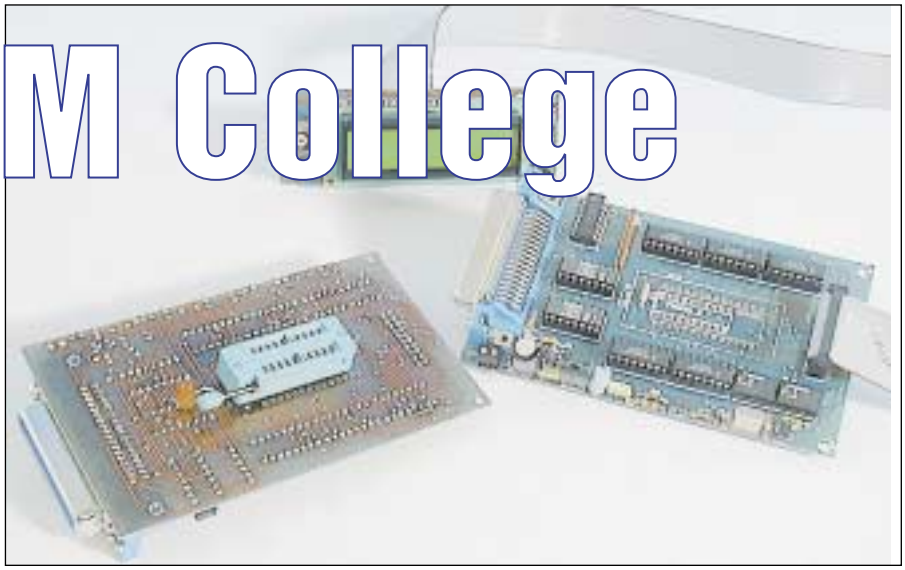
BASCOM College



Wykład 5

Serdecznie witam Studentów i Sympatyków BASCOM College na kolejnym wykładzie, który zmuszony jestem rozpocząć od przeprosin. Jak zapowiadałem, wykład ten miał być poświęcony tematowi obsługi magistrali I^2C, na który z pewnością wszyscy czekacie z niecierpliwością. Niestety, z rozmaitych powodów zmuszony jestem przełożyć ten temat na następny miesiąc, a dzisiaj zajmiemy się kilkoma innymi, równie ważnymi sprawami.

Temat główny naszego wykładu będzie nieco nietypowy. Prawdę mówiąc, lekcja ta jest bardziej rozszerzonym działem "Z ostatniej chwili", niż typowym wykładem. Mamy bowiem bardzo ciekawe sprawy do omówienia, które pojawiły się w "rozkładzie zajęć" dopiero w dniu dzisiejszym. Zanim jednak przejdziemy do omawiania dzisiejszego tematu, jak zwykle kilka spraw porządkowych, głównie związanych z nadchodzącą od Was korespondencją. Otrzymałem sporą ilość listów, głównie zawierających pytania i prośby o dodatkowe informacje i rozwiązanie dręczących Was problemów. Chciałbym jeszcze raz podkreślić, że KAŻDY list e-mail, który otrzymuję, jest przeze mnie czytany i analizowany. Jednak niejednokrotnie nie jestem w stanie na wszystkie listy odpowiedzieć, ponieważ prawdopodobnie nie mógłbym wówczas robić już nic innego. Mam do Was jeszcze jedną prośbę: starajcie się precyzyjnie formułować Wasze pytania. Podam Wam przykład dwóch listów, oczywiście bez wymieniania nazwisk Autorów:

"Nie mogę uruchomić klawiatury szesnastkowej KIT 2503. Nie działają SDA i SCL.

Gdzie jest "pies pogrzebany" ?"

