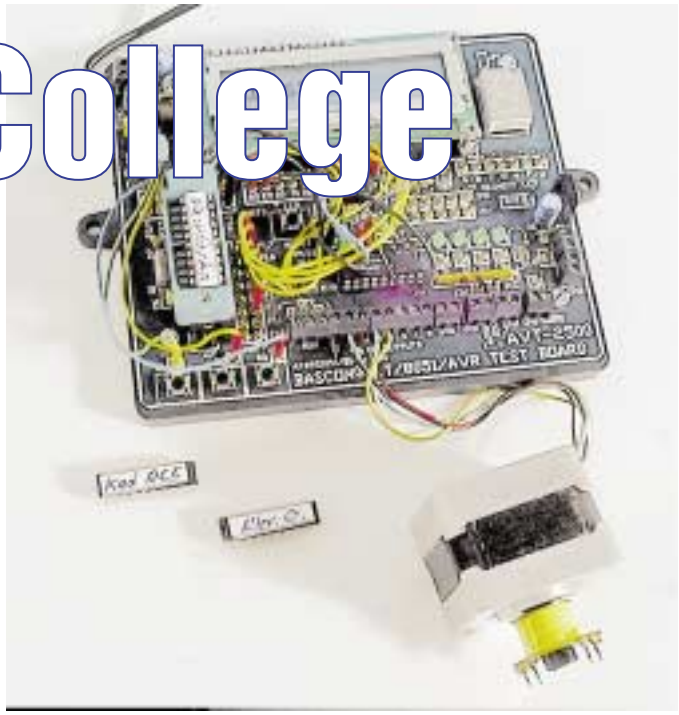
BASCOM College



Ćwiczenie 4

Nawijarka cewek,
czyli napędy silników krokowych

W tym artykule chciałbym pokazać Wam, jak posiadając tylko płytkę testową AVT-2500, procesor ’2051 i kilka drobnych elementów możemy rozwiązać niejednokrotnie skomplikowane problemy przygotować sobie potrzebne narzędzia.

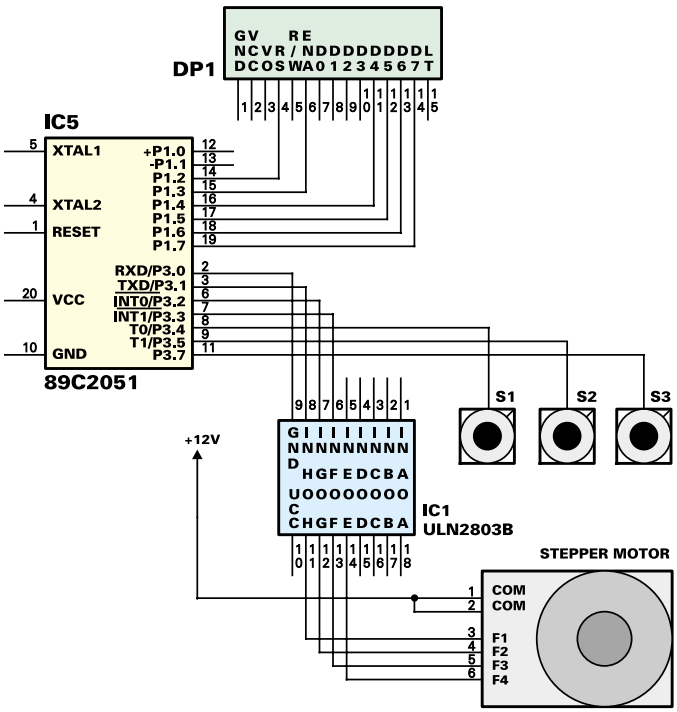
Chciałbym podać Wam pierwszy, ale nie ostatni przykład rozwiązywania za pomocą naszej płytki i zaprogramowanego procesora problemów mechanicznych, które zwykle sprawiają nam najwięcej kłopotów. Nawijanie cewek, czy może być coś bardziej okropnego, szczególnie jeżeli mamy do wykonania kilka sztuk tych powszechnie nie lubianych przez elektroników elementów? Pokażę Wam zatem, jak nawet tę okropną czynność możemy zautomatyzować i wykonać prostą nawijarkę do cewek.

