewnętrznego wzmacniacza PGA (*Programable Gain Amplifier*). Obydwa te parametry można zmieniać poprzez ustawianie wartości w rejestrach układu kodeka.

Do działania kodeka konieczne jest dostarczenie sygnału taktującego na wejście MCLK. Sygnał ten wytwarzany jest przez



Rysunek 3. Realizacja transmisji pomiędzy MPU, a kodekiem

mikrokontroler i wyprowadzony na odpowiednio skonfigurowane wyjście MCLK. Częstotliwość sygnału wynosi 24 576 000 Hz. Jest to zalecana wartość częstotliwości, która umożliwia wytworzenie wymaganych przebiegów sterujących dla standardowych prędkości próbkowania. Na **rysunku 2** pokazano strukturę wewnętrzną układu kodeka WM8904, natomiast **rysunek 3** obrazuje transmisję pomiędzy kodekiem a mikrokontrolerem.

Opis algorytmu DSP

Sygnał, który przeszedł próbkowanie, jest poddawany cyfrowej filtracji. Jako filtry zastosowałem cyfrowe imitacje szeregowych obwodów rezonansowych. Do ich zaprojektowania użyłem następujące zależności:

$$u_c = \frac{1}{C}\int i(t)\cdot dt$$

$$i_L = \frac{1}{L}\int u_L(t)\cdot dt$$

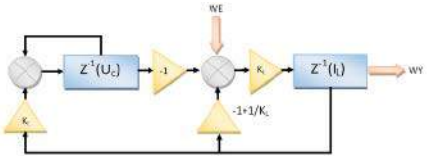
$$u_{we} = u_r + u_C + u_L$$

gdzie:

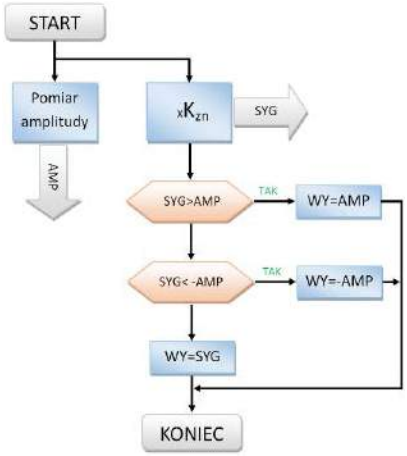
- UR – napięcie na rezystorze,
- UC – napięcie na kondensatorze,
- UL – napięcie na cewce.

Wartości L i C można wyliczyć na podstawie częstotliwości rezonansowej oraz wymaganej dobroci filtru. Natomiast wartość rezystancji najwygodniej przyjąć równą jeden. Struktura tego filtru zamieszczona jest na **rysunku 4**.

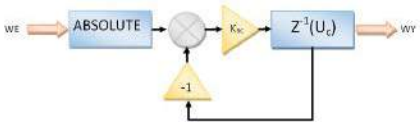
eprasa.pl 1fc8f31a49



Rysunek 4. Schemat blokowy filtru pasmowego

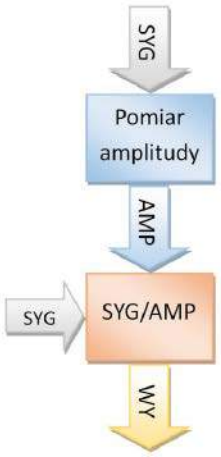


Rysunek 5. Działanie algorytmu wprowadzania zniekształceń



Rysunek 6. Algorytm układu pomiaru amplitudy

Na **listingu 1** zamieszczam fragment kodu odpowiedzialny za przetwarzanie sygnału akustycznego. Schemat algorytmu wprowadzającego zniekształcenia pokazany jest na **rysunku 5**. Jednym z elementów tego algorytmu jest układ pomiaru amplitudy. Algorytm jego działania jest pokazany na **rysunku 6**. Zasada jego działania jest bardzo prosta. Polega na wyliczeniu wartości bezwzględnej sygnału, a następnie poddaniu go działaniu filtru dolnoprzepustowego. Natomiast zasada działania układu kompresora (służy on do utrzymywania stałej wartości sygnału na wyjściu) jest pokazana na **rysunku 7**.



Rysunek 7. Zasada działania kompresora

Aby wykorzystać maksymalną częstotliwość procesora, procedura DSP została umieszczona w pamięci operacyjnej, a wszystkie powtarzające się procedury zostały napisane w postaci makr (listing 1).

Interfej GUI

Do opracowania interfejsu użytkownika zastosowano zaprojektowaną przez autora bibliotekę, która współpracuje z wyświetlaczem graficznym LCD o rozdzielczości 128×64 pikseli. Umożliwia ona utworzenie na ekranie siedmiu kontrolek:

- „Check box” – obiekt służy do wyboru dwóch stanów;

Listing 1. Fragment kodu odpowiedzialny za przetwarzanie sygnału akustycznego

```
#include „app.h”
#include „dsp.hpp”

#define T_PA  ((float)100/FP)

#define PomiarAmplitudy(n,x)\
    static float n##BA = 0;\
    static float n##C1 = 0;\
    n##C1 += (((x<0) ? -x :x)-n##C1)*T_PA;\
    n##BA += (n##C1-n##BA)*T_PA;

#define Filtr(f,q,x)\
        static float uc##f = 0;\
        static float IL##f = 0;\
        uc##f += IL##f * (float)((float)2 * STALA_PI / FP * (float)q * (float)f);\
        IL##f += (x – uc##f – IL##f) * (float)((float)2 * STALA_PI / FP * (float)f / (float)q);\

SParFuzz ParFuzz = {0.5,0.5,0.05,0.5,0.5,0.5,0.5,0.5,0.5,0,0,0,0,true};

void __attribute__((section(„RamFunction”))) PrzetwarzanieGlowne()
{
    volatile register float wl,wp;
    wl = (float)WartoscWE_L/(float)0x7FFF;
    wp = (float)WartoscWE_P/(float)0x7FFF;

    volatile register float ww = 0, wsp, ba, we = wl+wp, fk = 1/ParFuzz.Odc;

    Filtr(21,1,we);
    PomiarAmplitudy(P20_,IL21);
    ba = P20_BA;
    wsp = IL21*fk;
    ww += ((ba<wsp) ? ba : ((-ba>wsp) ? -ba : wsp))*ParFuzz.A20;

    Filtr(46,1,we);
    PomiarAmplitudy(P50_,IL46);
    ba = P50_BA;
    wsp = IL46*fk;
    ww += ((ba<wsp) ? ba : ((-ba>wsp) ? -ba : wsp))*ParFuzz.A50;

    Filtr(100,1,we);
    PomiarAmplitudy(P100_,IL100);
    ba = P100_BA;
    wsp = IL100*fk;
    ww += ((ba<wsp) ? ba : ((-ba>wsp) ? -ba : wsp))*ParFuzz.A100;

    Filtr(215,1,we);
    PomiarAmplitudy(P200_,IL215);
    ba = P200_BA;
    wsp = IL215*fk;
    ww += ((ba<wsp) ? ba : ((-ba>wsp) ? -ba : wsp))*ParFuzz.A200;

    Filtr(464,1,we);
    PomiarAmplitudy(P500_,IL464);
    ba = P500_BA;
    wsp = IL464*fk;
    ww += ((ba<wsp) ? ba : ((-ba>wsp) ? -ba : wsp))*ParFuzz.A500;

    Filtr(1000,1,we);
    PomiarAmplitudy(P1K_,IL1000);
    ba = P1K_BA;
    wsp = IL1000*fk;
    ww += ((ba<wsp) ? ba : ((-ba>wsp) ? -ba : wsp))*ParFuzz.A1K;

    Filtr(2150,1,we);
    PomiarAmplitudy(P2K_,IL2150);
    ba = P2K_BA;
    wsp = IL2150*fk;
    ww += ((ba<wsp) ? ba : ((-ba>wsp) ? -ba : wsp))*ParFuzz.A2K;

    Filtr(4640,1,we);
    PomiarAmplitudy(P5K_,IL4640);
    ba = P5K_BA;
    wsp = IL4640*fk;
    ww += ((ba<wsp) ? ba : ((-ba>wsp) ? -ba : wsp))*ParFuzz.A5K;

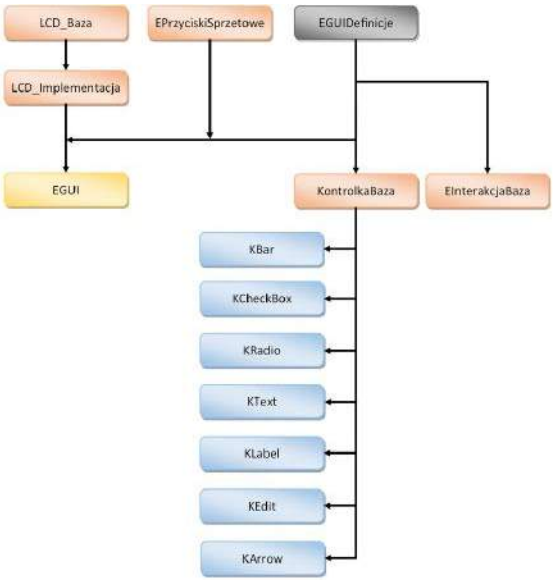
    Filtr(10000,1.5,we);
    PomiarAmplitudy(P10K_,IL10000);
    ba = P10K_BA;
    wsp = IL1000*fk;
    ww += ((ba<wsp) ? ba : ((-ba>wsp) ? -ba : wsp))*ParFuzz.A10K;

    ww*=2;

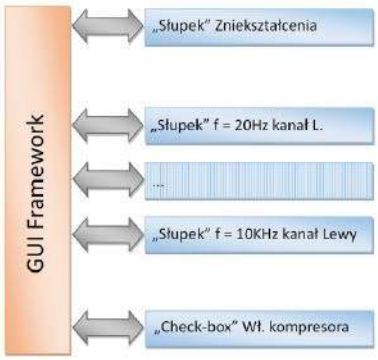
    if(ww>1)
        ww = 1;
    if(ww<-1)
        ww=-1;

    WartoscWY_P = WartoscWY_L = (int16_t)(ww*(float)0x7FFF);
}
```

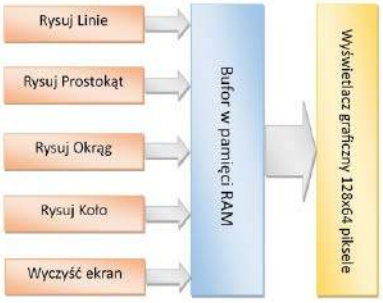

eprasa.pl 1fc8f31a49



Rysunek 8. Hierarchia klas interfejsu GUI



Rysunek 9. Schemat poglądowy struktury naszego interfejsu

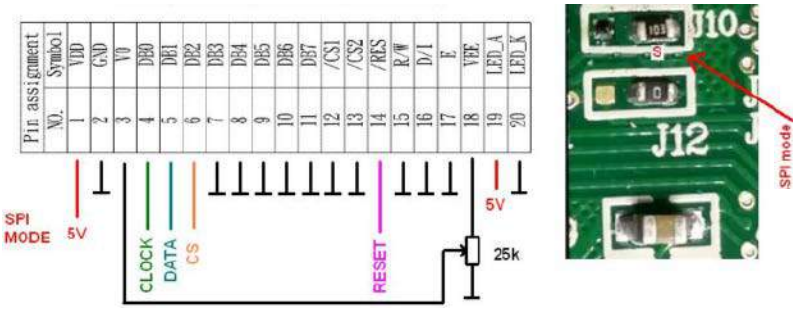


Rysunek 10. Koncepcja działania interfejsu graficznego obsługującego wyświetlacz LCD

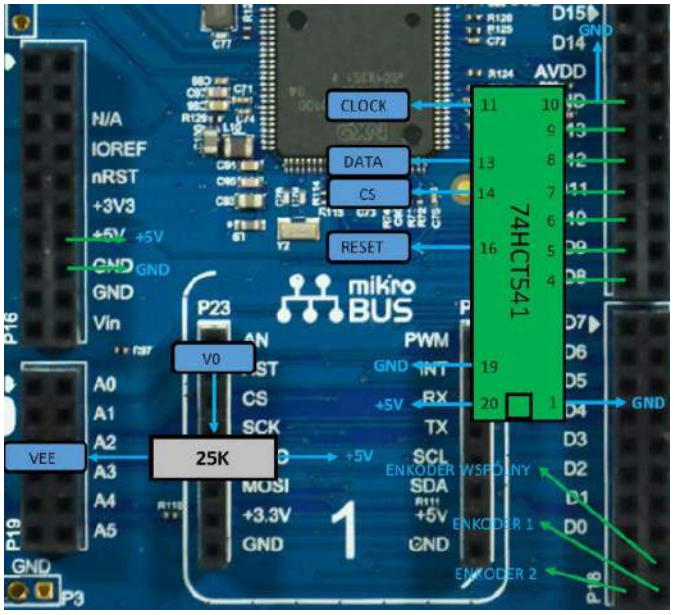
- „Radio Button" – ta kontrolka umożliwia wybór jednej z wielu opcji;
- „Text" – wyświetla tekst w ramce na ekranie i może być traktowana jako przycisk;
- „Label" – tylko wyświetla tekst;
- „Bar" – wyświetla słupek o zmieniającej się wysokości. Kontrolka ta jest używana do ustawiania wartości;
- „Edit" – ten obiekt umożliwia edycję napisu;
- „Arrow" – wyświetla strzałkę na ekranie.

Główną klasą do obsługi interfejsu użytkownika jest *EGUI*. Umożliwia ona wyświetlanie treści na ekranie oraz odczyt wartości z przycisków i enkodera. Również za pomocą jej funkcji składowych można do interfejsu dodawać okna kontrolne wywodzące się z klasy *EKontrolkaBaza*. Na **rysunku 8** pokazano hierarchię klas, natomiast na **rysunku 9** pokazano strukturę interfejsu GUI naszego programu.

Interfejs GUI korzysta z biblioteki graficznej, również opracowanej przez autora, służącej do wyświetlania różnych kształtów na ekranie wyświetlacza. Koncepcję tej biblioteki obrazuje **rysunek 10**.



Rysunek 11. Schemat podłączenia wyświetlacza



Rysunek 12. Schemat części elektrycznej

Opis układu elektrycznego

Chociaż układ zbudowany jest na bazie gotowego modułu z mikrokontrolerem firmy NXP do jego działania potrzebnych jest kilka elementów dodatkowych. Są to: wyświetlacz graficzny 128×64 piksele oraz układ do jego sterowania. Można również do modułu dołączyć dwa przyciski, chociaż nie jest to konieczne, ponieważ do sterowania urządzeniem można używać przycisków znajdujących się na płytce ewaluacyjnej.

Zastosowany w naszym układzie wyświetlacz graficzny może pracować w dwóch trybach. Może być sterowany za pomocą transmisji równoległej lub szeregowej w protokole SPI. W naszym układzie stosujemy tryb transmisji szeregowej. Ponieważ wyświetlacz fabrycznie jest skonfigurowany do pracy przy transmisji równoległej, należy przełączyć zworę, która znajduje się na jego tylnej części (**rysunek 11**). Dodatkowo, ponieważ układ logiczny wyświetlacza jest sterowany napięciami niezgodnymi z logiką standardu 3,3 V, czyli wartościami wyjściowymi mikrokontrolera (wysoki stan wejść wymaga napięcia wyższego niż 3,5 V), zastosowano bufor 74HCT541. Schemat części elektrycznej pokazany jest na **rysunku 12**.

Elementy zastosowane w układzie

Wyświetlacz monochromatyczny, graficzny LCD typu LCD-EG-128064H-FHW K/W-E6 daje dobre wrażenie. Charakteryzuje się niską ceną i jest dostępny w wielu sklepach internetowych m.in. ARTRONIC czy KAMAMI. Jako bufor służy układ 74HCT541. Potencjometr do regulacji kontrastu może być dowolnym potencjometrem montażowym o rezystancji około 25 kΩ. Jako enkoder można zastosować dowolny 3-wyjściowy enkoder inkrementalny.

Tomasz Krogulski
krogul70@gmail.com

eprasa.pl 1fc8f31a49

QWERTY

Twój sprawdzony dostawca klawiatur membranowych i paneli dotykowych

Klawiatury membranowe, szklane panele dotykowe czy zintegrowane panele czołowe stanowią podstawowy element konstrukcji tysięcy urządzeń w różnych branżach współczesnej techniki. Wieloletnie doświadczenie i doskonale wyposażony park maszynowy naszej firmy umożliwiają efektywną realizację nawet najbardziej złożonych zamówień w zakresie elektromechanicznych interfejsów HMI oraz paneli czołowych, zintegrowanych z wyświetlaczami dotykowymi.



Oferta firmy QWERTY obejmuje zarówno produkcję, jak i projektowanie klawiatur i paneli czołowych zgodnie z otrzymanymi od klienta wytycznymi. Kompleksowe usługi prowadzone przez nasze biuro konstrukcyjne obejmują ponadto tworzenie kompletnych urządzeń „od A do Z". Nasze wyroby przeznaczone są do maszyn i sterowników przemysłowych, aparatury medycznej i pomiarowej. Do ich wytwarzania stosujemy najnowsze technologie projektowania i produkcji oraz surowce i materiały najwyższej jakości. Dysponujemy wykwalifikowanym personelem, który dołoży wszelkich starań, by gruntownie i profesjonalnie zaspokoić Państwa potrzeby. Zespół specjalistów zaproponuje rozwiązania optymalne zarówno w sferze technologicznej, jak i ekonomicznej.

Szklane panele dotykowe

We współczesnych sterownikach elektronicznych modnym i nowoczesnym rozwiązaniem jest zastosowanie szklanej elewacji frontowej, połączonej z pojemnościowym panelem sterującym. Płyty szklane produkowane są na zamówienie klienta. Panel przedni wykonany ze szkła jest odporny na działanie czynników chemicznych i klimatycznych, a zastosowanie odpowiedniego materiału gwarantuje zarówno estetykę, jak i wysoką wytrzymałość mechaniczną. Grafikę, której forma jest praktycznie dowolna – w zależności od preferencji i potrzeb klienta – nadrukowuje się na spodniej stronie płyty szklanej, co gwarantuje jej odporność na ścieranie przy późniejszej eksploatacji. Integralną częścią takiego panelu jest ekran dotykowy, który zastępuje klawiaturę i umożliwia adaptacyjne sterowanie elektroniką urządzenia – co ważne, do aktywacji panelu pojemnościowego nie jest potrzebny najmniejszy nawet nacisk. Wystarcza zbliżenie do jego powierzchni lub delikatne przesunięcie palcem po powierzchni.
Zamiast tradycyjnej szyby możemy także zastosować płytę z tworzywa sztucznego, która spełnia wysokie normy IK. Daje ona wysoki poziom zabezpieczenia przed uszkodzeniami z zewnątrz.

Bonding optyczny w standardzie QWERTY

W ostatnich latach obserwujemy na rynku wzrost zapotrzebowania na sztywne panele frontowe, wykonane ze szkła lub tworzywa sztucznego. Firma QWERTY postanowiła wykorzystać swoje wcześniejsze doświadczenia z zakresu laminowania różnego rodzaju powierzchni (elastyczne-elastyczne, elastyczne-sztywne) do opracowania i wdrożenia do produkcji nowej technologii laminowania sztywnych materiałów, zwanej bondingiem optycznym. Bonding to proces zespolenia wyświetlacza (LCM) z panelem dotykowym i frontem ze szkła lub tworzywa. Wyróżniamy kilka metod bondingu.

Tabela 1. Parametry użytkowe klawiatur foliowych produkcji Qwerty

Lp.	Wyszczególnienie	Klawiatury foliowe standardowe płaskie	Klawiatury foliowe przetłoczone lub z blaszkami
1	Gwarantowana liczba zadziałań zestyku obciążonego prądem 25 mA	3×107	5×106
2	Rezystancja ścieżki o szerokości 1,25 mm	10 Ω/cm	10 Ω/cm
3	Rezystancja zestykowa klawiszy (wynika z konstrukcji klawiatury i wymagań odbiorcy)	od 20 do 1000 Ω	
4	Rezystancja izolacji	10 MΩ/cm	
5	Napięcie na zestyku (maksymalne)	30 V	
6	Prąd zestyku (maksymalny)	25 mA	
7	Dopuszczalna wilgotność otoczenia	95%	
8	Temperatura pracy	od –40°C do 70°C	
9	Temperatura przechowywania	od –40°C do 80°C	
10	Format elewacji (maksymalny)	350×700 mm	
11	Promień zagięcia paska	R>2 mm	
12	Zasilanie diod LED	od 2 do 30 V	
13	Zasilanie lamp elektroluminescencyjnych	130 V, 400 Hz	
14	Siła zadziałania klawiszy	>0,5 N	1,5...8,0 N

eprasa.pl 1fc8f31a49

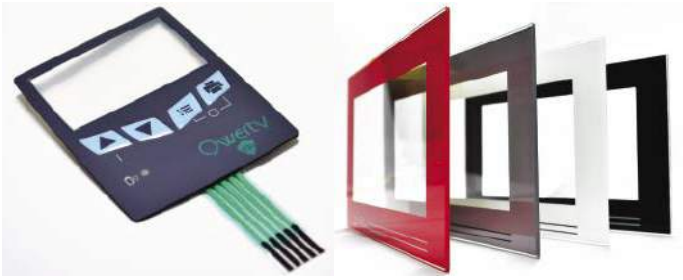
Air Gap to standardowa technika klejenia wyświetlaczy, polegająca na wykorzystaniu ramki z folii dwustronnie klejącej, umieszczonej dookoła wyświetlacza w celu zespolenia go z frontem. Jest to niedroga i uniwersalna technologia, która pozwala na łatwą wymianę wyświetlacza lub frontu szklanego bez konieczności użycia specjalistycznych narzędzi. Szczelina powietrzna może jednak wpływać na przejrzystość ekranu, szczególnie w jasnym otoczeniu, z powodu dodatkowego załamania światła. Wyświetlacze ze szczelinami powietrznymi są też bardziej podatne na uszkodzenia mechaniczne oraz te spowodowane wilgocią, dlatego gorzej nadają się do aplikacji na zewnątrz budynków lub w najtrudniejszych warunkach przemysłowych.

Bonding optyczny to zespolenie wyświetlacza z panelem dotykowym/frontem przy użyciu kleju optycznie czystego w postaci stałej (solid) lub ciekłej (liquid), na całej powierzchni wyświetlacza. Proces ten polega na naniesieniu kleju na powierzchnię ekranu, a następnie zespoleniu wyświetlacza z panelem dotykowym/frontem. Rozwiązanie to jest czasochłonne i wymaga specjalistycznego sprzętu. Cechuje się natomiast lepszymi parametrami optycznymi (mniejsze odbicia światła, poprawa kontrastu) oraz wyższą wytrzymałością na czynniki środowiskowe (kurz i kondensacja wilgoci) i narażenia mechaniczne.

Firma QWERTY przeprowadziła szereg badań i dopasowała technologię do rosnącego zapotrzebowania rynku na optycznie czyste laminowanie powierzchni sztywnych. Prowadzone prace przyczyniły się do pozyskania wiedzy na temat pełnego spektrum dostępnych technik, materiałów, maszyn oraz procedur, niezbędnych podczas bondingu optycznego. Będąc ekspertem w tej dziedzinie, jesteśmy w stanie dopasować rozwiązanie odpowiednie do każdej aplikacji. Oferujemy doradztwo w dziedzinie doboru wyświetlacza oraz najlepszej z dostępnych metod bondingu. Nasze działania charakteryzują się dużą elastycznością – możemy pracować na materiałach klienta lub własnych, kupowanych od zaufanych dostawców. W swoich produktach oferujemy wykończenie powierzchni akrylowych oraz szklanych AG (Anti-Glare), AF (Anti-Finger), AR (Anti-Reflective). Wieloletnie doświadczenie w druku cyfrowym i sitodruku oraz dostęp do szerokiego asortymentu wysokiej jakości materiałów, obrabianych za pomocą ploterów oraz maszyn CNC, czyni z firmy QWERTY eksperta w procesie integrowania wyświetlaczy i sensorów z panelami dotykowymi/frontami.

Tabela 2. Parametry klawiatur foliowych wykonanych sitodrukowo i drukiem cyfrowym

Lp.	Wyszczególnienie	Klawiatury foliowe wykonane sitodrukiem	Klawiatury foliowe wykonane drukiem cyfrowym
1	Szerokość linii graficznych	>0,3 mm	>0,2 mm
2	Dokładność druku zgodnie z ISO 2768-1 <120 mm od 120 do 400 mm >400 mm	±0,15 ±0,20 ±0,30	
3	Dopuszczalne odchylenie równomierności linii	<10%	<5%
4	Gładkość powierzchni poza przetłoczeniami	Gładkie na całej powierzchni z wyjątkiem miejsca wyprowadzenia paska	
5	Krycie kolorem	Wystarczające, jeżeli niewidoczne są warstwy poniżej	
6	Odwzorowanie koloru	DE 3	DE 6
7	Zanieczyszczenie – wielkość maksymalna	0,25 mm²	
8	Odległość pomiędzy zanieczyszczeniami – minimalna	50 mm	
9	Zanieczyszczenia w okienkach <1 dm²	1	
10	Zanieczyszczenia w okienkach 1 dm²	2/dm²	
11	Nachodzenie na siebie kolorów (zalewki)	Z powodów technologicznych zjawisko takie jest konieczne i dopuszczalne	
12	Kolory	Bez ograniczeń	
13	Forma graficzna	Na życzenie	



Klawiatury foliowe (membranowe)

Klawiatury foliowe są elementem wielu współczesnych urządzeń elektronicznych, a o ich szerokim zastosowaniu zdecydowały wszechstronność oraz niezawodność. Ze względu na swoje właściwości doskonale sprawdzają się w trudnych bądź nietypowych warunkach eksploatacyjnych, co stanowi o ich wyższości nad konkurencyjnymi klawiaturami mechanicznymi. Klawiatury membranowe są wykonywane na bazie folii poliestrowej, odpornej na szkodliwe czynniki chemiczne i klimatyczne, a – po naklejeniu na obudowę – również na większość uszkodzeń mechanicznych. Do realizacji formy graficznej naszych urządzeń stosujemy technologię druku cyfrowego, co pozwala tworzyć elementy atrakcyjne wizualnie. Integracja diod LED w wielowarstwowej strukturze klawiatury umożliwia pełni funkcjonalne sterowanie elektroniką, bez konieczności współpracy z dodatkowymi wyświetlaczami lub panelami LCD.

QWERTY
www.qwerty.com.pl

REKLAMA



eprasa.pl 1fc8f31a49

Klawiatury foliowe – przegląd technologii

Istotnym elementem dużej części współcześnie konstruowanych urządzeń elektronicznych jest interfejs użytkownika, poprzez który realizowana jest komunikacja między człowiekiem a maszyną. Elementami takiego interfejsu są najczęściej: wyświetlacz, przetwornik akustyczny i pojedyncze przełączniki lub klawiatura, przy czym nie wszystkie te elementy muszą występować jednocześnie.



Wymienione elementy interfejsu można zakwalifikować jako wejściowe i wyjściowe. Oczywiście każda z tych grup jest w rozwiązaniach praktycznych znacznie liczniejsza. Na przykład elementami wyjściowymi mogą być różnego rodzaju wskaźniki i wyświetlacze cyfrowe, diody LED, wskaźniki wychyłowe, wirtualne panele HMI realizowane na tabletach czy monitorach komputerowych itp. Do elementów wejściowych można zaś zaliczyć wszelkiego rodzaju jedno- lub wielostanowe przełączniki mechaniczne, przełączniki wieloobrotowe, enkodery, impulsatory itp. W artykule skoncentrujemy się na elementach stykowych realizowanych jako pojedyncze przyciski lub ich zespoły, połączone w matryce tworzące klawiatury membranowe (foliowe).

Klawiatury membranowe *versus* mechaniczne

Konstruktorzy urządzeń elektronicznych – szczególnie ci, którzy są odpowiedzialni za wykończenie mechaniczne wyrobów – często stają wobec dylematu, czy przyciski znajdujące się na panelu czołowym aparatury mają być wykonane jako elementy mechaniczne, czy choćby w postaci klawiatur membranowych. Są grupy urządzeń, dla których w zasadzie wyboru nie ma. Byłoby raczej trudno wyobrazić sobie na przykład audiofilski wzmacniacz akustyczny z przyciskami foliowymi. W tej grupie urządzeń obowiązują zdecydowanie rozwiązania klasyczne, czyli mechaniczne. Kanon wymaga, aby z mechaniki nie rezygnować nawet w urządzeniach sterowanych zdalnie za pośrednictwem pilota.

Wracamy jednak do przycisków membranowych. W urządzeniach pracujących w niekorzystnych warunkach środowiskowych, na przykład w dużym zawilgoceniu czy zapyleniu, elementy mechaniczne byłyby narażone na szybką degradację, z czym wiązałaby się konieczność częstego serwisowania. Podnosiłoby to koszty eksploatacji i wprowadzało dokuczliwe przerwy w pracy. Znacznie bardziej odporne na tak trudne warunki eksploatacji są klawiatury membranowe. Ich stosowanie stwarza pewne dodatkowe korzyści. Nie dość, że spełniają swoje funkcje sterujące, to jednocześnie stanowią ładny i nowoczesny, aczkolwiek dość charakterystyczny element designu urządzenia.

Na folii może być umieszczana niemal dowolna grafika z bogatą paletą barw, przyciski mogą być podświetlane, a dopełnienie stanowią świetlne elementy sygnalizacyjne. Patrząc na urządzenie z płytą czołową wykonaną w formie klawiatury foliowej, można odnieść wrażenie, że design jest płaski. Tego odczucia właściwie nie zmieniają nawet niewielkie przetłoczenia, których wykonanie jest technologicznie możliwe. Tradycjonalistom nie zawsze musi się to podobać, nawet mimo nienagannej estetyki. Na szczęście, jeśli projektant uzna, że na panelu czołowym powinny znaleźć się elementy mechaniczne, np. przełączniki hebelkowe, pokrętła potencjometrów lub impulsatorów, czy nawet wszelkiego rodzaju gniazda, możliwe jest wykonanie odpowiednich otworów w folii tak, aby części zewnętrzne tych elementów można było swobodnie wyprowadzić na zewnątrz obudowy. Istnieje ponadto możliwość wycinania okien pod wyświetlacze. W okna te może być wstawiana poliestrowa, przezroczysta szybka osłaniająca otwór i jednocześnie zabezpieczająca wyświetlacz przed czynnikami zewnętrznymi. Szybka może być ponadto wykonana w wersji matowej lub barwnej, stanowiąc tym samym filtr optyczny.