"Szanowny Panie. Emulator, płytka testowa, programator zmontowane z zestawów. Symulacja sprzętowa nie działa. Emulator testowany z wyswietl. 7-segm. I programem testującym z 3/00 EdW. Bez efektu. Symulacja programowa (na monitorze) funkcjonuje poprawnie. Próba z płytką testuj. i klawiaturą szesnastk. także nieudana. PCF typu 8574P (?). Wymieniłem PCF-y z klawiatury do emulatora. Test j.w. Bez efektu. Całym czasem wyświetlacz 7-segm. dołączony do emulatora minimalnie świeci (widzialne praktycznie tylko w ciemności). Port drukarki sprawny - drukarka działa normalnie. Wszystko wskazuje na brak komunikacji z komputerem. Program instalowany dwukrotnie - to samo. Skonfigurowany zgodnie z 4/00 EdW. Programator jeszcze nie testowany. Proszę o pomoc. Dziękuję." /pisownia i składnia jak w oryginale listów/

Moi Drodzy, co ja mam z takimi fantami robić? Nie jestem jasnowidzem i na podstawie takich dostarczonych mi danych mogę jedynie stwierdzić, że obaj Koledzy popełnili jakieś błędy w montażu układów albo w konfiguracji oprogramowania. Podobnych listów, nie zawierających żadnych konkretnych informacji otrzymałem więcej. Pamiętajcie, że chyba najważniejszą cechą programisty jest umiejętność precyzyjnego wyrażania swoich myśli i dostarczania "do obróbki" WSZYSTKICH potrzebnych danych!

Powiedzcie mi także, co mam zrobić z nadsyłanymi mi, długimi nieraz na kilka stron listingami, w dodatku nie opatrzonymi komentarzami? Otrzymałem kilka takich "wypracowań" i nawet ... ich nie przejrzałem, ponieważ przeanalizowanie długiego listingu, napisanego przez kogoś, wymaga znacznie więcej pracy, niż samodzielne napisanie całego programu od początku.

A zatem, reasumując: bardzo Was proszę: piszcie operując konkretami, a nie enigmatycznymi sformułowaniami typu "nie działa" i przysyłajcie tylko "podejrzane" fragmenty listingów.

Nowa wersja BASCOM 8051 DEMO "Special Edition for Elektronika Praktyczna"

Zaledwie tydzień temu ujrzała światło dzienne nowa edycja profesjonalnej wersji BASCOMa 8051, a już mamy, przeznaczoną specjalnie dla Czytelników EP i EdW, wersję DEMO! Podobnie jak poprzednia edycja BASCOM8051 SEfEP, z której korzystaliście do tej pory, wersja ta nie różni się niczym od komercyjnego BASCOM-a. Jedynym ograniczeniem jest długość kodu wynikowego, wynosząca 2kB. A zatem, w przypadku stosowania procesorów '2051 nie istnieją w tym programie żadne ograniczenia!

Najnowsza wersja naszego oprogramowania dostępna jest już pod adresem:

www.ep.com.pl/ftp

Jest to jedyne miejsce, z którego można ściągnąć ten program. Powszechnie dostępna wersja Demo ma ograniczenie kodu wynikowego do 1kB, co uniemożliwia pełne wykorzystanie obszaru pamięci ROM procesora '2051.

Drugą nowością, jaką otrzymaliśmy od MCS Electronics, jest program do automatycznego upgrade'owania BASCOM-a 8051 i AVR. Niestety, z oczywistych względów ten bardzo wygodny w użyciu (**rysunek 1**) program obsługuje jedynie wersję komercyjną BASCOM'a. Można go ściągnąć spod następującego adresu:

Rys. 1

http://www.mcselec.com/download/bascom-8051/autoupd.zip

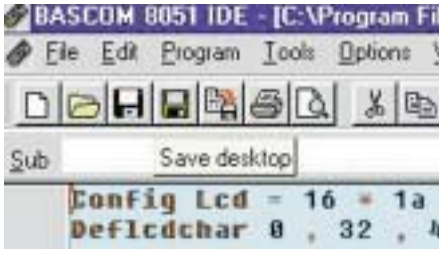
Omówmy teraz w największym skrócie nowości, jakie stworzył dla nas pan Mark Alberts z MCS Electronics. Pierwsze z nich widoczne jest już po uruchomieniu programu.

Poniżej paska narzędziowego umieszczone zostało nowe okienko, w którym można wyświetlić spis wszystkich etykiet i podprogramów wywoływanych przez przerwania zewnętrzne i pochodzące z timerów (**rysunek 2**). Jest to kolejne ułatwienie, znacznie przyspieszające pisanie długich, skomplikowanych programów. Kliknięcie na jedną z wyświetlonych w okienku etykiet powoduje natychmiastowe przejście do interesującego nas fragmentu programu.

