

MCS ELECTRONICS

Elektronika nie musi być trudna



Edukacyjny zestaw posiada:

- wyświetlacz LCD
- złącze PS/2 klawiatura/myszka
- uniwersalną płytkę
- interfejs USB
- interfejs RS-232
- złącze ISP
- obszerna instrukcja umożliwiająca zrobienie pierwszych kroków w elektronice

Edukacyjny zestaw EDB
Instrukcja

AKTUALNE INFORMACJE SĄ DOSTĘPNE NA STRONIE

www.mcselec.com/edb

MCS ELECTRONICS

Edukacyjny zestaw EDB - Instrukcja

Barry de Graaff
© 2005 MCS Electronics
www.mcselec.com

Spis treści

Wstęp	Błąd! Nie zdefiniowano zakładki.
1. Doświadczenia	8
1.1 Wymaganie sprzętowe	8
1.2 Uważaj na niezaprogramowane mikrokontrolery!	8
2. Pierwsze kroki	9
2.1 Montaż Twojego zestawu EDB	9
2.2 Instalacja Bascom AVR	14
2.3 Programowanie mikrokontrolera przez port szeregowy (procedura Boot loader)	20
2.4 Usuwanie błędów	20
3. Eksperymenty	23
3.1 Podstawowa obsługa linii I/O	25
3.1.2 Biegające światło diod LED	27
3.1.3a Wyjście sterujące tranzystorem	29
3.1.3b Wyjście sterujące tranzystorem FET	32
3.1.3c Wyjście sterujące przekaźnikiem	33
3.1.4 Wyjście sterujące układem ULN2803	34
3.1.5 Współpraca transoptorów z liniami I/O	36
3.2 Komunikacja przez UART	39
3.2.1 UART	39
3.2.2 Programowy UART	44
3.3 Obsługa wyświetlaczy	46
3.3.1 Wyświetlacz LCD (HD44780)	46
3.3.2 Wyświetlacz 7-segmentowy	48
3.4 Klawiatury	50
3.4.1 Klawiatura PS2 lub AT	50
3.4.2 Klawiatura matrycowa	52
3.4.3 Zdalny odbiór sygnałów RC5	54
3.5 Zaawansowana obsługa linii I/O	56
3.5.1 Licznik z wejściem eliminującym drgania	56
3.5.2 Komenda GetRC	58
3.5.3 Syrena z wykorzystaniem komendy SOUND	59

Spis treści *(kontynuacja)*

3.6 Przetwornik A/C i C/A	60
3.6.1 Generator sygnału PWM	60
3.6.2 Przetwornik A/C z fotorezystorem (LDR)	62
3.6.3 Tani woltomierz	65
3.7 Inne elementy mikrokontrolera	69
3.7.1 Pamięci	69
3.7.2 Układ Watchdog	71
3.7.3 Wykorzystanie czasomierza	72
3.8 Inne interfejsy	73
3.8.1 Interfejs I ² C bus	73
3.8.2 Interfejs USB	75
3.8.3 Procedury instalacji	77
3.8.3.1 Instalacja wirtualnego portu COM dla Windows	77
3.8.3.2 Instalacja urządzenia USB dla Windows	80
3.8.3.3 Odinstalowanie sterownika	83
3.8.4 Interfejs USB, urządzenie z wirtualnym portem COM	84
3.8.5 Interfejs USB, urządzenie z domyślnym PID&VID	86
3.8.6 Interfejs USB, urządzenie z własnym PID&VID	87
3.9 Sterowanie silnikami	88
3.9.1 Sterowanie silnikiem krokowym	88
3.9.2 Sterowanie sygnałem PWM silnika DC	89
4. Inne metody programowania	91
4.1 Programowanie programatorem STK200/300 ISP	92
4.2 Programowanie programatorem USB ISP Prog I	93
4.3 Przeprogramowanie bootloadera	94
Dodatek 1 Programator STK200/300 ISP	97
Annex 2 Tablica kodów ASCII	98
Annex 3 Wyprowadzenia wyświetlacza 7-segmentowego	101
Annex 4 Schemat płytki EDB	102
Prawa autorskie	104

Wstęp

W latach '80 ubiegłego wieku mikrokontrolery były stosunkowo drogie, przez co stosowane były w komputerach oraz elektronice domowej (sprzęt TV czy audio). Z powodu dużego popytu na prostą elektronikę, mikrokontrolery stały się łatwiej dostępne. Obecnie mikrokontrolery są dostępne dla każdego zainteresowanego elektroniką.

Edukacyjny zestaw EDB (Educational Development Board) jest wprowadzeniem do świata elektroniki i mikrokontrolerów. Zestaw EDB wraz z elektronicznymi elementami, pomoże Ci w nauce elektroniki i programowaniu mikrokontrolerów. Poznawanie elektroniki wspierane będzie przeprowadzonymi licznymi eksperymentami. Praktyczne eksperymenty, będą przebiegać w sposób logiczny, trzymając Cię w chęci do przeprowadzenia kolejnego eksperymentu.

W eksperymentach kładziony jest nacisk na współpracę zwykłych elementów elektronicznych jak rezystory, diody LED z mikrokontrolerami. Gdzie potrzebne jest proste wytłumaczenie działania elementu elektronicznego i mikrokontrolerów.

Nie musisz się martwić jeżeli nie rozumiesz mikrokontrolerów. W pierwszych eksperymentach, mikrokontrolery będą uważane za „czarne skrzynki”. Więcej o mikrokontrolerach dowiesz się z dalszych eksperymentów oraz Internetu z podanych linków.

Chcemy również wspomnieć o EDB-CD, która jest dostarczona z tą instrukcją. Na CD możesz znaleźć źródła programów dla eksperymentów, ćwiczenia związane z językiem Basic, dane katalogowe elementów elektronicznych i noty katalogowe. Całą zawartość CD możesz znaleźć w pliku “Contents of CD.pdf”, znajdującym się w głównym katalogu CD.

MCS Electronics życzy dużo zabawy, kiedy będziesz eksperymentować. Czekamy także na wszystkie Twoje komentarze, także o innych naszych produktach. Dla dodatkowych informacji proszę kontaktować się z Nami odwiedzając stronę www.mcselec.com.

1. Doświadczenia

Rozdział ten zawiera informacje jakie są wymagane umiejętności, jeśli zamierzasz rozpocząć zabawę z Edukacyjnym Zestawem EDB.

Zestaw Edukacyjny przeznaczony jest dla wszystkich, którzy mają małe lub nie mają doświadczenia z użyciem mikrokontrolerów. Ale powinieneś mieć jakieś doświadczenie w montowaniu i lutowaniu „elektronicznych zestawów”, więc możesz sam zbudować własny Zestaw Edukacyjny EDB. Wymagana jest także podstawowa znajomość obsługi komputera.

1.1 Wymagania sprzętowe

Aby uruchomić Bascom AVR wymagany jest komputer PC z zainstalowanym systemem Windows 95, 98, NT, 2000 lub XP. W tej instrukcji do zaprogramowania mikrokontrolera wykorzystywany będzie szeregowy port komputera. Jeśli nie posiadasz w komputerze portu szeregowego, możesz zaprogramować mikrokontroler programatorem STK200/300 dołączanym do portu LPT lub programatorem USB ISP Prog I dołączanym do portu USB. Jak użyć programatora STK200/300 lub USB ISP Prog I znajdziesz w rozdziale 4.



1.2 Uważaj na nie zaprogramowane mikrokontrolery!

Mikrokontrolery ATmega88 dostarczane przez MCS Electronics są już zaprogramowane aby umożliwić szeregowe programowanie. Jeśli (przypadkowo) skasujesz albo kupisz nowy mikrokontroler, będziesz potrzebował zaprogramować go programatorem STK200/300 lub USB ISP Prog I. W rozdziale 4 znajdziesz informacje jak to zrobić.

2. Pierwsze kroki

Rozdział ten zawiera informacje jak rozpocząć eksperymenty z zestawem EDB.

Zalecane jest wykonywanie punktów z rozdziału 2, krok po kroku. Postępowanie w ten sposób umożliwi rozpoczęcie eksperymentów w rozdziale 3, bez jakichkolwiek niespodzianek czy problemów.

2.1 Montaż Twojego zestawu EDB

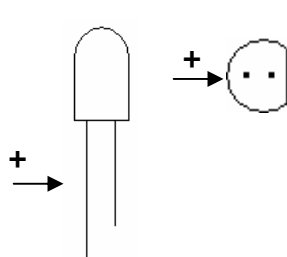
Rozdział 2.1 pomoże Ci w montażu Twojego zestawu EDB. Lutowanie elementów powinno następować zgodnie z wykazem elementów zamieszczonym na stronie 8.

Na PCB znajdziesz pola lutownicze typu:

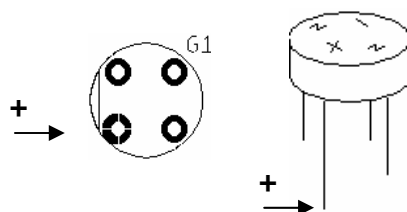


Kiedy lutujesz układ scalony IC, kwadratowe pola oznaczają nóżkę numer 1. Dla diod LED i kondensatorów kwadratowe pole oznacza dodatni biegun (+). Tutaj jest rysunek, który pomoże w zrozumieniu polaryzacji elementów elektronicznych:

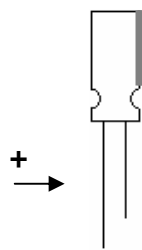
DIODA LED



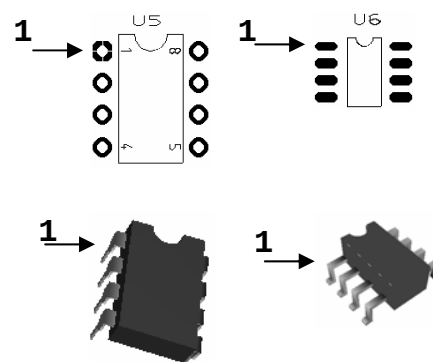
MOSTEK PROSTOWNICZY



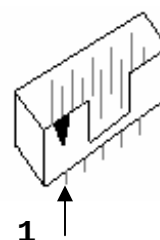
KONDENSATOR



IC



ZŁĄCZE



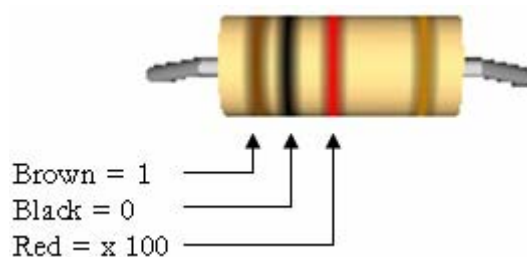
REZYSTORY

Rezystory są elementami nie biegunowymi. Więc nie ma znaczenia, która nóżka jest łączona do pola płytki PCB. Jednak potrzebujesz umieścić właściwą wartość rezystora, we właściwym miejscu płytki PCB.

Aby znaleźć wartość rezystora, możesz użyć omomierza albo odczytać ją z kolorowego kodu, który jest drukowany na rezystorze.

Kod koloru dla rezystorów:	Wartość pierwszego i drugiego paska	Wartość trzeciego paska, który jest mnożnikiem
Czarny	0	x1
Brown	1	x10
Czerwony	2	x100
Pomarańczowy	3	x1000
Żółty	4	
Zielony	5	
Niebieski	6	
Fioletowy	7	
Szary	8	
Biały	9	

Przykład jak przeczytać kolorowy kod:



A zatem $1000 \text{ OHM} = 1\text{kOHM}$

Czwarty pasek jest paskiem tolerancji. Żółty oznacza że rezystor posiada tolerancję 5 %.

WYKAZ ELEMENTÓW

Element	Opis	Wartość
R1,R5	REZYSTOR 1/4 W	330E
R4	REZYSTOR 1/4 W	220E
C3,C4,C9,C10, C11,C12,C13	KONDENSATOR CERAMICZNY	100N
R2	REZYSTOR 2W	47E
U1	PODSTAWKA DIP28	for U1
U3	PODSTAWKA DIP16	for U3
U4	STABILIZATOR	7805
R3	POTENCJOMETR	10K
G1	MOSTEK PROSTOWNICZY	
D1,D2,D3	DIODA LED	3mm
X15	ZŁĄCZE SW. UART	1x2
X5,X6, X7	ZŁĄCZE ŻEŃSKIE	2X5
X3,X4	ZŁĄCZE ŻEŃSKIE	1X24
X12	ZŁĄCZE ŻEŃSKIE LCD	1X16
X13	GNIAZDO USB	TYPU B
X9	GNIAZDO ISP	2X10
X1	ZASILANIE	ZŁĄCZE ZASILANIA
C5,C6,C7,C8	KONDENSATOR ELEKT.	1U
C2	KONDENSATOR ELEKT.	100U
C1	KONDENSATOR ELEKT.	220U
RESET	PRZYCISK RESET	SW
X16	GNIAZDO ŻEŃSKIE DB9	DB9FL
X2	ZASILANIE	ZŁĄCZE 1X2
X10	ZŁĄCZE KŁAWIATURY PS2	MINIDIN6
S1	PRZEŁĄCZNIK USB/UART	SW
-	WIELKOŚĆ PŁYTKI	7x7 CM

Nie montuj jeszcze elementów:

1x ATmega88 (U1) + 1x USB MODUŁ SDM-USB-QS1-S (U2)

ZALECENIA

Pod układy scalone IC, zalecane jest użycie podstawek, niż ich lutowanie do PCB. Można wykorzystać niskiej jakości podstawki, ale zalecane jest używanie wysokiej jakości podstawek.

Moduł USB może zostać wlutowany, **ale nie lutuj go jeszcze. Również w tym czasie nie wkładaj jeszcze układów scalonych IC do podstawek.**

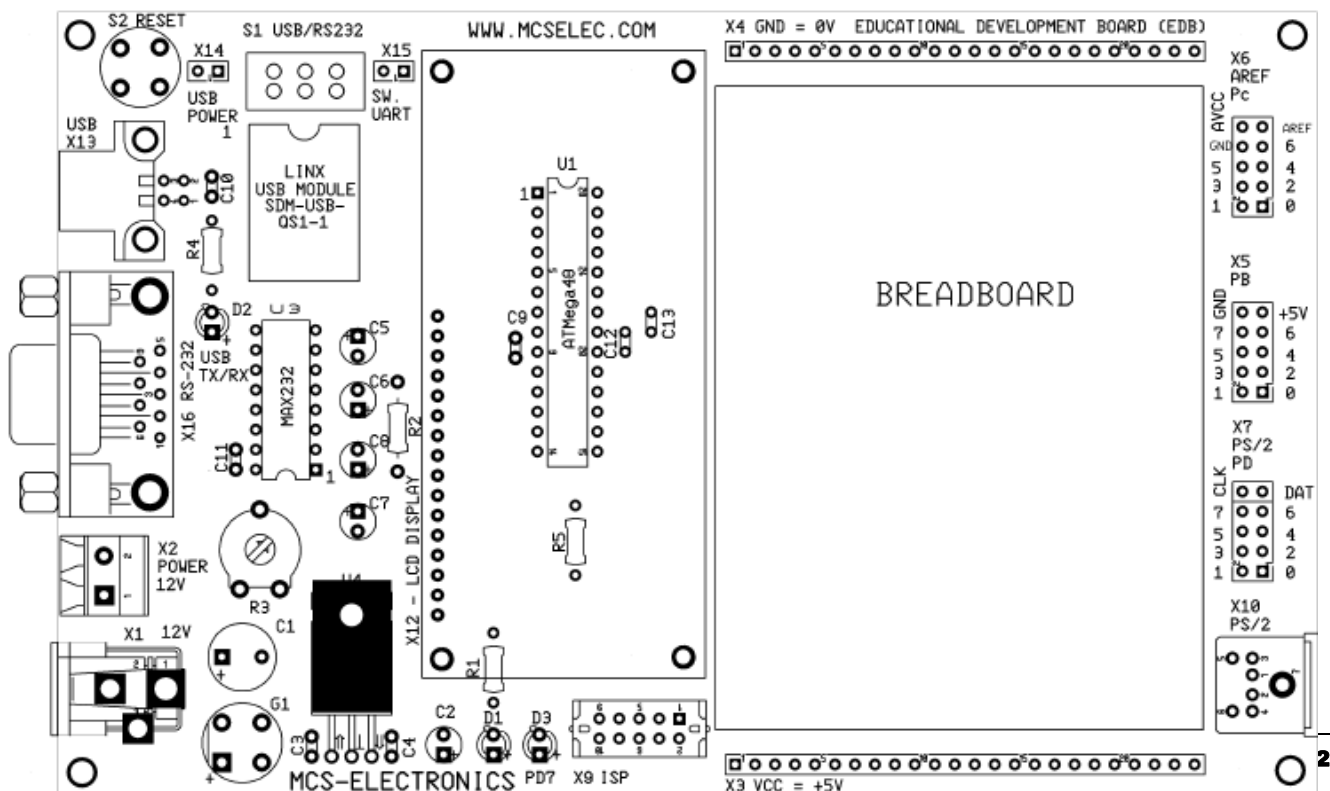
OPCJE ZASILANIA

Są dwie opcje zasilania, możesz zastosować zewnętrzne zasilanie dołączane do X2 lub zasilacz z odpowiednią wtyczką dołączany do złącza X1. Możesz także wykorzystać zasilanie z USB, wtedy musisz połączyć razem pola lutownicze X14. Jeśli użyjesz zasilania z USB, nie możesz zamontować stabilizatora U4 LM7805 i nie możesz wykorzystać do zasilania złącza X1 lub X2.

Pamiętaj, aby nie pobierać z USB dużego prądu. (MSC Electronic nie jest odpowiedzialny za możliwe uszkodzenie Twojego komputera z powodu wykorzystania zasilania z USB)

PŁYTKA DRUKOWANA (PCB)

Aby znaleźć właściwą pozycję elementu, możesz wykorzystać poniższy rysunek płytki drukowanej. Pozycje elementów są także namalowane na PCB, natomiast w większym powiększeniu rysunek PCB dostępny jest w dodatku 4.



PRZED PODŁĄCZENIEM ZASILANIA

Jeśli zlutowałeś całą płytkę, sprawdź czy pomiędzy połączeniami nie powstały zwarcia. Jeśli są, musisz je usunąć.

Zmierz rezystancje (oporność) pomiędzy wyprowadzeniami:

X2.1 i X2.2 powinien być około 18M OHM

X3 i X4 powinien być około 2,6k OHM.

Jeśli zmierzysz rezystancje około 0 OHM, na płytce posiadasz gdzieś zwarcie. Nie włączaj wtedy zasilania! Sprawdź ponownie płytkę drukowaną czy nie ma zwarć oraz sprawdź czy prawidłowo umieściłeś elementy U4 i G1.

Jeśli zmierzysz nieskończoną rezystancję, prawdopodobnie brak jest połączenia ze złączami X2, X3, lub X4. Spróbuj ponownie je przylutować.

Jest możliwe, że powyższe wartości rezystancji w Twoim EDB będą się różnić, ale nie powinny być większe niż $\pm 10\%$.

Jeśli pomiary przebiegły pomyślnie, możesz do EDB podłączyć zasilacz (lub zasilanie z USB). Zewnętrzne zasilanie (nie z USB) musi się zawierać w przedziale 9-15 V DC. Polaryzacja napięcia nie ma znaczenia, gdyż wykorzystany został mostek prostowniczy.

Po podłączenia zasilania powinna się zaświecić dioda LED. Zmierz napięcie między wyprowadzeniami 2 i 3 podstawki U4. Zmierzone napięcie musi wynosić 5 V.

Jeśli napięcie nie jest poprawne, odłącz zasilanie i sprawdź płytkę EDB.

Jeśli napięcie jest prawidłowe, odłącz je i zamontuj układ seryjnego bufora MAX232. Zamontuj również mikrokontroler ATmega88 i wlutuj moduł USB.

Przylutuj także do modułu LCD męskie złącze. Moduł wyświetlacza LCD możesz umieścić w złączu X12 (do złącza X12 należy dołączać LCD tylko z męskim złączem).

2.2 Instalacja Bascom AVR

Aby zaprogramować mikrokontroler ATMega88 w zestawie edukacyjnym EDB, potrzebujesz Bascom AVR. Możesz wybrać pomiędzy darmową wersją Bascom AVR, która umożliwia kompilację programu do 4 kB lub pełną wersją. Obie wersje są dostępne na stronie www.mcselec.com.

Procedura instalacji dla pełnej wersji

Pełna wersja dostępna jest na CD. Po włożeniu CD do napędu instalacja rozpocznie się automatycznie. W przeciwnym przypadku należy uruchomić plik setup.exe. Możesz następnie przejść do strony 11.

Procedura instalacji dla wersji demo

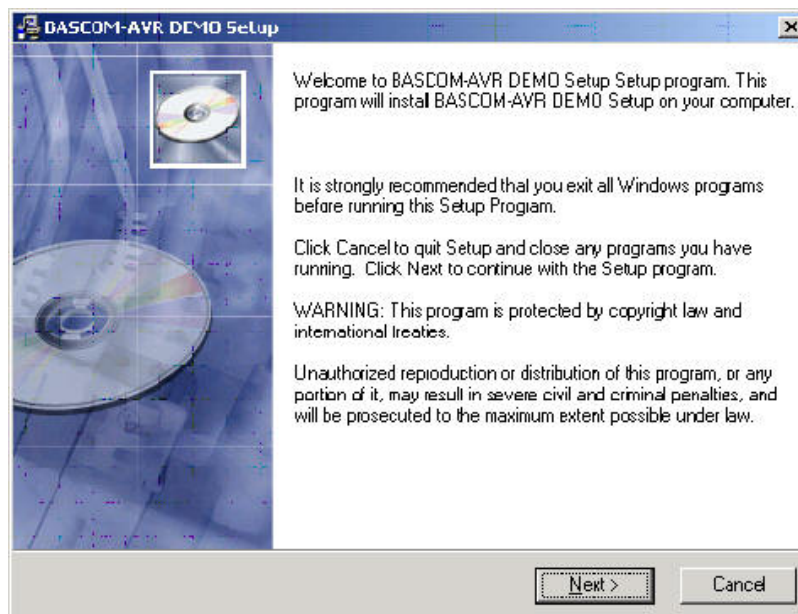
- Należy ściągnąć pliki instalacyjne (bcavrd.zip) ze strony www.mcselec.com.
- Domyślnie w Windows XP możesz otworzyć pliki ZIP. Wszystkie pliki należy rozpakować do tego samego katalogu. Jeśli nie możesz otworzyć pliku ZIP, możesz zainstalować program WinZip. (Dostępny na stronie www.winzip.com)

Nie uruchamiaj setup.exe w pliku ZIP!

- Po rozpakowaniu w katalogu będziesz posiadał plik wykonywalny setup.exe.
- Uruchom plik setup.exe, aby rozpocząć instalację.

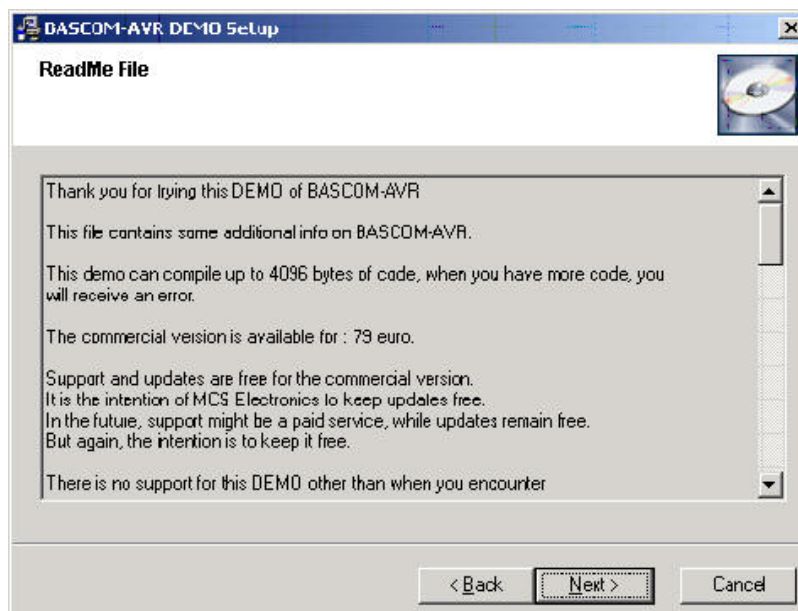
Przejdź do następnej strony.

Następnie wyświetlone zostanie okno:



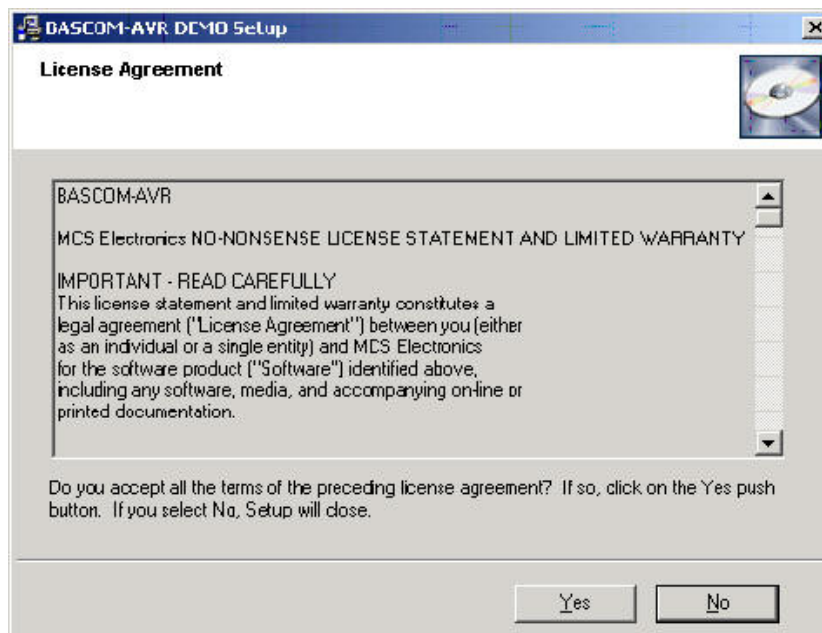
Naciśnij przycisk Next.

Następnie wyświetlone zostanie okno:



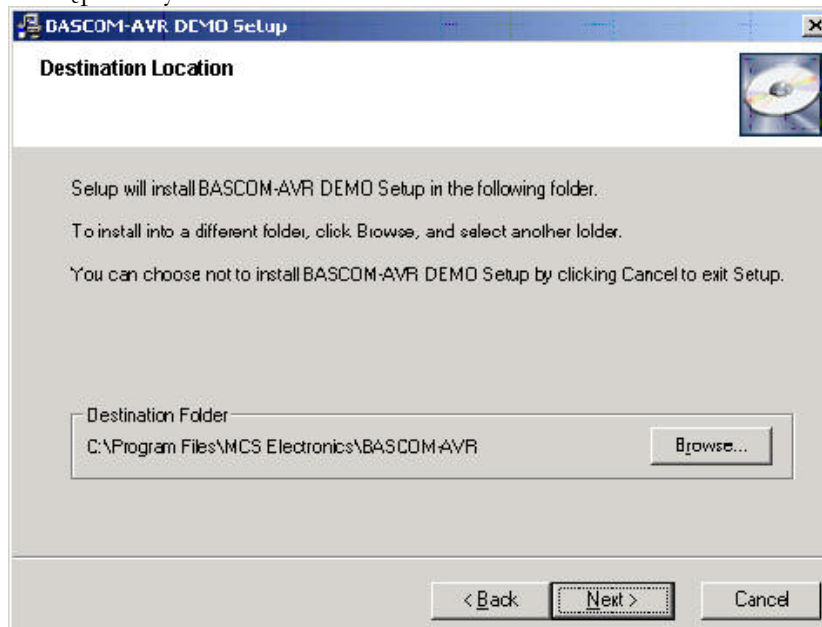
Naciśnij przycisk Next.

Następnie wyświetlone zostanie okno:



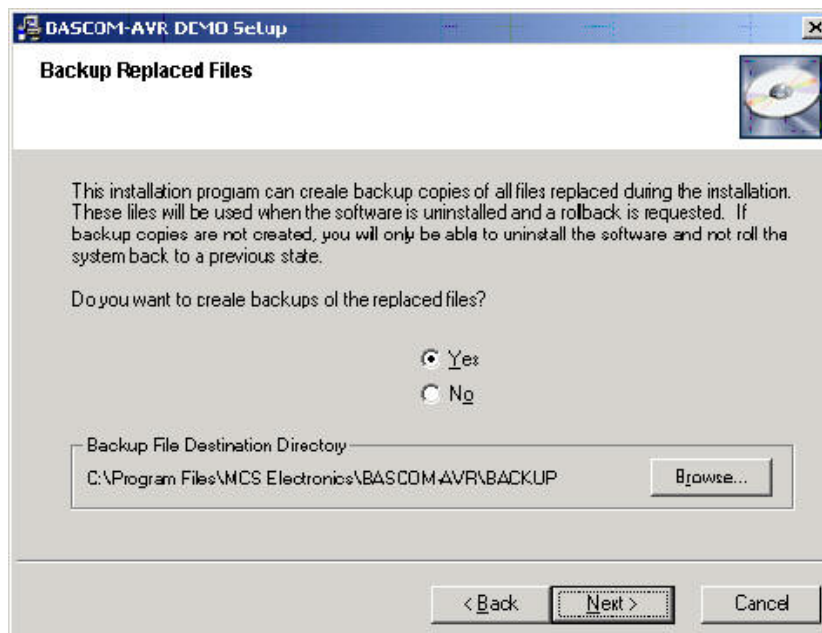
Naciśnij przycisk Yes.

