



Programowalne sztuczne obciążenie DC

W dzisiejszych czasach nie jest wielkim problemem zakup dobrej jakości zasilacza czy innego źródła energii elektrycznej, na przykład akumulatora czy baterii akumulatorów. Niestety, nie wszystkie te źródła energii spełniają zalecane normy czy deklarowane parametry. Największym problemem jest jednak to, że parametry tych urządzeń znacząco pogarszają się z upływem czasu. Prezentowane dalej urządzenie pozwala zweryfikować kluczowe parametry źródeł energii elektrycznej prądu stałego w zakresach: napięcia znamionowego do 60 V, prądu obciążenia do 10 A oraz całkowitej oddawanej mocy do 200 W.

Rekomendacje: użyteczny przyrząd, który przyda się w warsztacie każdego elektronika.

Układ sztucznego obciążenia DC został podzielony na dwa bloki elektryczne: wykonawczy oraz sterujący. Pierwszy z nich realizuje funkcje sterowanego odbioru energii elektrycznej oraz pomiarów podstawowych parametrów roboczych (prąd i napięcie obciążenia oraz temperatura tranzystorów odbierających obciążenie). Drugi blok, oparty o mikrokontroler AVR ATmega16, jest odpowiedzialny za nastawienie parametrów roboczych oraz kontrolę pracy urządzenia a także za pomiary wartości kluczowych z punktu widzenia jakości pracy testowanego źródła energii, m.in.: pobranego ładunku Q oraz energii E . Taki podział projektu na dwa osobne moduły znacznie ułatwił poprawną realizację jego kluczowych założeń, w ramach których wzajemnie wykluczają się: odprowadzenie znacznej ilości energii cieplnej

oraz prawidłowe sterowanie pracą układu. W dalszej części artykułu przedstawiono zasadę działania oraz szczegóły montażu, uruchomienia i eksploatacji prezentowanego urządzenia.

Moduł wykonawczy

Na rysunku 1 pokazano schemat elektryczny bloku wykonawczego sztucznego obciążenia DC. Wprowadzenie źródła energii jest realizowane przez złącze P1, z którego jest ona podawana na wyprowadzenia 3 i 8 przełącznika K1. Dla zwiększenia bezpieczeństwa pracy urządzenia prąd roboczy *I_{real}* jest załączany przez dwa styki podwójnego przełącznika K1 (każdy z nich odpowiada za podłączenie jednego z doprowadzeń prądu roboczego). Podwójny przełącznik K1 jest załączany z modułu sterowania poprzez wejście P3. Aby

uniknąć wpływu zakłóceń elektromagnetycznych na stabilność sterowania, wprowadzony sygnał przełączający jest filtrowany dolnoprzepustowo poprzez dwójnik z elementami R3 i C6. Odfiltrowany w ten sposób sygnał sterujący jest podawany na bramkę tranzystora polowego Q1 (2N7000), którego zadaniem jest załączanie prądu sterowania cewek przełącznika K1. Dioda D1 (1N4148) zabezpiecza całe urządzenie przed skutkami przepięć, występujących w stanie odłączenia obciążenia.

Pomiar napięcia obciążenia jest realizowany przez blok ze wzmacniaczem operacyjnym U2A (MCP6004), zasilanym przez elementy R2, C4 i C5 w celu eliminacji zakłóceń elektromagnetycznych, pochodzących

REKLAMA

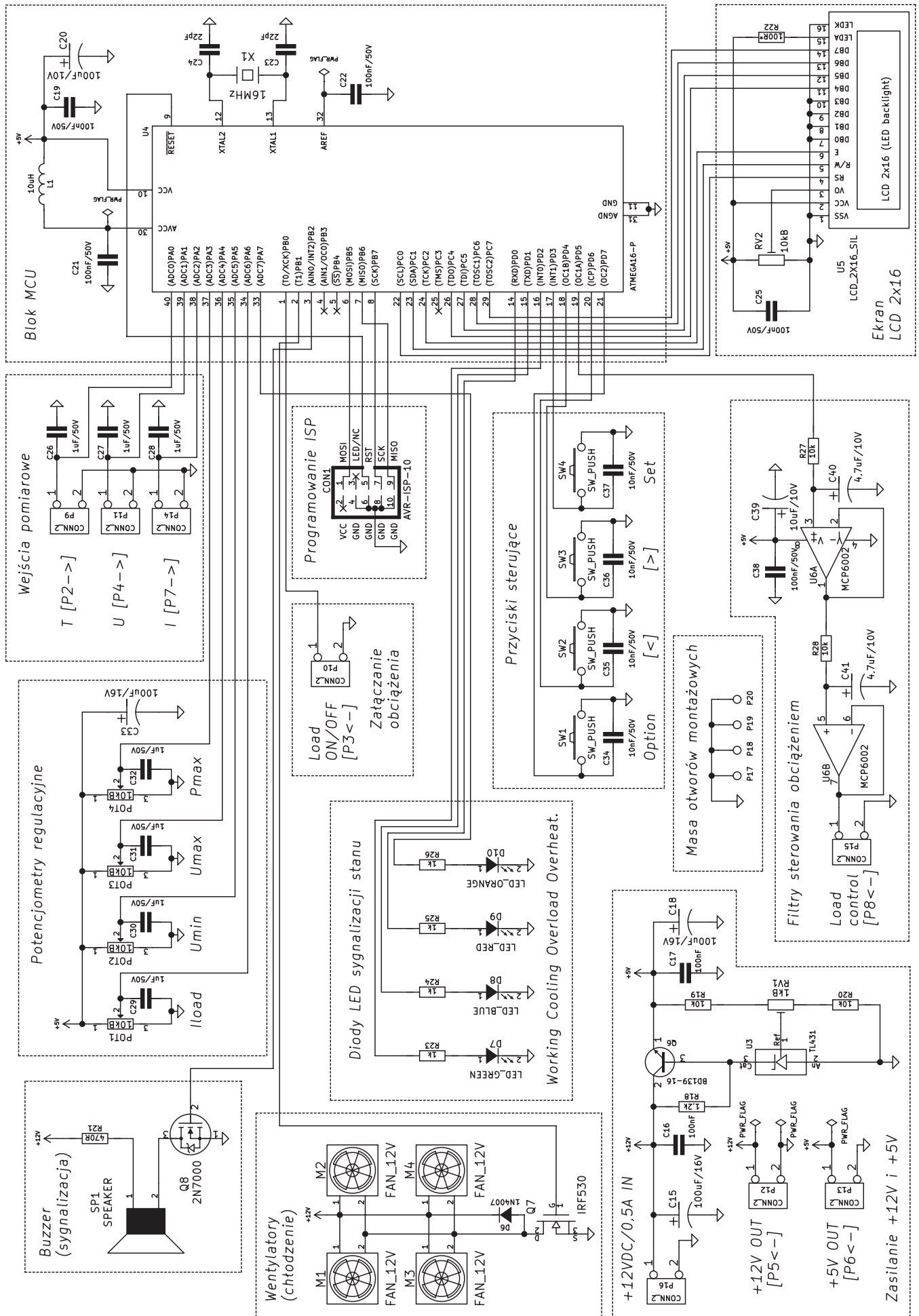
Specjalistyczne szkolenia dla elektroników i automatyków



TECHDAYS

techdays@techdays.pl
TECHDAYS.PL

CERTYFIKOWANY PARTNER SZKOLENIOWY
ST
life.augmented



Rysunek 1. Schemat ideowy modułu sterowania sztucznego obciążenia

od znacznych prądów obciążenia roboczego **Ireal**. Mierzone napięcie robocze **Ureal** jest podawane na dzielnik rezystancyjny z elementami R4 i R5, o współczynniku podziału **Ku** równym **0,0448902**. Wartości podanych elementów, o tolerancji równej 1%, są istotne dla dokładności pomiaru napięcia **Ureal**. Kondensator C7 blokuje zakłócenia w mierzonym napięciu, a dioda Zenera D2 ogranicza wartość przekazywanego dalej napięcia do poziomu bezpiecznego dla wejścia wzmacniacza operacyjnego, pracującego w trybie „reel-to-reel” z zasilaniem

równym +5 V. Wyjście zmniejszonego o podany wcześniej współczynnik **Ku** jest podawane na złącze P4.

Odbiór obciążenia jest realizowany poprzez podwójny układ z tranzystorami mocy Q4 i Q5 (2SC5200),ysterowanymi w układzie Darlingtona przez tranzystory Q2 i Q3 (BD911). Podzielenie na pół układu odbiornika energii ma na celu zmniejszenie do 100 W maksymalnej mocy wydzielanej w pojedynczym tranzystorze wykonawczym. Źródło obciążenia jest załączane poprzez tandem diod mocy typu Schottky D3 i D4 (MBR20200CT), których zadaniem jest ochrona układu przed skutkami podłączenia obciążenia z odwrotną polaryzacją. Wysterowanie par tranzystorów w układach Darlingtona: Q2-Q4 oraz Q3-Q5 jest realizowane z wyjść wzmacniaczy operacyjnych U2C i U2D (MCP6004), poprzez filtry dolno-przepustowe oparte o dwójniki R6-C8 oraz R7-C9. Pomiar prądu roboczego obciążenia

Ireal jest realizowany pośrednio poprzez pomiar napięć na rezystorach R10 i R11, pracujących w emiterach tranzystorów Q4 i Q5. Napięcia te są filtrowane poprzez dwójniki RC z elementami R8-C10 oraz R9-C11, a następnie sumowane i uśredniane za pomocą rezystorów: R12 i R13 na dalsze potrzeby pomiarowe. Należy podkreślić, że filtrujące dwójniki RC (R8, C10, R9, C11) zapobiegają oscylacjom w obwodach ujemnego sprzężenia zwrotnego wzmacniaczy operacyjnych U2C oraz U2D, sterujących parami tranzystorów w układzie Darlingtona: Q2-Q4 i Q3-Q5.

Sterowanie prądem obciążenia w module wykonawczym odbywa się poprzez złącze P8, za którym wprowadzony jest dzielnik rezystancyjny R15/R17 (oba oporniki o tolerancji 1%). Jego zadaniem jest realizacja podziału napięcia sterującego ze współczynnikiem **Ku=0,130435**. Kondensatory C13 i C14 mają za zadanie filtrować dolnoprzepustowo napięcie sterujące prądem obciążenia – głównie z uwagi na lokalne zakłócenia elektromagnetyczne, wywołane przepływem znacznych prądów obciążenia.

Pomiar prądu obciążenia **Ireal** jest realizowany w bloku ze wzmacniaczem operacyjnym U2B (MCP6004) o wzmocnieniu napięciowym **Ku=5,545455**, zapewnionym przez rezystory R14 i R16. Kondensator C12 zapewnia dodatkową filtrację mierzonego prądu, uzasadnioną wobec pracy bloku wykonawczego ze znacznymi prądami (być może także impulsowymi), natomiast dioda Zenera D5 chroni wzmacniacz operacyjny U2B przed negatywnymi konsekwencjami podania na jego wejście nadmiernej napięcia.

Z uwagi na możliwe znaczną moc odbieraną przez omawiane urządzenie, niezbędne są: pomiar oraz bieżąca kontrola temperatury **Treal** podzespołów odpowiedzialnych za odprowadzenie (w postaci ciepła) przyjętej energii. Zadanie to realizuje kalibrowany

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

W ofercie AVT* AVT-5586

Dodatkowe informacje:

Uwaga! Pełna treść listingów jest dostępna w materiałach dodatkowych do artykułu.

Podstawowe parametry:

- Aktywne obciążenie o mocy do 200 W (w prototypie – do 100 W).
- Złożony z dwóch płytek.
- Interfejs użytkownika: potencjometry, przyciski, moduł wyświetlacza LCD.
- Mikrokontroler ATmega16.
- Zasilanie +12 V DC/0,8 A.
- Oprogramowanie w języku AVR-GCC.

Projekty pokrewne na www.media.avt.pl:

AVT-5510	Sztuczne obciążenie (EP 8/2015)
AVT-1797	Sztuczne obciążenie wysokonapięciowe (EP 4/2014)
AVT-5318	Miernik mocy skutecznej wzmacniacza audio (EP 11/2011)
AVT-318	Obciążenie aktywne (EP 12/1996-1/1997)

Uwaga! Elektroniczne zestawy do samodzielnego montażu.