Rys. 2

Na pasku narzędziowym pojawił się nowy przycisk: "SAVE DESKTOP" (**rys. 3**). Po jego naciśnięciu program zapamiętuje aktualnie otwarte okna, tak że po ponownym uruchomieniu BASCOM-a znajdziemy się od razu w takim środowisku, jakie opuściliśmy robiąc sobie przerwę w pracy. Funkcja bardzo użyteczna, szczególnie podczas opracowywania kilku programów naraz i przy kompilacji jednego programu z kilku źródeł.

Rys. 3



W okienku wyświetlania informacji o skompilowanym programie (PROGRAM \ SHOW RESULT) dodany został także nowy przycisk: "COPY". Jego kliknięcie powoduje przeniesienie całej zawartości pliku informacyjnego do schowka i jego późniejszą edycję za pomocą dowolnego edytora tekstowego. Posiadacze drukarek barwnych mogą mieć także powód do zadowolenia: w najnowszej wersji BASCOM-a możliwe jest drukowanie listingów i informacji o pracy kompilatora w kolorze.

Jak dotąd omówiliśmy jedynie nowe dodatkowe "fajerwerki" BASCOM-a, niesłychanie ułatwiające i uprzyjemniające pracę, ale nie niezbędne do napisania poprawnego programu. Myślę więc, że jesteście ciekawi, jakie nowe polecenia zostały dodane do najnowszej wersji naszego programu.

Nowych poleceń i najróżniejszych dodatków do już istniejących komend jest sporo i nie sposób omówić je podczas jednej, nieco skróconej lekcji. Chciałbym więc bardziej szczegółowo zająć się poleceniem, które wprawdzie istniało już w poprzedniej wersji, ale jego sposób działania pozostawiał jeszcze wiele do życzenia. Obecnie wszystkie "pluskwy" zostały usunięte i możemy przystąpić do poznawania polecenia, którego z pewnością nie znajdziecie w żadnym innym języku programowania. Mam tu na myśli bezpośrednią obsługę serwomechanizmów modelarskich.

O serwomechanizmach pisałem już wielokrotnie, pozwoliłem sobie także zaprezentować kilka projektów wykorzystujących te urządzenia, będące znakomitym "przełożeniem" pomiędzy elektroniką i mechaniką. Przypomnijmy więc w największym skrócie, że serwomechanizm sterowany jest impulsami prostokątnymi o czasie trwania typowo od 1 do 2 ms. Częstotliwość powtarzania impulsów nie jest krytyczna i powinna wynosić około 50Hz. W zależności od szerokości impulsów podawanych na wejście, wał serwa ustawia się pod kątem proporcjonalnym do ich długości.

Kąt obrotu wału serwa wynosi typowo 90^{O}, ale praktyka wykazała, że stosując impulsy o niestandardowej długości (od 0,5 do 3 ms.) można go zwiększyć do 180^{O}. Pomimo małych wymiarów moment napędowy serwomechanizmu jest bardzo duży i wynosi w przypadku serwa standardowego aż 3,5kg/cm, a w przypadku serw wyczynowych może być o rząd wielkości większy.

Wszystkie te cechy serwomechanizmów sprawiają, że są one wręcz niezastąpione w konstrukcjach amatorskich. Wszędzie tam, gdzie trzeba poruszyć jakiś przedmiot ze znaczną siłą czy obrócić jakiś element o ściśle określony kąt, samodzielna budowa układu elektromechanicznego traci jakikolwiek sens. Najprostszy przykład: budowa części elektronicznej zamka szyfrowego przy obecnym stanie elektroniki nie przedstawia najmniejszych trudności. Jako element mechaniczny stosowane są zwykle tzw. rygle elektromagnetyczne, które można otworzyć jednym kopnięciem. Natomiast sterowane z procesora serwo może bez najmniejszych problemów poruszać nawet bardzo ciężką i solidną zasuwę.

Sterowanie serwomechanizmami z poziomu języka MCS BASIC

Do wyjść procesora może być dołączonych maksymalnie 16 serwomechanizmów. W naszym przypadku jest to liczba gwarantowana niejako "na zapas", ponieważ stosowane przez nas procesory 'X051 posiadają jedynie 15 aktywnych wyprowadzeń. Jednak kiedy "przesiądziemy się" na większe procesory, ta liczba serwomechanizmów może okazać się przydatna, np. do budowy robota czy skomplikowanego modelu.