Zaletą klawiatur foliowych jest możliwość wycinania nietypowych kształtów obrysów zewnętrznych i otworów. Ponadto folia nie musi być naklejana na płaską powierzchnię, dopuszczalne są niewielkie zakrzywienia. Dzięki temu klawiatura będzie zawsze idealnie dopasowana do kształtu obudowy, na której jest naklejana – najczęściej jest to panel czołowy urządzenia.

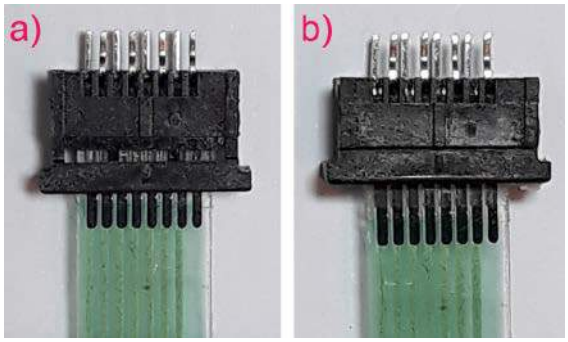
Naklejanie folii na sztywną powierzchnię nie jest jedyną możliwą metodą montażu. Spotykane są również samonośne odmiany klawiatur, w których folia jest mocowana punktowo za pomocą odpowiednich śrub.

Przegląd technologii klawiatur membranowych (foliowych)

Budowa klawiatury foliowej, jak można sądzić z samej nazwy, oparta jest na jakimś rodzaju elastycznej folii, na której naniesione są elementy stykowe i ścieżki wyprowadzające sygnały do tasiemki, która z kolei jest zakończona na przykład wtykiem, umieszczanym w gnieździe montowanym na płytce obwodu drukowanego elektroniki. Często stosowane jest rozwiązanie, w którym napylone na taśmę metalizowane ścieżki są wsuwane do odpowiednio dobranego gniazda szczelinowego ze specjalnym mechanizmem, zwiększającym docisk elementów stykowych (**fotografia 1**).

W realizacjach praktycznych budowa klawiatur foliowych nie jest tak prosta, jak mogłoby to wynikać z wcześniejszego opisu. Zanim temat zostanie rozwinięty, konieczne jest wyjaśnienie pewnych wątpliwości nomenklaturowych. Otóż na co dzień używane są dwa pojęcia: „klawiatura membranowa" i „klawiatura foliowa". Dotyczą one jednak tego samego rodzaju klawiatury, w której elementy stykowe są umieszczane na poliestrowej lub poliwęglanowej folii. Istnieje

eprasa.pl 1fc8f31a49



Fotografia 1. Taśma klawiatury i gniazdo z mechanizmem zaciskowym: a) przed zaciśnięciem, b) po zaciśnięciu



Rysunek 1. Przekrój klawiatury płaskiej

kilka technologii umożliwiających wykonanie wielu typów klawiatur, różniących się budową wewnętrzną, a co za tym idzie – spełniających rozmaite wymagania narzucane przez użytkowników. Wymagania te dotyczą własności mechaniczno-elektrycznych, ale też estetycznych, związanych z wyglądem klawiatury. W zasadzie każda, nawet najprostsza klawiatura jest zbudowana z kilku warstw pełniących odpowiednie funkcje. Styki przycisków umieszczone na wewnętrznych warstwach nie są widoczne dla użytkownika. Ich lokalizacje na powierzchni całej folii wskazują symbole graficzne widoczne z zewnątrz. Aby zwiększyć trwałość elementów graficznych, które mogłyby w stosunkowo niedługim czasie ulegać wycieraniu podczas intensywnego użytkowania, cały zadruk wierzchniej warstwy (elewacji) jest wykonywany najczęściej na wewnętrznej stronie folii.

Przyjrzyjmy się teraz budowie różnych rodzajów klawiatur membranowych.

Klawiatura płaska. To najprostszy rodzaj klawiatury, zbudowanej z 4 warstw folii. Klawiatura taka nie zawiera żadnych przetłoczeń ani elementów, np. diod świecących, umieszczanych między warstwami wewnętrznymi. Przekrój klawiatury płaskiej został przedstawiony na **rysunku 1**. Klawiatura składa się z 4 warstw. Na warstwie wierzchniej jest nanoszona grafika określająca wygląd zewnętrzny klawiatury. Pod nią znajduje się perforowana warstwa dystansowa, oddzielająca warstwę elewacyjną od poliestrowej folii, z naniesionymi na niej połączeniami i elementami stykowymi klawiatury. Na samym dnie instaluje się jeszcze warstwę klejącą, umożliwiającą trwałe zamocowanie klawiatury na powierzchni panelu czołowego urządzenia.

Klawiatura tłoczona. Budowa klawiatury tłoczonej (**rysunek 2**) jest niemal identyczna jak płaskiej. Różni się jedynie niewielkim wytłoczeniem przycisków, co sprawia, że użytkownik pewniej czuje je pod palcem. Przetłoczenia mogą być wykonywane w różnych kształtach, co nadaje klawiaturze indywidualnego charakteru. Najczęściej jest to koło, elipsa lub prostokąt/kwadrat.

Klawiatura tłoczona z blaszkami i diodami LED SMD. Użytkownicy płaskich klawiatur foliowych często narzekają, że nie wyczuwają momentu zadziałania przycisku. Faktycznie, o ile fakt aktywacji danego styku nie jest potwierdzony jakimś sygnałem dźwiękowym lub optycznym, a także nie widać tego od razu po zachowaniu się urządzenia, nie można mieć pewności, że przycisk faktycznie został naciśnięty. Często prowadzi to do powtórnego, nawet wielokrotnego naciskania



Rysunek 2. Przekrój klawiatury tłoczonej



Rysunek 3. Przekrój klawiatury z blaszkami i diodami SMD

przycisku z coraz większą siłą, co może z kolei powodować uszkodzenie klawiatury. Rozwiązaniem jest zastosowanie blaszek stykowych, które dadzą wyraźnie wyczuwalny „klik" po naciśnięciu przycisku, informujący użytkownika o prawidłowym działaniu klawiatury. Konsekwencją takiego rozwiązania jest zwiększenie liczby warstw wewnętrznych klawiatury, ale dodatkową korzyścią jest możliwość umieszczenia na jednej z wewnętrznych warstw na przykład diod LED, wykonanych w technologii SMD (**rysunek 3**). Wadą klawiatur z blaszką jest ich nieco krótsza żywotność, zaletą natomiast – mniejsza rezystancja styków w porównaniu z klawiaturami płaskimi.

Klawiatura z ekranem EMI/RFI. Pomiędzy złączem klawiatury a jej elementami stykowymi prowadzą zazwyczaj względnie długie połączenia. Konstruktorzy nie lubią takich przypadków, ponieważ w niekontrolowanych warunkach linie takie mogą nieoczekiwanie zamieniać się w anteny, zbierające lub promieniujące wysokoczęstotliwościowe sygnały zakłócające. Jeśli tę niekorzystną cechę skonfrontujemy z obowiązkowym badaniem zgodności elektromagnetycznej, uprawniającym do naklejania znaku CE, okaże się, że klawiatura może stwarzać bardzo poważne zagrożenie dla uzyskania pozytywnych wyników. Konieczne zatem jest podejmowanie odpowiednich środków zaradczych. Jednym z nich może być umieszczanie w wewnętrznej warstwie klawiatury ekranu EMI/RFI, skutecznie eliminującego wpływ zewnętrznego promieniowania elektromagnetycznego na działanie urządzenia z klawiaturą. Ekran jest wykonany najczęściej w postaci gęstej siatki przewodzącej umieszczonej na jednej z wewnętrznych warstw poliestrowych i połączonej z masą urządzenia. Zabieg taki w znacznym stopniu przyczynia się do zwiększenia odporności na zaburzenia elektromagnetyczne, a także skutecznie redukuje ich emisję. Budowa klawiatury z filtrem EMI/RFI została pokazana na **rysunku 4**. Niektóre firmy oferują ponadto ekranowanie w postaci siatki przewodzącej umieszczanej tuż nad ekranem. Wadą tego rozwiązania jest jednak zmniejszenie przepuszczalności światła, choć w praktyce na tyle małe, że niemal w ogóle nieodczuwalne.

Klawiatura z przyciskami silikonowymi. Znowu dotykamy zagadnień estetycznych. Można powiedzieć, że jest to powracający motyw w dyskusjach o klawiaturach foliowych, gdyż poza podstawowymi funkcjami, jakie pełnią, w dużym stopniu decydują również o wyglądzie urządzenia. Tym razem mowa będzie o samych przyciskach, które – jakkolwiekby było – są najważniejszym elementem klawiatury. W klawiaturach płaskich właściwie przycisków (klawiszy) w dosłownym rozumieniu nie było. Były jedynie elementy graficzne symbolizujące takie przyciski. W klawiaturach tłoczonych sytuacja wyglądała już znacznie lepiej, bo przyciski można było całkiem wyraźnie wyczuwać pod palcem. Były one również wyraźnie widoczne. Teoretycznie dałoby się korzystać z nich nawet wtedy, gdyby nie były zaznaczane jakimkolwiek nadrukiem. Oczywiście estetyka znacznie by w takim wypadku ucierpiała, więc niemal zawsze każdy przycisk jest akcentowany jakimś elementem graficznym. Tradycjonaliści mogą jednak odczuwać pewien niedosyt. Dla nich klawiatura, jak sama nazwa wskazuje, powinna składać się z prawdziwych klawiszy, a więc



Rysunek 4. Przekrój klawiatury z ekranem EMI/RFI

eprasa.pl 1fc8f31a49

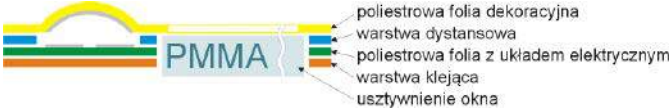


Rysunek 5. Przekrój klawiatury z przyciskami silikonowymi

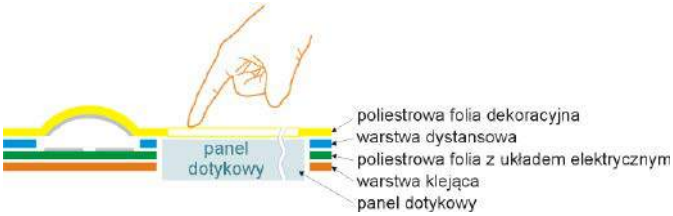
elementów trójwymiarowych, o wyraźnie wyczuwalnej wysokości. Teoretycznie możliwe wydaje się umieszczenie w otworze nad elementem stykowym jakiegoś plastikowego detalu, który miałby kształt tradycyjnego klawisza, jaki spotykamy w typowych klawiaturach mechanicznych. Problemem może być jednak stosunkowo duża twardość takiego elementu, przez co mogłoby dochodzić do uszkodzeń mechanicznych po stosunkowo krótkim czasie eksploatacji. Firmy produkujące klawiatury znalazły wyjście z tej sytuacji, zastępując klawisze plastikowe elementami silikonowymi. Materiał ten jest bardziej miękki i znacznie elastyczniejszyy od „zwykłego plastiku" i doskonale nadaje się do zastosowania w klawiaturach foliowych. Poliestrowa, górna folia dekoracyjna, stosowana w klawiaturach omawianych wcześniej, została zastąpiona matą silikonową z trójwymiarowymi przyciskami – i tylko tym różni się od innych rozwiązań. Budowa takiej klawiatury została przedstawiona na **rysunku 5**.

Klawiatury membranowe z laminowanym oknem. Wyświetlacze – czy to alfanumeryczne, czy graficzne, małe czy duże – są już stałym elementem większości urządzeń elektronicznych. Konstruktorzy i projektanci klawiatur foliowych musieli rozwiązać problem integracji elementów mechanicznych o bardzo zróżnicowanej budowie i wymiarach z elastyczną folią tworzącą klawiaturę. Najprostsze rozwiązanie, dopuszczalne na poziomie amatorskim i polegające na wycięciu okna w zewnętrznej warstwie klawiatury i umieszczeniu w nim na wcisk wyświetlacza, jest oczywiście niedopuszczalne na poziomie profesjonalnym. Po pierwsze, wyglądałoby bardzo nieestetycznie, po drugie zapewnienie choćby najmniejszej szczelności byłoby praktycznie niemożliwe. Problem został skutecznie rozwiązany i obecnie chyba nie ma firmy z tej branży, która nie miałaby w ofercie klawiatur z oknem. Rozwiązanie polega na zastosowaniu przezroczystej folii laminowanej, wzmacnianej szybką dystansową wykonaną z czystej optycznie PMMA (polimetakrylanu metylu, czyli powszechnie zwanej pleksi). Budowa takiej klawiatury została przedstawiona na **rysunku 6**. Należy zaznaczyć, że zarówno cała klawiatura, jak i okno są wycinane laserowo, co zapewnia bardzo dużą dokładność cięcia. Materiały, z których jest wykonywane okno, mają typowe grubości od 0,15 mm do 3 mm. Jeśli okno jest wykonywane nad wyświetlaczem LED, korzystne jest stosowanie kolorowych filtrów. Mogą być one nadrukowywane w oknie wyświetlacza, ale można też użyć PMMA o odpowiednio dobranej barwie. Filtry występują w różnych postaciach: jako powłoka antyrefleksyjna, powłoka twarda (hard-coated) lub matowa.

W klawiaturach z oknem można czasami obserwować niepożądany efekt optyczny, jakim są tzw. pierścienie Newtona. Zjawisko to powstaje w wyniku interferencji światła przechodzącego przez powierzchnię płaską i zakrzywioną (na nierównościach folii), i objawia się powstaniem wielobarwnych koncentrycznych pierścieni. Jedną z metod eliminacji tego zjawiska jest zastosowanie klejenia optycznego, w tym przypadku płytki PMMA i frontu foliowego. Pierścienie Newtona pojawiają się tylko wtedy, gdy grubość warstwy powietrza między stykającymi się ze sobą powierzchniami zawiera się w określonym przedziale. Drugą metodą minimalizacji, a nawet całkowitej eliminacji pierścieni Newtona jest stosowanie odpowiednio



Rysunek 6. Przekrój klawiatury z laminowanym oknem



Rysunek 7. Przekrój klawiatury z panelem dotykowym

dobranych pól dystansowych. Przepuszczalność światła w klawiaturach z oknem nie spada poniżej 90 procent. Użytkownik nie dostrzega praktycznie żadnego pogorszenia widoczności wyświetlacza.

Klawiatura membranowa z panelem dotykowym. Panele dotykowe są coraz częściej stosowane do obsługi urządzeń elektronicznych. Przyzwyczaiły nas do tego w dużym stopniu smartfony czy automatyczne kasy spotykane w sklepach. Wyświetlacz z panelem dotykowym obrazuje taką samą informację jak zwykły wyświetlacz graficzny, co więcej, oba te elementy praktycznie nie różnią się budową. Nic więc nie stoi na przeszkodzie, aby zamiast zwykłego wyświetlacza graficznego umieścić w oknie klawiatury panel dotykowy. Układ warstw takiej klawiatury pozostaje bez zmian w odniesieniu do zwykłej klawiatury z oknem laminowanym (**rysunek 7**).

Przedstawione klawiatury w zasadzie wyczerpują rozwiązania spotykane w ofertach różnych firm. We wszystkich ostatnia, spodnia warstwa jest używana do naklejenia folii na powierzchnię obudowy. Najczęściej w roli tej występuje błona klejąca firmy 3M, zabezpieczona papierem, który bezpośrednio przed naklejeniem jest zrywany. Taka metoda montażu gwarantuje doskonałe mocowanie folii do obudowy, ale osobom niemającym doświadczenia może sprawić sporo kłopotu. Problemem jest odpowiednie pozycjonowanie folii w trakcie naklejenia, które powinno być wykonane tylko jednym przyłożeniem. Klej jest na tyle silny, że nie daje szansy na skorygowanie położenia folii, jeśli tylko „złapie" obie powierzchnie. Należy ponadto uważać, aby w trakcie klejenia pod folię nie dostał się choćby najmniejszy pęcherzyk powietrza. Będzie on bardzo trudny do usunięcia, a spowoduje nieestetyczne wybrzuszenie folii. W najbardziej niekorzystnym przypadku taki błąd może być również przyczyną nieprawidłowego działania przycisków klawiatury.

I tu należy wspomnieć o dość nietypowym rodzaju **klawiatury membranowej umieszczanej na płytce obwodu drukowanego (rysunek 8)**. Na PCB montuje się blaszki przycisków, które są pozycjonowane dodatkową warstwą, umieszczaną bezpośrednio nad nimi. Kolejną warstwę tworzy poliestrowa folia trzymająca blaszki stykowe. Obwód drukowany zawiera oczywiście połączenia między stykami i gniazdem. Ponadto mogą być montowane na nim diody świecące, nad którymi są umieszczane przezroczyste okienka.

Technologie jak z sektora kosmicznego

Oglądając klawiaturę membranową jako gotowy wyrób, można odnieść wrażenie, że nie jest to element trudny w produkcji. Jednak gdy przyjrzymy się rysunkom przedstawiającym budowę różnego rodzaju klawiatur membranowych, z łatwością dostrzeżemy, że w procesie produkcyjnym wymagane jest zachowanie dość ostrych reżimów technologicznych. Klawiatury zawierają od czterech do sześciu warstw, które muszą być ze sobą bardzo dokładnie spasowane. Pewną trudność sprawia ponadto umieszczanie między warstwami elementów metalowych i elementów SMD (diod LED). Zupełnie innym zagadnieniem jest utrzymanie dużej czystości procesu, szczególnie w przypadku klawiatur z oknami. Klawiatury z zanieczyszczeniem widocznym



Rysunek 8. Przekrój klawiatury na PCB

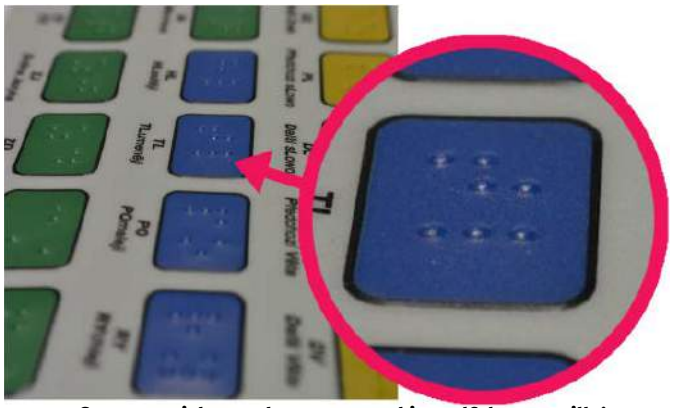
eprasa.pl 1fc8f31a49

pod szybką wyświetlacza prawdopodobnie nie kupiłby żaden klient. Dostanie się zanieczyszczeń na przykład w obszarze styków mogłoby z kolei spowodować ich błędne działanie. Co gorsza, takiego wyrobu nie da się naprawić i musi być traktowany jako brak. Z tego względu producenci sięgają po rozwiązania iście kosmiczne, wykonując niektóre procesy technologiczne (np. wspomniane wcześniej klejenie optyczne) w komorze bezpyłowej, tzw. *cleanroom*.

Wyśrubowane warunki pracy

Klawiaturom pracującym we względnie normalnych warunkach nie są narzucane specjalne, wygórowane wymagania techniczne. Istnieje jednak pewna grupa urządzeń wyposażonych w klawiatury membranowe, które są eksploatowane w często skrajnych warunkach środowiskowych i atmosferycznych. Przykładem niech będą klawiatury instalowane w kabinach prysznicowych, w saunach, na basenach, a także w aparaturze pomiarowej, obsługiwanej w halach produkcyjnych czy na otwartej przestrzeni, gdzie może występować duże zawilgocenie lub nasłonecznienie. Klawiatury takie, jak i całe urządzenia, w których są zainstalowane, muszą spełniać stopień ochrony co najmniej IP67. Oznacza to, że są całkowicie pyłoszczelne i odporne na krótkotrwałe zanurzenie w wodzie (30 sekund na głębokość 1 metra).

Są też inne zagrożenia, działające poniekąd w drugą stronę, czyli od klawiatury do środowiska. Dotyczy to urządzeń pracujących w środowisku zagrożonym wybuchem, np. w kopalni, na wieżach wiertniczych itp. Klawiatura wraz z układami elektronicznymi, do których jest dołączona, musi dawać absolutną gwarancję braku iskry w momencie zwierania lub rozwierania styków. Do takich klawiatur są stosowane specjalne folie z powierzchnią przewodzącą. Przed wyładowaniami elektrostatycznymi zabezpieczają natomiast odpowiednie warstwy i przestrzenie izolujące. Istnieje ponadto pewna specyficzna technologia produkcji klawiatur, która mogłyby się sprawdzać w omawianych zastosowaniach. Mowa o **klawiaturach pojemnościowych**, w których nie występują elementy stykowe, a więc z definicji nie może w nich dochodzić do iskrzenia. Przełączniki pojemnościowe są doskonale znane elektronikom. Wielu z nich prawdopodobnie samodzielnie implementuje już te elementy w swoich projektach. Niektóre mikrokontrolery mają specjalne porty przeznaczone do obsługi pojemnościowych sensorów dotykowych. Ten trend przechwyciły firmy produkujące klawiatury membranowe, co zaowocowało włączeniem klawiatur pojemnościowych do oferty. Warto zauważyć, że klawiatury takie wprowadzają szereg nowych możliwości, na przykład realizację sliderów o liniowych i okrągłych kształtach, czy też



Fotografia 2. Przycisk z wytłoczonym znakiem alfabetu Braille'a

tak lubianą funkcję multi-touch. Fronty klawiatur pojemnościowych są produkowane z takich materiałów jak szkło, utwardzone poliwęglany, PMMA i folie poliestrowe.

Podświetlanie

Kontynuując wątek środowiska pracy, należy wspomnieć o oświetleniu. Podświetlanie klawiatury staje się idealnym rozwiązaniem podczas eksploatacji urządzenia w warunkach małej jasności otoczenia. Wiemy już, że umieszczenie diod LED wewnątrz klawiatury nie stanowi większego problemu technologicznego. Wcześniej były już omówione klawiatury membranowe z wbudowanymi diodami LED. Pełniły one jednak jedynie funkcje sygnalizacyjne, a nie służyły do podświetlania klawiatury. Diody, których zadaniem byłoby podświetlanie całości panelu, wymagają zastosowania specjalnych elementów konstrukcyjnych. Niestety skutkuje to zwiększeniem grubości klawiatury do ok. 1,5 milimetra.

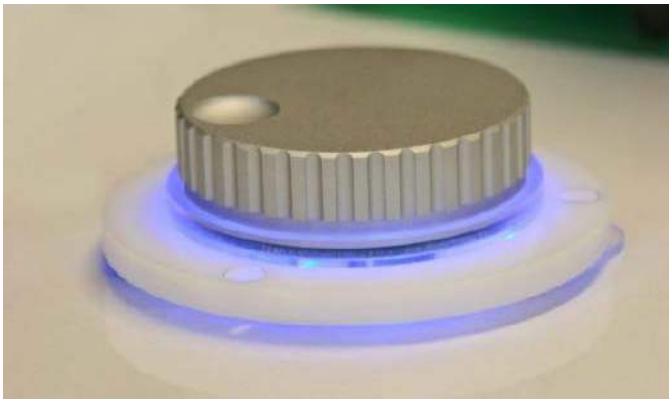
Do podświetlenia całej powierzchni klawiatury znacznie lepsze, choć niepozbawione wad, są lampy elektroluminescencyjne. Zapewniają one równomierną emisję światła na całej powierzchni. Lampa zajmuje dodatkową warstwę klawiatury. Niestety, taki rodzaj podświetlenia wymaga zastosowania dość nietypowego źródła zasilającego: 130 V, 400 Hz, co komplikuje nieco elektronikę urządzenia, ale gwarantuje doskonały efekt wizualny.

Udogodnienia dla osób słabowidzących

Jak pokazuje praktyka, użytkownikami urządzeń elektronicznych bywają osoby słabowidzące. Podświetlenie klawiatury nie zawsze jest

REKLAMA

SEMICON®
INNOVACYJNE PRODUKTY
INNOWACYJNE TECHNOLOGIE
Since 1987
Kontraktowy montaż elektroniki
Konwerting materiałów
Moduły laserowe
Szablony SMT
Dystrybucja
Wyznaczamy najwyższe standardy jakości w naszej branży
Semicon Sp. z o.o. ul. Zwoleńska 43/43A, 04-761 Warszawa
22 615 73 71
info@semicon.com.pl
semicon.com.pl

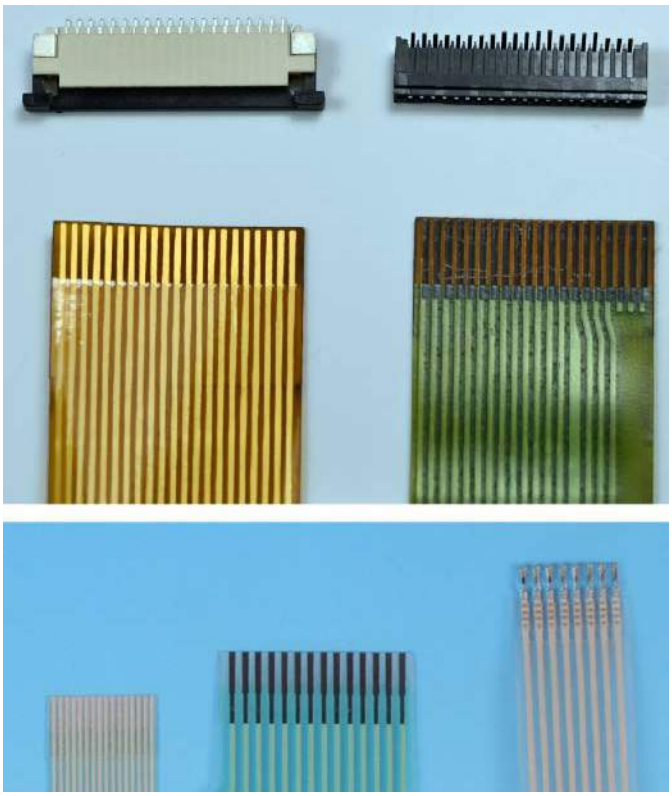


Fotografia 3. Pokrętło magnetyczne z podświetleniem

wystarczające do tego, by osoby takie mogły swobodnie używać obsługiwanych urządzeń. Firmy produkujące klawiatury wychodzą w takich przypadkach użytkownikom naprzeciw. Udogodnieniem dla osób słabowidzących jest wytłoczenie na przycisku napisu w alfabecie Braille'a (**fotografia 2**).

Nietypowe elementy regulacyjne

Wadą klawiatur foliowych/membranowych jest to, że są oparte na jednym elemencie konstrukcyjnym, jakim jest styk. Owszem, w klawiaturach pojemnościowych istniała możliwość korzystania ze sliderów, czyli liniowych lub okrągłych elementów regulacyjnych, ale zasada użytkowania samej klawiatury pozostawała praktycznie niezmienna. W każdym przypadku należy przycisnąć przycisk palcem albo przeciągnąć nim po sliderze. Tymczasem użytkownicy często mają odruchy nabyte podczas obsługi urządzeń z klasycznymi elementami regulacyjnymi: dwustanowymi przełącznikami mechanicznymi, potencjometrami czy przełącznikami obrotowymi. Pewną propozycją dla takich tradycjonalistów może być klawiatura z pokrętłem magnetycznym z czujnikiem Halla. Takie klawiatury są oferowane jednak tylko na indywidualne zamówienia. Dla wzmocnienia efektu wizualnego pokrętło może być wyposażone w kołnierz podświetlający (**fotografia 3**).



Fotografia 4. Końcówka taśmy klawiatury z pinami Crimplflex



Fotografia 5. Klawiatura z gniazdem służącym do połączenia zewnętrzną taśmą

Łączenie klawiatury z elektroniką

Klawiatura składa się z elementów stykowych i ewentualnie diod LED. Styki klawiatury mogą mieć indywidualne wyprowadzenia, ale znacznie częściej są łączone w matrycę. Konfiguracja matrycowa w znacznym stopniu minimalizuje liczbę połączeń. Klawiatura jest najczęściej łączona z elektroniką za pomocą elastycznej taśmy FFC, przystosowanej do umieszczania w złączu ZIF/NZIF. Końcówki taśmy mogą mieć ponadto zaciskane piny żeńskie typu Crimplflex (**fotografia 4**). Taśma stanowi w tym przypadku integralną część klawiatury, ale połączenie może być również wykonane zewnętrzną taśmą, łączoną z klawiaturą przez umieszczone na niej gniazdo (**fotografia 5**).

Projektując klawiaturę, należy bardzo dokładnie określić długość tasiemki. Jest ona wprawdzie elastyczna, ale powinna być umieszczana wewnątrz obudowy urządzenia tak, aby nie dochodziło do ostrych załamań. Warstwa koloidalnego srebra, tworzącego wyprowadzenie elektryczne, może łatwo zostać w takim przypadku przerwana, przez co zostanie utracone połączenie. Taka usterka jest dość trudna do naprawienia i zwykle kończy się złomowaniem całej klawiatury. Tasiemka nie może ponadto być zbyt napięta i powinna umożliwiać w miarę wygodny demontaż w razie konieczności serwisowania urządzenia.

I jeszcze raz o estetyce

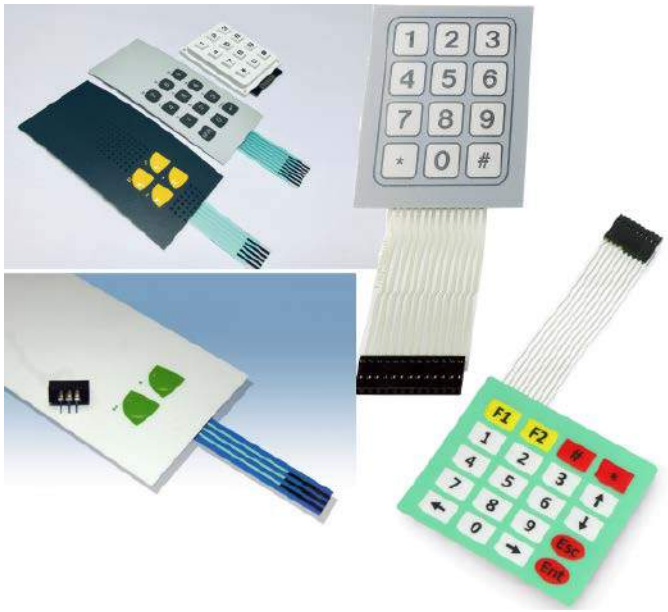
Klawiatura jest swego rodzaju wizytówką urządzenia. Zanim przekonamy się o parametrach użytkowych aparatury, odruchowo dokonujemy oceny wizualnej. Czy brzydki design może być przyczyną rezygnacji z zakupu wyrobu? To już sprawy indywidualne, ale gdybyśmy mieli dwa urządzenia o podobnych parametrach, lecz różniące się wyglądem zewnętrznym, to wydaje się, że jest bardziej prawdopodobne, że wybierzemy to ładniejsze.