Wymagana umiejętność lutowania!
Podstawową wersją zestawu jest wersja [B] nazywana potocznie KITEM (z ang. zestaw). Zestaw w wersji [B] zawiera elementy elektroniczne (w tym [UK] – jeśli występuje w projekcie), które należy samodzielnie wzlutować 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 wzlutowane w płytkę PCB)
• wersja [A] płytką drukowaną bez elementów i dokumentacją Kity w których występuje układ scalony wymagający zaprogramowania, posiadają następujące dodatkowe wersje:
• wersja [A+] płytką drukowaną [A] + zaprogramowany układ [UK] i dokumentacją
• wersja [UK] zaprogramowany układ
Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik pdf! Podczas składania zamówienia upewnij się, którą wersję zamawiasz!
<http://sklep.avt.pl>, osoby zainteresowane zakupem płytek drukowanych (PCB), prosimy o kontakt via email: kity@avt.pl.

czujnik termiczny U1 (MCP9700), fizycznie zlokalizowany w bezpośrednim sąsiedztwie tranzystorów wykonawczych: Q4 i Q5. Zasilanie czujnika U1 jest filtrowane przez kondensatory C1 i C2, natomiast jego wyjście napięciowe filtrują elementy: R1 i C3. Napięcie wyjściowe **Ut** z czujnika temperatury, o wartości **Ut=500+10*Treal [mV]** jest wyprowadzone na złącze P2 modułu wykonawczego.

Zasilanie modułu wykonawczego jest podawane z modułu sterującego poprzez porty: P5 (+12 V DC) oraz P6 (+5 V DC). Napięcie +12 V DC pochodzi bezpośrednio z wejścia zasilania zewnętrznego (i nie jest dodatkowo stabilizowane), natomiast napięcie zasilające +5 V DC jest pobierane ze stabilizatora, zrealizowanego w ramach modułu sterującego sztucznym obciążeniem.

REKLAMA

Specjalistyczne szkolenia dla elektroników i automatyków



TECHDAYS

techdays@techdays.pl
TECHDAYS.PL

ST CERTYFIKOWANY PARTNER SZKOLENIOWY
life augmented

Wykaz elementów:

Moduł wykonawczy

Rezystory: (0,25W/5%)
R1: 2,2 kΩ
R2: 10 Ω
R3: 10 kΩ
R4: 100 kΩ/1%
R5: 4,7 kΩ/1%
R7, R6: 510 Ω
R9, R8: 33 Ω
R11, R10: 0,1 Ω/5 W
R15, R12, R13, R14: 10 kΩ/1%
R16: 2,2 kΩ/1%
R17: 1,5 kΩ/1%

Kondensatory:

C1, C4, C6, C8, C9, C14: 100 nF
C2: 10 μF/10 V
C5: 100 μF/10 V
C7, C10..C12: 10 nF
C3, C13: 47 μF/10 V

Półprzewodniki:

D1: 1N4148 (D0-35)
D2, D5: dioda Zenera 5,1 V (D0-35)
D3, D4: MBR20200CT (TO-220)
Q1: 2N7000 (TO-92)
Q3, Q2: BD911 (TO-220)
Q5, Q4: 2SC5200 (TO-264)
U1: MCP9700AE (TO-92)
U2: MCP6004 (DIP14 + podstawa)

Inne:

K1: HFKA-012-2ZST (przełącznik)
M1..M4: wentylatory RDH6025S (12 V/0,14 A)
SP1: buzzer 5 V
Obudowa „KRADEX” Z15
Radiator żeberkowy 165 mm×70 mm×35 mm
P2..P6: goldpin

Moduł sterujący

Rezystory: (0,25W/5%)
R18: 1,2 kΩ
R19, R20, R27, R28: 10 kΩ
R21: 470 Ω
R22*: 100 Ω
R23..R26: 1 kΩ
POT1..POT4: 10 kΩ/B (liniowy)
RV1: 1 kΩ (montażowy RM-063)
RV2: 10 kΩ (montażowy RM-063)

Kondensatory:

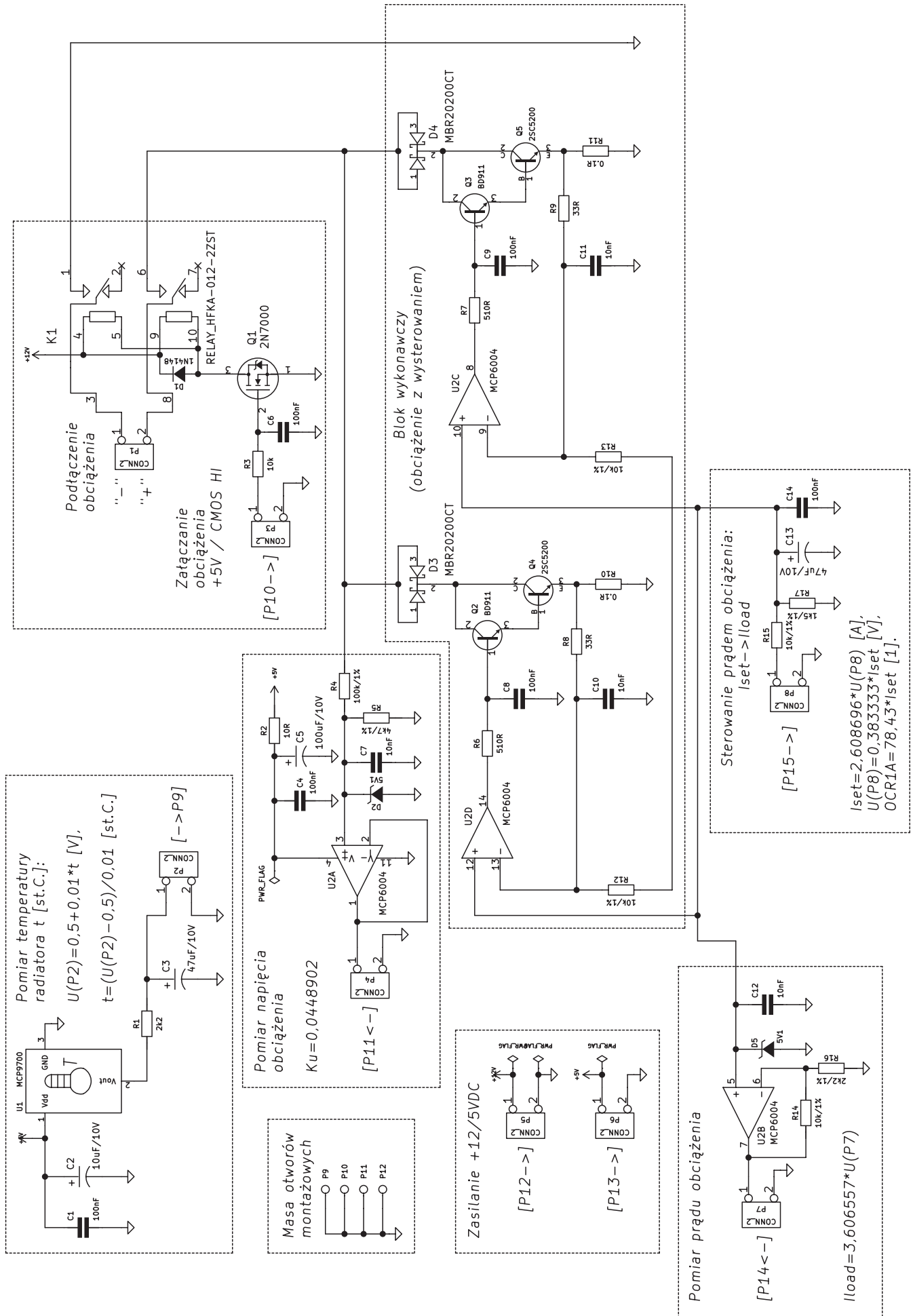
C15, C18, C33: 100 μF/16 V
C16, C17: 100 nF
C20: 100 μF/10 V
C19, C21, C22, C25, C38: 100 nF/50 V
C23..C24: 22 pF
C26..C32: 1 μF/50 V
C34..C37: 10 nF/50 V
C39: 10 μF/10 V
C40, C41: 4,7 μF/10 V

Półprzewodniki:

D6: 1N4007 (obudowa D0-41)
D7..D10: LED fi=3,0 mm (zielona, niebieska, czerwona i żółta)
Q6: BD139-16 (TO-126)
Q7: IRF530 (TO-220)
Q8: 2N7000 (TO-92)
U3: TL431 (TO-92)
U4: ATmega16 (DIP-40)
U6: MCP6002 (obudowa DIP-8)

Inne:

U5: LCD 2×16 znaków
L1: 10 μH (dławik osiowy)
CON1: 10 pinów, męskie (AVR ISP)
SP1: buzzer 5 V
X1: 16 MHz (HC49S)
SW1..SW4: przyciski (6×6×19) mm
P9..P16, M1..M4: męskie złącza goldpin



Rysunek 2. Schemat ideowy modułu wykonawczego sztucznego obciążenia

Modułu sterowania

Moduł sterowania, którego schemat przedstawiono na **rysunku 2**, oparty jest o mikrokontroler AVR ATmega16, zasilany poprzez pojemności filtrujące C19 i C20. Elementy C21 oraz L1 filtrują zasilanie bloku przetwornika analogowo-cyfrowego ADC, natomiast kondensator C22 blokuje dla zakłóceń napięcia referencyjne tego przetwornika. Kondensatory: C23 i C24 oraz rezonator X1 realizują bierny blok taktowania mikrokontrolera. Porty mikrokontrolera U1: 6, 7, 8 i 9 są podłączone do dedykowanego złącza programatora ISP, pozwalającego na zmianę ustawień sztucznego obciążenia DC.

Wejściowe porty mikrokontrolera U4: PA.0..PA.2 (piny: 38..40 i porty: P9, P11 oraz P14) wprowadzają sygnały pomiarowe: temperatury **Treal**, napięcia **Ureal** oraz prądu **Ireal**, pochodzących z modułu wykonawczego obciążenia. Z uwagi na możliwe zakłócenia impulsowe w pracy urządzenia, być może wygenerowane na dłuższych przewodach, łączących moduł wykonawczy z modułem sterującym, wejścia te zostały zablokowane dla takich zakłóceń kondensatorami: C26..C28.

Nastawy parametrów sterujących pracą urządzenia są wykonywane za pomocą potencjometrów POT1..POT4,ysterowanych z dokładnie wyregulowanego napięcia zasilającego +5 V DC, filtrowanego lokalnie przez kondensator C33. Nastawione wartości odczytywanych napięć są filtrowane przez pojemności: C29..C32 i podawane dalej na porty ADC6..ADC3 (porty: PA.6..PA.3 i jednocześnie piny 34..37 mikrokontrolera U4).

Załączanie obciążenia poprzez styki przekaźnika K1 w module wykonawczym jest realizowane poprzez podanie wysokiego poziomu logicznego (+5 V) na złącze P10. Natomiast za sygnalizację stanu pracy urządzenia odpowiadają diody LED: D7 (zielona, stan „**WORKING**”), D8 (niebieska, stan „**COOLING**”), D9 (czerwona, stan „**OVERLOAD**”), D10 (pomarańczowa, stan „**OVERHEATING**”). W przypadku wystąpienia jednego ze zdarzeń krytycznych, do których zaliczane są stany: „**OVERLOAD**” oraz „**OVERHEATING**”, jeżeli ustawiono na to zgodę w konfiguracji urządzenia, to załączana jest czasowa sygnalizacja buzzerem SP1. Sygnalizator dźwiękowy jest załączany wysokim poziomem logicznym (+5 V), podawanym z portu PB.2 (pin 3 mikrokontrolera U4) na bramkę tranzystora N-MOS Q8 (2N7000). Z uwagi na wyższe od nominalnego (+5 V) napięcie zasilania buzzera (+12 V) jest onysterowany poprzez rezystor szeregowy R21 (470 W). Chłodzenie radiatora, odbierające energię termiczną, jest realizowane przez maksymalnie cztery równoległe połączone wentylatory, zasilane napięciem +12 V każdy. Są one załączane w stanie „**COOLING**”,

za pomocą tranzystora mocy N-MOS Q7 (IRF530). Ilość wykorzystywanych wentylatorów M1..M4 zależy od ich wymiarów oraz od maksymalnej mocy **Preal**, odprowadzanej przez omawiane sztuczne obciążenie. Dioda D6 (1N4007) ma za zadanie chronić tranzystor Q7, załączający wentylatory, przed konsekwencjami przepięć pojawiających się w trakcie załączania i wyłączania wentylatorów. Przyciski typu „*micro-switch*”: SW1, SW2, SW3 oraz SW4, sterujące ustawieniami urządzenia, są wprowadzone na porty: PD.7 (pin 21), PD.6 (pin 20), PD.4 (pin 18) oraz PD.3 (pin 17) mikrokontrolera U4 i zablokowane do masy dla zakłóceń impulsowych za pomocą pojemności: C34..C37 (10 nF). Wartości nastawionych oraz pomierzonych parametrów są wyświetlane na ekranie alfanumerycznym LCD (U5) o wymiarze 16×2 znaki, którego zasilanie jest blokowane od zakłóceń kondensatorem C25 (100 nF). Liniowy potencjometr montażowy RV2 (10 kW) służy do ustawienia kontrastu wyświetlacza, natomiast rezystor R22 (w egzemplarzu modelowym o wartości 100 W) ogranicza prąd podświetlania ekranu LCD. Wartość tego rezystora może zostać zmieniona w zależności od typu użytego wyświetlacza LCD, a w przypadku ekranu z wbudowanym źródłem prądowym, ustalającym prąd podświetlających diod LED, można go zastąpić zworą.