Pierwszą czynnością, jaką musimy uczynić, jest poinformowanie kompilatora o liczbie serw, jakich zamierzamy używać, kącie, o jaki mają obracać się wały serw przy zmianie wartości przypisanej każdemu serwomechanizmowi zmiennej, o 1 i do których pinów procesora mają zostać dołączone poszczególne serwomechanizmy. Czynimy to za pomocą ciągu dyrektyw:

CONFIG SERVOS = [liczba zastosowanych serw], **RELOAD** [wartość określająca skok wału serwa], **SERVO1** = [pin portu], **SERVO2** = [pin portu] ....., **SERVO16**=[pin portu]

Realizacja obsługi serwomechanizmów przez typowe procesory rodziny '51 nie jest zadaniem prostym. Powód jest oczywisty: większość tych układów jest po prostu zbyt wolna, aby precyzyjnie odmierzać czas z zakresu 0,5 ... 3 ms. Dlatego też wał serwomechanizmu nigdy nie będzie poruszał się idealnie płynnie, ale zawsze mniejszymi lub większymi (skokami), których liczba, przypadająca na pełny obrót wału napędowego serwa zależna jest właśnie od częstotliwości zegara procesora. Nie ma najmniejszego sensu wdawać się w tym miejscu w skomplikowane obliczenia. Podczas testowania na prośbę Marka poleceń związanych z obsługą serwomechanizmów wykonałem serię prób, z których wynikły następujące wnioski:

Minimalna wartość parametru RELOAD dla procesora '2051 z rezonatorem kwarcowym 11059200 Hz wynosi 30. Z kolei maksymalna wartość zmiennej określającej położenie wału serwa może wynosić 70, a minimalna 15. Wynika z tego, że przy wykorzystywaniu pełnego kąta obrotu wału serwomechanizmu skok będzie wynosił dokładnie 3,2 stopnia. Nie jest to być może największa precyzja, ale nic więcej z typowym kwarcem nie osiągniemy. Przeprowadziłem także próby z rezonatorami kwarcowymi o większej częstotliwości. Przy częstotliwości oscylatora równej 24MHz osiągnąłem, zgodnie z przewidywaniami, ponad dwukrotnie większą precyzję poruszeń wału serwomechanizmu, czyli skok wynoszący około 1,5 stopnia.

Warto teraz sprawdzić działanie obsługi serwomechanizmu w praktyce. Niestety, na naszej płytce testowej AVT-2500 nie przewidziałem specjalnego złącza do dołączenia serwa. Powód tego "zaniedbania" był prosty: kiedy projektowałem tę płytkę, nikomu nawet nie śniło się o wbudowanych w

BASCOM-a poleceniach obsługi serwomechanizmów!

A zatem będziemy musieli dołączyć serwo do istniejących już na płytce wyprowadzeń. Kabelek dołączony do serwa ma trzy przewody: przewód czarny dołączamy do masy, czerwony do plusa zasilania (+5VDC), a żółty do dowolnego wyjścia procesora, z wyjątkiem pinów OPEN COLLECTOR (P1.0 i P1.1), np. do pinu P1.7.

Musimy teraz napisać sobie króciutki programik testowy i po skompilowaniu zaprogramować nim procesor. Niestety, jakiekolwiek próby testowania tego programu na emulatorze sprzętowym skazane są, z oczywistych przyczyn, na niepowodzenie.

```
$crystal = 11059200
Config Servos = 1 , Reload = 30 , Servo1 = P1.7
Dim R As Byte

Do
  For R = 15 To 70
    Servo1 = R
    Waitms 2
  Next R
  For R = 70 Downto 15
    Servo1 = R
    Waitms 2
  Next R
Loop
```

Po uruchomieniu programu wał napędowy serwomechanizmu powinien poruszać się, wykonując obroty o 180^{O}.