Klawiaturom foliowym, których wierzchnia warstwa tworzy elewację, nadawane są różne rodzaje powierzchni. Klient może zatem określać wygląd zamawianej klawiatury lub wybierać spośród gotowych. Możliwości jest wiele, m.in. są to powierzchnie: matowa, błyszcząca, częściowo błyszcząca, z efektem szczotkowanego aluminium, aksamitna (**fotografie 6a...e**). Nadruki w bogatej kolorystyce są wykonywane metodą sitodruku lub druku cyfrowego. W metodzie **sitodruku** farba jest nanoszona na powierzchnię folii za pomocą gumowej rakli przez drobnoziarnistą tkaninę (sito). Do powierzchni dociera tylko w tych miejscach, w których występują oczka sita. Są one wytworzone w procesie przenoszenia grafiki, która ma się znaleźć na folii. W jednym przebiegu nanoszony jest tylko jeden kolor grafiki, ważne jest zatem idealne pozycjonowanie sit dla elementów wielokolorowych. Oczka sita mogą mieć

eprasa.pl 1fc8f31a49



Fotografia 6. Różne rodzaje powierzchni klawiatur foliowych: a) matowa, b) błyszcząca, c) częściowo błyszcząca, d) z efektem szczotkowanego aluminium, e) aksamitna



Fotografia 7. Przykłady gotowych klawiatur uniwersalnych

różne rozmiary, co pozwala utrzymywać dużą kontrolę nad procesem rozprowadzania farby. W efekcie uzyskuje się równomierne pokrycie drukowanych kształtów. Niestety – technologia sitodruku, używana do produkcji klawiatur membranowych, nie umożliwia zwykle uzyskiwania przejść tonalnych, które pozwalałyby np. na umieszczanie fotografii na folii, ale i tak w niczym nie umniejsza to doskonałych efektów końcowych. Możliwe jest za to stosowanie specjalnych farb, np.: odblaskowych, neonowych, metalowych, a nawet mieniących się.

W metodzie **druku cyfrowego** farba jest nanoszona na powierzchnię bezpośrednio z maszyny drukarskiej za pośrednictwem nośnika cyfrowego. W tym przypadku możliwy jest druk przejść tonalnych i zdjęć. Uzyskują one przy tym wyjątkową ostrość. Ważną zaletą omawianej metody jest indywidualizacja druku. Można dzięki niej nanosić unikatowe numery seryjne urządzeń, kody paskowe lub QR. Ponadto jest to metoda, z której chętnie będą korzystać producenci urządzeń wytwarzanych w skali niskoseryjnej, gdyż nadaje się do realizacji krótkich serii wyrobów.

Klawiatura gotowa czy na zamówienie?

Firmy zajmujące się produkcją klawiatur foliowych są otwarte na szeroką współpracę z klientami. Gros ich wyrobów to klawiatury wykonywane ma zamówienie klienta i często według jego własnych projektów. Projekt musi być oczywiście tworzony w porozumieniu z konsultantami firmy, choćby po to, by w produkcji były spełnione reżimy technologiczne. Gotowy wyrób będzie w takim przypadku spełniał wszystkie oczekiwania klienta, a więc dokładne wymiary, długość taśmy i miejsce jej wypuszczenia spod folii, a także rodzaj złącza, kolorystyka, grafika itp. Oprócz tego firmy mają zwykle



pewną kolekcję własnych klawiatur standardowych (**fotografia 7**), które są na tyle uniwersalne, że mogą być używane w wielu urządzeniach. Z takich klawiatur chętnie korzystają amatorzy lub producenci urządzeń w skali jednostkowej bądź niskoseryjnej. W tym przypadku łatwiej jest dopasować jakąś standardową obudowę do klawiatury, niż inwestować w przygotowanie produkcji klawiatury na zamówienie. Nawiasem mówiąc, istnieją firmy, które podejmują się wykonania nawet jednej sztuki. Trzeba jednak mieć na uwadze, że jednorazowe przygotowanie produkcji, obejmujące pracę specjalisty i przygotowanie dokumentacji produkcyjnej, to wydatek rzędu kilkuset złotych. Koszt taki zależy oczywiście od rodzaju klawiatury, liczby warstw, elementów dodatkowych, takich jak metalowe blaszki, warstwa ekranująca, okno pod wyświetlacze itd. Koszt ten jest ponoszony jednorazowo. Dokumentacja produkcyjna pozostaje zwykle u producenta i w przypadku wznawiania produkcji może być ponownie użyta. Kolejnym składnikiem kosztów produkcji klawiatury foliowej jest wyświetlenie sit. Nie jest to już tak wysoka kwota, ale koszt ten musi być ponoszony przy każdym wznawianiu produkcji. Pamiętajmy, że każdy kolor użyty do wykonania klawiatury jest liczony osobno. I wreszcie, ostatnim składnikiem ceny jest koszt produkcji, zawierający takie składniki, jak materiały, praca operatorów maszyn, amortyzacja wyposażenia produkcyjnego itd.

Co dalej?

Wszystko wskazuje na to, że co najmniej o najbliższą przyszłość klawiatur foliowych/membranowych nie musimy się martwić. Co więcej, systematycznie pojawiające się nowe metody i technologie wytwarzania stwarzają nadzieję, że wraz z nimi będą ewoluowały same klawiatury. O tym, że wyrób ten wzbudza duże zainteresowanie producentów elektroniki, świadczy powiększająca się liczba przedsiębiorstw w tym segmencie rynku. Aktualnie w Polsce można doliczyć się ich kilkunastu. Taki rosnący trend może cieszyć, ale też lekko dziwić, gdyż otwarcie linii produkcyjnej jest dość kosztowne. Wiąże się to z zakupem specjalistycznych maszyn, wymagających obsługi przez doświadczonych operatorów. W niektórych operacjach technologicznych stosowane jest wyrafinowane oprzyrządowanie – przykładem są choćby komory bezpyłowe, w których wykonuje się klejenie optyczne.

Należy ponadto doliczyć przedstawicielstwa firm przyjmujących zlecenia w Polsce i przekazujących produkcję do zakładów zagranicznych. Jest to możliwe ze względu na w pełni cyfrowy charakter dokumentacji produkcyjnej i doskonale funkcjonujący transport kurierski.

Jarosław Doliński, EP

eprasa.pl 1fc8f31a49

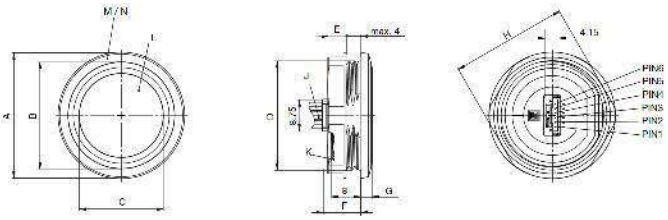
Rozwiązania dla nowoczesnych interfejsów HMI w ofercie firmy Semicon

Panel czołowy stanowi nie tylko „twarz" urządzenia, ale także wizytówkę jego producenta. Choć spore znaczenie ma sam projekt graficzny opisów i elementów dekoracyjnych, to w praktyce jeszcze ważniejsza jest jakość zastosowanych materiałów i dobór odpowiednich technologii. Do oferty firmy Semicon trafiły rozwiązania marki Schurter – jednego z czołowych producentów przełączników mechanicznych, membranowych oraz pojemnościowych, a także ekranów dotykowych za szkłem.

Tradycyjne technologie w nowoczesnej formie

Coraz większy odsetek urządzeń elektronicznych jest wyposażany w rozmaite modele ekranów dotykowych. Z różnych przyczyn klasyczne przełączniki mechaniczne wciąż pozostają jednak w użyciu, co więcej – ta grupa produktów wciąż intensywnie się rozwija. W ofercie marki Schurter dostępnych jest kilkadziesiąt serii przełączników panelowych, wyróżniających się wśród konkurencji nowoczesnym, estetycznym designem, bardzo wysoką niezawodnością oraz szerokimi możliwościami konfiguracji w zakresie wyglądu i funkcjonalności. Przykładowo – seria MSS (**fotografia 1**) obejmuje 54 warianty przełączników, które – pomimo dość dużej średnicy czoła (22,00 mm, 25,50 mm lub 34,75 mm) – oferują zaskakująco małą głębokość montażową, która w połączeniu z wygodnym sposobem podłączenia (za pomocą przewodu z wtykiem rastrowym) znacząco ułatwia implementację w przypadku urządzeń o istotnie ograniczonej przestrzeni wewnątrz obudowy (**rysunek 1**).

Produkty należące do linii MSM II (**fotografia 2**) występują w różnych wersjach materiałowych (aktuator i obudowa wykonane z aluminium bądź stali nierdzewnej), oferują szczelność na poziomie IP 66 lub IP 67 i trwałość do 1,5 miliona cykli łączeniowych. Zakres napięć podświetlenia obejmuje najczęściej spotykane wartości (5 V, 12 V oraz 24 V), zaś dostępne średnice nominalne przycisków to 16 mm, 19 mm, 22 mm oraz 30 mm. Co ciekawe, producent oferuje bardzo szeroką gamę standardowych oznaczeń (wykonywanych metodą laserową) – oprócz piktogramów (symbole zasilania, żarówki czy dzwonka) można także wybrać przyciski opisane literowo (STOP, ON, OFF, ON/OFF, START), a nietypowe nadruki pozostają dostępne na specjalne zamówienie.



Rysunek 1. Wymiary przełącznika z serii MSS

Przełączniki dotykowe i bezdotykowe – prosty montaż, potężne możliwości

Interesującą i niezwykle nowoczesną grupą produktów są przełączniki dotykowe oraz bezdotykowe (zbliżeniowe). Co ciekawe i ważne, w przypadku oferty firmy Schurter komponenty te występują w wersjach montażu nieróżniącym się od klasycznych przełączników panelowych, co pozwala na łatwe przejście na nowszą technologię, bez konieczności dokonywania jakichkolwiek większych zmian w konstrukcji obudowy urządzenia. Przyciski piezoelektryczne są dostępne w wersjach standardowych oraz iskrobezpiecznych, spełniających wymogi dyrektywy ATEX dla zastosowań w strefach zagrożonych wybuchem. Dobrym przykładem mogą być tutaj przełączniki piezo z serii PSE EX 22 (**fotografia 3**), których bardzo wysoki poziom szczelności (IP67 wg IEC 60529 oraz IP69K wg DIN 40050-9) oraz długa żywotność, dochodząca do 20 milionów cykli, pozwalają na bezpieczną i niezawodną pracę w najbardziej wymagających aplikacjach przemysłowych.

Przełączniki pojemnościowe marki Schurter występują w dwóch głównych odmianach. Seria CPS (**fotografia 4**) jest przeznaczona do montażu panelowego – ceramiczny aktuator umożliwia nie tylko niezakłóconą pracę detektora dotyku, ale także pozwala na podświetlenie całej powierzchni przycisku lub tylko pierścienia okalającego środek aktuatora. Wbudowany układ sterujący ma wyjście typu OptoMOS, zapewniające podstawową barierę galwaniczną, niezbędną w niektórych aplikacjach. Diametralnie inną konstrukcję prezentują przełączniki z serii CHS (**fotografia 5**), przeznaczone do montażu bezinwazyjnego poprzez przyklejenie do wewnętrznej strony panelu czołowego obudowy – także w tym przypadku użytkownicy mają do wyboru wersje z podświetleniem lub bez.



Fotografia 1. Przycisk metalowy z najnowszej serii MSS



Fotografia 2. Przełącznik z serii MSM II



Fotografia 3. Przełącznik iskrobezpieczny piezo z serii PSE EX 22



Fotografia 4. Przełącznik pojemnościowy z serii CPS



Fotografia 5. Pojemnościowy przełącznik z podświetleniem do montażu pod panelem czołowym

eprasa.pl 1fc8f31a49



Fotografia 6. Przełącznik bezdotykowy z serii TTS

Przełączniki bezdotykowe stanowią szczególnie ciekawy fragment oferty marki Schurter – seria TTS (**fotografia 6**) jest wprawdzie produkowana w standardowych obudowach panelowych z nakrętką mocowaną od tyłu, ale funkcję aktuatora pełni tu miniaturowy, refleksyjny sensor optyczny. Takie rozwiązanie zapewnia możliwość wygodnego sterowania np. urządzeniami medycznymi, w których zastosowanie zwykłych przełączników stwarzałoby ryzyko przenoszenia zakażeń pomiędzy kolejnymi użytkownikami.

Przełączniki membranowe

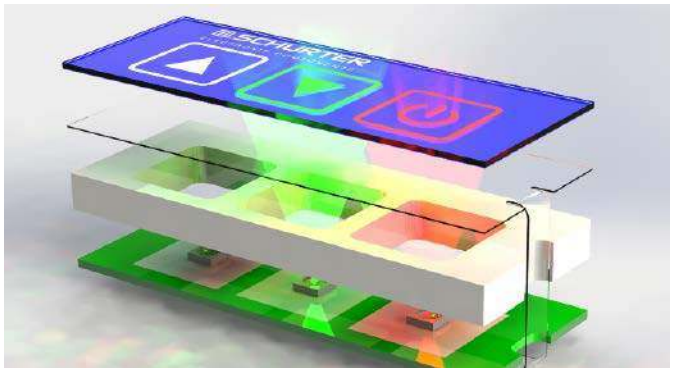
Marka Schurter posiada spore – bo ponad 40-letnie – doświadczenie w projektowaniu, produkcji oraz dystrybucji najwyższej klasy paneli klawiatur membranowych o żywotności rzędu min. 1 000 000 cykli przełączania, z opcjami podświetlenia klawiszy i tekstu oraz integracji wyświetlaczy 7-segmentowych. Dostępne są nawet specjalistyczne warianty bezdotykowe, klawiatury z osłoną metalową czy też wersje zróżnicowane pod względem siły przełączania. Przykład estetycznej i nowoczesnej klawiatury z oknem wyświetlacza graficznego i podświetlanymi przyciskami można zobaczyć na **fotografii 7**. Warto dodać, że skok klawiszy waha się w zależności od zastosowanej technologii w zakresie od 0,3 do nawet 0,8 mm, co pozwala dopasować efekty haptyczne do specyfiki danej aplikacji.

Klawiatury i ekrany dotykowe za szkłem

Najbardziej zaawansowane i zarazem dające najszersze pole do popisu projektantom wzornictwa przemysłowego są niewątpliwie planarne interfejsy użytkownika, złożone z paneli i/lub klawiatur dotykowych, ukrytych za szklanym panelem frontowym urządzenia. Technologia CapKey pozwala na łączenie z ekranami dotykowymi PCAP marki Schurter, umożliwia także dowolne podświetlanie struktur sensorów dotykowych – elektrody pojemnościowe są bowiem wykonane na bazie przeziernej folii ITO (**rysunek 2**), a dowolność w doborze kształtów i rozmiarów pól aktywnych daje możliwość wytwarzania nie tylko pojedynczych przycisków oraz całych matryc klawiszy, ale także dotykowych pokręteł i suwaków w szerokim zakresie wymiarów. Warto dodać, że przyciski mogą pracować pod niezwykle grubym szkłem (nawet do 20 mm!) lub tworzywem sztucznym (do 10 mm), zaś w przypadku pokręteł i sliderów wartości te plasują się na poziomie – odpowiednio – 4 mm/3 mm oraz 10 mm/5 mm. Użytkownicy mogą



Fotografia 7. Klawiatura membranowa marki Schurter zintegrowana z oknem wyświetlacza



Rysunek 2. Budowa prostej klawiatury podświetlanej na bazie technologii CapKey



Fotografia 8. Wysokiej klasy sterownik fotela bazujący na klawiaturze dotykowej marki Schurter

korzystać z najpopularniejszych kontrolerów dotykowych, wbudowanych w mikrokontrolery lub występujących w postaci samodzielnych układów scalonych – elementy CapKey wspierają m.in. technologie QTouch i Qmatrix, Cypress CapSense Express, Microchip CVD, CTMU i RightTouch, czy też kontrolery LC717 marki ON Semiconductor. Interfejsy HMI oparte na CapKey znajdują zastosowanie m.in. w aplikacjach medycznych, urządzeniach kosmetycznych, aparaturze laboratoryjnej, sterownikach HVAC, sprzęcie AGD i RTV oraz wielu innych (**fotografia 8**).

Warto dodać, że oferta producenta obejmuje projektowanie i produkcję ekranów o przekątnych od 3,5" do nawet 24", z zachowaniem najwyższych standardów jakości (**fotografia 9**) i kompleksowym wsparciem implementacji interfejsów HMI poprzez zastosowanie niestandardowych projektów układów sterujących, szkieł czołowych o niemal dowolnej grubości, kształcie i rozmiarze, a także zapewniają pełną integrację wyświetlacza z elektroniką oraz obudową urządzenia.

Nie sposób wspomnieć o jeszcze jednej innowacji w ofercie marki Schurter. DS11 PEM to pierwsze na świecie inteligentne gniazdo zasiania. Dzięki szczególnie kompaktowym wymiarom oferuje producentom urządzeń unikalną wartość dodaną, umożliwiając łatwe i ekonomiczne tworzenie „inteligentnych" produktów. Interfejs szeregowy, w połączeniu z elektroniką urządzenia, umożliwia dodanie funkcji wizualizacji danych, zarządzania energią i zasobami urządzenia, monitów o niezbędnych przeglądach i innych usług chmurowych.

www.semicon.com.pl



Fotografia 9. Inteligentne gniazdo zasilania DS11 PEM

eprasa.pl 1fc8f31a49

MIDIBox (2)

Niezwykły kontroler MIDI

W poprzednim wydaniu „Elektroniki Praktycznej" omawialiśmy modularną budowę projektu MIDIBox Thorstena Klose. Tym razem przyjrzymy się poszczególnym projektom MIDIBoxa. Każdy z nich składa się z dwóch podstawowych elementów: konfiguracji sprzętowej i aplikacji działającej pod kontrolą systemu MIOS.

Na stronie www.uCapps.de znaleźć można galerię pokazującą praktyczne realizacje różnych projektów, często bardzo dopracowane pod względem wyglądu. Projekty na stronie podzielone są na dwie kategorie: projekty wciąż wspierane oraz projekty stare, porzucone ze względu na ograniczenia sprzętu lub/i oprogramowania. Zanim jednak przejdziemy do projektów, warto przyjrzeć się rdzeniowi oprogramowania.

MIOS8 i MIOS32

Oprogramowanie MIDIBoxa rozwijane było już od roku 1998, ale w 2002 powstał system operacyjny IOS. Do 2008 roku MIOS działał tylko na mikrokontrolerach PIC, ale w ciągu kolejnych dwóch lat powstała też wersja 32-bitowa. Aplikacje dla MIOS8 pierwotnie pisane były w asemblerze, co mocno utrudniało zrozumienie, jak właściwie one działają, ale w zamian oferowało nieco wyższą wydajność. Pojawienie się MIOS32 i wsparcia dla mikrokontrolerów ARM pozwoliły na tworzenie aplikacji w języku C/C++ bez straty wydajności. Wręcz odwrotnie, wydajność jest znacząco wyższa, biorąc pod uwagę, iż mikrokontrolery PIC oferują maksymalnie 10 MIPS, podczas gdy STM32F103RE, czyli pierwszy wspierany przez MIOS32 mikrokontroler, oferuje 90 DMIPS. Szersza szyna danych i bardziej rozbudowane ALU dodatkowo jeszcze tę wydajność podnoszą względem ośmiobitowych PIC-ów w obliczeniach używających liczb większych niż ośmiobitowe. Nie wspominając już o przewadze w ilości pamięci RAM czy pamięci programu, oraz dzięki portowi USB, który znacząco ułatwia współpracę z komputerem – nie potrzeba (dobrego) interfejsu USB-MIDI.

Mimo wielu zalet wariantu 32-bitowego ja bym jednak nie odrzucał MIOS8 i mikrokontrolerów PIC. Na pewno wielką zaletą jest niższa cena, zwłaszcza że wariant 32-bitowy bazuje na płytkach deweloperskich, które są droższe, a i z ich dostępnością bywa różnie. MIOS8 i mikrokontrolery PIC oferują wystarczającą wydajność dla wielu użytecznych aplikacji, pod warunkiem że nie przeszkadza nam brak USB (ale i na to jest sposób, o czym będzie dalej). Wszystkie podstawowe aplikacje, jak kontrolery MIDI, sekwencery i syntezatory, powstały najpierw w wersji dla MIOS8. Co ważniejsze, nie każda aplikacja jest dostępna dla MIOS32, jak też nie każda jest dostępna dla MIOS8. Trzeba na to zwrócić uwagę, zanim zacznie się planować budowę swojego wymarzonego urządzenia MIDI.



Fotografia 1. Sekwencer MIDIBox SEQ V4 Lite



Ważnym elementem całej platformy jest aplikacja MIOS Studio. Jeśli tylko mikrokontroler ma załadowany odpowiedni bootloader, MIOS Studio pozwoli na załadowanie systemu MIOS8 (przez MIDI) lub MIOS32 (przez USB), a także wybranej aplikacji. MIOS Studio pozwala też na konfigurowanie naszego urządzenia MIDI, odczyt i zapis ustawień zapisanych w pamięci BankStick lub na karcie SD, i na wiele więcej. Oczywiście po zaprogramowaniu i skonfigurowaniu kontrolera nie trzeba już z tej aplikacji korzystać, ale może się ona przydać do zarządzania zachowanymi konfiguracjami urządzenia (zapisanymi w modułach BankStick lub na karcie SD).

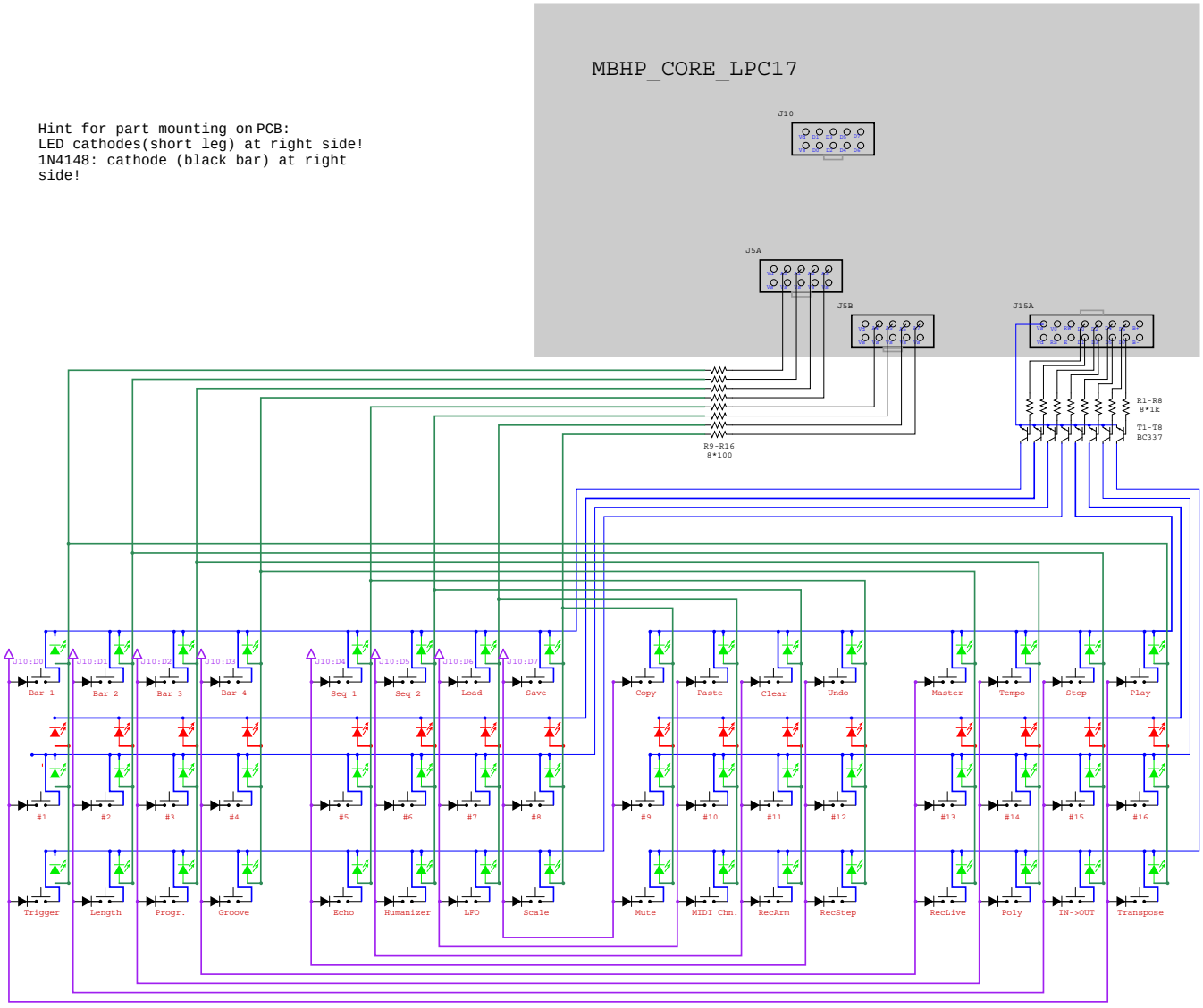
Rzeczą, o której warto pamiętać, jest to iż MIDIBox jest projektem otwartym i niekomercyjnym. Otwartość oznacza, że każdy może zgłębić tajniki kodu źródłowego i napisać własne aplikacje dla MIOS8 czy MIOS32, a także zaprojektować własne moduły czy wręcz całe kontrolery. Na stronie Thorstena Klose znajdziemy projekty od użytkowników oraz galerię zrealizowanych kontrolerów. Niekomercyjna natura licencji projektu oznacza jednak, że te kontrolery nie mogą się pojawić w szerszej dystrybucji ani w formie gotowych urządzeń, ani nawet jako kity. Inaczej pisząc, każda zainteresowana osoba musi samodzielnie wszystko wykonać. Moim zdaniem jest to wada znacząco ograniczająca popularność MIDIBoxów. Szkoda, bo projekty są bardzo użyteczne. Dzielą się na dwie kategorie: bieżące i archiwalne. Istnieje też trzecia kategoria, projekty użytkowników, ale należy je traktować bardziej jako przykłady realizacji.

Projekty bieżące

Te projekty są wciąż aktywnie wspierane. Część jest realizowana dla platform 32-bitowych, a część jest dostępna tylko dla wariantu ośmiobitowego. Zaczniemy od projektów z mikrokontrolerami 32-bitowymi.

Pierwszym z nich jest **MIDIBox SEQ V4 Lite**. Jest to bardzo prosty, choć mimo wszystko potężny sekwencer MIDI, czyli urządzenie pozwalające na zapisanie i odtwarzanie sekwencji poleceń MIDI, które kontrolują syntezatory i inne urządzenia MIDI. SEQ V4 Lite zawiera

eprasa.pl 1fc8f31a49



Rysunek 1. Schemat połączeń dla sekwencera MIDIBox SEQ V4 Lite

jedynie 48 przycisków i 64 diody LED, lecz jest to wystarczające, by zapisywać i odtwarzać różne sekwencje oraz dodawać do nich efekty oparte na poleceniach MIDI. **Fotografia 1** i **rysunek 1** prezentują wykonany sekwencer oraz jego schemat.

Jeśli jednak ktoś potrzebuje większej kontroli i więcej funkcji, to do wyboru jest **MIDIBox SEQ V4**. Ten sekwencer (**fotografia 2**) jest dużo bardziej rozbudowany, ma do czterech wejść MIDI i do 12 wyjść, podwójny wyświetlacz LCD niezbędny do obsługi wszystkich zaawansowanych funkcji, czy 17 enkoderów (po jednym na ścieżkę plus obsługa menu). SEQ V4 odtwarza cztery sekwencje naraz, każda składa się z czterech ścieżek, a każda ścieżka z 16 różnych „warstw", czyli kontrolowanych parametrów.

Rozbudowany Arpeggiator pozwala zmienić pojedynczą nutę lub akord w serię nut tworzących predefiniowany wzorzec, dzięki czemu grany utwór nabiera większej głębi.

Humanizer losowo zmienia wprowadzone wartości tak, by grane sekwencje nie brzmiały „jak z komputera", tylko były nieco bardziej „ludzkie".

Same ścieżki mają wiele dodatkowych funkcji kontrolnych, jak zapętlanie i skoki, co przypomina MOD Trackery (programy do tworzenia muzyki) z lat 80. i 90.

SEQ V4 jest tak bardzo rozbudowany, że mógłbym mu poświęcić cały osobny artykuł. Ograniczę się jednak do stwierdzenia, że z pomocą tego sekwencera można zmienić średniej klasy keyboard w odpowiednik profesjonalnego instrumentu do komponowania i grania na żywo, jak seria Yamaha Tyros.

Projekt MIDIBox NG ma nieco inne przeznaczenie. Nie służy bowiem do kontrolowania innych instrumentów (choć technicznie rzecz biorąc, może to robić), lecz do kontrolowania programów DAW (*Digital Audio Workstation* – cyfrowa stacja robocza audio) oraz pluginów VST (*Virtual Studio Technology* – format stworzony przez firmę Steinberg).



Fotografia 2. Sekwencer MIDIBox SEQ V4

eprasa.pl 1fc8f31a49



Fotografia 3. Syntezator MIDIBox SID V2 wykonany przez Wilbę



Fotografia 4. Kontroler MIDI na bazie MIDIBox64 wykonany przez Patricka

Każda osoba zainteresowana tworzeniem muzyki kojarzy takie programy, jak GarageBand, Ableton Live, FL Studio (dawniej FruityLoops, ale musieli zmienić nazwę, by nikt nie mylił programu z marką płatków kukurydzianych), Logic Pro czy Cubase. Posiadanie fizycznego kontrolera bardzo ułatwia zarówno tworzenie muzyki, jak i występowanie na żywo, i właśnie do tego przeznaczony jest MIDIBox NG. Nazwa oznacza następną generację kontrolera, gdyż projekt ten bazuje na projektach MIDIBox 64 V2 i 64E V2, o których opowiem w dalszej części artykułu.