Zasilanie całego modułu sterowania (+12 V DC/0,5 A) jest wprowadzane na złącze P16. Pojemności: C15 i C16 (100 μF i 100 nF) blokują potencjalne zakłócenia impulsowe i tak odfiltrowane napięcie jest podawane dalej na wyprowadzenie P12, którego zadaniem jest przekazanie zasilania obwodów sterujących w module wykonawczym. Zasilanie +5 V DC jest uzyskiwane z wykorzystaniem wysoko stabilnego temperaturowo źródła napięcia referencyjnego U3 (TL431). Precyzyjną wartość tego napięcia ustalają: rezystory R19 i R20 oraz potencjometr regulacyjny RV1, (liniowy, 1 kW), z którego odczepu pobierane jest napięcie sprzężenia zwrotnego, podawane na wyprowadzenie 1 układu U3. Układ ten współpracuje z tranzystorem wykonawczym NPN Q6 (BD139-16), którego zadaniem jest zapewnienie odpowiednio dużej wydajności prądowej omawianego stabilizatora. Bazę tego tranzystora polaryzuje rezystor R18, natomiast układ U3 pełni rolę „*regulowanej diody Zenera*”, która pobiera nadmiar prądu z rezystora R18 tak, by napięcie sprzężenia zwrotnego na odczepie potencjometru RV1 wynosiło 2,495 V DC (wartość katalogowa). Wyjście omawianego stabilizatora (emiter tranzystora Q6) jest dodatkowo blokowane dla zakłóceń impulsowych pojemnościami: C17 (100 nF) i C18 (100 μF) a także wyprowadzone na złącze P13 – celem dalszego wprowadzenia do modułu wykonawczego przez złącze P6.

Wysterowanie napięciowe tranzystorów odbierających moc strat termicznych, zlokalizowanych w module wykonawczym sztucznego obciążenia DC, jest realizowane przez 10-bitowy przetwornik cyfrowo-analogowy DAC/PWM z wyjściem OC1A na porcie PD.5 (pin 19) mikrokontrolera U4. Generowany w nim sygnał PWM jest dwustopniowo filtrowany poprzez pojedyncze dolnoprzepustowe trójniki R-C z elementami: R27 i C40 oraz R28 i C41. Oba stopnie filtrujące są sekwencyjnie separowane od wyjścia sterującego P15 poprzez wzmacniacze operacyjne U6A i U6B (MCP6002), które pracują w układach wtórników i są zasilane napięciem stabilizowanym +5 V, lokalnie blokowanym dla zakłóceń impulsowych kondensatorami: C38 (100 nF) i C39 (10 μF).

Montaż i uruchomienie

Część elektroniczna programowanego sztucznego obciążenia DC została rozdzielona między dwie płytki drukowane, pokazane na **rysunku 3** (PCB modułu sterowania) oraz na **rysunku 4** (PCB modułu wykonawczego). Montaż urządzenia z dwóch powodów najlepiej jest rozpocząć od płytki sterowania. Po pierwsze, na niej właśnie znajduje się precyzyjny obwód stabilizatora napięcia zasilającego i referencyjnego +5,00 V, potrzebnego do poprawnej pracy obu modułów. Po drugie, prawidłowe dopasowanie płytki sterowania do przedniego panelu obudowy (projekt dostosowano do standardowej obudowy typu „*KRADEX*” Z15 i zaproponowano w nim także dedykowany panel czolowy) pomoże w następnej kolejności optymalnie rozmieścić w obudowie: moduł wykonawczy z radiatorem oraz wentylatory i wszystkie pozostałe podzespoły. **Fotografia 5** przybliży szczegóły montażu elementów na „*górnej stronie*” PCB modułu sterowania. Wykonujemy standardowo, tzn. rozpoczynając od elementów najniższych i kolejno montując coraz wyższe elementy – jednak z pominięciem elementów: Q7 (tranzystor N-MOSFET, załączający zasilanie wentylatorów), POT1..POT4 (potencjometry regulacyjne – do wyprowadzenia na panelu

REKLAMA



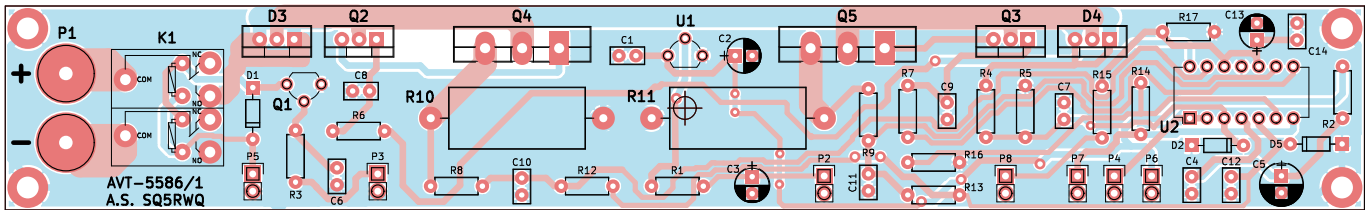
Specjalistyczne szkolenia dla elektroników i automatyków

STM32

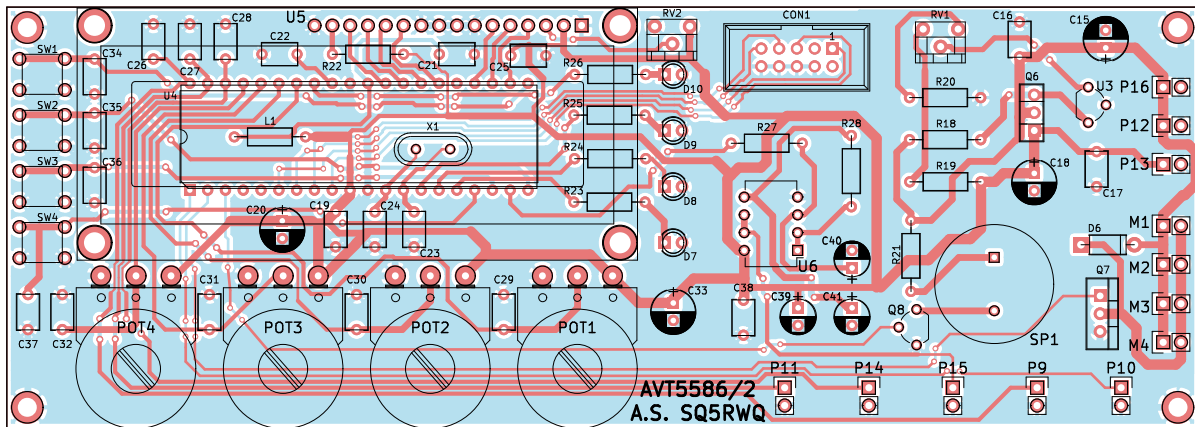
TECHDAYS

techdays@techdays.pl
TECHDAYS.PL

CERTYFIKOWANY PARTNER SZKOLENIOWY



Rysunek 3. Schemat montażowy modułu sterowania sztucznego obciążenia

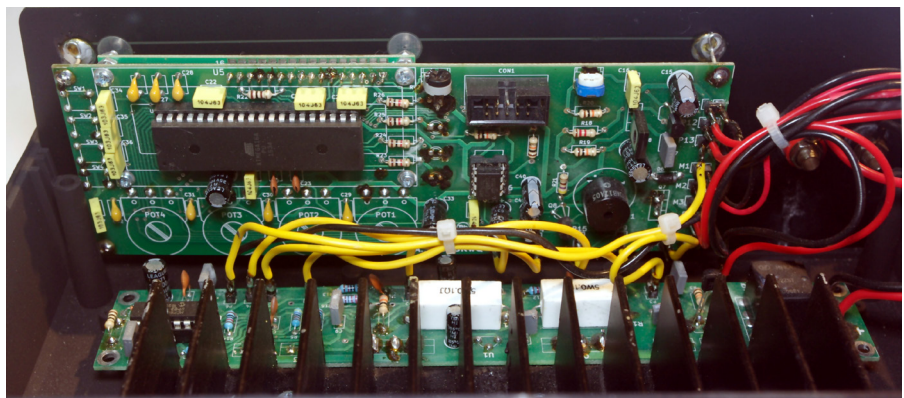


Rysunek 4. Schemat montażowy modułu wykonawczego sztucznego obciążenia

czołowym), diod LED D7...D10 (sygnalizacja stanu pracy urządzenia), wyświetlacza LCD U5 oraz przycisków SW1...SW4, którymi steruje się sztucznym obciążeniem. Są one przeznaczone do zamontowania na „spodniej stronie” PCB – w następnej kolejności.

Pod układy scalone U4 (mikrokontroler ATmega16) i U6 (wzmacniacz operacyjny MCP6002) należy zastosować podstawki. Warto też zamontować złącze programatora ISP (CON1, „KANDA”), które może okazać się bardzo dogodne w przypadku wprowadzania własnych poprawek i zmian do oprogramowania urządzenia.

Tranzystor średniej mocy Q6 w stabilizatorze +5 V nie wymaga stosowania radiatora. Wszystkie złącza na module sterowania (P9...P16 oraz wyprowadzenia do podłączenia sterowania wentylatorów M1...M4) są dostosowane do zalutowania w nich podwójnych męskich złączy „goldpin”. Na tym etapie prac można też już umieścić w podstawce zaprogramowany (fusebit’y + pamięć FLASH) mikrokontroler U6). Montaż elementów na „spodniej stronie” PCB warto jest rozpocząć od przygotowania potencjometrów regulacyjnych POT1...POT4. Potencjometrom tym należy najpierw ostrożnie dogiąć pod kątem prostym wyprowadzenia lutownicze (w trakcie wyginania warto jest zapobiegawczo zaciskać nity mocujące te wyprowadzenia), a następnie starannie zamontować je do przedniego panelu obudowy (fotografia 6). W kolejnym kroku należy dokonać dopasowania odległości montażu pozostałych elementów (POT1...POT4, D7...D10, LCD U5 oraz SW1...SW4). Przyciski SW1...SW4 powinny mieć prowadnice o długości 15...20 mm i można je przyłutować w pierwszej kolejności. Standardowy

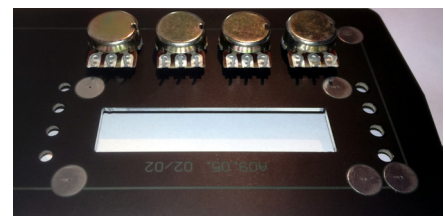


Fotografia 5. Szczegóły montażu elementów na „górnej” stronie płytki modułu sterowania

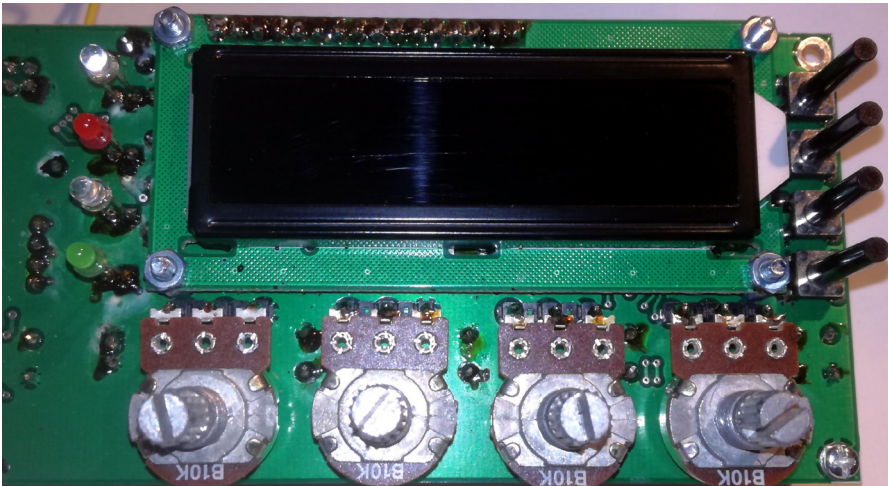
wyświetlacz LCD U5 należy od spodniej strony wyposażyć w 16-pinowe typowe męskie złącze goldpin.

