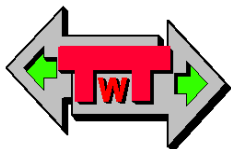
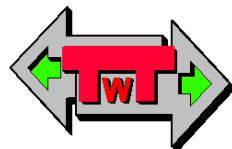


**ZESPÓŁ LABORATORIÓW TELEMATYKI TRANSPORTU
ZAKŁAD TELEKOMUNIKACJI W TRANSPORCIE**



**WYDZIAŁ TRANSPORTU
POLITECHNIKI WARSZAWSKIEJ**



**LABORATORIUM
SYSTEMÓW TELETRANSMISYJNYCH**

INSTRUKCJA DO ĆWICZENIA NR 4

Mikrokontrolery I

© TwT WT PW, DO UŻYTKU WEWNĘTRZNEGO

Warszawa 2016

1. Cel i zakres ćwiczenia

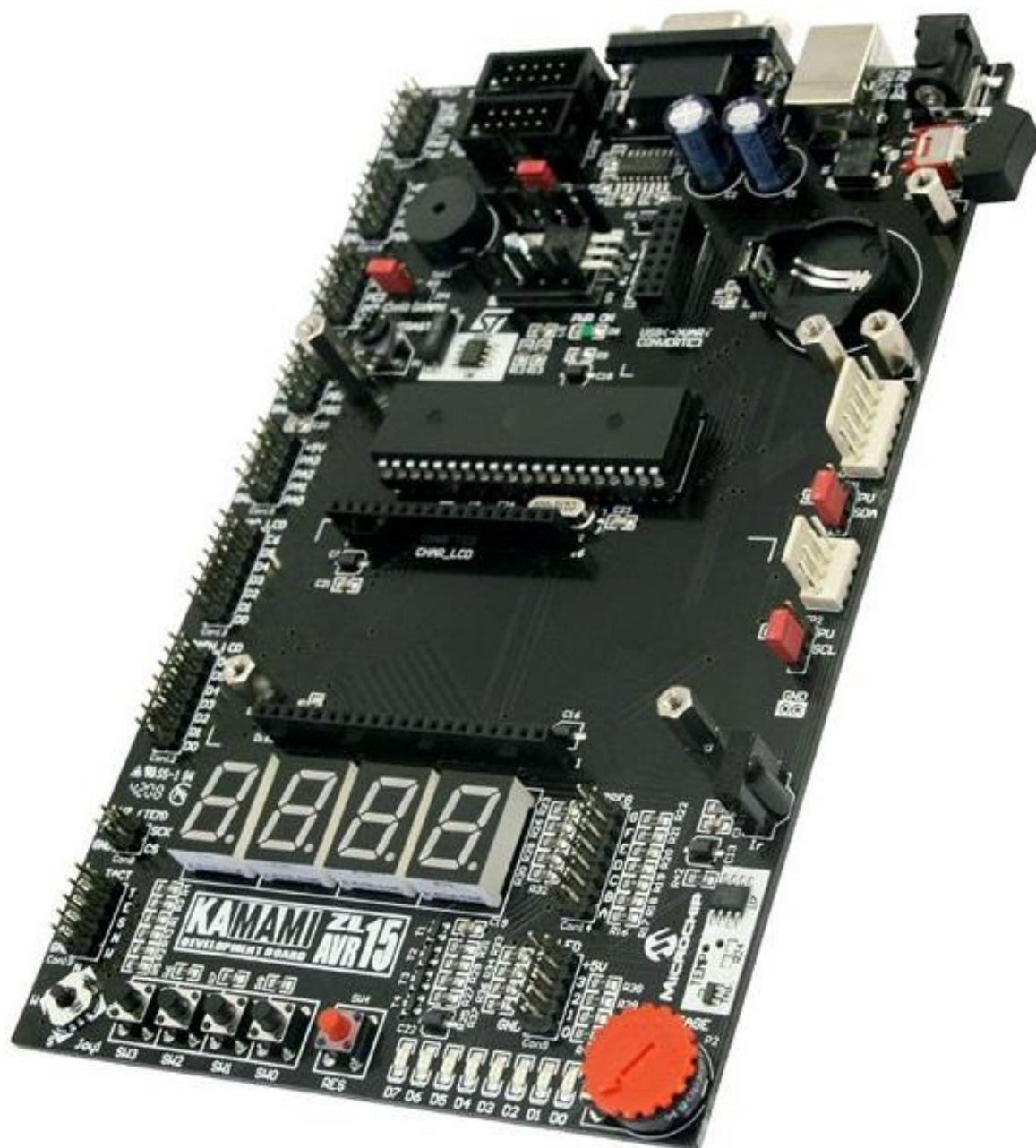
Celem ćwiczenia jest zapoznanie z fizyczną budową układu mikrokontrolera AVR ATmega32, identyfikacja jego wyprowadzeń, pomiar podstawowych właściwości elektrycznych jego układu zasilania oraz zapoznanie się z kompilatorem BASCOM i obsługą programatora ISP dla mikrokontrolerów AVR, zgodnego z STK500 v2.

Zakres ćwiczenia obejmuje identyfikację, obserwację, pomiar oraz rejestrację takich zagadnień jak:

- taktowanie mikrokontrolera,
- reset mikrokontrolera,
- zerowanie pamięci programu mikrokontrolera,
- zakres poziomów napięć zasilania i logicznych,
- programowanie mikrokontrolera,
- tworzenie i kompilacja programu,
- układy wejścia/wyjścia,
- porty mikrokontrolera,
- obsługa elementów sygnalizacyjnych (buzzer, led),
- obsługa wyświetlacza LED i LCD,
- obsługa pamięci wewnętrznej i zewnętrznej,
- obsługa klawiatury matrycowej,
- obsługa łączy RS232C oraz I2C,

2. Wykaz wykorzystanych przyrządów i oprogramowania

- zestaw uruchomieniowy ZL15AVR dla mikrokontrolerów ATmega32 (rys. 2.1),
- programator ISP ZL22PRG dla mikrokontrolerów AVR (rys. 2.2),
- multimetr elektroniczny (rys. 2.3),
- komputerowa przystawka oscyloskopowa PCSU1000 (rys. 2.4),
- komputer PC z systemem Windows 98/NT/2000/XP,
- kompilator języka Basic BASCOM AVR (rys. 2.5),
- program narzędziowy dla obsługi portów RS232C ComTest (rys. 2.6).



Rys. 2.1 Zestaw uruchomieniowy ZL15AVR dla mikrokontrolerów ATmega32



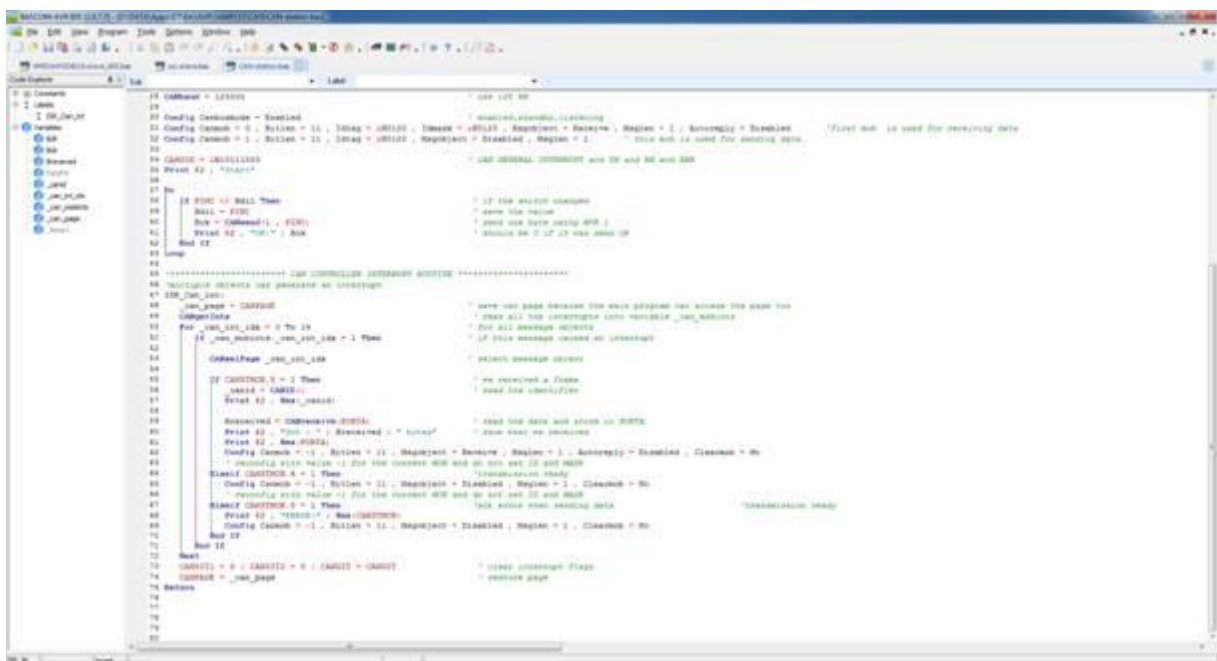
Rys. 2.2 Programator ISP ZL22PRG dla mikrokontrolerów AVR



Rys. 2.3 Multimetr elektroniczny



Rys. 2.4 Komputerowa przystawka oscyloskopowa PCSU1000



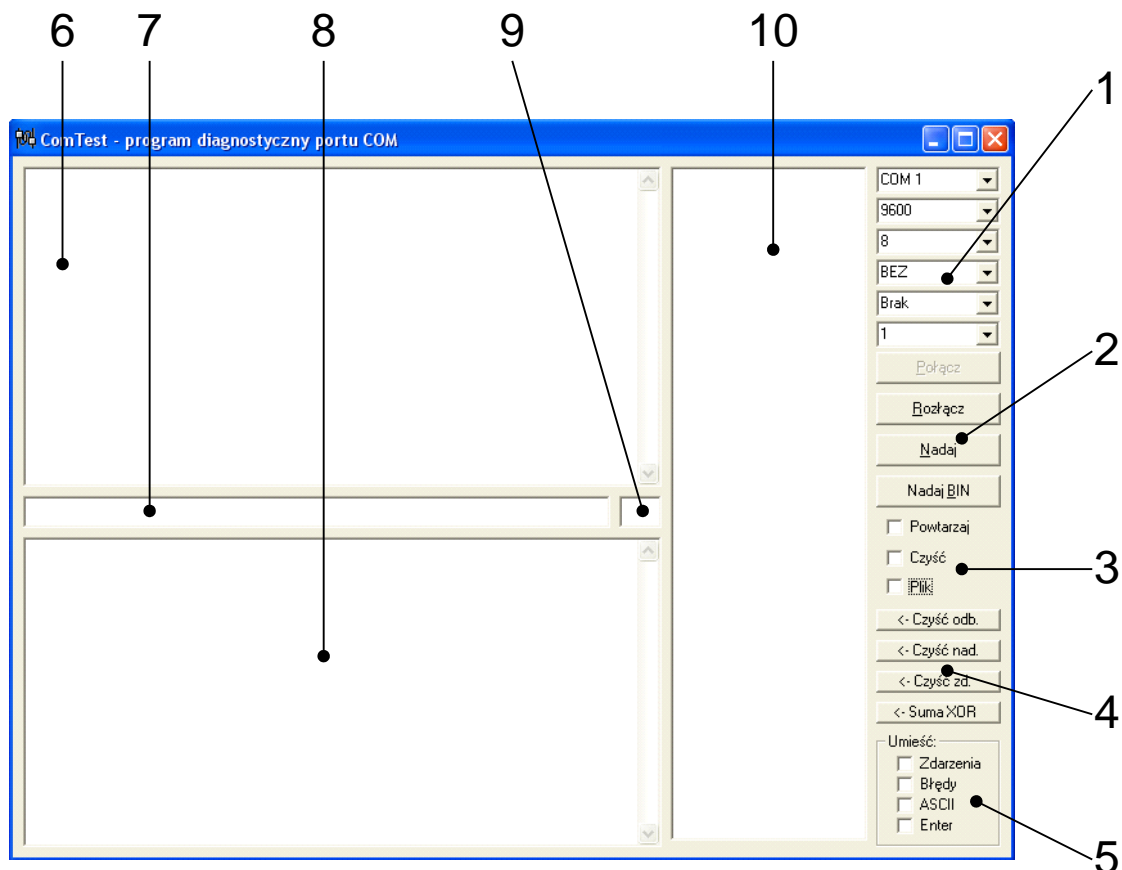
Rys. 2.5 Kompilator języka Basic BASCOM AVR

Do pobrania:

http://www.mcselec.com/index.php?option=com_docman&task=doc_download&gid=139&Itemid=54

Opis języka:

http://www.mcselec.com/index.php?option=com_docman&task=doc_download&gid=140&Itemid=54



Rys. 2.6 Program diagnostyczny i obsługi portów RS232C ComTest

3. Przygotowanie do zajęć

W trakcie wykonywania wszystkich ćwiczeń / projektów wymagane jest uruchomienie tylko jednej aplikacji:

- BASCOM – kompilator języka BASIC.

Jako aplikacje dodatkowe uruchamiane są w określonych sytuacjach:

- PCS2000 – program obsługi przystawki oscyloskopowej,
- COMTEST – program narzędziowy obsługi portów COM – RS232C.

3.1 Kompilator BASCOM

Programowanie w języku Bascom jest stosunkowo łatwe, gdyż standardowo zawiera on bogatą bibliotekę funkcji do obsługi modułów peryferyjnych wbudowanych w mikrokontroler (m.in. przetwornik AC, łatwa konfiguracja i sterowanie linii portów we/wy, obsługa transmisji szeregowej, obsługa pamięci EEPROM).

Największą jednak zaletą Bascom AVR jest biblioteka funkcji zapewniających obsługę najczęściej wykorzystywanych układów w systemach mikroprocesorowych, np. znakowych i graficznych wyświetlaczy LCD, układów z interfejsami SPI, I2C, UART i 1-Wire (obsługa programowa i sprzętowa), klawiatury PC, klawiatury matrycowej. Wśród zaawansowanych funkcji warto zwrócić uwagę na obsługę stosu TCP/IP z wykorzystaniem układu W3100A.

Bascom AVR to także przede wszystkim całe zintegrowane środowisko programistyczne (IDE) przeznaczone do programowania mikrokontrolerów z rodziny AVR w języku programowania o tej samej nazwie. Język ten jest zbliżony do języka Basic.

Obsługa kompilatora sprowadza się do trzech kroków:

1. Napisanie programu. W standardowym oknie tekstowym, typowym dla wszystkich programów Windows. Zapis i odczyt pliku .bas zawierającego program jest również taki sam, jak w każdym innym programie.
2. Kompilacji programu, w wyniku której powstaje plik gotowy do wgrania do układu mikrokontrolera – klawisz F4.
3. Zaprogramowania mikrokontrolera – klawisz F7.

3.2 Program obsługi programatora ISP ZL22PRG

Służy on (w najprostszym zastosowaniu) do wpisania do pamięci mikrokontrolera napisanego i skompilowanego programu. Nie jest to odrębne urządzenie, ale moduł (płytką) dołączona do zestawu uruchomieniowego ZL15AVR, która dokonuje programowania układu poprzez standard ISP. ZL22PRG jest bowiem programatorem ISP dla procesorów całej rodziny AVR firmy Atmel. Zgodność programatora z protokołem STK500 v2 zapewnia bezproblemową współpracę ze środowiskiem AVRStudio firmy Atmel, Bascom AVR oraz z wieloma popularnymi i bezpłatnymi programami (np. AVR Dude). Do poprawnej pracy programatora konieczne jest zainstalowanie sterowników obsługujących wirtualny port szeregowy (Virtual COM Port) dla układu FT232. W zależności od używanego systemu operacyjnego odpowiednie sterowniki należy pobrać ze strony <http://www.ftdichip.com> z

działu Drivers. Instalacja sterowników przebiega w sposób standardowy dla używanego systemu operacyjnego. Może też w ogóle nie być potrzebna, kiedy nowszy system Windows pobierze sterowniki automatycznie.

*UWAGA !!! Programowanie przeprowadza się przy wyłączonym zasilaniu
zestawu uruchomieniowego ZL15 AVR!*

Pełny proces programowania układu zawiera się w czterech krokach:

1. Skasowanie zawartości pamięci układu.
2. Sprawdzenie, czy pamięć układu jest „czysty”.
3. Wgranie programu do pamięci układu.
4. Weryfikacja, czy program został poprawnie wgrany.

3.3 Zestaw uruchomieniowy ZL15 AVR

Zestaw ZL15AVR dzięki wyposażeniu w bogaty zestaw typowych układów peryferyjnych stanowi środowisko do uruchamiania prototypów układów budowanych w oparciu o mikrokontrolery ATmega32 (oraz inne w obudowie 40-wyprowadzeniowej). Może być również wykorzystywany jako zestaw edukacyjny podczas nauki programowania mikrokontrolerów AVR. Podstawowe parametry:

- mikrokontroler ATmega32 w obudowie DIP osadzony w podstawce,
- 4-przyciskowa klawiatura,
- 5-pozycyjny joystick,
- 8 diod LED,
- czterocyfrowy wyświetlacz siedmiosegmentowy LED,
- złącze dla alfanumerycznego wyświetlacza LCD 2x16 znaków (LCD1602),
- złącze dla graficznego wyświetlacza LCD 128x64 pikseli ze sterownikiem KS0108 (LCD12864),
- układ zegara M41T00 (STMicroelectronics) pracujący na magistrali I2C wraz z podstawką na baterię CR2032,
- układ termometru TC77 (Microchip) pracujący na magistrali SPI,
- czujnik temperatury MCP9701 (Microchip),
- odbiornik podczerwieni TSOP31236 (36 kHz),
- przetwornik piezoelektryczny,

- złącze dla modułu USB<->UART (ZL1USB_A, ZL4USB),
- 10-pinowe złącze programowania ISP,
- 10-pinowe złącze JTAG,
- złącze USB przeznaczone do zasilania zestawu z portu USB komputera,
- złącze DB9 wraz z konwerterem napięć MAX232 do realizacji transmisji RS232,
- zestaw złącz szpilkowych z wszystkimi użytecznymi sygnałami mikrokontrolera oraz układów peryferyjnych,
- potencjometr umożliwiający podawanie napięcia z zakresu 0...5 V na wejście przetwornika analogowo-cyfrowego wbudowanego w mikrokontroler.

Ponieważ jest to zestaw uniwersalny, to dostęp do poszczególnych elementów zestawu poprzez mikrokontroler wymaga wcześniejszego dokonania szeregu połączeń przewodowych i taśmowych (płaska grupa przewodów), które są typowe dla danego ćwiczenia. Są one wykonane wcześniej i nie wymagają zmian ani ingerencji studenta.

UWAGA !!! Pod żadnym pozorem nie należy rozłączać zestawionych w zestawie połączeń przewodowych i taśmowych!