MIDIBox NG korzysta w pełni z możliwości 32-bitowych mikrokontrolerów, by obsługiwać nawet 2048 przycisków czy diod LED. Ale to nie wszystko, bo można też podłączyć do 128 enkoderów, 128 potencjometrów i do 32 potencjometrów zmotoryzowanych (których pozycja jest kontrolowana też przez oprogramowanie, jak w profesjonalnych stołach mikserskich). Niestety, sam projekt jest jeszcze w wersji beta, więc nie wszystko może działać poprawnie. Na stronie brakuje też dokładnych informacji na temat budowy tego kontrolera, a i instrukcja obsługi nie jest jeszcze skończona.

Warto w tym miejscu wspomnieć jeszcze o **MIDIO128 V3**, projekcie opracowanym po to, by dodać funkcjonalność kontrolera MIDI instrumentom i urządzeniom niemającym takiej możliwości. Typowym przykładem są organy – wersje klasyczne (piszczałkowe), elektromechaniczne (elektroniczna klawiatura kontroluje zawory pneumatyczne w skrzyniach wiatrowych) czy całkowicie elektroniczne – używające jakiejś formy syntezy (jak organy Hammonda) czy odtwarzania dźwięku (jak Mellotron). Z pomocą MIDIO 128 V3 możemy zamienić manuały i pedały oraz przełączniki czy suwaki w takich organach na sygnały kontrolne MIDI dla organów wirtualnych (pluginów VST). Może to też działać w drugą stronę, pozwalając komputerowi kontrolować fizyczny instrument normalnie niemający tej opcji.

Jednym z dwóch aktywnych projektów z 8-bitowym mikrokontrolerem PIC jest syntezator **MIDIBox SID V2**. Na **fotografii 3** widać bardzo dopracowany pod względem wykonania egzemplarz zrobiony przez Wilbę. Syntezator ten używa dwóch układów SID firmy MOS, znanych z komputerów Commodore 64, do polifonicznej syntezy dźwięku. Są też wersje używające nawet ośmiu układów. Sam syntezator jest bardzo rozbudowany, i daje autentyczne brzmienie muzyki lat 80., ale z powodu wysokiej ceny oryginalnych układów SID jego wykonanie może być poza zasięgiem typowego hobbysty. Można użyć zamiennika w rodzaju emulacji SwinSID, ale „puryści syntezy" nie będą takiej wersji poważać (niesłusznie).

Drugim projektem wciąż bazującym na układzie 8-bitowym jest inny syntezator – **MIDIBox FM Synth**. W tym przypadku do wytwarzania dźwięku używany jest układ firmy Yamaha, YMF262, zwany też OPL3. Układ ten odpowiada za 16-bitowe brzmienie wielu gier końca lat 80. i początku lat 90. Co więcej, ze względu na mnogość urządzeń produkowanych z tym układem, jego ceny są relatywnie niskie, i nie ma aż tak dużych problemów z dostępnością. Na stronie poświęconej modułowi OPL3 wypisane są modele kart dźwiękowych, w których znajdziemy YMF262 i współpracujące z nim układy YAC512. Sam projekt oferuje 6 głosów i 5 instrumentów perkusyjnych. Każdy głos jest ze swej natury monofoniczny, ale syntezator może użyć dwóch lub więcej głosów jako jednego instrumentu celem uzyskania polifonii.

Zarówno MIDIBox SID V2, jak i MIDIBox FM Synth same z siebie „nie grają" i potrzebują źródła komunikatów MIDI, jak klawiatura sterująca, komputer z programem DAW czy sekwencer (jak wspomniany wyżej SEQ V4/V4 Lite). Za to użytkownik może kontrolować wszystkie parametry obu syntezatorów „na żywo", by osiągnąć pożądane brzmienie. Ciekawą opcją jest rozbudowanie projektu MIDIBox FM Synth o moduły AOUT pozwalające na użycie bloków, głównie filtrów sterowanych napięciem, z analogowych syntezatorów modularnych (jak Moog) do dalszej modyfikacji brzmienia.

Projekty archiwalne

Te projekty są umieszczone na dedykowanej im podstronie głównie ze względu na to, że imć Klose skupił się na 32-bitowych wersjach niektórych z nich. Moim zdaniem jednak te projekty są jak najbardziej wartościowe i mogą wzbudzić zainteresowanie osoby o ograniczonym budżecie. O problemach z dostępnością zestawów rozwojowych z 32-bitowymi mikrokontrolerami nie wspominając.

Pierwszym z tych archiwalnych projektów jest odpowiedzialny za moje zainteresowanie całą platformą **MIDIBox64E V2** i pokrewne mu **MIDIBox64 V2** oraz **MIDIBox16E**. MIDIBox64 V2 to dość rozbudowany kontroler MIDI udostępniający do 64 potencjometrów, przycisków i diod LED. Latencja na potencjometrach jest poniżej 1 ms, co jest świetnym wynikiem, biorąc pod uwagę użycie ośmiobitowego mikrokontrolera. **Fotografia 4** pokazuje przykładową realizację, **fotografia 5** kontroler programu DJskiego Traktor zbudowany przez Pay.C na bazie MIDIBox64E. MIDIBox16E i 64E V2 to dwa pokrewne projekty, które zamieniają potencjometry na enkodery. MIDIBox 16E miał ich szesnaście, stąd nazwa, plus dodatkowe 8 przycisków. To wystarcza, by regulować do 128 parametrów na jeden bank ustawień poprzez wybranie jednej z ośmiu „stron" parametrów. Wersja „wypasiona" może mieć 64 enkodery – przyciski obsługi menu są podłączone do portu J6 na płytce mikrokontrolera PIC. Dla mnie wariant, mający 32 enkodery z przyciskami plus

eprasa.pl 1fc8f31a49



Fotografia 5. Kontroler programu Traktor na bazie MIDIBox64E wykonany przez Pay.C

kolejne 32 przyciski dodatkowe, wydaje się najbardziej optymalnym rozwiązaniem pod względem kosztów i możliwości.

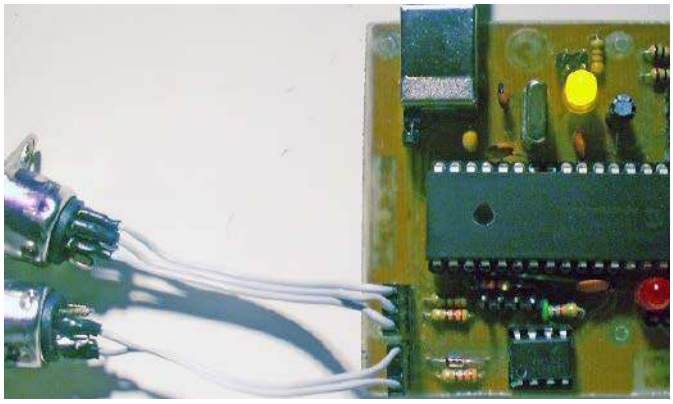
Dodatkowo wokół każdego enkodera można umieścić pierścień z maksymalnie 16 diod LED do wskazywania bieżącej nastawy danego parametru. Oczywiście przełączenie się na kolejną „stronę" lub bank aktualizuje te wartości.

Swego czasu zaprojektowałem własną płytkę pod budowę MIDIBoxa 64E V2, na której zintegrowane zostały wszystkie moduły DIN i DOUT oraz sam mikrokontroler, by zredukować liczbę połączeń. Ze wszystkich projektów rodziny MIDIBox najbardziej polecam na początek właśnie 64E V2. Warto też wspomnieć o projekcie **MIDIBox LC**, który pozbawiony jest jakiegokolwiek opisu poza diagramami połączeń, ale który pozwala na budowę dość rozbudowanych kontrolerów.

Wśród archiwalnych projektów znajdziemy też wcześniejsze wersje sekwencerów, z czego najbardziej rozbudowanym wariantem jest oczywiście **MIDIBox SEQ V3**. Ze względu na ograniczenia sprzętu ten sekwencer ma 16 ścieżek w czterech grupach (patternach), z czego każda ma tylko trzy „warstwy" parametrów. Są też inne ograniczenia, jak długość sekwencji w każdej ścieżce licząca maksymalnie 32 kroki. Mimo tych i innych ograniczeń ten projekt jest wart uwagi, gdyż jest wystarczający do współpracy z syntezatorami MIDIBox SID V2 i MIDIBox FM Synth.

Wśród tych projektów jest też **MIDIbox 808**, czyli SEQ V3 zoptymalizowany do pracy w roli sekwencera perkusyjnego, opracowany na potrzeby projektu **MB808**, ale może współpracować również z komputerami, syntezatorami i modułami dźwiękowymi MIDI mającymi stosowne brzmienia. Układ YMF262/OPL3 ma 5 brzmień perkusyjnych uzyskiwanych drogą uproszczonej syntezy FM, ale nie są one rewelacyjne. Układ SID też potrafi wytwarzać brzmienia perkusyjne. Ale lepszą drogą może być użycie analogowych modułów tworzących dźwięki perkusyjne w połączeniu z kolejnym, archiwalnym projektem, **MIDIBox CV V1**, który zamienia komunikaty MIDI na napięcia kontrolne dla syntezatorów analogowych w stylu Mooga. Niestety, syntezatory analogowe w tym stylu są wyjątkowo drogie, zwłaszcza na polskim rynku, więc taki interfejs nie każdemu się przyda. Oczywiście można samodzielnie zbudować taki syntezator – nie brakuje schematów i gotowych projektów DIY, a i ceny w częściach zwykle nie są wygórowane, ale sumaryczny koszt rozsądnego zestawu modułów może być znaczny.

Na koniec chciałbym wspomnieć o małym projekcie, którego nie znajdziecie w sekcji projektów, lecz w sekcji nieaktualnych modułów. Chodzi dokładniej o moduł/projekt oznaczony jako **MBHP_USB_PIC**. Jest to specjalna wersja firmware dla 8-bitowych PIC-ów używanych przez MIDIBoxa, zmieniająca je w całkiem dobry interfejs USB-MIDI.



Fotografia 6. Interfejs USB-MIDI MBHP_USB_PIC w wykonaniu autora artykułu

Ten projekt zrealizowałem pod koniec 2012 r. po tym, jak tani kabel interfejsowy rozczarował mnie opóźnieniami na poziomie nawet pół sekundy, zaś ceny dobrych interfejsów USB-MIDI były zbyt wysokie na mój skromny budżet. Projekt ten dostarcza jedno wejście i jedno wyjście MIDI, ale można go rozbudować, dokładając do czterech modułów IIC MIDI, by mieć łącznie pięć par portów. Na własne potrzeby zaprojektowałem dedykowaną płytkę (**fotografia 6**). Układ po zlutowaniu działał od razu i zapewniał sprawną komunikację bez żadnych odczuwalnych opóźnień. Co ciekawe, projekt jest zoptymalizowany pod kątem przesyłania konfiguracji i programów dla MIOS8, z czym tańsze interfejsy MIDI sobie zupełnie nie radzą.

Podsumowanie

MIDIBox to przykład, jak używając relatywnie tanich komponentów i prostych modułów, można stworzyć budżetowy odpowiednik kosztownych i dość niszowych produktów. Gdy sam po raz pierwszy zainteresowałem się budową lub zakupem kontrolera MIDI, urządzenia takie, nawet budżetowe, kosztowały po kilkaset złotych prawie 20 lat temu. Obecnie ceny prostych urządzeń są podobne, ale modele bardziej rozbudowane są wciąż sporo droższe.

Nadal budowa własnego urządzenia, pod własne potrzeby, jest opłacalna, zwłaszcza gdy dostępnych jest wiele darmowych lub darmowych z ograniczeniami programów DAW (Cakewalk, Waveform Free, LMMS, Reaper – wieczny trial, czy Studio One Prime). MIDIBoxy w wersji ośmiobitowej są szczególnie wdzięcznymi projektami na początek, gdyż koszt jest niski, a możliwości ogromne. Plus, mając już programator układów PIC, dość łatwo można rozpocząć przygodę z tą przeogromną i bogatą rodziną układów ośmio-, szesnasto- i trzydziestodwubitowych. Przed erą Arduino to właśnie PIC były najczęściej używanymi układami w amatorskich projektach w USA, bo były tanie i dostępne.

Na przykładzie MIDIBoxa widać też, jak przemyślany projekt o konstrukcji modularnej może przekształcić się w mnogość gotowych urządzeń i wariantów realizacji skrojonych pod potrzeby konkretnych osób. Realizację własnego projektu na bazie MIDIBoxa w obecnych czasach ułatwi też dostęp do produkcji małych serii (5–10 szt.) profesjonalnie wykonanych płytek drukowanych z Chin czy rozpowszechnienie się tanich, aczkolwiek dobrych drukarek 3D, przecinarek i ploterów laserowych czy frezarek CNC, oraz usług z nimi związanych. Kompletną obudowę wedle własnego projektu można wyciąć ze sklejki, a front wyciąć w akrylu lub poliwęglanie, zaś wszystkie inne części wydrukować, używając tworzyw w rodzaju PLA czy PET-G.

MIDIBox może być dobrym startem, a zarazem motywacją do realizacji bardziej rozbudowanego urządzenia DIY. Dlatego właśnie lubię MIDIBoxa i polecam go każdemu elektronikowi-hobbyście zainteresowanemu też tworzeniem muzyki.

Paweł Kowalczyk
urgon@wp.pl

eprasa.pl 1fc8f31a49

Zamek do drzwi oparty na Arduino, który wykrywa rytm pukania

Zaprezentowany projekt to bardzo ciekawe urządzenie, które można zastosować jako nietypowe zabezpieczenie drzwi przed intruzami czy nawet zamknięcie sekretnej kryjówki. Zamek otworzy się tylko wtedy, gdy usłyszy sekretny stukot. To prawie jak otwieranie drzwi specjalnym hasłem.

Cały projekt miał żartobliwy początek, jak wskazuje autor, ale okazało się, że system jest zdumiewająco precyzyjny w rozpoznawaniu rytmu pukania do drzwi. Jeśli czułość algorytmu jest ustawiona na maksimum, może nawet rozpoznawać różne osoby, nawet jeśli dają ten sam stukot! Powoduje to wiele fałszywych niepowodzeń, co nie jest dobre, szczególnie gdy spieszymy się i chcemy jak najszybciej wejść do pomieszczenia.

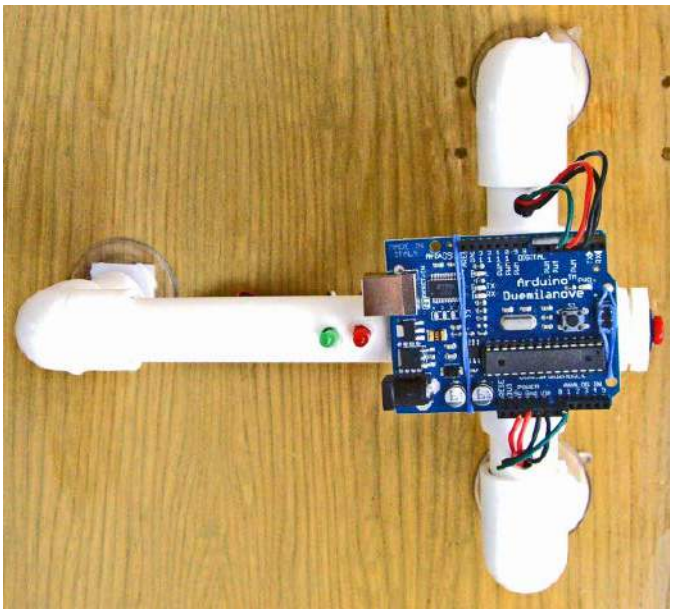
Układ jest programowalny. Wystarczy nacisnąć przycisk programowania i wystukać nowy rytm, a wtedy układ otworzy drzwi tylko przy nowym haśle. Domyślny stukot to znana melodyjka „Shave and a Haircut", ale można go bez problemu przeprogramować na cokolwiek innego, o długości do 20 stuków. Można użyć swojej ulubionej piosenki, ciągu znaków zaszyfrowanych morsem, słowem – cokolwiek.

Uwaga: Ten projekt ma wyłącznie charakter rozrywkowy. Poprzez zastosowanie tego systemu faktycznie zmniejszamy bezpieczeństwo swoich drzwi, wprowadzając dodatkowy sposób ich otwierania i ujawniając kod odblokowujący każdemu w zasięgu słuchu. Jeśli decydujemy się zainstalować to na swoich drzwiach, zdecydowanie zaleca się mieć przy sobie klucz. Baterie mogą się wyczerpać, przyssawki mogą odczepić, albo po prostu można zapomnieć swojego kodu. Ten system to tylko zabawka i nie polegajmy na nim, jak na niezawodnym zabezpieczeniu!

Potrzebne elementy

Realizacja tego projektu zajmie kilka godzin i wymaga co najmniej podstawowych umiejętności mechanicznych, znajomości elektroniki i programowania na platformie Arduino. Oprócz tych umiejętności potrzebne będą podstawowe narzędzia, takie jak wiertarka, piła, która potrzebna będzie do cięcia rur PCV, lutownica i podstawowe narzędzia. Wśród komponenty potrzebne będą:

- płytka Arduino Duemilanove (lub inna zgodna płytka lub po prostu mikrokontroler z co najmniej 1 wejściem analogowym i 3 wyjściami cyfrowymi);
- silnik redukcyjny 5 V – im większy będzie jego moment obrotowy, tym lepiej. Autor rekomenduje silnik o momencie co najmniej 11 Ncm przy zasilaniu 5 V;
- głośnik piezoelektryczny o średnicy 30 mm. Można użyć większego lub mniejszego, pamiętając, że mniejsze będą mniej wrażliwe;
- przycisk zwykły chwilowy (normalnie wyłączony);
- czerwona i zielona dioda LED;
- tranzystor NPN P2N2222A;
- dioda prostownicza (1N4001 lub podobna);



- rezystory: 2,2 kΩ, 10 kΩ, 1 MΩ oraz 2×560 Ω (lub inne, potrzebne do zapalenia diody LED przy 5 V);
- mały kawałek płytki uniwersalnej do montażu (wystarczy 5×15 otworów);
- złącze na baterię 9 V i bateria 9 V (lub dowolny inny sposób dostarczenia 7...12 V do Arduino. Można zastąpić ją zasilaczem wtyczkowym).

Układ nie ma obudowy. Zamiast tego elementy są zainstalowane na stelażu z rurek PCV. Do konstrukcji stelaża potrzebne jest około 50 cm rurki o średnicy 1/2" i trzy łączniki do rurek do połączenia pod kątem prostym, pojedynczy łącznik do połączenia pięciu rurek i dwie zaślepki do rurek. Do zamocowania układu na drzwiach autor przewidział przyssawki – trzy sztuki o średnicy 1/2" każda, ale nie każde drzwi się do tego nadają. Jeśli na naszych drzwiach nie można zamontować przyssawek, należy rozważyć inny sposób montażu.

Dodatkowo potrzebny będzie płaskownik o szerokości 12 mm i grubości około 0,25...0,5 mm i blaszka o szerokości 25 mm i grubości 1 mm. Może ona być wykonana ze stali, miedzi itp. Do montażu całości potrzebne są śruby, dwie o średnicy 1,5 mm lub 3 mm i długości 10 mm i 2 śruby M1,6 o długości 16 mm. Najlepiej, aby wszystkie były wyposażone w podkładki. Te ostatnie potrzebne są do mocowania silnika – jeśli nasz ma inne otwory, może wymagać M1,6 lub M2.

Oprogramowanie

Najlepiej jest wgrać program do układu przed przystąpieniem do montażu elementów elektronicznych, tak aby można je było testować na bieżąco. Postępujmy zgodnie z poniższymi krokami:

1. Pobierz plik *secret_knock_detector.pde* na dole tej sekcji i skopiuj go do swojego folderu z programami Arduino (lub wyświetl tekst i skopiuj go do nowego programu).

eprasa.pl 1fc8f31a49

2. Otwórz program i skompiluj go. Powinien skompilować się poprawnie za pierwszym razem, ale zawsze warto się upewnić.
3. Podłącz Arduino i wgraj program.

I na tym operacja mogłaby się zakończyć – w oprogramowaniu nie ma potrzeby wprowadzania żadnych zmian, aby dostosować je do naszej aplikacji.

Przyjrzyjmy się jednak strukturze programu i ciekawszym fragmentom kodu, które mogą być przydatne, jeżeli chcielibyśmy lepiej zrozumieć jego działanie czy też wprowadzić jakieś modyfikacje. Numery linii są orientacyjne, gdyż to, jak wyświetlać będą się u nas, zależeć może od ustawień edytora itp. Kompletny program pokazano na **listingu 1**.

Linia *const int threshold = 4;* Jest to czułość detektora stukania. Jeśli otrzymujemy dużo zakłóceń, należy podnieść tę wartość (maksymalnie do 1023), jeśli z kolei trudno jest wykryć stukanie, można ten parametr obniżyć (nawet do 1).

Linia *const int rejectValue = 25;* oraz *const int averageRejectValue = 15;* służą do określenia, jak dokładnie ktoś musi stukać. Są to procenty i powinny mieścić się w zakresie 0...100. Ich obniżenie oznacza, że trzeba mieć bardziej precyzyjny rytm, a wyższa wartość zapewnia więcej wyrozumiałości. Wartość *averageRejectValue* powinna zawsze być niższa niż *rejectValue*. Ustawienia równe około 10 i 7 sprawiają, że dwie osoby mają trudność z wykonaniem tego samego stukania, nawet jeśli znają rytm. Jednakże zwiększa to liczbę fałszywie negatywnych detekcji (tj. gdy stukamy rytm poprawnie, a drzwi i tak się nie otwierają).

Linia *const int knockFadeTime = 150;* to prymitywny licznik czasu dla czujnika stukania. Po usłyszeniu stukania przestaje on słuchać przez tyle milisekund, aby nie liczyć tego samego stukania więcej niż raz. Jeśli pojedyncze stukanie jest liczone jako dwa, należy zwiększyć ten czas. Jeśli nie rejestruje dwóch szybkich stuknięć, czas należy zmniejszyć.

Linia *const int lockTurnTime = 650;* to liczba milisekund, przez które uruchamiamy silnik, aby otworzyć drzwi. Jak długo to powinno trwać, zależy od projektu silnika i zamka. Nie ma nic złego w tym, jeśli trochę za długo działa, ponieważ w projekt został wprowadzony prosty sprzęg, ale lepiej, aby wszystkie elementy działały z umiarem.

Linia *const int maximumKnocks = 20;* to maksymalna liczba stuknięć jakie rejestrujemy. 20 to sporo. Można zwiększyć ten parametr, jeśli mamy dobrą pamięć. Jeśli za dużo zwiększymy długość sekwencji, może skończyć się pamięć w układzie.

Linia *const int knockComplete = 1200;* to maksymalna liczba milisekund przez które układ będzie czekał na stuknięcie. Jeśli układ nie usłyszy stuknięcia przez ten czas, założy, że użytkownik skończył i sprawdzi, czy sekwencja jest poprawna.

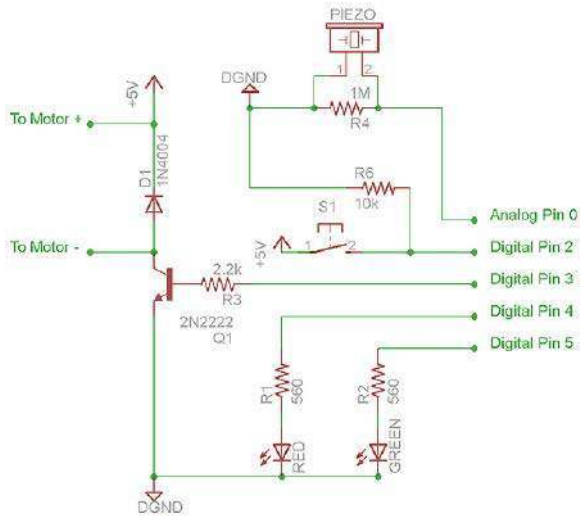
Linia *int secretCode[maximumKnocks] = {50, 25, 25, 50, 100, 5.....* to domyślny rytm stukania, który układ rozpoznaje po włączeniu. Jest to dziwna notacja rytmiczna, ponieważ każda wartość jest procentem najdłuższego czasu stukania.

Jeśli chodzi o debugowanie kodu, w programie zawarto kilka istotnych narzędzi:

Linia *Serial.begin(9600)* oraz *Serial.println("Program start.");* – jeśli odkomentujemy te linie, będziemy mogli zobaczyć kilka informacji debugujących na porcie szeregowym. Jest kilka innych linii kodu do debugowania rozmieszczonych w reszcie kodu, które można odkomentować, aby zobaczyć, co dzieje się wewnątrz algorytmu. Należy upewnić się, że port jest ustawiony na odpowiednią prędkość.

Budowa układu

Teraz możemy przystąpić do budowy układu, najpierw np. na płytce stykowej, aby upewnić się, że wszystko działa poprawnie. Na **rysunku 1** pokazany jest schemat podłączenia zewnętrznych elementów. Montaż realizujemy krok po kroku, zgodnie z opisaną poniżej kolejnością, w ten sposób najprościej będzie sprawdzać po kolei poprawność instalacji poszczególnych komponentów.



Rysunek 1. Schemat ideowy układowy

Podłączenie czujnika piezoelektrycznego

Do głośnika piezoelektrycznego należy przylutować dwa przewody o długości 30 cm. Podłączamy je pomiędzy wejściem analogowym 0 a masą. Należy również dołączyć rezystor 1 MΩ między pinem analogowym 0 a masą. Aby przetestować ten sensor z podłączonym Arduino do komputera (za pośrednictwem USB lub kabla szeregowego), należy otworzyć monitor szeregowy w Arduino IDE. Po włączeniu Arduino powinniśmy zobaczyć tekst „Program start". Należy stuknąć w głośnik piezoelektryczny, co powinno wywołać przesłanie kolejnego tekstu na terminal – „knock starting" i „knock", za każdym razem, gdy uderzymy w sensor. Jeśli poczekamy teraz chwilę, to zobaczymy „Secret knock failed" lub „Door unlocked!", jeśli rytm był poprawny.

Jeśli nie widzimy na ekranie wiadomości albo widzimy śmieci, to należy upewnić się, że port szeregowy szeregowy jest ustawiony na 9600 bodów i zresetować zasilanie Arduino. Jeśli system przesyła komunikaty o stukaniu bez stukania, detektor może być zbyt czuły. W takim przypadku trzeba edytować program. W okolicach linii 27 należy zwiększyć wartość progową. Można ją podnieść nawet do 1032, jeśli mamy do czynienia z bardzo czułym sensorem. Parametr *const int threshold = 3;* określa minimalny sygnał z głośnika piezoelektrycznego, potrzebny, aby zarejestrować stuknięcie. Gdy detekcja już działa, można zakomentować lub usunąć linie zaczynające się od *Serial...* Nie powinny być już dalej potrzebne.

Podłączenie diody LED

W dalszej kolejności podłączamy diody LED, dzięki czemu będziemy mogli śledzić działanie układu bez podłączania kabla szeregowego do komputera. Podłączamy czerwoną diodę LED do wyjścia cyfrowego 4, a zieloną diodę LED do 5, z odpowiednimi rezystorami – np. o wartości 560 Ω (lub innymi, zależnie od naszych diod LED). Pamiętajmy o biegunowości diod i odpowiednim ich podłączeniu pomiędzy wyjścia Arduino i masę.

Jeśli zasilimy obwód, zielona dioda LED powinna się zapalić. Jeśli nie, należy sprawdzić połączenia i upewnić się, że dioda LED jest poprawnie podłączona. Za każdym razem, gdy stukniemy, zielona dioda powinna przygasać. Po zastukaniu właściwej sekwencji zielona dioda powinna mignąć kilka razy. Stuknięcie złej sekwencji powinno zakończyć się miganiem czerwonej diody LED.