Po wprowadzeniu diod LED D7...D10 w otwory lutownicze oraz otwory w przednim panelu można dokonać wstępnego dopasowania elementów. Wyświetlacz LCD U5 należy zamocować do płytki sterowania śrubami M2.5. W miejsce podłączenia potencjometrów POT1..POT4 należy włutować 5-pinowe męskie złącza „goldpin”, pozbawione pinów 2 i 4. Po ustaleniu odpowiedniego dystansu i pozycji między płytką drukowaną sterowania, a przednim panelem można zalutować końcówki diod LED D7...D10. Na tym etapie należy wstępnie przyłączyć do modułu sterowania zasilanie +12 V DC/0,5 A, a następnie potencjometrem montażowym RV1 dokładnie ustawić napięcie 5 V na emiterze tranzystora Q6 (wyjście stabilizatora). Dociskając bocznie wyprowadzenia „goldpin”, włutowane do modułu LCD (U5) do otworów lutowniczych w PCB, powinno udać się zapewnić połączenie galwaniczne, wystarczające do zadziałania

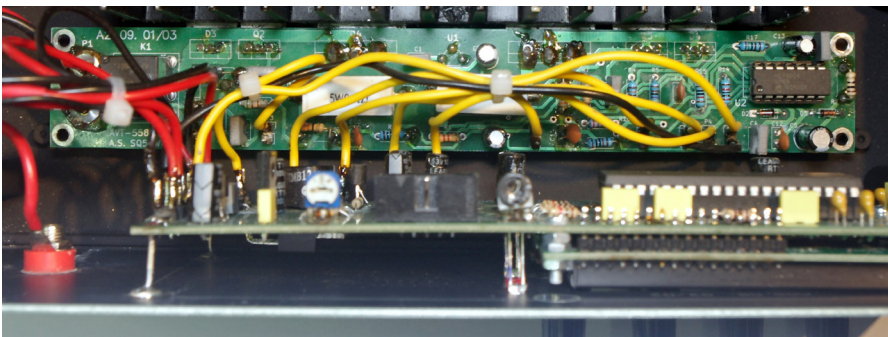
wyświetlacza LCD. Po włączeniu zasilania +12 V DC w takim stanie ekran LCD powinien wystartować, możliwe też powinno być ustawienie jego kontrastu potencjometrem montażowym RV2. Jeśli opisany test zakończył się sukcesem, to można już też trwale zalutować do PCB wszystkie wyprowadzenia „goldpin” wyświetlacza LCD. Na fotografii 7 pokazano płytkę sterowania z przyłutowanymi potencjometrami. Na koniec, także po tej stronie płytki kontrolnej, należy zamontować tranzystor Q7. Nóżki tranzystora Q7, który nie wymaga stosowania radiatora, należy odpowiednio dogiąć pod kątem



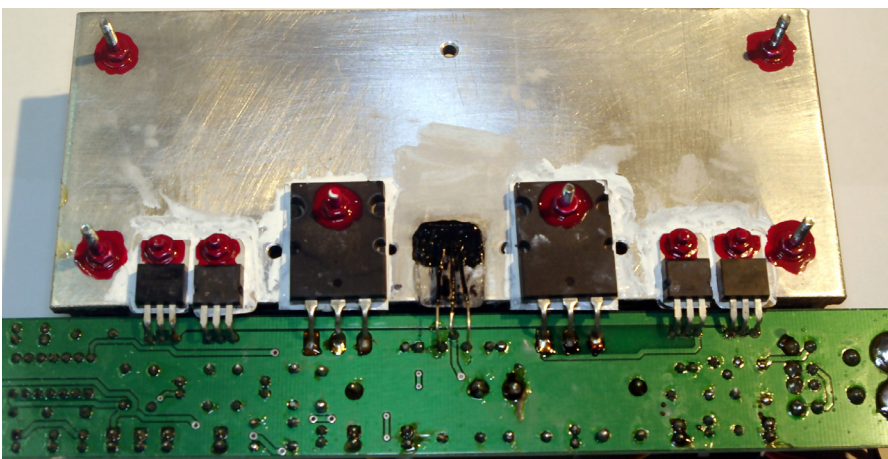
Fotografia 6. Przygotowanie potencjometrów regulacyjnych POT1...POT4 do montażu (A)



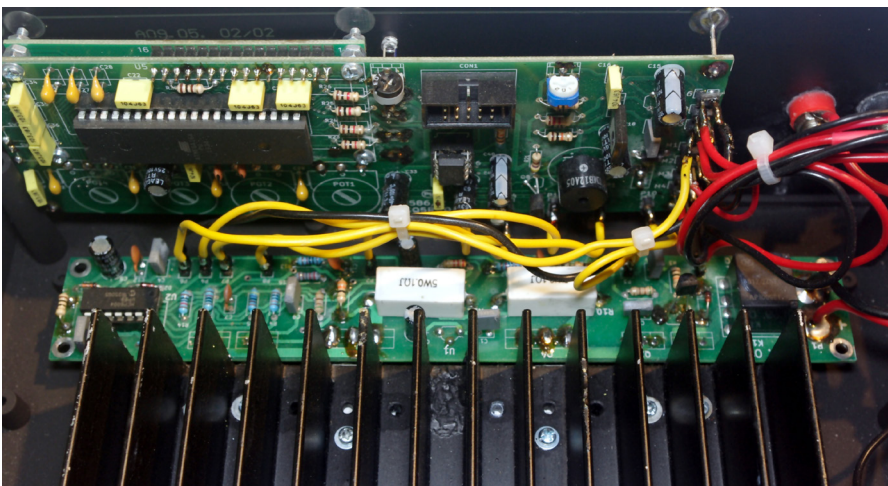
Fotografia 7. Widok zamontowanych potencjometrów regulacyjnych POT1...POT4



Fotografia 10. Montaż płytki drukowanej modułu wykonawczego (widok od góry)



Fotografia 11. Montaż płytki drukowanej modułu wykonawczego (widok od dołu)



Fotografia 12. Montaż i okablowanie obu modułów z radiatorem wewnątrz obudowy

prostim, aby po zalutowaniu swobodnie mieścił się on pomiędzy omawianą płytką sterowania a przednim panelem obudowy. Ostatecznie, zmontowaną płytkę sterowania można ponownie uruchomić na tymczasowym zasilaniu +12 V DC, podanym na złącze wejściowe P16, a następnie zweryfikować i ewentualnie skorygować: wartość napięcia stabilizowanego +5 V DC (RV1) oraz kontrast wyświetlacza LCD (RV2).

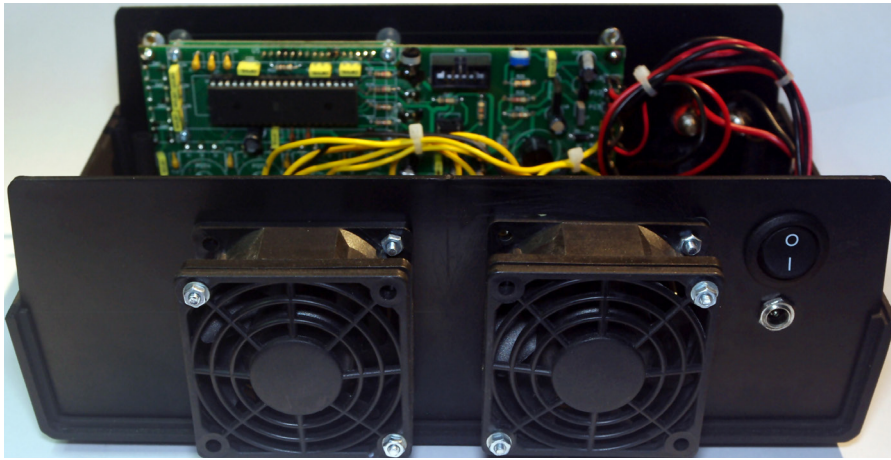
Fotografie 8 i 9 przybliżają szczegóły montażu płytki drukowanej modułu wykonawczego. Od spodu należy zamontować elementy, które w trakcie normalnej pracy mogą oddawać znaczne ilości ciepła: zabezpieczające, podwójne diody mocy Schottky (D3 i D4), tranzystory stopni sterujących w układach Darlingtona (Q2 i Q3) oraz główne tranzystory wykonawcze dużej mocy (Q4 i Q5). Pomiędzy tranzystorami Q4 i Q5 należy zamontować czujnik temperatury U1. Wymienionym elementom, mocowanym do radiatora, przed zalutowaniem należy w odpowiednim miejscu delikatnie wygiąć nóżki pod kątem prostym.

W modelu zastosowano standardowy radiator żeberkowy o wymiarach około 165 mm×70 mm×35 mm i rozstawie żeber 12 mm. Tranzystory Q2...Q5 oraz diody D3...D4 zamontowano na podkładkach izolacyjnych. Czujnik termiczny U1 należy zamocować najlepiej dokładnie pośrodku między tranzystorami mocy Q4 i Q5. Otwór pod czujnik należy wykonać wiertłem o średnicy 6 mm, a sam czujnik starannie wkleić w ten otwór nieprzewodzącym klejem dwuskładnikowym, który zapewni dobre przewodzenie ciepła z radiatora.

Wyprowadzenia P2...P8 modułu wykonawczego należy wykonać za pomocą podwójnych, męskich łączówek „goldpin”, natomiast dołączenie mierzonego źródła energii (złącze P1) trzeba wykonać odcinkami izolowanego miedzianego przewodu (linki) o przekroju min. 1,5 mm²

Poprawnie zmontowany moduł wykonawczy nie wymaga uruchomienia ani regulacji. W dolnej części obudowy warto jest wykonać cztery „postumenty”, na których zostaną oparte oddające znaczne ilości ciepła radiator. Fotografia 10 przybliży detale mocowania i okablowania obu modułów wraz z radiatorem wewnątrz obudowy. Odcinkami przewodów o przekroju około 0,75 mm² należy połączyć ze sobą parami złącza: P2-P9, P3-P10, P4-P11, P5-P12, P6-P13, P7-P14 oraz P8-P15, przy czym nie ma potrzeby prowadzenia wszystkich połączeń masowych. Wystarczy dwa – trzy z nich we w miarę odległych od siebie miejscach, o ile faktycznie oba moduły będą zamontowane blisko siebie.

Po zamontowaniu na przednim panelu obudowy gniazd podłączeniowych dla badanego źródła energii należy odpowiednio skrócić i solidnie dolutować do nich grubsze



Fotografia 13. Montaż wentylatorów oraz włącznika i gniazda zasilania 12 V DC – widok z tyłu obudowy



Fotografia 14. Montaż wentylatorów oraz włącznika i gniazda zasilania 12 V DC – widok wewnętrzny obudowy

przewody. Fotografii 11 i 12 przybliżają szczegóły montażu wentylatorów oraz włącznika i gniazda zasilania +12 V DC/1,0 A.

Z racji niewielkiej przestrzeni wewnątrz obudowy w egzemplarzu modelowym wentylatory zamocowano na zewnątrz tylnego

panelu. W egzemplarzu modelowym zastosowano dwa wentylatory 12 V DC/0,14 A i wymiarach ok. 60 mm×60 mm×20 mm, co wystarczyło do odprowadzenia mocy strat cieplnych na poziomie 100 W.

Oprogramowanie sterujące

Oprogramowanie sterujące urządzeniem zostało napisane w języku C (AVR-GCC). W projekcie wykorzystano standardową darmową bibliotekę AVR-LIBC oraz dodatkową bibliotekę w pliku „lcd.c”, opracowaną przez Lucjana Bryndzę SQ5FGB. Biblioteka „lcd.c” dostarcza podstawowych definicji i funkcji, przeznaczonych do obsługi alfanumerycznego ekranu LCD ze sterownikiem kompatybilnym z HD44780.

Program można podzielić logicznie na cztery główne moduły: definicji plików nagłówkowych, definicji parametrów i zmiennych globalnych, a także prototypów funkcji lokalnych, głównego bloku programu, funkcji pomocniczych oraz funkcji obsługi konfiguracji, przechowywanej w EEPROM.