Następnie wyświetlone zostanie okno:



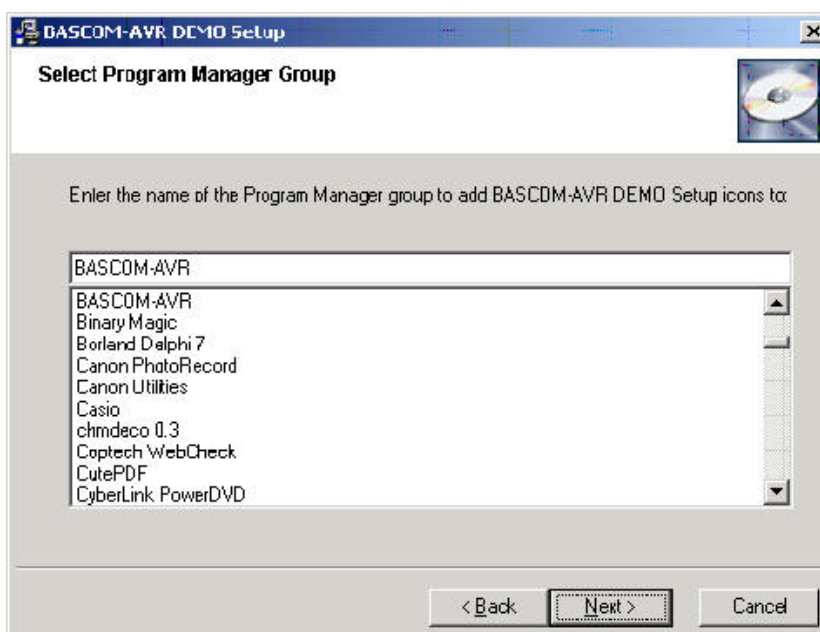
Możesz wybrać dysk oraz katalog. Możesz też nacisnąć przycisk Next dla domyślnego dysku i katalogu. Naciśnij przycisk Next, gdy wybrałeś docelowy katalog instalacji.

Następnie wyświetlone zostanie okno:



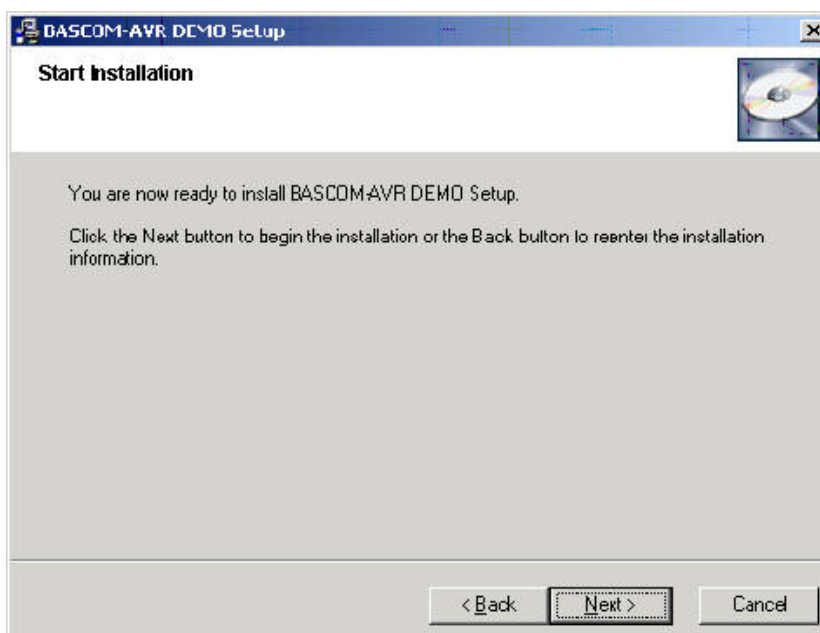
W tym oknie możesz wybrać, czy chcesz stworzyć kopię plików, które będą zastąpione. Normalnie, gdy pliki nie istnieją (przy pierwszej instalacji), nie będą zastąpione innymi. Ale wybierz opcję Yes i kliknij na przycisku Next.

Następnie wyświetlone zostanie okno:

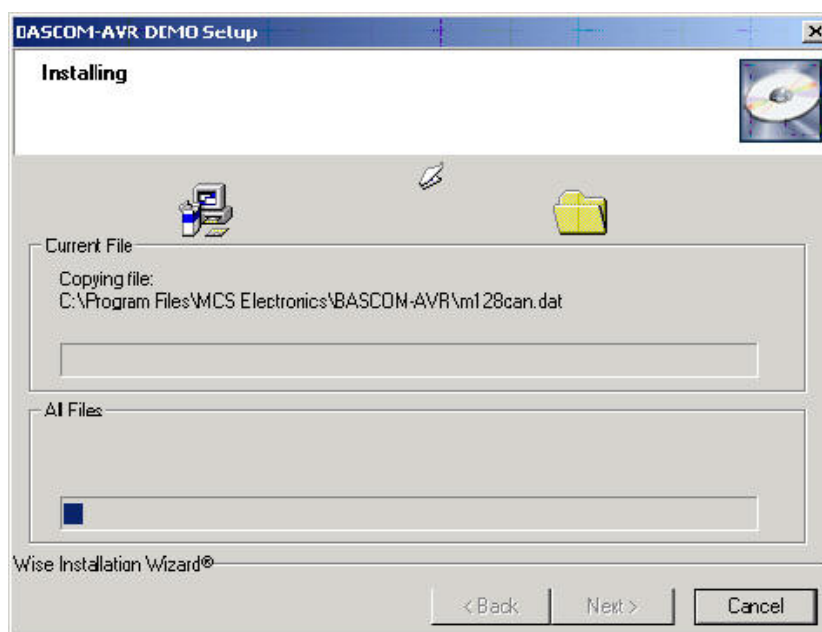


Możesz wybrać w jakiej grupie program BASCOM zostanie zainstalowany. Naciśnij przycisk Next, gdy wybrałeś grupę programów.

Następnie wyświetlone zostanie okno:



Program instalacyjny ma teraz wszystkie informacje, aby rozpocząć instalację. Naciśnij przycisk Next, aby rozpocząć instalację.



Podczas instalacji wyświetlane jest powyższe okno.

BASCOM-AVR jest teraz gotowy do użycia.

Kiedy instalujesz BASCOM w Windows NT, Windows 2000 lub XP, będziesz potrzebował praw administratora. Prawa administratora są wymagane tylko podczas instalacji. Po instalacji możesz korzystać z niego jako inny użytkownik.

2.3 Programowanie mikrokontrolera

Przez port szeregowy (procedura Bootloader)

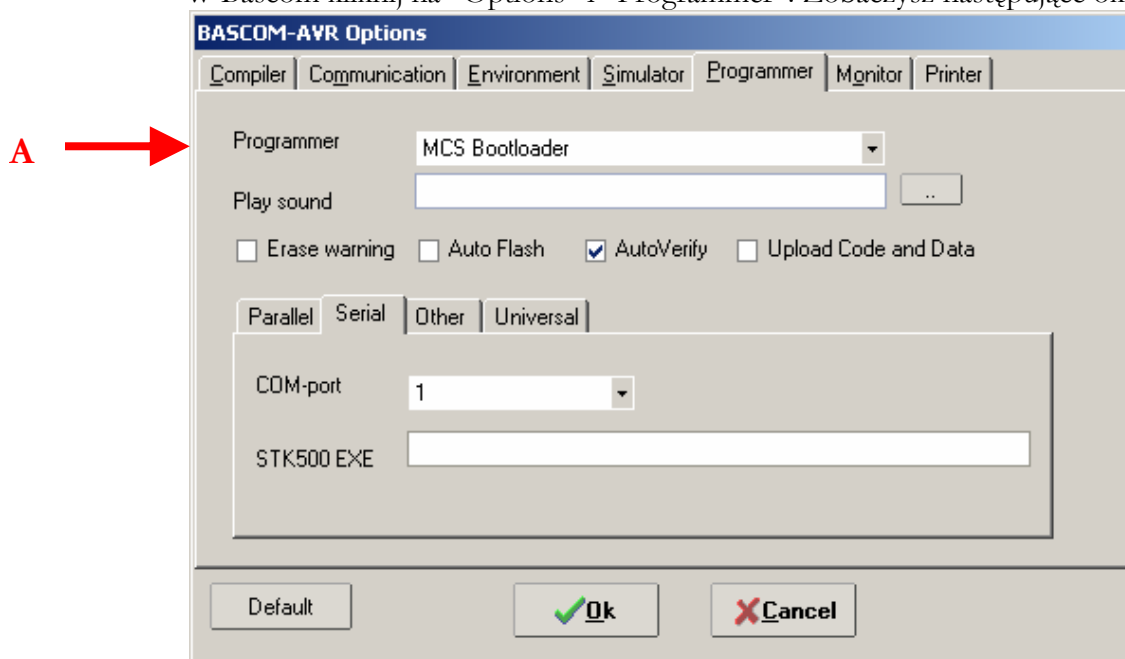
Ten rozdział opisuje jak możesz skonfigurować Bascom, aby było możliwe zaprogramowanie mikrokontrolera przez port szeregowy.

Metoda, którą zaprogramujesz mikrokontroler przez port szeregowy nazywana jest “Bootloader”. Bootloader jest kawałkiem programu, który załaduje Twój program do pamięci mikrokontrolera (Bootloader również jest umieszczony w pamięci mikrokontrolera, ale MCS Electronics zatroszczyła się już o to).

Do programowania EDB przez port szeregowy możesz użyć szeregowego przewodu, który dostarczany jest razem z zestawem. Ale możesz użyć również innego prostego przewodu, w którym przewody są połączone następująco: linia 2->2, linia 3->3, linia 4->4 i linia 5->5.

Jeden koniec przewodu należy dołączyć do portu szeregowego komputera, a drugi koniec do złącza X16 (DB-9) oznaczonego RS-232 w edukacyjnym zestawie EDB.

W Bascom kliknij na “Options” i “Programmer”. Zobaczysz następujące okno:



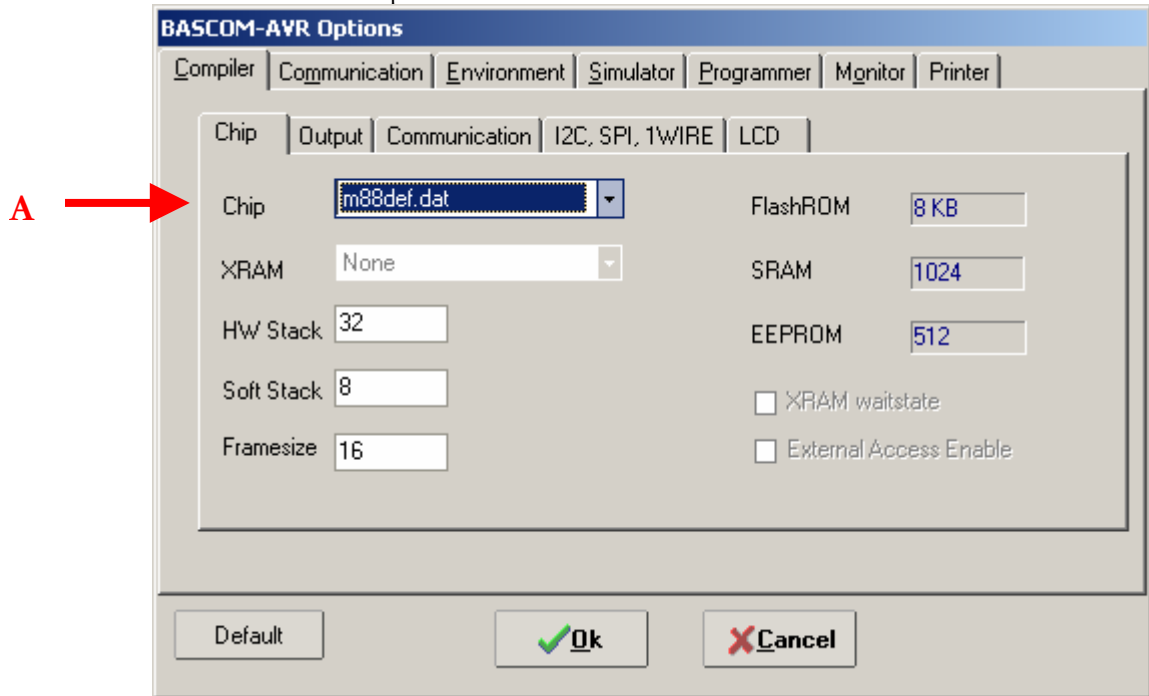
- Wybierz “MCS Bootloader” z menu oznaczonego literą A
- Wybierz prawidłowy port COM i naciśnij przycisk Ok.

“Teraz zaprogramujemy pamięć mikrokontrolera”

Zanim zaprogramujesz mikrokontroler, w pierwszej kolejności musisz otworzyć i skompilować program testowy. Program testowy możesz znaleźć na płycie EDB-CD. Otwórz plik EDBtest.bas w Bascom AVR. (kliknij “File” > “Open”)

Teraz otwórz menu opcji (“Options” > “Compiler” > “Chip”)

Powinieneś widzieć poniższe okno:



- Wybierz m88def.dat z menu “Chip” (oznaczonego literą A).
- Nie zmieniaj innych opcji i naciśnij przycisk OK
- Teraz naciśnij klawisz F7, aby skompilować testowy program
- Naciśnij klawisz F4, aby automatycznie zaprogramować mikrokontroler.

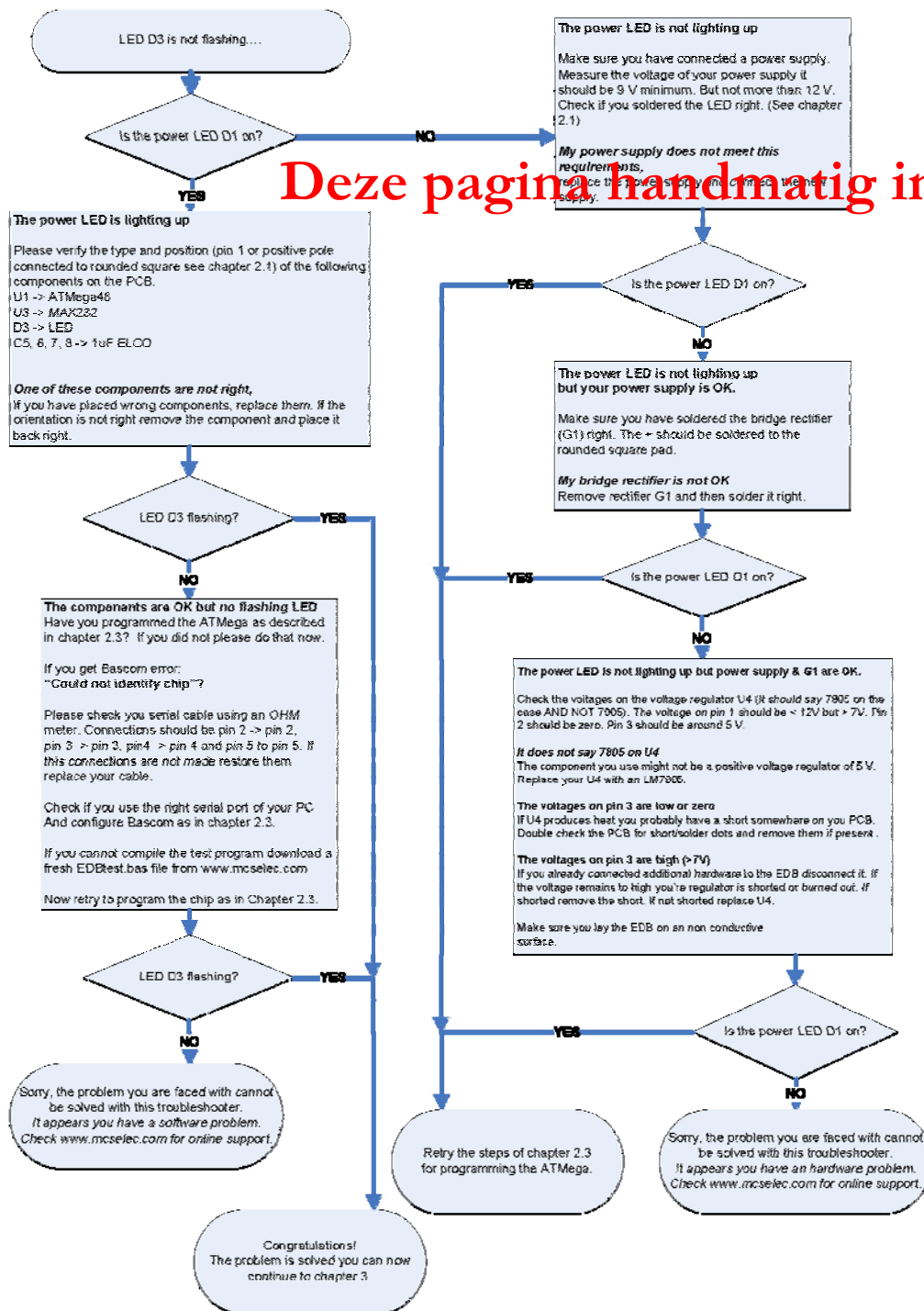
Jeśli wszystko poszło dobrze, powinieneś zobaczyć migającą diodę LED.

Jeśli używasz mikrokontroler ATmega88, który nie jest dostarczany przez MCS Electronics albo zmieniłeś Fuse bity, przeczytaj rozdział 4.3 przed kontaktem z Nami.

- Czy widzisz migającą diodę LED D3 (PD7)?

Jeśli odpowiesz TAK, to gratulacje! Możesz przejść do rozdziału 3

Jeśli odpowiesz NIE, nie martw się, kontynuuj czytanie następnej strony.



Deze pagina handmatig invoegen

3. Eksperymenty

W tym rozdziale poznasz elementy elektroniki oraz mikrokontrolery.

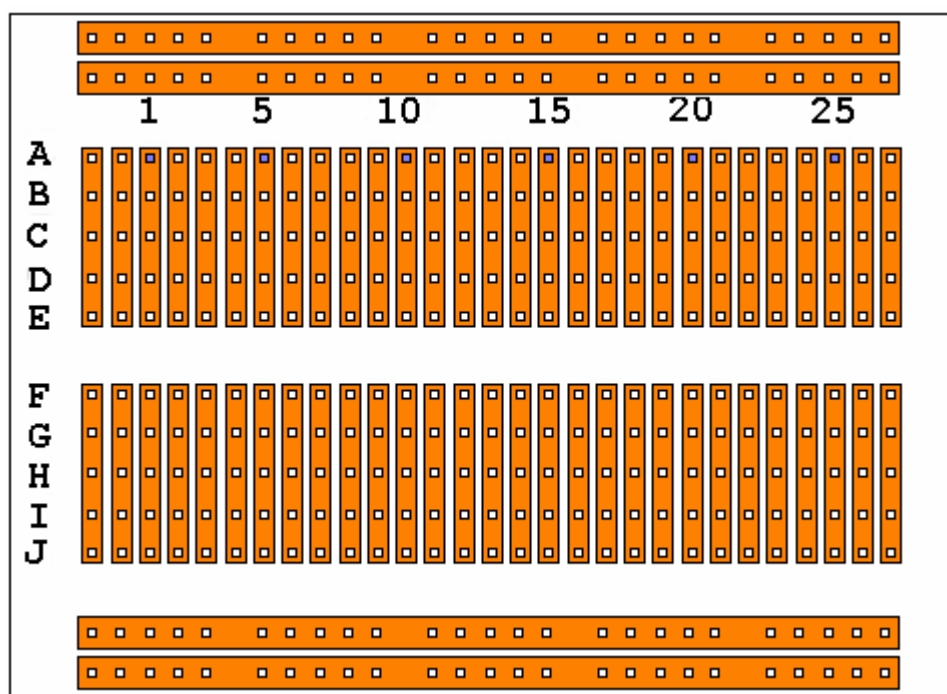
Złożyłeś i przetestowałeś własną płytkę EDB, więc możesz zacząć eksperymentować. Eksperymenty będą przebiegać w logiczny sposób trzymając Cię w zachęceniu do kolejnych eksperymentów.

Eksperymenty zostały podzielone na kategorie,

- Podstawowa obsługa linii I/O
- Szeregowa komunikacja z wykorzystaniem UART
- Obsługa wyświetlaczy (LCD, 7 segmentowy)
- Obsługa klawiatur (PS/2, matrycowa, odbiornik podczerwieni RC5)
- Zaawansowana obsługa linii I/O
- Przetworniki A/C i C/A
- Inne elementy mikrokontrolera
- Inne interfejsy (I²C, USB)
- Sterowanie silnikami

UNIWERSALNA PŁYTKA

Uniwersalna płytka służy do umożliwienia budowania bez lutowania połączeń pomiędzy elementami elektronicznymi podczas eksperymentów. Będziesz ją wykorzystywał podczas przeprowadzanych eksperymentów. Poniższy rysunek pokazuje połączenia jakie znajdują się w uniwersalnej płytce.



Kolorem **pomarańczowym** zaznaczone zostały wewnętrzne połączenia płytki. Możesz w niej umieszczać elementy, które posiadają odległość pomiędzy wyprowadzeniami nie mniejsze niż 2,54 mm. Przewody połączeniowe mogą posiadać średnice od 0,3 do 0,8 mm. Nigdy na siłę nie umieszczaj elementów i przewodów w uniwersalnej płytce.



UWAGA! RYZYKO POWAŻNEGO USZKODZENIA PŁYTKI

**Uniwersalna płytka nie jest odpowiednia do
dołączenia do sieci elektrycznej.**

Nigdy nie podłączaj uniwersalnej płytki do sieci elektrycznej

3.1 Podstawowa obsługa linii I/O

I/O oznacza “wejscie/wyjście”. Wejścia są łączone do wejściowych linii mikrokontrolera, a wyjścia są łączone do jego linii wyjściowych. (Dwukierunkowe linie w mikrokontrolerze mogą być skonfigurowane jako wejścia lub wyjścia). Obsługę linii I/O można pokazać na wiele sposobów, ale najłatwiejsze jest pokazanie sterowania wyjściem z użyciem diody LED oraz odczytu stanu wejścia z użyciem przycisku.

Eksperyment

1

3.1.1 Wyjście dla LED, wejście dla przycisku

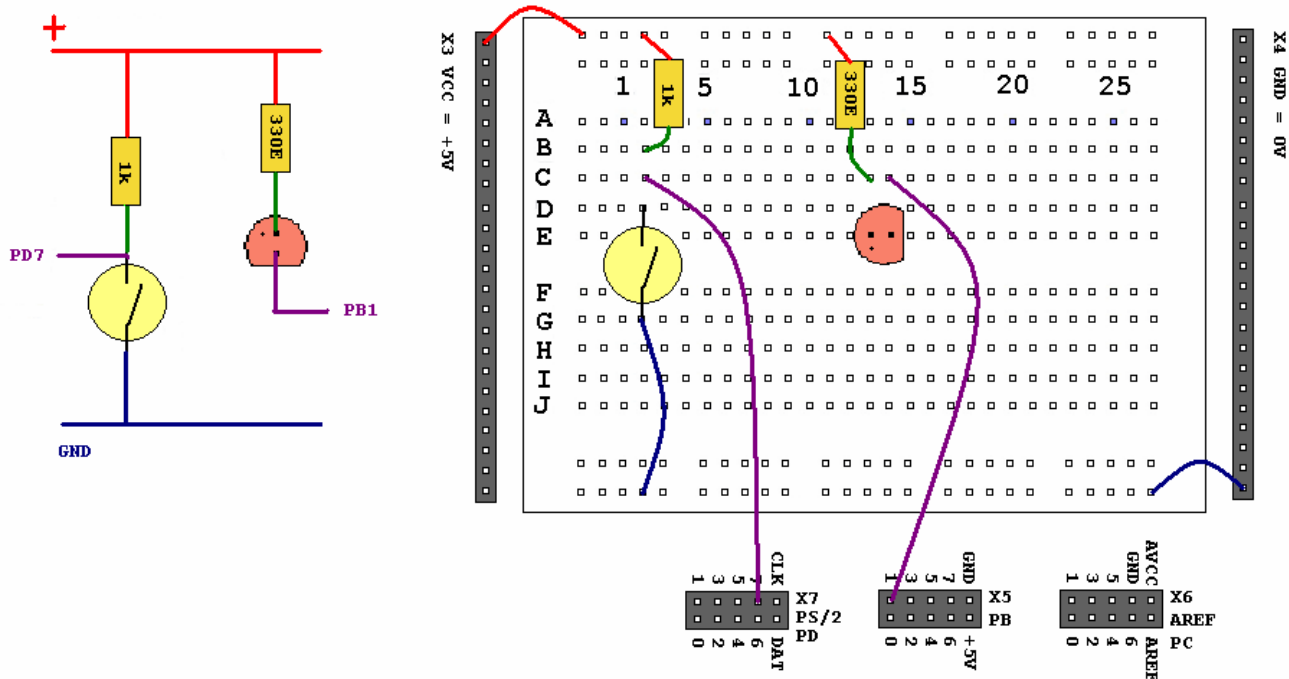
Lista elementów

- 1x Rezystor 1k
- 1x Rezystor 330E
- 1x dioda LED
- 1x Przycisk lub przewód
- 6x Przewody

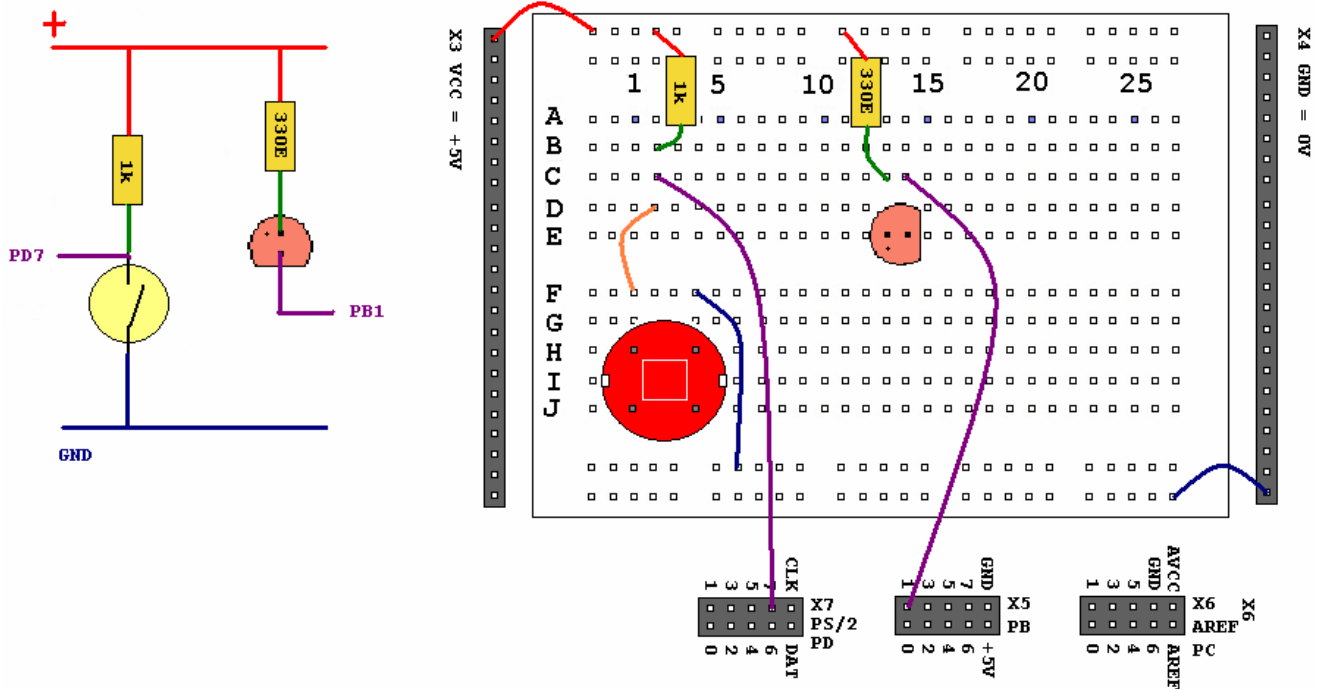
Cele

- Zrozumienie podstaw linii wejścia/wyjścia (I/O)

Teraz zbuduj poniżej pokazany układ. Jest pokazane jak dołączyć przycisk oraz diodę LED do mikrokontrolera. Jeśli nie masz przycisku, możesz użyć przewodu.



Możesz także użyć przycisku dostarczonego przez MCS Electronics, wtedy połączenia będą wyglądać jak poniżej:



Teraz otwórz plik EDBexperiment1.bas z płyty EDB-CD.
Program powinien wyglądać następująco:

```

EDBexperiment1.bas
Experiment 1 for the Educational Development Board
(c) 1995-2005, MCS Electronics
Fileversion 1.0

'Purpose:
'This program assigns the value of Pind.7 to Pinb.1.
'Conclusions:
'You should be able to switch the led with a key press.

$regfile = "m88def.dat"

Config Pind.7 = Input
Config Pinb.1 = Output

Do
  Portb.1 = Pind.7
Loop

End
    
```

'To tell Bascom which chip we use
'Configure Pin D7 as input
'Configure Pin B1 as output
'Do...loop will loop forever
'Assign the output with the input value

Program ten musisz umieścić w Twoim mikrokontrolerze podobnie jak robiłeś to z programem testowym w rozdziale 2.3 (*"Programowanie mikrokontrolera"*). Działanie programu możesz sprawdzić naciskając przycisk (gdy naciśniesz przycisk, dioda LED powinna zaświecić).