Podłączenie przycisku programowania

Do przycisku należy przylutować przewody o długości 20 cm. Jedną stronę przycisku podłączamy do +5 V. Drugi pin podłączony zostanie do wejścia cyfrowego 2 w Arduino i przez opornik 10 kΩ do masy. Po podłączeniu zasilania i naciśnięciu przycisku powinna zapalić się czerwona lampka. Przytrzymanie przycisku i wystukanie sekwencji powinno zapalić obie diody. Podczas stukania i programowania

eprasa.pl 1fc8f31a49

Listing 1.

```
// Definicje pinów
// Piezo sensor
const int knockSensor = 0;
// Przycisk programowania
const int programSwitch = 2;
// Silnik otwierania zamka
const int lockMotor = 3;
const int redLED = 4; // LED
const int greenLED = 5; // LED

// Stałe sterowania detektorem
const int threshold = 3;
const int rejectValue = 25;
const int averageRejectValue = 15;
const int knockFadeTime = 150;
const int lockTurnTime = 650;
const int maximumKnocks = 20;
const int knockComplete = 1200;

// Zmienne
int secretCode[maximumKnocks] = {
50, 25, 25, 50, 100, 50, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
// Gdy ktoś puka ta macierz wypełnia się mierzonymi opóźnieniami
int knockReadings[maximumKnocks];
// Ostatni pomiar sensora uderzenia
int knockSensorValue = 0;
// Flaga do zapamiętywania naciśnięcia przycisku programowania
int programButtonPressed = false;

void setup() {
  pinMode(lockMotor, OUTPUT);
  pinMode(redLED, OUTPUT);
  pinMode(greenLED, OUTPUT);
  pinMode(programSwitch, INPUT);
  Serial.begin(9600);
  Serial.println("Program start.");
  // Zapalenie zielonej diody LED
  digitalWrite(greenLED, HIGH);
}

void loop() {
  // Nasłuchiwanie pikania
  knockSensorValue = analogRead(knockSensor);
  // Naciśnięcie przycisku programowania
  if (digitalRead(programSwitch)==HIGH){
    // Jeśli tak, to zapisujemy stan programu
    programButtonPressed = true;
    // i zapalamy czerwoną diodę, aby to zasygnalizować
    digitalWrite(redLED, HIGH);
  } else {
    programButtonPressed = false;
    digitalWrite(redLED, LOW);
  }
  // Wykryto pukanie
  if (knockSensorValue >=threshold){
    // Dekodowanie pukania
    listenToSecretKnock();
  }
}

// Zapisuje czas uderzeń
void listenToSecretKnock(){
  Serial.println("knock starting");
  int i = 0;
  // resetuje wartości w macierzy
  for (i=0;i<maximumKnocks;i++){
    knockReadings[i]=0;
  }
  // Licznik
  int currentKnockNumber=0;
  // Start pomiaru czasu
  int startTime=millis();
  int now=millis();
  digitalWrite(greenLED, LOW);
  if (programButtonPressed==true){
     digitalWrite(redLED, LOW);
  }
  // Czekamy po uderzeniu pewien czas
  delay(knockFadeTime);
  digitalWrite(greenLED, HIGH);
  if (programButtonPressed==true){
     digitalWrite(redLED, HIGH);
  }
  do {
    // Czekamy na kolejne uderzenie lub timeout
    knockSensorValue = analogRead(knockSensor);
    // Jeśli wystąpiło kolejne uderzenie
    if (knockSensorValue >=threshold){
      Serial.println("knock.");
      // zapisuujemy czas,
      now=millis();
      knockReadings[currentKnockNumber] = now-startTime;
      // zwiększamy licznik
      currentKnockNumber ++;
      // i resetujemy pomiar czasu
      startTime=now;
      digitalWrite(greenLED, LOW);
      if (programButtonPressed==true){
        digitalWrite(redLED, LOW);
      }
      // Czekamy po uderzeniu pewien czas
      delay(knockFadeTime);
      digitalWrite(greenLED, HIGH);
      if (programButtonPressed==true){
        digitalWrite(redLED, HIGH);
      }
    }
    now=millis();
    // Czy nastąpił timeout lub ditarliśmy do limitu uderzeń?
  } while ((now-startTime < knockComplete)
    && (currentKnockNumber < maximumKnocks));
  // sprawdzanie poprawności sekwencji uderzeń
  if (programButtonPressed==false){
    if (validateKnock() == true){
      triggerDoorUnlock();
    } else {
      Serial.println("Secret knock failed.");
      digitalWrite(greenLED, LOW);
      for (i=0;i<4;i++){
        digitalWrite(redLED, HIGH);
        delay(100);
        digitalWrite(redLED, LOW);
        delay(100);
      }
      digitalWrite(greenLED, HIGH);
    }
  } else {
    validateKnock();
    Serial.println("New lock stored.");
    digitalWrite(redLED, LOW);
    digitalWrite(greenLED, HIGH);
    for (i=0;i<3;i++){
      delay(100);
      digitalWrite(redLED, HIGH);
      digitalWrite(greenLED, LOW);
      delay(100);
      digitalWrite(redLED, LOW);
      digitalWrite(greenLED, HIGH);
    }
  }
}

// Otwiera drzwi silnikiem
void triggerDoorUnlock(){
  Serial.println("Door unlocked!");
  int i=0;
  // Obracanie silnikiem
  digitalWrite(lockMotor, HIGH);
  digitalWrite(greenLED, HIGH);
  // ...przez jakiś czas...
  delay (lockTurnTime);
  // po czym wyłączamy silnik
  digitalWrite(lockMotor, LOW);
  for (i=0; i < 5; i++){
      digitalWrite(greenLED, LOW);
      delay(100);
      digitalWrite(greenLED, HIGH);
      delay(100);
  }
}

// Sprawdza, czy nasza sekwencja zgadza się z zapisaną
boolean validateKnock(){
  int i=0;
  int currentKnockCount = 0;
  int secretKnockCount = 0;
  int maxKnockInterval = 0;
  for (i=0;i<maximumKnocks;i++){
    if (knockReadings[i] > 0){
      currentKnockCount++;
    }
    if (secretCode[i] > 0){
      secretKnockCount++;
    }
    if (knockReadings[i] > maxKnockInterval){
      maxKnockInterval = knockReadings[i];
    }
  }
  if (programButtonPressed==true){
      for (i=0;i<maximumKnocks;i++){
        secretCode[i]= map(
            knockReadings[i],0, maxKnockInterval, 0, 100);
      }
      digitalWrite(greenLED, LOW);
      digitalWrite(redLED, LOW);
      delay(1000);
      digitalWrite(greenLED, HIGH);
      digitalWrite(redLED, HIGH);
      delay(50);
      for (i= 0; i< maximumKnocks ; i++){
        digitalWrite(greenLED, LOW);
        digitalWrite(redLED, LOW);
        if (secretCode[i] > 0){
          delay(map(secretCode[i],0, 100, 0, maxKnockInterval));
          digitalWrite(greenLED, HIGH);
          digitalWrite(redLED, HIGH);
        }
        delay(50);
      }
      return false;
  }
  // Czy liczba uderzeń się zgadza?
  if (currentKnockCount != secretKnockCount){
    return false;
  }
  int totaltimeDifferences=0;
  int timeDiff=0;
  // Normalizacja czasu
  for (i=0;i<maximumKnocks;i++){
    knockReadings[i]= map(
        knockReadings[i],0, maxKnockInterval, 0, 100);
    // Różnice czasu uderzeń
    timeDiff = abs(knockReadings[i]-secretCode[i]);
    // Pojedynczy czas zbyt odbiega od zapisanego
    if (timeDiff > rejectValue){
      return false;
    }
    totaltimeDifferences += timeDiff;
  }
  if (totaltimeDifferences/secretKnockCount>averageRejectValue){
    // Średnia rozbieżność z zapisanym czasem
    return false;
  }
  return true;
}
```


eprasa.pl 1fc8f31a49

obie diody powinny migać. Po zakończeniu odtwarzania nowa sekwencja stuknięć jest zapisywana, a światła będą migać przez chwilę na przemian na czerwono i zielono, aby o tym poinformować.

Podłączenie silnika

Do silnika lutujemy przewody o długości 20 cm i podłączamy go zgodnie ze schematem. Należy upewnić się, że dioda jest podłączona we właściwy sposób, a także sprawdź piny tranzystora, aby upewnić się, że pasują do schematu – w zależności od konkretnego typu tranzystora wyprowadzenia mogą być inne.

Aby przetestować działanie silnika, wystarczy po uruchomieniu układu wystukać prawidłową melodię, aby silnik się uruchomił i działał przez około pół sekundy. Jeśli nie, to należy sprawdzić połączenia i kierunek umieszczenia diody w układzie. Teraz pozostaje tylko umieścić układ w czymś, co możemy zamocować do naszych drzwi.

Obudowa i montaż na drzwiach

Zanim zmontujemy całość, należy dopasować do siebie wszystkie elementy i je przetestować – po sklejeniu rurki PCV będą już niemożliwe do przemieszczenia czy skrócenia, więc lepiej jest sprawdzić wszystko, zanim skleimy szkielet urządzenia w całość. Na **fotografii 1** pokazano przygotowane do montażu elementy.

W pierwszej kolejności konstruujemy uchwyt przycisku. Bierzemy jedno zakończenie rurki PCV i wiercimy w nim otwór przez środek, o średnicy odpowiedniej na nasz przycisk. Przycisk montujemy w otworze. Zatyczkę tę montujemy w jednym z otworów pięciodrożnego łącznika (tj. nie ten, który wskazuje w dół).

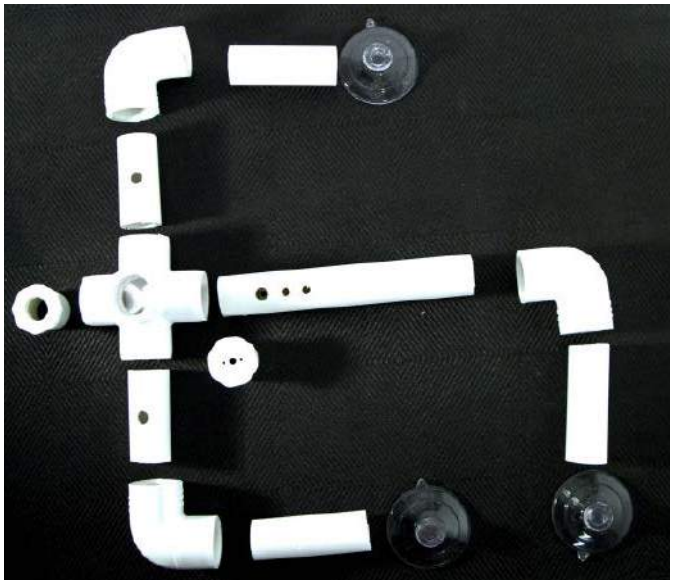
Drugie zakończenie rurki PCV wykorzystane zostanie jako uchwyt dla silnika. Wiercimy w nim otwór wystarczająco duży, aby wałek silnika mógł przez niego przejść. Można również zrobić go jeszcze większym, jeśli wybrany silnik ma wystające łożysko. W zależności od konstrukcji silnika można trochę oszlifować na grubość zatyczkę, aby więcej osi silnika wystawało z obudowy. Następnie umieszczamy w zatyczce silnik, by sprawdzić, czy wszystko pasuje. Jeśli jest zbyt ciasno, można szlifować wnętrze zatyczki.

Następnie za pomocą np. papierowego szablonu lokalizujemy na zatyczce otwory na śruby mocujące, które musimy wywiercić, aby zamocować silnik do zatyczki za pomocą dwóch niewielkich śrub – w przypadku tego silnika M2. Śruby powinny być osadzone możliwie głęboko, aby nie wystawały zbyt dużo powyżej zatyczki.

Kolejnym krokiem jest dobranie „ramion". Aby je wykonać, wkładamy najpierw kawałek rurki PCV o długości 125 cm. Nazwijmy to ramię „długim ramieniem". Na jednym końcu umieszczamy prosty łącznik. Drugi koniec wkładamy do centralnego łącznika naprzeciw przycisku. Ołówkiem lub markerem rysujemy linię wzdłuż środka góry i dna długiego ramienia. Na górnej stronie zaznaczamy dwa miejsca na otwory, jeden ok. 20 mm (3/4 cala) od łącznika, a drugie 37 mm (1 i 1/2 cala) od łącznika. Wiercimy tam otwory o średnicy ok. 4 mm (3/16 cala) w każdym z nich. To jest miejsce, gdzie zamontowane będą diody LED.

Następnie rysujemy linię wzdłuż dna, gdzie długie ramię łączy się z centralnym łącznikiem. Za pomocą piły nacinamy krótko rurkę od dołu w górę, aż powstanie otwór o szerokości około 12 mm (1/2 cala) na rurze (to jest miejsce, gdzie zamocowana będzie sprężynka dociskająca sensor urządzenia). Również na dole należy wywiercić otwór o średnicy 3 mm około 6 mm dalej w głąb rurki (dalej od centralnego łącznika). Przez ten otwór przewleczony zostanie przewód do sensora piezoelektrycznego.

Finalny element, jaki wykonamy, to „nogi", które trzymać będą układ do drzwi. Ich długość zależy od konstrukcji zamka, długości wałka silnika i ostatecznego projektu zacisku do obracania zamka, jaki opisany jest w dalszej części artykułu. W przypadku projektu autora wszystkie trzy miały długość 72 mm. Dopasowując je do konstrukcji, pamiętajmy, że lepiej jest je odciąć za długie i przyciąć do odpowiedniego rozmiaru później. Jeśli są za długie, silnik nie sięgnie zamka. Jeśli są za krótkie, przyssawki nie będą dosięgać do drzwi – układ oprze się na zamku.



Fotografia 1. Przygotowane do montażu elementy urządzenia

Po odcięciu nóg na odpowiednią długość osadzamy na ich końcach przyssawki (lub dowolny inny element, jakiego używamy do montażu urządzenia na naszych drzwiach).

Otwieranie zamka

Autor skonstruował prosty uchwyt, który z jednej strony zamocowany jest na osi silnika, a z drugiej strony chwyta za zamek od wewnętrznej strony drzwi. Elastyczny zacisk z giętej blachy łapie za zamek, obracając nim za pomocą silnika.

Najpierw musimy przygotować uchwyt. Bierzemy większy z przygotowanych arkuszy blachy i tniemy go na pół, aby otrzymać dwa kawałki. Sklejamy je taśmą, zaznaczając każdą stronę, aby wiedzieć, która strona jest „na zewnątrz" i zaznaczając jedną z dłuższych krawędzi jako „do góry". To wszystko pomoże nam utrzymać wszystko w porządku na bieżąco.

Mierzymy i zaznaczamy linię środkową od góry do dołu. 12 mm z każdej strony tej linii środkowej i 6 mm od górnej krawędzi zaznaczamy miejsce do wywiercenia otworów. W tych miejscach wiercimy otwory o średnicy 3 mm. Zaznaczenie punktów stemplem lub uderzenie młotkiem i gwoździem sprawi, że wiercenie będzie dokładniejsze.

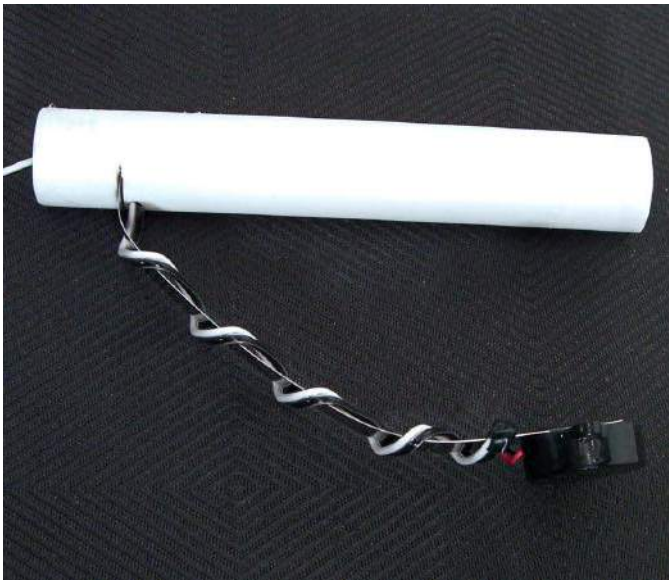
Mierzymy szerokość zatrzasku zamka i dzielimy ją przez 2. Ta odległość określa, ile krawędzi zagniemy w każdym kawałku metalu. Zaznaczamy tę odległość wzdłuż każdego paska. Gniemy jeden element tak, aby zwijał się w lewo, a drugi tak, aby zwijał się w prawo. Musimy upewnić się, że otwory na śruby w górnej części elementów są wyrównane, a zagięcia nie uniemożliwiają połączenia elementów na górze.

Przez otwory w górnej części płytek możemy teraz przewlec śruby, tak aby utworzyły odwróconą literę „Y" i umieszczamy oś silnika na górze, pomiędzy śrubami. Dodajemy z drugiej strony nakrętki i mocno



Fotografia 2. Uchwyt zamka zamontowany do osi silnika

eprasa.pl 1fc8f31a49



Fotografia 3. Detektor zainstalowany na sprężynie dociskowej

skręcamy, aż będą dobrze przymocowane. Niewielki luz pomiędzy metalem a wałem sprawi, że silnik zacznie się obracać, jeśli napotka zbyt duży opór. Konstrukcję taką pokazano na **fotografii 2**.

Uchwyt powinien lekko przylegać do zatrzasku zamka. Nie tak ściśle, aby był mocno zaciśnięty, ale też nie tak luźno, aby można było go obrócić bez obracania zamka. W razie potrzeby można wyregulować to poprzez dogięcie blaszek bocznych. Po dokonaniu prawidłowej regulacji na końcach śrub nakręcamy drugą parę nakrętek, aby je skontrować. Pomoże to zablokować je na miejscu i uniemożliwić im samodzielne odkręcanie.

Sprężyna detektora

Ostatni element, jaki należy wykonać, to sprężyna dociskająca detektor do drzwi. Jest ona wykonana z cienkiej blaszki, dzięki czemu układ jest bardziej przenośny. Mocujemy czujnik do metalowego paska, np. za pomocą taśmy dwustronnej, kleju na ciepło itp. Metoda jest dowolna. Przewody możemy owinąć wokół paska blachy, żeby były dobrze ułożone i nie przeszkadzały. Pasek mocujemy do długiego ramienia – rurki z PCV, w której wycinaliśmy wcześniej otwór. Pasek przewlekamy w szczelinę, którą przygotowaliśmy, a przewód wsuwamy w pobliski otwór, aby podłączyć go do elektroniki. Pasek z blachy powinien wysunąć się z wyczuwalnym oporem, aby nie wypadał.

Zamocowany na sprężynie dociskającej detektor powinien wyglądać jak na **fotografii 3**.

Podsumowanie i perspektywy na przyszłość

Najpierw upewnijmy się, że nogi mają odpowiednią długość. Upewnijmy się, że uchwyt zamka jest przymocowany do silnika, a nogi są na swoim miejscu. Spróbujmy przymocować układ do drzwi, tak aby zacisk pasował do zamka. Przy odrobinie szczęścia wszystko uda się od ręki. Zamiast przyssawek można użyć innych metod montażu – dwustronnej taśmy piankowej lub wkrętów przez zaślepki, jeśli nie mamy nic przeciwko robieniu otworów w drzwiach.

To już wszystko. Układ można przetestować i używać. Po testach zaprogramujmy własny szyfr i używajmy do woli. Być może na tym etapie będziemy musieli wprowadzić pewne zmiany, wymagające podłączenia modułu do komputera. Na przykład, jeśli silnik nie obraca się wystarczająco daleko, aby odblokować drzwi, trzeba zaktualizować szkic, aby silnik działał dłużej.

Jeśli jesteśmy ambitni, możemy natomiast ulepszyć lub zmienić ten projekt na wiele sposobów. Autor wymienia kilka pomysłów na rozwój systemu:

- dodanie mostka H do obwodu, aby mógł też zamykać, a nie tylko otwierać drzwi (potrzebne jest obracanie silnikiem w obie strony);
- można sprawić, aby działał w trybie cichym, usuwając czujnik stukowy i podłączając czujnik pojemnościowy (dotykowy) do klamki i rejestrując sekwencje dotknięć;
- można dodać potencjometr, aby wyregulować czułość sensora stukowego i inne wartości parametrów;
- warto zastanowić się nad redukcją wymiarów układu, aby cały system dało się zamaskować na drzwiach;
- dodanie kołatki, aby przekazać informację zwrotną przez drzwi. Mógłby wówczas oferować zabezpieczenie typu wezwanie-odpowiedź, w którym drzwi rozpoczynają sekwencję pukania, a użytkownik musi ją poprawnie zakończyć;
- usunąć czujnik piezoelektryczny i nagrywań naciśnięcia dzwonka lub innego ukrytego przycisku;
- zamienienie czujnika piezoelektrycznego na fotosensor w wizjerze – kod byłby przesyłany sekwencją mrugnięć latarki.

Nikodem Czechowski, EP

Źródło
https://www.instructables.com/Secret-Knock-Detecting-Door-Lock/

REKLAMA



eprasa.pl 1fc8f31a49

Kurs FPGA Lattice (15)

Pamięć EBR

W tej części kursu poznamy bardzo często używane peryferium, jakim jest blok pamięci EBR. Może pełnić funkcje pamięci ROM oraz RAM, a poza tym można go skonfigurować także do innych funkcjonalności.

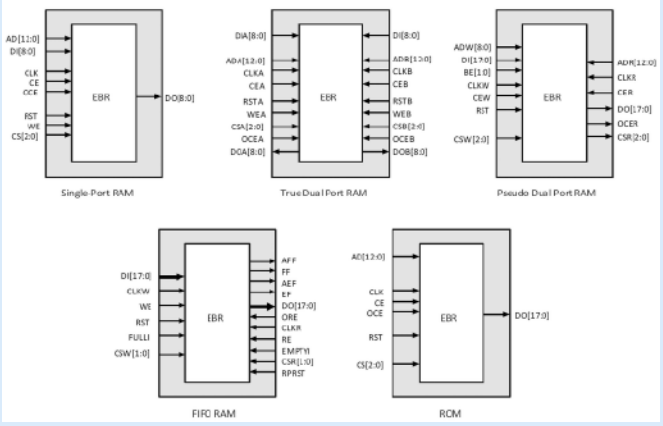
Pamięć można pokazać jako tablicę stałych lub zmiennych, w której wszystkie dane mają swój unikalny adres i każdy wpis ma tę samą liczbę bitów. W FPGA najczęściej stosujemy pamięci równoległe. To znaczy, że w jednym takcie zegarowym dokonujemy odczytu lub zapisu całej danej z pamięci, wskazanej przez wejście adresowe. Dla kontrastu, w przypadku pamięci szeregowych, dane odczytujemy lub zapisujemy do pamięci bit po bicie.

Zwykle mamy do czynienia z pamięciami, gdzie dane mają długość 8, 16 lub 32 bitów. Dane wchodzą i wychodzą z pamięci w sposób równoległy, zatem wejście i wyjście danych musi mieć tyle bitów, ile ma pojedynczy wpis. Najczęściej w równoległych pamięciach RAM, Flash czy EPROM, dostępnych w formie samodzielnych układów scalonych (np. 62256, 39Fxxx, 27Cxxx), wejście i wyjście realizowane jest w formie dwukierunkowego portu wyposażonego w bufor trójstanowy. Takie rozwiązanie ma na celu zmniejszenie liczby wyprowadzeń układu scalonego. Natomiast wewnątrz FPGA nie musimy przejmować się takimi ograniczeniami, więc stosuje się osobny port dla wejścia danych i osobny na wyjście.

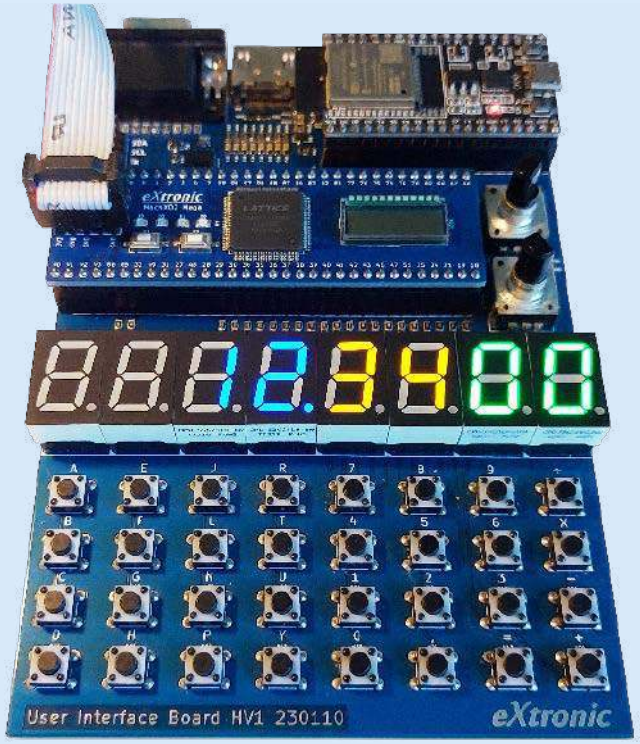
Wejście adresowe wskazuje nam, którą daną chcemy odczytać lub zapisać. Liczba bitów tego wejścia jest ściśle powiązana z pojemnością pamięci. Mając N-bitowe wejście adresowe, możemy zaadresować 2^N danych (obojętnie, ile bitów mają dane), czyli przykładowo 8-bitowa magistrala adresowa może obsłużyć maksymalnie 256 danych.

Pojemność pamięci to iloczyn liczby bitów danych oraz liczby wszystkich danych w pamięci. Przykładowo, pamięć obsługująca 1024 danych 8-bitowych ma pojemność 8 kilobitów. Pamięć 512 danych 16-bitowych również ma pojemność 8 kilobitów, ale to przecież nie jest taka sama pamięć! Podawanie pojemności pamięci bez informacji o jej organizacji stwarza możliwość nieporozumień. Z tego powodu dobrze jest podawać pojemność pamięci w formie 1024×8 lub 512×16.

W FPGA mamy dwa typy pamięci pod względem rodzaju używanych zasobów. Są to pamięć rozproszona (*distributed memory*) i blokowa (*block memory*). Pamięć rozproszona powstaje z uniwersalnych zasobów logicznych. Dzięki temu pamięć może mieć najróżniejszą



Rysunek 1. Możliwe tryby pracy bloku EBR





Poprzednie odcinki znajdują się pod adresem:
https://ulubionykiosk.pl/media

architekturę oraz nietypowe funkcjonalności. Jednak jej wadą jest to, że pamięć rozproszona jest bardzo zasobochłonna i może się okazać, że nawet mała i nieskomplikowana pamięć rozproszona wymaga zastosowania dużego i drogiego układu FPGA.

W praktyce pamięć rozproszona jest stosowana tylko do prostych tablic lub bardzo nietypowych zastosowań. W naszym kursie już raz spotkaliśmy się z pamięcią rozproszoną, choć nie używałem wtedy takiej nazwy. Mowa o module **Decoder7seg**, którego używaliśmy do sterowania 7-segmentowym wyświetlaczem LED. Moduł ten zawierał tablicę (co prawda opisaną za pomocą instrukcji case) z szesnastoma 7-bitowymi stałymi. Każdej 4-bitowej liczbie binarnej był przypisany 7-bitowy kod, sterujący segmentami wyświetlacza. W gruncie rzeczy jest to asynchroniczna pamięć ROM. Wejście liczby do pokazania na wyświetlaczu można by nazwać 4-bitowym wejściem adresowym, a wyjście segmentów byłoby 7-bitowym wyjściem danych z pamięci.

Pamięć jest peryferium na tyle często stosowanym, że większość układów FPGA ma w swojej strukturze umieszczone gotowe bloki pamięci. Możemy je konfigurować na wiele sposobów oraz łączyć, aby uzyskać większe bloki pamięci. W MachXO2 nazywają się EBR, czyli **Embedded Block RAM**, choć te bloki mogą funkcjonować także jako ROM. Układy MachXO2 mają także osobny blok pamięci Flash, ale nie będziemy go omawiać w tej części kursu.

Pamięć EBR

Pamięć EBR została dokładnie omówiona w rozdziale 2.5 dokumentacji MachXO2 Family Datasheet [2], a także w Memory Usage Guide for MachXO2 Devices [3]. Blok EBR może pracować w jednym

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie
https://ulubionykiosk.pl/media

eprasa.pl 1fc8f31a49

z pięciu możliwych trybów. Ich symbole wraz z wejściami i wyjściami pokazano na **rysunku 1**.

- **Single Port RAM** – Jest to najprostsza i najczęściej stosowana konfiguracja. Pamięć ma 13-bitowe wejście adresowe AD, 9-bitowe wejście danych DI, 9-bitowe wyjście danych DO, wejście zegarowe oraz kilka sygnałów sterujących.
- **Pseudo Dual Port RAM** – Pamięć ma dwie osobne magistrale adresowe oznaczone jako ADW (do zapisu) oraz ADR (do odczytu). Celem tego jest możliwość rozdzielenia operacji zapisywania i odczytywania, a co bardzo ważne, zegar i sygnały sterujące do odczytu i zapisu mogą być w zupełnie różnych domenach zegarowych. Najczęściej takie rozwiązanie wykorzystuje się w roli bufora, gdzie jeden moduł zapisuje pamięć jakimiś danymi, a inny moduł te dane odczytuje. Dzięki rozdzieleniu portów moduł zapisujący i odczytujący mogą mieć swobodny dostęp do pamięci w dowolnym momencie, bez konieczności stosowania jakichś systemów kontroli dostępu do pamięci.
- **True Dual Port RAM** – Jest to rozwinięcie powyższej pamięci w taki sposób, aby oba porty miały możliwość zapisu i odczytu po obu stronach pamięci. Porty oznaczane są literami A i B.
- **FIFO** – Jest to konfiguracja podobna do Pseudo Dual Port RAM, ale nie zawiera wejść adresowych. Pamięć tego typu służy do buforowania danych. Z jednej strony są zapisywane przez jakiś moduł, a z drugiej są odczytywane przez inny moduł. Idea polega na tym, że dane odczytywane są w takiej kolejności, w jakiej zostały zapisane (*First In First Out*).
- **ROM** – Działa tak samo jak Single Port RAM, ale nie ma możliwości zapisywania danych w trakcie pracy. Dane muszą być zapisane w osobnym pliku, który syntezuje się równolegle z innymi modułami w języku Verilog.

Pamięć EBR ma sumarycznie 9216 bitów, które można zorganizować na wiele sposobów, w zależności od wybranego trybu pracy. Ta „dziwna" liczba wynika z możliwości zapisywania danych 9-bitowych, których można zapisać 1024. Jednak najczęściej operujemy danymi 8-bitowymi, czyli bajtami. W takim przypadku dziewiąty bit danych pozostaje nieużywany i w pojedynczym bloku EBR możemy zgromadzić 1024 bajty. Wszystkie możliwe konfiguracje pokazano na **rysunku 2**.