Celowo piszę: "nawijarka do cewek", a nie "nawijarka transformatorów", ponieważ wykonane, a właściwie zaprogramowane przeze mnie urządzenie nie bardzo nadaje się do nawijania cięższych lub wykonywanych grubszym drutem cewek. Powód jest prosty: stosunkowo mała moc silnika krokowego. Zadanie było dość proste: napisać program wprawiający w ruch silnik krokowy, a następnie uzupełnić go o możliwość programowania liczby obrotów, jaką silnik ma jednorazowo wykonać. Mechanika jest zupełnie banalna: cewkę o niewielkich rozmiarach można osadzić "na wcisk" na wale silnika, w razie potrzeby zwiększając jego średnicę za pomocą odpowiedniej liczby zwojów taśmy izolacyjnej. Z układu prowadzącego nawijany przewód na razie zrezygnowałem, pomimo że jego wykonanie byłoby dość proste. Przy odrobinie wprawy można nawijany przewód trzymać po prostu w palcach.

Temat silników krokowych był już przeze mnie poruszany wielokrotnie i mam nadzieję, że wszyscy Czytelnicy pamiętają podstawowe zasady posługiwania się tymi elementami. Do naszej konstrukcji wykorzystamy typowy silnik krokowy, najczęściej pochodzący z demontażu przestarzałych stacji dysków 5". **Musi to być silnik czterofazowy (posiadający cztery niezależne cewki). Zastosowanie silnika dwufazowego, wymagającego odwracania kierunku przepływu prądu w cewkach w przypadku współpracy z naszą płytką testową nie jest możliwe.**

Montujemy układ zgodnie z **rysunkiem 1**, na którym pokazano tylko połączenia przewodowe, które musimy wykonać na płytce testowej (możemy także wzorować się na fotografii przedstawiającej naszą nawijarkę) i bierzemy się za pisanie programu. Z zasady działania silnika krokowego wynika, że aby wprawić go w ruch, musimy cyklicznie włączać zasilanie kolejnych jego cewek. Wynika z tego, że na wyjściach P3.0, P3.1, P3.2 i P3.3 procesora, do których dołączone

Rys. 1



są drivery mocy zasilające cewki silnika, musi kolejno pojawiać się stan wysoki. Przerwa pomiędzy kolejnymi zmianami stanów decyduje o prędkości obrotowej silnika. Przerwa ta nie może być zbyt krótka, ponieważ silnik krokowy ma ściśle określoną maksymalną prędkość obrotową, powyżej której silnik raptownie traci moc, a wreszcie się zatrzymuje. Nie ma natomiast jakichkolwiek ograniczeń "w dół" - silnik krokowy, w przeciwieństwie do silników DC, może obracać się dowolnie wolno.

Napiszmy zatem podprogram powodujący obrót wału silnika o cztery kroki, czyli jeden cykl:

```
Sub Foursteps
Set P3.0 : Reset P3.1 : Reset P3.2 : Reset P3.3
Call Ddelay
Reset P3.0 : Set P3.1 : Reset P3.2 : Reset P3.3
Call Ddelay
Reset P3.0 : Reset P3.1 : Set P3.2 : Reset P3.3
Call Ddelay
Reset P3.0 : Reset P3.1 : Reset P3.2 : Set P3.3
Call Ddelay
End Sub
```

Każdorazowe wywołanie tego podprogramu powodować będzie wykonanie przez silnik czterech kroków, w czasie zależnym od opóźnienia wnoszonego przez podprogram DDELAY. Takie rozwiązanie ogranicza nieco precyzję sterowania silnikiem, ale w przypadku nawijarki, którą mamy zbudować, nie ma to znaczenia. Pamiętajmy jednak, że silnikiem krokowym możemy w prosty sposób sterować z precyzją do połowy kroku. Jak? No to popatrzcie:

```
Set P3.0 : Reset P3.1 : Reset P3.2 : Reset P3.3
                    'prąd płynie tylko przez
pierwszą cewkę
Call Ddelay
Set P3.0 : Set P3.1 : Reset P3.2 : Reset P3.3
                    'prąd płynie przez cewkę 1
i 2, wał silnika zatrzymał 'się w środkowym
położeniu, czyli wykonał pół kroku
Call Ddelay
Reset P3.0 : Set P3.1 : Reset P3.2 : Reset P3.3
                    ' dokończenie kroku
Call Ddelay
```

Zostawmy jednak te dygresje i wracajmy do naszej nawijarki. Kolejnym podprogramem, który musimy napisać, jest realizacja opóźnienia pomiędzy kolejnymi krokami:

```
Sub Ddelay
Waitms X        '10 lub inna wartość, patrz tekst
dalej
End sub
```

Jak już wspomniałem, dla silnika krokowego istnieje ściśle określona wartość maksymalnej prędkości obrotowej, natomiast nie ma ograniczeń co do prędkości minimalnej. Zmienna X decyduje właśnie o tej prędkości i w przypadku nawijarki powinna być dobrana tak, aby silnik obracał się możliwie szybko. W przypadku silnika, którego używałem podczas prób wartość X wyniosła 10, czyli czas przerwy pomiędzy kolejnymi krokami wynosił 0,01 sekundy. Sądzę, że taka sama wartość X będzie odpowiednia dla większości typowych silników krokowych, zastosowanych w Waszych konstrukcjach, jednak najlepiej ustalić tę wartość doświadczalnie.