3.1.2 Biegające światło diod LED

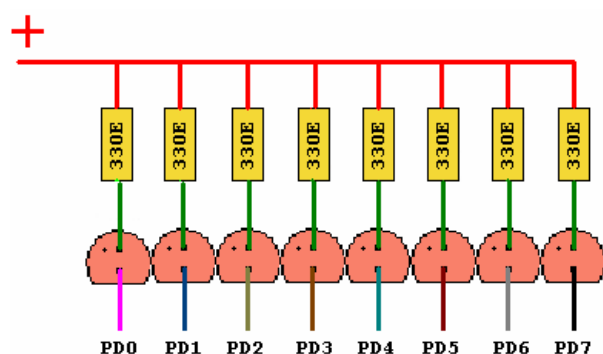
Lista elementów

- 8x Rezystor 330E
- 8x Dioda LED
- 9x Przewody

Cele

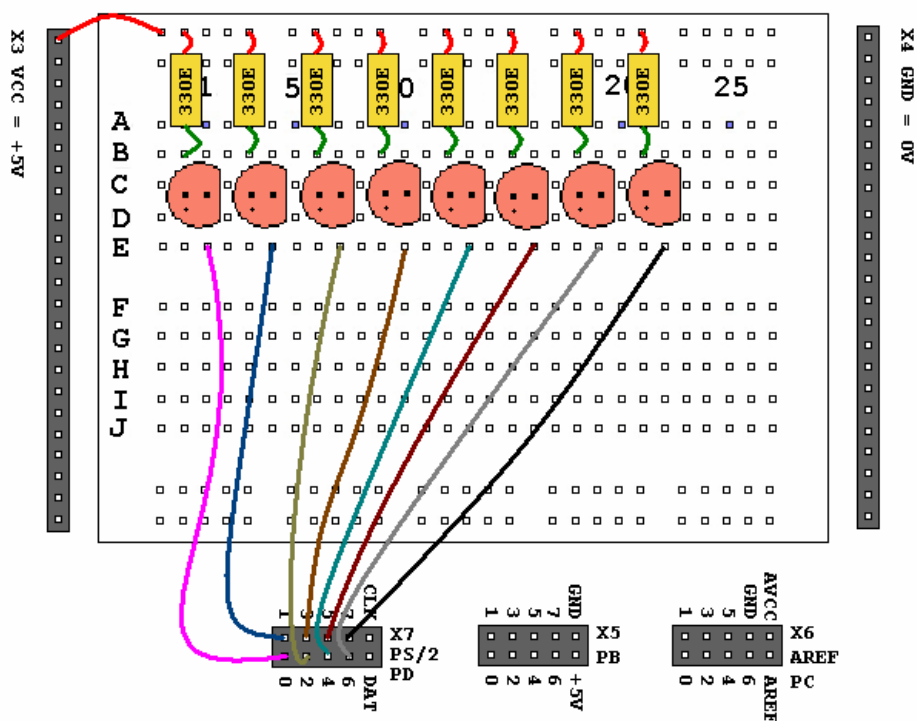
- Zrozumienie instrukcji przesunięcia,
- Zrozumienie linii I/O

Inny przykład z podstaw linii I/O, to operacja przesunięcia.



Nie musisz wykonywać
jakichkolwiek połączeń do masy.
Masa będzie dostarczana przez
mikrokontroler.

Sugerowane połączenia na uniwersalnej płytce:



Teraz otwórz plik EDBexperiment2.bas z płyty EDB-CD.
Program powinien wyglądać następująco:

```
'-----
'               EDBexperiment2.bas
'   Experiment 2 for the Educational Development Board
'   (c) 1995-2005, MCS Electronics
'   Fileversion 1.0
'-----
'Purpose:
'This program 'rotates' the LED's on port D.
'
'Conclusions:
'You should be able to see a moving LED effect.

$regfile = "m88def.dat"                'To tell Bascom which chip we use
Config Portd = Output                  'Configure Pin D7 as output
Portd = &B1111110                     'We start by making pd0 low (led on)

Do
  Rotate Portd , Right , 1              'Move the 'bits' one place to the right
  Waitms 100                            'Now PD1 will light and PD0 goes out.
Loop                                    'Wait otherwise we cannot see te rotate

End
```

Program ten musisz umieścić w Twoim mikrokontrolerze, podobnie jak robileś to z programem testowym w rozdziale 2.3 (*"Programowanie mikrokontrolera"*). Teraz widzisz jak diody LED zapalają się od lewej do prawej strony.

Zmień linię:

```
Rotate Portd , Right , 1
na:
Rotate Portd , Right , 2
```

Czy zauważyłeś różnice?

Teraz znajdź jak zmienić kierunek zapalania diod LED, aby zapalały się od prawej do lewej strony. Możesz znaleźć pomoc przez najechanie kursorem na komendę rotate i przyciśnięciu klawisza F1.

3.1.3a Wyjście sterujące tranzystorem

Lista elementów

1x Tranzystor BC547 NPN
 1x Dioda LED
 5x Przewody
 1x Rezystory 330, 47k ohm

Cele

- Podstawy zrozumienia elektrycznych parametrów mikrokontrolera
- Podstawy użycia tranzystora FET

Teoria

W eksperymentach 1 i 2 używane były “bezpośrednio linie I/O”, to znaczy że dioda LED była bezpośrednio dołączona do mikrokontrolera. Jest to proste rozwiązanie, ale niekorzystne. Po pierwsze z linii I/O mikrokontrolera nie możesz pobrać dużego prądu. Po drugie używanie linii I/O bezpośrednio, niesie ryzyko uszkodzenia mikrokontrolera przez wyładowania statyczne (ESD). (ESD krótko zostały opisane w rozdziale 3.1.5).

Prąd pobierany z linii mikrokontrolera nie może przekroczyć określonych wartości. Te wartości można znaleźć w dokumencie opisującym ATMEGA88 na stronie www.atmel.com. Wartości prądów można znaleźć w punkcie “**Electrical Characteristics**”. Przykład elektrycznych charakterystyk jest na kolejnej stronie.

Pierwszy podpunkt “**Absolute Maximum Ratings**” przedstawia maksymalne wartości parametrów, których przekroczenie spowoduje uszkodzenie mikrokontrolera. Tam nie ma niczego o pracy Twojej aplikacji. (Więc jeśli pobierzesz z jednej linii prąd do 40 mA, Atmel gwarantuje, że mikrokontroler nie ulegnie uszkodzeniu, ale nie ma gwarancji, gdy pobierany będzie większy prąd)

Jak dużo prądu możesz pobierać?

Jeśli pobierany jest prąd z linii, która ma poziom 0 V:

1. Suma prądu dla wszystkich linii portu C0 - C5, ADC7, ADC6 nie powinna przekroczyć 100 mA.
2. Suma prądu dla wszystkich linii portu B0 - B5, D0 - D4, D5 - D7, XTAL1, XTAL2, RESET nie powinna przekroczyć 100 mA.

Jeśli pobierany jest prąd z linii, która ma poziom 5 V:

1. Suma prądu dla wszystkich linii portu C0 - C5, D0 - D4, ADC7, RESET nie powinna przekroczyć 150 mA
2. Suma prądu dla wszystkich linii portu B0 - B5, D5 - D7, ADC6, XTAL1, XTAL2 nie powinna przekroczyć 150 mA.

Dla całego układu prąd nie powinien przekroczyć 200 mA (Absolute Max.)

Dla pełnych informacji proszę się odnieść do dokumentu opisującego mikrokontroler. Odtąd będą umieszczane informacje, gdzie w dokumentacji można znaleźć potrzebną informację.

27. Electrical Characteristics

27.1 Absolute Maximum Ratings*

Operating Temperature.....	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on any Pin except RESET with respect to Ground	-0.5V to $V_{CC}+0.5V$
Voltage on RESET with respect to Ground.....	-0.5V to +13.0V
Maximum Operating Voltage	6.0V
DC Current per I/O Pin	40.0 mA
DC Current V_{CC} and GND Pins.....	200.0 mA

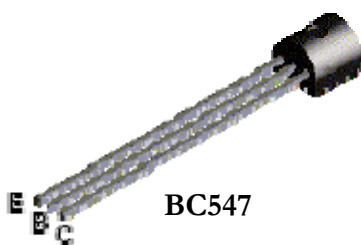
***NOTICE:** Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

27.2 DC Characteristics ATmega48/88/168*

* Note: The following table contains preliminary data for the ATmega88 and ATmega168.

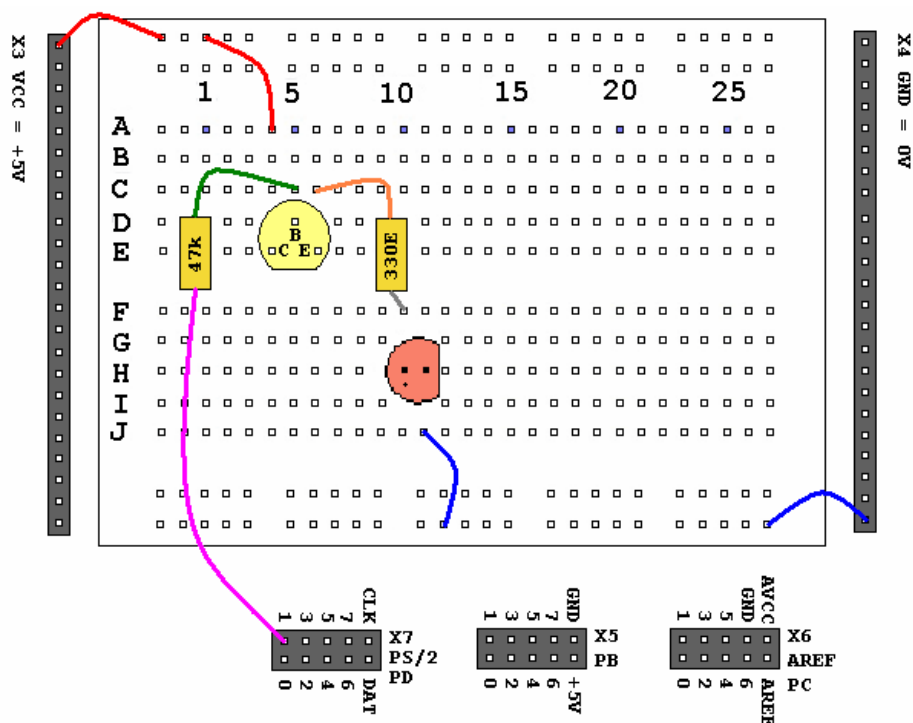
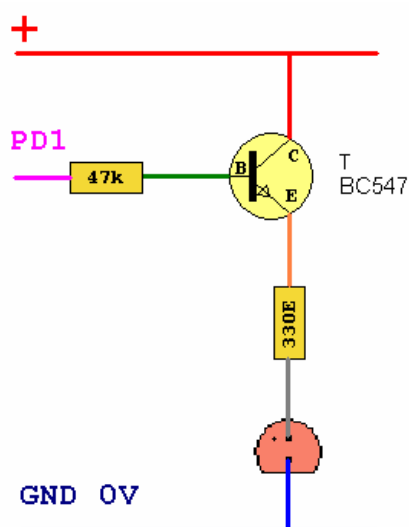
$T_A = -40^{\circ}C$ to $85^{\circ}C$, $V_{CC} = 1.8V$ to $5.5V$ (unless otherwise noted)

Symbol	Parameter	Condition	Min. ⁽⁵⁾	Typ.	Max. ⁽⁵⁾	Units
V_{IL}	Input Low Voltage, except XTAL1 and RESET pin	$V_{CC} = 1.8V - 2.4V$ $V_{CC} = 2.4V - 5.5V$	-0.5 -0.5		$0.2V_{CC}^{(1)}$ $0.3V_{CC}^{(1)}$	V
V_{IH}	Input High Voltage, except XTAL1 and RESET pins	$V_{CC} = 1.8V - 2.4V$ $V_{CC} = 2.4V - 5.5V$	$0.7V_{CC}^{(2)}$ $0.6V_{CC}^{(2)}$		$V_{CC} + 0.5$ $V_{CC} + 0.5$	V
V_{IL1}	Input Low Voltage, XTAL1 pin	$V_{CC} = 1.8V - 5.5V$	-0.5		$0.1V_{CC}^{(1)}$	V
V_{IH1}	Input High Voltage, XTAL1 pin	$V_{CC} = 1.8V - 2.4V$ $V_{CC} = 2.4V - 5.5V$	$0.8V_{CC}^{(2)}$ $0.7V_{CC}^{(2)}$		$V_{CC} + 0.5$ $V_{CC} + 0.5$	V
V_{IL2}	Input Low Voltage, RESET pin	$V_{CC} = 1.8V - 5.5V$	-0.5		$0.2V_{CC}^{(1)}$	V
V_{IH2}	Input High Voltage, RESET pin	$V_{CC} = 1.8V - 5.5V$	$0.9V_{CC}^{(2)}$		$V_{CC} + 0.5$	V
V_{IL3}	Input Low Voltage, RESET pin as I/O	$V_{CC} = 1.8V - 2.4V$ $V_{CC} = 2.4V - 5.5V$	-0.5 -0.5		$0.2V_{CC}^{(1)}$ $0.3V_{CC}^{(1)}$	V
V_{IH3}	Input High Voltage, RESET pin as I/O	$V_{CC} = 1.8V - 2.4V$ $V_{CC} = 2.4V - 5.5V$	$0.7V_{CC}^{(2)}$ $0.6V_{CC}^{(2)}$		$V_{CC} + 0.5$ $V_{CC} + 0.5$	V
V_{OL}	Output Low Voltage ⁽³⁾ , RESET pin as I/O	$I_{OL} = 20mA$, $V_{CC} = 5V$ $I_{OL} = 6mA$, $V_{CC} = 3V$			0.7 0.5	V
V_{OH}	Output High Voltage ⁽⁴⁾ , RESET pin as I/O	$I_{OH} = -20mA$, $V_{CC} = 5V$ $I_{OH} = -10mA$, $V_{CC} = 3V$	4.2 2.3			V
V_{OL3}	Output Low Voltage ⁽³⁾ , RESET pin as I/O	TBD			TBD	V
V_{OH3}	Output High Voltage ⁽⁴⁾ , RESET pin as I/O	TBD	TBD			V



BC547

Możesz użyć tranzystora, aby pracować z większym prądem, niż dostarczany przez mikrokontroler. Poniższy rysunek pokazuje jak użyć tranzystora jako przełącznika



Otwórz plik EDBexperiment2.bas z płyty EDB-CD i załaduj program do mikrokontrolera ATmega88.

Zasoby internetowe:

BC547: <http://www.fairchildsemi.com/pf/BC/BC547.html>

ATmega88: http://www.atmel.com/dyn/products/product_card.asp?part_id=3302

Eksperyment 3b

3.1.3b Wyjście sterujące tranzystorem FET

Lista części

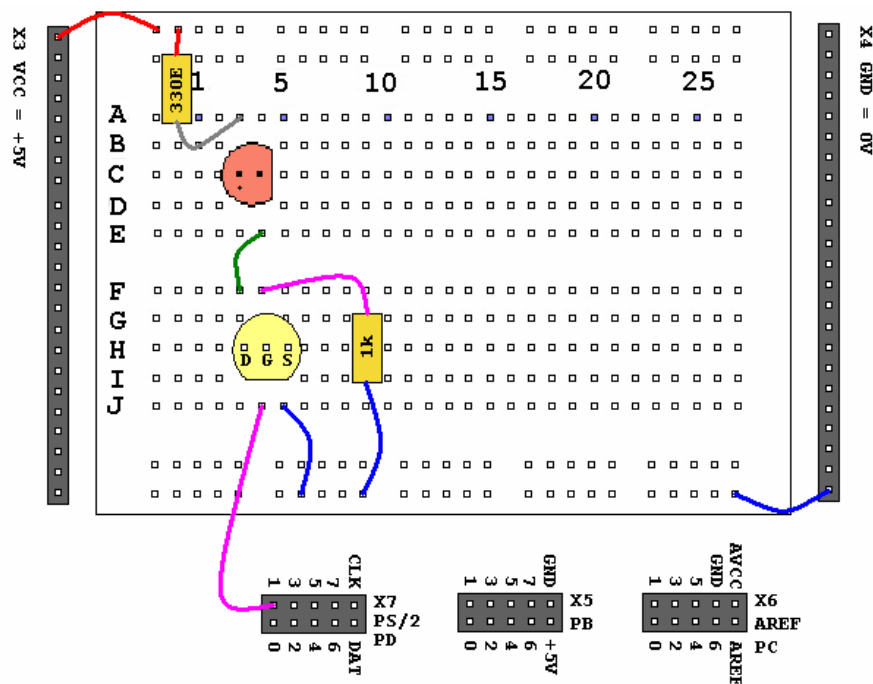
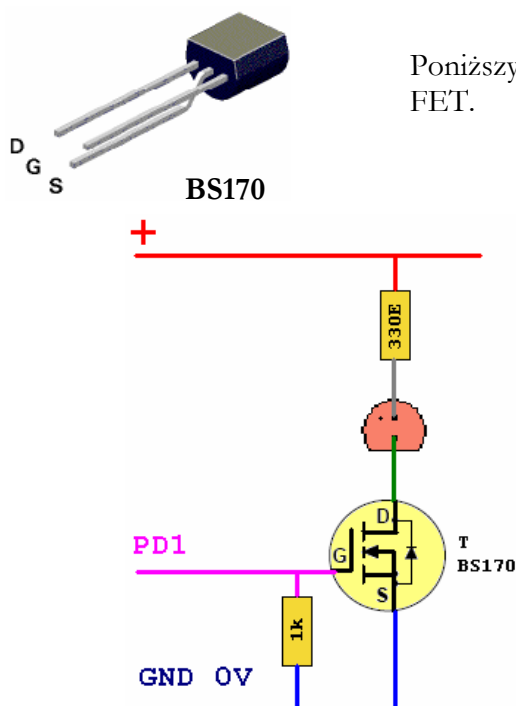
- 1x BS170 N-kanalowy FET
- 1x Dioda LED
- 5x Przewody
- 1x Rezystory 330 i 47k ohm

Cele

- Podstawy użycia tranzystora FET

Użycie MOSFETa, zamiast klasycznego tranzystora BC547, daje kilka korzyści. Tranzystor MOSFET może dostarczać więcej prądu, a jego bramka pobiera prąd bliski 0 mA.

Poniższy rysunek pokazuje sposób dołączenia do mikrokontrolera tranzystora FET.



Jeśli zaprogramowałem mikrokontroler ATmega88 w eksperymencie 3a, nie programuj go ponownie. Jeśli tego nie zrobisz, otwórz plik EDBexperiment3.bas z płyty EDB-CD i zaprogramuj mikrokontroler ATmega88. Nie usuwaj elementów z uniwersalnej płytki.

Zasoby internetowe:

BS170: <http://www.fairchildsemi.com/pf/BS/BS170.html>

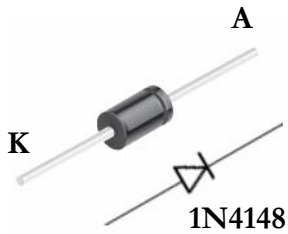
3.1.3c Wyjście sterujące przekaźnikiem

Lista elementów

- 1x BS170 N-kanalowy FET
- 1x Przekaźnik (Schrack PE014-005)
- 7x Przewody
- 1x Rezystor 1k ohm
- 1x Dioda 1N4148

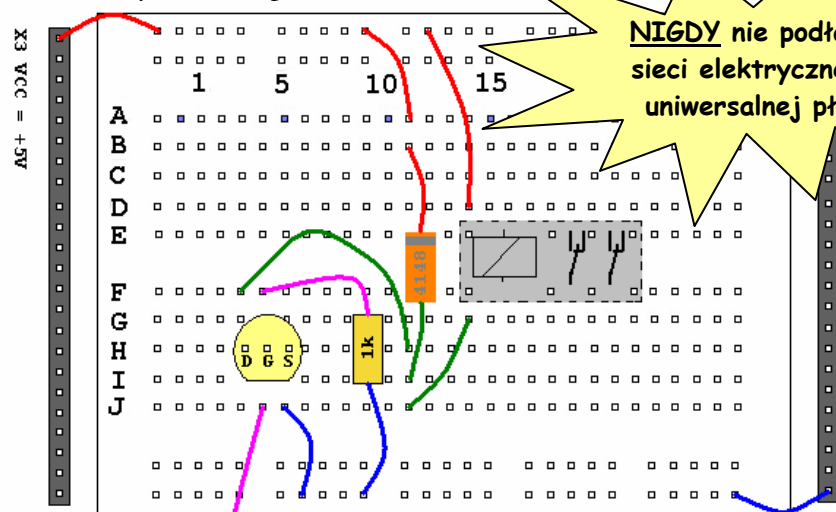
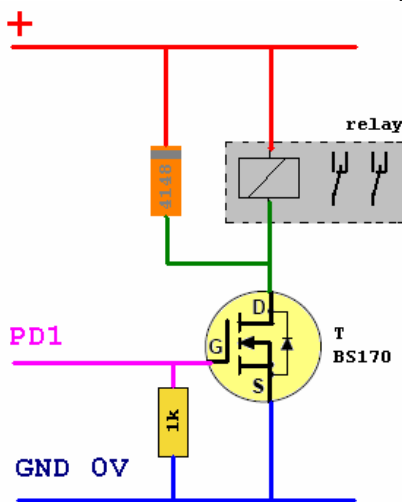
Cele

- Nauczysz się jak użyć przekaźnika



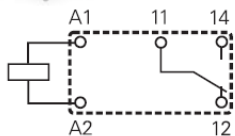
MOSFETy oraz tranzystory klasyczne są elementami półprzewodnikowymi. Te których użyłeś w eksperymentach mogą przełączyć tylko napięcia stałe (DC). Możesz użyć przekaźnika, aby uzyskać dodatkowe odseparowane styki od układu. Umożliwi to przełączanie sygnałów zmiennych (AC) oraz prądów z sieci elektrycznej.

Potrzebny jest FET, aby załączać przekaźnik.



Przykład przekaźnika

Dostępnych jest wiele innych typów przekaźników!



Jeśli zaprogramowałeś mikrokontroler ATmega88 w eksperymencie 3a lub 3b, nie programuj go ponownie. Jeśli tego nie zrobiłeś, otwórz plik EDBexperiment3.bas z płyty EDB-CD i zaprogramuj mikrokontroler ATmega88.

Zasoby internetowe:

Przekaźniki: <http://relays.tycoelectronics.com/>

3.1.4 Wyjście sterujące układem ULN2803

Lista elementów

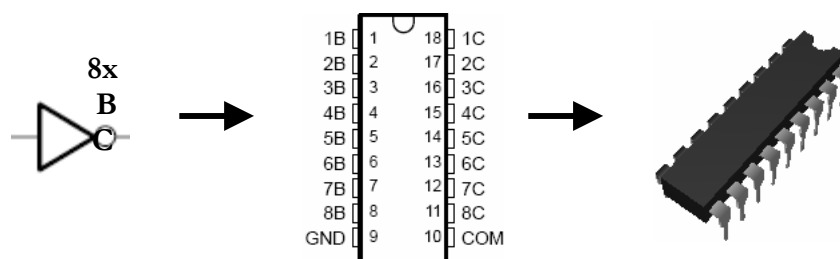
1x Układ ULN2803 lub odpowiednik
 2x Dioda LED
 7x Przewody
 2x Rezystor 330E

Cele

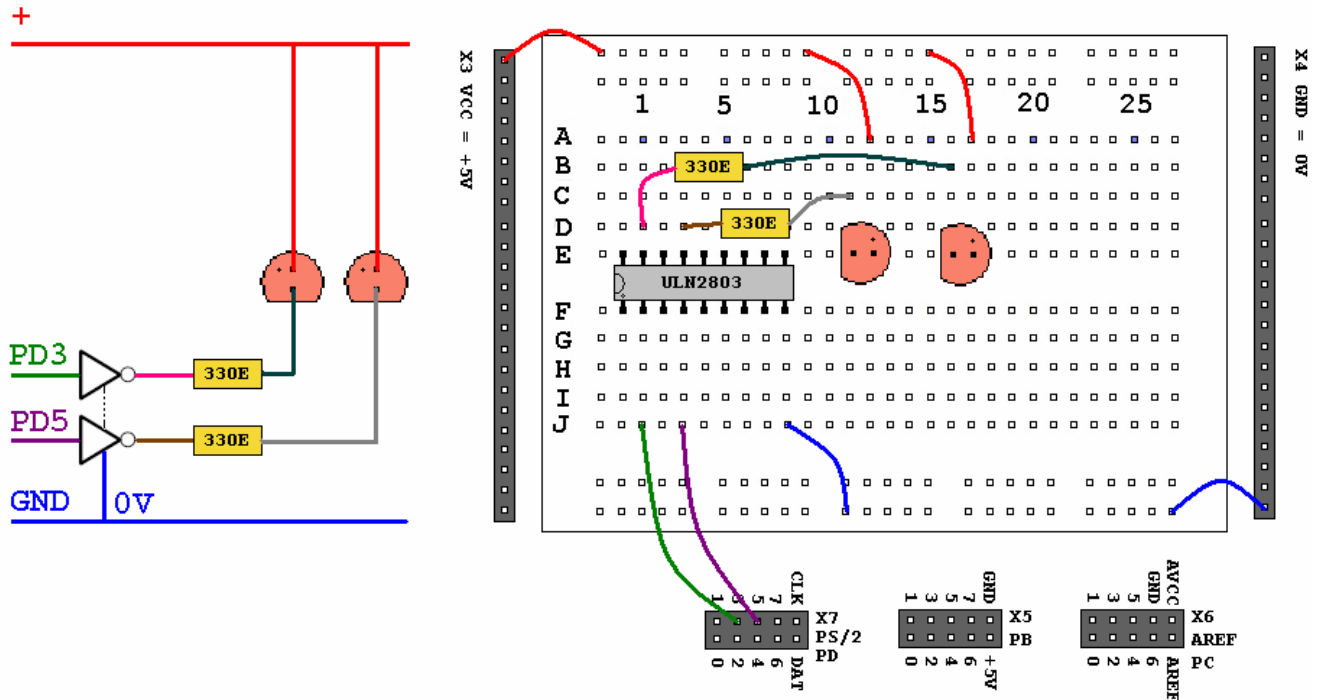
- Nauczysz się jak użyć ULN2803

W wielu aplikacjach jest wykorzystywane więcej niż jedno wyjście. Wyobraź sobie, że do sterowania kilkoma elementami będziesz potrzebował 7 lub 8 tranzystorów! (Patrz rozdział 3.1.3a). Nie tylko będzie to drogie, ale również nieeleganckie.

Aby rozwiązać to zagadnienie, możesz kupić tranzystory w obudowie DIL. Układ ULN2803 posiada 8 tranzystorowych portów, w którym z każdego można pobrać prąd do 30 mA.



Zbuduj układ pokazany na poniższym rysunku:



Otwórz plik EDBexperiment4.bas z płyty EDB-CD i załaduj program do mikrokontrolera ATmega88.

W tym eksperymencie wykorzystane zostały tylko dwa porty układu ULN2803. Oczywiście możliwe jest dołączenie 8-miu diod LED. Mógłbyś także dołączyć przekaźniki, ale musiałbyś sprawdzić ile potrzebują prądu. Dla dodatkowych informacji odnieś się do dokumentacji układu ULN2803.

Zasoby internetowe:

ULN2803: <http://focus.ti.com/docs/prod/folders/print/uln2803a.htm>

3.1.5 Współpraca transoptorów z liniami I/O

Lista elementów

- 1x PC817, SFH601 lub odpowiednik
- 1x Dioda LED
- 7x Przewody
- 1x Rezystory 330E, 820E

Cele

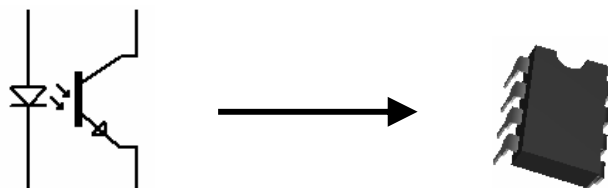
- Nauczysz się jak używać transoptorów
- Poznasz istnienie ESD

Teoria ESD

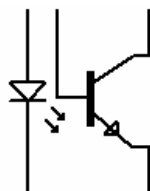
Jeśli kiedykolwiek zobaczyłeś iskrę dotykając czegoś lub kogoś, doświadczyłeś elektrostatycznego wyładowania (ESD). Półprzewodniki są bardzo wrażliwe na szkody powstałe przez ESD. ESD nie tylko jest wywołany przez człowieka, i to nie zawsze jest widoczny i odczuwalny.

Projektowanie elektroniki odpornej na ESD nie jest łatwe i jest poza ramy tej instrukcji. Jednakże pokażemy Tobie jak uzyskać **galwaniczną izolację** z wykorzystaniem transoptorów.

Transoptor jest specjalnym tranzystorem z wbudowaną diodą LED. Tranzystor nie potrzebuje wyprowadzenia BAZY. Jeśli płynie prąd przez diodę LED, jej światło generuje prąd BAZY, gdzie normalnie występuje wyprowadzenie BAZY w zwykłym tranzystorze. Wygląd tego typu elementu przedstawia poniższy rysunek:

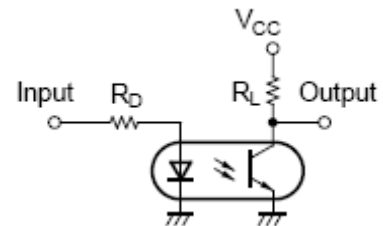
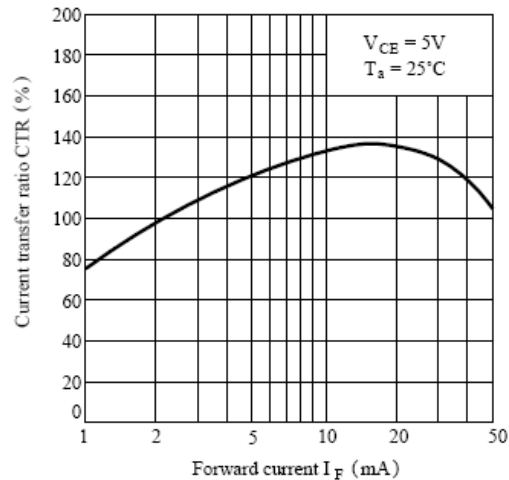


Są też typy transoptorów, które posiadają wyprowadzenie BAZY. W tym przypadku tranzystor można załączać światłem diody LED i/lub prądem podanym na BAZĘ. Wygląd tego typu elementu pokazano na poniższym rysunku:



W tym eksperymencie możesz użyć transoptora PC817, SFH601 lub innego odpowiednika. Jeżeli używasz PC817 spójrz do dokumentacji tego elementu, którą możesz znaleźć na płycie EDB-CD. Zobacz rys. 4, który pokazuje jaki potrzebny jest prąd diody LED, aby można było go zastosować.