Listing 1. Przykłady różnych pamięci

```
reg       Memory;               // 1
reg [7:0] Memory;               // 2
reg       Memory [0:15];        // 3
reg [7:0] Memory [0:15];        // 4
reg [7:0] Memory [15:0];        // 5
reg [7:0] Memory [16];          // 6
reg [7:0] Memory [0:15][0:255]; // 7
```

Memory Mode	Configurations
Single Port	8,192 x 1 4,096 x 2 2,048 x 4 1,024 x 9
True Dual Port	8,192 x 1 4,096 x 2 2,048 x 4 1,024 x 9
Pseudo Dual Port	8,192 x 1 4,096 x 2 2,048 x 4 1,024 x 9 512 x 18
FIFO	8,192 x 1 4,096 x 2 2,048 x 4 1,024 x 9 512 x 18

Rysunek 2. Możliwe opcje organizacji pamięci dla każdego trybu pracy

Listing 2. Kod pliku rom.v

```
// Plik rom.v
`default_nettype none
module ROM #(
 parameter ADDRESS_WIDTH = 16,                              // 1
 parameter DATA_WIDTH    = 8,                               // 2
 parameter MEMORY_DEPTH  = 2**ADDRESS_WIDTH,                // 3
 parameter MEMORY_FILE   = "data.mem"                       // 4
)(
 input wire Clock,
 input wire Reset,
 input wire ReadEnable_i,   // 5
 input wire [ADDRESS_WIDTH-1:0] Address_i,                  // 6
 output reg [   DATA_WIDTH-1:0] Data_o                      // 7
);
 / Deklaracja pamięci
    /* synthesis syn_romstyle = "EBR" */
    reg [DATA_WIDTH-1:0] Memory [0:MEMORY_DEPTH-1];         // 8

    // Kontrola czy pamięć mieści się w przestrzeni adresowej
    initial begin                                           // 9
        if(MEMORY_DEPTH > 2**ADDRESS_WIDTH)
            $fatal(0, "Required memory depth is larger than address space");
    end

    // Inicjalizacja pamięci plikiem
    initial begin                                           // 10
        $readmemh(MEMORY_FILE, Memory);
    end

    // Inicjalizacja pamięci poprzez przypisania
    initial begin                                           // 11
        Memory[0] = 8'h00;
        Memory[1] = 8'h35;
        Memory[2] = 8'h7A;
        Memory[3] = 8'hFF;
        // ...i tak dalej
    end

    // Inicjalizacja pamięci poprzez przypisania w pętli
    integer i;
    initial begin                                           // 12
        for(i=0; i<MEMORY_DEPTH; i=i+1) begin
            Memory[i] = 0;
        end
    end


    // Logika pamięci
    always @(posedge Clock, negedge Reset) begin
        if(!Reset)                                          // 13
            Data_o <= 0;
        else if(ReadEnable_i)                               // 14
            Data_o <= Memory[Address_i];
    end

endmodule
`default_nettype wire
```

Istnieje jedna zasadnicza różnica między klasycznymi pamięciami RAM a pamięciami EBR w FPGA. Klasyczne pamięci nie mają wejścia zegarowego. W celu odczytania danych wystarczyło ustawić żądany adres na wejściu adresowym, a dane na wyjściu pojawiały się po upływie czasu propagacji pamięci. **W FPGA musimy dodatkowo poczekać na zbocze rosnące sygnału zegarowego i dopiero wtedy możemy odczytywać dane na wyjściu.**

Moduł ROM

Zanim poznamy, jak utworzyć moduł obsługujący pamięć, musimy nauczyć się tworzyć obiekty pamięciowe w języku Verilog. Zobaczmy **listing 1**, w którym pokazano przykłady deklaracji różnych elementów przechowujących dane. W linii 1 widzimy coś, co już dobrze znamy. Jest to 1-bitowy rejestr, mogący przechowywać stan wysoki lub niski, który zostanie zaimplementowany w sprzęcie jako przerzutnik D. Linia 2 również niczym nowym nie zaskakuje. Jest to rejestr 8-bitowy, gdzie bity ponumerowane są od 7 do 0. Taka konstrukcja spowoduje syntezowanie ośmiu przerzutników D. Deklaracja w linii 3, choć podobna do linii 2, tworzy pamięć złożoną z 16 danych 1-bitowych, które oznaczone są indeksami od 0 do 15. Pamięci, w których dane są 1-bitowe, w gruncie rzeczy są dość podobne do rejestrów wielobitowych.

Najczęściej mamy do czynienia z pamięciami, które przechowują dane 8-bitowe. Przykład deklaracji pamięci, która przechowuje 16 danych 8-bitowych, widzimy w linii 4. Dane w tej pamięci ponumerowane są od 0 do 15. Linia 5 jest bardzo podobna do linii 4 z tą różnicą, że dane ponumerowane są w odwrotnej kolejności. Nie ma to praktycznego znaczenia. Większość deweloperów stosuje

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie
https://ulubionykiosk.pl/media

eprasa.pl 1fc8f31a49

sposób pokazany w linii 4, ale niektórzy wolą metodę z linii 5. Jest to kwestia indywidualnych upodobań.

W linii 6 widzimy deklarację takiej samej pamięci, co w linii 4 i 5, jednak zamiast numerować elementy „od–do", podajemy po prostu liczbę danych w pamięci. Jednak należy podkreślić, że taki zapis został wprowadzony dopiero ze standardem SystemVerilog. Niektóre syntezatory i symulatory języka Verilog również wspierają taki zapis, ale nie wszystkie.

Następnie w linii 7 widzimy przykład utworzenia pamięci dwuwymiarowej, składającej się z tablicy 16 rzędów i 256 kolumn danych 8-bitowych. Dostęp do tak utworzonych pamięci jest podobny do odczytywania danych w C i C++. W nawiasach kwadratowych podajemy, który elementy pamięci nas interesuje. Wyrażenie Memory[n] zwraca n-ty element pamięci, gdzie n może być zarówno stałą, jak i zmienną.

Nadszedł czas, aby pokazać, jak w języku Verilog opisać moduł pamięci tylko do odczytu. Przeanalizujmy kod z **listingu 2**. Zaprezentowany kod modułu pamięci jest napisany w taki sposób, aby był jak najbardziej uniwersalny. Moduł jest konfigurowany za pomocą trzech lub czterech parametrów w liniach od 1 do 4. Są to:

- **ADDRESS_WIDTH** – Szerokość wejścia adresowego; wynika z niej maksymalna możliwa pojemność pamięci;
- **DATA_WIDTH** – Szerokość danych zapisanych w pamięci, najczęściej to 8, 16 lub 32 bity;
- **MEMORY_DEPTH** – Liczba danych, jaką pamięć ma przechowywać. Jest to parametr opcjonalny i ma zastosowanie tylko wtedy, kiedy pamięć ma mieć mniejszą pojemność, niż wynika to z przestrzeni adresowej. Jeżeli parametr ten zostanie pominięty podczas tworzenia instancji modułu, to zostanie wykorzystana jego domyślna wartość, obliczana w linii 3 na podstawie parametru ADDRESS_WIDTH. Przykładowo, jeżeli wejście adresowe ma 16 bitów, to przestrzeń adresowa pozwala zaadresować 65536 danych, ale pamięć można skonfigurować tak, by jej pojemność była mniejsza;
- **MEMORY_FILE** – Nazwa pliku z danymi służącymi do inicjalizacji pamięci. Nazwę pliku należy wziąć w cudzysłów.

W liniach 5, 6 i mamy porty modułu. Wejście zegarowe i resetujące działają tak samo jak we wszystkich dotychczas omawianych modułach, więc nie będę ich opisywał po raz kolejny.

- **ReadEnable_i** – Wejście uruchamiające odczyt z pamięci. Jeżeli jest w stanie niskim, to pamięć ignoruje wejście adresowe i zegarowe. Ta funkcjonalność jest zaimplementowana na poziomie bloku EBR. Usunięcie tego wyjścia nie spowoduje żadnej oszczędności w zasobach FPGA. Jeżeli nie interesuje nas taka funkcjonalność, można do tego wejścia na stałe podać 1'b1;
- **Address_i** – Wejście, na którym podajemy adres danej, którą chcemy odczytać w następnym takcie zegarowym;
- **Data_o** – Wyjście danych z pamięci.

Następnie w linii 8 deklarujemy pamięć, którą chcemy powołać do życia. Szerokość danych oraz ich liczbę podajemy za pomocą zdefiniowanych wcześniej parametrów. Niespodzianką jest komentarz `/* synthesis syn_romstyle = "EBR" */`. Jest to dyrektywa syntezatora Lattice Synthesis Engine, informująca, w jaki sposób pamięć ma zostać zaimplementowana w strukturze FPGA. Możliwe są następujące opcje:

- **auto** – Syntezator podejmie decyzję automatycznie na podstawie rozmiaru pamięci, dostępnych zasobów w FPGA, a także wybranej strategii syntezy;
- **logic** – Pamięć zostanie zaimplementowana z wykorzystaniem uniwersalnych zasobów logicznych;
- **EBR** – Pamięć zostanie zaimplementowana z wykorzystaniem bloków pamięci EBR.

Brak takiego komentarza jest równoznaczny z wybraniem opcji **auto**. Więcej na ten temat znajdziesz w instrukcji **Lattice Synthesis Engine for Diamond User Guide** [link 4].

Listing 3. Kod pliku data.mem

```
// Plik data.mem
0F 1E 2D 3C 4B // Komentarz a
5A 69 78 87 96 // Komentarz b
A5 B4 C3 D2 E1 // Komentarz c
F0
```

W dalszej części mamy kilka bloków **initial**. Są to bloki, które wykonują się jednorazowo na początku syntezy lub symulacji i definiują warunki startowe układu po uruchomieniu aplikacji. Pierwszy z nich (linia 9) stanowi proste zabezpieczenie na wypadek błędnej konfiguracji pamięci, kiedy użytkownik chce utworzyć pamięć o większym rozmiarze, niż pozwala na to przestrzeń adresowa. Maksymalna liczba elementów pamięci to $2^{ADDRESS_WIDTH}$, zatem żądany rozmiar pamięci, określony parametrem **MEMORY_DEPTH** nie może być większy. Jeżeli taka sytuacja wystąpi, to wtedy wykonujemy instrukcję **$fatal**, która wyświetla stosowny komunikat na konsoli i natychmiast przerywa dalsze operacje.

Istnieje kilka sposobów, aby zainicjalizować pamięć. W przypadku pamięci ROM wręcz musimy ją zainicjalizować jakimiś konkretnymi danymi, które będą wykorzystywane w naszej aplikacji. Najpopularniejszy i najbardziej uniwersalny sposób przedstawiono w linii 10. W bloku **initial** wykorzystujemy instrukcję **$readmemh**, która otwiera plik podany w pierwszym argumencie i ładuje jego zawartość do pamięci, wskazanej w argumencie drugim. Plik otwierany za pomocą tego polecenia powinien zawierać dane zapisane w notacji szesnastkowej. Dane mogą być oddzielone spacjami, enterami i tabulatorami, a ponadto w pliku mogą znajdować się także komentarze. Przykład pliku zdatnego do załadowania poprzez **$readmemh** przedstawiono na **listingu 3**. Innym sposobem jest wykorzystanie **$readmemb**, która różni się tym, że dane zapisane są w formie binarnej.

Taki sposób inicjalizacji pamięci jest bardzo praktyczny, ponieważ zawartość pamięci oddzielamy od modułu pamięci. Ponadto nazwę pliku z zawartością pamięci możemy przekazać w instancji za pomocą parametru, co zobaczymy za chwilę, analizując testbench. Często zdarza się tak, że plik z zawartością pamięci jest generowany przez jakiś inny program, skrypt w Pythonie lub chociażby arkusz w Excelu – oddzielenie plików z uniwersalnym modułem pamięci i jej zawartością jest po prostu wygodne. W ten sposób można utworzyć wiele instancji tego samego modułu pamięci, a każdą z nich można zainicjalizować innym plikiem z danymi.

W linii 11 przedstawiono inny sposób, najprostszy z możliwych. W bloku **initial** wszystkim danym w pamięci przypisujemy jakieś wartości ręcznie, wpisując je za pomocą operatora =, tak jak byśmy je wpisywali do normalnych zmiennych. Taki sposób może być przydatny tylko, kiedy pamięć jest bardzo mała. Ponadto tak zainicjalizowana pamięć przestaje być uniwersalna, bo każda instancja tego modułu ma taką samą zawartość pamięci.

Sposób przedstawiony w linii 12 ma większe zastosowanie dla pamięci RAM niż ROM. Wykorzystuje on pętlę for, która każdemu elementowi pamięci przypisuje wartość początkową. Najczęściej przypisuje się 0 lub FF, czyli wszystkie bity ustawiane są w stan niski lub wysoki. Można także pokusić się o zastosowanie jakiegoś wzoru, który przypisze elementom tablicy wartość na podstawie ich adresu lub jeszcze jakichś innych zmiennych.

Z tych trzech wymienionych sposobów wybieramy tylko jeden. W dalszej części kursu i we wszystkich kolejnych odcinkach moduł pamięci ROM będzie wykorzystywał inicjalizację poprzez odczytanie pliku z danymi, natomiast moduły pamięci RAM będą inicjalizowane zerami.

Przejdźmy wreszcie do jedynego bloku **always** w tym module, gdzie zawarta jest właściwa logika pamięci ROM. Blok ten reaguje na zbocze rosnące sygnału zegarowego i zbocze opadające sygnału resetującego. Jeżeli **Reset** jest w stanie niskim, to w linii 13 resetujemy rejestr wyjściowy **Data_o** i tylko tyle. W żaden sposób nie

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie
https://ulubionykiosk.pl/media

eprasa.pl 1fc8f31a49

resetujemy zawartości pamięci. Jeżeli nie, to sprawdzamy, czy wejście **ReadEnable_i** jest w stanie wysokim (linia 14) i jeżeli tak, to do rejestru wyjściowego **Data_o** wpisujemy wpis z pamięci, wskazywany przez aktualny stan wejścia **Address_i**.

I to cała logika pamięci ROM! Całkiem proste, prawda?

Testbench pamięci ROM

Opracujemy teraz testbench, aby sprawdzić w symulatorze, jak działa moduł pamięci ROM. Jego kod widzimy na **listingu 4**. Zadanie testbencha będzie bardzo proste – ma on jedynie odczytać wszystkie dane z pamięci.

Standardowo rozpoczynamy od utworzenia generatora sygnału zegarowego, po czym deklarujemy zmienne (linia 1). W linii 2 tworzymy instancję testowanego modułu pamięci. Konfigurujemy go w taki sposób, aby przechowywał dane 8-bitowe, a liczbę bitów wejścia adresowego ustawiamy na 4 bity. Oznacza to, że w przestrzeni adresowej będziemy mogli pomieścić tylko 16 danych i będziemy wykorzystywać całą pamięć. W takiej sytuacji można by pominąć linię 3. Podczas symulacji będziemy zmieniać stan wejścia **ReadEnable_i**, ale to wejście można by ustawić na stałe w stan wysoki (linia 4).

Zanim rozpoczniemy sekwencję testową, musimy jeszcze utworzyć zmienną **i**, która jest iteratorem pętli for (linia 5). Niestety w Verilogu nie ma możliwości, by utworzyć iterator pętli w pierwszym argumencie pętli **for**, tak jak w C++, więc trzeba to zrobić przed rozpoczęciem bloku **always** lub **initial**.

Sekwencję testową rozpoczynamy od wyświetlenia kilku komunikatów za pomocą instrukcji **$display**. Między innymi wyświetlamy liczbę elementów pamięci określoną parametrem **MEMORY_DEPTH** (linia 6).

W linii 7 tworzymy pętlę **for**, której zadaniem będzie wyświetlenie na konsoli wszystkich danych z pamięci. Składnia pętli for jest taka sama jak w języku C z wyjątkiem tego, że iterator pętli inkrementujemy instrukcją `i=i+1`, a nie `i++`, ponieważ operator `++` w Verilogu nie istnieje (a szkoda, boby się bardzo przydał). Elementy pamięci wyświetlamy za pomocą funkcji **$display**. Warto podkreślić, że cała ta operacja jest wykonywana przez testbench w zerowym czasie, a nie przez testowany moduł pamięci. Dopiero w linii 8 ustawiamy stan zmiennej **Reset** na wysoki, co uruchamia testowany moduł i rozpoczyna właściwą symulację.

Czekamy na kolejne zbocze rosnące sygnału zegarowego (linia 9), po czym ustawiamy **ReadEnable** w stan wysoki, aby moduł pamięci reagował na sygnał zegarowy i wejście adresowe (linia 10).

W linii 11 mamy pętlę iterującą po wszystkich elementach pamięci, podobnej do pętli, którą widzieliśmy przed chwilą. Jest jednak zasadnicza różnica między tymi dwoma pętlami. Ta druga modyfikuje zmienną **Address**, która połączona jest z wejściem adresowym pamięci. Ponadto po każdej zmianie czekamy na zbocze rosnące sygnału zegarowego. W ten sposób nie tylko odczytujemy pamięć

Listing 4. Kod pliku rom_tb.v

```
// Plik rom_tb.v

`timescale 1ns/1ns
`default_nettype none
module ROM_tb();

    parameter      CLOCK_HZ       = 10_000_000;
    parameter real HALF_PERIOD_NS = 1_000_000_000.0 / (2 * CLOCK_HZ);

    // Generator sygnału zegarowego
    reg Clock = 1'b1;
    always begin
        #HALF_PERIOD_NS;
        Clock = !Clock;
    end

    // Zmienne
    reg        Reset      = 1'b0;                                  // 1
    reg        ReadEnable = 1'b0;
    reg  [3:0] Address;
    wire [7:0] Data;

    // Instancja testowanego modułu
    ROM #(                                                         // 2
        .ADDRESS_WIDTH(4),
        .DATA_WIDTH(8),
        .MEMORY_DEPTH(16),                                         // 3
        .MEMORY_FILE("data.mem")
    ) DUT(
        .Clock(Clock),
        .Reset(Reset),
        .ReadEnable_i(ReadEnable),                                 // 4
        .Address_i(Address),
        .Data_o(Data)
    );

    // Eksport wyników symulacji
    initial begin
        $dumpfile("rom.vcd");
        $dumpvars(0, ROM_tb);
    end

    // Sekwencja testowa
    integer i;                                                     // 5
    initial begin
        $timeformat(-6, 3, "us", 12);
        $display("===== START =====");
        $display("MEMORY_DEPTH: %0d", DUT.MEMORY_DEPTH);           // 6

        // Wyświetl zawartość całej pamięci na konsoli
        for(i=0; i<DUT.MEMORY_DEPTH; i=i+1) begin                  // 7
            $display("Memory[%d] = %h", i, DUT.Memory[i]);
        end

        @(posedge Clock);
        Reset <= 1'b1;                                             // 8

        @(posedge Clock);                                          // 9

        // Odczytaj wszystkie dane z pamięci
        ReadEnable <= 1'b1;                                        // 10
        for(i=0; i<DUT.MEMORY_DEPTH; i=i+1) begin                  // 11
            Address <= i;
            @(posedge Clock);
        end

        ReadEnable <= 1'b0;                                        // 12

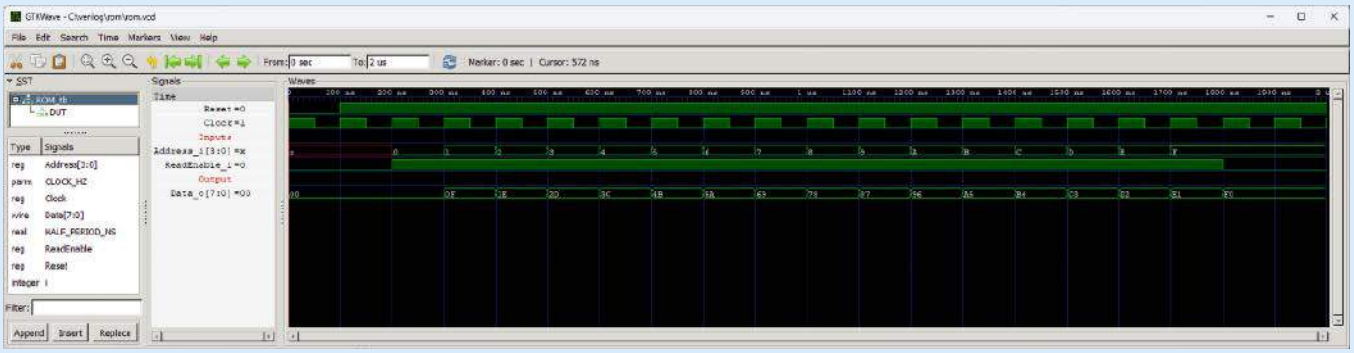
        repeat(2) @(posedge Clock);

        $display("===== END =====");
        $finish;
    end

endmodule
`default_nettype wire
```

Listing 5. Kod pliku rom.bat

```
@echo off
iverilog -o rom.o rom.v rom_tb.v
vvp rom.o
del rom.o
```



Rysunek 3. Przebiegi sygnałów podczas symulacji pamięci ROM

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie
https://ulubionykiosk.pl/media

eprasa.pl 1fc8f31a49

jako taką, ale badamy reakcje testowanego modułu na zmiany sygnałów wejściowych, co będziemy obserwować na wykresie (będzie o tym kilka akapitów dalej). Pozostaje już tylko ustawić **ReadEnable** w stan niski (linia 12) i poczekać dwa takty zegarowe przed zakończeniem symulacji, aby wykres był bardziej czytelny.

Listing 5 to zawartość skryptu *rom.bat*, który służy do wykonania symulacji programem Icarus Verilog, który omawialiśmy w 12 odcinku kursu. Po przeprowadzeniu symulacji powinniśmy zobaczyć komunikaty, jakie pokazano na **listingu 6**.

Musimy teraz otworzyć plik *rom.vdc*, który zawiera informacje o zmianach wszystkich symulowanych sygnałów. Otwórz ten plik za pomocą GTKWave i skonfiguruj program, by otrzymać rezultat taki, jak pokazano na **rysunku 3**. Widzimy, że zmienna **Address** w każdym cyklu zegarowym zmienia swój stan, odliczając od 0 do F szesnastkowo, czyli do 15 dziesiętnie. Zwróć uwagę, że dane odpowiadające żądanemu adresowi pojawiają się dopiero w następnym takcie zegarowym. Dzieje się tak dlatego, że bloki pamięci EBR są zsynchronizowane z zegarem – w związku z tym, przed odczytaniem danych z pamięci musimy mieć na uwadze to opóźnienie. Powinny szczególnie o tym pamiętać osoby przyzwyczajone do klasycznych pamięci równoległych, które funkcjonują asynchronicznie i wyrzucają dane na wyjście od razu po zmianie wejścia adresowego (mówię tutaj także o sobie samym – przyzwyczajenie się do takiego sposobu działania pamięci było dla mnie mocno irytujące).

Listing 6. Log z konsoli po wykonaniu symulacji

```
VCD info: dumpfile rom.vcd opened for output.
===== START =====
MEMORY_DEPTH: 16
Memory[          0] = 0f
Memory[          1] = 1e
Memory[          2] = 2d
Memory[          3] = 3c
Memory[          4] = 4b
Memory[          5] = 5a
Memory[          6] = 69
Memory[          7] = 78
Memory[          8] = 87
Memory[          9] = 96
Memory[         10] = a5
Memory[         11] = b4
Memory[         12] = c3
Memory[         13] = d2
Memory[         14] = e1
Memory[         15] = f0
===== END =====
rom_tb.v:71: $finish called at 400 (1ns)
```

Pamięć RAM

Pamięć RAM jest bardzo podobna do pamięci ROM. Z tego powodu analizując kod z **listingu 7**, przedstawimy tylko istotne różnice. Pod względem funkcjonalnym ROM od RAM różni się tylko tym, że można do niej zapisywać dane podczas pracy.

Pierwszą zmianę widzimy już na liście portów modułu. W linii 1 pojawiło się wejście **WriteEnable_i**. Stan wysoki na tym wejściu będzie powodować, że pamięć zapisze pod wskazanym adresem dane, które zostały doprowadzone do wejścia **Data_i** (linia 2).

W linii 3 deklarujemy pamięć. Składnia języka Verilog jest w tym przypadku identyczna. Zmienia się tylko instrukcja dla syntezatora. Parametr **syn_ramstyle** może przyjmować następujące wartości:

- **block_ram** – implementacja pamięci w blokach EBR (uwaga, aby uzyskać taki sam efekt dla pamięci ROM, trzeba było wpisać „**EBR**”);
- **distributed** – implementacja w postaci pamięci rozproszonej;
- **registers** – implementacja w przerzutnikach;
- **no_rw_check** – ta opcja ma zastosowanie tylko do dwuportowych pamięci RAM i pozwala wyeliminować logiką kontrolującą jednoczesny zapis do tej samej komórki pamięci z obu portów. Jeżeli nie ma ryzyka, że taka sytuacja wystąpi, to można tę logikę usunąć, aby zaoszczędzić trochę zasobów. Należy tę opcję dodać po przecinku do jednej z trzech powyższych opcji.

W linii 4 mamy blok **initial**, w których inicjalizujemy stan początkowy pamięci. Wykorzystując pętlę **for**, każdemu elementowi pamięci przypisujemy wartość zerową. Możliwe jest także zainicjalizowanie pamięci RAM wartościami odczytanymi z pliku, tak jak to robiliśmy w przypadku pamięci ROM. Jednak w praktycznych rozwiązaniach najczęściej zależy nam, aby po starcie aplikacji pamięć RAM była czysta, a nie zapełniona jakimiś danymi.

Dalej mamy blok **always** z logiką pamięci. Jedyne, co zostało dodane, to linia 5, w której sprawdzamy, czy wejście **WriteEnable_i** jest w stanie wysokim. Jeżeli tak, to do komórki pamięci wskazywanej przez wejście adresowe wpisywane są dane z wejścia **Data_i**.

Listing 7. Kod pliku ram.v

```
// Plik ram.v

`default_nettype none
module RAM #(
 parameter ADDRESS_WIDTH = 16,
 parameter DATA_WIDTH    = 8,
 parameter MEMORY_DEPTH  = 2**ADDRESS_WIDTH
)(
 input wire Clock,
 input wire Reset,
 input wire ReadEnable_i,
 input wire WriteEnable_i,                                  // 1
 input wire [ADDRESS_WIDTH-1:0] Address_i,
 input wire [   DATA_WIDTH-1:0] Data_i,                     // 2
 output reg [   DATA_WIDTH-1:0] Data_o
);
 / Deklaracja pamięci
    /* synthesis syn_ramstyle = "block_ram" */
    reg [DATA_WIDTH-1:0] Memory [0:MEMORY_DEPTH-1];    // 3

    // Kontrola czy pamięć mieści się w przestrzeni adresowej
    initial begin
        if(MEMORY_DEPTH > 2**ADDRESS_WIDTH)
            $fatal(0, "Required memory depth is larger than address space");
    end

    // Inicjalizacja pamięci zerami
    integer i;
    initial begin                                           // 4
        for(i=0; i<2**ADDRESS_WIDTH; i=i+1) begin
            Memory[i] = 0;
        end
    end

    // Logika pamięci
    always @(posedge Clock, negedge Reset) begin
        if(!Reset) begin
            Data_o <= 0;
        end else begin
            if(ReadEnable_i)
                Data_o <= Memory[Address_i];
            if(WriteEnable_i)                               // 5
                Memory[Address_i] <= Data_i;
        end
    end

endmodule
`default_nettype wire
```

Testbench pamięci RAM

Opracujmy testbench, który zapisze do wszystkich komórek pamięci losowo wybrane wartości, a następnie wszystkie je odczyta. Kod testbencha modułu pamięci RAM pokazano na **listingu 8**. Jest on podobny do testbencha pamięci ROM, więc omówimy tylko istotne różnice.

Dotychczas do eksportowania wyników symulacji stosowaliśmy taką instrukcję, jaką przedstawiono w linii 4. Funkcja **$dumpvars()**, której pierwszym argumentem jest 0, eksportuje wszystkie zmienne z modułu podanego w drugim argumencie oraz jego modułach podrzędnych. Niestety jednak ta funkcja nie eksportuje tablic pamięci. Aby mieć możliwość zaglądania do środka pamięci i obserwowania zmian poszczególnych danych wewnątrz pamięci, musimy wyeksportować każdą z interesujących nas komórek pamięci. Z tego powodu w linii 5 tworzymy pętlę for, która wywołuje funkcję **$dumpvars()** dla każdego elementu tablicy (linia 6).

Przeskoczmy teraz do linii 11. Znajduje się tam coś, czego w tym kursie jeszcze nie widzieliśmy – jest to **task**. W języku Verilog mamy możliwość grupowania często wykorzystywanego kodu za pomocą

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie
https://ulubionykiosk.pl/media

eprasa.pl 1fc8f31a49

funkcji i tasków, podobnie jak to się dzieje w różnych językach programowania. Taskom i funkcjom poświęcimy osobny odcinek, a w tym odcinku poznamy tylko elementarne podstawy tasków.

Największa różnica między taskiem a funkcją jest taka, że funkcja w założeniu ma służyć do obliczenia czegoś, a task ma wykonywać jakąś czynność. Funkcja pobiera co najmniej jeden argument i zwraca jakąś wartość, a co najważniejsze, musi wykonać się w zerowym czasie. Funkcja nie może zawierać żadnych opóźnień ani wywoływać jakichkolwiek innych operacji, które zawierają opóźnienia. Task służy do wykonywania jakichś zadań, które mogą być natychmiastowe, ale także mogą trwać przez jakiś czas. Może mieć jakieś wejścia, ale nie musi. Nie zwraca żadnych wartości, tak jak to robi funkcja, chociaż task może mieć wyjścia, tak jak moduły mają porty wejściowe i wyjściowe. Task może modyfikować różne zmienne wewnątrz modułu, w którym on się znajduje.

W linii 8 tworzymy task, który zapisuje bajt danych do pamięci RAM. Ma on dwa porty wejściowe: **Adr**, który określa adres modyfikowanej komórki pamięci oraz **Dat**, czyli dane do zapisania. Ponieważ task może modyfikować zmienne modułu, dobrze jest, aby jego wejścia i wyjścia nazywały się inaczej niż inne zmienne wykorzystywane w kodzie. Deklaracja portów tasku jest bardzo podobna do portów modułu.

Jeżeli task ma zawierać więcej niż jedną instrukcję, to musimy je objąć blokiem `begin… end`, pomimo że wszystkie taski zaczynamy słowem kluczowym **task** i kończymy **endtask**. Wewnątrz tasku jedynie przepisujemy wejścia **Adr** i **Dat** do rejestrów **Address** oraz **DataIn**, utworzonych w liniach 2 i 3, które doprowadzone są do wejść testowanej pamięci. Aby pamięć zapisała dane, musimy jeszcze tylko ustawić sygnał **WriteEnable** w stan wysoki (linia 1). Pozostaje już tylko poczekać na zbocze rosnące sygnału zegarowego i można opuścić task.

Wróćmy do linii 7. Rozpoczynamy w tym miejscu pętlę, która iterując po wszystkich elementach pamięci, będzie przypisywać im losowo wybrane wartości. W linii 8 widzimy wywołanie tasku **WriteData**, który przyjmuje dwa argumenty. Pierwszym z nich jest iterator pętli i, zmieniający się od 0 do 15. Drugim jest wartość zwracana przez funkcję **$urandom_range(x, y)**. Ta funkcja zwraca nam losowo wybraną liczbę całkowitą bez znaku, znajdującą się w zakresie od x do y.

Po dokonaniu zapisu do wszystkich szesnastu elementów pamięci musimy pamiętać o sprowadzeniu linii **WriteEnable** w stan niski. Bez tego w każdym cyklu zegarowym pamięć zapisywałaby dane, jakie aktualnie znajdują się na wejściu **Data_i** pod adres, jaki jest wskazywany przez **Address_i**. W tym celu utworzymy task podobny do tego, który omawialiśmy przed chwilą. W linii 12 rozpoczyna się task **WriteEnd**, który zmiennej **WriteEnable** przypisuje stan niski, a zmiennym **Address** oraz **DataIn** przypisujemy stan nieokreślony X, aby podkreślić to, że zmienne te nie mają znaczenia, kiedy **WriteEnable** jest w stanie niskim (**Address** jest używany także do odczytu pamięci, ale wtedy **ReadEnable** musi być w stanie wysokim).