Na listingu 1 podano właściwe ustawienia fusebitów (E=FF, H=D7, L=3E) oraz częstotliwość taktowania mikrokontrolera, równą 16 MHz. Następnie podano deklaracje parametrów i zmiennych globalnych: *linebuf* (bufor znakowy LCD), *indLCD* (wskaźnik zestawu parametrów wyświetlanych na LCD), *Iload*, *Umin*, *Umax*, *Pmax* (wartości nastaw realizowanych potencjometrami: POT1..POT4), *Ireal*, *Ureal*, *Preal*, *Treal* (rzeczywiste wartości: prądu, napięcia i mocy obciążenia oraz temperatury radiatora),

Listing 1. Deklaracje stałych i zmiennych używanych w programie

```
// Fusbit-y do zaprogramowania w MCU AVR ATmega16: E=FF, H=D7, L=3E
// Częstotliwość F_CPU (=16MHz): zdefiniowana w pliku 'Makefile'
#include <stdio.h>
#include <float.h>
#include <math.h>
#include <avr/pgmspace.h>
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#include <avr/eeprom.h>
#include "lcd.h"

char linebuf[17] = ""; // Bufor znakowy dla LCD
unsigned char indLCD = 1; // wskaźnik zestawu parametrów do wyświetlenia na LCD (1 -> pierwszy zestaw)
float Iload=0, Umin=0, Umax=0, Pmax=0; // Wartości nastaw: Iload=0, Umin=0, Umax=0, Pmax=0
float Ireal=0, Ureal=0, Preal=0, Treal=0; // Prąd obciążenia, napięcie na obciążeniu, moc w obciążeniu, temperatura radiatora
float Ereal=0, Qreal=0; // Energia i ładunek pobrane z badanego źródła
unsigned long int tics=0; // Czas pomiaru liczony przepełnieniami licznika TCNT2, czyli co 1/(16MHz/1024/256) = 16,384 msec.
const float dt0=0.016384; // Stała czasowa [sek.], odpowiadająca odstępowi czasowemu między dwoma przerwaniem OVF2 dla TCNT2
const float dt1=4.551111111e-6; // Stała czasowa [godz.], odpowiadająca odstępowi czasowemu między dwoma przerwaniem OVF2 dla TCNT2
char OK_FLAG=0, ON_FLAG=0; // Flagi statusowe: zgody (na załączenie i wysterowanie) oraz stanu (załączenia i wysterowania)
unsigned char pos_changed=1; // Flaga zmian w menu: 1 -> wyświetlenie nadrzędnego menu (wymuszone na początku obsługi)
unsigned char selected_item=1; // Aktualnie wybrana pozycja menu konfiguracyjnego
unsigned char show_pars_flag; // Pokazywanie parametrów konf. menu na starcie: 1 -> pokazuj, 0 -> nie pokazuj
unsigned char buzzer_flags; // Zachowanie buzzera: b0==1 -> buzzer ON @ 'overheat', b1==1 -> buzzer ON @ 'overload'
float T1; // Próg temperatury wyłączenia wentylatora: zdarzenie [COOLING OFF]; default: T1=40.0
float T2; // Próg temperatury załączenia wentylatora: zdarzenie [COOLING ON]; default: T2=50.0
float T3; // Próg temperatury dla stanu przegrzania: zdarzenie [OVERHEATING]; default: T3=60.0

void adc_init(void); // Inicjalizacja przetwornika ADC
unsigned int adc_read(unsigned char channel); // Odczyt wartości przez ADC (pojedyncza konwersja) z kanału o #-rze 'channel'
void pwm_init(void); // Inicjalizacja przetwornika DAC/PWM
void OutParsLCD(unsigned char KindOf, unsigned char WhichLine); // LCD - wyświetlanie wybranych parametrów
void tcnt2_cfg(void); // TCNT2 - konfiguracja preskalera /1024 i przerwania OVF2
void tcnt2_start(void); // TIMER2 - start pracy z przerwaniami OVF2 (odliczania czasu)
void tcnt2_stop(void); // TIMER2 - zatrzymanie pracy z przerwaniami OVF2 (odliczania czasu)
void load_on(void); // START pracy urządzenia: załączenie obciążenia i pomiaru
void load_off(void); // STOP pracy urządzenia: wyłączenie obciążenia i pomiaru
void change_config(void); // Konfiguracja parametrów sterowania
void show_config(void); // Odczyt i warunkowa prezentacja parametrów zapisanych w EEPROM
void show_pars_flag_cfg(void); // Modyfikacja flagi show_pars_flag
void buz_on_overload_cfg(void); // Modyfikacja flagi buz_on_overload
void buz_on_overheat_cfg(void); // Modyfikacja flagi buz_on_overheat
void T1_cfg(void); // Modyfikacja progu temperaturowego T1
void T2_cfg(void); // Modyfikacja progu temperaturowego T2
void T3_cfg(void); // Modyfikacja progu temperaturowego T3
```


Listing 2. Program główny

```

int main(void)
{
    // PORTA: IN - pomiary ADC: wartości T,U,I (PA.0..2) oraz nastawy Iload, Umin, Umax, Pmax (PA.6..3); OUT: LED_OVERHEATING (PA.7)
    DDRA=0b10000000; //DDRA=0x80;
    PORTA=0b00000000; //PORTA=0x00; // LED_OVERHEATING OFF
    // PORTB: prog. ISP (PB.5..7); nieużywane piny (PB.3..4); OUT: buzzer (PB.2), wentylatory (PB.1), Load On/Off (PB.0)
    DDRB=0b00000111; //DDRB=0x07;
    PORTB=0b00000000; //PORTB=0x00; // buzzer, wiatraki i obciążenie wyłączone
    // PORTC - obsługa ekranu LCD: wszystkie linie, poza nieużywaną PC.3 są wyjściowe (OUT)
    DDRC=0b11110111; //DDRC=0xF7;
    PORTC=0b00000000; //PORTC=0x00;
    // PORTD: IN: obsługa przycisków PD.3..4, PD.6..7; OUT: DAC/PWM OC1A PD.5, sterowanie LED (3 szt.): LED_OVERLOAD PD.0, LED_COOLING
    PD.1, LED_WORKING PD.2
    DDRD=0b00100111; //DDRD=0x27;
    PORTD=0b11011000; //PORTD=0xD8;
    // 1-pull-up'y dla przycisków, 0-wyłączone 3 szt. LED-ów, DAC/PWM/OC1A output LOW
    pwm_init(); // DAC/PWM - inicjalizacja kanału OCR1A/PD.5
    adc_init(); // ADC - inicjalizacja
    tcnt2_cfg(); // TCNT2 - preskaler=1024 i przerwanie od OVF2
    InitLcd();
    InstLcd(LCDLINE1); PutsLcd_P(PSTR("DC 'dummy load'"));
    InstLcd(LCDLINE2); PutsLcd_P(PSTR("AVT-5586 v.1.0 "));
    _delay_ms(1500);
    show_pars_flag = eeprom_read_byte( (unsigned char *) 0x01);
    buzzer_flags = eeprom_read_byte( (unsigned char *) 0x02);
    T1 = eeprom_read_float( (float *) 0x03);
    T2 = eeprom_read_float( (float *) 0x07);
    T3 = eeprom_read_float( (float *) 0x0B);
    // wciśnięte oba przyciski: "UP"/PD.6 i "DOWN"/PD.4
    if((PIND & _BV(PD6))==0 && (PIND & _BV(PD4))==0)
        change_config();
    show_pars_flag = eeprom_read_byte( (unsigned char *) 0x01);
    if(show_pars_flag) show_config();
    OK_FLAG=0; // Brak zgody na załączenie obciążenia
    ON_FLAG=0; // Obciążenie odłączone i niewysterowane
    while(1)
    {
        // PA.6 -> ADC6; (1 / 1023.0 * 10.0 A) = 0.00977517106549 A
        Iload = 0.00977517106549 * adc_read(6);
        // PA.5 -> ADC5; (1 / 1023.0 * 60.0V) = 0.058651026393 V
        Umin = 0.058651026393 * adc_read(5);
        // Operacja uniemożliwiająca ustawienie Umin > Umax
        Umin = fmin(Umin, Umax);
        // PA.4 -> ADC4; (1 / 1023.0 * 60.0V) = 0.058651026393 V
        Umax = 0.058651026393 * adc_read(4);
        // Operacja uniemożliwiająca ustawienie Umax < Umin
        Umax = fmax(Umin, Umax);
        // PA.3 -> ADC3; (1 / 1023.0 * 200.0W) = 0.19550342131 W
        Pmax = round(0.19550342131 * adc_read(3));
        // PA.5 -> ADC5; (2 * 5 / (0.1 * (1 + 10/2.2) * 1023) ) = 0.0176273576591 A
        Ireal = 0.0176273576591 * adc_read(2);
        // PA.4 -> ADC4; (1 / 1023.0 * 5.0 / 0.0448901623687) = 0.108878767081 V
        Ureal = 0.108878767081 * adc_read(1);
        // PA.3 -> ADC3; (((x/1023.0)*5.0)-0.5)/0.01=x*0.488758553275-50.0 st.C
        Treal = 0.488758553275 * adc_read(0) - 50.0;
        Preal = Ureal * Ireal;
        if((PIND & _BV(PD4))==0) // przycisk "DOWN"/PD.4
        {
            indLCD++; // Wybór następnego zestawu parametrów do wyświetlenia
            if(indLCD>6) indLCD=6; // indLCD = (indLCD>6) ? 6 : indLCD;
            _delay_ms(200); // Zapobieganie zbyt częstemu przełączaniu
        }
        if((PIND & _BV(PD6))==0) // Przycisk "UP"/PD.6
        {
            indLCD--; // Wybór poprzedniego zestawu parametrów do wyświetlenia
            if(indLCD<1) indLCD=1; // indLCD = (indLCD<1) ? 1 : indLCD;
            _delay_ms(200); // Zapobieganie zbyt częstemu przełączaniu
        }
        OutParsLCD(indLCD, 1);
        OutParsLCD(indLCD+1, 2);
        OK_FLAG = 1; // Założenie a'priori, że jest zgoda na pracę układu z obciążeniem
        PORTD &= ~_BV(PD0); // Wyłączenie sygnalizacji LED_OVERLOAD (PD.0=0)
        PORTB &= ~_BV(PB2); // Wyłączenie sygnalizacji buzzerem (PB.2=0)
        if(ON_FLAG==1 && (Ureal<Umin || Ureal>Umax || Preal>Pmax))
        {
            PORTB &= ~_BV(PB0); // Load OFF (PB.0=0) - odłączenie obciążenia
            OK_FLAG = 0; // Brak zgody na dalszą pracę układu
            PORTD |= _BV(PD0); // Załączenie sygnalizacji LED_OVERLOAD (PD.0=1)
            if( (buzzer_flags & _BV(1)) > 0 ) // Załączenie sygnalizacji buzzerem
                PORTB |= _BV(PB2);
            _delay_ms(1000); // Opóźnienie pozwalające na percepcję wystąpienia zdarzenia [OVERLOAD]
        }
        if(Treal < T1) // Zatrzymanie chłodzenia [COOLING OFF]
        {
            PORTB &= ~_BV(PB1); // Wyłącz wentylatory: PB.1=0
            PORTD &= ~_BV(PD1); // Wyłącz LED_COOLING: PD.1=0
        }
        if(Treal > T2) // Rozpoczęcie chłodzenia [COOLING ON]
        {
            PORTB |= _BV(PB1); // Załącz wentylatory: PB.1=1
            PORTD |= _BV(PD1); // Załącz LED_COOLING: PD.1=1
        }
        if(Treal < T3) // Koniec obsługi stanu przegrzania [NO OVERHEATING]
        {
            PORTA &= ~_BV(PA7); // Wyłącz LED_OVERHEATING: PA.7=0
            PORTB &= ~_BV(PB2); // Wyłącz buzzer: PB.2=0
        }
        if(Treal > T3) // Początek obsługi stanu przegrzania [OVERHEATING]
        {
            OK_FLAG=0; // Brak zgody na pracę (załączenie) układu
            PORTA |= _BV(PA7); // Załączenie sygnalizacji LED_OVERHEATING (PA.7=1)
            // Załączenie sygnalizacji buzzerem
            if( (buzzer_flags & _BV(0)) > 0 ) PORTB |= _BV(PB2);
            _delay_ms(1000);
        }
        if( OK_FLAG==1 && (PIND & _BV(PD7))==0 )
        {
            ON_FLAG=1; // Warunek START: ustawiona flaga OK_FLAG i wciśnięty OPT
            load_on(); // PRACA: ZAŁĄCZENIE obciążenia i wysterowanie
            _delay_ms(5); // Opóźnienie na ustalenie warunków pomiarowych
            tcnt2_start(); // Rozpoczęcie pomiarów: tic, Qreal, Ereal
        }
        if( OK_FLAG==0 || (PIND & _BV(PD3))==0 )
        {
            ON_FLAG=0; // Warunek STOP: wyzerowana flaga OK_FLAG lub wciśnięty SET
            load_off(); // WYŁĄCZENIE: wyłączenie obciążenia
            tcnt2_stop(); // Zatrzymanie pomiarów - koniec pracy z przerwaniami OVF2
        }
        if(ON_FLAG==1) load_on(); // PRACA: załączenie obciążenia i WYSTEROWANIE
    }
    return 0;
}

```

Ereal, *Qreal* (energia i ładunek pobrane z testowanego źródła) oraz *tics* (czas pomiaru, liczony ilością przepełnień licznika TCNT2). Stała $dt0=0.016384$ jest stałą czasową w sekundach, odpowiadająca odstępowi czasowemu między dwoma przerwaniem OVF2 dla TCNT2, natomiast stała $dt1=4.551111111e-6$ odpowiada wartości stałej *dt0*, ale liczonej w godzinach. Zmienne: *OK_FLAG* oraz *ON_FLAG* pełnią rolę flag statusowych: zgody na załączenie i występowanie obciążenia oraz bieżącego jego stanu. Natomiast *pos_changed* jest flagą zmian w menu, a *selected_item=1* pełni rolę wskaźnika aktualnie wybranej pozycji menu konfiguracyjnego. Deklaracje parametrów, przechowywanych w pamięci EEPROM, obejmują kolejno: *show_pars_flag* (pokazywanie parametrów konf. menu na starcie), *buzzer_flags* (zachowanie buzzera w przypadku wystąpienia zdarzeń: [OVERHEAT] oraz [OVERLOAD]), *T1* (próg temperatury wyłączenia wentylatora – zdarzenie [COOLING OFF]), *T2* (próg temperatury załączania wentylatora – zdarzenie [COOLING ON]), *T3* (próg temperatury dla stanu przegrzania – zdarzenie [OVERHEATING]).