Fig. 4 Current Transfer Ratio vs. Forward Current

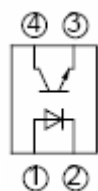
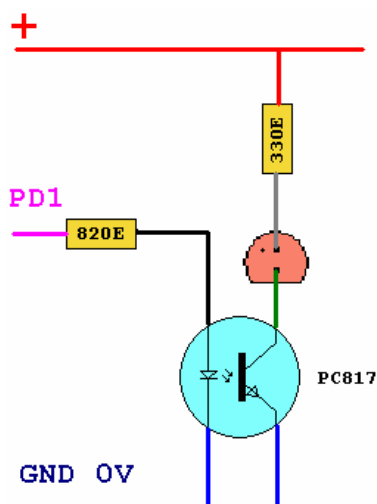


Dla bezpieczeństwa lepiej jest zastosować prąd diody LED około 5 mA. Powiedzmy, że spadek napięcia na diodzie LED jest większy niż 1,3 V (napięcie przewodzenia) i napięcie VCC ma poziom 5 V. Napięcie na rezystorze R_D będzie wynosić $5 - 1,3 = 3,7$ V. Wartość R_D powinna wynosić $R_D = 3,7V / 5mA = 740$ OHM.

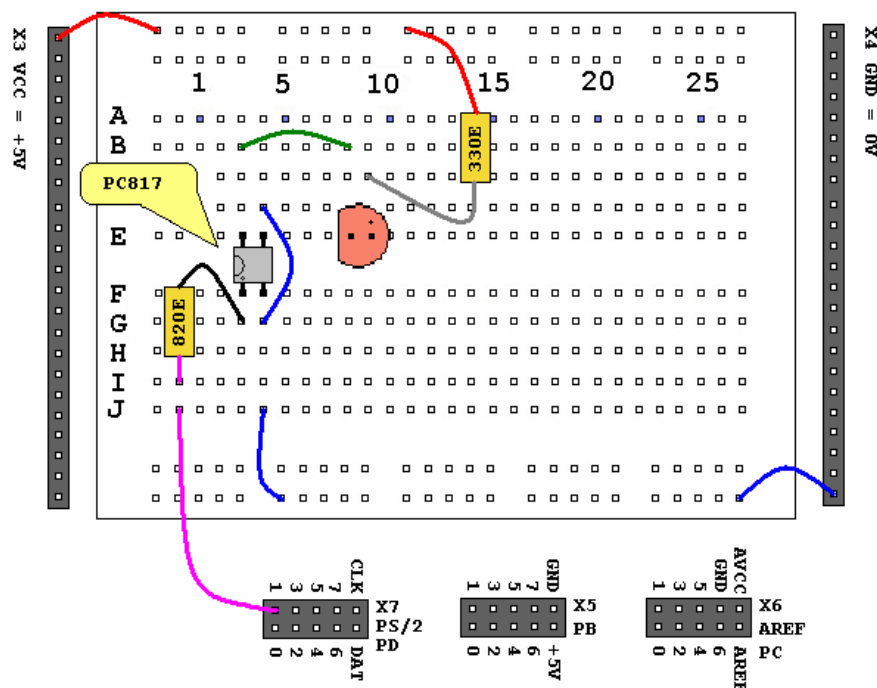
Wartość 740 OHM nie jest wartością, którą można znaleźć wśród rezystorów z szeregu E24. Najbliższą wartością jest 820. Ta wartość da prąd 4,5 mA, który jest w 120% wystarczający.

Zasoby internetowe:

SFH601: <http://www.vishay.com/optocouplers/list/product-83663/>

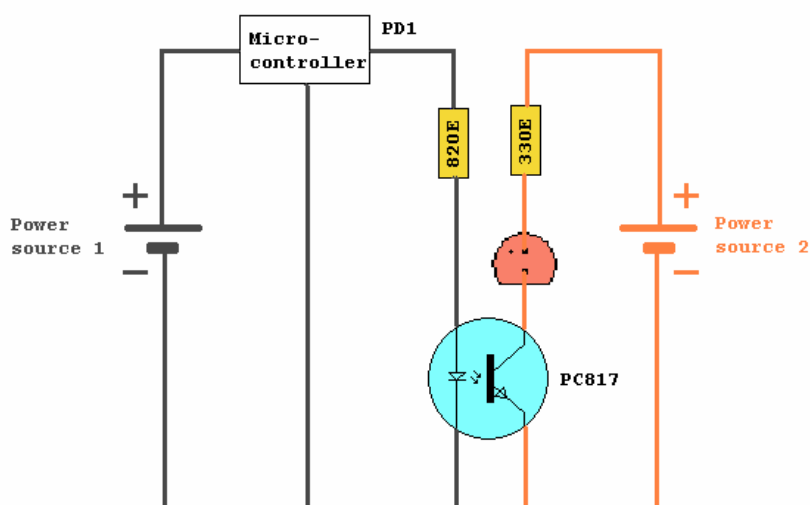


PC817



Otwórz plik EDBexperiment3.bas z płyty EDB-CD i załaduj program do mikrokontrolera ATmega88.

Widzisz że dioda LED błyska. ESD może mieć wpływ przez wszystkie *rodzaje* materiałów. Na rysunku pin PD1 jest zabezpieczony. Ale ESD może przejść także przez linie GND i +, niszcząc tak czy tak mikrokontroler. Aby temu zapobiec możesz zastosować pełną izolację galwaniczną jak pokazano na rysunku:



Ta sama zasada może być wykorzystana do zabezpieczenia wejść.

“Tylko spójrz na to, nie musisz tego układu teraz budować.

3.2 Komunikacja przez UART

Zobaczyłeś wiele rozwiązań prostej obsługi linii I/O. Możemy użytkownikowi pozwolić nacisnąć przycisk potwierdzając to zaświeceniem diody LED. Ale co, gdy chcielibyśmy komunikować się z użytkownikiem po imieniu, pokazując jego imię?

Ten rozdział pokaże Ci najpopularniejszy sposób komunikacji na świecie wśród mikrokontrolerów.

3.2.1 UART

Eksteryment

6a

Lista części

- 1x Kabel RS232
- 1x Komputer z portem COM

Cele

- Nauczysz się używać szeregowej transmisji

Teoria

UART oznacza uniwersalny asynchroniczny odbiornik/nadajnik. Możesz użyć UART, aby transmitować dane pomiędzy PC a płytką EDB.

UART związany jest z portem COM Twojego komputera. Dużą różnicą jest że Twój port COM używa RS232/V24 z sygnałami o poziomie +15 i -15V, kiedy UART używa poziomów TTL (5 V) lub LVTTL (3 V lub mniej). Aby UART mógł komunikować się z PC używamy *konwertera napięć MAXX232 (U3)*. MAX232 jest układem generującym odpowiednie napięcia, aby była możliwa komunikacja z portem COM komputera.

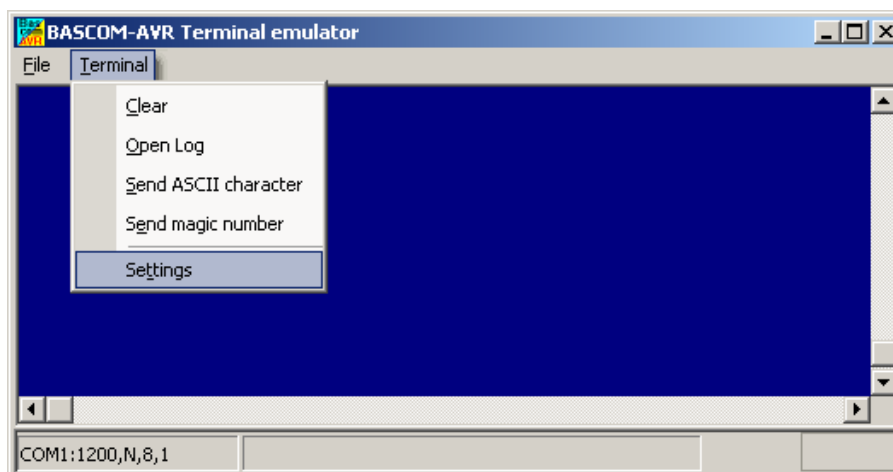
Dla tego eksperymentu potrzebujesz połączyć szeregowym kablem złącze X16 (wtyczka DB-9 nazywana jest RS-232) płytki EDB z portem COM twojego PC.

Aby połączenie było prawidłowe, kabel powinien posiadać następujące połączenia:
pin 2 do pin 2,
pin 3 do pin 3
pin 4 do pin4 i
pin 5 do pin 5.

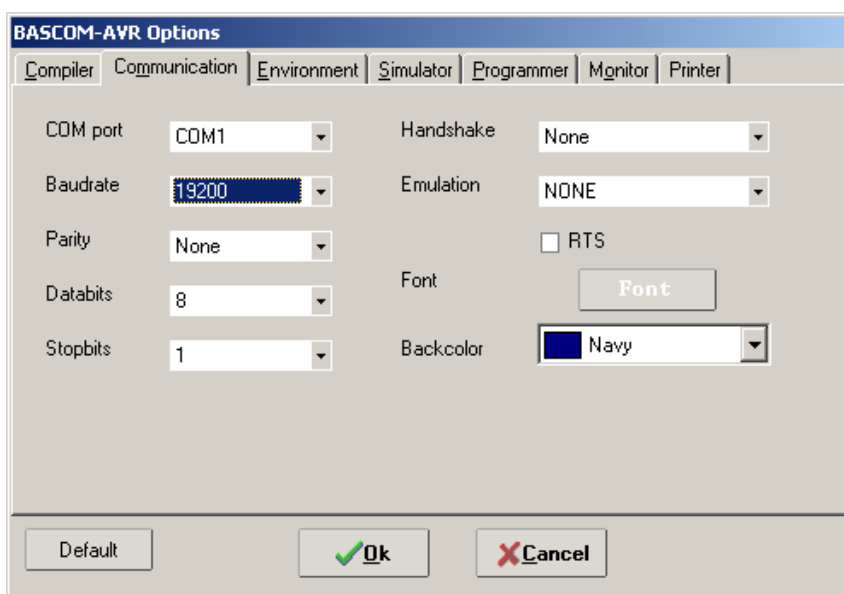
Dopilnuj aby przełącznik "S1 USB/RS232", płytki EDB był ustawiany na RS232.

Gdy połączyłeś płytkę kablem, uruchom Bascom AVR i otwórz Bascomowy emulator terminala klikając na ikonę .

Powinieneś zobaczyć następujące okno:



Kliknij na “Terminal” i “Settings”



Wybierz port COM jaki użyłeś w Twoim PC oraz wybierz prędkość transmisji cyfrowych danych równą 19200.

Jeśli w PC nie posiadasz portu COM, możesz kupić konwerter USB<->COM, który utworzy wirtualny port COM.

Możesz zamknąć okno emulatora terminala.

Jak już wiemy będziemy używać UART do komunikacji płytki EDB z komputerem PC. Sposób transmisji będzie następujący:

Od strony płytki EDB:

Dane wysyłane -> dane z EDB do komputera PC i

Dane odbierane -> dane z komputera do EDB.

Różne dane przesyłane są w identyczny sposób, ale należy rozumieć co to znaczy. Danymi mogą być zmienne, wartości rejestrów, stałe (tekstowe lub liczbowe) lub dane z obliczeń. Mikrokontroler może również wysyłać dane o procedurach wykonywanych w Twoim programie co umożliwia łatwe znalezienie błędów.

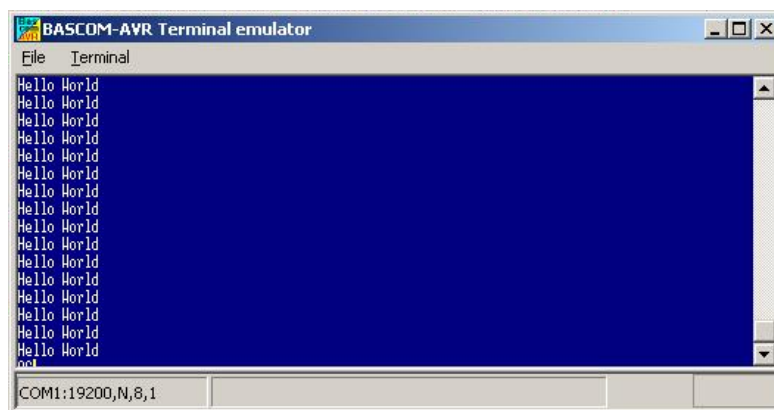
Wysyłane dane mogą również pochodzić z klawiatury, co pozwala użytkownikowi wysyłać wiadomości czy korzystać z menu. Zostanie to wyjaśnione później.

Eksperyment 6a

Otwórz plik EDBexperiment6a.bas z płyty EDB-CD i załaduj program do mikrokontrolera ATmega88. Przeczytaj uwagi jakie można znaleźć w pliku EDBexperiment6a.bas, które pokazują jak zainicjować i używać UART.

Gdy zaprogramowałem mikrokontroler i podłączyłem go do komputera przewodem, otwórz emulator terminala klikając w Bascom na ikonę .

Powinieneś zobaczyć na ekranie PC odbierany z EDB napis "Hello World", jak pokazano na poniższym rysunku:

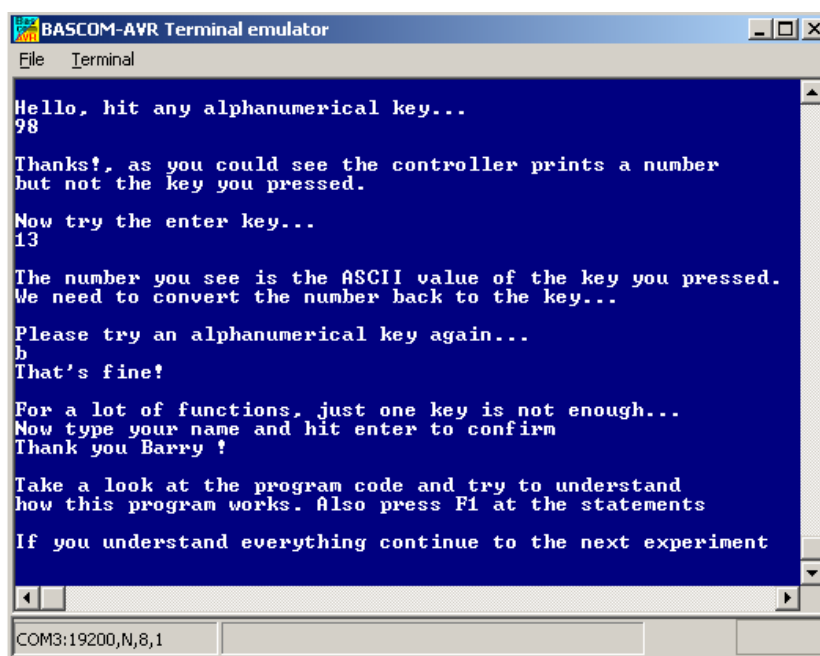


Eksperyment 6b

Dopilnuj aby przełącznik “S1 USB/RS232”, płytki EDB był ustawiany na RS232.

Otwórz plik EDBexperiment6b.bas z płyty EDB-CD i załaduj program do mikrokontrolera ATmega88. Gdy zaprogramowałeś mikrokontroler i dołączyłeś go do komputera przewodem, otwórz emulator klikając w Bascom na ikonę .

Wykonuj instrukcje wyświetlane na ekranie przez program EDBexperiment6b.bas, podobnie jak pokazano poniżej:



```
BASCOM-AVR Terminal emulator
File Terminal

Hello, hit any alphanumerical key...
98

Thanks!, as you could see the controller prints a number
but not the key you pressed.

Now try the enter key...
13

The number you see is the ASCII value of the key you pressed.
We need to convert the number back to the key...

Please try an alphanumerical key again...
b
That's fine!

For a lot of functions, just one key is not enough...
Now type your name and hit enter to confirm
Thank you Barry !

Take a look at the program code and try to understand
how this program works. Also press F1 at the statements
If you understand everything continue to the next experiment

COM3:19200,N,8,1
```

Zauważ że z wykorzystaniem RS232 możesz połączyć ze sobą komputer PC z PC lub mikrokontroler z mikrokontrolerem.

Więcej teorii znajdziesz na następnej stronie.

Zasoby internetowe:

Sprzętowy UART

MAX232: http://www.maxim-ic.com/quick_view2.cfm/qv_pk/1798

Kody ASCII

W programie EDBexperiment6b.bas możesz zauważyć, że do wysyłania danych przez UART używana jest komenda PRINT. Faktycznie nie jest wysyłany tekst. Wysyłane są znaki w kodzie ASCII. ASCII tłumaczony jest jako Amerykański standardowy kod wymiany informacji. Podstawowa grupa kodów ASCII składa się z listy 127 znaków.

Tabela kodów ASCII (Niekompletna)

Decimal	Hex	Binary	Value	
----	---	-----	-----	
000	000	00000000	NUL	(Null char.)
008	008	00001000	BS	(Backspace)
009	009	00001001	HT	(Horizontal Tab)
010	00A	00001010	LF	(Line Feed)
012	00C	00001100	FF	(Form Feed)
013	00D	00001101	CR	(Carriage Return)
048	030	00110000	0	
049	031	00110001	1	
052	034	00110100	4	
065	041	01000001	A	
066	042	01000010	B	
067	043	01000011	C	

Kompletną tablicę znaków ASCII możesz znaleźć w dodatku 2.

KODY POWROTU (CR) I NOWEJ LINII (LF)

Jak widzisz w programie EDBexperiment6b.bas wysyłamy dane przez UART używając polecenia PRINT. Zauważyłeś także że kolejne teksty są wyświetlane w nowych liniach. Jest to spowodowane tym, że do wyświetlanego tekstu zawsze są dodawane znaki CR i LF.

Można stwierdzić że wykonując polecenie:

Print "ABC"

Wysłane zostaną przez UART znaki 65 66 67 13 10. (W formacie binarnym)

Znak powrotu (13) spowoduje, że kursor zostanie umieszczony na pozycji 0 bieżącej linii. Znak nowej linii (10) spowoduje umieszczenie kursora w kolejnej linii.

Print "ABC" ;

Możemy na końcu linii umieścić średnik (;). Bascom nie wyśle znaków CR i LF, więc w tej samej linii można wyświetlać kolejny tekst (po znakach ABC).

Print "ABC" ; Chr(13) ;

Ta instrukcja wysła tylko znaki ABC CR. Następne znaki zastąpią znaki ABC.

3.2.2 Programowy UART

Lista elementów

- 1x Kabel RS232
- 1x Komputer z portem COM

Cele

- Nauczysz się korzystać z programowego UART

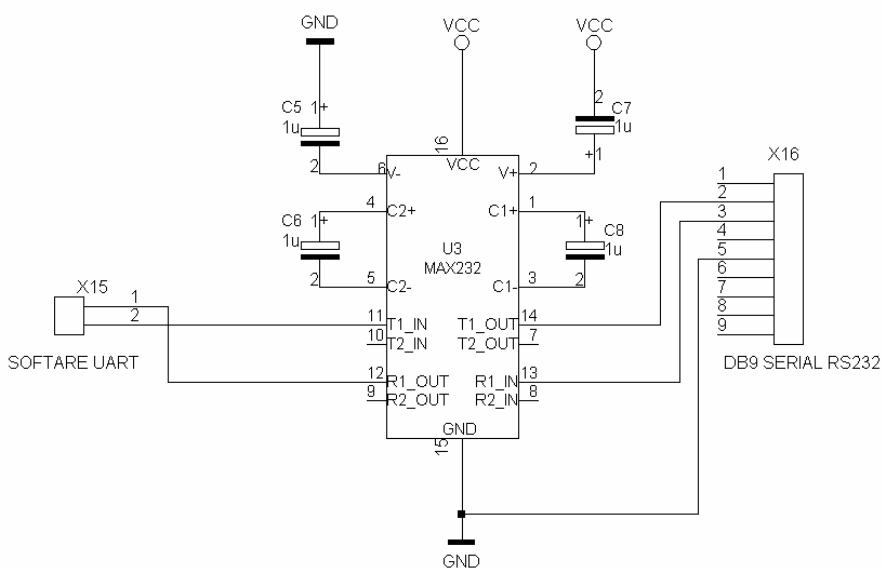
Poprzednie przykłady wykorzystywały sprzętowy UART. Tym sposobem komunikacji kompilator używa wewnętrznych rejestrów UART oraz wewnętrznego sprzętu (portd.0 i portd.1) w mikrokontrolerze ATmega88. Czasami potrzebny jest więcej niż jeden UART lub chcemy jednocześnie używać USB i RS232 (komunikacja przez USB jest wyjaśniona w rozdziale 3.8.2 i kolejnych).

Kompilator w Bascom umożliwia w prosty sposób tworzenie dodatkowych układów UART. Bascom tworzy programowe układy UART i może wykorzystywać prawie każde wyprowadzenie portu.

Dopilnuj aby przełącznik “S1 USB/RS232”, płytki EDB był ustawiany na USB.

Nie użyjemy teraz interfejsu USB, ale przełącznik S1 należy ustawić na USB, aby piny portd.0 i portd.1 były odłączone od układu konwertera poziomów MAX232.

Rysunek poniżej przedstawia sytuację, gdy S1 jest ustawiony na USB:



Co teraz musisz zrobić, to podłączyć używane linie programowego UARTu do złącza X15.

Dołącz pin złącza X15, który jest najbliżej S1 do linii portc.1.
Dołącz pozostały pin złącza X15 do linii portc.2.

Zauważ że jeśli używasz bootloadera, aby zaprogramować mikrokontroler w płytce EDB, kiedy programujesz musisz przełącznik (USB/RS232) ustawić na RS232 i aby przetestować programowy UART będziesz musiał po zaprogramowaniu ustawić go na USB (w celu ponownej inicjalizacji naciśnij przycisk zerujący, gdy przełączysz S1 na USB. W przeciwnym przypadku w oknie terminala nic nie zobaczysz).

Otwórz plik EDBexperiment6c.bas z płyty EDB-CD i załaduj program do mikrokontrolera ATmega88. Przeczytaj uwagi jakie można znaleźć w pliku EDBexperiment6c.bas, które pokazują jak zainicjować i używać programowy UART.

Gdy zaprogramowałeś mikrokontroler i dołączyłeś go do komputera przewodem, otwórz emulator klikając w Bascom na ikonę .

Powinieneś widzieć działanie programu EDBexperiment6c.bas, który zachęca Cię do wprowadzanie tekstu, który następnie wyświetla w oknie terminala.

3.3 Obsługa wyświetlaczy

Nie zawsze możemy użyć PC aby coś wyświetlić. Zwłaszcza, gdy do wyświetlenia jest tylko kilka linii. Dlatego do tego celu lepszym rozwiązaniem będzie użycie alfanumerycznego wyświetlacza LCD.

3.3.1 Wyświetlacz LCD (HD44780)

Lista elementów

1x Wyświetlacz LCD z HD44780

1x Komputer z portem COM

Cele

- Nauczysz się jak używać wyświetlacza LCD

Otwórz plik EDBexperiment7.bas z płyty EDB-CD i załaduj program do mikrokontrolera ATMega88. Przeczytaj uwagi jakie można znaleźć w pliku EDBexperiment7.bas, które pokazują jak zainicjować i używać wyświetlacz LCD.

Musisz umieścić wyświetlacz LCD w złączu X12 płytki EDB.

Gdy zamontowałeś LCD i zaprogramowałeś mikrokontroler, na wyświetlaczu powinieneś zobaczyć tekst "Hello World".

Spróbuj także skorzystać ze sprzętowego symulatora dla tego programu. Jeśli masz otwarty plik EDBexperiment7.bas kliknij w Bascom na ikonę .

Wtedy kliknij w oknie symulatora na ikonę . Następnie naciśnij kolejno przycisk "step-into-code" (kilkakrotnie). Zobaczysz tekst jak na rysunku:



Zasoby internetowe:

Producentem Hitachi Semi jest teraz Renesas

Producent LCD: <http://www.renesas.com/>

Obsługa LCD: <http://home.iae.nl/users/pouwheha/lcd/lcd.shtml>

Ćwiczenie

1

Ćwiczenie

Najlepszą drogą do nauki są praktyczne ćwiczenia. Spróbuj napisać program pytający o “Twoje imię” przez UART, ale tak aby wpisane imię nie było wyświetlane na ekranie komputera, ale na wyświetlaczu LCD (Rozwiązanie możesz znaleźć na płycie EDB-CD w pliku Exercise1.bas)

3.3.2 Wyświetlacz 7-segmentowy

Lista elementów

- 1x Wyświetlacz 7-Segmentowy
(Wspólna Anoda SL119-OS516HWA)
lub inny odpowiednik
8x Przewody

Cele

- Nauczysz się jak używać
wyświetlacza 7-segmentowego

Są dwie wady wyświetlaczy LCD, po pierwsze mogą być drogie, a po drugie wyświetlane znaki nie są dobrze widoczne jak w starym dobrym wyświetlaczu LED. Jeśli potrzebujesz pokazać jeden znak możesz użyć wyświetlacza LED 7-segmentowego.

Wyświetlacz 7-segmentowy składa się z kilku odpowiednio umieszczonych diod LED umożliwiających wyświetlenie znaków poprzez załączanie odpowiednich diod. Są ogólnie dwa typy wyświetlaczy LED. Ze wspólną anodą oraz ze wspólną katodą.

Wyświetlacze ze wspólną anodą posiadają wspólny DODATNI biegun, który należy dołączyć do dodatniego bieguna zasilania (VCC). Aby zapalić dane segmenty LED wyświetlacza, mikrokontroler powinien odpowiedni jego pin dołączyć do 0V/GND.

Wyświetlacze ze wspólną katodą posiadają wspólny UJEMNY biegun, który należy dołączyć do ujemnego bieguna zasilania (0V/GND). Aby zapalić dane segmenty LED wyświetlacza, mikrokontroler powinien odpowiedni jego pin dołączyć do +5V/VCC.

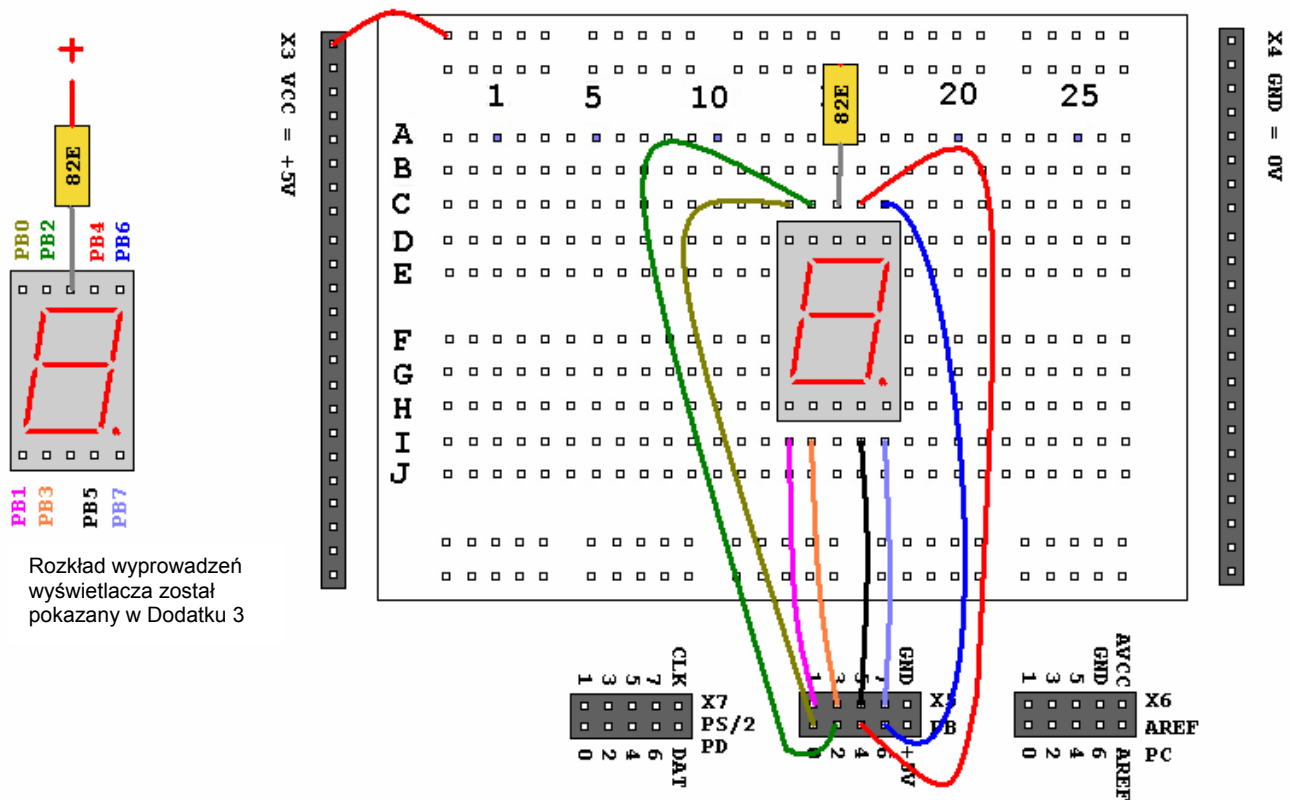
Najczęściej wykorzystywane są wyświetlacze ze wspólną anodą ponieważ mikrokontrolery mogą oddawać więcej prądu, gdy ich linie są ustawione do masy niż do napięcia VCC.