4. Program narzędziowy ComTest

Jego zadaniem jest obsługa dostępnych w Windows portów COM (RS232C) komputera laboratoryjnego. Po uruchomieniu programu widoczne jest jego standardowe okno (rys. 2.6), którego organizacja nie ulega zmianie podczas pracy.

- 1 – pole wyboru portu i parametrów transmisji,
- 2 – klawisze poleceń operacji na porcie,
- 3 – pola wyboru opcji operacji na porcie,
- 4 – klawisze operacji na polach tekstowych programu,
- 5 – pola wyboru opcji dodatkowych informacji,
- 6 – pole danych odbieranych,
- 7 – pole danych do wysłania lub nazwa pliku,
- 8 – pole danych nadanych,
- 9 – pole współczynnika skali czasu,
- 10 – pole zdarzeń i komunikatów.

Dostęp do danego portu uzyskuje się po wybraniu jego numeru z pierwszej listy (COM 1, COM 2) oraz jego konfiguracji do której należą kolejno: wybór szybkości pracy (300, ... 9600, ...), ilości bitów danych (5, 6, 7, 8), rodzaj kontroli parzystości (brak, parzysty, nieparzysty), sposobu sterowania przepływem danych (brak, xon/xoff, ...) oraz ilości bitów stopu (1 lub 2). Najczęstsze ustawienie wykorzystywane podczas ćwiczeń to: COM n (gdzie n zależy od zestawionego połączenia Bluetooth), 4800, 8, bez kontroli parzystości, brak sterowania przepływem, 1 bit stopu. Po dokonaniu konfiguracji należy nacisnąć klawisz *POŁĄCZ* i program otwiera dostęp do wybranego portu. W przypadku błędu, np. z powodu zajętości portu przez inną aplikację, program wyświetli odpowiedni komunikat. Zamknięcie portu następuje po naciśnięciu klawisza *ROZŁĄCZ*. W polu nr 6 pokazywane są wszystkie znaki odbierane z otwartego portu COM. Należy jednak pamiętać, że nie wszystkie znaki ASCII będą widziane w czytelnej postaci. Pole nr 7 służy do bezpośredniego wysyłania danych do portu COM lub określenia nazwy pliku do odczytu lub zapisu. W celu wysłania danych należy wpisać w polu nr 7 odpowiednią sekwencję znaków i nacisnąć Enter lub klawisz *NADAJ*. W przypadku zaznaczenia opcji *POWTÓRZ* program będzie „w kółko” nadawał zapisaną sekwencję. Jeżeli umieścimy tam nazwę pliku i zaznaczymy opcję *PLIK* (domyślnym katalogiem jest c:\gps\), to program wyśle do portu zawartość tego pliku. Przy zaznaczonej opcji *POWTÓRZ* wysyłanie będzie realizowane „w kółko”. Jeżeli zaznaczono opcję *CZYŚĆ*, to po wysłaniu sekwencji nastąpi skasowanie zawartości pola nr 8, gdzie umieszczane są wszystkie znaki wysłane do portu COM. Pole nr 10 służy natomiast do wyświetlania zachodzących w porcie COM zdarzeń i informowania o wykrytych błędach transmisji. Możliwe jest to po zaznaczeniu odpowiedniej opcji w polu *UMIEŚĆ*. Zaznaczenie tam opcji *ASCII* spowoduje wyświetlenie w polu nr 6 nie znaków, lecz ich kodów ASCII, dla lepszej czytelności pomiędzy znakami „<” i „>”. Opcja *ENTER* powoduje dodanie do każdej wysyłanej do portu sekwencji znaków kodów nowej linii CR/LF, czyli 13 i 10. Kasowanie zawartości pól nr 6, 7 i 8 jest możliwe dzięki trzem klawiszom *CZYŚĆ*, odpowiednio *ODB.*, *NAD.*, *ZD*. Klawisz *SUMA XOR* służy do obliczenia sumy kontrolnej ramki danych protokołu NMEA, umieszczonej w polu nr 7.

Program może zapisywać odbierane dane bezpośrednio do pliku tekstowego. Podczas operacji na plikach ważna jest jednak kolejność wykonywanych działań:

- przy zapisie do pliku najpierw wpisujemy nazwę pliku w polu nr 7, zaznaczamy opcję *PLIK* i dopiero (po konfiguracji) otwieramy port poleceniem *POŁĄCZ*,
- przy odczycie danych z pliku najpierw odznaczamy opcję *PLIK* i dopiero (po

konfiguracji) otwieramy port poleceniem *POŁĄCZ*, następnie wpisujemy nazwę pliku w polu nr 7 i zaznaczamy opcję *PLIK*.

UWAGA!!! Program można uruchamiać wielokrotnie i obsługiwać kilka portów (urządzeń) jednocześnie. Opis standardu RS232C znajduje się w innych instrukcjach.

5. Wykaz użytych skrótów i oznaczeń

Dla zwiększenia przejrzystości wprowadzono poniższe skróty oraz oznaczenia graficzne, które zostały wykorzystane w tekście instrukcji:



- zapisz przebieg lub zrzut ekranu na dysku,



- pytanie, na które odpowiedź musi znaleźć się w sprawozdaniu,

6. Uwagi praktyczne

Pod żadnym pozorem nie należy „upraszczać” ćwiczenia i próbować wykonywać jego części w sposób łączny lub w zmienionej kolejności. Jest to najszybsza droga do pomyłki w przeprowadzonych pomiarach i obserwacjach, co skutkuje odrzuceniem sprawozdania.

Najlepiej nadawać plikom z programami nazwy zgodne z numerem punktu instrukcji. Nigdy nie należy dokonywać zmian w już działającym poprawnie (dla danego punktu instrukcji) programie. Zawsze po poprawnym zrealizowaniu konkretnego punktu instrukcji zapisujemy plik programu pod nową nazwą, która odpowiada kolejnemu punktowi części wykonawczej instrukcji i dopiero w niej dokonujemy kolejnych modyfikacji.

7. Przebieg ćwiczenia

7.1 Odszukaj na płytce zestawu uruchomieniowego ZL15 AVR wszystkie charakterystyczne bloki funkcjonalne i elementy, wyszczególnione i omówione w opisie zestawu. Zidentyfikuj je zarówno przy pomocy opisów płytki, jak i schematów zestawu, dołączonych do instrukcji. się z budową stanowiska laboratoryjnego.

7.2 Uruchom kompilator Bascom AVR.

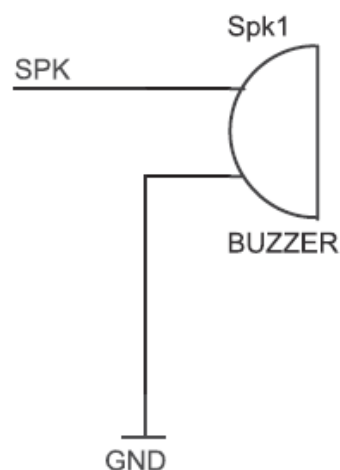
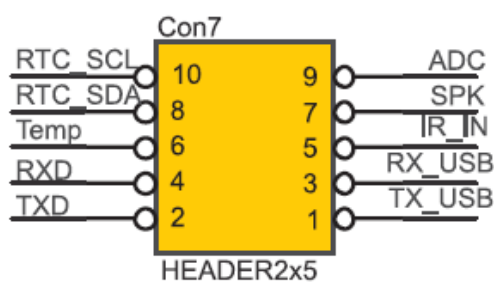
7.3 Wykonaj testowe kasowanie pamięci programu mikrokontrolera. Pamiętaj, że zasilanie zestawu ZL15 AVR ma być wyłączone.

7.4 Włącz zasilanie zestawu ZL15 AVR. Co się dzieje? Co można zaobserwować? 

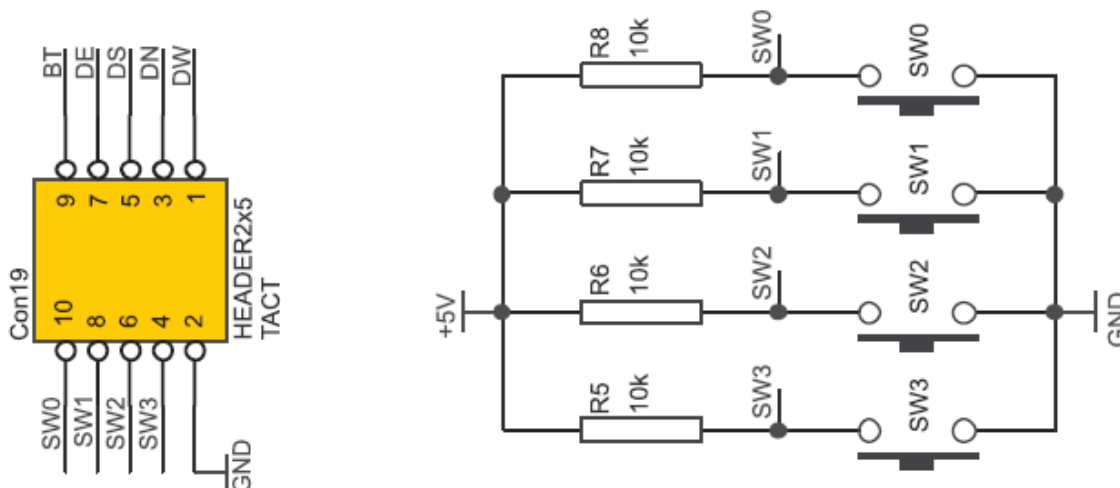
7.5 Załaduj do kompilatora program *deklaracje.bas* – znajduje się również na końcu

instrukcji. UWAGA!!! Program ten zawiera wszystkie niezbędne do wykonania ćwiczenia deklaracje sprzętu (fizycznych wyprowadzeń i połączeń) oraz minimalny zestaw procedur. Ma także zaimplementowaną obsługę przerw, niezbędną do multipleksowej obsługi wyświetlacze 7-segmentowego LED. Część zadań może wymagać ingerencji i zmian wewnątrz tej procedury. Zapisz program w katalogu swojego protokołu, nadając mu własną, początkową nazwę.

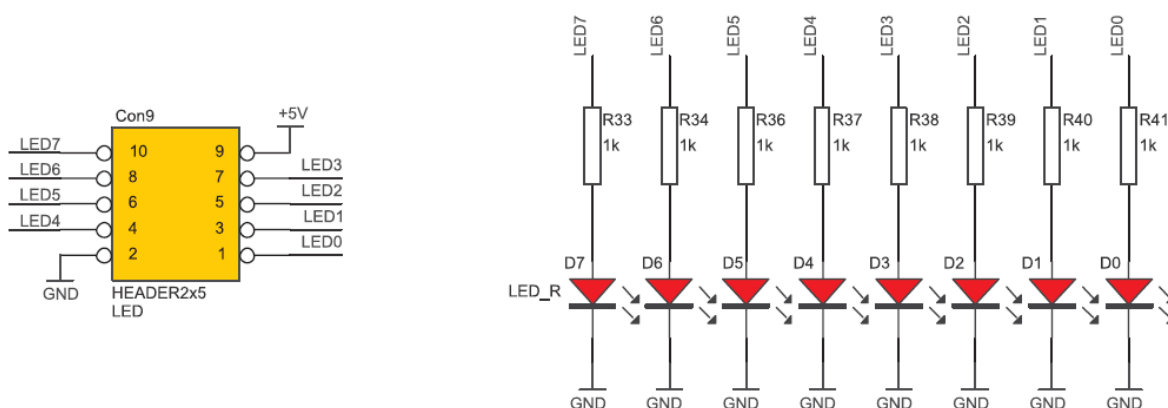
- 7.6 Wykorzystaj program i kompilator BASCOM AVR i napisz swój pierwszy program, którego zadaniem będzie aktywacja sygnału przetwornika piezoelektrycznego Spk1 (tzw. buzzer). Przetwornik piezoelektryczny Spk1 służący do generowania dźwięku. Do generowania dźwięków można wykorzystać jeden z układów licznikowych mikrokontrolera ATmega32 lub też generować dźwięki programowo. Wyprowadzenie przetwornika oznaczone SPK jest dostępne na złączu Con7 i podłączone do linii PinD.7 portu mikrokontrolera D. Użyj w programie nazwy symbolicznej BUZZER oraz instrukcję **Sound**. 🖨



- 7.7 Zestaw ZL15AVR jest wyposażony w układ czterech przycisk SW0...SW3, podłączonych do linii PinD.0, PinD.1, PinD.2, PinD.3 portu D. Każdy z przycisków ma własny rezystor podciągający. Wyprowadzenia przycisków są dostępne na złączu Con8. Zmień program w taki sposób, aby naciśnięcie klawisza SW0 prowadziło do generowania dźwięku przez BUZZER. W programie używaj nazw symbolicznych SW0...SW3. Wykorzystaj do tego instrukcję **Debounce** i wcześniej poznaną instrukcję **Sound** oraz już zadeklarowane w programie procedury Klaw0, Klaw1, Klaw2 i Klaw3. 🖨



7.8 Zestaw ZL15 wyposażono w osiem czerwonych diod LED przeznaczonych do sygnalizacji stanu portów mikrokontrolera (aktywny stan wysoki). Linie sterujące czterema (tutaj tylko czterema) diodami LED (Led0, Led1, Led2, Led3) są wprowadzone na złączu Con9 i podłączone do portu A, odpowiednio do PinA.4, PinA.5, PinA.6, PinA.7. Spraw, aby generowaniu przez BUZZER dźwięku towarzyszyła sygnalizacja (świecenie) diodą Led0, podłączoną do linii PinA.4 a chwilom ciszy świecenie diodą Led1, podłączonej do linii PinA.5. W programie używaj nazw symbolicznych SW0, BUZZER, Led0 i Led1. 🖱️ Świecenie diody jest aktywowane poziomem „1”.



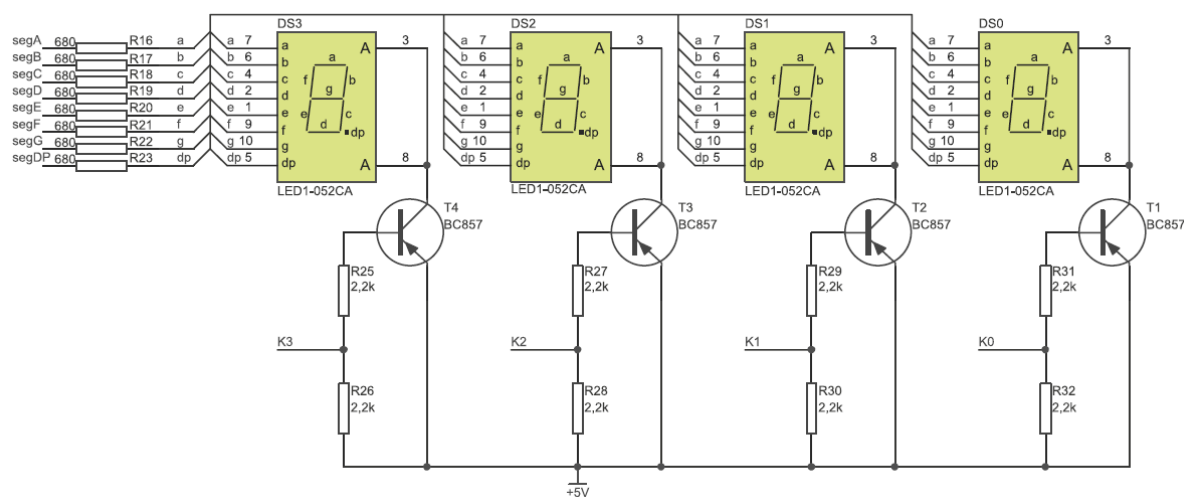
7.9 Zmień poprzedni program w taki sposób, aby nie było potrzebne naciskanie klawisza SW0. Niech program z przerwą 1 sekundy generuje dźwięk i świeci (miga) raz parzystymi a raz nieparzystymi diodami LED. 🖱️

7.10 Zestaw ZL15AVR wyposażono w czterocyfrowy siedmiosegmentowy wyświetlacz LED (DS0...DS3) wraz z tranzystorami sterującymi anody wyświetlaczy. Sygnały sterujące anodami (0...3) oraz poszczególnymi segmentami wyświetlacza zostały wprowadzone na złączu Con14 i podłączone do portu A i wyprowadzeń PinA.0, PinA.1,

PinA.2 i PinA.3. Napisz program, który na wyświetlaczu LED wyświetli dzisiejszą datę – naprzemiennie rok oraz dzień i miesiąc z czasem przerwy 1 sekunda. Musisz wykorzystać do tego istniejące procedury **Wyswietlacz7segmentowy** oraz **Zgaswysled**. Za cyfry wyświetlane na wyświetlaczu odpowiadają zmienne `Wartosc0`, `Wartosc1`, `Wartosc2`, `Wartosc3`. Za wygląd cyfr na wyświetlaczu odpowiadają wartości dziesiętne odczytywane instrukcją **Data**. Ostatnia z nich (szesnasta) wygasza wyświetlacz. Do podstawienia konkretnej wartości pod zmienną `Wartosc` wykorzystaj instrukcję **Lookup** 🖱️

7.11 Napisz program, który po uruchomieniu na wyświetlaczu LED wyświetli cyfry od 9 do 0 z czasem przerwy 1 sekunda. Po osiągnięciu 0 niech program wygeneruje BUZZEREM 3-sekundowy dźwięk i zakończy działanie. 🖱️

7.12 Rozbuduj poprzedni program w taki sposób, by zmiana wartości na wyświetlaczu była realizowana od wartości 30 do 0. 🖱️



8. Wykonanie sprawozdania

Nie należy umieszczać w sprawozdaniu podstaw teoretycznych, ani opisów stanowiska laboratoryjnego. Wnioski powinny zawierać omówienie zaznaczonych fragmentów programu – jaką realizują funkcję, co poprzedza wywołanie tej funkcji, jakie są uwarunkowania, jakie powiązania z innymi funkcjami (procedurami, częściami programu).

Do sprawozdania należy dołączyć stworzone w BASICu programy.

9. Literatura

- Pełka Ryszard, *Mikrokontrolery – architektura, programowanie, zastosowania*, WKŁ 1999,
- Dąca Wieńczysław, *Mikrokontrolery od układów 8-bitowych do układów 32-bitowych*, Mikom 2000,
- Wiązania Marcin, *Programowanie mikrokontrolerów AVR w języku Bascom*, BTC 2004,
- Wiązania Marcin, *Bascom AVR w przykładach*, BTC 2008,
- Chromik Rafał, *RS232 w przykładach na PC i AVR*, BTC 2010.

OBOWIĄZKOWE

- Kurs *Bascom*
http://www.mcselec.com/index.php?option=com_docman&task=doc_download&gid=140&Itemid=54 .