Testbench gotowy. Na **listingu 9** pokazano skrypt BAT, który umożliwi przeprowadzenie symulacji w symulatorze Icarus Verilog. W wyniku symulacji powstaje plik *ram.vcd*, którym otwieramy przeglądarką GTKWave. Skonfiguruj ją jak tak, by uzyskać przebiegi widoczne na **rysunku 4**. Widzimy, że w pierwszej części symulacji na wejściu **Address_i** pojawiają się adresy kolejno od 0 do F (lub dziesiętnie: 15), a na wejściu **Data_i** są losowo wybrane dane (uwaga, przy ponownym uruchomieniu symulacji zostaną wylosowane dokładnie takie same liczby!). Dane te pojawiają się w pamięci dopiero po wystąpieniu zbocza rosnącego sygnału zegarowego. W drugiej części symulacji dokonujemy odczytu danych z pamięci w taki sam sposób, jak to robiliśmy w pamięci ROM.

Listing 8. Kod pliku ram_tb.v

```
// Plik ram_tb.v

`timescale 1ns/1ns
`default_nettype none
module RAM_tb();

    parameter CLOCK_HZ            = 10_000_000;
    parameter real HALF_PERIOD_NS = 1_000_000_000.0 / (2 * CLOCK_HZ);

    // Generator sygnału zegarowego
    reg Clock = 1'b1;
    always begin
        #HALF_PERIOD_NS;
        Clock = !Clock;
    end

    // Zmienne
    reg        Reset       = 1'b0;
    reg        ReadEnable  = 1'b0;
    reg        WriteEnable = 1'b0;                          // 1
    reg  [3:0] Address;                                     // 2
    reg  [7:0] DataIn;                                      // 3
    wire [7:0] DataOut;
    integer    i;

    // Instancja testowanego modułu
    RAM #(
        .ADDRESS_WIDTH(4),
        .DATA_WIDTH(8)
    ) DUT(
        .Clock(Clock),
        .Reset(Reset),
        .ReadEnable_i(ReadEnable),
        .WriteEnable_i(WriteEnable),
        .Address_i(Address),
        .Data_i(DataIn),
        .Data_o(DataOut)
    );

    // Eksport wyników symulacji
    initial begin
        $dumpfile("ram.vcd");
        $dumpvars(0, RAM_tb);                               // 4

        // Eksportuj także wszystkie dane z pamięci
        for(i=0; i<DUT.MEMORY_DEPTH; i=i+1) begin           // 5
            $dumpvars(2, DUT.Memory[i]);                    // 6
        end
    end

    // Sekwencja testowa
    initial begin
        $timeformat(-6, 3, "us", 12);
        $display("===== START =====");
        $display("MEMORY_DEPTH: %d", DUT.MEMORY_DEPTH);
        $display("        Time Address DataIn DataOut");
        $monitor("%t       %H      %H       %H",
            $realtime,
            Address,
            DataIn,
            DataOut
        );

        @(posedge Clock);
        Reset <= 1'b1;
        @(posedge Clock);

        // Zapis losowych danych do pamięci
        for(i=0; i<=15; i=i+1) begin                        // 7
            WriteData(i, $urandom_range(8'h00, 8'hFF));     // 8
        end
        WriteEnd();                                         // 9

        // Odczytaj wszystkie dane z pamięci
        ReadEnable <= 1'b1;                                 // 10
        for(i=0; i<=15; i=i+1) begin
            Address <= i;
            @(posedge Clock);
        end

        ReadEnable <= 1'b0;

        repeat(2) @(posedge Clock);

        $display("===== END =====");
        $finish;
    end

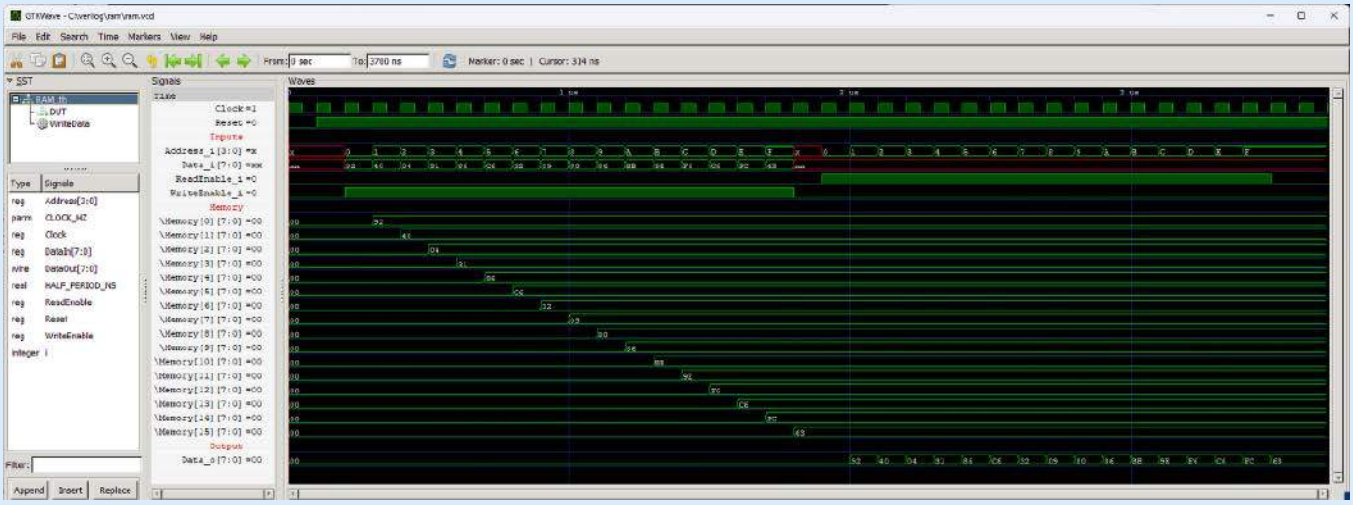
    // Task zapisujący dane do pamięci
    task WriteData(input [3:0] Adr, input [7:0] Dat);       // 11
        begin
            Address     <= Adr;
            DataIn      <= Dat;
            WriteEnable <= 1'b1;
            @(posedge Clock);
        end
    endtask

    // Task kończący zapis danych
    task WriteEnd();                                        // 12
        begin
            Address     <= 4'dX;
            DataIn      <= 8'dX;
            WriteEnable <= 1'b0;
            @(posedge Clock);
        end
    endtask

endmodule
`default_nettype wire
```

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie
https://ulubionykiosk.pl/media

eprasa.pl 1fc8f31a49



Rysunek 4. Przebiegi sygnałów podczas symulacji pamięci RAM

Pamięć Pseudo Dual Port RAM

Pamięć pseudodwuportowa to taka pamięć, gdzie dostępne są dwa porty adresowe. Jeden wskazuje adres danych, które chcemy odczytać, a drugi wskazuje adres danych do zapisu. Ponadto operacje zapisu i odczytu mogą być taktowane zupełnie różnymi zegarami. Z tego powodu pamięć tego typu świetnie nadaje się jako bufor pomiędzy dwiema domenami zegarowymi.

Z tego powodu mamy kilka zmian na liście portów modułu. Na **listingu 10** w liniach 1 i 2 widzimy dwa osobne wejścia zegarowe – jedno do operacji odczytu danych z pamięci, drugie do zapisu. Analogicznie, w liniach 3 i 4 tworzymy dwa osobne porty adresowe.

W przeciwieństwie do omawianych wcześniej pamięci ROM i RAM, w przypadku pamięci dwuportowych mamy dwa bloki always. Ten, który zaczyna się z linii 5, obsługuje operację odczytu, a ten z linii 6 odpowiedzialny jest za zapis. Zwróć uwagę na listę wrażliwości instrukcji **always**. Mamy tam dwa różne sygnały zegarowe. Ponadto blok odpowiedzialny za zapis nie reaguje na sygnał zegarowy. Nie ma takiego powodu – rejestr wyjściowy pamięci **Data_o** służy tylko do odczytywania danych i zerowany jest przez blok always związany z odczytem.

I to wszystko, jeżeli chodzi o pamięć pseudodwuportową. Okazuje się to bardzo proste. Możemy także utworzyć pamięć prawdziwie dwuportową (true dual port RAM), który ma dwa porty zdolne do zapisu i odczytu. Z uwagi na długość tego artykułu nie będziemy omawiać pamięci prawdziwie dwuportowej, ale powinieneś być w stanie opracować ją samodzielnie, łącząc w jedną całość kody pamięci RAM oraz pseudodwuportowej pamięci RAM.

W tym odcinku kursu nie będziemy omawiać zastosowania pamięci EBR skonfigurowanej jako FIFO.

Testbench pamięci Pseudo Dual Port RAM

Testbench pamięci dwuportowej, jak można się spodziewać, jednym portem będzie zapisywał losowe dane, a drugim będzie je odczytywać w tym samym czasie. Kod testbencha przedstawiono na **listingu 11**.

W naszej symulacji nowością będą dwa różne sygnały zegarowe. W liniach 1 i 2 definiujemy częstotliwość **CLOCK_READ_HZ** oraz **CLOCK_WRITE_HZ** na odpowiednio 10 MHz i 15 MHz. Następnie tworzymy generatory sygnałów zegarowych do odczytu i zapisu pamięci RAM w liniach 3 i 4.

Listing 9. Kod pliku ram.bat

```
@echo off
iverilog -o ram.o ram.v ram_tb.v
vvp ram.o
del ram.o
```

Listing 10. Kod pliku ram_pdp.v

```
// Plik ram_pdp.v

`default_nettype none
module PseudoDualPortRAM #(
    parameter ADDRESS_WIDTH = 16,
    parameter DATA_WIDTH    = 8,
    parameter MEMORY_DEPTH  = 2**ADDRESS_WIDTH
)(
    input wire ReadClock,                                   // 1
    input wire WriteClock,                                  // 2
    input wire Reset,
    input wire ReadEnable_i,
    input wire WriteEnable_i,
    input wire [ADDRESS_WIDTH-1:0] ReadAddress_i,           // 3
    input wire [ADDRESS_WIDTH-1:0] WriteAddress_i,          // 4
    input wire [   DATA_WIDTH-1:0] Data_i,
    output reg [   DATA_WIDTH-1:0] Data_o
);

    // Deklaracja pamięci
    reg [DATA_WIDTH-1:0] Memory [0:2**ADDRESS_WIDTH-1];

    // Kontrola czy pamięć mieści się w przestrzeni adresowej
    initial begin
        if(MEMORY_DEPTH > 2**ADDRESS_WIDTH)
            $fatal(0, "Required memory depth is larger than address space");
    end

    // Inicjalizacja pamięci zerami
    integer i;
    initial begin
        for(i=0; i<2**ADDRESS_WIDTH; i=i+1) begin
            Memory[i] = 0;
        end
    end

    // Operacja odczytu
    always @(posedge ReadClock, negedge Reset) begin    // 5
        if(!Reset)
            Data_o <= 0;
        else if(ReadEnable_i)
            Data_o <= Memory[ReadAddress_i];
    end

    // Operacja zapisu
    always @(posedge WriteClock) begin                  // 6
        if(WriteEnable_i)
            Memory[WriteAddress_i] <= Data_i;
    end

endmodule
`default_nettype wire
```

Listing 11. Kod pliku ram_pdp_tb.v

```
// Plik ram_pdp_tb.v

`timescale 1ns/1ns
`default_nettype none
module PseudoDualPortRAM_tb();

    parameter CLOCK_READ_HZ             = 10_000_000;     // 1
```

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie https://ulubionykiosk.pl/media

eprasa.pl 1fc8f31a49

W taki sposób powstają dwa niezależne od siebie sygnały zegarowe **ReadClock** oraz **WriteClock**. Następnie tworzymy różne zmienne, które wykorzystywane będą w sekwencji testowej. W szczególności uwagę należy zwrócić na osobne zmienne sterujące wejściami adresowymi **ReadAddress** (linia 5) oraz **WriteAddress** (linia 6). W linii 7 i 8 tworzymy zmienne **r** i **w** typu integer. Są to iteratory pętli for, odpowiednio do odczytu i zapisu. W poprzednich testbenchach używaliśmy do obu operacji zmiennej o nazwie **i**, ponieważ tak się przyjęło. W tym testbenchu będziemy jednocześnie zapisywać i odczytywać, a więc potrzebujemy dwóch różnych zmiennych sterujących pętlami for.

W liniach 9 oraz 11 rozpoczynamy dwie sekwencje testowe, które wykonują się równolegle, w tym samym czasie. W linii 9 rozpoczyna się blok always podobny do tego, jaki wykorzystywaliśmy do testowania zwykłej pamięci RAM. W linii 10 mamy pętlę for, która zapełnia pamięć losowo wybranymi danymi.

Równolegle wykonuje się sekwencja z bloku **initial**, zaczynającego się w linii 11, której celem jest odczytanie danych z pamięci. W linii 12 mamy opóźnienie o kilka taktów zegarowych. Celem tego jest zawieszenie wykonywania tego bloku do czasu, aż w pamięci zostanie zapisane pierwsze kilka danych. Potem będziemy jednocześnie zapisywać kolejne dane i odczytywać te, które zostały zapisane chwilę wcześniej. Odczytywania dokonujemy w pętli for w taki sam sposób, jak w poprzednich testbenchach.

Uruchamiamy symulację plikiem ram_pdp.bat, którego kod przedstawiono na **listingu 12**, a plik ram_pdp.vcd, powstający po zakończeniu symulacji, otwieramy przeglądarką GTKWave.

Przebiegi sygnałów uzyskane podczas symulacji przedstawiono na **rysunku 5**.

Listing 11. Kod pliku ram_pdp_tb.v – cd.

```
    parameter real HALF_PERIOD_READ_NS  = 1_000_000_000.0 / (2 * CLOCK_READ_HZ);

    parameter CLOCK_WRITE_HZ            = 15_000_000;     // 2
    parameter real HALF_PERIOD_WRITE_NS = 1_000_000_000.0 / (2 * CLOCK_WRITE_HZ);

    // Generator sygnału zegarowego dla odczytu
    reg ReadClock = 1'b1;                                 // 3
    always begin
        #HALF_PERIOD_READ_NS;
        ReadClock = !ReadClock;
    end

    // Generator sygnału zegarowego dla zapisu
    reg WriteClock = 1'b1;                                // 4
    always begin
        #HALF_PERIOD_WRITE_NS;
        WriteClock = !WriteClock;
    end

    // Zmienne
    reg        Reset       = 1'b0;
    reg        ReadEnable  = 1'b0;
    reg        WriteEnable = 1'b0;
    reg  [3:0] ReadAddress;                               // 5
    reg  [3:0] WriteAddress;                              // 6
    reg  [7:0] DataIn;
    wire [7:0] DataOut;
    integer    r;                                         // 7
    integer    w;                                         // 8

    // Instancja testowanego modułu
    PseudoDualPortRAM #(
        .ADDRESS_WIDTH(4),
        .DATA_WIDTH(8)
    ) DUT(
        .ReadClock(ReadClock),
        .WriteClock(WriteClock),
        .Reset(Reset),
        .ReadEnable_i(ReadEnable),
        .WriteEnable_i(WriteEnable),
        .ReadAddress_i(ReadAddress),
        .WriteAddress_i(WriteAddress),
        .Data_i(DataIn),
        .Data_o(DataOut)
    );

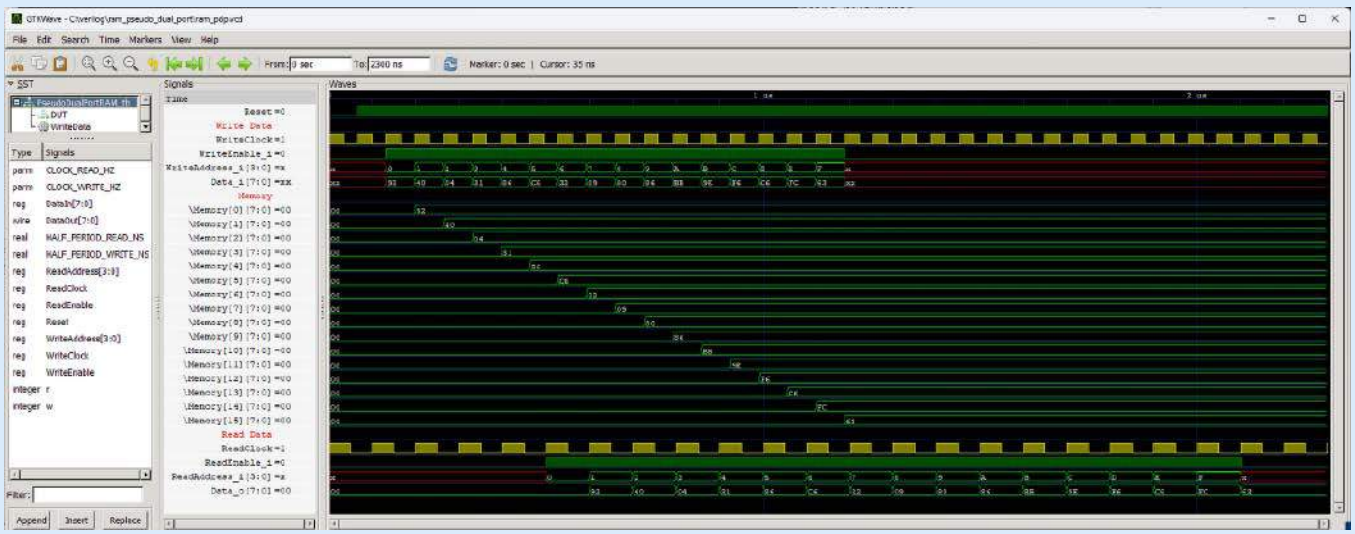
    // Eksport wyników symulacji
    initial begin
        $dumpfile("ram_pdp.vcd");
        $dumpvars(0, PseudoDualPortRAM_tb);

        // Eksportuj także wszystkie dane z pamięci
        for(r=0; r<=15; r=r+1) begin
            $dumpvars(2, DUT.Memory[r]);
        end
    end

    // Sekwencja testowa zapisu
    initial begin                                         // 9
        $timeformat(-6, 3, "us", 12);
        $display("===== START =====");
        $display("        Time WriteAddress DataIn ReadAddress DataOut");
        $monitor("%t          %H     %H          %H      %H",
            $realtime,
            WriteAddress,
            DataIn,
            ReadAddress,
            DataOut
        );

        @(posedge WriteClock);
        Reset <= 1'b1;
        @(posedge WriteClock);

        // Zapis losowych danych do pamięci
```



Rysunek 5. Przebiegi sygnałów podczas symulacji pseudodwuportowej pamięci RAM

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie
https://ulubionykiosk.pl/media

eprasa.pl 1fc8f31a49

Dla zwiększenia czytelności sygnały zegarowe zaznaczono żółtym kolorem. Porównaj ten screenshot z rysunkiem 4. W tym przypadku widzimy wyraźnie, że operacja zapisu i odczytu wykonywana jest równolegle.

Moduł top

Czas na ćwiczenie praktyczne, aby przetestować, jak działa zwykła pamięć RAM o organizacji 256x8bit, czyli komórki pamięci będą miały adresy od 0x00 do 0xFF i będzie można w nich przechowywać wartości również w takim zakresie. Zobacz **rysunek 6**. Zrobimy prostą aplikację, gdzie za pomocą jednego enkodera będziemy wybierać interesujący nas adres pamięci. Będzie on wyświetlany na dwóch cyfrach wyświetlacza w formacie szesnastkowym – adres na wyświetlaczu zaznaczono kolorem niebieskim. Natychmiast po zmianie adresu będzie wyświetlana zawartość wybranej komórki pamięci na dwóch cyfrach zaznaczonych kolorem zielonym.

Będziemy mogli zapisać do pamięci jakąś wartość pod aktualnie wskazywany adres. W tym celu służy drugi enkoder, którym będziemy mogli modyfikować wartość na wyświetlaczu zaznaczoną na żółto. Wartość ta zostanie zapisana do pamięci dopiero wtedy, kiedy zostanie wciśnięty enkoder wybierający adres. Po zapisaniu zielone cyfry wyświetlacza powinny natychmiast pokazać nową wartość odczytaną z pamięci.

Utwórz nowy projekt w Diamond i dodaj do niego pliki, które widoczne są na **rysunku 7**. Są to moduły, które omawialiśmy już w poprzednich odcinkach. Możesz je pobrać pod adresami widocznymi w ramce.

Kod pliku *top.v* pokazano na **listingu 13**. Na liście portów modułu top znajdują się tylko sygnały związane z enkoderem, wyświetlaczem i sygnał resetujący. Logika odpowiedzialna za inkrementację i dekrementację adresu oraz danych do zapisu jest bardzo podobna, więc z tego powodu omówimy tylko obsługę jednego enkodera. W linii 4 widzimy instancję modułu obsługującego enkoder obrotowy, który opracowaliśmy w 14 odcinku kursu. Jego wyjścia **Increment_o**, **Decrement_o** i **ButtonPress_o** (linie 5, 6, 7) informują o wykryciu obrotu gałki enkodera i wciśnięcia jej poprzez ustawienie stanu wysokiego na czas jednego cyklu zegarowego. Do tych wyjść doprowadzono sygnały wire utworzone w liniach 1, 2 i 3.

Sygnały te sprawdzane są w bloku always, który rozpoczyna się w linii 9. Jeżeli rozpoznano obrót pokrętła enkodera, to jest zwiększany lub zmniejszany stan rejestru **Address**, który utworzono w linii 8.

W linii 11 tworzymy instancję pamięci RAM. Konfigurujemy szerokość wejścia adresowego oraz wejścia danych na 8 bitów. Bajt wskazany przez wejście adresowe ma być odczytywany w każdych warunkach, więc dlatego na wejściu **ReadEnable_i** ustawiony jest na sztywno stan wysoki (linia 12). Zapis do pamięci odbywa się wtedy, kiedy doprowadzony do wejścia **WriteEnable_i** sygnał

Listing 11. Kod pliku ram_pdp_tb.v – cd.

```
        for(w=0; w<=15; w=w+1) begin                    // 10
            WriteData(w, $urandom_range(8'h00, 8'hFF));
        end
        WriteEnd();
    end

    // Sekwencja testowa odczytu
    initial begin                                       // 11
        repeat(5) @(posedge ReadClock);                 // 12

        // Odczytaj wszystkie dane z pamięci
        ReadEnable <= 1'b1;
        for(r=0; r<=15; r=r+1) begin                    // 13
            ReadAddress <= r;
            @(posedge ReadClock);
        end

        ReadAddress <= 4'dX;
        ReadEnable  <= 1'b0;

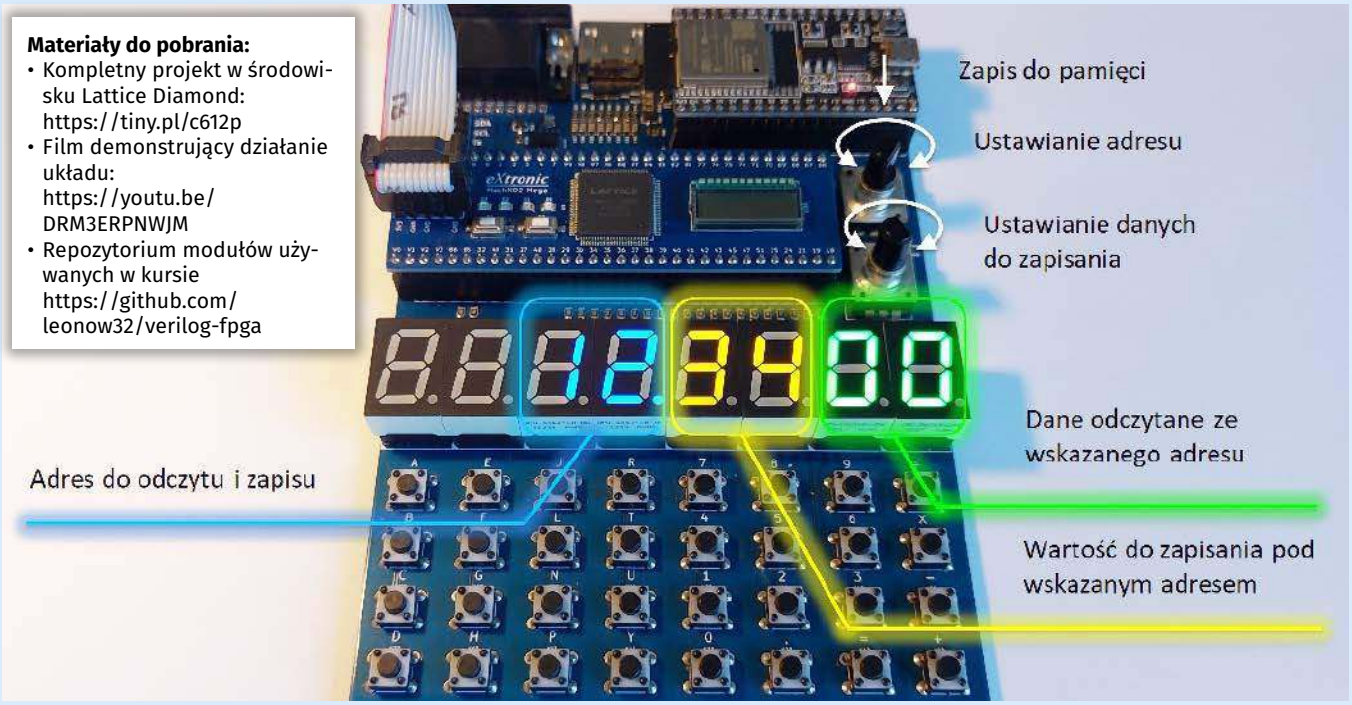
        // Pauza
        repeat(2) @(posedge ReadClock);

        $display("===== END =====");
        $finish;
    end

    // Task zapisujący dane do pamięci
    task WriteData(input [3:0] Adr, input [7:0] Dat);
        begin
            WriteAddress <= Adr;
            DataIn       <= Dat;
            WriteEnable  <= 1'b1;
            @(posedge WriteClock);
        end
    endtask

    // Task kończący zapis danych
    task WriteEnd();
        begin
            WriteAddress <= 4'dX;
            DataIn       <= 8'dX;
            WriteEnable  <= 1'b0;
            @(posedge WriteClock);
        end
    endtask

endmodule
`default_nettype wire
```



Materiały do pobrania:
- Kompletny projekt w środowisku Lattice Diamond: https://tiny.pl/c612p
- Film demonstrujący działanie układu: https://youtu.be/DRM3ERPNWJM
- Repozytorium modułów używanych w kursie https://github.com/leonow32/verilog-fpga

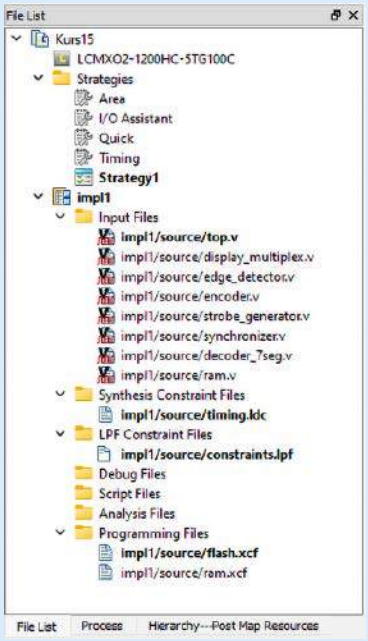
Rysunek 6. Opis działania enkoderów i wyświetlacza w testowej aplikacji

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie
https://ulubionykiosk.pl/media

eprasa.pl 1fc8f31a49

Listing 12. Kod pliku ram_pdp.bat

```
@echo off
iverilog -o ram_pdp.o ram_pdp.v ram_pdp_tb.v
vvp ram_pdp.o
del ram_pdp.o
```



Rysunek 7. Drzewko projektowe i lista plików wykorzystywanych w projekcie

WriteExecute jest w stanie wysokim – ten sygnał pochodzi z wyjścia **ButtonPress_o** sterownika enkodera (linia 7).

Dane odczytane z pamięci dostępne są na jej wyjściu **Data_o** (linia 16) i przekazywane są do sterownika wyświetlacza za pomocą sygnału wire **DataReadFromMemory** (linia 10).

Instancję sterownika 8-cyfrowego wyświetlacza 7-segmentowego LED widzimy w linii 17. Najbardziej nas interesuje jego wejście **Data_i**, które jest wejściem 32-bitowym. Każdy z czterech bitów tego wejścia steruje pojedynczą cyfrą wyświetlacza, która może wyświetlać cyfry od 0 do 9 i znaki

Listing 13. Kod pliku top.v

```
/ Plik top.v

`default_nettype none
module top(
    input wire Reset,                                          // Pin 17
    input wire Encoder1A_i,                                    // Pin 68
    input wire Encoder1B_i,                                    // Pin 67
    input wire Encoder1S_i,                                    // Pin 66
    input wire Encoder2A_i,                                    // Pin 71
    input wire Encoder2B_i,                                    // Pin 70
    output wire [7:0] Cathodes_o,
    output wire [7:0] Segments_o
);

    // Generator sygnału zegarowego
    parameter CLOCK_HZ = 14_000_000;
    wire Clock;
    OSCH #(
        .NOM_FREQ("14.00")
    ) OSCH_inst(
        .STDBY(1'b0),
        .OSC(Clock),
        .SEDSTDBY()
    );

    // Enkoder 1 do ustawiania adresu
    wire AddressIncrement;                                     // 1
    wire AddressDecrement;                                     // 2
    wire WriteExecute;                                         // 3

    Encoder Encoder1(                                          // 4
        .Clock(Clock),
        .Reset(Reset),
        .AsyncA_i(Encoder1A_i),
        .AsyncB_i(Encoder1B_i),
        .AsyncS_i(Encoder1S_i),
        .Increment_o(AddressIncrement),                        // 5
        .Decrement_o(AddressDecrement),                        // 6
        .ButtonPress_o(WriteExecute),                          // 7
        .ButtonRelease_o(),
        .ButtonState_o()
    );

    // Enkoder 2 do ustawiania zapisywanego bajtu danych
    wire DataIncrement;
    wire DaraDecrement;

    Encoder Encoder2(
        .Clock(Clock),
        .Reset(Reset),
        .AsyncA_i(Encoder2A_i),
        .AsyncB_i(Encoder2B_i),
        .AsyncS_i(1'b0),
        .Increment_o(DataIncrement),
        .Decrement_o(DaraDecrement),
        .ButtonPress_o(),
        .ButtonRelease_o(),
        .ButtonState_o()
    );

    // Rejestr wskazujący aktualnie wybrany adres pamięci
    reg [7:0] Address;                                         // 8

    // Logika inkrementacji/dekrementacji wybranego adresu
    always @(posedge Clock, negedge Reset) begin               // 9
        if(!Reset)
```



Rysunek 8. Konfiguracja pinów w Spreadsheet

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie
https://ulubionykiosk.pl/media

eprasa.pl 1fc8f31a49

od A do F, aby wyświetlać dane w formacie szesnastkowym. W naszej testowej aplikacji mamy trzy zmienne 8-bitowe do wyświetlenia. Są to **Address**, **DataToWrite** i **DataReadFromMemory**. Te trzy zmienne zajmują 24 bity, więc pozostałe 8 bitów zerujemy na stałe przy pomocy wyrażenia 8'd0. Wszystkie te cztery 8-bitowe wyrażenia sklejamy w jedną 32-bitową całość za pomocą operatora konkatenacji {}.