Zasoby internetowe:

Aplikacja multipleksowego wyświetlacza 7-segmentowego:

http://www.maxim-ic.com/appnotes.cfm/appnote_number/3210

Zbuduj układ jak na poniższym rysunku:



Otwórz plik EDBexperiment8.bas z płyty EDB-CD i załaduj program do mikrokontrolera ATmega88. Po zaprogramowaniu mikrokontrolera powinieneś zobaczyć na wyświetlaczu znaki szesnastkowego licznika (znaki 0....9 A....F)

Przeczytaj uwagi jakie można znaleźć w pliku EDBexperiment8.bas. Sprawdź i zrozum działanie komend Data oraz Read.

3.4 Klawiatury

Dobrze by było, gdyby do mikrokontrolera można było dołączyć PS2 lub klawiaturę AT zamiast dołączać do niego komputer (i odczytywać klawisze przez UART). W tym rozdziale poznasz jak łatwo jest dołączyć klawiaturę do EDB z wykorzystaniem Bascom.

3.4.1 Klawiatura PS2 lub AT

Eksperyment

9

Lista elementów

- 1x Wyświetlacz LCD HD44780
- 1x Klawiatura AT lub PS2
- 1x Opcjonalnie komputer z portem COM

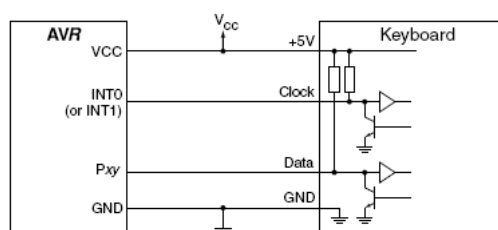
Cele

- Nauczysz się jak dołączyć klawiaturę AT lub PS2

Płytką EDB została już wyposażona w standardowy interfejs PS2. Możesz do niego dołączyć standardową klawiaturę PS2 (złącze X10 oznaczone PS2 na płytce EDB). Możesz także dołączyć starszą klawiaturę AT, ale będziesz potrzebował odpowiedniej przejściówki ze złącza DIN42524 na złącze DIN PS2.

Na poniższym rysunku pokazano interfejs oraz rozkład i opis wyprowadzeń w złączu DIN, które można znaleźć w twojej klawiaturze.

Interfejs klawiatury



Rozkład i opis wyprowadzeń

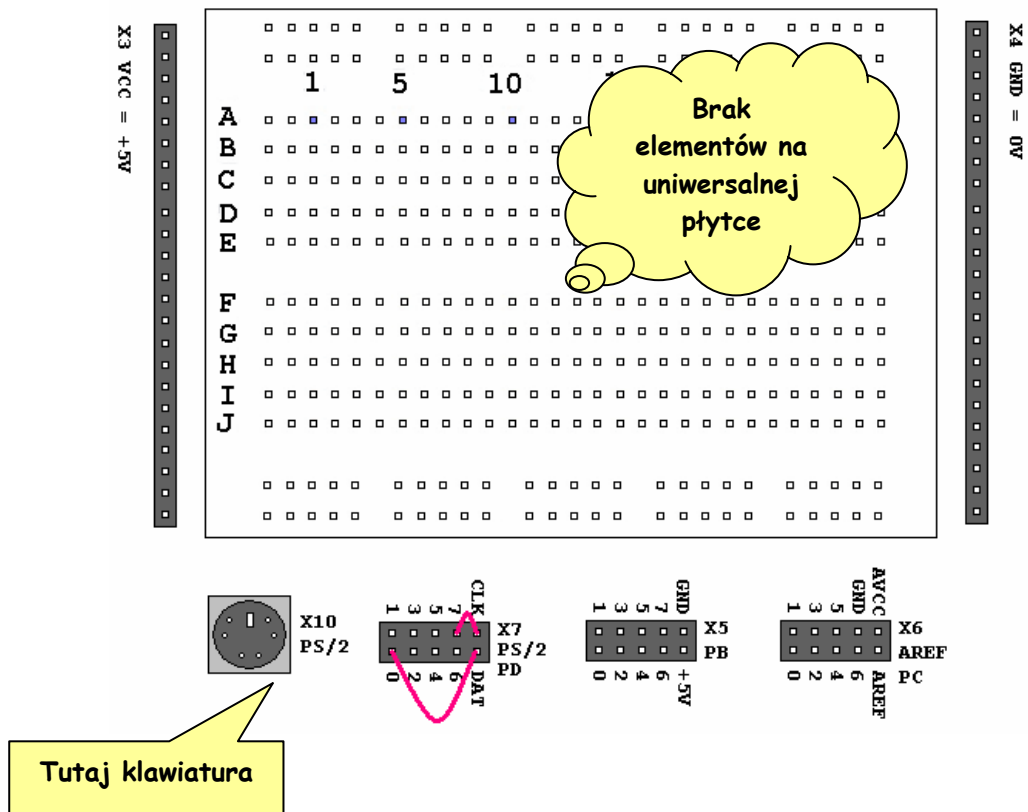
AT Computer	DIN41524, Female at Computer, 5-pin DIN 180°	6-pin Mini DIN PS2 Style Female at Computer
Signals		
Clock	1	5
Data	2	1
nc	3	2,6
GND	4	3
+5V	5	4
Shield	Shell	Shell

Źródło: Atmel Nota aplikacyjna "AVR313: Interfejs klawiatury PC AT"

Zasoby internetowe:

Noty aplikacyjne firmy Atmel:

http://www.atmel.com/dyn/products/app_notes.asp?family_id=607



Podłącz klawiaturę i zobacz powyższy rysunek. Dołącz pin CLK do PD.7 oraz pin DAT do PD.0. Otwórz plik EDBexperiment9.bas i zaprogramuj mikrokontroler. .

Teraz napisz jakiś tekst wykorzystując klawiaturę i naciśnij klawisz Enter. Twój tekst zostanie wyświetlony na wyświetlaczu LCD.

3.4.2 Klawiatura matrycowa

Lista elementów

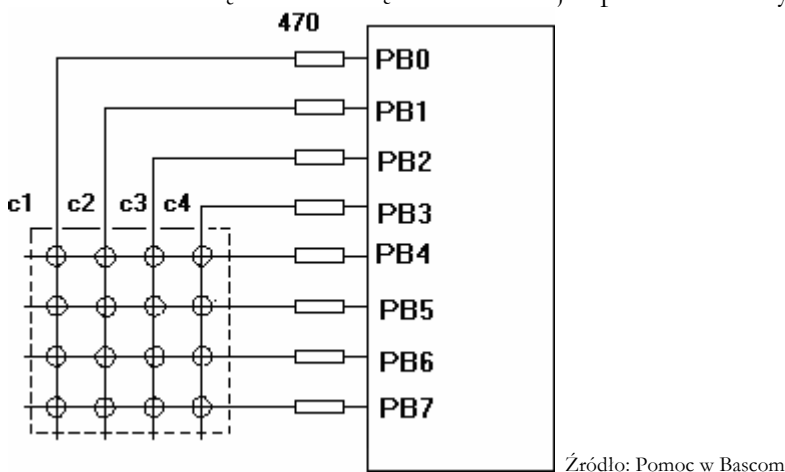
- 1x Klawiatura matrycowa
dostępna w Conrad 709840
- 1x Komputer z portem COM
- 8x Rezystor 470E

Cele

- Nauczysz się jak dołączyć klawiaturę matrycową

Teoria

Mikrokontroler AVR ATmega88 jest (kiedy była pisana instrukcja) jednym z najnowszych mikrokontrolerów z rodziny AVR. Nowe mikrokontrolery AVR dodatkowo posiadają więcej źródeł przerwań, niż ich poprzednicy. Obsługę przerwań zobaczysz w eksperymencie z klawiaturą matrycową z 16 lub 9 klawiszami. Podłącz klawiaturę do AVR tak jak pokazano na rysunku:



Jak widzisz do obsługi 16 przycisków wykorzystywanych jest tylko 8 linii I/O. Są dwa sposoby sprawdzania naciśnięcia przycisków. Pierwszy sposób to ciągłe sprawdzanie klawiatury. To znaczy, że oprogramowanie cały czas sprawdza czy nie został naciśnięty jakiś przycisk (w pętli do...loop). W tym sposobie jest problem, gdyż mikrokontroler nie może wykonywać innych zadań, prócz sprawdzania matrycowej klawiatury.

Drugi sposób polega na wykorzystaniu przerwań. Ja napisano prawie każda linia portu mikrokontrolera ATmega88 może zostać wybrana jako źródło przerwania. Kiedy używane jest przerwanie, procedura „czytania klawiatury”, wykonywana jest tylko kiedy użytkownik nacisnął przycisk. (Procedura „czytania klawiatury” wewnątrz procedur przerwań jest taka sama jak w przypadku ciągłego sprawdzania klawiatury, tylko że przy wykorzystaniu przerwań procedura „odczytu klawiatury” zostanie wykonana tylko kiedy naciśniemy przycisk)

Krótkie wytłumaczenie działania przerwań znajduje się na następnej stronie.

PRZERWANIA

Przerwanie jest zdarzeniem, w którym wykonywanie programu głównego jest wstrzymywane i następuje skok do podprogramów obsługi przerwań. Przerwania mogą być wywołane przez linie I/O, ale również przez linie RxD układu UART czy czasomierz/licznik.

Po wykonaniu przerwania, następuje powrót do wykonywania programu głównego do miejsca w którym zostało wywołane przerwanie.

Możesz użyć przerwań jeśli chcesz, aby niektóre wykonywane zadania posiadały priorytet.

Nie będziemy w tym eksperymencie sugerować się uniwersalną płytką, ponieważ klawiatura matrycowa dokładnie nie pasuje do uniwersalnej płytki. Będziesz musiał wykonać połączenia jak pokazano na poprzedniej stronie.

Połącz pin1 matrycy przez rezystor 470 Ohm do portb.0,
Połącz pin2 matrycy przez rezystor 470 Ohm do portb.1, itd...

Teraz zaprogramuj mikrokontroler ATmega88 programem EDBexperiment10.bas i dołącz płytkę EDB szeregowym przewodem do Twojego komputera.

W oknie programu terminal powinieneś przeczytać wartość 16, i jeśli przyciśniesz któryś z przycisków powinieneś zobaczyć jak zmienia się wartość która będzie < 16.

Przeczytaj uwagi z pliku EDBexperiment10.bas file i zobacz czy rozumiesz komendę Getkbd, którą inicjujemy z użyciem przerwań.

Ten eksperyment może być wykonany tylko, gdy używasz Bascom w wersji 1.11.8.1 lub nowszej.

Zobacz również plik EDBexperiment10polling.bas w którym matryca klawiszy czytana jest bez wykorzystywania przerwań. Zobacz czy rozumiesz różnice pomiędzy dwoma sposobami odczytu klawiatury.

Jeśli używasz starszej wersji Bascom, możesz pobrać nowszą wersję demonstracyjną ze strony www.mcselec.com lub wykorzystać obsługę klawiatury w programie głównym zamieszczoną w pliku EDBexperiment10polling.bas.

3.4.3 Zdalny odbiór sygnałów RC5

Lista elementów

- 1x TSOP1736 lub odpowiednik
- 1x Komputer z portem COM
- 1x Pilot podczerwieni z RC5

Cele

- Nauczysz się jak używać zdalnego sterowania RC5

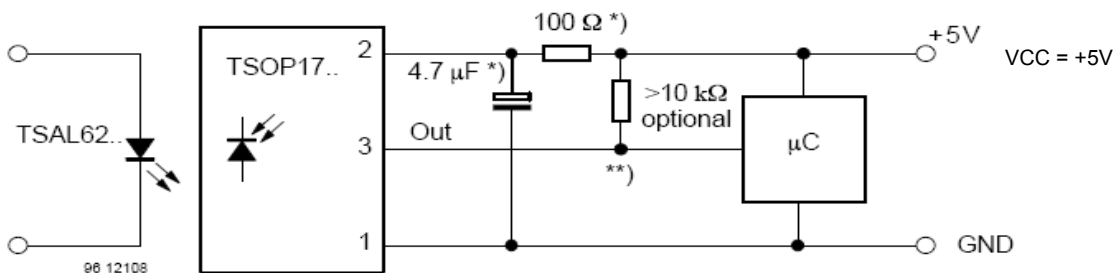
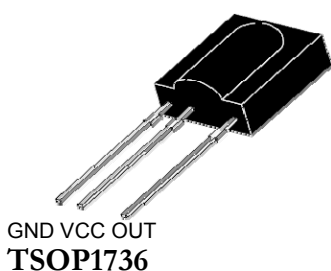
Można dyskutować czy rozdział o “zdalnym sterowaniu” powinien zostać umieszczony w rozdziale „klawiatury”. Tak czy owak możemy użyć zdalnego sterowania jako układu wejściowego.

Obecnie w każdym domu jest wiele urządzeń zdalnie sterowanych podczerwienią. Wiele z tych urządzeń używa protokołu RC5. W Bascom AVR już zaimplementowano obsługę protokołu RC5. Dlatego też możliwość zdalnego sterowania może być łatwo dodana do płytki EDB.

Jeśli chcesz się więcej dowiedzieć o protokole RC5, poczytaj aplikacje dostarczane przez firmę Philips:

http://www.semiconductors.philips.com/acrobat_download/applicationnotes/AN10210_2.pdf

Moduł TSOP1736 jest odbiornikiem podczerwieni. Poniższy rysunek pokazuje interfejs układu TSOP1736.



*) recommended to suppress power supply disturbances

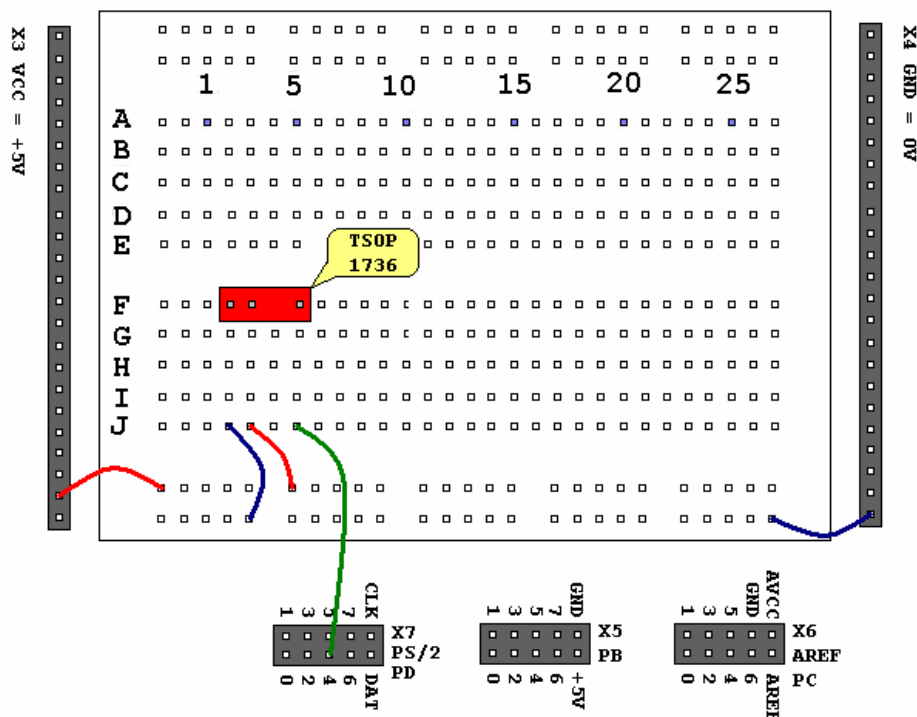
**) The output voltage should not be hold continuously at a voltage below 3.3V by the external circuit.

Źródło: Vishay Semiconductors

(Dioda TSAL jest wewnątrz pilota zdalnego sterowania)

Możesz użyć różnych odbiorników podczerwieni (takie jak SFH506 lub SFH 5110-36). Jeśli używasz innego odbiornika, sprawdź rozkład jego wyprowadzeń przed podłączeniem do płytki EDB.

Sugerowane połączenia na uniwersalnej płytce dla tego eksperymentu:



Co teraz musisz zrobić to znaleźć pilot zdalnego sterowania z kodowaniem RC5. Urządzenia Philipsa w większości używają protokołu RC5, ale uniwersalne piloty również mogą używać protokołu RC5.

Gdy znalazłeś pilot z RC5 lub masz wątpliwości z jakim protokołem działa, otwórz program z pliku EDBexperiment11.bas. Połącz płytkę EDB z komputerem za pomocą szeregowego kabla. Nakieruj pilota na odbiornik podczerwieni umieszczony na płytce EDB. Jeśli naciskasz przyciski pilocie, powinieneś zobaczyć na ekranie komputera (w oknie terminala) odpowiadające im numeryczne wartości.

Ten eksperyment może być wykonany tylko, gdy używasz Bascom w wersji 1.11.8.1 lub nowszej.

Przeczytaj uwagi w pliku EDBexperiment11.bas i zobacz czy wszystko rozumiesz.

Zasoby internetowe:

Informacje o RC5: <http://www.clearwater.com.au/rc5/>
<http://www.epanorama.net/links/irremote.html>

3.5 Zaawansowana obsługa linii I/O

W tym rozdziale będziemy mówić o zaawansowanej obsłudze linii I/O i kwestiach, które nie było łatwo przedstawić w rozdziałach, w których nie był używany UART. W rozdziale 3.5.1 zobaczymy, że przyciski nie są idealne, dalej dowiemy się jak odczytywać pozycję potencjometru oraz jak z wykorzystaniem fotorezystora (LDR) wykrywać ciemność. Rozdział kończy się generatorem tonów z wykorzystaniem głośnika.

3.5.1 Licznik z wejściem eliminującym drgania

Eksperyment 12

Lista elementów

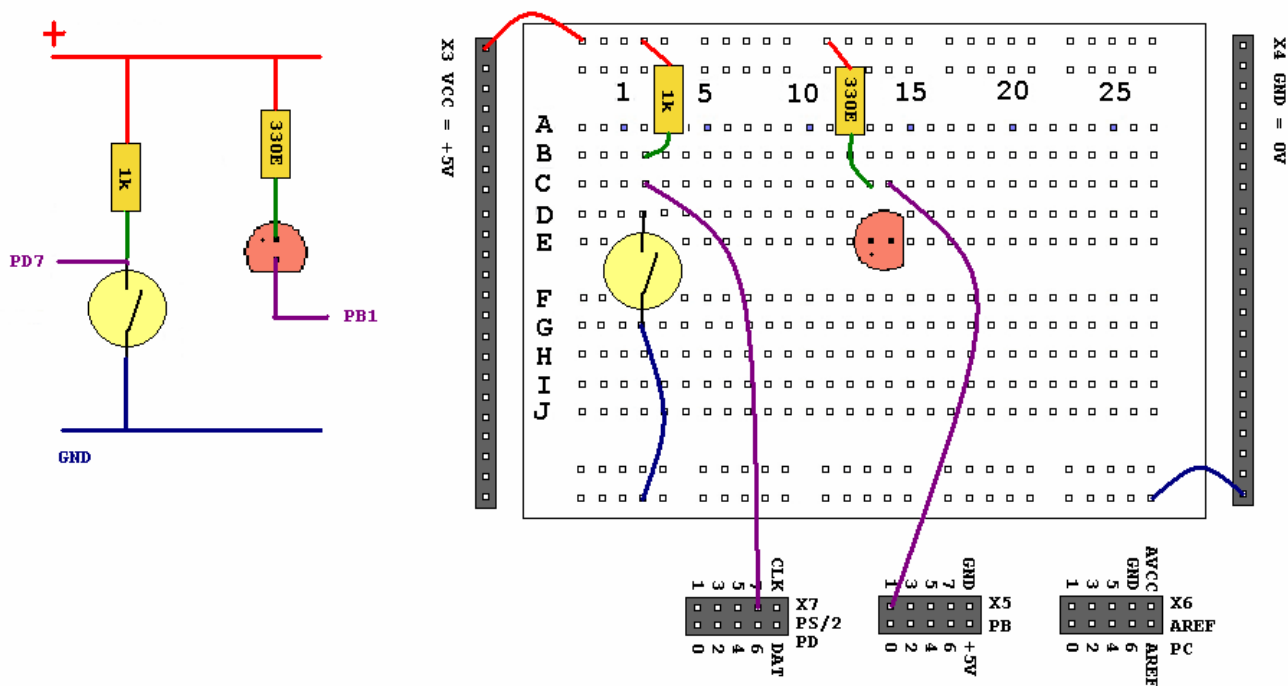
- 1x Dioda LED
- 1x Przycisk lub przewód
- 6x Przewody
- 1x Rezystory 330E, 1k

Cele

- Poznasz nieidealne przyciski
- Programowa eliminacja drgań styków

Przyciski nie są idealne. To znaczy że jeśli dołączysz przycisk do mikrokontrolera jak w eksperymencie 1, twój program może nie działać prawidłowo. To nie jest duży problem, gdy dołączysz diodę LED, ale problem może być, gdy chcesz wykorzystać i sterować licznikiem.

Ponownie zbuduj układ z eksperymentu 1:

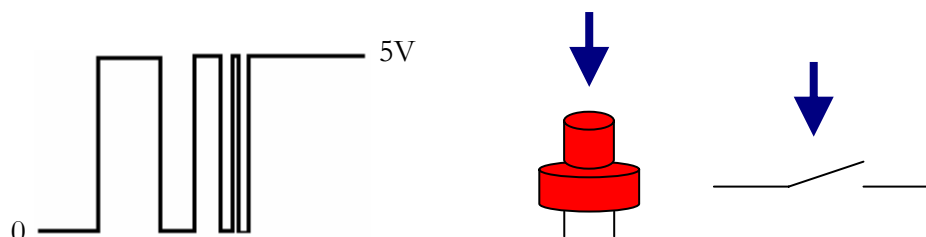


Zaprogramuj mikrokontroler ATmega88 programem EDBexperiment12a.bas oraz dołącz płytke EDB do komputera szeregowym kablem.

Program EDBexperiment12a.bas “zlicza” ilość naciśnieć przycisku. Zliczoną wartość zobaczysz w oknie terminala. Jeśli będziesz miał szczęście, będziesz widział że program zlicza prawidłowo naciśnięcia przycisku.

Teraz zastąp przycisk przewodem, którym zwieraj i rozwieraj miejsce przycisku obserwując zachowanie licznika. Jak widzisz licznik działa nieprawidłowo. Licznik zlicza prawidłowo, ale czasami zlicza kilka impulsów przy pojedynczym zwarcie przewodu.

Pojawia się to zawsze, gdy kontakt styków nie jest stabilny. Inaczej mówiąc *styki drgają*:



To jest zjawisko *drgania styków*. Stabilny kontakt styków uzyskuje się po kilkunastu milisekundach, ale mikrokontroler pracuje z częstotliwością 8 MHz i dostrzega opadające i rosnące zbocza przebiegów drgań trwających kilkadziesiąt milisekund. Jeśli naciśnięty zostanie przycisk w powyższej sytuacji, mikrokontroler zliczy 4 impulsy zamiast jednego.

Rozwiązaniem jest wprowadzenie *czasu martwego*, w którym mikrokontroler nie zauważa zmian w poziomie sygnału. Czas martwy można łatwo wprowadzić przez dodanie opóźnienia „Waitms 100” za linią w programie, gdzie sprawdzane jest wejście z dołączonym przyciskiem.

Możesz zaprogramować mikrokontroler programem EDBexperiment12b.bas, przeczytaj uwagi i zwróć uwagę na linię “Waitms 100”. Jeśli Twój mikrokontroler nadal nieprawidłowo liczy, możesz zwiększyć wartość w Waitms do 250.

Dodanie komendy “Waitms” do dostarczenia poprawnych sygnałów dla licznika jest nazywane “programowa eliminacją drgań styków”.

3.5.2 Komenda GetRC

Lista elementów

- 1x Kondensator ceramiczny 100nF - Poznasz działanie komendy GetRC
- 1x Potencjometr 10k
- 7x Przewody

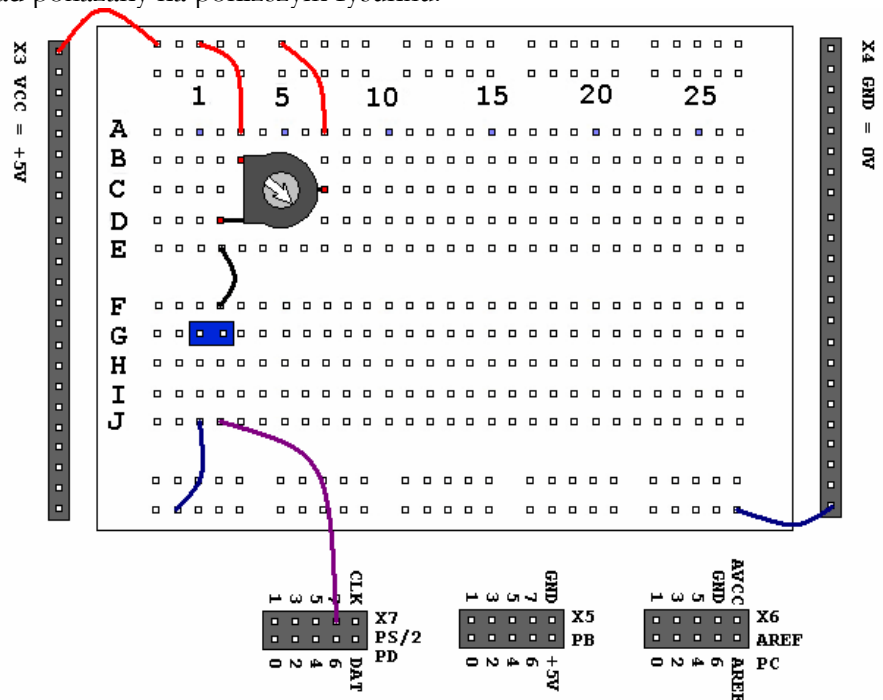
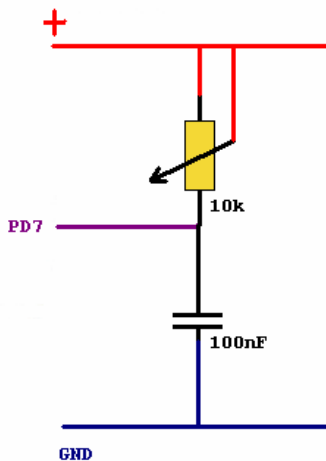
Cele

Komenda GetRC w Bascom daje możliwość odczytu pozycji potencjometru. Może być użyta, gdy nie jest potrzebna duża dokładność.

Zbuduj układ pokazany na poniższym rysunku:



Kondensator
100 nF
"104"



Tekst na kondensatorze pokazuje wartość w piko Faradach. Więc 100 nF = 100000 pF. Ostatnia cyfra na kondensatorze jest liczbą '0' (Więc 104 to kondensator = 100000 pF = 100 nF)

Potencjometr może nie pasować do uniwersalnej płytki. Jeśli nie pasuje, należy dolutować do niego przewody o średnicy 0,5mm, którymi można dołączyć go do płytki..

Zaprogramuj mikrokontroler ATmega88 programem EDBexperiment13.bas i połącz płytkę EDB z Twoim komputerem szeregowym przewodem. W oknie terminala zobaczysz wartości, które możesz zmieniać przez obracanie osi potencjometru. Przeczytaj uwagi z pliku EDBexperiment13.bas i sprawdź czy wszystko rozumiesz.

3.6 Przetwornik A/C i C/A

Choc mikrokontrolery są cyfrowe, mogą przetwarzać sygnał analogowy na cyfrowy oraz cyfrowy na analogowy. Daje to możliwość mierzenia sygnałów analogowych. Ale również jest możliwe uzyskanie “analogowych” wyjść. W tym rozdziale nauczysz się jak mierzyć i uzyskiwać analogowe sygnały.

Eksperyment 15a

3.6.1 Generator PWM

Lista elementów

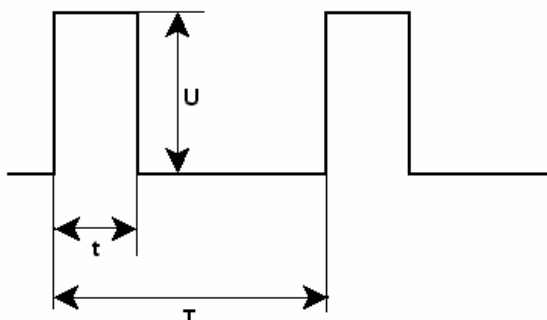
- 1x MOSFET IRF520 lub odpowiednik
- 1x Żarówka 6 V
- 6x Przewody

Cele

- Poznasz działanie PWM

PWM jest formą cyfrowego sygnału, który można użyć jako sygnału analogowego.