Program deklaracje.bas.

```
$regfile = "m32def.dat"
```

```
$crystal = 16000000
```

```
$baud = 1200
```

```
$hwstack = 64
```

```
$swstack = 64
```

```
$framesize = 64
```

```
Dim Ktorawartosc As Byte
```

```
Dim Wartosc0 As Byte
```

```
Dim Wartosc1 As Byte
```

```
Dim Wartosc2 As Byte
```

```
Dim Wartosc3 As Byte
```

```
'Deklaracja procedury wygaszenia wyświetlacza LED
```

```
Declare Sub Zgaswysled
```

```
'Deklaracja obsługi przerw
```

```
Config Timer0 = Timer , Prescale = 256
```

```
On Timer0 Wyswietlacz7segmentowy
```

```
'Aktywacja przerw
```

```
Enable Interrupts
```

```
Enable Timer0
```

```
'Konfiguracja portów
```

```
Config Porta = Output
```

```
Config Portb = Output
```

```
Config Portc = Input
```

```
Config Pinc.7 = Output
```

```
Config Portd = Output
```

```
'Wstępne ustawienia wartości portów
```

```
Porta = 0
```

```
Portb = 255
```

```
Portc = 0
```

```
Portd = 0
```

```
'Nadanie nazw poszczególnym liniom portów
```

```
Buzzer Alias Pind.7
```

```
Sw0 Alias Pind.0
```

```
Sw1 Alias Pind.1
```


Sw2 Alias Pind.2

Sw3 Alias Pind.3

Led0 Alias Pina.4

Led1 Alias Pina.5

Led2 Alias Pina.6

Led3 Alias Pina.7

'Wygazzenie wyświetlaczy LED

Call Zgaswysled

Do

'Zasadnicza część programu

Loop

End

'Procedura obsługi przycisku SW0

Klaw0:

Return

'Procedura obsługi przycisku SW1

Klaw1:

Return

'Procedura obsługi przycisku SW2

Klaw2:

Return

'Procedura obsługi przycisku SW3

Klaw3:

Return

'Wygazzenie wyswietlacza LED

Sub Zgaswysled

 Wartosc0 = Lookup(16 , Cyfry)

 Wartosc1 = Lookup(16 , Cyfry)

 Wartosc2 = Lookup(16 , Cyfry)

 Wartosc3 = Lookup(16 , Cyfry)

End Sub Zgaswysled

'Obsługa przerwania dla Timer0
Wyswietlacz7segmentowy:

Incr Ktorawartosc

Select Case Ktorawartosc

Case 1 :

Set Porta.3
Portb = Wartosc0
Reset Porta.0

Case 2 :

Set Porta.0
Portb = Wartosc1
Reset Porta.1

Case 3 :

Set Porta.1
Portb = Wartosc2
Reset Porta.2

Case 4 :

Set Porta.2
Portb = Wartosc3
Reset Porta.3
Ktorawartosc = 0

End Select

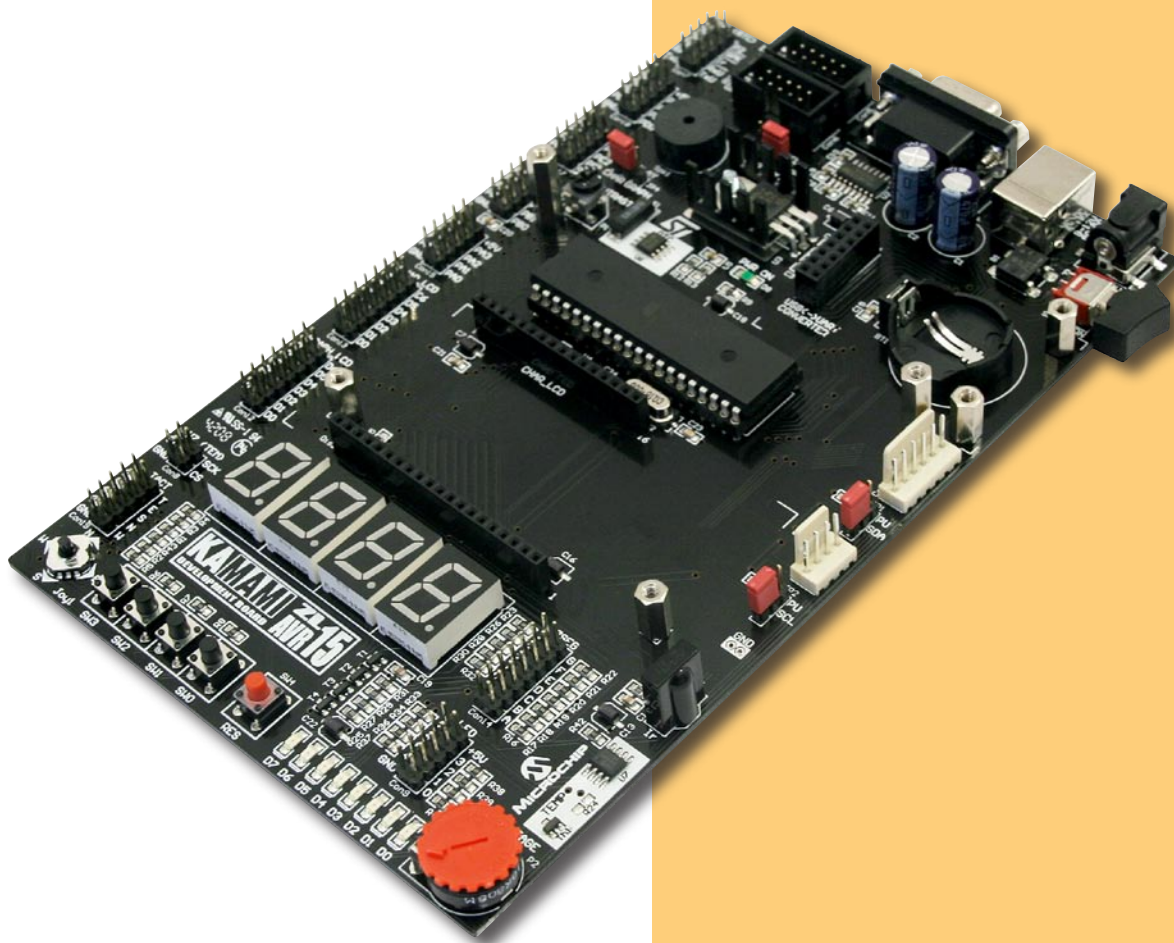
Return

'Definicja cyfr na wyświetlaczy 7-segmentowym: 0,1,2,3,4,5,6,7,8,9, zgaś

Cyfry:

Data 192 , 249 , 164 , 176 , 153 , 146 , 130 , 248 , 128 , 144 , 136 , 131 , 198 , 161 , 134 , 142
, 255

Zestaw uruchomieniowy dla mikrokontrolerów ATmega32



ZL15AVR jest uniwersalnym zestawem uruchomieniowym dla mikrokontrolerów ATmega32 (oraz innych w obudowie 40-wyprowadzeniowej). Dzięki wyposażeniu w szeroką gamę peryferiów może być stosowany jako środowisko do opracowywania prototypów urządzeń lub też jako zestaw edukacyjny.

Wprowadzenie

Zestaw ZL15AVR dzięki wyposażeniu w bogaty zestaw typowych układów peryferyjnych stanowi środowisko do uruchamiania prototypów układów budowanych w oparciu o mikrokontrolery ATmega32 (oraz inne w obudowie 40-wyprowadzeniowej). Może być również wykorzystywany jako zestaw edukacyjny podczas nauki programowania mikrokontrolerów AVR.

Podstawowe parametry

- ▶ mikrokontroler ATmega32 w obudowie DIP osadzony w podstawce
- ▶ 4-przyciskowa klawiatura
- ▶ 5-pozycyjny joystick
- ▶ 8 diod LED
- ▶ czterocyfrowy wyświetlacz siedmiosegmentowy LED
- ▶ złącze dla alfanumerycznego wyświetlacza LCD 2x16 znaków (LCD1602)
- ▶ złącze dla graficznego wyświetlacza LCD 128x64 pikseli ze sterownikiem KS0108 (LCD12864)
- ▶ układ zegara M41T00 (STMicroelectronics) pracujący na magistrali I²C wraz z podstawką na baterię CR2032
- ▶ układ termometru TC77 (Microchip) pracujący na magistrali SPI
- ▶ czujnik temperatury MCP9701 (Microchip)
- ▶ odbiornik podczerwieni TSOP31236 (36 kHz)
- ▶ przetwornik piezoelektryczny
- ▶ złącze dla modułu USB<->UART (ZL1USB_A, ZL4USB)
- ▶ 10-pinowe złącze programowania ISP
- ▶ 10-pinowe złącze JTAG
- ▶ złącze USB przeznaczone do zasilania zestawu z portu USB komputera.
- ▶ złącze DB9 wraz z konwerterem napięć MAX232 do realizacji transmisji RS232
- ▶ zestaw złącz szpilkowych z wszystkimi użytecznymi sygnałami mikrokontrolera oraz układów peryferyjnych
- ▶ potencjometr umożliwiający podawanie napięcia z zakresu 0...5 V na wejście przetwornika analogowo-cyfrowego wbudowanego w mikrokontroler

Wyposażenie standardowe

Kod	Opis
ZL15AVR	▶ Zmontowana płytką zestawu ZL15AVR z mikrokontrolerem ATmega32



BTC Korporacja
 05-120 Legionowo
 ul. Lwowska 5
 tel.: (22) 767-36-20
 faks: (22) 767-36-33
 e-mail: biuro@kamami.pl
<http://www.kamami.pl>

Zastrzegamy prawo do wprowadzania zmian bez uprzedzenia.

Oferowane przez nas płytki drukowane mogą się różnić od prezentowanej w dokumentacji, przy czym zmianom nie ulegają jej właściwości użytkowe.

BTC Korporacja gwarantuje zgodność produktu ze specyfikacją.

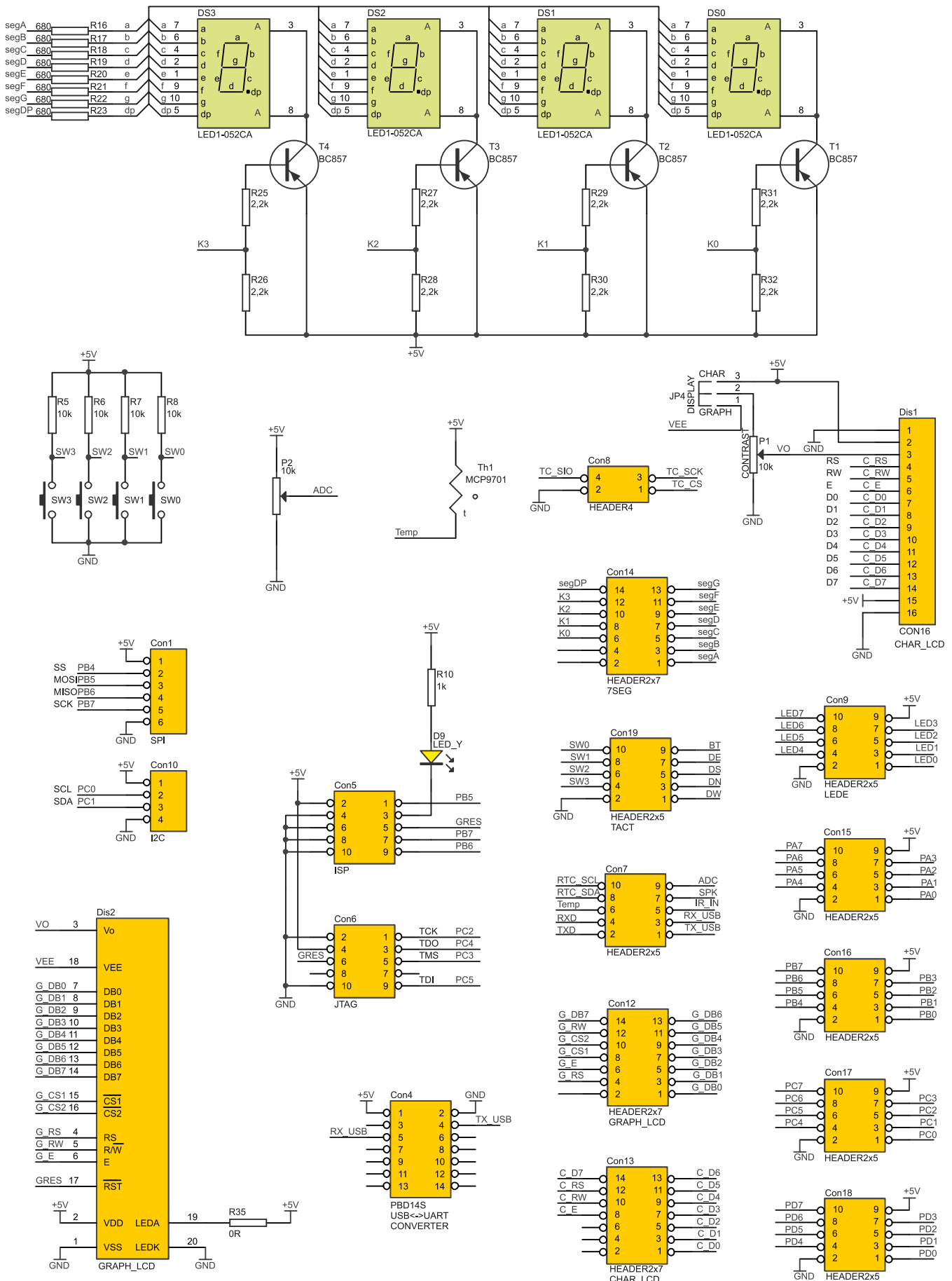
BTC Korporacja nie ponosi odpowiedzialności za jakiegokolwiek szkody powstałe bezpośrednio lub pośrednio w wyniku użycia lub nieprawidłowego działania produktu.

BTC Korporacja zastrzega sobie prawo do modyfikacji niniejszej dokumentacji bez uprzedzenia.

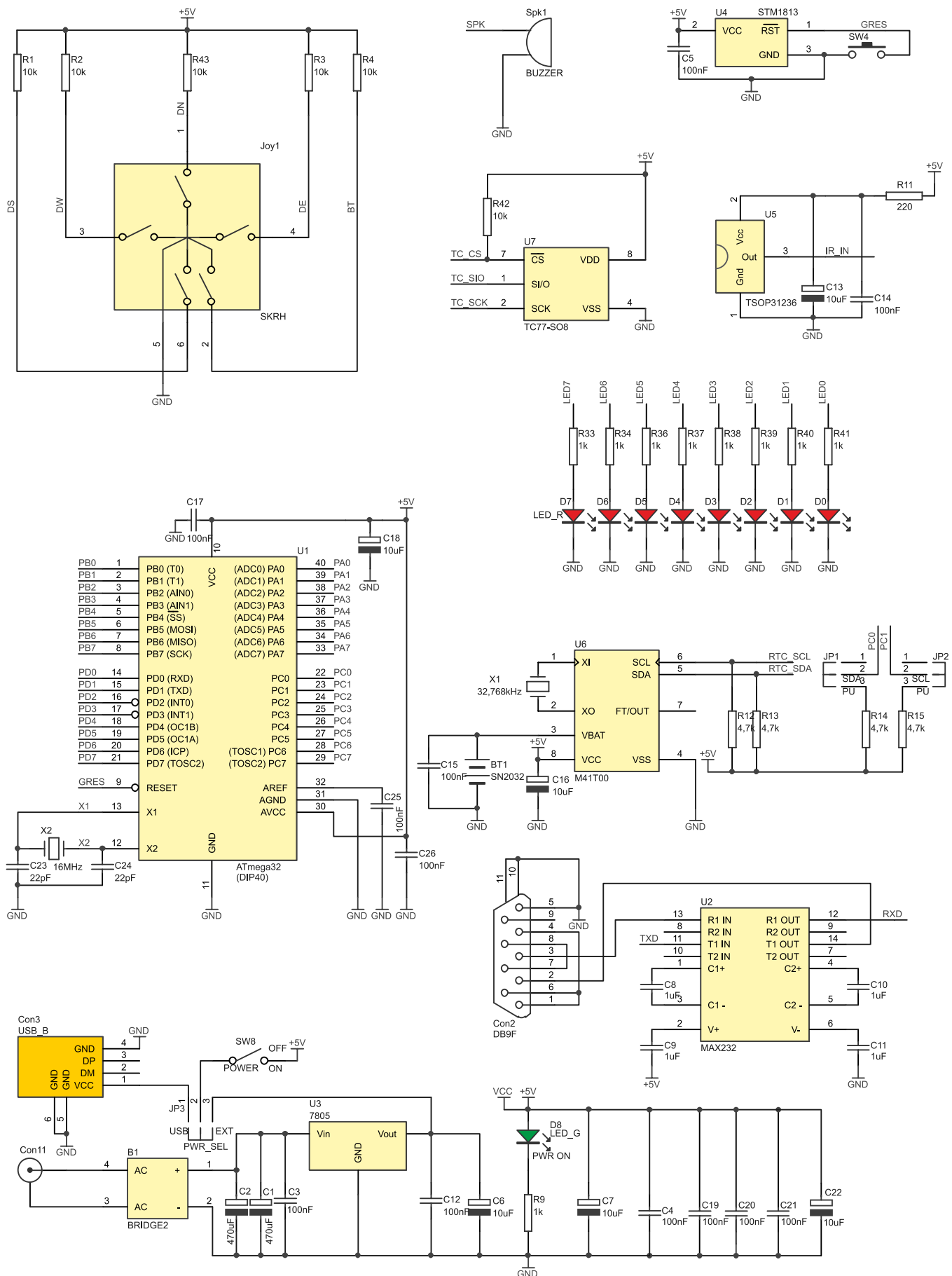
Spis treści

▶ Schemat elektryczny	4
▶ Zasilanie	6
▶ Klawiatura	7
▶ Joystick.....	6
▶ Diody LED.....	5
▶ Wyświetlacz LED	10
▶ Potencjometr analogowy.....	11
▶ Termometr cyfrowy TC77	12
▶ Termistor	13
▶ Odbiornik podczerwieni.....	14
▶ Złącze graficznego wyświetlacza LCD	15
▶ Złącze alfanumerycznego wyświetlacza LCD	16
▶ Złącze magistrali I ² C	17
▶ Złącze magistrali SPI.....	18
▶ Złącze konwertera USB<->UART	19
▶ Zegar czasu rzeczywistego M41T00	20
▶ Przetwornik piezoelektryczny.....	21
▶ Złącze JTAG i ISP	22
▶ Złącze RS232	23
▶ Wyprowadzenia portów wejścia/wyjścia.....	24