No dobrze, wiemy już jak obrócić wał silnika o cztery kroki, ale w nawijarce będziemy raczej liczyć pełne obroty wału silnika. I znowu pojawia się problem, który rozwiązać możemy jedynie drogą empiryczną. Większość, ale tylko większość silników od stacji dysków, z którymi miałem do czynienia wykonywała sto kroków na jeden pełny obrót wału napędowego. A zatem, w celu wykonania pełnego obrotu przymocowanej do wału napędowego cewki, musimy podprogram FOURSTEPS powtórzyć 25 razy. Jednak i to musicie sprawdzić doświadczalnie, ponieważ spotykałem się już z silnikami o 90 i 120 krokach na obrót.

A zatem, wykonanie kolejnego podprogramu:

```
For R = 1 To Rotations          ' z m i e n n a
ROTATIONS określa liczbę obrotów silnika,
które chcemy wykonać
 For X = 1 To 4Steps            ' z m i e n n a
4STEPS określa liczbę "poczwórnych", kroków
potrzebnych do      'obrócenia wału silnika o
360 stopni
                    Call Foursteps
                  Next X
 Next R
```

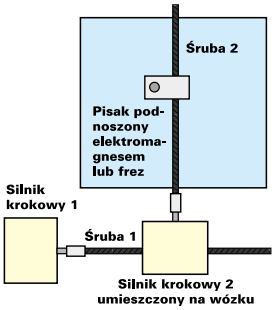
spowoduje wykonanie określonej zmienną ROTATIONS liczby obrotów silnika, czyli nawinięcie takiej samej liczby zwojów na naszej cewce.

I na tym moglibyśmy już właściwie zakończyć pracę. Posklejać te wszystkie podprogramy razem, zadeklarować zmienne i podprogramy i zamiast zmiennej ROTATIONS wstawić stałą, określającą liczbę zwojów naszej cewki. Tak też postąpimy, jeżeli mamy do wykonania tylko kilka identycznych cewek i nie zamierzamy korzystać z nawijarki w przyszłości. Jeżeli jednak mamy zamiar nawijać wiele różnych cewek o rozmaitej liczbie zwojów, to warto jeszcze trochę popracować i napisać program, który umożliwi programowe wprowadzanie liczby zwojów i obsługuje wyświetlacz LCD, na którym prezentowane są aktualnie wprowadzone wartości a następnie postępy w pracy. Napisałem taki programik i przetestowałem go na naszej płytce. Jego listing zamieszczony jest poniżej, ale celowo nie będziemy go komentować. Jest to Wasza praca domowa: przeanalizujcie ten program, wypróbujcie go i postarajcie się znaleźć inne jeszcze zastosowania dla silników krokowych sterowanych z naszej płytki testowej.

```
'Nawijarka do cewek
'$sim
Config Lcd = 16 * 1a
Cursor Off
Declare Sub Mainprogram
Declare Sub Foursteps
Declare Sub Ddelay
Dim R As Word
Dim 4Steps As Byte
Dim Rotations As Word
Dim X As Byte
S1 Alias P3.4
S2 Alias P3.5
S3 Alias P3.7
Rotations = 100
Steps = 25
Reset P3.0 : Reset P3.1 : Reset P3.2 : Reset P3.3

Sub Mainprogram
Cls
Lcd "Turns: " ; Rotations
  Do
                    Debounce S1 , 0 , Decrturns ,
Sub
                    Debounce S2 , 0 , Incrturns ,
Sub
                    Debounce S3 , 0 , Go_on , Sub
  Loop
End Sub

Decrturns:
Decr Rotations
                    If Rotations = 0 Then
                    Rotations = 1
                    End If
Locate 1 , 8
Lcd "    "
Locate 1 , 8
Lcd Rotations
Waitms 50
Return

Incrturns:
Incr Rotations
                    If Rotations = 65536 Then
                    Rotations = 65535
                    End If
Locate 1 , 8
Lcd "    "
Locate 1 , 8
Lcd Rotations
Waitms 50
Return

Go_on:
Cls
Lcd "Waiting..."
Wait 3
Cls
Lcd "Working..."
  For R = 1 To Rotations
    For X = 1 To 4Steps
      Call Foursteps
    Next X
      Locate 2 , 4
      Lcd R
  Next R
Reset P3.0 : Reset P3.1 : Reset P3.2 : Reset P3.3
Cls
Call Mainprogram
Return

Sub Foursteps
Set P3.0 : Reset P3.1 : Reset P3.2 : Reset P3.3
Call Ddelay
Reset P3.0 : Set P3.1 : Reset P3.2 : Reset P3.3
Call Ddelay
Reset P3.0 : Reset P3.1 : Set P3.2 : Reset P3.3
Call Ddelay
Reset P3.0 : Reset P3.1 : Reset P3.2 : Set P3.3
Call Ddelay
End Sub

Sub Ddelay
Waitms 10
End Sub
```