Sygnały PWM są sygnałami cyfrowymi DC, jak pokazano na rysunku:



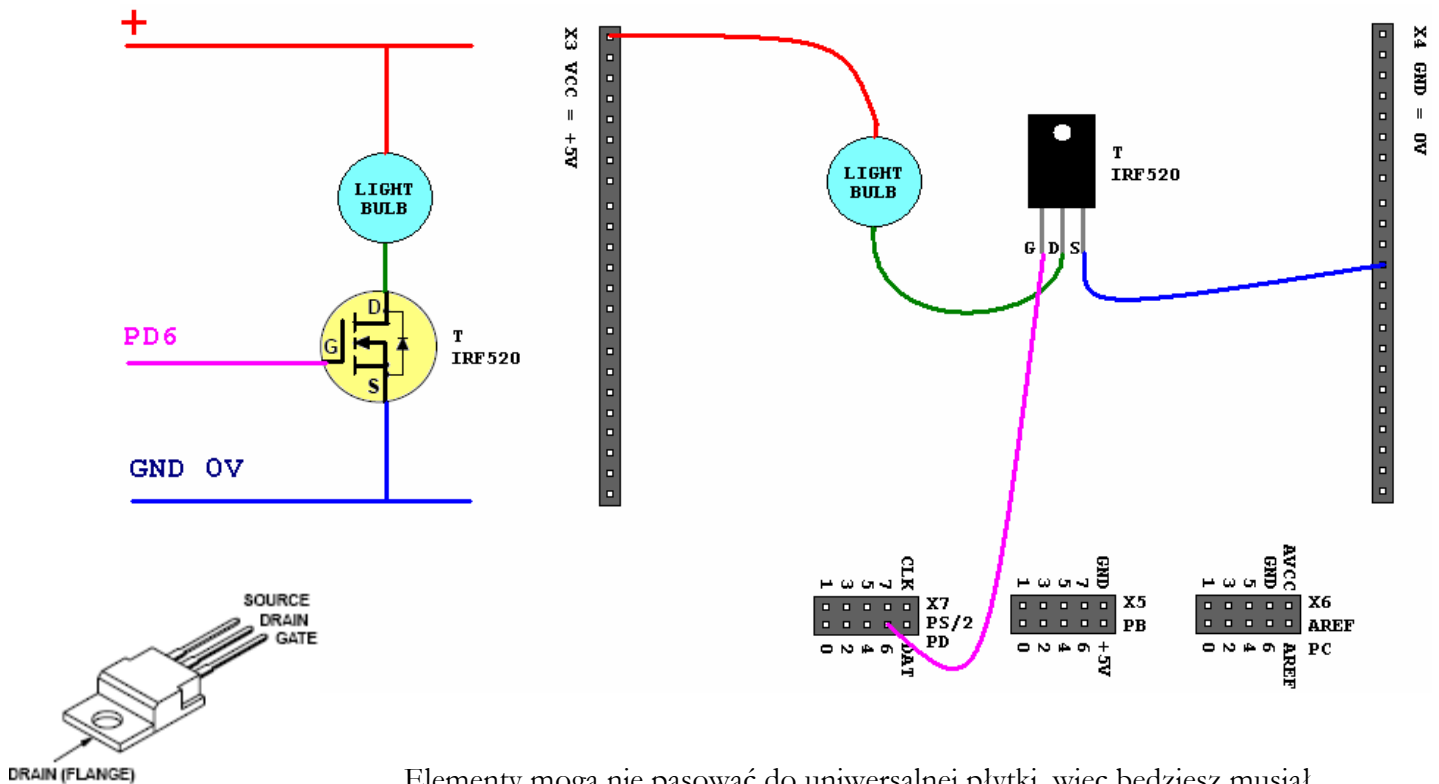
Wyobraź sobie, że zasilimy takim sygnałem żarówkę. Jeśli będziemy zwiększać czas t , średnie napięcie na żarówce także będzie się zwiększać. Czas T należy wybrać taki, aby nie było widoczne miganie żarówki. Wtedy wygląda to tak jakby żarówka była sterowana analogowo.

Zamiast żarówki, możesz dołączyć do sygnału PWM kondensator, co jest wystarczające aby uzyskać analogowy sygnał.

Opisany wyżej sygnał, nazywany jest sygnałem o zmiennym wypełnieniu (PWM). PWM jest implementowany w mikrokontrolerach AVR i obsługiwany przez Bascom.

W tym eksperymencie pokazany zostanie przykład wykorzystania PWM.

Zbuduj pokazany poniżej układ:



Elementy mogą nie pasować do uniwersalnej płytki, więc będziesz musiał dolutować do nich przewody.

Zaprogramuj mikrokontroler ATmega88 programem EDBexperiment15a.bas i połącz płytkę EDB z komputerem przez szeregowy przewód.

Jasność świecenia żarówki powinna się zmieniać, natomiast wartość sygnału PWM powinna być widoczna w oknie terminala.

Przeczytaj uwagi z pliku EDBexperiment15a.bas i zobacz czy wszystko rozumiesz. Program EDBexperiment15a.bas używa czasomierza o których więcej możesz się dowiedzieć z rozdziału 3.7.3.

Zasoby internetowe:

Opis PWM: <http://www.netrino.com/Publications/Glossary/PWM.html>
<http://www.4qdttec.com/pwm-01.html>

IRF520: <http://ec.irf.com/v6/en/US/adirect/ir?cmd=catProductDetailFrame&productID=IRF520>

3.6.2 Przetwornik A/C z fotorezystorem (LDR)

Lista elementów

- 1x Fotorezystor NSL19-MS51
RS596-141 lub odpowiednik
- 1x Komputer z portem COM
- 2x Rezystor 2k2

Cele

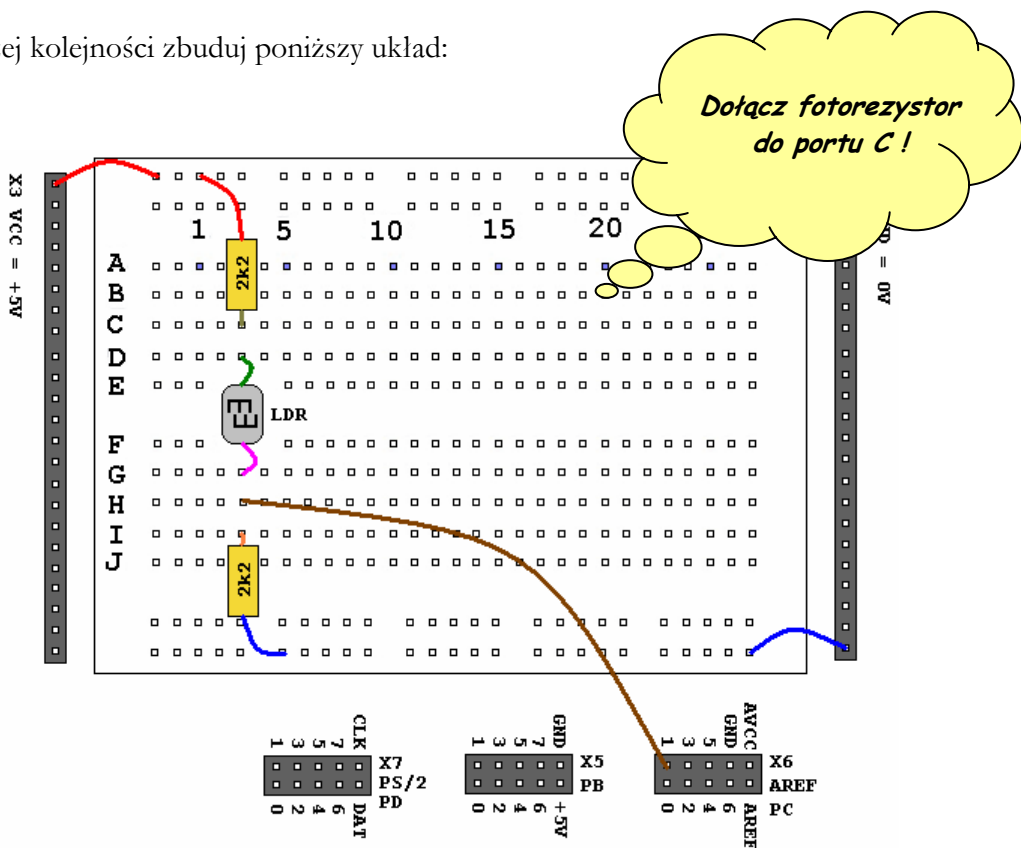
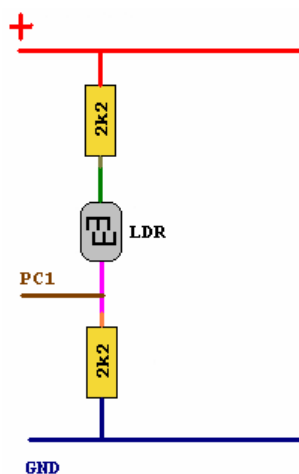
- Nauczysz się jak użyć przetwornik A/C
- Poznasz fotorezystory (LDR)

W tym rozdziale zobaczysz przetwarzanie A/C oraz zdobędziesz informacje do ćwiczenia 2 “Światlny przełącznik”.

Fotorezystory (LDR) są rezystorami, w których rezystancja zależy od padającego na nie światła. Do tego eksperymentu może zostać użytych wiele typów fotorezystorów.

Fotorezystor NSL19-MS51 przy świetle dziennym (bez jasnego słońca) posiada rezystancję 5k. O zmroku posiada rezystancję 1M2, a w ciemności rezystancja jest > 35M. Dla tego eksperymentu pasować będzie większość fotorezystorów. Jeśli użyjesz innego typu fotorezystora, w oknie terminala zobaczysz inne wartości.

W pierwszej kolejności zbuduj poniższy układ:



Zaprogramuj mikrokontroler ATmega88 programem EDBexperiment15b.bas oraz połącz płytkę EDB z komputerem szeregowym przewodem.

Powinieneś zobaczyć w oknie terminala wartości, które powinny się zmieniać przy różnym oświetleniu fotorezystora. Do oświetlenia fotorezystora możesz użyć latarki. Możesz również przy dużym oświetleniu zakrywać fotorezystor dłonią.

Przeczytaj uwagi w pliku EDBexperiment15b.bas i zobacz czy wszystko rozumiesz.

> Zaobserwuj co się dzieje, gdy linię PC1 dołączysz do masy oraz gdy ją dołączysz do $VCC = +5V$.

Zasoby internetowe:

Opis fotorezystora (LDR): http://www.radio-electronics.com/info/data/resistor/ldr/light_dependent_resistor.php

<http://www.technologystudent.com/elec1/ldr1.htm>

Ćwiczenie

2

Ćwiczenie 2, Świetlny przełącznik

Posiadane wiadomości z poprzednich rozdziałów przydadzą Ci się w tym ćwiczeniu.

Zbuduj świetlny przełącznik,

- Dołącz do Twojego mikrokontrolera diodę LED lub żarówkę (taką jaką możesz znaleźć w rowerze). Upewnij się **że nie dołączasz jej bezpośrednio**, ale użyj tranzystora BS170 jako przełącznika, aby uzyskać większy prąd.
- Jeśli dołączyłeś żarówkę, sprawdź czy możesz ją przełączać wykorzystując prosty program jaki był w eksperymencie 1.
- Teraz użyj fotorezystora z poprzedniego rozdziału, aby włączać żarówkę, kiedy jest ciemno i wyłączać kiedy jest jasno. Jeśli Ci się udało, gratuluje!!! Jeśli Ci się nie udało możesz zobaczyć rozwiązanie tego ćwiczenia. Znajdziesz je w pliku **Exercise2.bas** na płycie EDB-CD.

3.6.3 Tani woltomierz

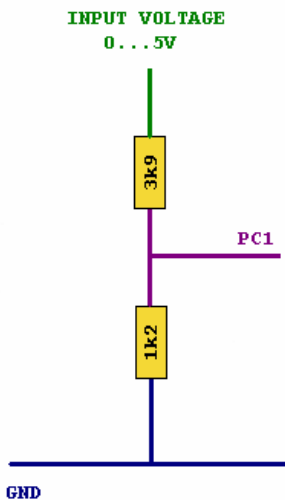
Lista elementów

- 1x Komputer z portem COM
- 1x Rezystory 1k2, 3k9
- 1x Potencjometr 10k
- 1x Woltomierz

Cele

- Nauczysz się jak używać przetwornika A/C
- Zobaczysz zastosowanie A/C

Teoria



Jeśli obserwowałeś wartości z przetwornika A/C w Twoim terminalu w eksperymencie 15, mogłeś zauważyć że przy napięciu 0 V wartość z przetwornika A/C wynosi 0, a przy napięciu 5 V wynosi 1024. Właściwie wartość 1024 z przetwornika jest już osiągnięta przy napięciu 1,1 V.

Teraz zbudujemy woltomierz z zakresem mierzonych napięć od 0 do około 5,0 V z rozdzielczością 0,1 V.

Teraz możemy mierzyć napięcia do 1,1 V, ale potrzebne jest kilka rezystorów do budowy dzielnika napięcia. Tą metodą uzyskamy że zmierzone napięcie 1 V odpowiadać będzie wejściowemu napięciu 5 V.

Rysunek obok pokazuje jakie wartości rezystorów są potrzebne do uzyskania potrzebnego dzielnika.

Wniosek jest taki, że musimy podzielić wartość 1024 przez 11 (1024/11) co daje 93,09. Preferowana jest praca z wartościami całkowitymi, gdyż mikrokontrolery nie potrafią dobrze dzielić. (Mikrokontrolery mogą dzielić z reszta $1024:11 = 93$ reszta 1)

V	Wartość A/C
1,1	1024
1	931
0,9	838
0,8	745
0,7	652
0,6	559
0,5	465
0,4	372
0,3	279
0,2	186
0,1	93
0	0

Tak więc zaokrąglimy wartość 93,09 do 93. Kiedy teraz będziemy *obliczać* wartość z A/C, dla każdego wolta możemy znaleźć wartość z przetwornika w tablicy konwersji po prawej stronie.

Powiedzmy że dołączamy napięcie do portu C.1 i odczytujemy z A/C wartość 190. Wartość 190 nie znajduje się w tablicy, więc możemy ją zaokrąglić.

Powiedzmy, że wszystkie wartości poniżej 93 są 0,0V.
Wartości od 93 do 185 są 0,1V,
Wartości od 186 do 278 są 0,2V, itd.

Jedyną rzeczą jaką musimy zrobić, to napisać program, który bada zakres wartości z A/C w określonym zakresie jak pokazano poniżej:

```

Test_value = 93      `(Początkowa wartość "93" ,gdyż poniżej wartości 93 jest 0,0V)
AD_value = 190
Voltage = 00

```

```

Not_ready:
  If AD_value ≤ Test_value then goto ready
  Test_value = Test_value + 93
  Voltage = Voltage + 1
goto not_ready

```

```

Ready:
Print Voltage

```

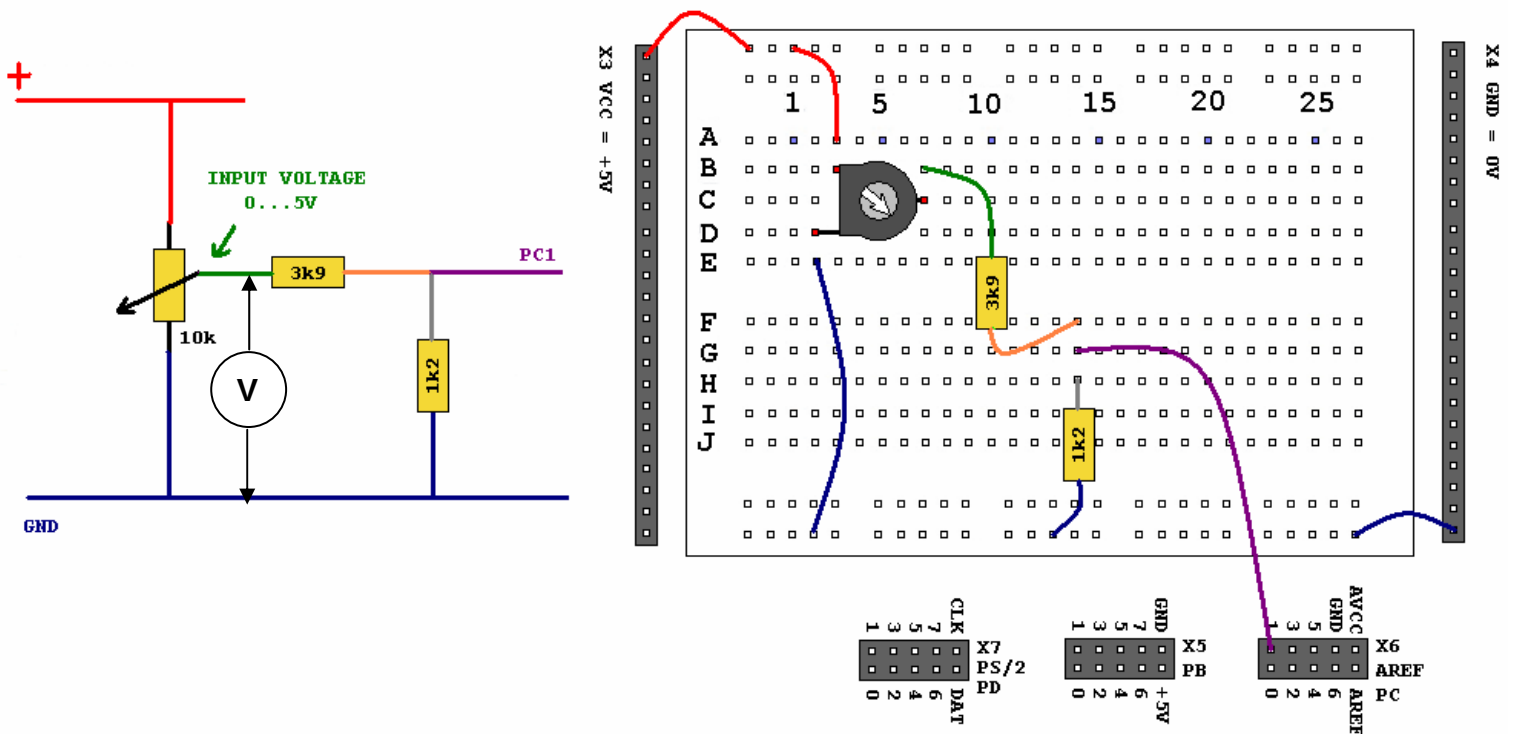
W przykładzie zmienna AD_value ma wartość 190, a wartość początkowa dla Test_value jest 93. Jeśli komenda IF jest FALSE (fałsz), to zostaje dodana wartość 93 i wartość 1 do voltage. (Teraz wiemy że zmierzone napięcie jest inne od 0,0 V).

Zmienna Test_value wynosi teraz 186 i IF jest nadal fałszem, więc napięcie nie wynosi 0,1 V. Wciąż ponownie zostaje dodana wartość 93 do test_value i 1 do zmiennej voltage.

Test_value wynosi teraz 279 i komenda IF jest TRUE (prawda)! Następuje skok do etykiety Ready. Jeśli wyświetlimy wartość Voltage zobaczymy 0,2 V. Już wiesz jak prawidłowo zinterpretować wartość z A/C bez używania dzielenia.

Jeśli użyjemy programu z powyższą metodą, można mierzyć napięcia od 0 do 1,1 V. Podział wejściowego napięcia przez 4 pozwoli mierzyć napięcia w pełnej skali.

→ Zaprogramuj mikrokontroler programem EDBExperiment16.bas, połącz płytkę z komputerem przewodem oraz zbuduj poniższy układ. **Zweryfikuj** pracę mikrokontrolera z teorią zmieniając położenie osi potencjometru i mierząc napięcie woltomierzem pomiędzy GND a PC1.



Ćwiczenie

3

Ćwiczenie 3, Woltomierz z bargrafem

Przez wykonanie tego ćwiczenia sprawdzisz znajomość poprzednich rozdziałów.

Zbuduj bargraf (graficzna linijkę) dla woltomierza z poprzedniego eksperymentu.

- Dołącz 8 diod LED do portu B. Jeśli napięcie wynosi $= 5\text{ V}$ wszystkie diody LED powinny być włączone. Jeśli napięcie wynosi $2,5\text{ V}$, powinny zostać włączone diody LED dołączone do PB0...4 itd.

Podpowiedź:

```
Select Case AD_value
  Case Is < 128 : Portb = &B11111110
  Case 128 To 256 : Portb = &B111111100
  Case 256 To 384 : Portb = &B111111000
  Case 384 To 512 : Portb = &B111110000
  Case 512 To 640 : Portb = &B111100000
  Case 640 To 768 : Portb = &B111000000
  Case 768 To 896 : Portb = &B100000000
  Case Is > 896 : Portb = &B000000000
End Select
```

- Dołącz również wyświetlacz LCD pokazujący mierzone napięcie.

Jeśli Ci się udało, gratulacje!!!. Jeśli nie to spójrz na rozwiązanie tego ćwiczenia. Znajdziesz je w pliku **Exercise3.bas** na płycie EDB-CD.

Ćwiczenie 4, Regulowane źródło światła

Przez wykonanie tego ćwiczenia sprawdzisz znajomość poprzednich rozdziałów.

Zbuduj regulowane źródło światła, a dokładnie powinieneś móc regulować jasność światła za pomocą dołączonego potencjometru.

- Dołącz do Twojego mikrokontrolera żarówkę (taką jaką możesz znaleźć w rowerze). Upewnij się **że nie dołączasz jej bezpośrednio**, ale użyj tranzystora IRF520 jako przełącznika, aby uzyskać relatywnie duży prąd.
- Jeśli dołączyłeś żarówkę, sprawdź czy możesz ją przełączać wykorzystując prosty program jaki był w eksperymencie 1
- Użyj sygnału PWM do zmiany jasności świecenia żarówki.
- Użyj potencjometru dołączonego do przetwornika A/C, aby można było regulować jasność świecenia żarówki.

Jeśli Ci się udało, gratulacje!!!. Jeśli nie to spójrz na rozwiązanie tego ćwiczenia. Znajdziesz je w pliku **Exercise4.bas** na płycie EDB-CD.

3.7 Inne elementy mikrokontrolera

W tym rozdziale będziemy dyskutowali o różnych elementach mikrokontrolera, które są często wykorzystywane, ale nie zobaczymy ich tylko w jednym eksperymencie. Zobaczysz jak używać pamięci, watchdoga oraz wewnętrznych czasomierzy.

3.7.1 Pamięci

Teoria

Lista elementów

- brak -

Cele

- Poznasz różne pamięci

Mikrokontroler ATmega88 posiada 3 różne pamięci:

- FLASH -> Bootloader FLASH
Aplikacja FLASH
- RAM
- EEPROM

Pamięć FLASH (8 kB) może być programowana **tylko** za pomocą zewnętrznego programatora STK200/300 lub USB ISP Prog I (rozdział 4).

Jednakże jest jeden wyjątek, którym jest Bootloader. Kiedy używasz bootloadera, pamięć FLASH dzielona jest na 2 obszary. Bootloader znajduje się w obszarze Bootloader FLASH i jest programowany programatorem STK lub USB. Wtedy obszar pamięci FLASH dla aplikacji może być programowany za pomocą szeregowego przewodu (rozdział 2.3).

W pamięci FLASH zapisywany jest program/oprogramowanie, które ma być wykonywane. Jeśli nie jest używany bootloader, cała pamięć FLASH jest dostępna dla programu sterującego. Pamięć FLASH jest **nie ulotna**, to znaczy że jej zawartość zachowana jest, gdy wyłączysz zasilanie.

Kiedy w pamięci FLASH zapisany jest Twój program/kody instrukcji, **pamięć RAM** używana jest do zapisu danych wejściowych, wyników obliczeń oraz logicznych operacji. Również zawartość pamięci RAM może być wysyłana i odbierana przez UART. Także w pamięci RAM Bascom zapisuje adres powrotu po wykonaniu instrukcji skoku (JUMP).

Pamięć RAM (Random Access Memory) jest pamięcią tymczasową. Po wykonaniu niektórych operacji potrzebujesz zapisać lub usunąć dane. Pamięć RAM jest

ulotna, to znaczy że jej zawartość jest tracona po wyłączeniu napięcia zasilania mikrokontrolera.

Specjalną formą pamięci jest **pamięć EEPROM** (Electrically Erasable/Programmable Read Only Memory). Stare pamięci EEPROM mogły być tylko czytane i służyły jako pamięć ROM mikrokontrolera. NIGDY nie mogły być kasowane przez mikrokontroler. Można je było tylko programować za pomocą specjalnego programatora i kasować światłem ultrafioletowym UV (Układy miały przezroczyste okienko).

Obecnie, mikrokontrolery mogą **elektrycznie** kasować i programować EEPROM, dając Tobie możliwość gromadzenia w nich danych, których się nie straci po wyłączeniu zasilania. Są to pamięci **nie ulotne**.

EEPROM jest polecany, kiedy twoja aplikacja musi zapamiętać ustawienia. Wyobraź sobie, co by się działo gdybyś musiał programować swój TV zawsze po włączeniu zasilania. Ale tego nie musisz robić, bo w Twoim telewizorze znajduje się pamięć EEPROM.

Eksperyment

17

Połącz płytkę EDB z komputerem szeregowym przewodem, zaprogramuj mikrokontroler programem w pliku EDBexperiment17.bas oraz włącz terminal znajdujący się z Bascom.

Ten program zachęca Cię do wpisywania znaków, które zapisywane są w pamięci EEPROM. Wykonuj instrukcje które pojawiają się na ekranie terminala. Zobacz co się dzieje, gdy wyłączysz zasilanie płytki i ponownie je włączysz.

Czy widzisz że wartości w pamięci EEPROM zostały zachowane?

Zasoby internetowe:

Rodzina mikrokontrolerów AVR firmy ATMEL:

http://www.atmel.com/dyn/products/param_table.asp?family_id=607

3.7.2 Układ Watchdog

Lista elementów

- brak -

Cele

- Poznasz działanie Watchdoga

Teoria

Czasomierz watchdog zabezpiecza działanie mikrokontrolera. Jeśli oprogramowanie mikrokontrolera przestanie prawidłowo pracować, watchdog po pewnym czasie wyzeruje mikrokontroler, aby program zaczął działać prawidłowo.

Jeśli mikrokontroler zawiesi się przez wykonanie nieprawidłowej instrukcji (przy pojawieniu się elektrycznego problemu) lub przestały pracować zewnętrzne układy, czasomierz watchdog nie zostanie wyzerowany. Kiedy tak się stanie watchdog przepelni się i wyzeruje mikrokontroler. Wtedy mikrokontroler zacznie wykonywać program od początku.

Typowa implementacja watchdoga może wyglądać następująco:

Config Watchdog = 2048
Start Watchdog

'zerowanie po 2048 mS
'włączenie watchdoga

Do
'Print "Hello"
Reset Watchdog
'Tutaj program w którym powtarzana będzie komenda Reset Watchdog
Loop
End

Jeśli chcesz możesz poeksperymentować z watchdog, którego implementację możesz znaleźć w pliku Watchdog.bas na płycie EDB-CD. Odblokuj linię "Print "Hello" oraz zablokuj linię "Reset watchdog". W twoim terminalu zauważysz cykliczne zerowanie mikrokontrolera.

Maksymalny czas watchdoga dla ATMega88 to 8 sekund. Ale dla większości aplikacji nie jest zalecane używanie tak długich czasów. Wystarczy czas 2048 lub 1024 mS.

3.7.3 Wykorzystanie czasomierza

*Lista elementów**-brak-**Cele**- Nauczysz się jak używać czasomierza*

Mikrokontroler ATmega88 ma wbudowane czasomierze. Możesz je użyć do mierzenia czasu lub zbudowania własnych procedur opóźniających. Zobaczysz przykład sekundowego czasomierza.

Połącz płytkę EDB z komputerem za pomocą szeregowego kabla, zaprogramuj mikrokontroler programem z pliku EDBexperiment18.bas oraz uruchom Bascomowy terminal.

Zobaczysz w oknie terminala sekundowy czasomierz. Przeczytaj uwagi z pliku EDBexperiment18.bas i zobacz czy wszystko rozumiesz co się dzieje.

Można zauważyć że program EDBexperiment18.bas jest dokładny dla czasu około 10 minut, co jest spowodowane użyciem mało dokładnego wewnętrznego oscylatora RC mikrokontrolera ATmega88. Jeśli chcesz uzyskać dokładniejszy czas, lepiej jest użyć prawdziwego zegara RTC jak na przykład DS1307 firmy Maxim-Dallas. Układ DS1307 posiada interfejs I²C. Jak używać I²C, nauczysz się w następnym rozdziale.

INFORMACJE O DOSTĘPNYCH CZASOMIERZACH

Mikrokontroler ATmega88 posiada następujące wbudowane czasomierze:

8-bitowy Czasomierz/Licznik0 z PWM

16-bitowy Czasomierz /Licznik1 z PWM

8-bitowy Czasomierz /Licznik2 z PWM i operacjami asynchronicznymi

Jak inicjować czasomierze i wykorzystywać zgłaszane przez nie przerwania, możesz znaleźć w pomocy Bascoma. Naciśnij klawisz F1 w Bascom AVR i znajdź w spisie "CONFIG TIMER".

3.8 Inne interfejsy

Oprócz UART jest wykorzystywanych wiele innych interfejsów. W tym rozdziale zobaczysz jak używać protokołu I²C oraz jak używać modułu USB, który jest dostępny dla zestawu EDB.