Tak napisany program służy jedynie celom poznawczym i nie posiada większych wartości użytkowych. Aby do końca przekonać Studentów BASCOM College o użyteczności serwomechanizmów, podam Wam tylko jeden przykład. Wyobraźmy sobie, że konstruujemy układ zamka elektronicznego, np. otwieranego za pomocą "magicznych" tabletek DALLAS, z którymi zapoznamy się w najbliższej przyszłości. Hardware i program są sprawą prostą, ale jak wykonać solidne zamknięcie do drzwi domu? Przy zastosowaniu serw sterowanych z mikroprocesora zaprogramowanego z wykorzystaniem BASCOM-a sprawa jest banalnie prosta: wał napędowy serwa wykorzystujemy jak mimośród, łącząc go za pomocą sztywnego stalowego drutu z dowolnie ciężką zasuwą. Natomiast w programie, w momencie, kiedy chcemy, aby zamek się otworzył, piszemy po prostu:

SERVO1 = 15

Natomiast polecenie:

SERVO1 = 70

spowoduje obrót wału serwa o 180^{O} i zamkniecie zamka. Przebiegi na wyjściu procesora, do którego dołączone powinno być wejście serwomechanizmu, zostały pokazane na **rysunku 4**. Proste i użyteczne, nieprawdaż?

Rys. 4

Polecenia związane z obsługą serwomechanizmów nie są jedynymi dodanymi do nowej wersji BASCOM-a. Jak dotąd nie omawialiśmy jeszcze bardzo użytecznego polecenia SOUND, głównie dlatego, że napotkałem na problemy z jego wykonywaniem w programach wykorzystujących przerwania. Generowany dźwięk był zniekształcony, przerywany i wolałem zaczekać na rozwiązanie przez Marka tego problemu. Obecnie kłopoty z generowaniem sygnałów akustycznych zostały usunięte i możemy już zapoznać się z poleceniem:

SOUND pin, czas trwania, częstotliwość [,NOINT]

Zanim jednak przejdziemy do szczegółowego opisu polecenia SOUND, musimy wyjaśnić sobie pewną sprawę. Polecenie to w żadnym wypadku nie może służyć do generowania sygnałów o precyzyjnie określonej częstotliwości, czy też, tym bardziej, do tworzenia "kompozycji muzycznych". Jego przeznaczeniem jest tworzenie "some noise", jak napisał Mark w helpie, czyli generowania prostych sygnałów akustycznych, o z grubsza określonej częstotliwości i czasie trwania.

Wydając polecenie SOUND musimy określić, do jakiego wyprowadzenia procesora dołączony będzie przetwornik elektroakustyczny, którym może być element piezo lub miniaturowy głośniczek. Następnie określamy liczbę impulsów, z których ma się składać generowany dźwięk. Liczbę impulsów, czyli wartość wpływająca na czas trwania dźwięku, może być liczbą z zakresu, od 1 (trochę bez sensu) do 326768. Ostatnim parametrem potrzebnym do wygenerowania dźwięku jest jego, z grubsza określona częstotliwość, czyli informacja o tym, co ile mikrosekund stan na wyjściu wybranego pinu ma zmieniać się na przeciwny.

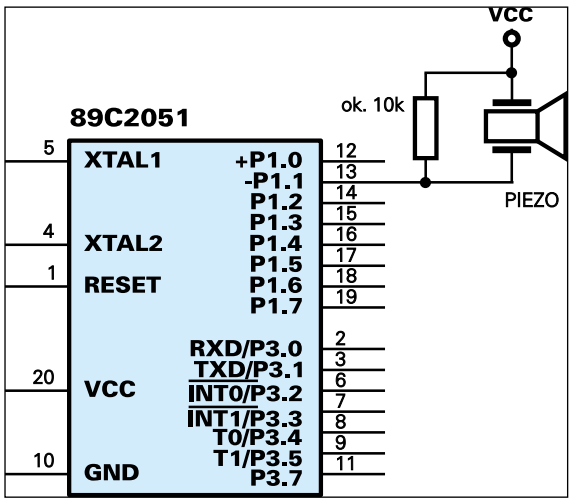
Parametrem opcjonalnym jest nowo dodane polecenie NOINT. Dołączenie go do polecenia SOUND spowoduje, że obsługa przerwań zostanie zawieszona na czas generowania sygnału akustycznego. Parametr ten musimy stosować z największą rozwagą, po dokładnej analizie celu stosowania przerwań w pisanym programie. Jest oczywiste, że jeżeli przerwania stosowane są np. do generacji impulsów sekundowych w zegarze czasu rzeczywistego, to o jakimkolwiek zawieszaniu ich obsługi nie może być mowy, pomimo że po zakończeniu generacji dźwięku obsługa przerwań jest natychmiast przywracana.