Schemat elektryczny

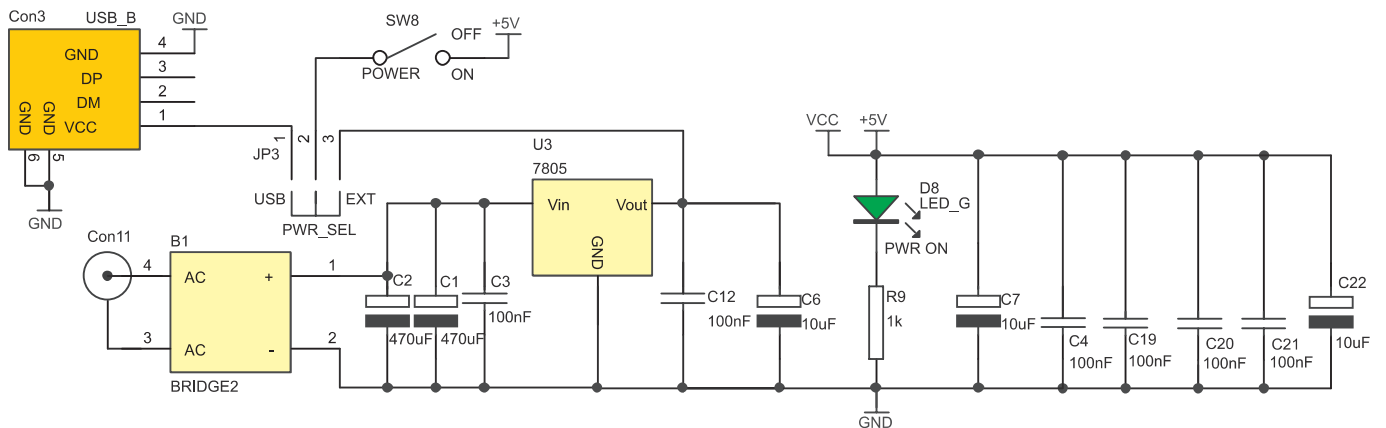
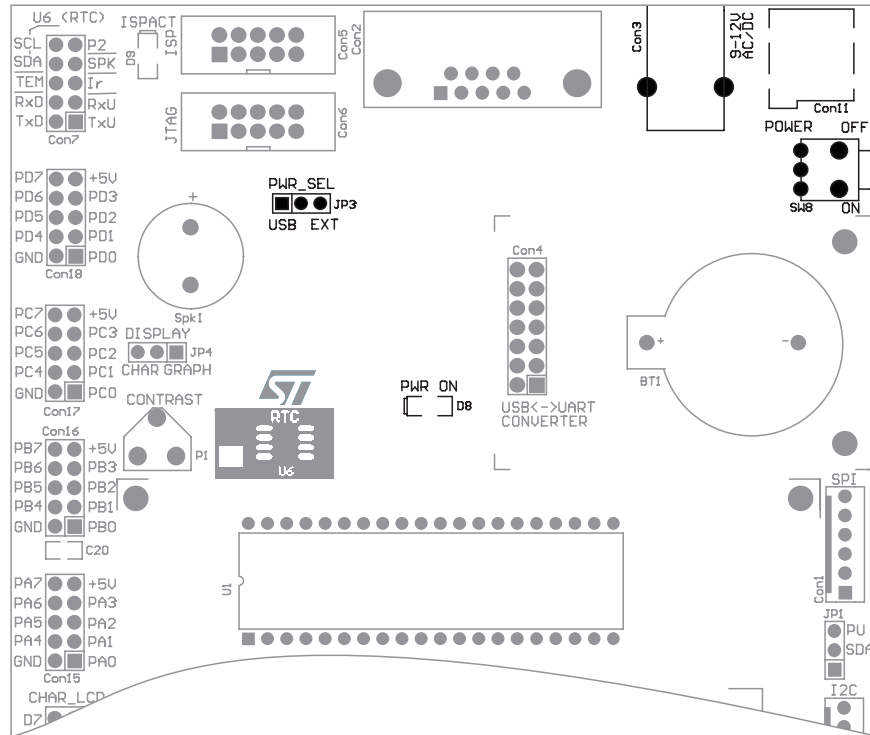


ZL15AVR – zestaw uruchomieniowy dla mikrokontrolerów ATmega32



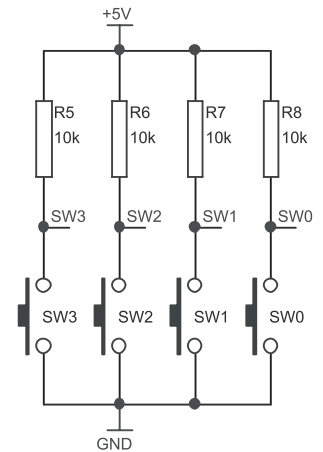
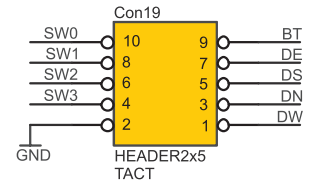
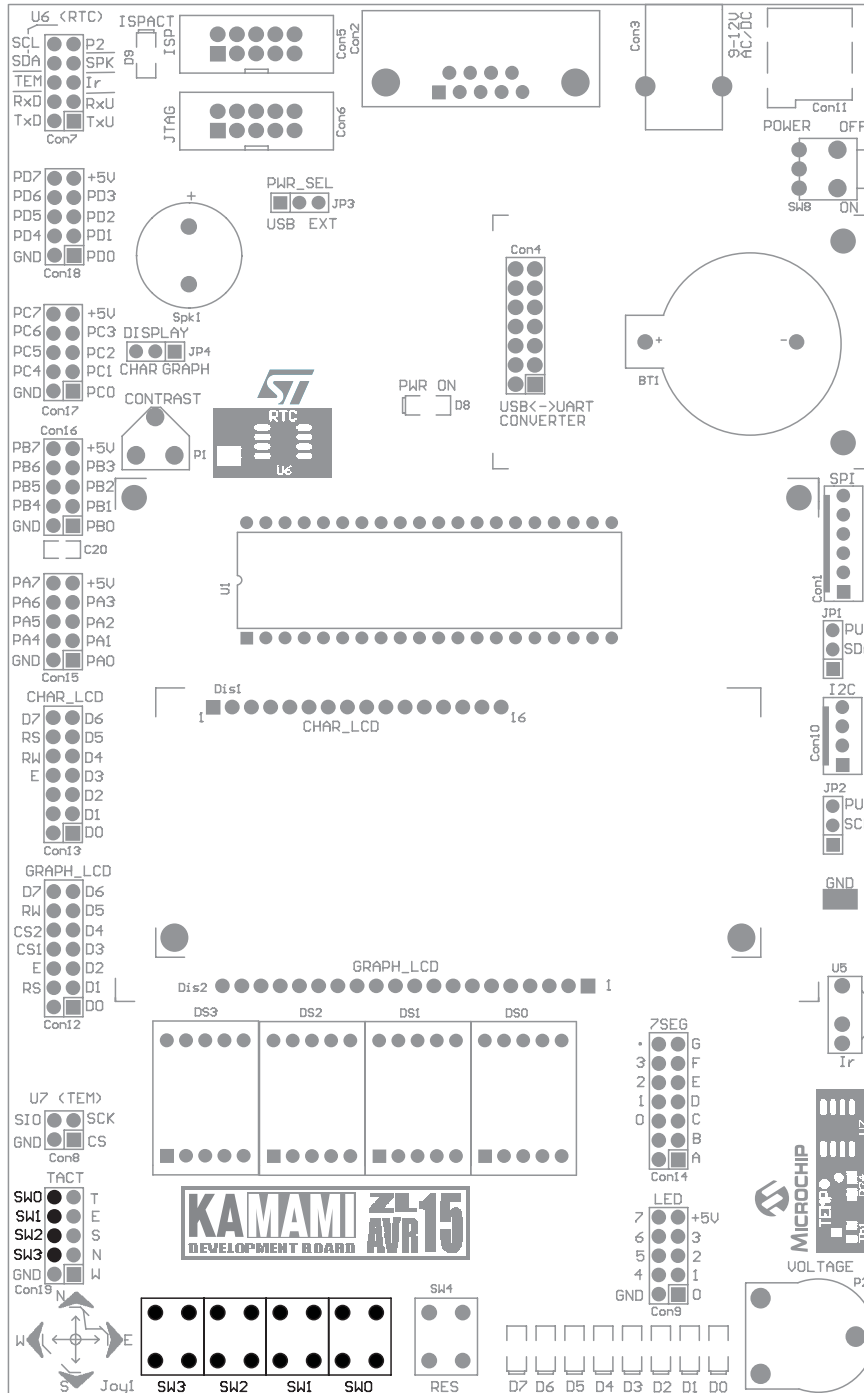
Zasilanie

Zestaw ZL15AVR może być zasilany z portu USB komputera (złącze USB-B Con3, zworka PWR_SEL w pozycji USB) lub z zewnętrznego źródła napięcia o wartości 9...12 V (złącze DC Con11, zworka PWR_SEL w pozycji EXT). Zestaw jest wyposażony w przełącznik SW8 umożliwiający wyłączenie zasilania zestawu bez konieczności odłączania wtyczki ze złącza Con3 lub Con11. Dioda LED D8 sygnalizuje włączenie napięcia zasilania.



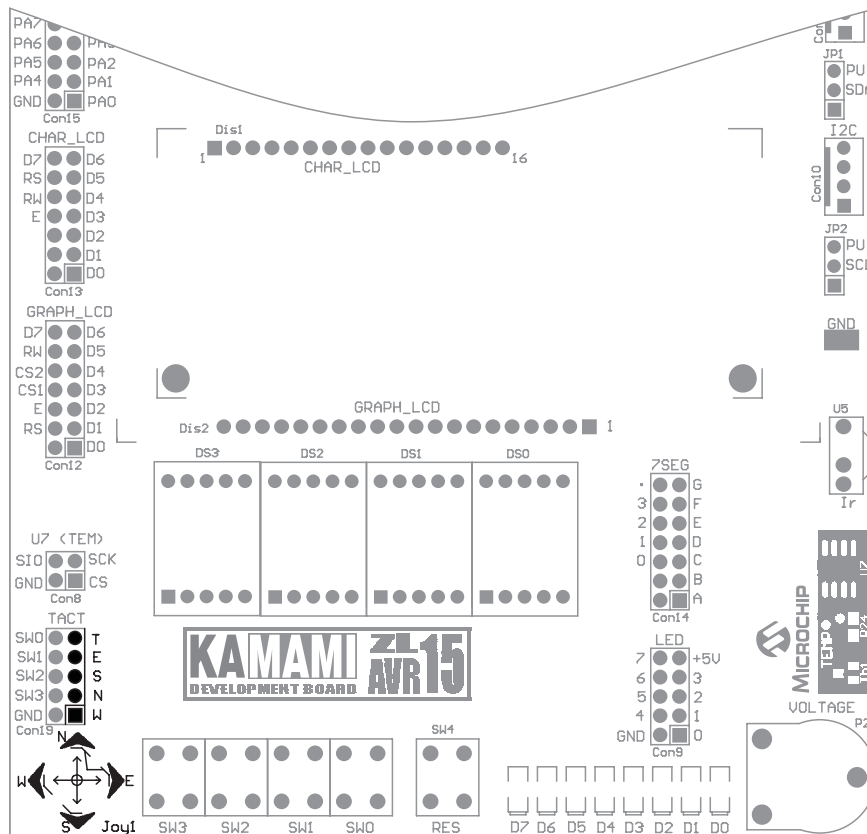
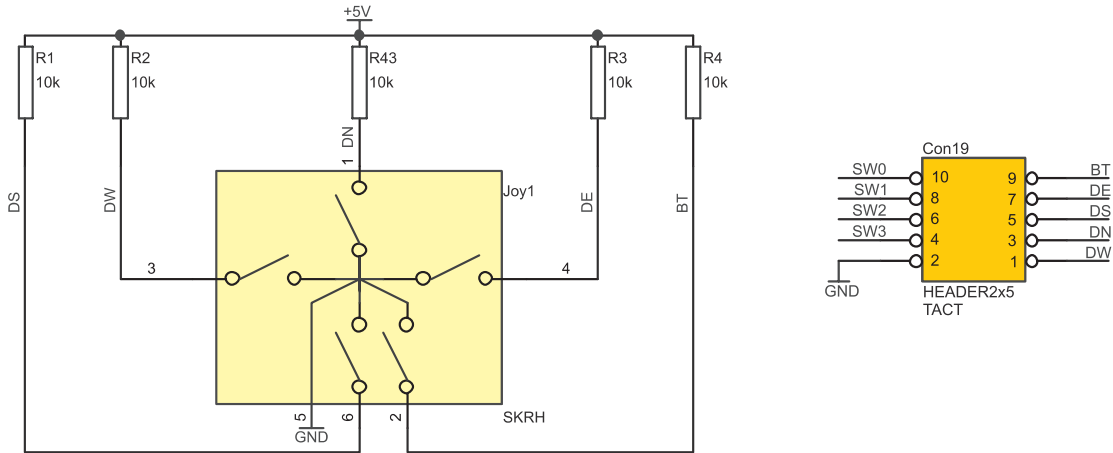
Klawiatura

Zestaw ZL15AVR jest wyposażony w 4-przyciskową klawiaturę składającą się z przycisków SW0...SW3. Każdy z przycisków ma własny rezystor podciągający. Wyprowadzenia przycisków są dostępne na złączu Con8.



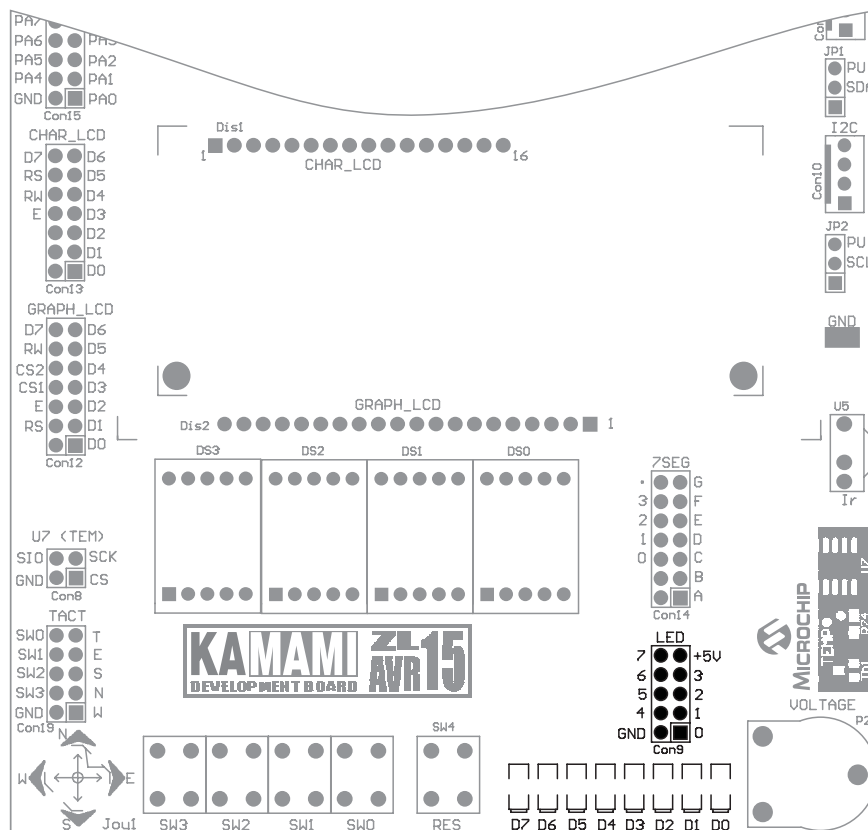
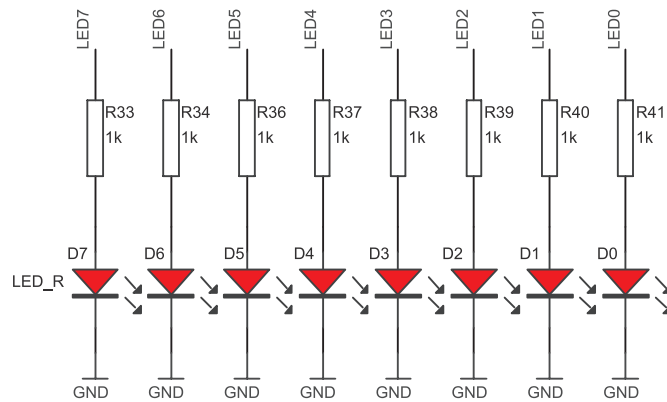
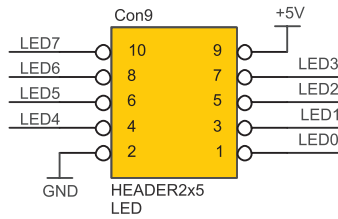
Joystick

Zestaw ZL15AVR został wyposażony w 5-pozycyjny joystick. Każdy ze styków joysticka ma rezystor podciągający do napięcia zasilania. Wszystkie linie joysticka są dostępne na złączu Con19.