Teoria

3.8.1 Interfejs I²C

Lista elementów

1x Układ DS1624

2x Rezystor 4k7

1x Komputer z portem COM

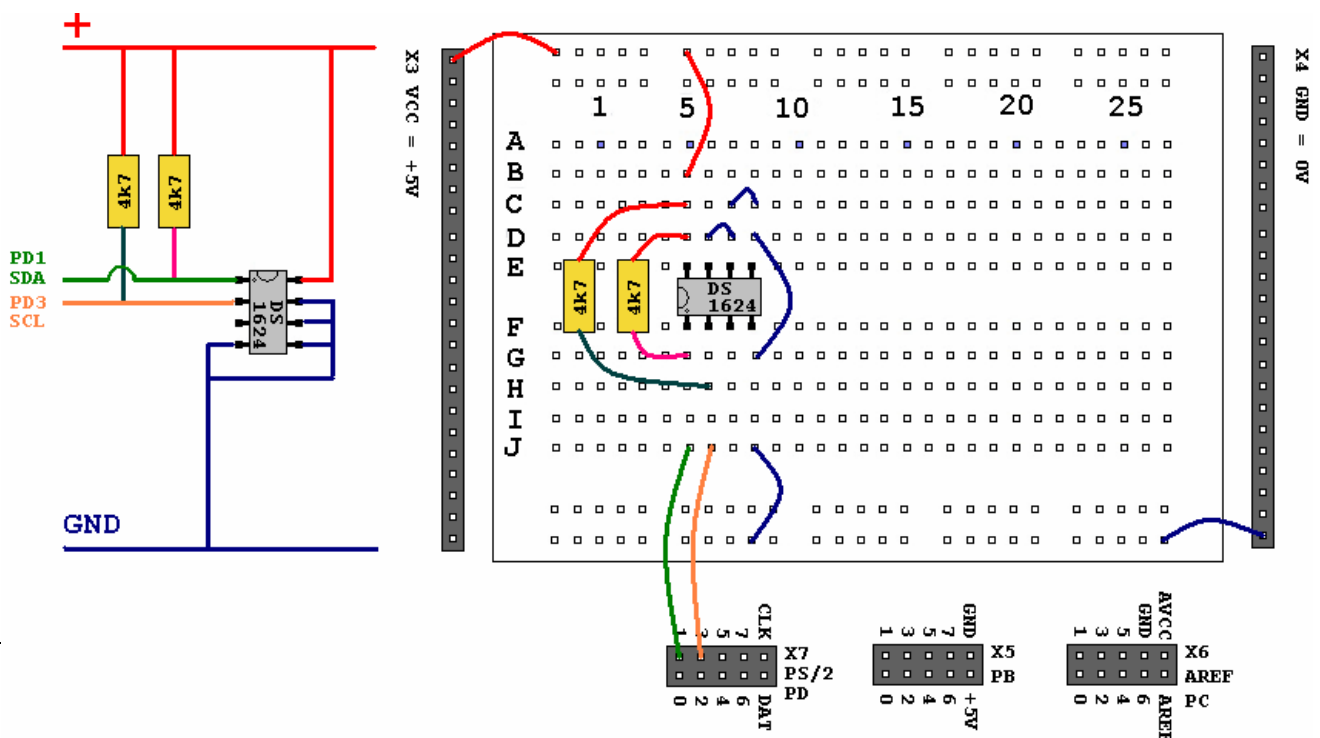
Cele

- Poznasz interfejs I²C

Układ DS1624 jest cyfrowym termometrem, który mierzy temperaturę od -55°C do +125°C. Jest on dołączany przez interfejs I²C. Szeregowy interfejs I²C posiada dwie linie komunikacyjne. Dwie linie komunikacyjne nazywane są Serial Data (SDA) oraz Serial CLock (SCL).

W interfejsie I²C mikrokontroler będzie zazwyczaj układem nadrzędnym (MASTER), a DS1624 układem podrzędnym (SLAVE). Sygnał zegara SCL generowany jest przez Mastera. Można dołączać do jednego Mastera wiele układów podrzędnych Slave.

Zbuduj układ pokazany na poniższym rysunku:



Teoria I²C (2-przewodowy interfejs)

Jeśli chcesz się dowiedzieć więcej o interfejsie I²C, przeczytaj dokumentację układu DS1307 firmy Maxim/Dallas. Układ DS1307 nie jest jedyny jaki można użyć w tym eksperymencie, może być każdy inny z identycznym interfejsem. Ale opis układu DS1307 jest bardzo dobry jeśli chcesz zrozumieć protokół I²C.

Opis układu DS1307 można znaleźć na płycie EDB-CD lub ściągnąć ze strony www.maxim-ic.com. Zacznij czytać od strony 5 sekcji “2-WIRE SERIAL DATA BUS”, nie czytając strony 9 i następnych (zapamiętaj że można użyć różnych układów, więc komendy i adresy nie będą takie same).

Zaprogramuj mikrokontroler ATmega88 programem z pliku EDBexperiment19.bas oraz połącz płytkę EDB z komputerem za pomocą szeregowego przewodu. .

W oknie terminala powinieneś widzieć mierzona temperaturę.

Przeczytaj uwagi z pliku EDBexperiment19.bas i zobacz czy wszystko rozumiesz.

Zasoby internetowe:

Opis I²C:

<http://www.semiconductors.philips.com/markets/mms/protocols/i2c/>

Obsługa interfejsu I²C w Windows:

<http://www.microsoft.com/whdc/archive/i2c.msp>

3.8.2 Interfejs USB

Chociaż większość komputerów posiada USB, nie jest on wykorzystywany często we własnych aplikacjach. W tym rozdziale poznasz jak we własnych aplikacjach korzystać z USB.

Po pierwsze powinieneś wiedzieć, że są 3 opcje wykorzystania we własnych urządzeniach interfejsu USB.

1. Urządzenie USB z wirtualnym portem COM

To jest najłatwiejsza opcja jeśli chcesz używać USB. Łączysz Twoje *urządzenie* do portu USB komputera, a system operacyjny będzie się z nim komunikował jak by był dołączony do portu COM.

Zalety:

- Łatwy w użyciu
- Kompatybilność z programami terminalowymi oraz innym oprogramowaniem

Wady:

- Użytkownik widzi nieprofesjonalny port COM
- Port COM może zostać źle skonfigurowany
- Użytkownik musi zainstalować 2 sterowniki.

2. Urządzenie USB z domyślnym VID&PID oraz sterownikiem

Ogólnie: Aplikacja uzyskuje dostęp do portu USB, kiedy posiada zgodne numery VID&PID (Vendor & part ID).

W tej drugiej opcji, możesz użyć sterowników dostarczanych przez Linx Technologies (www.linxtechnologies.com) aby uzyskać dostęp do urządzenia.

Zalety:

- Stosunkowo łatwy z użyciu

Wady:

- Będziesz musiał użyć VID&PID, które są zarejestrowane przez Linx.

3. Urządzenie USB z własnym VID&PID oraz sterownikiem

Z tą opcją możesz zbudować własne urządzenie. Jeśli kupisz dla siebie VID w MCS Electronics, także otrzymasz narzędzia programistyczne oraz plik instalacyjny *.inf. Jeśli kupisz VID gdzie indziej możliwe, że będziesz musiał napisać plik *.inf.

Zalety:

- Możesz użyć własnego VID&PID

Wady:

- Będziesz musiał kupić VID w MCS lub w konsorcjum USB
- Potrzebujesz napisać własny plik *.inf.

W większości przypadków, urządzenia USB są używane do współpracy z komputerem, dlatego prawie zawsze oprócz programu mikrokontrolera, będziesz musiał napisać obsługujący program dla komputera. To i koszty modułu USB (i opcjonalnie VID) mogą utrudniać implementację w urządzeniu interfejsu USB. Porty COM nie będą długo już dostępne w komputerach a moduły USB będą coraz bardziej powszechne i dostępne. To jest powód dlaczego został dodany rozdział o USB do instrukcji zestawu EDB.

Wykład o działaniu USB nie wchodzi w ramy tej instrukcji, ale jest dostępny pod adresem www.usb.org/developers. Opiszemy jak dołączyć EDB do USB w opcji pierwszej i drugiej. Pokazane zostaną także przykłady dla systemu Windows napisane w Visual Basic 6.

Trzecia opcja nie będzie opisywana, ale pokażemy narzędzia do zmiany pamięci EEPROM modułu USB w celu zmiany numerów VID&PID.

Zasoby internetowe:

Moduł USB: <http://www.linxtechnologies.com/>

Układ USB w module firmy Linx:

<http://www.ftdichip.com/Products/FT232BM.htm>

Opis USB : www.usb.org

Możliwość kupienia własnego VID: www.mcselec.com

3.8.3 Procedury instalacji



Jeśli chcesz użyć opcji 1 (wirtualny COM), przeprowadź instalację według opisu w rozdziale 3.8.3.1. Dla pozostałych opcji przeprowadź instalację opisaną w rozdziale 3.8.3.2.

Jeśli po opcji 1 (wirtualny COM), chcesz przejść do opcji 2 lub 3. Albo z opcji 2 lub 3 przejść do opcji 1 (wirtualny COM), musisz w pierwszej kolejności odinstalować poprzednie sterowniki. Instrukcje odinstalowania sterowników można znaleźć w rozdziale 3.8.3.3

Dopilnuj aby Twój komputer miał sprawny port USB. Porty USB nie będą pracowały pod Windows NT oraz z różnymi wersjami Windows 95. Doradzamy byś używał, dołączając płytkę EDB, tego samego portu USB. Unikniesz wtedy ponownej instalacji sterowników.

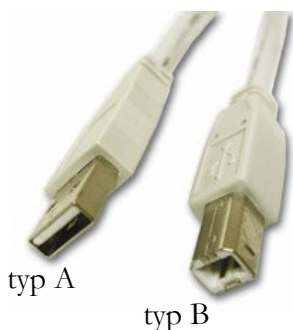
3.8.3.1 Instalacja wirtualnego portu COM dla Windows

Dołącz zasilanie do Twojej EDB oraz dołącz ją do portu USB komputera przewodem USB-A/USB-B.

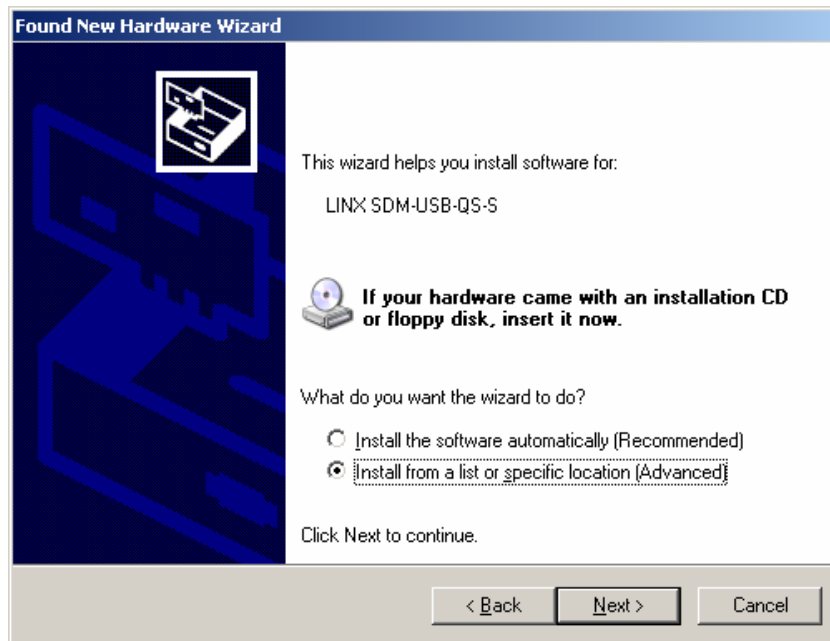
Powinieneś wtedy zobaczyć okno:



Wybierz “No, not this time” i kliknij przycisk “Next”

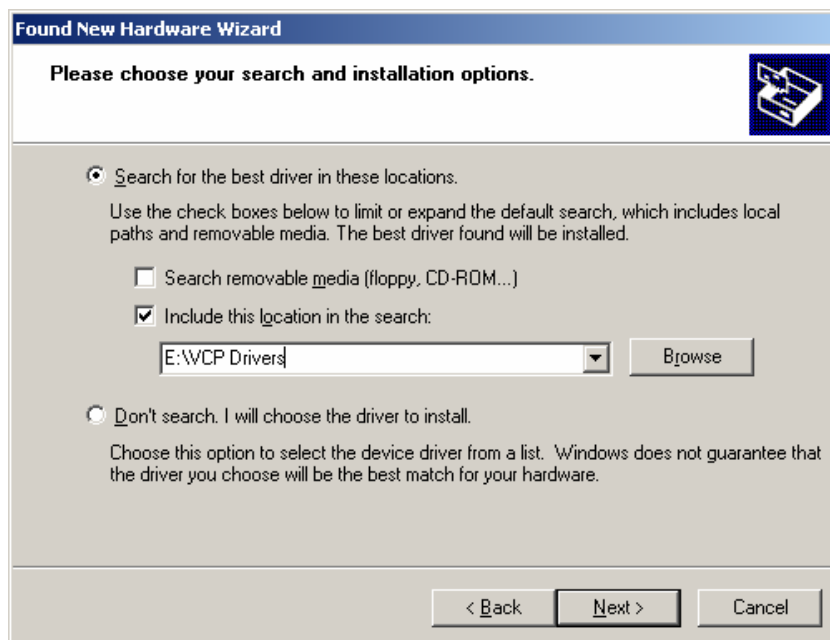


Po chwili powinieneś zobaczyć okno:



Wybierz “Install from a list or specific location (Advanced)” i kliknij przycisk “Next”

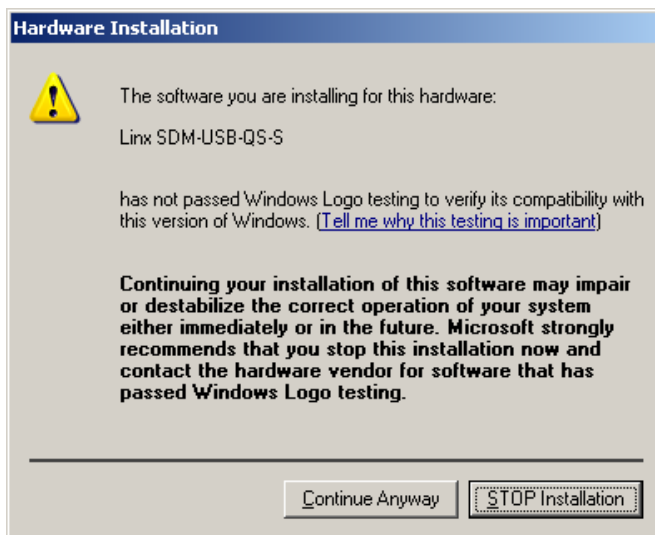
Po kolejnej chwili zobaczysz następujące okno:



Włóż płytę EDB-CD i użyj “Browse” aby wybrać folder:

Twój napęd CD-rom:\USB_Module\Virtual_COM_Drivers\ i kliknij “Next”

Po momencie zobaczysz następujące okno:



Okienko to jest ostrzeżeniem, że sterownik nie przeszedł procesu certyfikacji Microsoftu i może stanowić problemem dla systemu. Sterowniki modułu USB zostały przetestowane i nie powinny stanowić żadnych problemów dla systemu operacyjnego, chyba że sterowniki zostały zmodyfikowane przez użytkownika. Tak więc kliknij na przycisk "Continue Anyway".

Po chwili pojawi się okno z informacją że "instalacja oprogramowania została zakończona", po czym można kliknąć przycisk Finish.

Nie odłączaj płytki od USB lub jej zasilania, gdy instalacja nie została zakończona.

Wtedy komputer ponownie wykryje nowy sprzęt. Właściwie to teraz zostaje znaleziony wirtualny port COM. Aby zainstalować wirtualny port COM, będziesz musiał ponownie wykonać wszystkie kroki instalacyjne, pokazane na poprzednich stronach (zaczynij czytanie od punktu 8.3.1 "Found new hardware wizard").

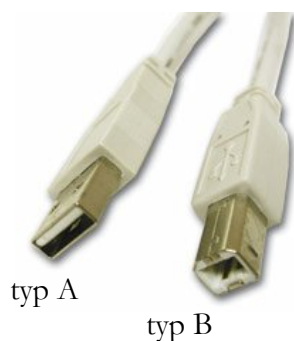
Wirtualny port COM jest teraz zainstalowany.

-> Kontynuuj czytanie rozdziału 3.8.4

3.8.3.2 Instalacja urządzenia USB dla Windows

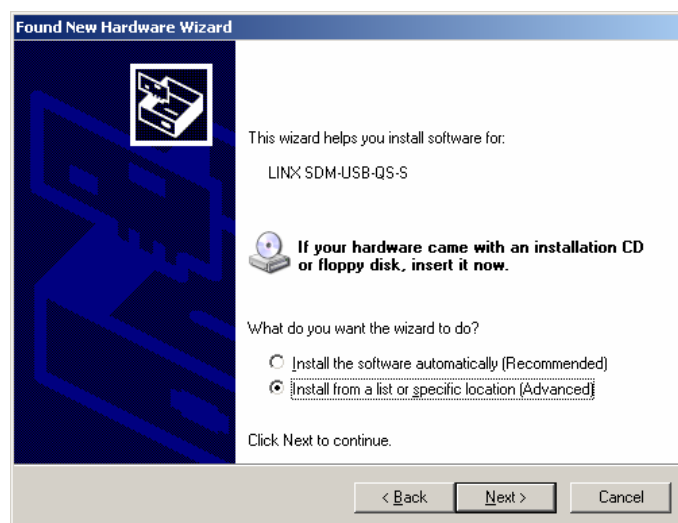
Dołącz zasilanie do Twojej EDB oraz dołącz ją do portu USB komputera przewodem USB-A/USB-B.

Powinieneś wtedy zobaczyć okno:



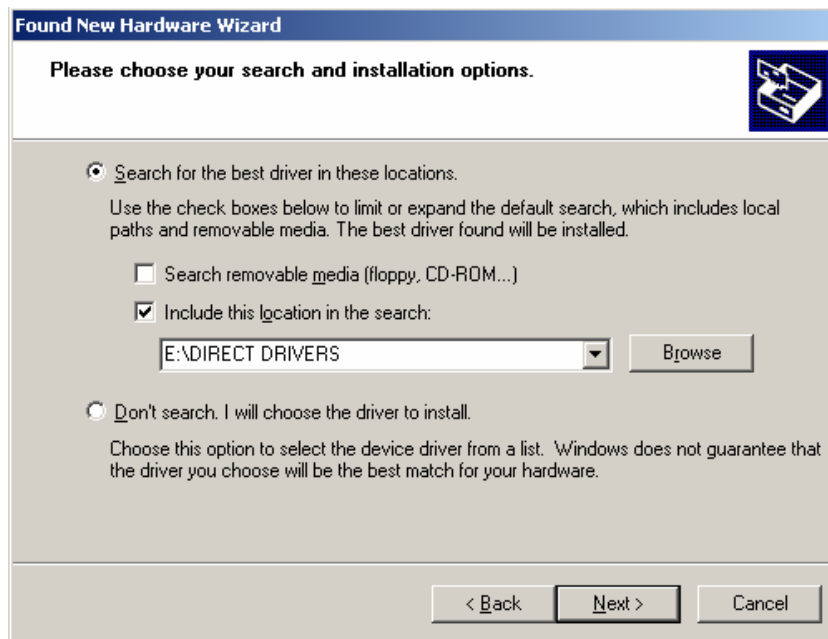
Wybierz “No, not this time” i kliknij przycisk “Next”

Za moment zobaczysz następujące okno:



Wybierz “Install from a list or specific location (Advanced)” i kliknij “Next”

Po momencie zobaczysz następujące okno:



Włóż płytę EDB-CD i użyj “Browse” aby wybrać folder:

Twój napęd CD-rom: \USB_Module\Direct_Drivers\ i kliknij “Next”

***Jeśli programowałeś własny VID&PID,
musisz wyszczególnić go w instalowanych sterownikach.***

Po momencie zobaczysz następujące okno:



Okienko to jest ostrzeżeniem ze sterownik nie przeszedł procesu certyfikacji Microsoftu i może stanowić problemem dla systemu. Sterowniki modułu USB zostały przetestowane i nie powinny stanowić żadnych problemów dla systemu operacyjnego, chyba że sterowniki zostały zmodyfikowane przez użytkownika. Tak więc kliknij na przycisk "Continue Anyway".

Po chwili pojawi się okno z informacją że "instalacja oprogramowania została zakończona", po czym można kliknąć przycisk Finish.

Urządzenie USB jest teraz zainstalowane.

-> Kontynuuj czytanie rozdziału 3.8.5

3.8.3.3 Odinstalowanie sterownika

Jeśli po opcji 1 (wirtualny COM), chcesz przejść do opcji 2 lub 3. Albo z opcji 2 lub 3 przejść do opcji 1 (wirtualny COM) musisz w pierwszej kolejności odinstalować poprzednie sterowniki. W tym rozdziale znajdziesz informacje jak to zrobić

W pierwszej kolejności włóż do napędu płytę EDB-CD.

Jeśli zainstalowałeś sterowniki dla wirtualnego portu COM.

Użyj Explorera plików w celu odczytu Twojego CD-romu.

Otwórz folder “VCP Drivers folder” oraz uruchom plik “FTDIUNIN.EXE”

USB_Module\Virtual_COM_Drivers\FTDIUNIN.EXE

Powinieneś zobaczyć okienko:



Kliknij przycisk “Continue”, a za moment przycisk “Finish”. Jeśli chcesz zainstalować sterowniki dla urządzenia USB zobacz rozdział 3.8.3.2.

Jeśli zainstalowałeś sterowniki dla urządzenia USB.

Użyj Explorera plików w celu odczytu Twojego CD-romu.

Otwórz folder “Direct Drivers” oraz uruchom plik “FTD2XXUN.EXE”

USB_Module\Direct_Drivers\FTD2XXUN.EXE

Powinieneś zobaczyć okienko:



Kliknij “Continue”, a za moment przycisk “Finish”. Jeśli chcesz zainstalować sterowniki dla wirtualnego portu COM patrz rozdział 3.8.3.1.

3.8.4 Interfejs USB, urządzenie z wirtualnym portem COM

Lista elementów

- 1x Kabel USB-A/USB-B
- 1x Komputer z portem USB

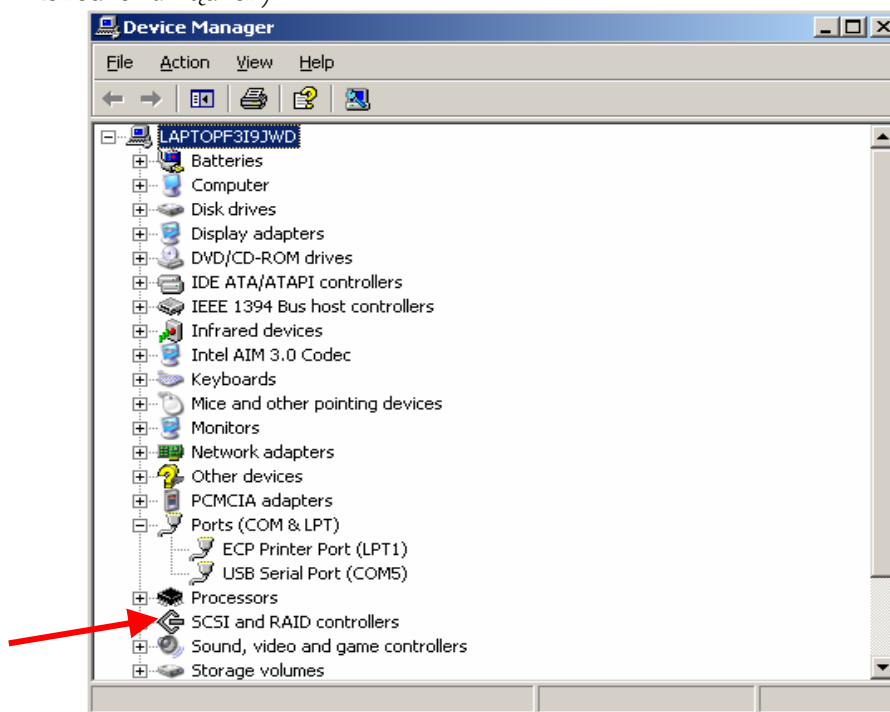
Cele

- Nauczysz się korzystać z wirtualnego portu COM

Ten eksperyment może być przeprowadzony, gdy zainstalowałeś sterowniki opisane w rozdziale 3.8.3.1. **Dopilnuj aby przełącznik “S1 USB/RS232”, był ustawiony w pozycji USB.**

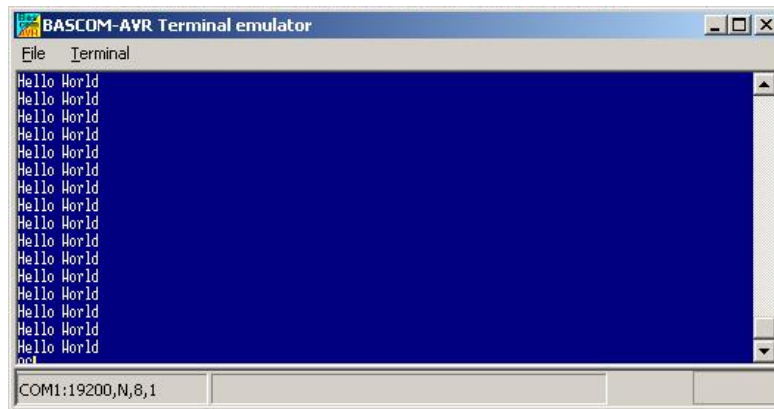
Otwórz plik EDBexperiment6a.bas z płyty EDB-CD oraz zaprogramuj mikrokontroler ATmega88. Używamy tego samego programu co w eksperymencie 6a, aby przetestować wirtualny port COM. Gdy zaprogramowałeś mikrokontroler i podłączyłeś płytkę EDB kablem do portu USB, uruchom emulator terminala klikając w Bascom ikonę  .

Ustawienia terminala nadal powinny być: 19200 baud, brak bitu parzystości, 8 bitów danych oraz 1 bit stopu. Dopilnuj, aby używać odpowiedniego (wirtualnego) portu COM, którego numer możesz znaleźć w panelu sterowania (System, Sprzęt, i Menedżer urządzeń).



W tym przypadku musisz użyć COM5, ale w Twoim systemie może być inaczej.

W przypadku programu EDBexperiment6a.bas na ekranie Twojego PC powinieneś zobaczyć tekst “Hello World”:



Jest możliwe napisanie własnego oprogramowania dla Windows/Linux itd. Jeśli używasz Windows, możesz użyć do tego Visual Basic (Basic jest w Bascom AVR). Możesz także użyć innych języków popieranych przez system operacyjny jak na przykład C++ Borland Builder, Pascal oraz jakiegokolwiek inny, który umożliwia użycie komponentów ActiveX.

Używamy komponentu ActiveX aby wysyłać i otrzymywać dane z portu COM dla aplikacji Windows/Linux itd. Sugerujemy używanie darmowego komponentu OCX (mcscomm.ocx) dostarczanego przez MCS Electronics. (Microsoft dostarcza mscmm32.ocx, ale MS OCX nie pracuje ze zdarzeniami).

Przykładowe programy możesz znaleźć w katalogu Visual Basic na płycie EDB-CD. W tym katalogu znajduje się także plik mcscomm.ocx.

3.8.5 Interfejs USB, urządzenie z domyślnym PID&VID

Lista elementów

- 1x Kabel USB-A/USB-B
- 1x Komputer z portem COM

Cele

- Nauczysz się korzystać z urządzenia z domyślnym PID&VID

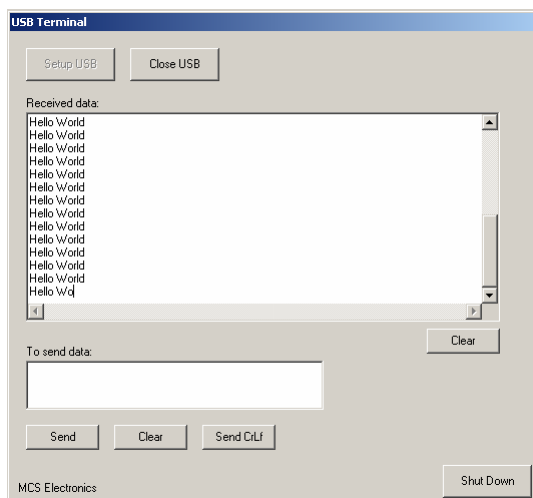
Ten eksperyment może być przeprowadzony tylko, gdy zainstalowałeś sterowniki opisane w rozdziale 3.8.3.2.

Dopilnuj aby przełącznik “S1 USB/RS232”, był ustawiony w pozycji USB.

Otwórz plik EDBexperiment6a.bas z płyty EDB-CD oraz zaprogramuj mikrokontroler ATmega88. Używamy tego samego programu co w eksperymencie 6a, aby przetestować interfejs USB z domyślnym PID&VID.

Teraz uruchom program “USB Terminal” znajdujący się w katalogu “Visual_Basic\USB_Terminal” na płycie EDB-CD.

Kliknij “Setup USB”, po czym powinieneś zobaczyć na ekranie tekst “Hello World”. Spróbuj to zrobić również z plikiem EDBexperiment6b.bas.



Jest możliwe napisanie własnego oprogramowania dla Windows/Linux itd. Jeśli używasz Windows możesz użyć do tego Visual Basic (Basic jest w Bascom AVR). Możesz także użyć innych języków popieranych przez system operacyjny jak na przykład C++ Borland Builder, Pascal oraz jakiegokolwiek inny, który umożliwia użycie komponentów DLL.

Używamy komponentu DLL do wysyłania oraz odbierania danych z portu USBn dla aplikacji w Windows/Linux itd.

Przykładowe programy możesz znaleźć w katalogu Visual Basic na płycie EDB-CD. W tym katalogu znajduje się także plik DLL. Więcej przykładów w różnych językach programowania można znaleźć na stronie FTDI:
<http://www.ftdichip.com/Projects/CodeExamples.htm>.

3.8.6 Interfejs USB, urządzenie z własnym PID&VID

Ten rozdział jest przeznaczony dla zaawansowanych programistów, którym krótko pokazano zaawansowane możliwości korzystania z USB. To nie jest przewodnik, który pokazuje wszystkie kroki jakie należy wykonać, aby posiadać własny numer PID urządzenia. .

OSTRZEWNIENIE: Zmiana zawartości pamięci EEPROM (PID&VID) w module USB może spowodować, że układ będzie niedostępny. Zmiany w EEPROM oraz sterownikach są wprowadzane na własne ryzyko. MCS Electronics nie odpowiada za zmiany wprowadzone do EEPROM oraz za zmiany w sterownikach.

Pamięć EEPROM może być programowana, gdy zostały zainstalowane sterowniki opisane w rozdziale 3.8.3.2.

Interfejs USB zrealizowany został za pomocą modułu USB SDM-QS-S1-S firmy Linx Technologies, który posiada wbudowany układ FT232xx firmy FTDI.

Tutaj znajduje się strona internetowa firmy Linx: www.linxtechnologies.com
Tutaj znajduje się strona internetowa firmy FTDI: www.ftdichip.com

Możesz czytać oraz zapisywać pamięć EEPROM modułu USB SDM-QS-S1-S. Daje to Tobie możliwość zmiany numerów PID&VID.