Przykładowo, polecenie SOUND może mieć postać:

SOUND P1.1 , 10000, 10 [NOINT]

Należy pamiętać, że po zakończeniu generacji sygnału akustycznego wyjście użytego do tego celu portu **pozostaje w stanie niskim**!

Jako przetwornika elektroakustycznego można użyć "blaszki" piezo, najlepiej w obudowie wzmacniającej siłę generowanego dźwięku. Na **rysunku 5** pokazany jest prosty schemat takiego układu. Możliwe jest też zastosowanie miniaturowego głośniczka (dołączonego tylko do pinów portu P1!)



Rys. 5

To wszystko na dzisiaj. Zapraszam Was teraz do przerobienia materiału zawartego w ćwiczeniu, a następnie na otwarcie nowego działu „µProjekty - 3000", który z całą pewnością spotka się z Waszym zainteresowaniem.

Zbigniew Raabe
e-mail: **zbigniew.raabe@edw.com.pl**

BASCOM Forum

Witam Panie Zbigniewie.

Zainspirowany pomysłem kolegi Artura Krajewskiego postanowiłem podzielić się swoim spostrzeżeniem. Sprawa dotyczy - oczywiście wyświetlacza LCD.

```
'Programik ujawniający błąd
'funkcji LOCATE
'Na P3.5 zapięta LED -sygnalizacja pracy
$sim
$noinit
Dim A As Byte
Dim X As Byte
Dim Korekta As Byte
Config Lcd = 16 * 1a
Cursor Off
Cls
For X = 1 To 80
Set P3.5
Cls
Locate 1 , X
Lcd X
For A = 1 To 30
Next A
Reset P3.5
For A = 1 To 30
Next A
Next X
```

```
' Program wyświetlający na LCD 1*16
' numer pozycji znaku.
' zmienna KOREKTA - dodanie offsetu o 8 pozycji
'UWAGA - funkcja LOCATE na pozycji Y
'dla Y =1 Locate działa do 9 znaku
'dla Y=2 Locate działa od 9 do 16 znaku
$sim
$noinit
Dim A As Byte
Dim X As Byte
Dim Korekta As Byte
Config Lcd = 16 * 1a
Cursor Off
Cls
For X = 1 To 8
Cls
Locate 1 , X
Lcd X
For A = 1 To 30
Next A
Next X
For X = 1 To 8
Cls
Korekta = X + 8
Locate 2 , X
Lcd Korekta
For A = 1 To 30
Next A
Next X
```

Sprawa ujawniła się, kiedy chciałem zapisać funkcją LOCATE zmienną na pozycji 10. Nic się nie pojawiło (źródło niżej):

"Bawiąc się" parametrami LOCATE, wydaje mi się, że znalazłem na to "łatę" podobną do "łaty" kol. Krajewskiego.

Krótko mówiąc, pierwsze 8 znaków wystawiam przez LOCATE 1, X, gdzie x od 1 do 8; drugie 8, czyli znaki od 9 do 16 pozycji, wystawiam przez LOCATE 2, X gdzie x od 1 do 8 (w programiku Korekta czyli X zwiększona o 8 powoduje, że na LCD pojawia się faktyczna pozycja znaku - cyfra pierwsza z lewej dla numeracji od 10 do 16). Działanie "łaty" przedstawia programik obok.

P.S.

*1. Może pan Mark powinien napisać osobne funkcje dla wyświetlaczy 1*16.*

2. Coraz bardziej widzę konieczność powstania listy dyskusyjnej, jednak nie chcę żeby Koledzy nie mający dostępu do Internetu czuli się dyskryminowani.

pozostaję z szacunkiem

Krzysztof Pokorski

krzysztof_pokorski@elb2.pl

Wejherowo, 9.07.2000

Szanowny Panie Zbigniewie!