Prototypy funkcji lokalnych uwzględniają następujące podprogramy:

- *adc_init()* inicjalizująca przetwornik ADC,
- *adc_read()* – pobierająca wartości odczytane przez przetwornik A/C z kanału o wskazanym numerze,
- *pwm_init()* – inicjalizująca przetwornik DAC/PWM,
- *OutParsLCD()* – wyświetlająca wybrane parametry na ekranie LCD,
- *tcnt2_cfg()* – konfiguracja preskalera i przerwania OVF2 dla licznika TCNT2,
- *tcnt2_start()* – start odliczania czasu przez TIMER2,
- *tcnt2_stop()* – zatrzymanie odliczania czasu przez TIMER2,
- *load_on()* – załączenie obciążenia i rozpoczęcie pomiaru,
- *load_off()* – odłączenie obciążenia i zakończenie pomiaru,
- *change_config()* – konfiguracja parametrów sterowania,
- *show_config()* – odczyt i warunkowa prezentacja parametrów zapisanych w EEPROM,
- *show_pars_flag_cfg()* – modyfikacja flagi *show_pars_flag*,
- *buz_on_overload_cfg()* – modyfikacja flagi *buz_on_overload*,
- *buz_on_overheat_cfg()* – modyfikacja flagi *buz_on_overheat*,
- *T1_cfg()* – modyfikacja progu temperaturowego *T1*,
- *T2_cfg()* – modyfikacja progu temperaturowego *T2*,

Listing 3. Funkcje pomocnicze

```
// DAC/PWM - inicjalizacja
void pwm_init(void)
{
    // INICJALIZACJA PWM - TIMER1, kanał OCR1A, port PD.5
    // make sure to make OC1A pin (pin PD.5 for ATmega16) as output pin
    DDRD |= (1<<PD5);
    // Clear OC1A/OC1B on Compare Match, set OC1A/OC1B at TOP
    TCCR1A |= (1<<COM1A1)|(1<<COM1B1);
    // Mode 7: Fast PWM, 10 bit: WGM13=0, WGM12=1, WGM11=1, WGM10=1
    TCCR1A |= (1<<WGM10)|(1<<WGM11)|(1<<WGM12);
    TCCR1B |= (1<<WGM12); // Mode 7: Fast PWM, 10 bit: WGM13=0, WGM12=1
    // Prescaler = 1, Fclk=CLK_io; CS12=0, CS11=0, CS10=1; Fpwm = CLK_io, 10 bit res.
    TCCR1B |= (1<<CS10);
}

// ADC - inicjalizacja
void adc_init(void)
{
    ADMUX = (1<<REFS0); // AREF = AVcc
    ADCSRA = (1<<ADEN)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0); // ADC Enable
}

// ADC - odczyt wartości z wybranego kanału w zakresie 0..7
unsigned int adc_read(unsigned char channel)
{
    channel &= 0b0000111; // ograniczenie zakresu #-ów kanału do 0..7
    ADMUX = (ADMUX & 0xF8)|channel; // zerowanie 3 najmłodszych
    ADCSRA |= (1<<ADSC); // wpisanie 1 do bitu ADSC rozpoczyna pojedynczą konwersję
    while(ADCSRA & (1<<ADSC)); // oczekiwanie na zakończenie konwersji, ADSC==0
    return (ADC);
}

// LCD - wyświetlanie wybranych parametrów
void OutParsLCD(unsigned char KindOf, unsigned char WhichLine)
{
    switch(KindOf) // Formatowanie
    {
        case 1:
            sprintf(linebuf, sizeof(linebuf), "%.1fV<=U<=%.1fV ", Umin, Umax);
            break; // (1) Umin, Umax
        case 2:
            sprintf(linebuf, sizeof(linebuf), "I<=%.1fA P<=%.0fW ", Iload, Pmax);
            break; // (2) Iload, Pmax
        case 3:
            sprintf(linebuf, sizeof(linebuf), "U=%4.1fV I=%4.1fA ", Ureal, Ireal);
            break; // (3) Ureal, Ireal
        case 4:
            sprintf(linebuf, sizeof(linebuf), "P=%4.1fW T=%2.fdeg. ", Preal, Treal);
            break; // (4) Preal, Treal
        case 5: // (5) t(ime)
            float t;
            t = tics * dt0; // czas liczony w sekundach (na potrzeby prezentacji)
            int h=0, m=0; float s=0;
            h = floor( t/3600.0 );
            m = floor( ( t - (3600.0*h) ) / 60.0 );
            s = t-3600.0*h-60.0*m;
            sprintf(linebuf, sizeof(linebuf), "t=%dh %dm %.1fs ", h, m, s);
            break;
        case 6:
            sprintf(linebuf, sizeof(linebuf), "Q=%.3f Ah ", (Qreal*dt1) );
            break; // (6) Qreal
        case 7:
            sprintf(linebuf, sizeof(linebuf), "E=%.3f Wh ", (Ereal*dt1) );
            break; // (7) Ereal
        default:
            break;
    }
}
```

- *T3_cfg()* – modyfikacja progu temperaturowego *T3*.

Na listingu 2 pokazano główny blok programu z funkcją *main()*. W pierwszej kolejności realizowana jest konfiguracja portów mikrokontrolera. Po zakończeniu konfiguracji portów następują kolejno: konfiguracje oraz inicjalizacje przetworników C/A, PWM oraz A/C, a także licznika-timera TCNT2. Następnie jest inicjalizowany wyświetlacz LCD i wyświetlany ekran powitalny urządzenia. Dalej, następuje odczyt parametrów zapisanych w pamięci EEPROM. Jeśli w kolejnej chwili zostanie wykryte jednoczesne naciśnięcie przycisków *UP* i *DOWN*, to nastąpi wywołanie funkcji konfiguracyjnej *change_config()*. Następnie jest wykonywana autokonfiguracja oraz warunkowa prezentacja parametrów przechowywanych w pamięci EEPROM (funkcja *show_config*), przy czym wystąpienie tego zdarzenia zależy od wartości bitu *show_pars_flag*.

Przed rozpoczęciem głównej, „nieskończonoj” pętli programu, ustawiane są główne flagi sterujące pracą sztucznego obciążenia: *OK_FLAG=0* (brak zgody na załączenie

obciążenia) oraz *ON_FLAG=0* (obciążenie odłączone i niewystępowanie). Pętla nieskończona jest wykonywana przez cały okres normalnej pracy urządzenia: zarówno w trakcie pracy obciążenia jak i w stanie jego wyłączenia. Rozpoczyna się ona blokiem odczytów z wejść przetwornika A/C, mających na celu pomiary (odczyt) nastaw, zrealizowanych potencjometrami regulacyjnymi: *Iload*, *Umin*, *Umax*, *Pmax*:

- $Iload=0,00977517106549*adc_read(6)$, gdzie $wsp. K=1/1023*10,0 A$,
- $Umin=0,058651026393*adc_read(5)$, gdzie $wsp. K=1/1023*60,0 V$,
- $Umax=0,058651026393*adc_read(4)$, gdzie $wsp. K=1/1023*60,0 V$,
- $Pmax=round(0,19550342131*adc_read(3))$, gdzie $wsp. K=1/1023*200,0 W$.

Funkcje $Umin=fmin(Umin,Umax)$ oraz $Umax=fmax(Umin,Umax)$ uniemożliwiają dokonanie nieprawidłowych ustawień: $Umin>Umax$ oraz $Umax<Umin$. Pomiar wartości chwilowych: *Treal*, *Ureal* oraz *Ireal*, a także wyliczenie mocy strat termicznych

Listing 3. cd.

```

}
switch(WhichLine) // Wybór linii
{
    case 1:
        InstLcd(LCDLINE1);
        break; // UpperLine
    case 2:
        InstLcd(LCDLINE2);
        break; // LowerLine
    default:
        break;
}
PutsLcd(linebuf); // Wyprowadzenie znaków
}
// TIMER2 - konfiguracja jako zwykły licznik z preskalerem /1024 i aktywnym przerwaniem OVF2
void tcnt2_cfg(void)
{
    TCCR2 = (_BV(CS22)|_BV(CS21)|_BV(CS20)); // preskaler /1024
    TIMSK |= _BV(TOIE2); // int. OVF2 enabled
    TIMSK &= ~_BV(OCIE2); // int. OC2 disabled
}
// TIMER2 - obsługa cyklicznego przerwania OVF2
ISR(TIMER2_OVF_vect, ISR_BLOCK)
{
    tics++; // inkrementacja licznika przepełnień TCNT2
    Ereal += Preal; // mnożenie przez dt1 dopiero na etapie prezentacji Ereal
    Qreal += Ireal; // mnożenie przez dt1 dopiero na etapie prezentacji Qreal
}
// ISR_ANY - obsługa domyślna dla pozostałych przerw (nic nie robi - tylko powrót z przerwania)
ISR(BADISR_vect) {}
// TCNT2 - start pracy z przerwaniami OVF2 (odliczania czasu)
// start pomiarów: tic (czas - licznik przerw), Qreal, Ereal
void tcnt2_start(void)
{
    tics = 0; // reset licznika przepełnień TCNT2
    Ereal = 0; // reset licznika pobranej energii
    Qreal = 0; // reset licznika pobranego ładunku
    TCNT2 = 0; // reset licznika TCNT2
    sei(); // odblokowanie przerw
}
// TIMER2 - zatrzymanie pracy z przerwaniami OVF2 (odliczania czasu)
void tcnt2_stop(void)
{
    cli(); // zablokowanie przerw
}
// PRACA urządzenia: załączenie obciążenia iysterowanie
void load_on(void)
{
    PORTB |= _BV(PB0); // Load ON (PB.0=1)
    PORTD |= _BV(PD2); // LED_WORKING = ON (PD.2=1)
    OCR1A = round(78.43 * Iload); // Nastawienie prądu Iload poprzez DAC/PWM: (1023 * (10+1.5) 1.5 / 5.0 / 2 / 0.1) = 78.43
}
// WYŁĄCZENIE urządzenia: wyłączenie obciążenia i brakysterowania
void load_off(void)
{
    PORTB &= ~_BV(PB0); // Load OFF -> PB.0=0
    PORTD &= ~_BV(PD2); // LED_WORKING = OFF -> PD.2=0;
    OCR1A = 0; // Nastawienie prądu Iload poprzez wyzerowanie wyjścia DAC/PWM
}

```

w obciążeniu P_{real} , następuje zgodnie z wyrażeniami:

- $I_{real}=0,0176273576591*adc_read(2)$, gdzie: $wsp. K=(2^5 \cdot 0 V / (0,1 W * (1+10 kW/2,2 kW) * 1023)) A$,
- $U_{real}=0,108878767081*adc_read(1)$, gdzie $wsp. K=(1/1023 * 5,0 / 0,0448901623687) V$,
- $T_{real}=0,488758553275*adc_read(0)-50,0$, co odpowiada przekształceniu wartości X , zmierzonej przez przetwornik ADC3, wg formuły: $(((X/1023) * 5,0 V - 0,5 V) / 0,01 V = X * 0,488758553275 - 50,0 [^{\circ}C]$.
- $P_{real}=U_{real} * I_{real}$.