Polecamy odwiedzenie wspomnianych wyżej stron i przeczytanie zawartych tam instrukcji przed programowaniem pamięci EEPROM modułu USB. Aplikację przeznaczoną do programowania EEPROM, możesz znaleźć w katalogu "USB Module" na płycie EDB-CD, jak również na stronach producentów.

Zanim możesz użyć własnego numeru PID urządzenia w Windows/Linux, itd., w pierwszej kolejności musisz zmodyfikować plik *.inf, znajdujący się w plikach sterownika. Jako przykład możesz przeczytać plik *.inf, który możesz znaleźć na płycie EDB-CD.

Zasoby internetowe:

Własny numer VID możesz kupić na stronie: www.mcselec.com

3.9 Sterowanie silnikami

Oprócz komunikacji z użytkownikiem, transmisji danych, mikrokontrolery mogą również sterować dołączonymi do niego układami wykonawczymi. Popularnymi urządzeniami wykonawczymi są silniki DC oraz krokowe. Ten rozdział pokaże Ci jak sterować silnikiem DC oraz silnikiem krokowym.

3.9.1 Sterowanie silnikiem krokowym

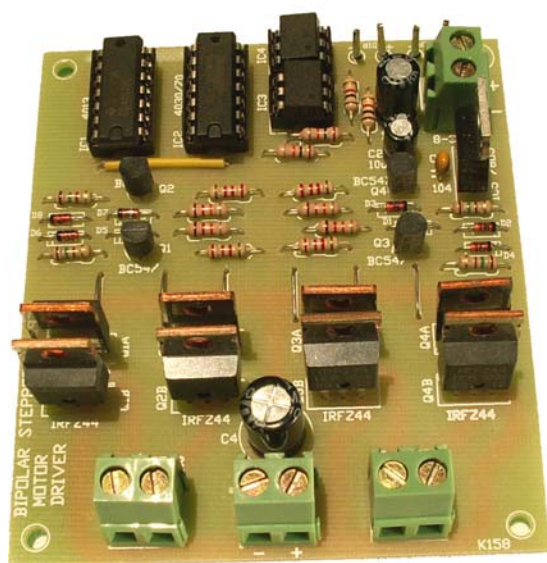
Lista elementów

- 1x Bi-polarny silnik krokowy
- 1x Sterownik silnika krokowego

Cele

- Nauczysz się jak używać silnika krokowego

Do tego eksperymentu używany jest sterownik dla silnika krokowego „Quasar Bi-Polar Stepper Motor Driver 3158” o którym więcej informacji można znaleźć na stronie www.quasarelectronics.com/kit-files/3000/3158.pdf.



Jeśli nie posiadasz silnika krokowego, możesz taki znaleźć w stacji dyskiek. Tam są silniki nie bipolarnie, więc są one dobre jedynie dla eksperymentów (w połączeniu z tym układem sterownika), ale nie są dobre do pracy w rzeczywistych aplikacjach.

Dołącz silnik krokowy do złącza “TO MOTOR” układu sterownika.

Dołącz Portd.0 do pinu “STEP +”,
dołącz Portd.1 do pinu “DIR +”,
dołącz GND z EDB do “STEP -” oraz “DIR -”

Dołącz napięcie 8...30 V do obu gniazd zasilających na płycie układu sterownika. Zacznij od niskiego napięcia, jeśli nie znasz opisu Twojego silnika (jest możliwe także zastosowanie galwanicznej separacji, której opis znajdziesz w pliku 3158.pdf).

Jeśli używasz silnika krokowego ze stacji dyskiek, masz tylko 3 przewody do dołączenia silnika. Wtedy musisz zostawić jedną szpilkę w złączu “TO MOTOR” nie podłączoną.

Zaprogramuj ATmega88 programem EDBexperiment22. Zobacz jak silnik obraca się raz w jedną stronę, i raz w drugą stronę. Przeczytaj uwagi znajdujące się w pliku EDBexperiment22.bas i zobacz czy wszystko rozumiesz.

3.9.2 Sterowanie sygnałem PWM silnika DC

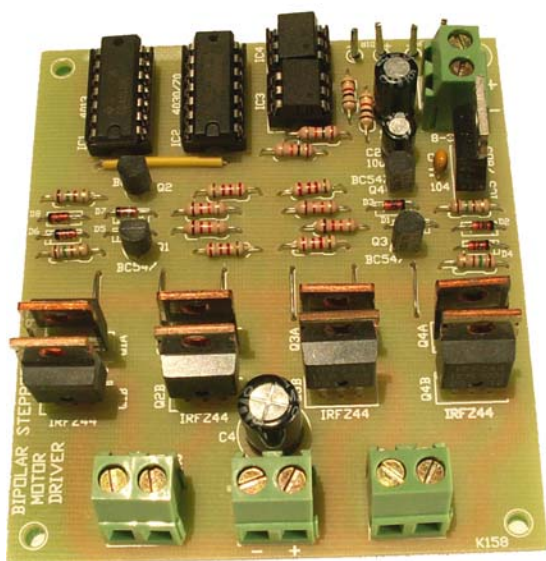
Lista elementów

- 1x Silnik DC
- 1x Sterownik silnika krokowego

Cele

- Nauczysz się jak używać silnika DC

Do tego eksperymentu używany jest sterownik silnika krokowego „Quasar Bi-Polar Stepper Motor Driver 3158” o którym więcej informacji można znaleźć na stronie www.quasarelectronics.com/kit-files/3000/3158.pdf.



Jeśli nie masz silnika DC, możesz użyć takiego silnika z magnetofonu kasetowego.

Dołącz silnik DC do jednego złącza “TO MOTOR” sterownika.

Dołącz pin Portd.0 do pinu “STEP +”,
dołącz pin Portd.1 do pinu “DIR +”,
dołącz GND z EDB do “STEP -” oraz “DIR-”

Dołącz napięcie 8...30 V do obu gniazd zasilających na płytce układu sterownika. Zacznij od niskiego napięcia jeśli nie znasz opisu Twojego silnika (jest także możliwe zastosowanie galwanicznej separacji, której opis znajdziesz w pliku 3158.pdf)

Zaprogramuj ATMega88 programem EDBexperiment22. Zobacz jak silnik obraca się chwilę w jedną stronę po czym zmienia kierunek obrotów na przeciwny.

Przeczytaj uwagi znajdujące się w pliku EDBexperiment22.bas i zobacz czy wszystko rozumiesz.

Zasoby internetowe:

Sterownik silnika: http://www.quasarelectronics.com/motor_controllers_drivers.htm

4. Inne metody programowania

W tym rozdziale znajdziesz sposób programowania mikrokontrolera ATmega88 programatorem równoległym oraz z interfejsem USB.

Morzesz zaprogramować mikrokontroler ATmega88 używając programatora równoległego lub z interfejsem USB. Są trzy powody dla użycia tego sposobu.

1. Nie posiadasz portu szeregowego,
2. Twój mikrokontroler posiada wykasowaną pamięć FLASH,
3. Chcesz wykorzystać całą dostępną pamięć na program.

To nie jest możliwe, kiedy używany jest bootloader dla którego wydzielono określona wielkość pamięci FLASH.

W rozdziale 4.3 znajduje się opis jak przeprogramować bootloader.

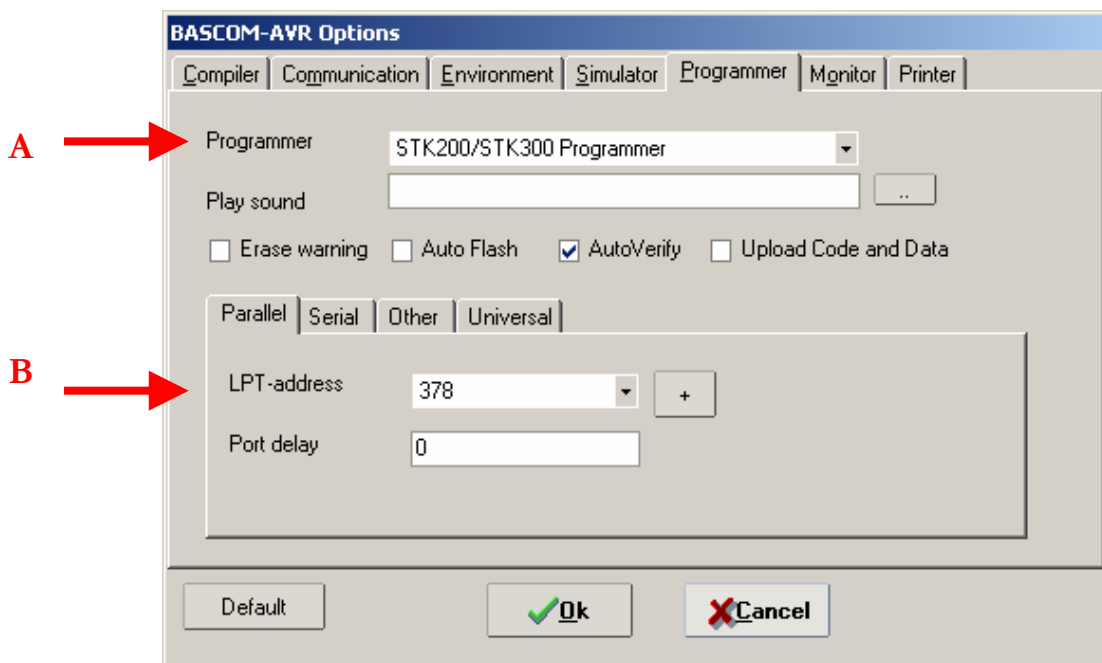
4.1 Programowanie programatorem STK200/300 ISP

Możesz kupić programator STK200/300 w MCS Electronics, ale możesz także go zbudować samemu. Schemat programatora STK możesz znaleźć w dodatku 1.

- Zanim będziesz kontynuował, sprawdź ustawienia portu równoległego w BIOS, który powinien być ustawiony w tryb ECP. Programator nie pracuje w innych trybach pracy portu, jak na przykład w trybie dwukierunkowym.

Dołącz programator STK200/300 do portu równoległego komputera oraz do złącza X9 oznaczonego “ISP” zestawu EDB.

W Bascom kliknij menu “Options” i “Programmer”. Zobaczysz następujące okno:



- W pierwszej kolejności wybierz “STK200/STK300 Programmer” z menu (A) .
- Następnie wybierz adres portu LPT. Jeśli używasz LPT1 to domyślnym adresem jest “378” a dla LPT2 adres “278” (B).
- Następnie kliknij przycisk OK.

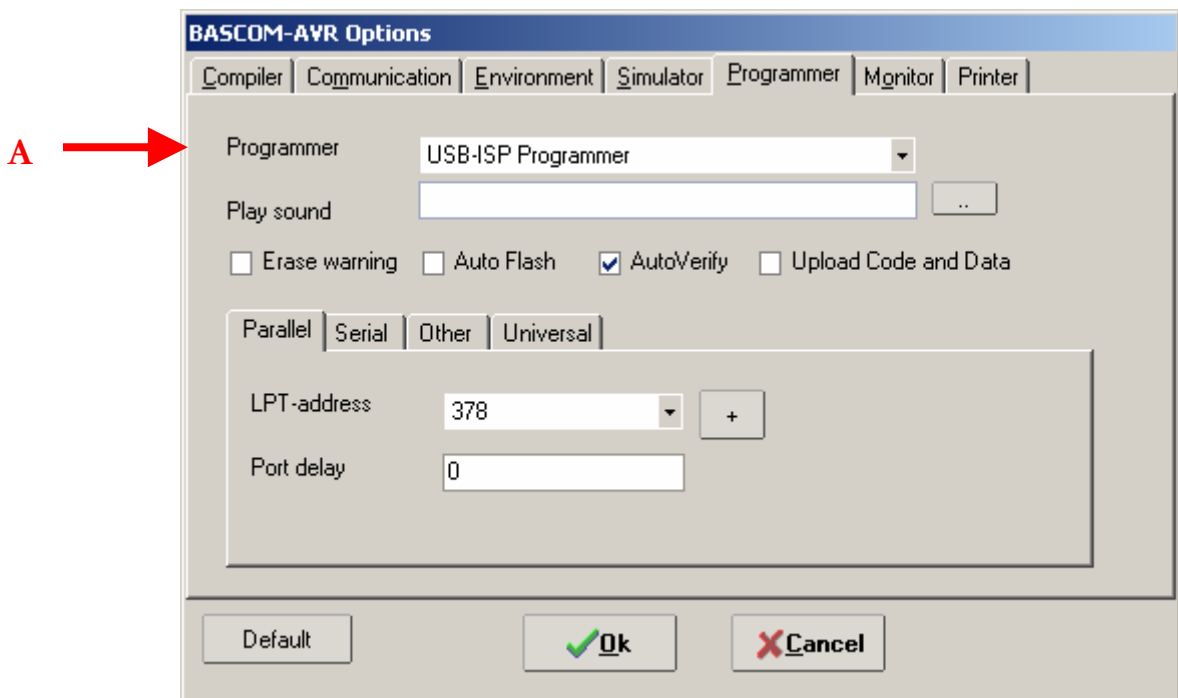
Jeśli chcesz przeprogramować bootloader, kontynuuj czytanie rozdziału 4.3, w przeciwnym razie możesz zaprogramować mikrokontroler w sposób opisany w rozdziale 2.3 zatytułowanym **“Programowanie mikrokontrolera”**.

4.2 Programowanie programatorem USB ISP Prog I

Możesz kupić programator USB ISP Prog I w MCS Electronics. Programator USB ISP posiada te same możliwości co STK200/300, ale zamiast portu równoległego używa wygodnego portu USB.

Dołącz programator USB ISP Prog I do portu USB komputera oraz przewodem do złącza X9 oznaczonego “ISP” płytki EDB.

W Bascom kliknij menu “Options” i “Programmer”. Zobaczysz następujące okno:



- Wybierz “USB-ISP Programmer” z menu oznaczonego (A).
- Następnie kliknij przycisk OK.

Jeśli chcesz przeprogramować bootloader, kontynuuj czytanie rozdziału 4.3, w przeciwnym razie możesz zaprogramować mikrokontroler w sposób opisany w rozdziale 2.3 zatytułowanym **“Programowanie mikrokontrolera”**.

4.3 Przeprogramowanie bootloadera

Zastosuj metodę programowania z rozdziału 4.1 lub 4.2.

- Otwórz plik bootloader.bas (EDB-CD) w Bascom AVR.
- Skompiluj program bootloadera (naciśnij F7)
- Sprawdź czy dobrze podłączyłeś programator opisany w rozdziale 4.1 lub 4.2, po czym zaprogramuj mikrokontroler (naciśnij F4)
- Zaprogramuj bity Fuse mikrokontrolera jak opisano poniżej.
- Na końcu wybierz “MCS Bootloader” z listy programatorów jak opisano w rozdziale 2.3

Programowanie bitów Fuse

Rodzina mikrokontrolerów AVR ATMega posiada tak zwane bity bezpieczników (Fuse) za pomocą których można ustawiać zaawansowane opcje układu. Normalnie są one programowane raz, choć mogą być zaprogramowane wiele razy. Możesz zaprogramować bity Fuse w Bascom z wykorzystaniem programu obsługującego programator.

Możesz wykonać pokazane na tej stronie kroki, jeśli jeden z poniższych warunków jest spełniony:

1. Nie używasz ATMega88 dostarczanego przez MCS Electronics lub zostały zmienione bity “lock” układu dostarczanego przez MCS Electronics.
2. Programowanie nie pracującego mikrokontrolera w sposób pokazany na poprzednich stronach.

Do zaprogramowania bitów “Fuse” potrzebujesz programatora STK200/300 lub USB ISP Prog I.

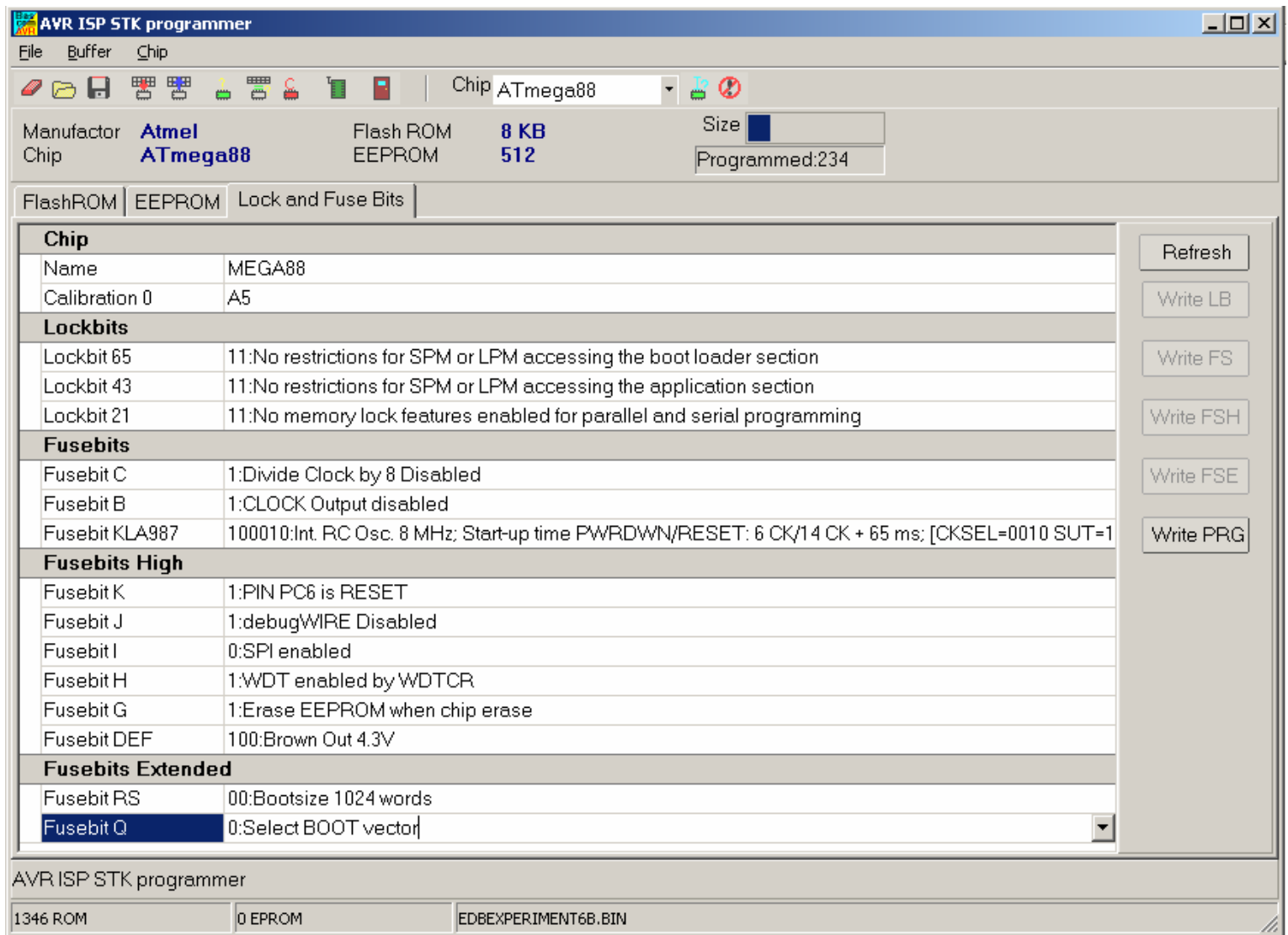
Podane niżej kroki doprowadzą do ustawienia bezpieczników mikrokontrolera ATMega88 do pracy w zestawie EDB.

Uruchom Bascom IDE, oraz utwórz nowy lub otwórz dowolny plik *.bas.

Otwórz program obsługi programatora jak pokazano obok:



Wybierz zakładkę “Lock and Fuse Bits” oraz zmaksymalizuj okno programatora.



Upewnij się że widzisz zakładkę “fuse and lock bits” oraz naciśnij przycisk “Refresh”, aby zobaczyć aktualne ustawienia. Fusebit Q powinien być ustawiony na BOOT.

Ostrzeżenie, uważaj na Fusebit KLA987, który konfiguruje oscylator. Powinien posiadać wartość “100010: Int. RC Osc.... SUT = 1”. Jeśli zmienisz te opcje, mikrokontroler nie będzie pracował i już nie zmienisz tego z powrotem bez zewnętrznego oscylatora częstotliwości (generatora częstotliwości). Dopilnuj aby były ustawione na 100010.

Zmienione bity możesz zapisać naciskając przycisk “Write XXX” dostępny dla zmienionych bitów.

Atmel AVR In System Programming Dongle

Atmel offers a software package called the Atmel AVR ISP which allows the programming of AVR Microcontrollers in circuit with a simple dongle which is attached to the Parallel Port. This dongle is detailed below. It can be built cheaply, making it an ideal starting point for developing with ATMEL AVR micros.

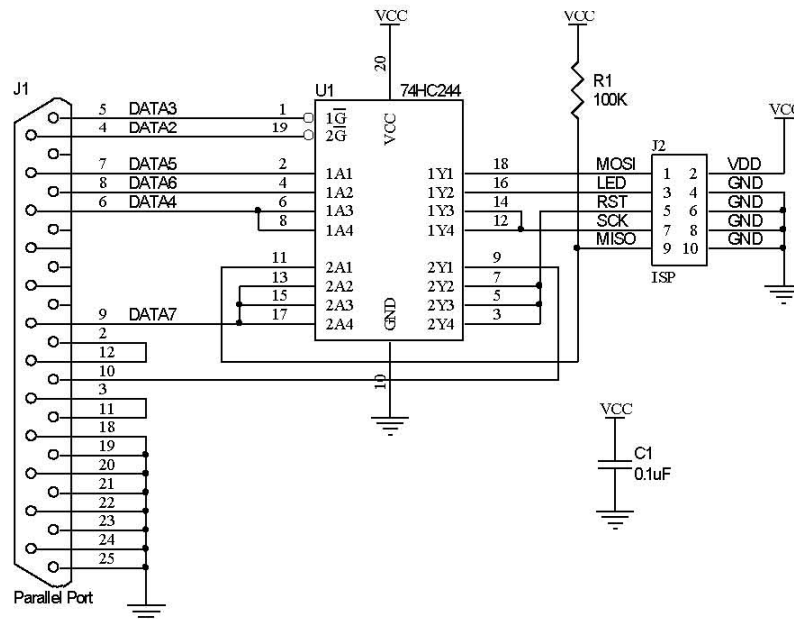


Figure 1 : Schematic for the ATMELE ISP Value Added Dongle

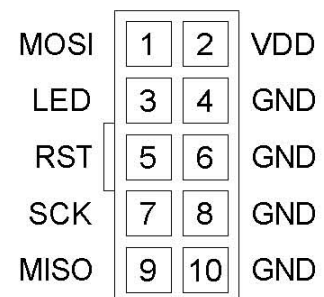
Using a 74HC244 Tri-State buffer as the main component, operation is extremely simple. The two loopback connections, pin 2 to 12 and 3 and 11 is used to identify the dongle. With both links in place the dongle is identified as a Value Added Pack Dongle. With only pins 2 and 12 links, it is reported as a STK300 or AVR ISP Dongle. With only 3 and 11 the dongle is reported as an STK200 or old Kanda ISP Dongle.

DATA2 and DATA3 of the Parallel Port Drive the TriState Outputs. A low will allow the passing of the serial clock and data during programming. MOSI, LED, SCK and Reset being outputs are buffered from the Parallel Port's DATA5, DATA6, DATA4 and DATA7 Respectively. The only input, MISO is fed into nACK, a status input of the Parallel Port.

There are two standard ISP Connectors for Atmel AVR Microcontroller ISP Programming. One standard is the 10 pin version using a DIL 5x5 header of 0.1" Pitch, shown in the above schematic. This is used on the ATMEL STK Kits. The other is a more compact 6 pin version, once again using a DIL 3x3 header of 0.1" pitch. This 6 pin version is the standard connector for ATMEL ISP Programmers.

The main advantage of the 10 pin header is the clean and easy use of 10 pin IDC crimp headers.

Name	Function	Description
MOSI	Master Out - Slave In	Data being transmitted to the part being programmed is sent on this pin
LED	Program LED	Optional Programming LED
RST	Target MCU Reset	Connects to Target AVR. Target AVR is programmed while in Reset State.
SCK	Shift Clock	Serial Clock Generated by the Programmer
MISO	Master In - Slave Out	Data received from the part being programmed is sent on this pin
VCC	ISP Power	Power Supply for the ISP. ISP Header must supply power to the dongle.
GND	Ground	Common Ground



10 Pin Header

Dodatek 2 Tablica kodów ASCII

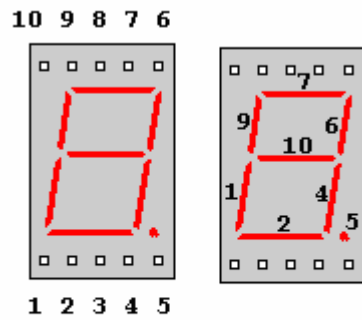
Decimal	Hex	Binary	Value
-----	----	-----	-----
000	000	00000000	NUL (Null char.)
001	001	00000001	SOH (Start of Header)
002	002	00000010	STX (Start of Text)
003	003	00000011	ETX (End of Text)
004	004	00000100	EOT (End of Transmission)
005	005	00000101	ENQ (Enquiry)
006	006	00000110	ACK (Acknowledgment)
007	007	00000111	BEL (Bell)
008	008	00001000	BS (Backspace)
009	009	00001001	HT (Horizontal Tab)
010	00A	00001010	LF (Line Feed)
011	00B	00001011	VT (Vertical Tab)
012	00C	00001100	FF (Form Feed)
013	00D	00001101	CR (Carriage Return)
014	00E	00001110	SO (Shift Out)
015	00F	00001111	SI (Shift In)
016	010	00010000	DLE (Data Link Escape)
017	011	00010001	DC1 (XON) (Device Control 1)
018	012	00010010	DC2 (Device Control 2)
019	013	00010011	DC3 (XOFF) (Device Control 3)
020	014	00010100	DC4 (Device Control 4)
021	015	00010101	NAK (Negative Acknowledgement)
022	016	00010110	SYN (Synchronous Idle)
023	017	00010111	ETB (End of Trans. Block)
024	018	00011000	CAN (Cancel)
025	019	00011001	EM (End of Medium)
026	01A	00011010	SUB (Substitute)
027	01B	00011011	ESC (Escape)
028	01C	00011100	FS (File Separator)
029	01D	00011101	GS (Group Separator)
030	01E	00011110	RS (Request to Send)(Record Separator)
031	01F	00011111	US (Unit Separator)
032	020	00100000	SP (Space)
033	021	00100001	! (exclamation mark)
034	022	00100010	" (double quote)
035	023	00100011	# (number sign)
036	024	00100100	\$ (dollar sign)
037	025	00100101	% (percent)
038	026	00100110	& (ampersand)
039	027	00100111	' (single quote)
040	028	00101000	((left/opening parenthesis)
041	029	00101001	) (right/closing parenthesis)
042	02A	00101010	* (asterisk)
043	02B	00101011	+ (plus)
044	02C	00101100	, (comma)
045	02D	00101101	- (minus or dash)
046	02E	00101110	. (dot)
047	02F	00101111	/ (forward slash)
048	030	00110000	0
049	031	00110001	1
050	032	00110010	2

Decimal -----	Hex ---	Binary -----	Value -----
051	033	00110011	3
052	034	00110100	4
053	035	00110101	5
054	036	00110110	6
055	037	00110111	7
056	038	00111000	8
057	039	00111001	9
058	03A	00111010	:
059	03B	00111011	;
060	03C	00111100	<
061	03D	00111101	=
062	03E	00111110	>
063	03F	00111111	?
064	040	01000000	@
065	041	01000001	A
066	042	01000010	B
067	043	01000011	C
068	044	01000100	D
069	045	01000101	E
070	046	01000110	F
071	047	01000111	G
072	048	01001000	H
073	049	01001001	I
074	04A	01001010	J
075	04B	01001011	K
076	04C	01001100	L
077	04D	01001101	M
078	04E	01001110	N
079	04F	01001111	O
080	050	01010000	P
081	051	01010001	Q
082	052	01010010	R
083	053	01010011	S
084	054	01010100	T
085	055	01010101	U
086	056	01010110	V
087	057	01010111	W
088	058	01011000	X
089	059	01011001	Y
090	05A	01011010	Z
091	05B	01011011	[
092	05C	01011100	\
093	05D	01011101	]
094	05E	01011110	^
095	05F	01011111	_
096	060	01100000	
097	061	01100001	a
098	062	01100010	b
099	063	01100011	c
100	064	01100100	d
101	065	01100101	e
102	066	01100110	f
103	067	01100111	g
104	068	01101000	h

Decimal -----	Hex ----	Binary -----	Value -----
105	069	01101001	i
106	06A	01101010	j
107	06B	01101011	k
108	06C	01101100	l
109	06D	01101101	m
110	06E	01101110	n
111	06F	01101111	o
112	070	01110000	p
113	071	01110001	q
114	072	01110010	r
115	073	01110011	s
116	074	01110100	t
117	075	01110101	u
118	076	01110110	v
119	077	01110111	w
120	078	01111000	x
121	079	01111001	y
122	07A	01111010	z
123	07B	01111011	{ (left/opening brace)
124	07C	01111100	(vertical bar)
125	07D	01111101	} (right/closing brace)
126	07E	01111110	~ (tilde)
127	07F	01111111	DEL (delete)

Dodatek 3 Wyprowadzenia wyświetlacza 7-segmentowego

Wyprowadzenia wyświetlacza 7-segmentowego LED dla typu SL119 lub OS516HWA



VCC to pin 3 or 8

Dodatek 4 Schemat płytki EDB

--> PCB LAYOUT LANDSCAPE 170%