Piszę do Pana ten list z dwóch powodów. Po pierwsze, tak jak kiedyś obiecałem, przesyłam program sterujący pojazdem napędzanym dwoma silnikami krokowymi (coś na wzór AVT-2059) W obecnej wersji tor ruchu pojazdu nie jest zbyt skomplikowany - odcinek do przodu (tyłu), obrót w prawo, do przodu, obrót w lewo, do przodu, zatrzymanie się i przejście do stanu uśpienia. Program umożliwia (za pomocą przycisków dołączonych do wejść INT0 i INT1) wybór kierunku ruchu, w którym pojazd zaczyna się przemieszczać na samym początku. Schemat połączeń jest bardzo prosty. Port P1 steruje silnikami krokowymi poprzez driver ULN2803. Wyjście P3.7 jest połączone z wejściami liczników T0 i T1. Wejścia INT0 oraz INT1 mogą być zwierane poprzez przyciski do masy. Linie P1.0 - P1.3 prowadzą do nóżek 12, 11, 10, 9 ULN2803, a linie P1.4 - P1.7 do pinów 7, 6,5, 4 (w tej kolejności). Do wyjść 15 - 18 drivera podpiąłem lewy silnik, a 11 - 14 - prawy.

Trochę o programie. Początki sięgają maja 2000, gdy napisałem program sterujący pracą silnika krokowego jako zadanie domowe z Bascomu. Właściwe prace rozpoczęły się w czerwcu, gdy poznałem obsługę przerwań i timerów. Mam zamiar dalej rozwijać program, lecz wykorzystując inną platformę pojazdu i napęd. Silniki krokowe okazały się niedokładne (choć pochodzą z tego samego modelu stacji dysków, to pracują nierówno), a ich mała prędkość obrotowa nie pozwala na stosowanie jakiegoś sprzężenia zwrotnego umożliwiającego korektę obrotów. Postanowiłem stworzyć coś na wzór robota kolegi Marka Lewandowskiego. Jego artykuły w EdW bardzo sobie cenię.

Druga sprawa to pomysł stworzenia plotera z EdW 7/00. Na taki sam pomysł wpadłem na początku roku, z tym, że miał być sterowany przez port równoległy i służyć bardziej do rysowania grafiki, a nie płytek drukowanych. Jeśli chodzi o opisywany przez Pana ploter, to zauważyłem dwa problemy. Primo: konwersją plików HPGL na postać przyswajalną dla sterownika plotera powinni się zająć programiści znający dobrze języki programowania na Peceta, mniej Bascom (może czytelnicy EdW lub EP?). Secundo: jak długo czytam Pańskie artykuły, tak pamiętam, że zawsze był Pan stanowczym przeciwnikiem wszelkich prac mechanicznych, a z tego, co mi się wydaje, to wykonanie w warunkach amatorskich takiej konstrukcji, która by zapewniała wystarczającą dokładność do frezowania płytki drukowanej nie jest wcale takie proste.

Z poważaniem

Adam Robaczewski

badworm@poczta.fm

```
'_____________________________________
'       (r)  20-06-2000 Bad Worm
'_____________________________________
' File : robot.bas
'            program sterujący pojazdem
'       napędzanym silnikami krokowymi
'                                  4-fazowymi
'_____________________________________
'$sim
Dim Krok(16) As Byte          'deklaracja tablicy
Dim X As Byte                 'deklaracja zmiennej
                              pomocniczej X (opóźnienie)
Dim R As Byte                 'deklaracja zmiennej
                              pomocniczej R (główny licznik)
Dim A As Byte                 'deklaracja zmiennej
                              warunku ruchu Forward
Dim C As Byte                 'deklaracja zmiennej
                              dolnej granicy kroków
Dim D As Byte                 'deklaracja zmiennej
                              górnej granicy kroków
Dim E As Byte                 'deklaracja zmiennej
                              warunku obrotu Wprawo
Dim F As Byte                 'deklaracja zmiennej
                              warunku wyłączenia Power
Config Timer0 = Counter , Gate = Internal , Mode
= 2       'konfiguracja timera0
Config Timer1 = Counter , Gate = Internal , Mode
= 2       'konfiguracja timera1
                              'ładowanie kroków do tablicy
Krok (1) = 129                'ładowanie do tablicy
                              kodu kroku 1
Krok (2) = 66                 'ładowanie do tablicy
                              kodu kroku 2
Krok (3) = 36                 'ładowanie do tablicy
                              kodu kroku 3
```

Cały listing programu Adama oraz wszystkie ciekawe propozycje przychodzące do Redakcji znajdziecie na stronie internetowej EdW.