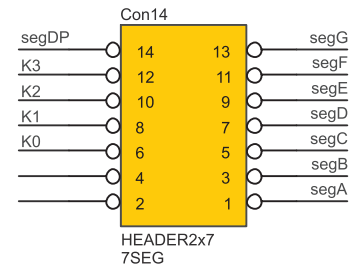
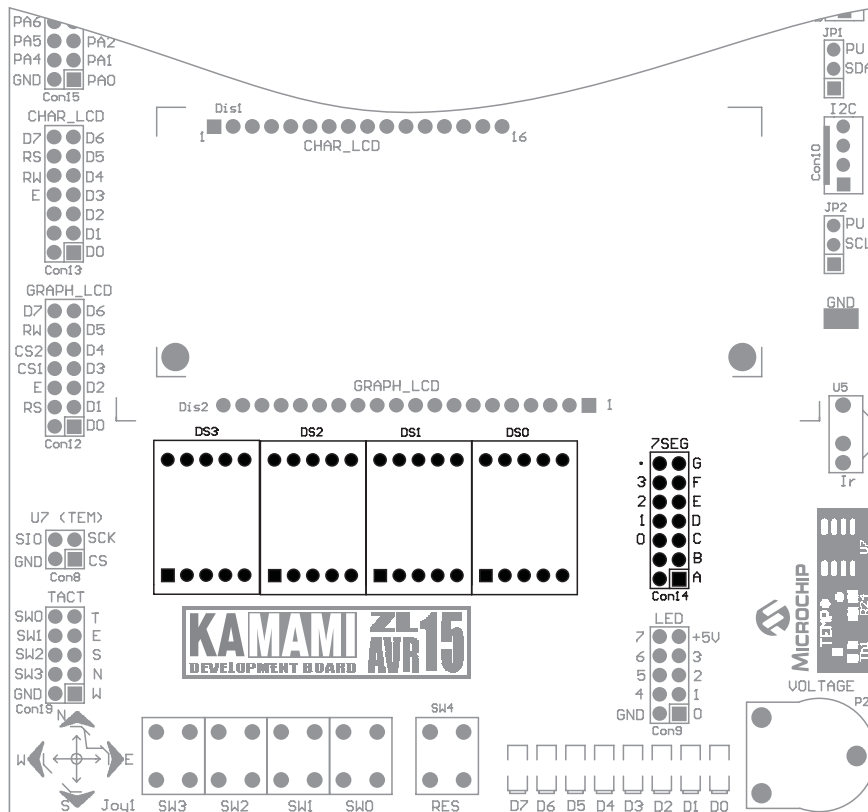
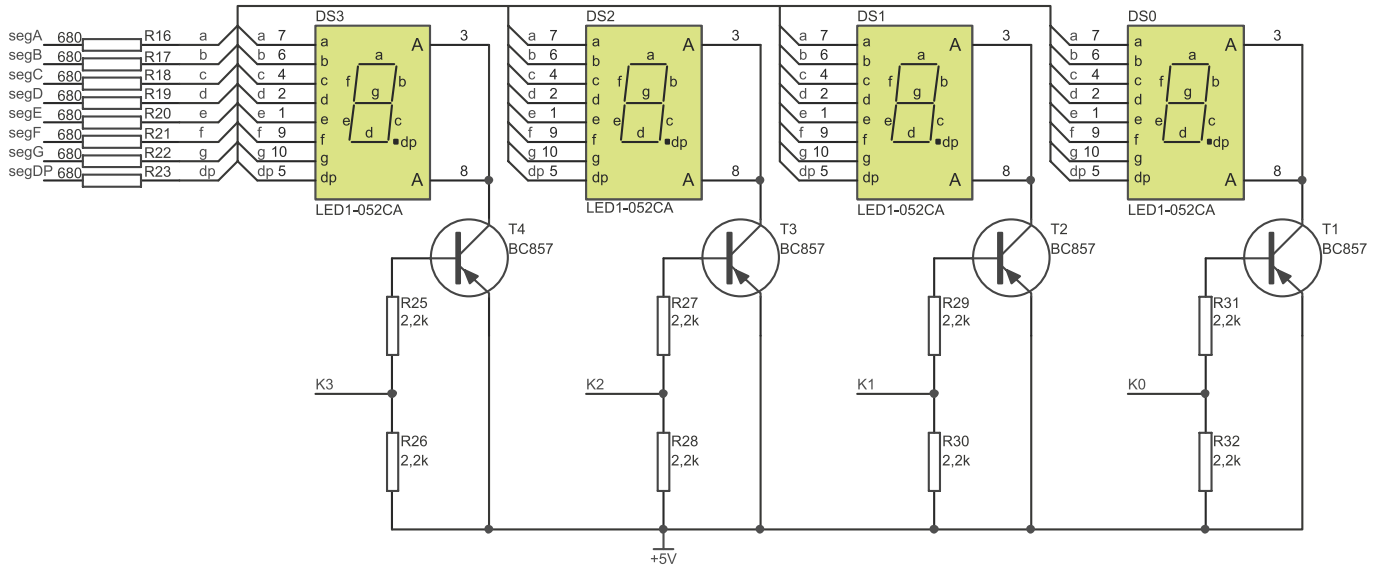
Diody LED

Zestaw wyposażono w osiem diod LED przeznaczonych do sygnalizacji stanu portów mikrokontrolera (aktywny stan wysoki). Linie sterujące diodami LED są wprowadzone na złączu Con9.



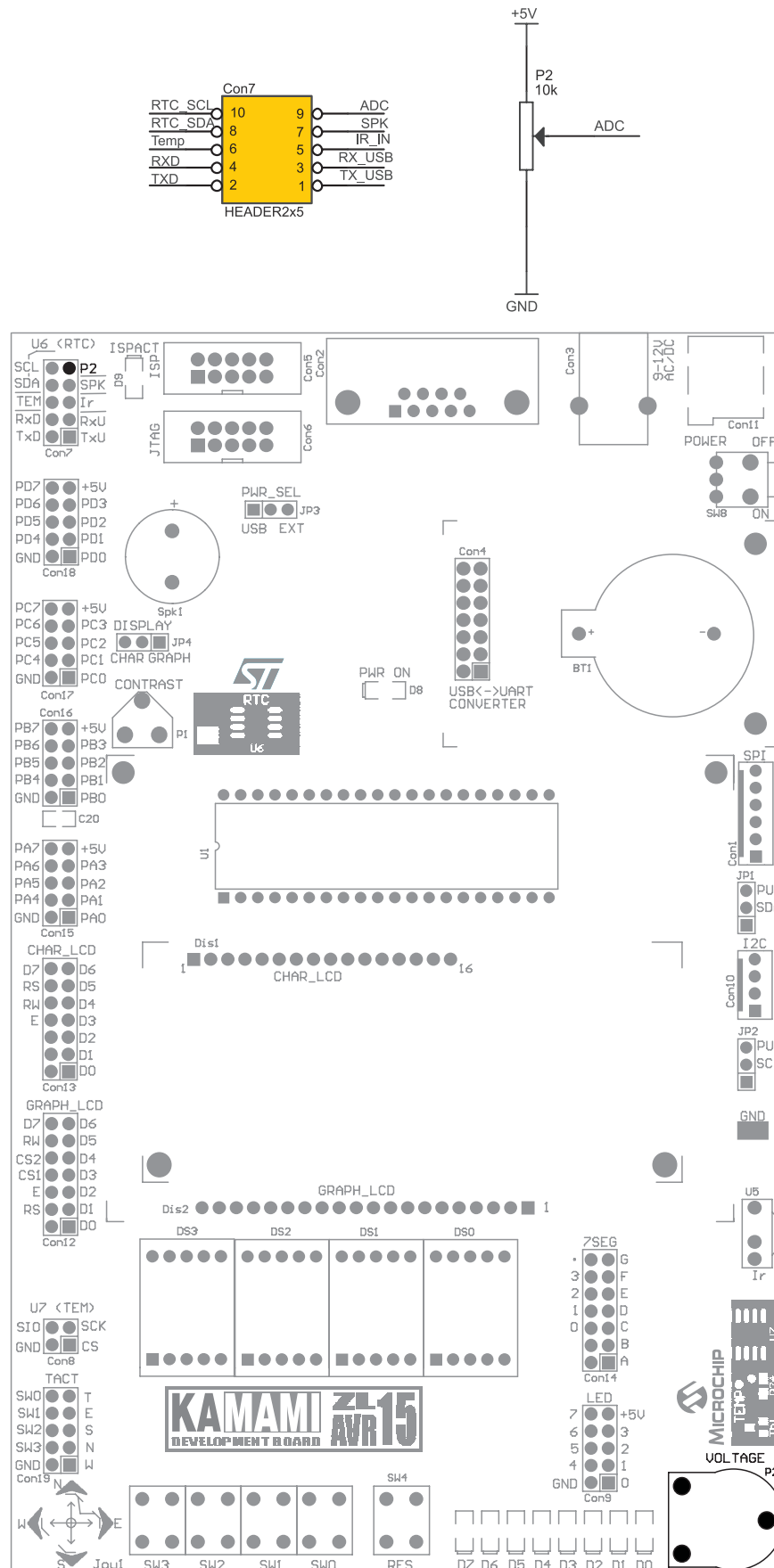
Wyświetlacz LED

Zestaw ZL15AVR wyposażono w czterocyfrowy siedmiosegmentowy wyświetlacz LED (DS0...DS3) wraz z tranzystorami sterującymi anody wyświetlaczy. Sygnały sterujące anodami (0...3) oraz poszczególnymi segmentami wyświetlacza zostały wyprowadzone na złącze Con14.



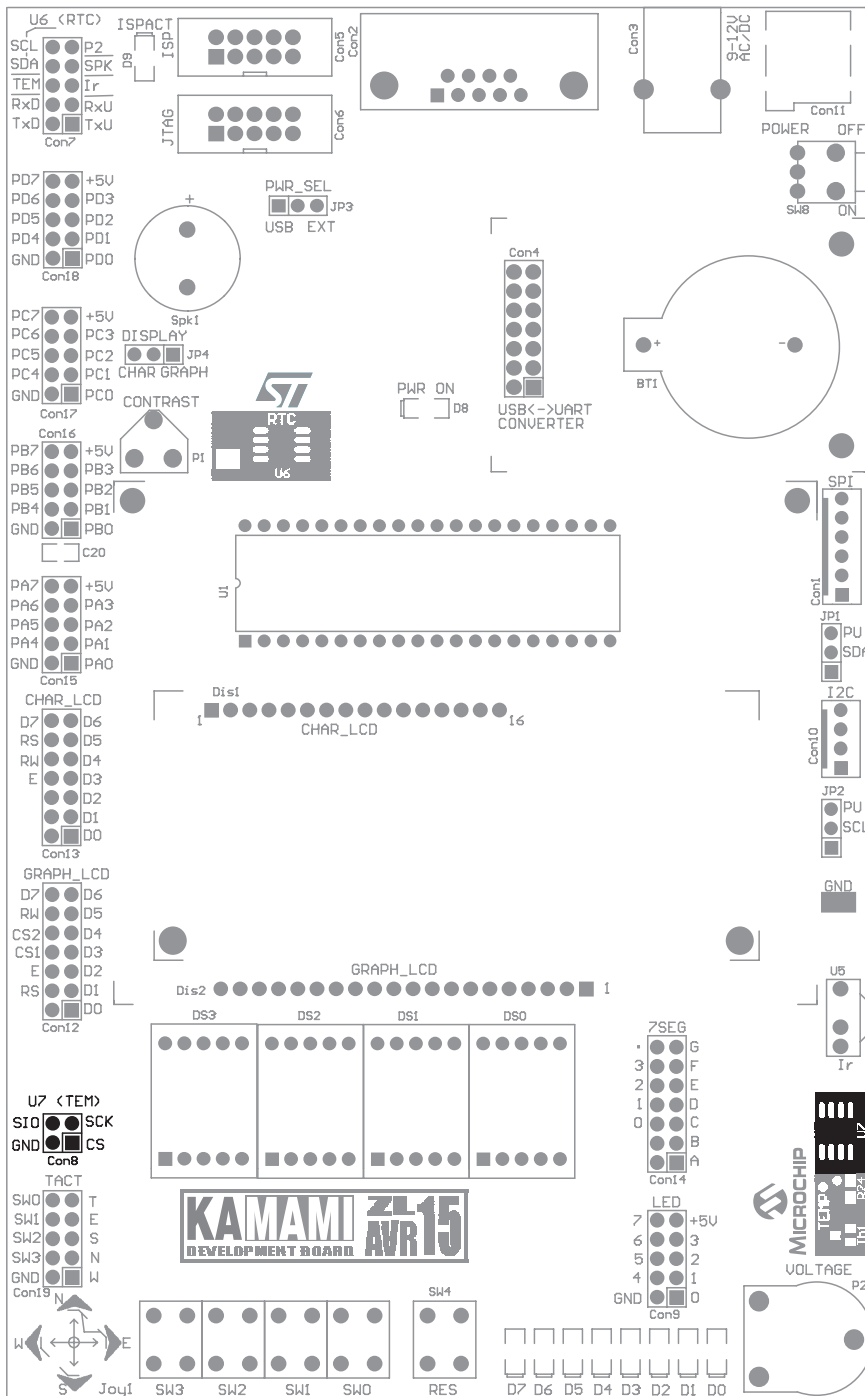
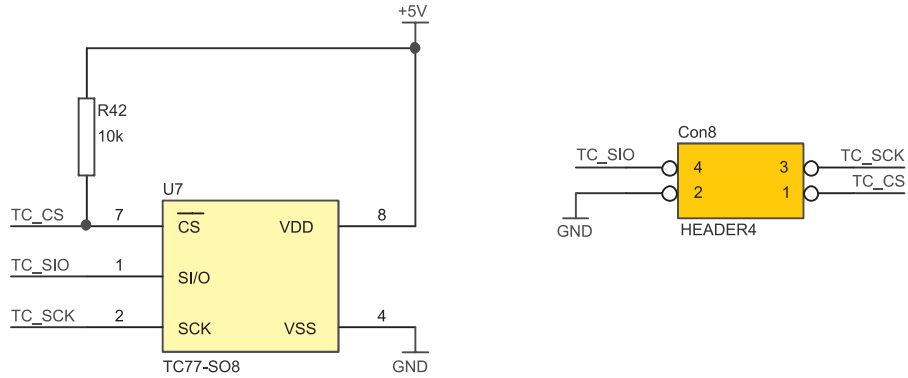
Potencjometr analogowy

Zestaw ZL15AVR wyposażono w potencjometr P2, który może zostać wykorzystany do podawania napięcia z zakresu 0...5 V na wejścia przetwornika analogowo-cyfrowego mikrokontrolera AVR. Środkowe wyprowadzenie potencjometru dostępne jest na złączu Con7 (pin oznaczony P2).



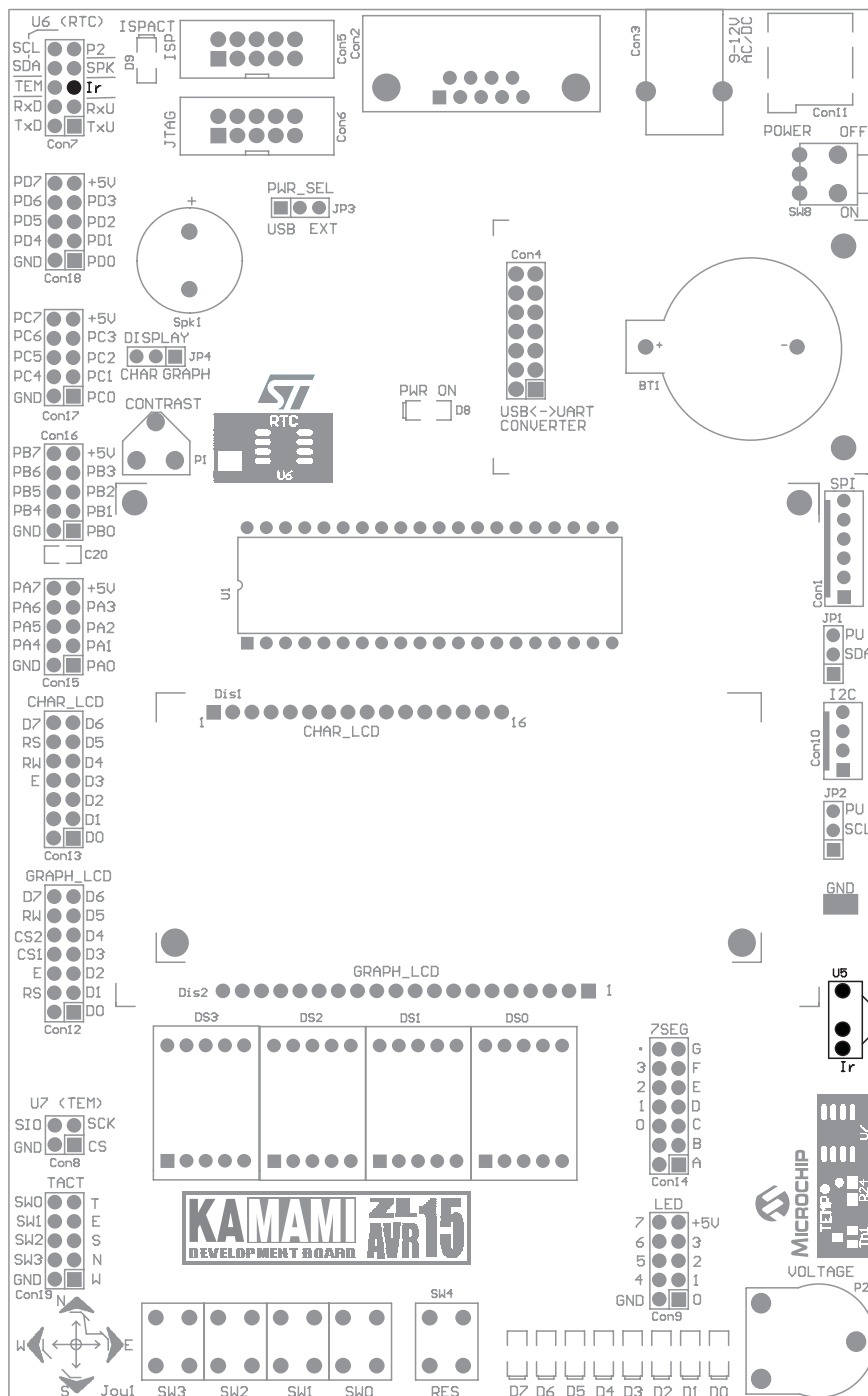
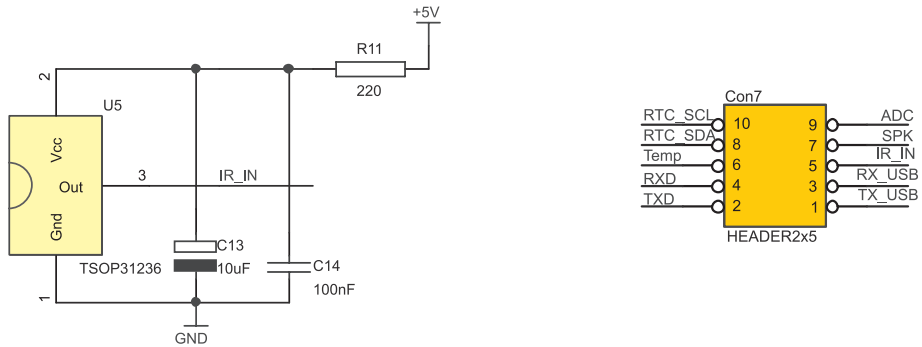
Termometr cyfrowy TC77

Zestaw wyposażono w cyfrowy termometr TC77 (Microchip) pracujący na magistrali SPI. Linie sterujące termometrem (SCK, SIO, CS) są dostępne na złączu Con19.