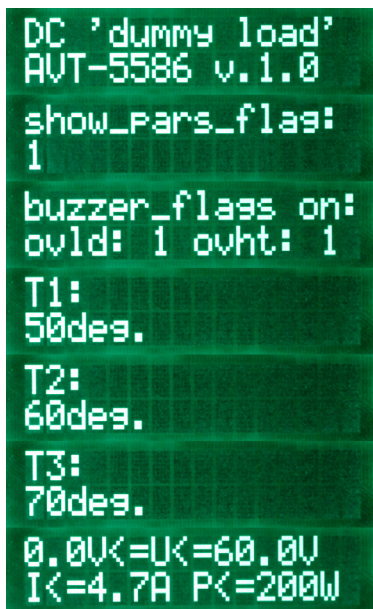
Wynik obliczeń jest prezentowany na wyświetlaczu LCD. Odczyty podzielono na 7 wierszy wyświetlane w 6 parach. Zmianę wyświetlanej pary realizują przyciski UP i DOWN. Za wyświetlanie danego wiersza parametrów odpowiada funkcja $OutParsLCD(x,y)$. Funkcja jest wywoływana dwukrotnie: raz dla górnego a drugi raz dla dolnego wiersza wyświetlacza LCD.

Dalej następuje właściwy blok kontrolno-sterujący, którego zadaniem jest sterować bezpieczną pracą całego urządzenia. Cała koncepcja bezpiecznego sterowania oparta jest o pewien schemat logiczny, w którym cyklicznie w nieskończonej pętli, na początku każdego wykonania bloku kontrolno-sterującego czynione jest założenie *a priori*, że jest zgoda na pracę układu z obciążeniem (brak przeciwwskazań), co wyrażane jest ustawieniem flagi $OK_FLAG=1$. Naturalną konsekwencją jest wyłączenie sygnalizacji stanu [OVERLOAD] na diodzie LED ($PD.0=0$) oraz buzzerze ($PB.2=0$). Oczywiście, w sytuacji, gdy taki stan nie miał miejsca, po prostu nie zadzieje się nic znaczącego. Natomiast w przypadku, gdy ten stan miał miejsce, to po prostu wymienione indykatory zostaną „zagaszone” na krótką, niezauważalną dla obserwującego człowieka chwilę – dla momentu bardzo szybkiej, ponownej aktywacji. Dalej, jeśli obciążenie jest w stanie pracy, (sprawdzany warunek: $ON_FLAG==1$), to następuje kontrola stanu parametrów roboczych:

U_{real} oraz P_{real} (temperatura T_{real} jest kontrolowana w osobnym bloku programu). Jeśli stwierdzony zostanie któryś z podanych warunków: $U_{real} < U_{min}$ lub $U_{real} > U_{max}$ lub $P_{real} > P_{max}$, to nastąpi natychmiastowe odłączenie obciążenia ($PB.0=0$), ustawiona zostanie flaga $OK_FLAG=0$, wskazującą na brak zgody na dalszą aktywną pracę obciążenia, oraz czasowo na okres jednej sekundy załączona zostanie sygnalizacja wystąpienia stanu [OVERLOAD]: za pomocą diody LED ($PD.0=1$) oraz, jeśli ustawiony jest odpowiedni bit w zmiennej flagowej $buzzer_flags$, to także załączony zostanie buzzer ($PB.2=1$). Opóźnienie rzędu jednej sekundy ma pozwolić na percepcję wystąpienia zdarzenia [OVERLOAD], natomiast po upływie tego okresu czasu, z racji odłączenia obciążenia (flaga $ON_FLAG==0$) warunek na wystąpienie zdarzenia [OVERLOAD] nie będzie testowany, zatem stan ten nie zostanie podtrzymany.

Dalej następuje kontrola temperatury T_{real} oraz warunkoweysterowanie wentylatorów i buzzera, modyfikacja flagi zgody

na pracę obciążenia *OK_FLAG* oraz przełączanie odpowiednich sygnalizacyjnych diod LED. Kontrola ta oparta jest o kolejne testowanie przekroczenia (w górę lub w dół) poszczególnych progów temperaturowych (*T1*, *T2*, *T3*) i przypomina nieco realizację tzw. automatu stanowego, w którym wystąpienie danego warunku przełącza urządzenie do innego stanu. Spełnienie pierwszego z warunków (*Treal<T1*) spowoduje przejście do stanu [COOLING OFF], a w konsekwencji: zatrzymanie pracy wentylatorów (*PB.1=0*) oraz wyłączenie sygnalizacji odpowiednią diodą LED (*PD.1=0*). Z kolei spełnienie kolejnego warunku (*Treal>T2*, gdzie: *T2>T1*) wprowadzi urządzenie w stan [COOLING ON] i wywoła rozpoczęcie działania wentylatorów (*PB.1=1*) oraz załączenie sygnalizacji odpowiednią diodą LED (*PD.1=1*). Spełnienie następnego warunku (*Treal<T3*, gdzie: *T3>T2>T1*) prowadzi do zakończenia obsługi stanu przegrzania [NO OVERHEATING], realizowanego poprzez wyłączenie sygnalizacji odpowiednią diodą LED (*PA.7=0*) oraz buzzera (*PB.2=0*). Ostatni ze sprawdzanych warunków (*Treal>T3*, gdzie: *T3>T2>T1*), w wypadku spełnienia spowoduje rozpoczęcie obsługi stanu przegrzania [OVERHEATING]. Następuje wówczas: ustawienie flagi *OK_FLAG=0*, oznaczającej brak zgody na pracę (załączenie) układu oraz załączenie sygnalizacji tego stanu odpowiednią diodą LED (*PA.7=1*). Dodatkowo, jeśli ustawiony jest odpowiedni bit w zmiennej flag buzzera *buzzer_flags*, to załączona zostanie także sygnalizacja buzzerelem (*PB.2=1*). Po jednosekundowym opóźnieniu, mającym za zadanie pozwolić operatorowi urządzenia na percepcję wystąpienia zdarzenia [OVERHEATING], program przechodzi do dalszej standardowej obsługi w nieskończonej pętli.



Fotografia 15. Zawartość informacyjnych ekranów LCD, wyświetlanych po załączeniu urządzenia

Ostatnim, niezwykle istotnym blokiem w tejże pętli jest tzw. blok wykonawczy obciążenia, który realizuje dwa główne zadania: załączanie (start) lub wyłączenie (zatrzymanie) pracy obciążenia. W pierwszym przypadku sprawdzany jest warunek dla flagi *OK_FLAG==1* oraz stan naciśnięcia przycisku "OPT" (*PD.7==0*) i, jeśli oba te warunki są spełnione jednocześnie, to następuje: zmiana stanu flagi *ON_FLAG* na 1, uruchomienie (załączenie) pracy obciążenia funkcją *load_on()*, po pięciu milisekundach opóźnienia, przeznaczonych na ustalenie jednoznacznych warunków pomiarowych, wywołanie funkcji *tcnt2_start()* inicjalizuje pracę licznika *TCNT2* z przerwaniem *OVF2*, co faktycznie rozpoczyna pomiary (zliczanie) w zmiennych: *tic* (czas – licznik przerwania), *Qreal* (pobrano ładunek) oraz *Ereal* (odebrana energia elektryczna). W drugim przypadku sprawdzany jest warunek dla flagi *OK_FLAG==0* oraz stan naciśnięcia przycisku "SET" (*PD.3==0*) i, jeśli spełniony jest którykolwiek z tych warunków, to następuje: zmiana stanu flagi *ON_FLAG* na 0, zatrzymanie pracy (odłączenie) obciążenia funkcją *load_off()* oraz zatrzymanie pomiarów (koniec pracy z przerwaniem *OVF2* licznika *TCNT2*) czasu, ładunku i energii. Po wykonaniu tych dwóch prostych testów sprawdzany jest warunek na wartość flagi *ON_FLAG==1* i, w przypadku spełnienia

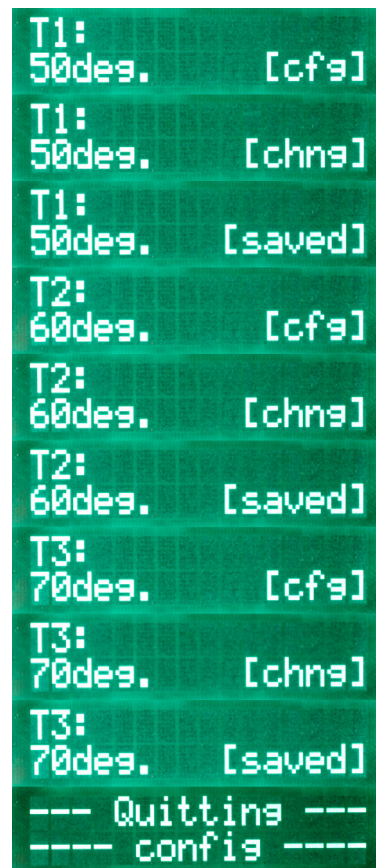


Fotografia 16. Zawartość ekranów konfiguracyjnych LCD sztucznego obciążenia DC (cz. I)

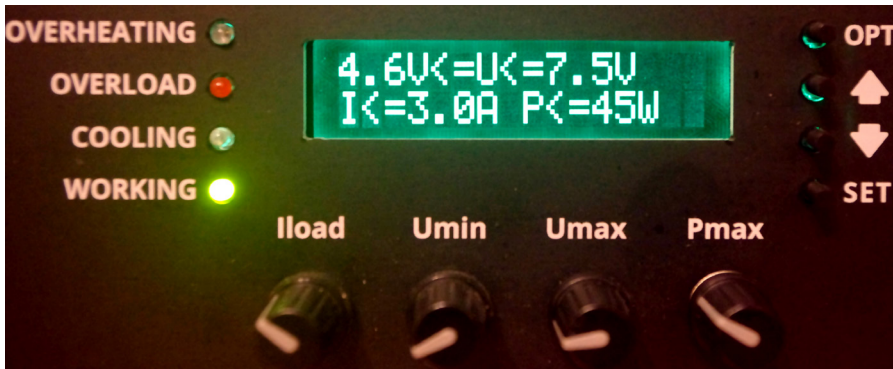
tego warunku, ponownie (cyklicznie) wywoływana jest funkcja *load_on()*, przy czym to cykliczne wywołanie jest kluczowe w tym miejscu programu nie ze względu na potrzebę włączenia (zazwyczaj włączonego już w tym stanie flagi *ON_FLAG*) obciążenia, ale z powodu możliwej konieczności modyfikacji poziomuysterowania prądu obciążenia *Iload* poprzez zmianę nastawienia przetwornika DAC/PWM (wartość zmiennej *OCR1A*).

Na **listingu 3** pokazano funkcje pomocnicze. Funkcja *pwm_init()* odpowiada za inicjalizację przetwornika DAC/PWM, polegającą na ustawieniu jako wyjściowy pinu *PD.5* oraz ustawieniu *TIMER*-a 1-ego do pracy z kanałem *OCR1A*. Zadanie to jest realizowane poprzez ustawienie właściwych wartości (bitów) w rejestrach *TCCR1A* oraz *TCCR1B*. Przetwornik DAC/PWM z wyjściem *OCR1A* jest skonfigurowany tak, że po osiągnięciu zadanej parametrem wejściowym *OC1A* wyjście przetwornika zmienia stan z wysokiego na niski. Dodatkowo, ustawiany jest tryb „Fast PWM” z rozdzielczością 10 bitów oraz preskalerem równym 1 (brak podziału częstotliwości taktującej przetwornik DAC/PWM). Jakkolwiek, wobec ustawionej 10-bitowej rozdzielczości przetwornika, częstotliwość sygnału PWM będzie wynosiła $F_{pwm} = F_{CPU} / (2^{10}) = F_{CPU} / 1024 = 15,625 \text{ kHz}$.

Kolejna funkcja *adc_init()* ma za zadanie zainicjować pracę przetwornika A/C. Odbywa



Fotografia 17. Zawartość ekranów konfiguracyjnych LCD sztucznego obciążenia DC (cz. II)



Fotografia 18. Widok panelu sterującego sztucznego obciążenia DC

się to poprzez ustawienie właściwych bitów w rejestrach ADMUX oraz ADCSRA. Funkcja `adc_read()` powoduje próbkowanie oraz konwersję wskazanego kanału A/C. Funkcja zwraca wartość pobranej próbki. Kolejną funkcją jest `OutParsLCD()`, której rolą jest wyświetlanie wybranych parametrów na ekranie LCD. Funkcja `tcnt2_cfg()` realizuje konfigurację TIMER2 jako zwykłego licznika z preskalerem /1024 i aktywnym przerwaniami OVF2. Funkcja `tcnt2_start()` uruchamia pracę z przerwaniami od flagi OVF2 oraz przygotowuje do rozpoczęcia pomiarów poprzez wyzerowanie zmiennych: `tics=0` (czas – licznik wystąpienia przerwań), `Qreal=0` (pobrano ładunek), `Ereal=0` (odebrana energia). Zerowany jest również licznik TCNT2 oraz załączane przerwanie za pomocą `sei()`. Funkcja `tcnt2_stop()` powoduje zatrzymanie pomiarowego odliczania czasu poprzez licznik TCNT2. Funkcja `load_on()` powoduje załączenie (rozpoczęcie pracy) urządzenia oraz wysterowanie tranzystorów wykonawczych. Odłączenie obciążenia i anulowanie jego wysterowania realizuje funkcja `load_off()`.