Syntezujemy, po czym otwieramy narzędzie Spreadsheet i konfigurujemy piny układu FPGA tak, jak to pokazano na **rysunku 8**. Pozostaje już tylko wygenerować bitstream i wgrać go do FPGA. Miłej zabawy! Pamiętaj, że moduł wyświetlacza wygasza zera z lewej strony wyświetlacza, czyli adres 0x00 i dane do zapisu 0x00 będą niewidoczne.

W następnej części kursu dowiemy się, jak za pomocą FPGA można generować dźwięki, a potem użyjemy pamięci, aby wykonać odtwarzacz melodyjek.

Dominik Bieczyński
leonow32@gmail.com

Zobacz więcej:

1. Repozytorium modułów wykorzystywanych w kursie https://github.com/leonow32/verilog-fpga
2. MachXO2 Family Datasheet https://tiny.pl/cgwbb
3. Memory Usage Guide for MachXO2 Devices https://tiny.pl/c61gc
4. Lattice Synthesis Engine for Diamond User Guide https://tiny.pl/c61gd

Listing 13. Kod pliku top.v – cd.

```
            Address <= 0;
        else if(AddressIncrement)
            Address <= Address + 1'b1;
        else if(AddressDecrement)
            Address <= Address - 1'b1;
    end

    // Dane do zapisu
    reg [7:0] DataToWrite;

    // Logika inkrementacji/dekrementacji bajtu danych do zapisu
    always @(posedge Clock, negedge Reset) begin
        if(!Reset)
            DataToWrite <= 0;
        else if(DataIncrement)
            DataToWrite <= DataToWrite + 1'b1;
        else if(DaraDecrement)
            DataToWrite <= DataToWrite - 1'b1;
    end

    // Instancja pamięci RAM
    wire [7:0] DataReadFromMemory;                                  // 10

    RAM #(                                                          // 11
        .ADDRESS_WIDTH(8),
        .DATA_WIDTH(8)
    ) RAM_inst(
        .Clock(Clock),
        .Reset(Reset),
        .ReadEnable_i(1'b1),                                        // 12
        .WriteEnable_i(WriteExecute),                               // 13
        .Address_i(Address),                                        // 14
        .Data_i(DataToWrite),                                       // 15
        .Data_o(DataReadFromMemory)                                 // 16
    );

    // Instancja sterownika wyświetlacza
    DisplayMultiplex #(                                             // 17
        .CLOCK_HZ(CLOCK_HZ),
        .SWITCH_PERIOD_US(1000),
        .DIGITS(8)
    ) DisplayMultiplex_inst(
        .Clock(Clock),
        .Reset(Reset),
        .Data_i({8'd0, Address, DataToWrite, DataReadFromMemory}), // 18
        .DecimalPoints_i(8'b00010000),
        .Cathodes_o(Cathodes_o),
        .Segments_o(Segments_o),
        .SwitchCathode_o()
    );


endmodule

`default_nettype wire
```

REKLAMA

świat radio
Magazyn wszystkich użytkowników eteru
KRÓTKOFALARSTWO CB RADIOTECHNIKA

przejrzyj i kup na
www.ulubionykiosk.pl



eprasa.pl 1fc8f31a49

koktajl niusów



Krótka historia o tym, jak fotel dentystyczny uniemożliwiał otwieranie samochodów

Była to poważna sprawa, która rozpoczęła się zgłoszeniami od kilkunastu właścicieli lokali usługowych, zlokalizowanych na parterze budynku wielorodzinnego. Informowali o problemie z otwieraniem bądź zamykaniem samochodów zaparkowanych przed budynkiem za pomocą pilotów. Wysłany na miejsce zespół kontrolerów UKE sam odczuł uciążliwość sytuacji, nie mogąc zabezpieczyć auta, którym udano się w teren. Na szczęście mechanicznego zamka w drzwiach nic nie zakłócało. Dokonane za pomocą odbiornika monitoringowego pomiary jednoznacznie wykazały, że występuje nieprzerwana emisja w paśmie zarezerwowanym do sterowania drogą radiową urządzeniami powszechnego użytku. Stosując manualną antenę kierunkową, pracownicy trafili od razu pod drzwi przychodni stomatologicznej, a w jej wnętrzu pod jeden z gabinetów. Z uwagi na to, że na fotelu siedział pacjent, trzeba było poczekać z dalszymi pomiarami. Tuż przed wyjściem pacjenta sygnał zakłócający „przepadł". Wynikało z tego, że niepożądana emisja związana jest z działaniem fotela dentystycznego. Po tym jak kontrolerzy UKE weszli do środka, poprosili obsługę o uruchomienie kolejnych elementów wyposażenia stanowiska zabiegowego. Winny okazał się ssak odciągający ślinę z ust pacjenta. Po jego podniesieniu fotel zaczął „nadawanie". Po rozmowie z właścicielami przychodni ustalono, że w gabinetach niedawno został przeprowadzony remont. Przez niedopatrzenie firmy realizującej remont nie została położona instalacja do sterowania pompą podciśnieniową, która ma współpracować z ssawkami. Żeby nie niszczyć dopiero co położonych płytek zastosowano sterowanie zdalne z użyciem dzwonka bezprzewodowego. I tak oto przez cały czas przebywania pacjenta na fotelu nie było możliwe otwieranie samochodów zaparkowanych przed lokalami usługowymi.

https://tiny.pl/c61s5

Słońce pomoże w produkcji silników i napędów ABB

Jest większa niż boisko piłkarskie oraz zasila energią słoneczną fabryki wytwarzające urządzenia, które zmniejszają pobory mocy. Zaprezentowane zdjęcie powstało w Aleksandrowie Łódzkim, gdzie firma ABB wybudowała farmę fotowoltaiczną o mocy 2 MW. Wykonana przez ABB farma fotowoltaiczna zajmuje powierzchnię 10 000 m². Rozmieszczono w niej ponad 5 tysięcy modułów PV oraz kilkanaście inwerterów. Tak zrealizowana instalacja zasila dostępne w pobliżu dwa zakłady ABB: silników elektrycznych i napędów przemysłowych.



Farma fotowoltaiczna w Aleksandrowie Łódzkim została podłączona do systemu zarządzania energią ABB Ability Energy Management, który monitoruje ilość wytwarzanej energii z paneli fotowoltaicznych. Wkrótce rozwiązanie zyska całkiem nowe funkcjonalności, które pozwolą na szczegółowe określenie energochłonności poszczególnych linii produkcyjnych, w zależności od pory dnia, oraz wydajniejszą dystrybucję dopasowaną do aktualnych potrzeb.

Elektrownie słoneczne mają powstać także w innych lokalizacjach ABB w Polsce – w drugiej połowie 2024 roku powstanie farma fotowoltaiczna o mocy 1,6 MW w Przasnyszu, co sprawi, że tamtejsza fabryka aparatury czy rozdzielnic średniego napięcia będzie zasilana wyłącznie energią słoneczną. W planach jest również instalacja paneli przy siedzibie spółki w Warszawie. Jednocześnie firma rozpoznaje opcję budowy farmy słonecznej o mocy 2 MW, która zasiliłaby jej hub produkcyjno-inżynieryjny w Bielsku-Białej.

https://tiny.pl/c61sp



Zwiększone bezpieczeństwo kodu dzięki rodzinie mikrokontrolerów PIC18-Q24 Microchip Technology

Coraz więcej przedmiotów codziennego użytku zostaje podłączonych do Internetu – od telefonów komórkowych i pojazdów, po nowoczesne termostaty i urządzenia domowe. Żeby skutecznie bronić się przed ciągle rosnącymi zagrożeniami bezpieczeństwa, firma Microchip Technology wprowadziła na rynek nową serię mikrokontrolerów PIC18-Q24 z wbudowaną funkcją Programming and Debugging Interface Disable (PDID). Po aktywacji funkcji Programming and Debugging Interface Disable (PDID), zapewnia się nadzwyczaj efektywną ochronę kodu w celu zablokowania prób dostępu do interfejsu

eprasa.pl 1fc8f31a49

programowania/debugowania i eliminacji nieautoryzowanych prób odczytu, modyfikacji bądź kasowania oprogramowania układowego.

Ponieważ wiele rozwiązań często łączy się albo też komunikuje z dość szeroką gamą czujników, układów pamięci i procesorów, mikrokontrolery PIC18-Q24 obejmują wielonapięciowe wejścia/wyjścia (MVIO). Pozwalają one wyeliminować konwertery napięcia, umożliwiając dołączanie się do szeregu cyfrowych wejść/wyjść (I/O) – przy różnych napięciach. Oprócz zmniejszenia złożoności rozwiązań mikrokontrolery PIC18-Q24 idealnie sprawdzają się jako układy do zarządzania systemami. Występuje w nich bootloader dla aplikacji, które wymagają bezpiecznego sposobu aktualizacji oprogramowania układowego. Jak wyjaśnia wiceprezes ds. korporacyjnych w dziale mikrokontrolerów ośmiobitowych firmy Microchip Technology, Greg Robinson: „Bezpieczeństwo systemu jest ważne tak samo jak jego najsłabsze ogniwo. Każdy element, w praktyce dający się skonfigurować, może być podatny na ataki i dlatego konieczne jest pełne, wręcz całościowe, wdrożenie ulepszonych funkcji ochrony, żeby zapobiec włamaniom. Seria mikrokontrolerów PIC18-Q24 powstała z myślą o zaawansowanych zabezpieczeniach, żeby w całości pomóc klientom w walce z zagrożeniami”.

https://tiny.pl/c61ss



QN800C firmy Samsung – telewizor linii premium, który wykracza ponad przeciętność

Telewizor QN800C wyróżnia zaawansowana technologia wyświetlania Quantum Mini LED, dzięki której nie można przegapić żadnego, istotnego momentu wydarzenia. Dzięki głębokiej czerni i żywym światłom wszystko widać w połączeniu ze spektakularnym i niespotykanym wcześniej kontrastem. Dostępny w telewizorze procesor neuronowy AI Quantum 8K automatycznie polepsza jakość scen, niezależnie od oryginalnej zawartości. Dzięki temu widzowie mogą w zupełności doświadczać rozdzielczości możliwie jak najbardziej zbliżonej do 8K. Prócz dobrej jakości obrazu telewizor ten odznacza się również niesłychanym brzmieniem. Technologia Dolby Atmos współpracuje z 8 zintegrowanymi głośnikami, tworząc realistyczne i wielowymiarowe odczucie dźwięku. Jak przystało na przedstawiciela najbardziej luksusowej linii telewizorów QN800C z pewnością odznacza się wybornym wyglądem. Jego cechą charakterystyczną jest absolutnie bezgraniczny design, który pozwala jeszcze bardziej zanurzyć się w sportowym świecie albo przenieść wrażenia z oglądania na kompletnie nowy poziom. Dodatkowo moduł One Connect Pro pozwala podłączyć do telewizora opcjonalne urządzenia. Dzięki temu można pousuwać nieestetyczną plątaninę kabli. Telewizor QN800C występuje w 3 wariantach różniących się przekątną ekranu: 65”, 75” i 85”.

https://tiny.pl/c61sv

Większy komfort dzięki robotom usługowym CLOi ServeBot od LG Electronics

Przewiduje się, że w najbliższej przyszłości roboty usługowe będą pomagać ludziom z wszelakimi zadaniami: od produkcji i służby zdrowia, po budownictwo czy pomoc w domu. Roboty te powstają z myślą o wykonywaniu praktycznych zadań oraz przekazywaniu informacji.



Potrafią też zwiększyć komfort, wydajność, a nawet bezpieczeństwo korzystających z nich osób. Najlepszym tego przykładem jest CLOi ServeBot z firmy LG Electronics, który ma swoje zastosowanie m.in. w bibliotece publicznej w Changwon – Korea Południowa. Robot CLOi ServeBot z chęcią pomaga dzieciom i osobom z problemami z poruszaniem się w transporcie książek do dowolnego miejsca w bibliotece.

Użytkownicy mogą poprosić robota o wybraną pozycję na stanowisku wyszukiwania, po czym pracownik biblioteki wkłada książkę do szuflady robota LG CLOi ServeBot, który dostarcza zamówiony tytuł. Aby upewnić się, że książka trafi do odpowiedniej osoby, szuflada otwiera się dopiero po wpisaniu hasła, które utworzono na stanowisku wyszukiwania.

Dzięki temu, że roboty LG Electronics są w stanie korzystać z windy, mogą udawać się na wszystkie piętra biblioteki. Prócz tego robot CLOi ServeBot z półkami odciąża personel biblioteki w Changwon, przekazując oddane książki do zautomatyzowanych stanowisk zwrotu. Pracownicy biblioteki nie muszą już pchać i rozładowywać wózków z ciężkimi książkami, dzięki czemu mają więcej czasu, żeby pomagać osobom odwiedzającym bibliotekę, służąc im poradami lub odpowiadając na ich pytania.

https://tiny.pl/c616h



Udane testy procesorów Intel Xeon z technologią vRAN Boost za nami

Były to eksperymenty, które dowiodły wzrostu wydajności energetycznej, możliwości zarządzania większymi obciążeniami i wyższej przepustowości, w wirtualnej radiowej sieci dostępowej (virtual Radio Access Network, vRAN). Testów procesorów Intel Xeon dokonano w ramach sieci vRAN opracowanej przez firmę Samsung. Sprawdzone z sukcesem procesory Intel Xeon z technologią vRAN Boost wytworzono w celu efektywnego napędzania wysokowydajnych, energooszczędnych sieci vRAN. Dzięki pełnej integracji akceleracji vRAN, w procesorach Intel Xeon firma Intel pousuwała potrzebę korzystania z różnych, zewnętrznych rozwiązań.

Jest to pierwszy taki przypadek na rynku vRAN. Nowe procesory oferują kompleksowe funkcje, które optymalizują wydajność obciążeń roboczych, dynamiczne zarządzanie energią i akcelerację dla rozwiązań sztucznej inteligencji (AI) i uczenia maszynowego. Krytyczne funkcje dla zdezagregowanych rozwiązań vRAN, takie jak m.in. polepszone taktowanie i synchronizacja sieci, są możliwe dzięki mocy

eprasa.pl 1fc8f31a49

procesorów Intel Xeon – w połączeniu z kartami sieciowymi Intel Ethernet serii 800.

Jak przyznaje dyrektor generalny grupy Network and Edge Solutions Group w firmie Intel, Dan Rodriguez: „Ciągłe transformacje sieci 5G wspierane poprzez zróżnicowane innowacje projektowe, w tym w pełni zintegrowaną akcelerację vRAN, pokazują bezkompromisowe korzyści płynące z wirtualizacji na dużą skalę. Tym sposobem umożliwiamy swobodne korzystanie w całości z wysokowydajnych i energooszczędnych sieci vRAN, zapewniając lepsze doświadczenia, które związane są z siecią 5G".

https://tiny.pl/c61vs



Łatwość uczenia modeli LLM dzięki platformie NVIDIA NeMo

Jak sama nazwa modeli wskazuje – Large Language Models, są to rozwiązania, dla których potrzeba dużych zbiorów danych. Dla wielu podmiotów zajmujących się na dobrą sprawę generatywną sztuczną inteligencją jest to ogromne wyzwanie. Jednak w zupełności wystarczy zastosować opracowaną przez firmę NVIDIA platformę NeMo w celu sprostania owemu wyzwaniu. Jest to rozwiązanie umożliwiające tworzenie, a także, w praktyce, dostosowywanie oraz wdrażanie modeli LLM na ogromną skalę.

Jedną z firm, która zaufała platformie NVIDIA NeMo, jest firma Amazon, która rozwija modele fundamentalne Amazon Titan dla potrzeb Amazon Bedrock, tzn. usługi generatywnej sztucznej inteligencji. Zapewniane przez NVIDIA NeMo techniki zrównoleglania umożliwiają efektywne uczenie modeli LLM na niesłychaną skalę, a wraz z interfejsem Elastic Fabric Adapter proces uczenia realizowany jest na mnóstwie procesorów graficznych naraz (przynajmniej 10 tysiącach sztuk). Dało to niespotykaną dokładność działania modeli LLM, której nie da się osiągnąć, polegając wyłącznie na zrównolegleniu wykonywania operacji.

Jak wyjaśnia osoba pracująca dla firmy Amazon, Leonard Lausen: „Jednym z powodów, dla którego zdecydowaliśmy się na użycie platformy NVIDIA NeMo, jest jej skalowalność i dostępność optymalizacji, które umożliwiają pracę z wieloma procesorami, oczywiście graficznymi. W szczególności można ją dopasowywać do ogromnych klastrów przetwarzania danych, co sprawia, że można szybciej nauczać i dostarczać klientom modele LLM. Jest to o tyle łatwe, że NVIDIA NeMo bazuje na bardzo popularnych bibliotekach, np. PyTorch Lightning, które standaryzują proces uczenia modeli LLM".

https://blogs.nvidia.com/blog/nemo-amazon-titan/

Ricoh Pro Z75 – pierwszy na świecie system do druku arkuszowego B2 zawierający atrament wodny

System ten pozwala uzyskać wysoką jakość produkcji przy utrzymaniu dosyć szybkiej realizacji zadań i obniżeniu nakładu pracy ze strony operatora. Własności te sprawiają, że Ricoh Pro Z75 wspiera transformację z druku offsetowego na cyfrowy. Dostępne w systemie

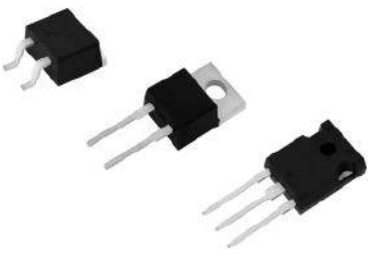


Ricoh Pro Z75 głowice piezoelektryczne ze stali nierdzewnej zapewniają rozdzielczość natywną druku do 1200 dpi – przy szybkościach pracy z czterokolorowymi atramentami pigmentowymi (CMYK). Gwarantowana jest prędkość druku jednostronnego do 4500 arkuszy na godzinę, a także do 2250 arkuszy na godzinę przy druku dwustronnym. Opatentowany system suszenia sprawia, że wydruki można poddawać bez problemu dalszemu wykończaniu.

Ricoh Pro Z75 obsługuje szeroką gamę podłoży – m.in. papiery offsetowe i nośniki o gramaturze do 400 g/m^2 oraz maksymalnym rozmiarze arkusza 58,5×75 cm. Trwała i przypominająca urządzenia offsetowe konstrukcja, jak też zaawansowane, wytrzymałe głowice drukujące zostały zaprojektowane tak, żeby maksymalizować czas pracy systemu. Dbałość o szczegóły – od wspomaganego powietrzem podawania arkuszy, które eliminuje problemy z transportem papieru, po konfigurowalną tacę odbiorczą o rewelacyjnej pojemności, która zapewnia równe układanie papieru, usprawnia cały proces produkcji. System Ricoh Pro Z75 jest zgodny z bardzo szeroką gamą oprogramowania wspierającego automatyzację produkcji np. Ricoh TotalFlow BatchBuilder i Ricoh Supervisor.

https://tiny.pl/c61v3

Węglikowo-krzemowe (SiC) diody MPS firmy Vishay Intertechnology dostępne w ofercie Rutronik



Diody Merged PIN Schottky'ego (MPS) wyróżnia niskie napięcie przewodzenia, a także małe pojemności. Działają one niezwykle niezawodnie i z powodzeniem przeszły trwający aż 2 tysiące godzin test HTBR (*Higher Temperature Reverse Bias*) oraz 2 tysiące cykli termicznych. Odpowiada to dwukrotnie wyższej liczbie godzin testów, niż wymaga tego standard AEC-Q101.

Wprowadzenie diod MPS firmy Vishay Intertechnology w stan przewodzenia wymaga napięcia poniżej 1,46 V, prąd wsteczny nie przekracza 1,3 µA, a ładunek pojemnościowy to zaledwie 12 nC. Pracują z napięciami nie większymi niż 650 V i oferują zgodność z normą J-STD-020. Predestynowane do pracy w szerokim przedziale częstotliwości diody MPS działają w bardzo szerokim zakresie temperatury. Są dostępne w 3 obudowach: D2PAK 2L, TO-220AC 2L i TO-247AD 3L. Zastosowania diod obejmują m.in.: przetwornice DC/DC, przetwornice rezonansowe LLC, zasilacze impulsowe, a także inwertery fotowoltaiczne i serwery.

https://tiny.pl/c61b5

Jakub Tyburski
jakub.tyburski@elportal.pl

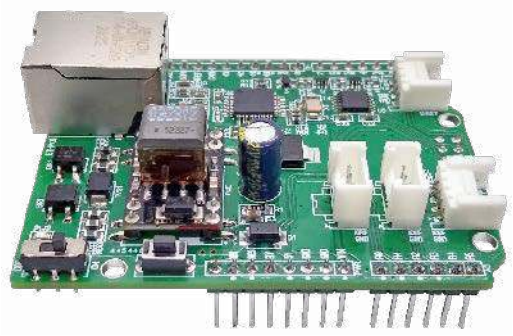
eprasa.pl 1fc8f31a49

Hity następnego numeru • „Elektronika Praktyczna” 2/2024

Bezprzewodowy włącznik dwukanałowy

Tradycyjne, elektromechaniczne włączniki sieciowe wymagają wpięcia bezpośrednio w obwód zasilający załączanego urządzenia. Ale nie zawsze jest to możliwe – zwłaszcza wtedy, gdy chcielibyśmy sterować odbiornikiem z innego pomieszczenia lub nawet budynku. Można wprawdzie pociągnąć w tym celu bardzo długie przewody, ale wiązałoby się to z przewiercaniem ścian i/lub stropów. A gdyby tak przekazać informację drogą radiową, tak jak ma to miejsce w nowoczesnych systemach automatyki domowej?

Problem zdalnego załączania odbiorników sieciowych rozwiązuje z powodzeniem przedstawiony przez nas projekt bezprzewodowego włącznika dwukanałowego. Prosta konstrukcja, bazująca na gotowych modułach radiowych, zapewnia niezawodne sterowanie o teoretycznym zasięgu w terenie otwartym nawet do 200 m.



AVTDuino UNO_R4_EthPoE – nakładka Ethernet + PoE dla Arduino

Pojawienie się na rynku płytki Arduino UNO R4 jest dobrym pretekstem do odświeżenia dostępnych nakładek, które albo nie do końca są zgodne z nową wersją, albo nie wykorzystują pełnych możliwości procesora zastosowanego w R4.

Pierwszą z przygotowanych nakładek jest przewodowy interfejs Ethernet, oparty na popularnym układzie WIZnet W5500 z dodaną funkcjonalnością zasilania PoE. Nakładka usprawni realizację systemów sterowania i automatyki domowej oraz innych projektów, korzystających z sieci Ehernet.



I²C_UDCM sterownik mikrosilników prądu stałego

Nie zawsze realizacja zadań robotyki DIY wymaga sterowania dużymi silnikami o poborze prądu na poziomie kilku amperów lub więcej – w wielu wypadkach wystarczy mikrosilnik prądu stałego z odpowiednim modułem sterującym. W artykule zaprezentowano taki właśnie kompaktowy moduł, bazujący zaledwie na dwóch układach scalonych: podwójnym mostkowym driverze silników A3901 o prądzie wyjściowym do 400 mA/kanał oraz generatorze PWM z interfejsem I²C typu PCA9632.

Temat numeru: Akumulatory i BMS

Współczesna elektronika w dużej mierze bazuje na akumulatorowych źródłach zasilania. Urządzenia przenośne i ubieralne, zdalne systemy sensoryczne IoT, roboty mobilne, czy wreszcie pojazdy elektryczne – wszędzie tam, gdzie źródłem energii jest akumulator, pojawia się potrzeba odpowiedniego zarządzania procesami ładowania oraz rozładowania, a niebagatelne znaczenie tak dla bezpieczeństwa, jak i użyteczności, mają funkcje precyzyjnego monitorowania stanu całego pakietu, a nawet poszczególnych jego ogniw. W lutowym numerze „Elektroniki Praktycznej” przyjrzymy się zagadnieniom, związanym właśnie z zarządzaniem akumulatorami – i to zarówno tymi maleńkimi, stosowanymi w urządzeniach przenośnych, jak i ogromnymi pakietami, zdolnymi do przechowywania energii na poziomie kilkudziesięciu kWh.

Elektronika w Praktyce: Przełączniki, przyciski i złącza

Konwencjonalne podzespoły elektromechaniczne są niezastąpione w wielu obszarach współczesnej techniki. O ile coraz liczniejsze urządzenia bazują już na nowoczesnych interfejsach dotykowych bądź nawet sterowaniu gestami, to w licznych zastosowaniach proste, niezawodne elementy stykowe wciąż stanowią klucz do bezpieczeństwa, niezawodności czy też użyteczności urządzeń i maszyn. Wiele dzieje się także w obszarze złączy – wraz z rosnącymi wymaganiami użytkowników zmieniają się także rozmiary, standardy oraz techniczne aspekty wykonania złączy zasilających, sygnałowych i komunikacyjnych. Warto zatem co pewien czas odświeżyć posiadane informacje i poznać aktualne trendy, widoczne w stale rozwijającym się rynku elementów elektromechanicznych.

Wykaz firm ogłaszających się w tym numerze „Elektroniki Praktycznej”

AKSOTRONIK 43
BORNICO 5
BTC KORPORACJA 23
COMPUTER CONTROLS 11
ELMAX 9
FERYSTER 15
HAMMOND 7
LASTENIC LASER 15
PCB WAY 59, 61
QWERTY 74, 75
SEMICON 79, 82

Miesięcznik **„Elektronika Praktyczna”** (12 numerów w roku) jest wydawany przez AVT-Korporacja Sp. z o.o. we współpracy z wieloma redakcjami zagranicznymi.

Wydawnictwo:
AVT-Korporacja Sp. z o.o.
03-197 Warszawa, ul. Leszczynowa 11
tel. 22 257 84 99, e-mail: avt@avt.pl

Wydawca:
Wiesław Marciniak

Adres redakcji:
03-197 Warszawa, ul. Leszczynowa 11
e-mail: redakcja@ep.com.pl, www.ep.com.pl

Redaktor Naczelny:
Przemysław Musz

Redaktor Programowy, Przewodniczący Rady Programowej:
Piotr Zbysiński

Menedżer Magazynu:
Katarzyna Gugała

Szef Pracowni Konstrukcyjnej:
Jakub Sobański

Zespół marketingu i reklamy:
Katarzyna Gugała,
Bożena Krzykawska,
Grzegorz Krzykawski,
Grzegorz Lalak

Stali współpracownicy:
Lucjan Bryndza, Nikodem Czechowski, Jarosław Doliński, Andrzej Gawryluk, Krzysztof Górski, Tomasz Jabłoński, Paweł Kowalczyk, Henryk Kowalski, Rafał Kozik, Michał Kurzela, Szymon Panecki, Damian Sosnowski, Ryszard Szymaniak, Adam Tatuś, Jakub Tyburski, Robert Wołgajew

Uwaga!
Kontakt z wymienionymi osobami jest możliwy via e-mail, według schematu: imię.nazwisko@ep.com.pl

DTP i okładka:
MAD Sp. z o.o.

Redakcja strony internetowej www.ep.com.pl
MAD Sp. z o.o.

Prenumerata w Wydawnictwie AVT
www.ulubionykiosk.pl lub tel. 22 257 84 22 (godz. 10.00–14.00)
e-mail: prenumerata@avt.pl

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

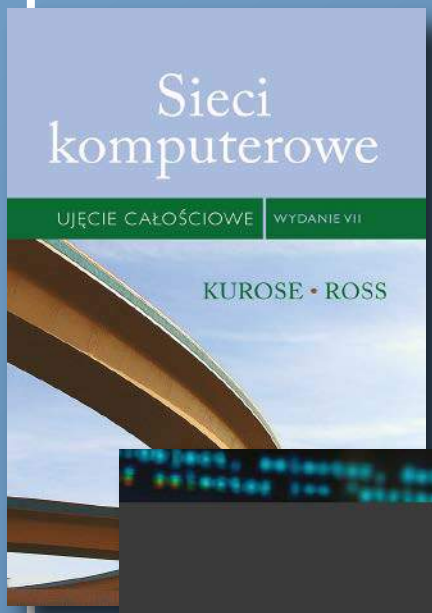
Wydawnictwo AVT-Korporacja Sp. z o.o. należy do **Izby Wydawców Prasy**

Copyright AVT-Korporacja Sp. z o.o.
03-197 Warszawa, ul. Leszczynowa 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 umieszczanie na stronach internetowych całości lub fragmentów publikacji zamieszczanych w „Elektronice Praktycznej”. jest dozwolone wyłącznie po uzyskaniu zgody redakcji. Redakcja nie odpowiada za treść reklam i ogłoszeń zamieszczanych w „Elektronice Praktycznej”.

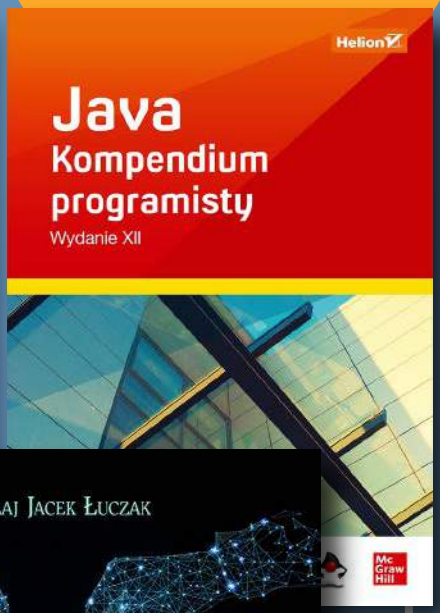
eprasa.pl 1fc8f31a49

Książki z rabatem
w Ulubionym Kiosku

do -30%











Zobacz pełną ofertę - **ponad 500 tytułów!**

www.UlubionyKiosk.pl

eprasa.pl 1fc8f31a49