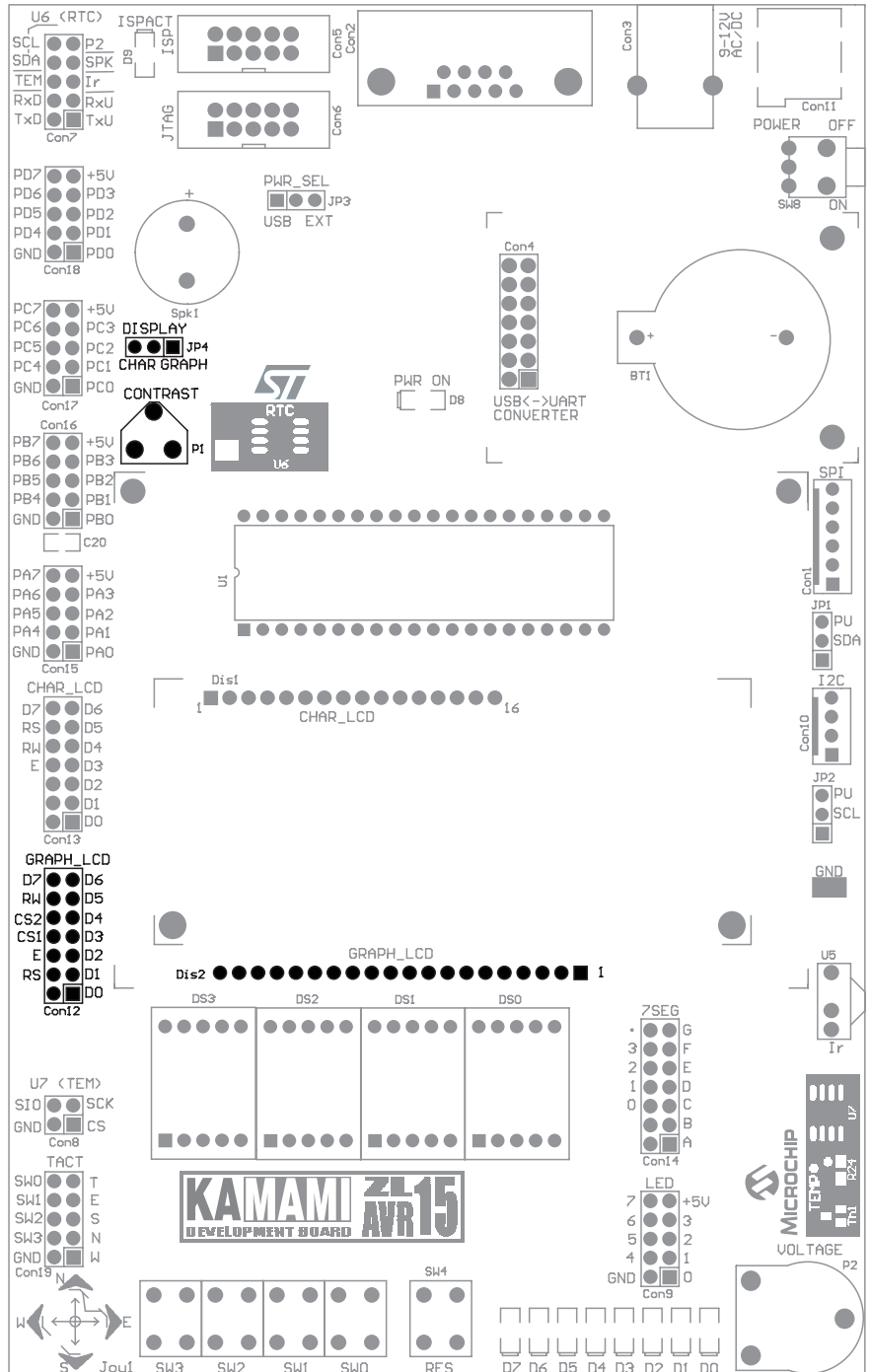
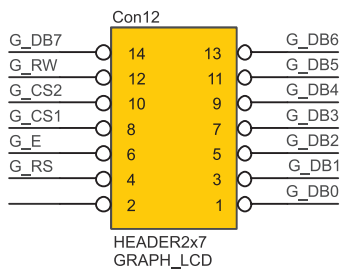
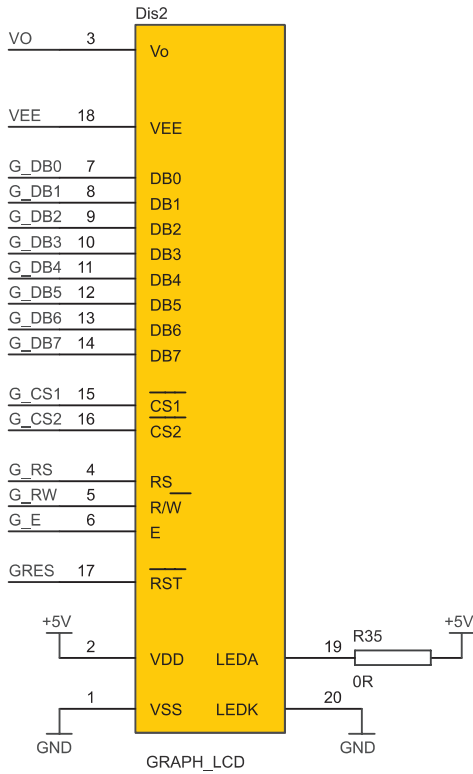
Odbiornik podczerwieni

Zestaw ZL15AVR wyposażono w układ odbiornika podczerwieni TSOP31236, który umożliwia odbiór sygnałów nadawanych przez piloty zdalnego sterowania od sprzętu RTV lub od innego urządzenia nadającego sygnał w podczerwieni o częstotliwości nośnej 36 kHz. Wyjście odbiornika jest wyprowadzone na złącze Con7 (wyprowadzenie oznaczone Ir).



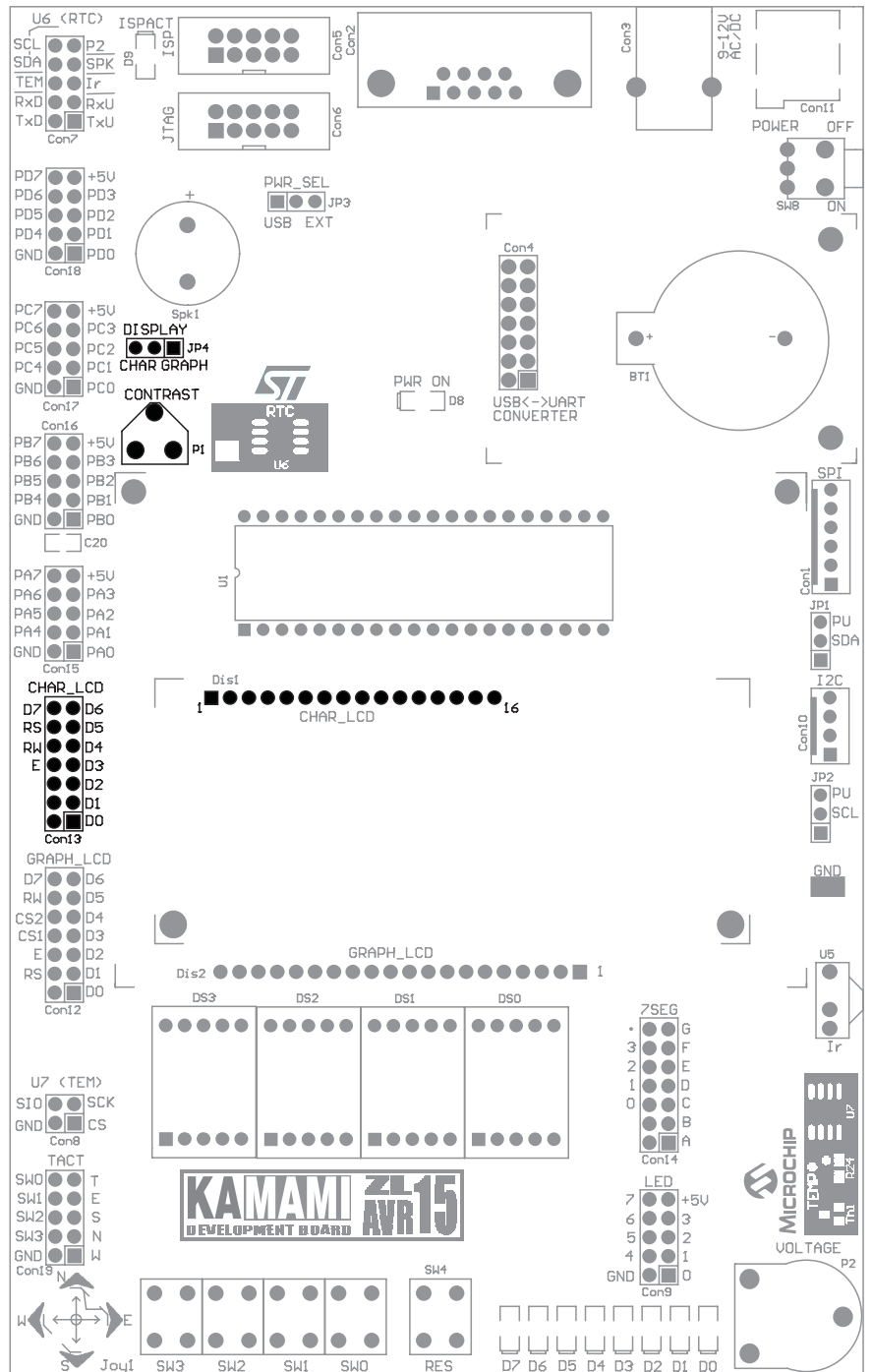
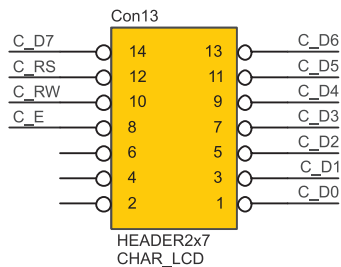
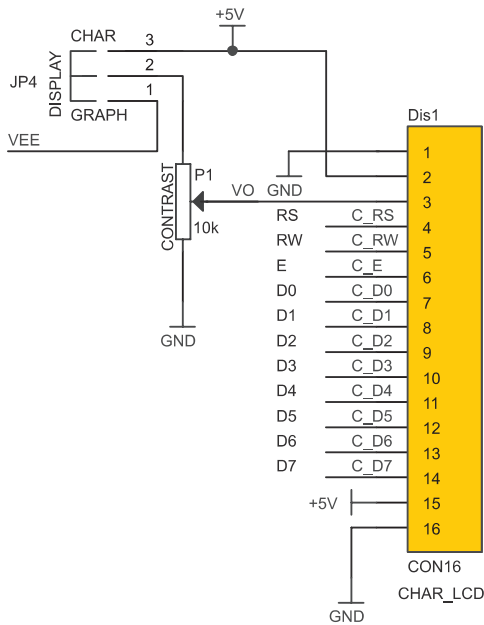
Złącze graficznego wyświetlacza LCD

Zestaw wyposażono w złącze umożliwiające dołączenie do zestawu wyświetlacza graficznego o organizacji 128x64 piksele ze sterownikiem KS0107/KS0108 (oznaczenie GLCD12864). Regulacja kontrastu wyświetlacza jest możliwa poprzez potencjometr P1 (zworka JP4 DISPLAY w pozycji GRAPH). Linie sterujące wyświetlaczem dostępne są na złączu Con12/LCD_GRAPH.



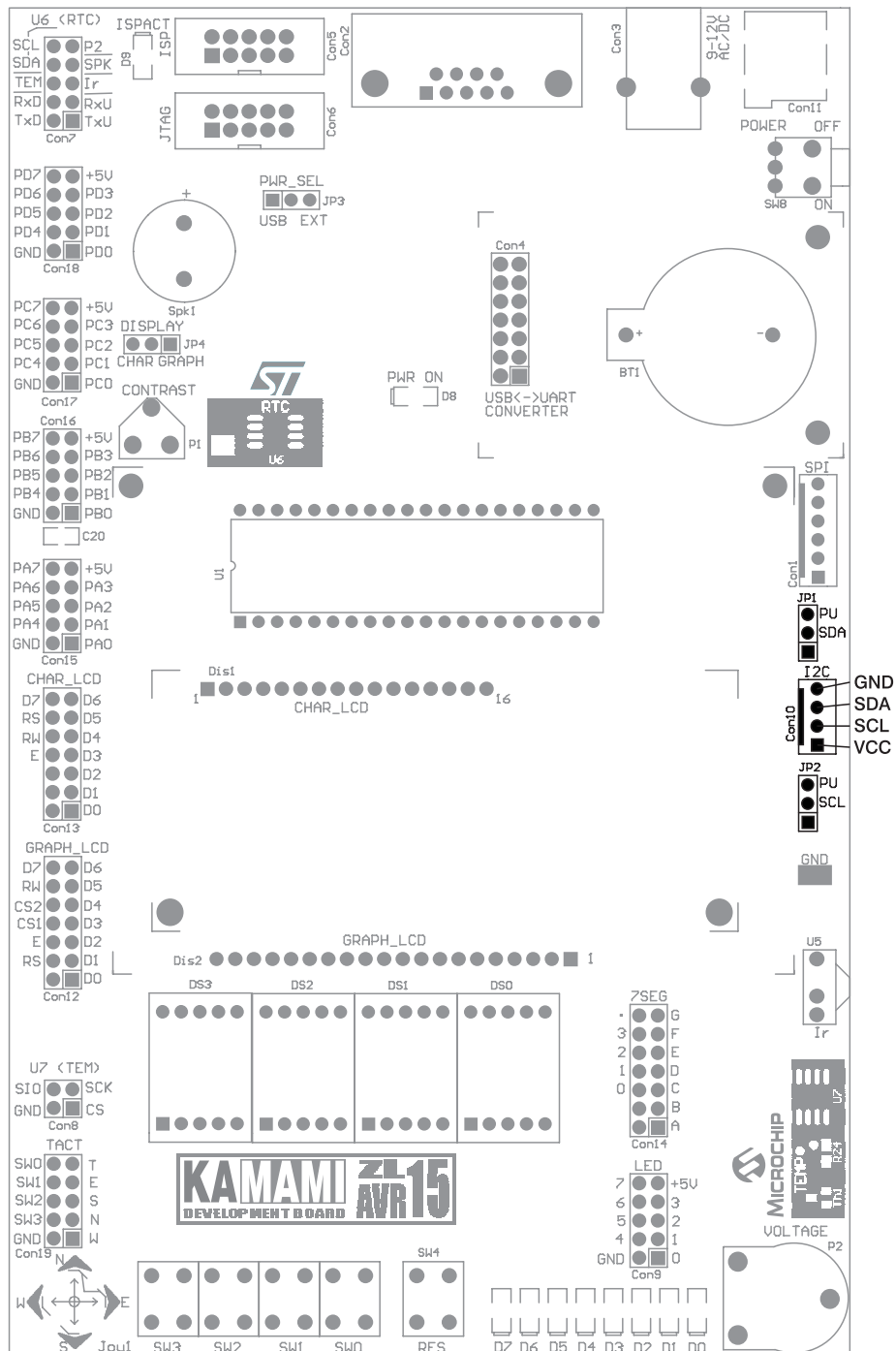
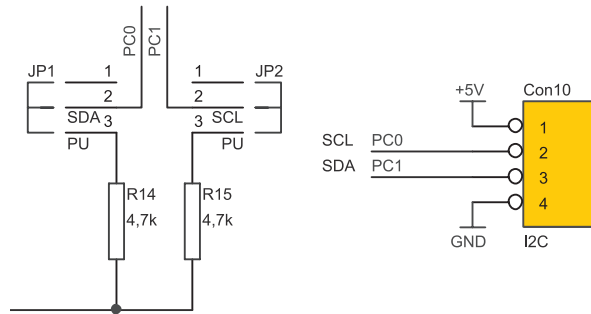
Złącze alfanumerycznego wyświetlacza LCD

Zestaw ZL15AVR wyposażono w złącze umożliwiające dołączenie alfanumerycznego wyświetlacza LCD 2x16 znaków ze sterownikiem zgodnym z HD44780 (oznaczenie LCD1602). Regulacja kontrastu wyświetlacza jest możliwa poprzez potencjometr P1 (zworka JP4 DISPLAY w pozycji CHAR). Linie sterujące wyświetlaczem są dostępne na złączu Con13/LCD_2x16.



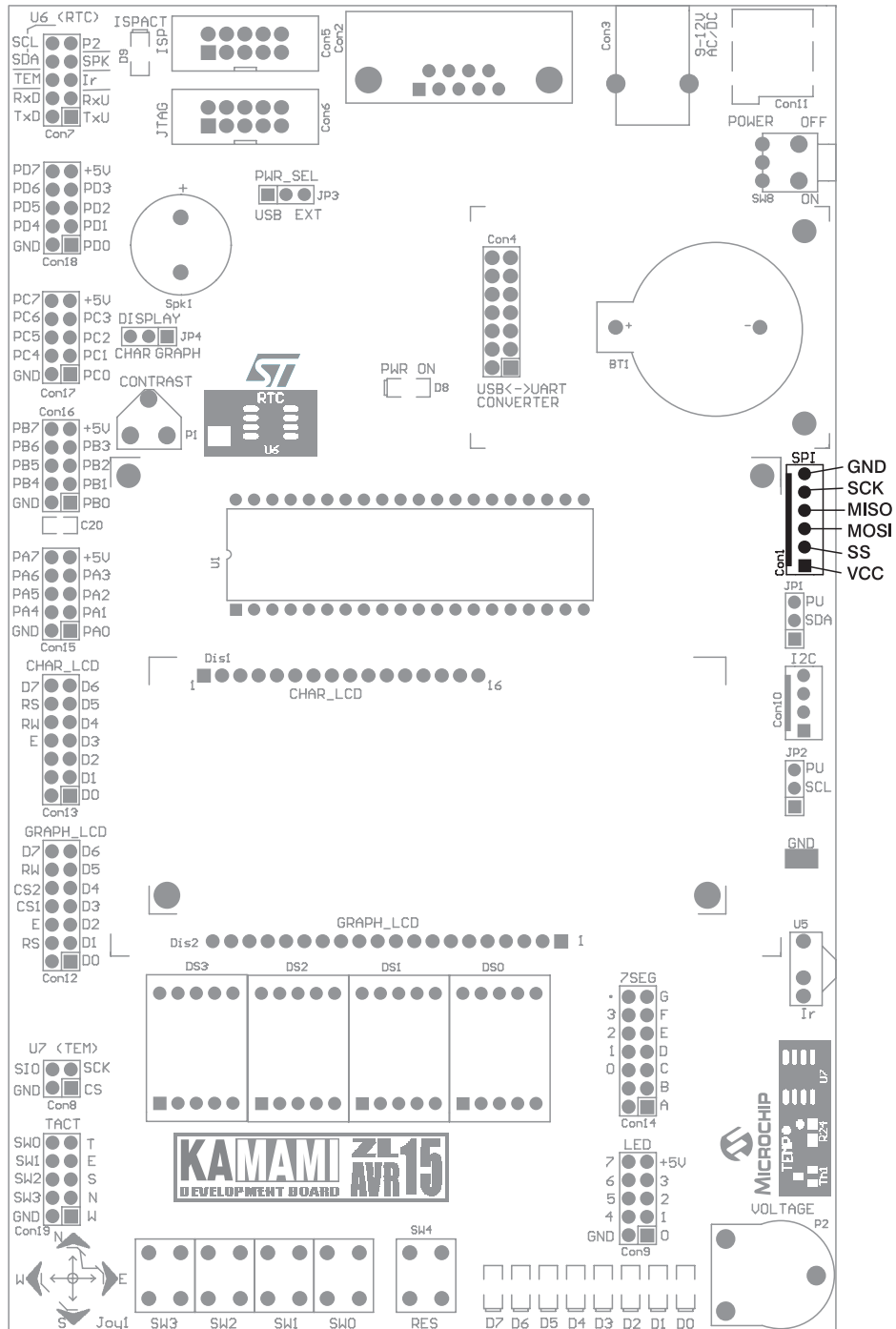
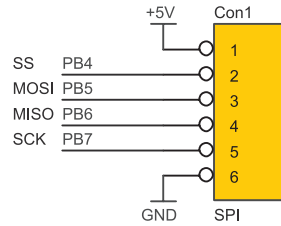
Złącze magistrali I2C

Zestaw ZL15AVR został wyposażony w złącze Con10/I2C umożliwiające podłączenie zewnętrznych układów pracujących na magistrali I2C. Możliwe jest również dołączenie rezystorów podciągających do linii SDA za pomocą zworki JP1 w pozycji PU/SDA oraz do linii SCL za pomocą zworki JP2 w pozycji PU/SCL.