Chciałbym nawet podsunąć Wam parę pomysłów, a właściwie tematów do przemyślenia. Wróćmy jeszcze do naszej nawijarki, która jest urządzeniem bardzo prostym i nie posiadającym możliwości automatycznego prowadzenia nawijanego przewodu. A może by tak dodać do niej drugi silnik krokowy, na którego wale zamocowana by była zwyczajna śruba, M5 lub M6, jakakolwiek, byle można by było łatwo połączyć ją z wałem za pomocą kawałka rurki. Ten drugi silnik musiałby obracać się synchronicznie z pierwszym lub nieco wolniej, lub szybciej, w zależności od średnicy nawijanego drutu, zmieniając po nawinięciu każdej warstwy kierunek obrotów. Gwint śruby mógłby służyć jako prowadnica przewodu i w ten sposób otrzymalibyśmy całkowicie zautomatyzowaną nawijarkę!

Pójdźmy dalej: wyobraźmy sobie urządzenie, którego szkic (szkic, a nie rysunek techniczny!) został pokazany na **rysunku 2**. Dwa silniki krokowe z nasadzonymi na ich wałach śrubami. Pierwszy silnik zamocowany jest do podstawy, a drugi umieszczony został przesuwnie, na wózku lub jakichś szynach... Śruba, a właściwie długi nagwintowany pręt (do nabycia w każdym sklepie ze śrubami i artykułami budowlanymi), zamocowana do wału drugiego silnika przesuwa jakiś element, do którego przymocowany jest podnoszony i opuszczany za pomocą elektromagnesu pisak wodoodporny. A może nie pisak, tylko mały silniczek z miniaturowym frezem dentystycznym ...?



Rys. 2

Chyba już wiecie, do czego zmierzam: wykorzystując dwa silniki krokowe i garść elementów mechanicznych moglibyśmy zbudować ploter, który mógłby rysować ścieżki bezpośrednio na laminacie albo nawet usuwać zbędny laminat za pomocą frezu. Niczego nowego nie wymyśliłem, takie urządzenia są od dawna produkowane i umożliwiają wykonanie prototypowej płytki obwodu drukowanego w ciągu paru minut, bez konieczności taplania się w chlorku czy innych paskudztwach. Tylko że cena tych urządzeń ... no, lepiej zmieńmy temat.

Wykonanie w amatorskich warunkach frezarki do płytek jest całkowicie możliwe. Prawie każdy program inżynierski, w tym także EASY i AUTOTRAX, potrafią wygenerować kod w formacie HPGL przeznaczony do sterowania ploterem. Problem jest jedynie z "przetłumaczeniem" przez procesor kodu HPGL i odpowiednim wysterowaniu silników plotera. No cóż, jest to temat dla BARDZO ambitnych programistów!

Jednak ci bardzo ambitni mogą natrafić na poważny problem: braku wystarczającej ilości nóżek w małych procesorach 'X051. Dwa silniki krokowe to 8 pinów, zostaje tylko siedem. Wyświetlacz alfanumeryczny to dalsze 6 pinów i już pozostała nam tylko jedna wolna nóżka! W przypadku budowy plotera, frezarki czy też innego urządzenia sterowanego z komputera nie ma to większego znaczenia, bo i tak nie będziemy w nich raczej wykorzystywać wyświetlacza. Ale w innych projektach te 15 aktywnych pinów może być problemem. Pomyślimy nad jakimś rozwiązaniem na potrzeby BASCOM College, a na razie mogę polecić Wam prosty, opracowany przeze mnie układ, opublikowany w numerze 4/00 Elektroniki Praktycznej (AVT-860). Jest to sterownik dwóch silników krokowych, cztero- lub dwufazowych, lub czterech silników prądu stałego średniej mocy. Sterownik ten dołączany jest do procesora za pomocą magistrali I2C, która będzie tematem jednej z następnych lekcji.

Zbigniew Raabe
e-mail: **zbigniew.raabe@edw.com.pl**
Konsultacje: **Sławomir Surowiński**
e-mai: **slawomir.surowinski@edw.com.pl**