Na listingu 4 zamieszczono funkcje obsługi zmian konfiguracji. Funkcja `show_config()` dokonuje inicjowania oraz prezentacji parametrów zapisanych w EEPROM – odczytuje wszystkie parametry przechowywane

w EEPROM (flagi `show_pars_flag` i `buzzer_flags` oraz progi temperaturowe $T1$, $T2$ i $T3$) a następnie – jeśli któryś z tych parametrów ma wartość spoza dopuszczalnego zakresu, to automatycznie nadawana mu jest wartość domyślna (`show_pars_flag=1`, `buzzer_flags=3`, $T1=50.0$, $T2=T1+10.0$, $T3=T2+10.0$). Jeśli wartość flagi `show_pars_flag` wynosi 1, to w trakcie realizacji omawianej procedury wyświetlane są kolejno przyjęte wartości analizowanych parametrów z opóźnieniami równymi 1,5 sek. Funkcja `change_config()` umożliwia zmianę konfiguracji parametrów sterowania.

Regulacja, konfiguracja i obsługa zasilacza

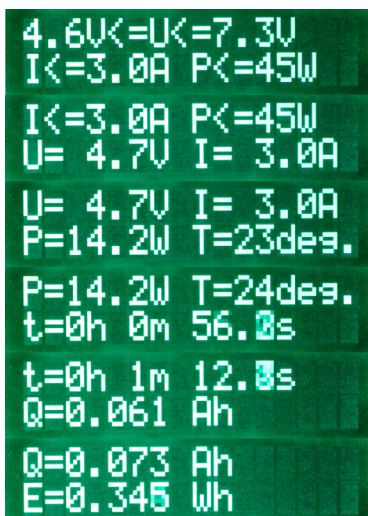
Regulacja zasilacza sprowadza się wyłączenie do ustawienia potencjometrów RV1: (referencyjne napięcie zasilania +5,00 V DC) oraz RV2 (kontrast wyświetlacza LCD), omówione w dziale poświęconym montażowi i uruchomieniu. Natomiast nieco więcej wysiłku wymaga poprawne skonfigurowanie parametrów, przechowywanych w pamięci EEPROM. Po załączeniu urządzenia po raz pierwszy nieskonfigurowanego urządzenia na ekranie LCD zostanie pokazany ciąg komunikatów, zbliżony do pokazanego na fotografii 13.

Pierwszy ekran jest ekranem powitalnym. Następnie prezentowane są kolejno wartości ustawień parametrów-flag – w tym przypadku: `show_pars_flag=1`, `buzzer_flag_on_ovld=1` i `buzzer_flag_on_ovht=1`, $T1=50.0$, $T2=60.0$ oraz $T3=70.0$, przy czym, jeśli miałyby miejsce pierwotna autokonfiguracja przyrządu, to podane progi temperaturowe miałyby wartości odpowiednio: 40, 50 i 60 stopni Celsjusza. Ostatni z pokazanych na fotografii 13 wyświetlanych ekranów LCD jest jednocześnie pierwszym normalnym ekranem roboczym i zostanie omówiony dokładniej w dalszej części, poświęconej normalnej eksploatacji sztucznego obciążenia.

Jeśli w momencie włączenia urządzenia (a dokładnie: w chwili zakończenia wyświetlania ekranu powitalnego LCD) będą jednocześnie wciśnięte przyciski: „UP” (strzałka w górę) oraz „DOWN” (strzałka w dół),

to nastąpi wejście do menu konfiguracji parametrów, przechowywanych w pamięci EEPROM. Na fotografiach 14 i 15 (w dwóch częściach) pokazane zostały przykłady wszystkich możliwych ekranów tego menu, przy czym pierwszy i ostatni, to ekrany: powitalny i pożegnalny, natomiast pozostałe, to „trójki” ekranów, jakie zostaną wyświetlone w pełnych cyklach zmian parametrów: `show_pars_flag`, `buzzer_flag_on_ovld`, `buzzer_flag_on_ovht`, $T1$, $T2$ oraz $T3$. Jak trudno zauważyć, każdy z wymienionych parametrów może być wyświetlony na trzech „poziomach zagłębienia” w menu, wskazanych dodatkowym napisem w prawym dolnym rogu ekranu LCD: „[cfg]” (poziom podstawowy), „[chng]” (poziom niższy, na którym można zmieniać dany parametr) oraz „[saved]” (poziom niższy po zapisaniu danego parametru w pamięci EEPROM). W zależności od aktualnie wybranego poziomu menu poszczególne przyciski mają następujące przeznaczenia: „OPT” – wyjście poziom wyżej lub wyjście z menu, „UP” – zmiana pozycji menu na kolejną lub zwiększenie wartości modyfikowanego parametru, „DOWN” – zmiana pozycji menu na wcześniejszą lub zmniejszenie wartości modyfikowanego parametru, „SET” – wejście poziom niżej lub zatwierdzenie (zapis w EEPROM) wartości danego parametru.

Po skonfigurowaniu parametrów roboczych w pamięci EEPROM można przejść do normalnej eksploatacji urządzenia. Fotografia 16 przedstawia wszystkie elementy operacyjne, dostępne na przednim panelu urządzenia. Po lewej stronie ekranu LCD znajdują się diody LED, sygnalizujące stany: [WORKING] (normalna praca załączonego obciążenia), [COOLING] (załączenie wentylatorów chłodzących radiator), [OVERLOAD] (przekroczenie wartości któregoś z parametrów: $Umin$, $Umax$, $Pmax$) oraz [OVERHEATING] (przekroczenie wartości górnego, maksymalnego progu temperaturowego $T3$). Pod wyświetlaczem LCD znajdują się (kolejno od lewej) potencjometry nastaw: prądu roboczego $Ireal$, dopuszczalnego napięcia minimalnego $Umin$, dopuszczalnego napięcia maksymalnego $Umax$ oraz dopuszczalnej mocy maksymalnej $Pmax$. Należy tu podkreślić, że funkcje obsługujące odczyty parametrów $Umin$ i $Umax$ mają wbudowane ograniczenia, dzięki którym nie jest możliwe dokonanie błędnych nastaw, dla których $Umin > Umax$. Po prawej stronie ekranu LCD znajdują się cztery przyciski „micro-switch”, których role w trakcie obsługi menu konfiguracyjnego omówiono w tekście powyżej. Natomiast w trakcie normalnej pracy operacyjnej urządzenia są one następujące: „OPT” – załączenie obciążenia i rozpoczęcie pomiarów, „UP” (strzałka w górę) – przewinięcie do poprzedniego dwuwiersza parametrów na ekranie LCD, „DOWN” (strzałka w dół)



Fotografia 19. Zawartość ekranów LCD wyświetlanych w trybie roboczym urządzenia (kolejne możliwe opcje)

– przewinięcie do kolejnego dwuwiersza parametrów na ekranie LCD, „SET” – odłączenie obciążenia i zatrzymanie pomiarów. Na ostatniej już **fotografii 17** widzimy sześć możliwych kombinacji informacji, wyświetlanych na ekranie LCD. Są one skomponowane z siedmiu możliwych wierszy – zestawów informacji (parametrów), które występują w parach będących połączeniem dwóch kolejnych wierszy. Dlatego właśnie w każdym z sześciu zaprezentowanych ekranów ostatni wiersz poprzedniego ekranu zawiera dokładnie te same parametry, co poprzedni wiersz ekranu następnego. Przełączanie między poszczególnymi kombinacjami informacji (dwuwierszy) następuje za pomocą przycisków „UP” oraz „DOWN” (strzałka w dół), a kolejne wiersze składowe (patrzac od góry na fotografię 17) prezentują: ustawione ograniczenia napięcia *Umin* i *Umax*, ustawiony prąd roboczy *Iload* oraz limit mocy strat *Pmax*, rzeczywiste (zmierzone) wartości napięcia *Ureal* oraz prądu *Ireal* dla podłączonego obciążenia, rzeczywistą (zmierzoną) wartość mocy strat *Preal*, odebranej ze źródła oraz temperatury radiatora *Treal* w pobliżu tranzystorów wykonawczych, czasu *t* (w formacie: *hh:mm:ss.s*), naliczonego od momentu rozpoczęcia pracy obciążenia, ładunku *Qreal* [Ah] i – na koniec – energii *Ereal* [Wh], pobranych z testowanego źródła.

Praca ze sztucznym obciążeniem nie wymaga żadnej szczególnej, dodatkowej wiedzy. Urządzenie uruchamiamy i zatrzymujemy przyciskami „SET” i „OPT” (start/stop), a nastawiane potencjometrami i mierzone parametry odczytujemy z ekranu LCD, który „przewijamy” przyciskami „UP” oraz „DOWN”. Pracując ze znacznymi prądami obciążenia należy koniecznie zapewnić połączenia galwaniczne o jak najmniejszej rezystancji szeregowej. Stany pracy urządzenia są sygnalizowane przez diody LED i ewentualnie przez buzzer. Po zatrzymaniu pracy urządzenia wyniki pomiarów można nadal przeglądać (bez ograniczeń czasowych), a ich wartości zostaną wyzerowane dopiero po ponownym uruchomieniu sztucznego obciążenia. Choć w oprogramowaniu sterującym jako górny limit temperatury radiatora ustawiono 70°C (stan [OVERHEATING]), to dla zwiększenia bezpieczeństwa pracy i trwałości urządzenia warto jest pracować ze znacznie niższymi progami temperatur, np. $T1=40^{\circ}\text{C}$, $T2=50^{\circ}\text{C}$, $T3=60^{\circ}\text{C}$, przy czym przy odbieranych przez sztuczne obciążenie mocach strat przekraczających 100 W należy mieć na uwadze konieczność zastosowania więcej niż dwóch (3...4 sztuki) wentylatorów chłodzących radiator.

Podsumowanie projektu

W artykule opisano projekt stałoprądowego sztucznego obciążenia, sterowanego

mikroprocesorem i o dość zaawansowanych możliwościach pomiarowych oraz o stosunkowo znacznej maksymalnej mocy strat, równej 200 W. Mimo nieskomplikowanej budowy urządzenie jest w stanie w większości typowych zastosowań konkurować z licznymi urządzeniami fabrycznymi, które nie należą do bardzo tanich. Istotną wartością dodaną są spore możliwości regulacji parametrów granicznych urządzenia – także już w trakcie jego pracy. Można także dokonywać modyfikacji konstrukcji pod kątem indywidualnych potrzeb czytelników z większymi ambicjami konstruktorskimi.

Projekt ten jest dedykowany zarówno osobom, które chcą wejść w posiadanie sztucznego obciążenia DC o niezłych parametrach, wysokich walorach użytkowych oraz niskim koszcie pozyskania, jak i tym osobom, które pragną poszerzyć swoją wiedzę konstruktorską w zakresie własności źródeł energii elektrycznej i przyrządów przeznaczonych do ich pomiarów. Więcej informacji o dalszych losach tego projektu można będzie znaleźć na stronie internetowej autora pod adresem <http://sq5rwq.pl/?p=1093>.

Adam Sobczyk SQ5RWQ
sq5rwq@gmail.com
<http://sq5rwq.pl>



Moduł Espressif ESP32-DevKitC dla czytelników Elektroniki Praktycznej

ESP32-DevKitC to minimalistyczny zestaw rozwojowy, który umożliwia rozwinięcie potencjału modułu Wi-Fi/Bluetooth ESP-WROOM-32. Dzięki zastosowaniu interfejsu USB, w szybki i prosty sposób można dostosować działanie zestawu do swoich możliwości oraz potrzeb. Ponadto, produkt ten obsługuje USB-UART bridge, ma przyciski reset i boot-mode oraz regulator LDO. Każdy niezbędny port GPIO został oddany do dyspozycji użytkownika.

Dzięki uprzejmości firmy Soyter Components, Elektronika Praktyczna ma do zaoferowania czytelnikom 10 egzemplarzy modułu Espressif ESP32-DevKitC. Osoby zainteresowane otrzymaniem modułu na własność, proszone są o zgłoszenie swojego zainteresowania drogą mailową, na adres naszego Szefa Pracowni Konstrukcyjnej: Grzegorz.becker@ep.com.pl. Prosimy o podanie planu zastosowania próbek. Spośród zgłoszeń wybierzemy te najciekawsze. Wartość każdej płytki to około 55 zł.