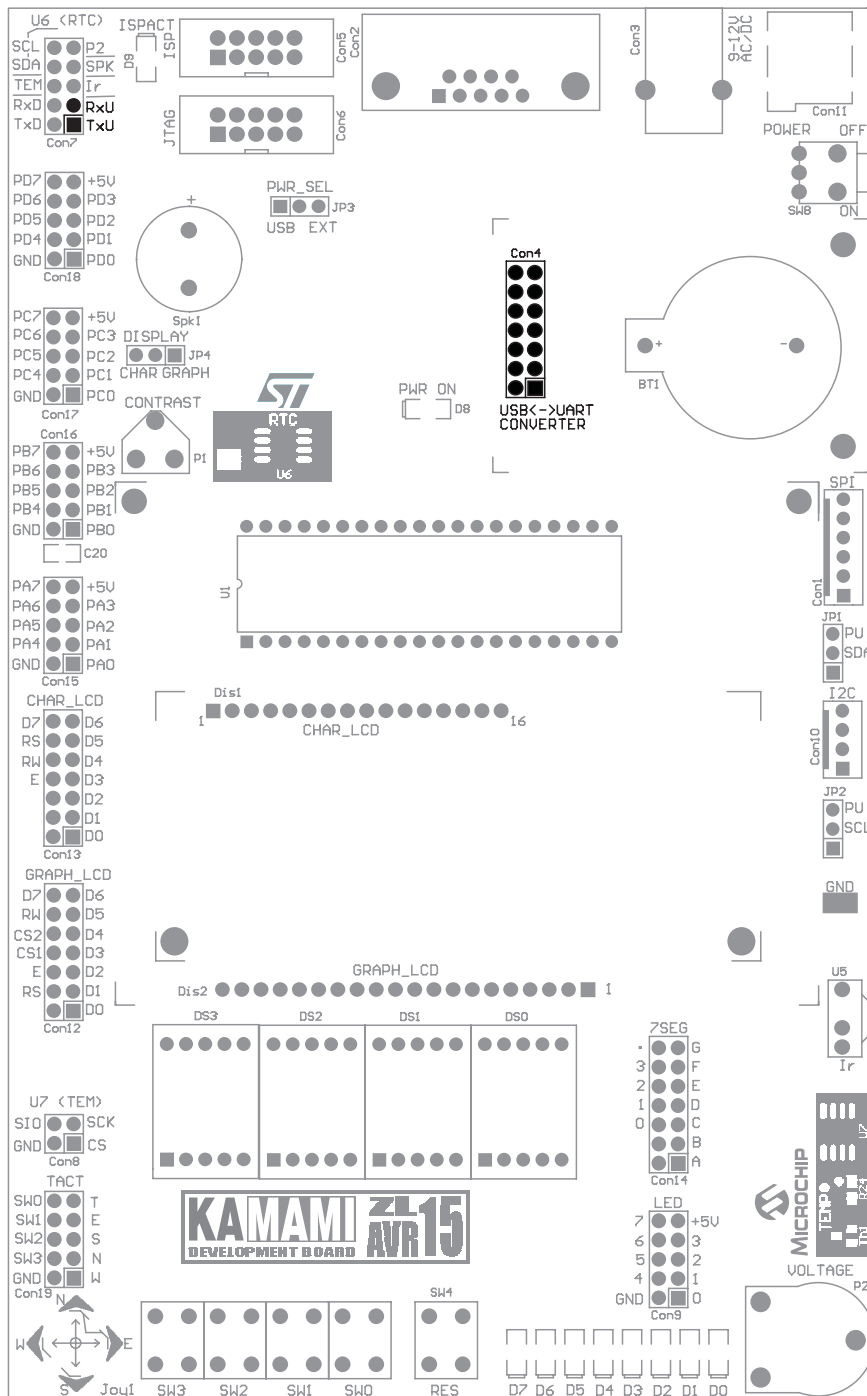
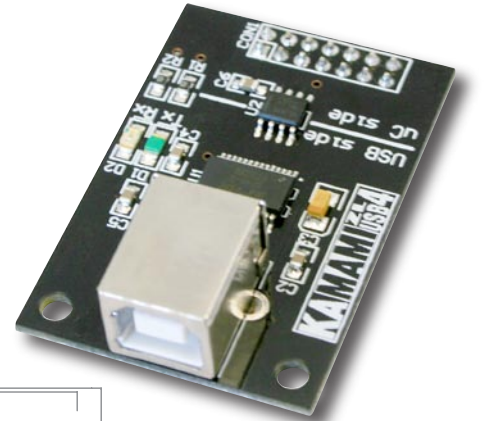
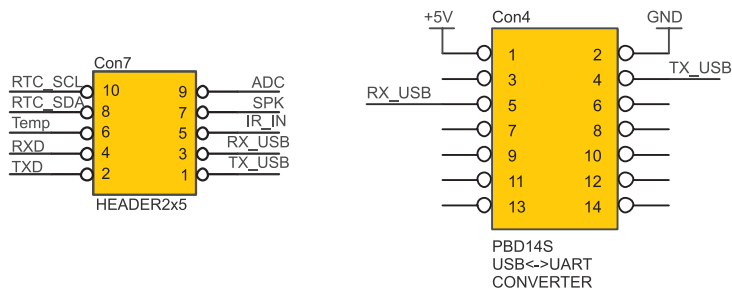
Złącze magistrali SPI

Zestaw ZL15AVR został wyposażony w złącze Con1/SPI umożliwiające podłączenie zewnętrznych układów pracujących na magistrali SPI.



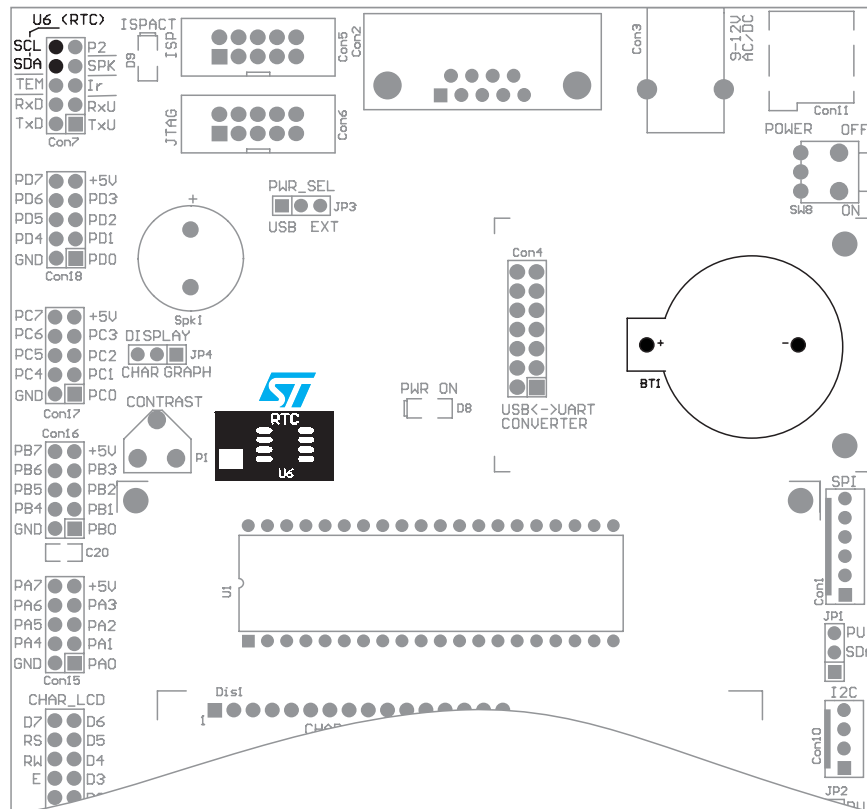
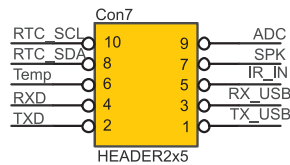
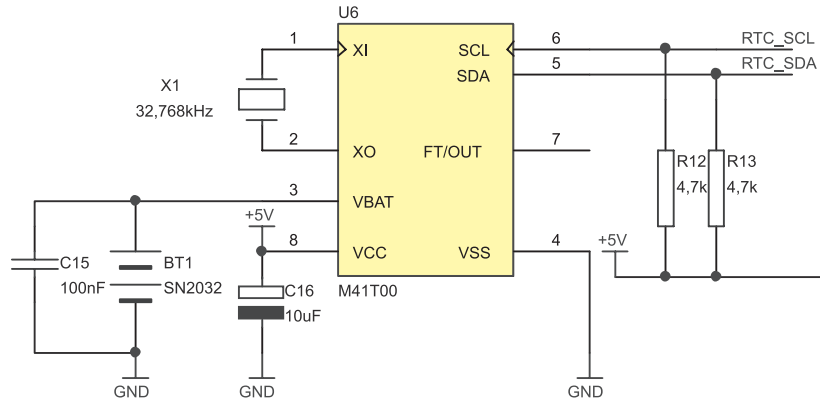
Złącze konwertera USB <-> UART

Zestaw ZL15AVR wyposażono w złącze Con4 umożliwiające dołączenie modułu konwertera USB<->UART (np. ZL1USB_A lub ZL4USB). Linie interfejsu UART są dostępne na złączu Con7 i oznaczone są TxU oraz RxU.



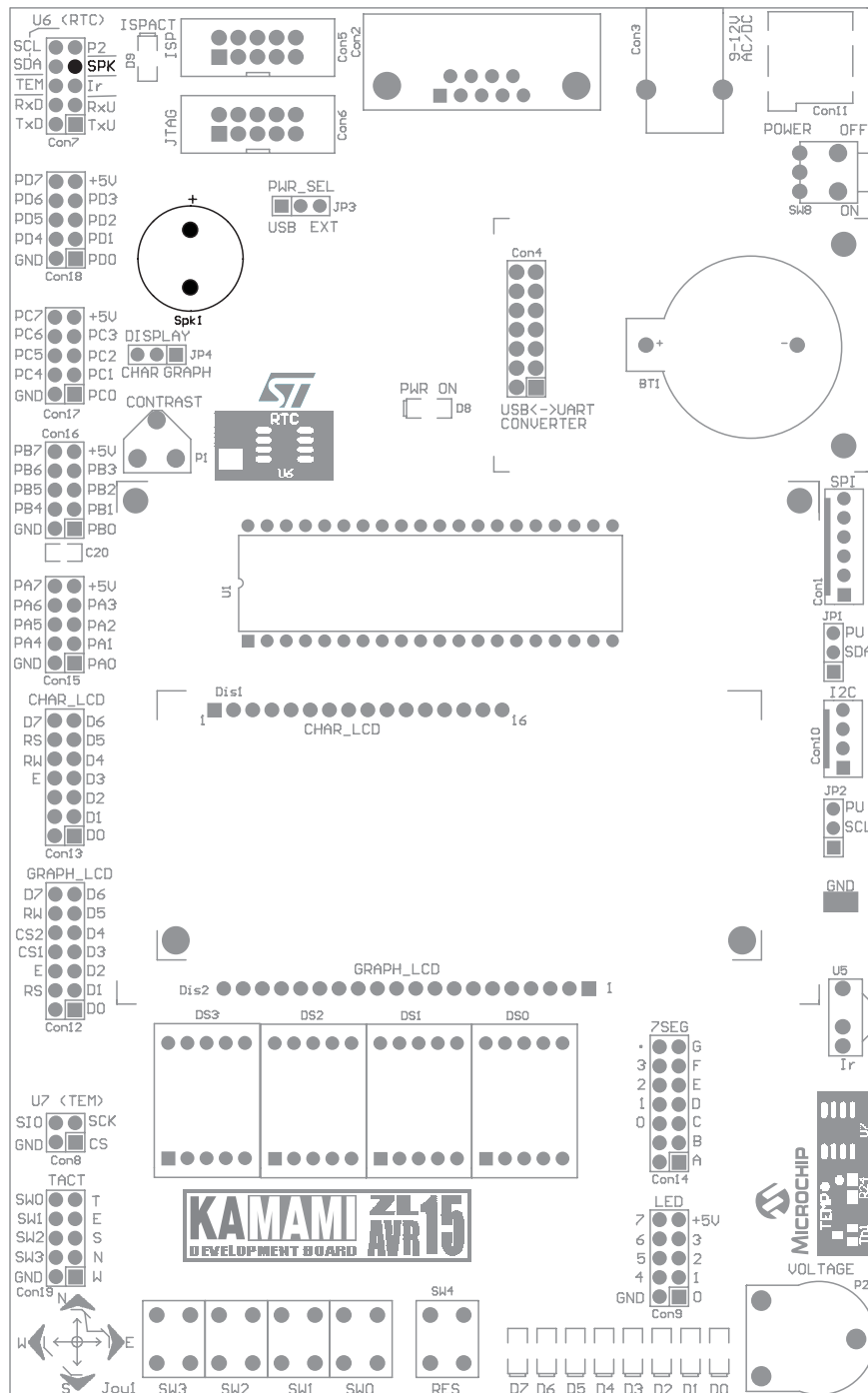
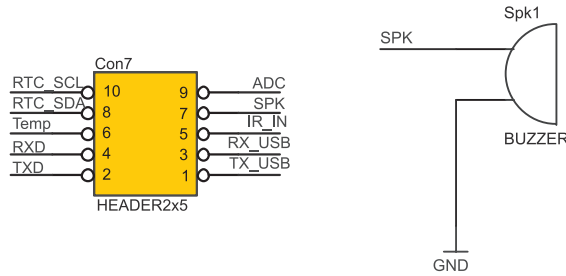
Zegar czasu rzeczywistego M41T00

Zestaw ZL15AVR został wyposażony w układ zegara czasu rzeczywistego M41T00 (STMicroelectronics). Oprócz samego układu M41T00 na płytce zestawu znajduje się również rezonator kwarcowy 32768 Hz oraz podstawa pod baterię litową CR2032. Wyprowadzenia SDA i SCL zegara dostępne są na złączu Con7.



Przetwornik piezoelektryczny

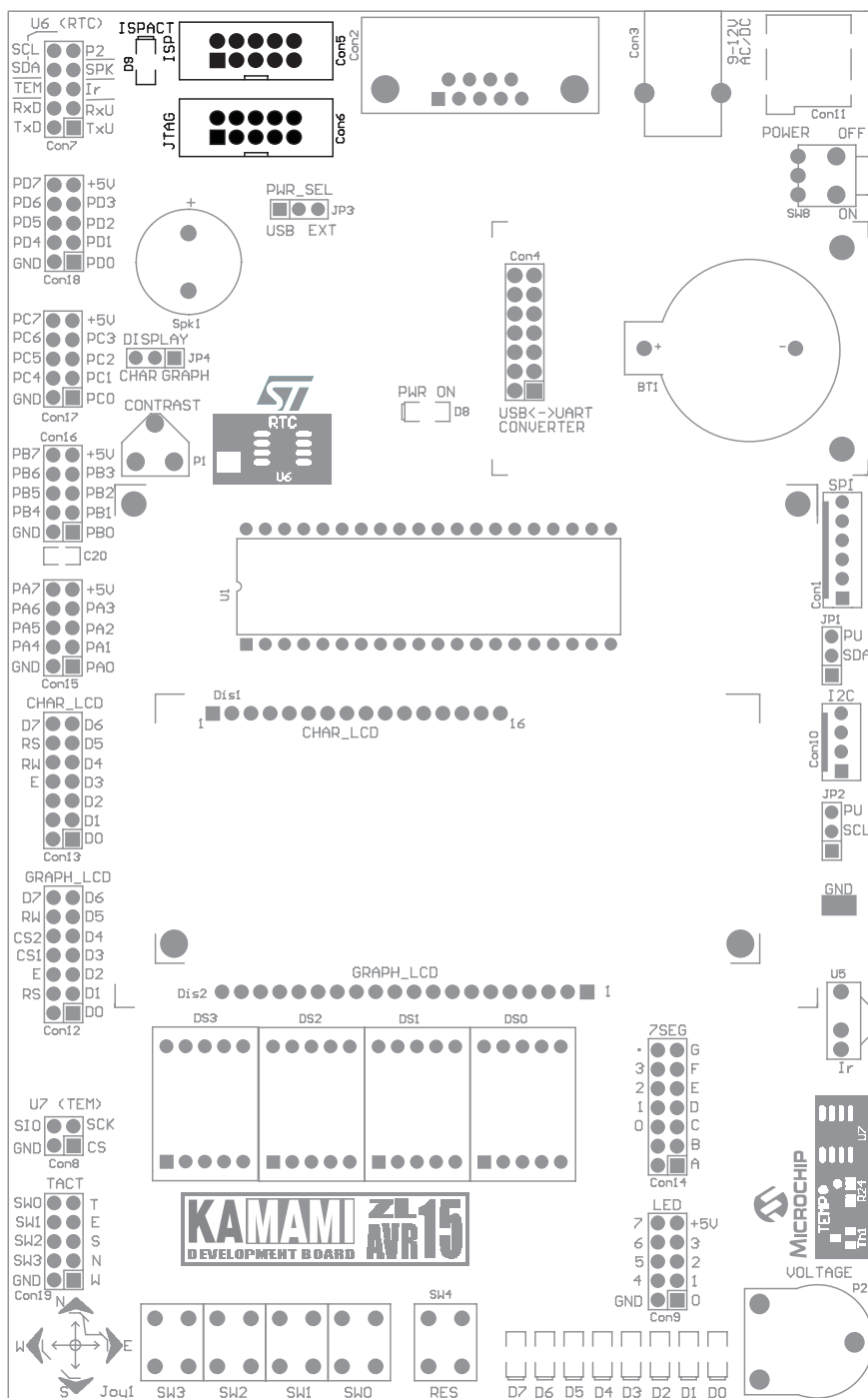
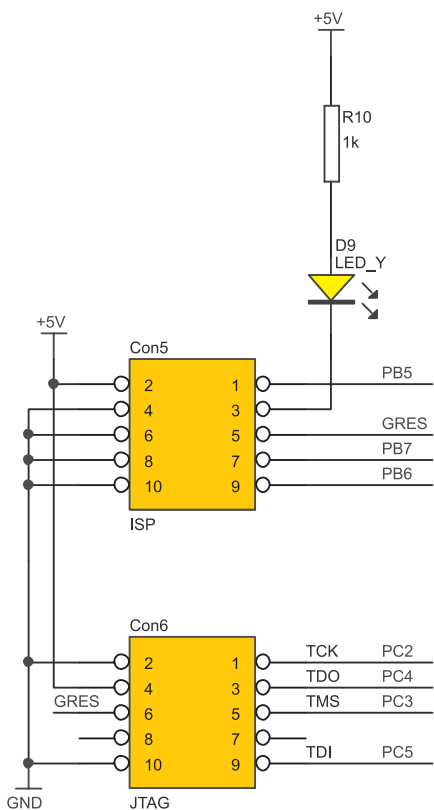
Zestaw ZL15AVR wyposażono w przetwornik piezoelektryczny Spk1 służący do generowania dźwięku. Do generowania dźwięków można wykorzystać jeden z układów licznikowych mikrokontrolera ATmega32 lub też generować dźwięki programowo. Wyprowadzenie przetwornika oznaczone SPK jest dostępne na złączu Con7.



Złącze JTAG i ISP

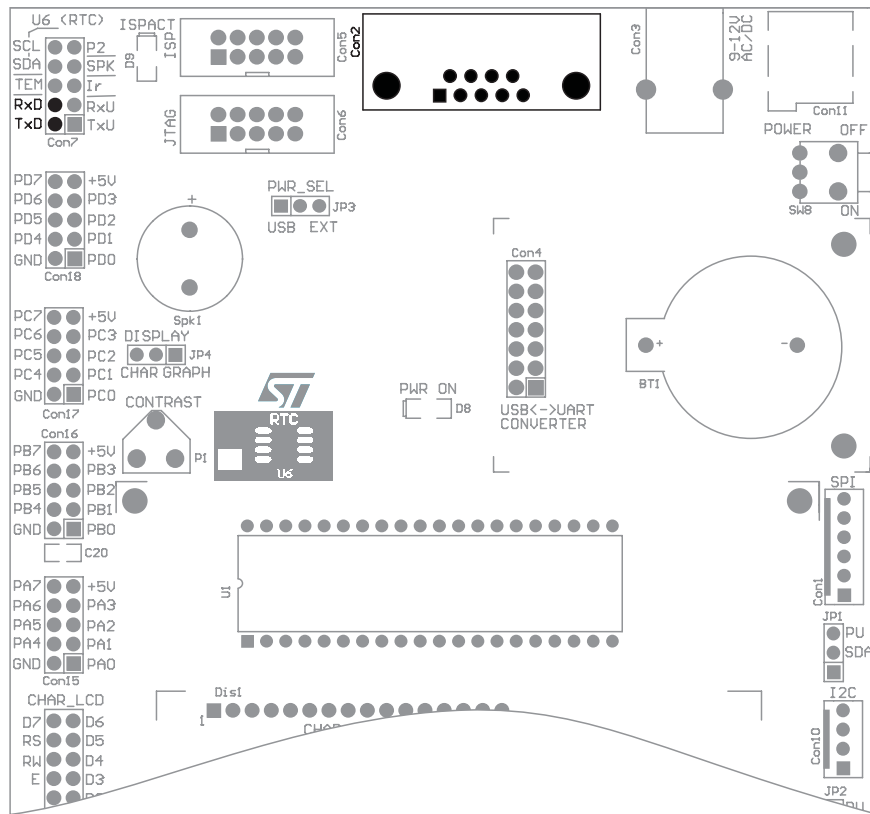
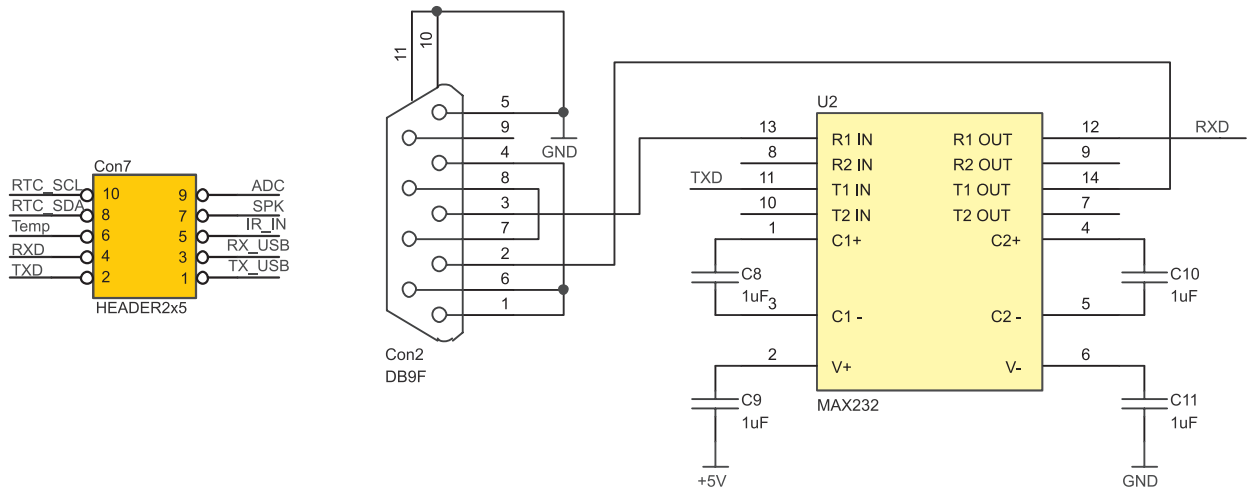
Zestaw ZL15AVR wyposażono w złącze Con6 umożliwiające programowanie oraz debugowanie programu z wykorzystaniem interfejsu JTAG (np. ZL16PRG). W przypadku korzystania z interfejsu JTAG nie należy do wyprowadzeń podłączać żadnych obciążeń o niskiej impedancji.

Zestaw ZL15AVR został wyposażony w 10-wyprowadzeniowe złącze Con5 przeznaczone do dołączenia programatora ISP. Na czas programowania do wyprowadzeń PB5...PB7 nie należy podłączać obciążeń o niskiej impedancji, gdyż mogą one zakłócić proces programowania mikrokontrolera.

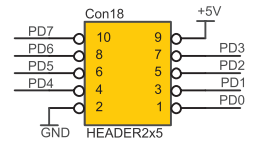
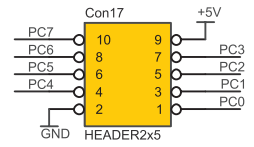
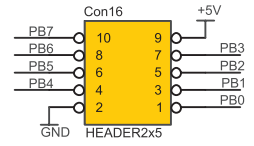
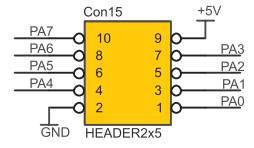
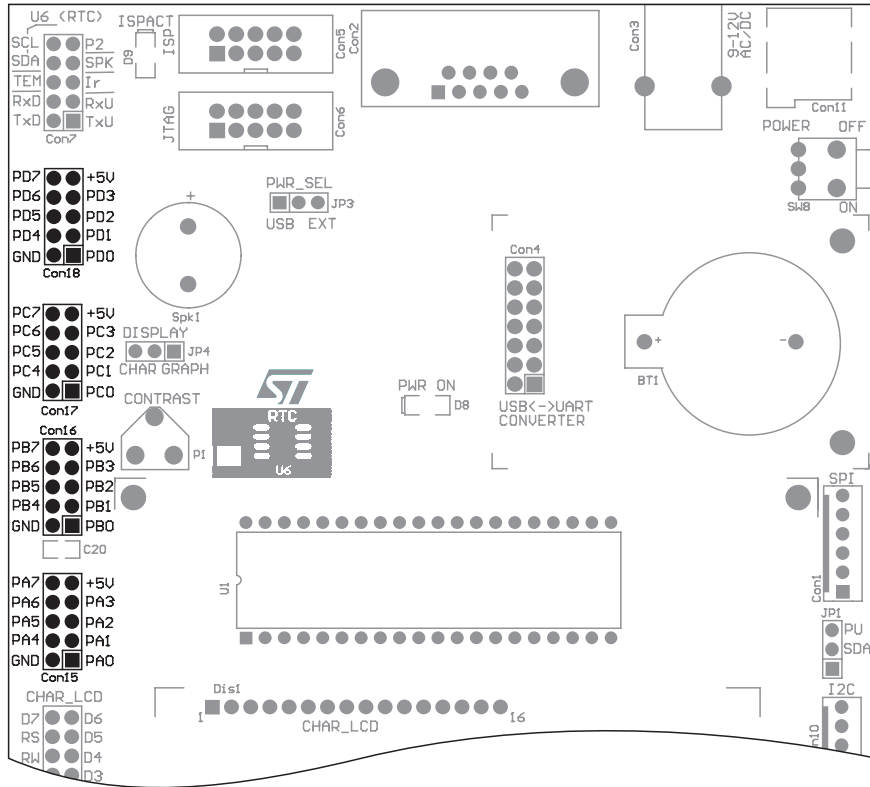


Złącze RS232

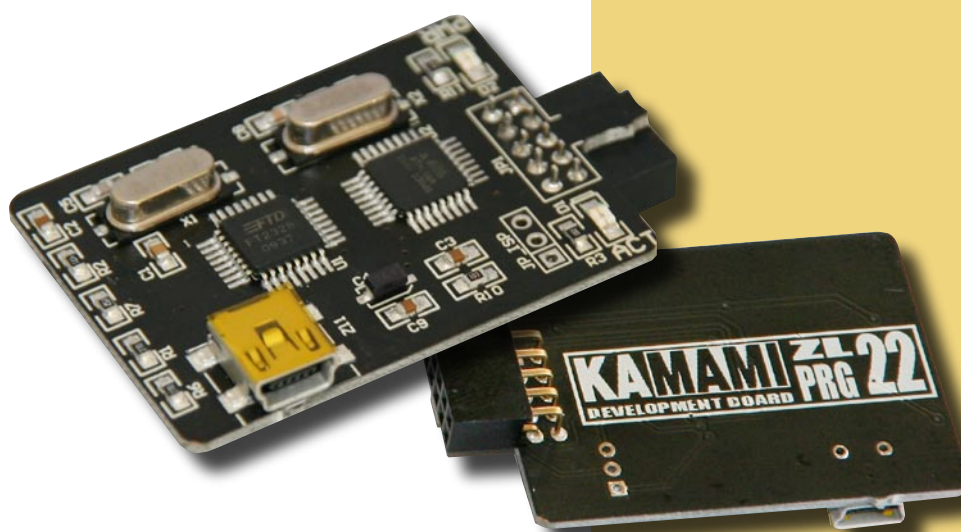
Zestaw ZL15AVR został wyposażony w złącze typu DB9F (Con2), umożliwiające komunikację z komputerem PC poprzez interfejs RS232. Sygnały RxD oraz TxD są dostępne na złączu Con7.



Wyprowadzenia portów wejścia/wyjścia



Programator ISP mikrokontrolerów AVR z interfejsem USB (zgodny z STK500 v2)



ZL22PRG jest programatorem ISP dla procesorów rodziny AVR firmy Atmel. Zgodność programatora z protokołem STK500 v2 zapewnia bezproblemową współpracę ze środowiskiem AVRStudio firmy Atmel, Bascom AVR oraz z wieloma popularnymi i bezpłatnymi programami (np. AVR Dude).

Wprowadzenie

Programator ZL22PRG umożliwia programowanie w systemie mikrokontrolerów z rodziny AVR. Dzięki zgodności z protokołem STK500v2 możliwe jest sterowanie programatorem z poziomu wielu popularnych aplikacji (m.in. AVR Studio, AVR Dude).

Podstawowe parametry

- ▶ programator zgodny z protokołem STK500 v2
- ▶ współpracuje m.in. z programem Atmel AVR Studio, Bascom AVR, AVR Dude
- ▶ programowanie w systemie (ISP) wszystkich mikrokontrolerów AVR obsługiwanych przez AVRStudio i STK500 w trybie ISP
- ▶ zakres napięć zasilających programowanych mikrokontrolerów: +3...5 V
- ▶ komunikacja z komputerem przez USB (gniazdo mini USB 5 pin)
- ▶ szybkość transmisji: 115 kB – wirtualny port szeregowy (sterowniki FTDI dla Windows/Linux)
- ▶ stan pracy programatora sygnalizowany przez wbudowaną diodę LED
- ▶ standardowe 10-wyprowadzeniowe złącze do układu docelowego
- ▶ zasilanie programatora z portu USB

Wyposażenie standardowe

Kod	Opis
ZL22PRG	<ul style="list-style-type: none">▶ Programator▶ Kabel USB A/B-mini

	btc
---	------------

BTC Korporacja
05-120 Legionowo
ul. Lwowska 5
tel.: (22) 767-36-20
faks: (22) 767-36-33
e-mail: biuro@kamami.pl
<http://www.kamami.pl>

Zastrzegamy prawo do wprowadzania zmian bez uprzedzenia.

Oferowane przez nas płytki drukowane mogą się różnić od prezentowanej w dokumentacji, przy czym zmianom nie ulegają jej właściwości użytkowe.

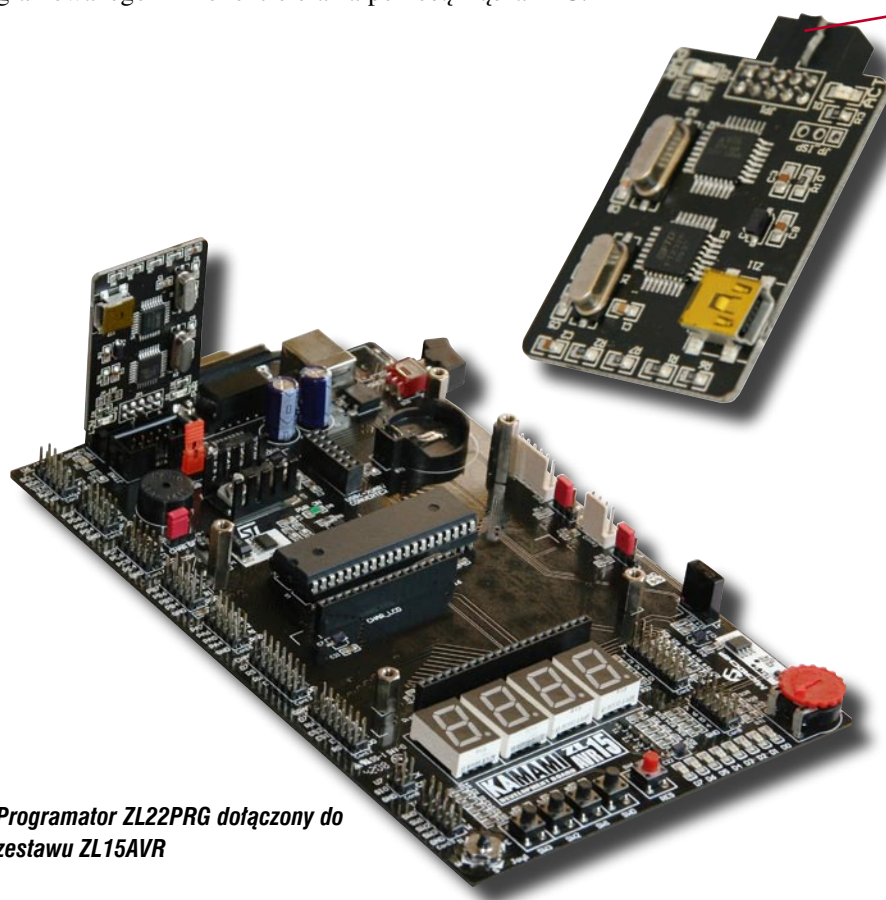
BTC Korporacja gwarantuje zgodność produktu ze specyfikacją.

BTC Korporacja nie ponosi odpowiedzialności za jakiegokolwiek szkody powstałe bezpośrednio lub pośrednio w wyniku użycia lub nieprawidłowego działania produktu.

BTC Korporacja zastrzega sobie prawo do modyfikacji niniejszej dokumentacji bez uprzedzenia.

Podłączenie programatora

Programator ZL22PRG wyposażono w 10-stykowe złącze żeńskie umożliwiające jego wygodne dołączenie do programowanego mikrokontrolera za pomocą złącza IDC.



Wypust pomagający w prawidłowym dołączeniu programatora do 10-wyprowadzeniowego gniazda ISP

Programator ZL22PRG dołączony do zestawu ZL15AVR

Sygnaly służące do programowania mikrokontrolerów AVR wyprowadzono na złącze 10-pinowe. Rozmieszczono je zgodnie ze standardem Kanda STK200 (rysunek poniżej).

MOSI	●1	●2	VCC
	●3	●4	GND
RST	●5	●6	GND
SCK	●7	●8	GND
MISO	●9	●10	GND

Rozmieszczenie sygnałów w złączu programowania układu docelowego (widok od strony elementów)

UWAGA

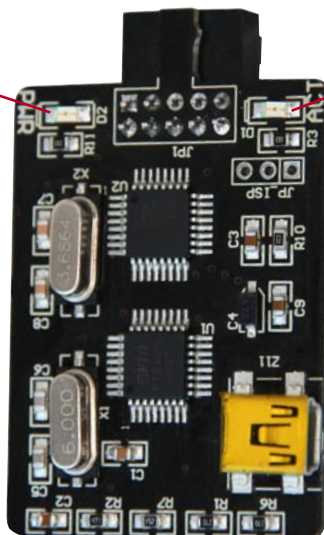
Programator ZL22PRG jest zasilany ze złącza USB. Zasilanie nie jest przekazywane do programowanego/debugowanego mikrokontrolera, trzeba więc zasilać go niezależnie.

Elementy sygnalizacyjne

Programator ZL22PRG wyposażono w diody LED:

- ACT – sygnalizującą aktywność programatora;
- PWR – sygnalizującą obecność napięcia zasilającego programator.

Dioda LED sygnalizująca zasilanie programatora



Dioda LED sygnalizująca aktywność programatora

Sterowniki

Do poprawnej pracy programatora konieczne jest zainstalowanie sterowników obsługujących wirtualny port szeregowy (*Virtual COM Port*) dla układu FT232. W zależności od używanego systemu operacyjnego odpowiednie sterowniki należy pobrać ze strony <http://www.ftdichip.com> z działu *Drivers*. Instalacja sterowników przebiega w sposób standardowy dla używanego systemu operacyjnego.