

**Dokumentacja techniczna projektu:**

***Sterowanie urządzeniami  
elektronicznymi przy użyciu  
portu LPT***



**Zespół Szkół Elektronicznych w Rzeszowie**

16 kwietnia 2007

**Spis treści:**

<u>Wstęp.....</u>	<u>3</u>
<u>Zrealizowane prace.....</u>	<u>3</u>
<u>Dane techniczne:.....</u>	<u>3</u>
<u>Program „Sterowanie portem LPT”.....</u>	<u>3</u>
<u>Wstęp.....</u>	<u>3</u>
<u>Główne okno programu.....</u>	<u>4</u>
<u>Menu programu.....</u>	<u>5</u>
<u>Programator.....</u>	<u>6</u>
<u>Silnik krokowy - unipolarny.....</u>	<u>9</u>
<u>Sterowanie portem LPT z użyciem Delphi i WinApi.....</u>	<u>10</u>
<u>Sekwencja sterująca silnikiem.....</u>	<u>11</u>
<u>Połączenie z portem LPT.....</u>	<u>11</u>
<u>Sterowanie silnikiem.....</u>	<u>12</u>
<u>Spis ilustracji.....</u>	<u>12</u>
<u>Informacje o autorze.....</u>	<u>13</u>

## **Wstęp**

Projekt „Sterowanie urządzeniami elektronicznymi przy użyciu portu LPT” został zrealizowany w ramach specjalizacji programowanie w Zespole Szkół Elektronicznych w Rzeszowie jako praca końcowa. Podsumowująca wiedzę zdobytą w ciągu specjalizacji oraz własnych doświadczeń.

## **Zrealizowane prace**

W ramach projektu zostały zrealizowane następujące prace:

- Program komputerowy „Sterowanie portem LPT” zrealizowany w środowisku Borland Delphi 5,
- Program „Czytnik LPT” dla mikrokontrolera 8051 napisany w Asemblerze,
- Przewód łączący komputer (LPT) z silnikiem krokowym lub DSM51’,
- Niniejsza dokumentacja techniczna.

## **Dane techniczne:**

Do realizacji projektu zostały użyte:

- Silnik krokowy MSJE200A53 firmy Sankyo, (z starej stacji dyskietek 5,25”),
- Przewód LPT (z nieużywanej drukarki HP DeskJet 640C),
- Komputer klasy PC (WinXp + SP2, 660MHz, 256 MB Ram),
- DSM51,
- Środowisko programistyczne:
  - Borland Delphi 5,
  - Asembler, (dsm51asm.exe).

## **Program „Sterowanie portem LPT”**

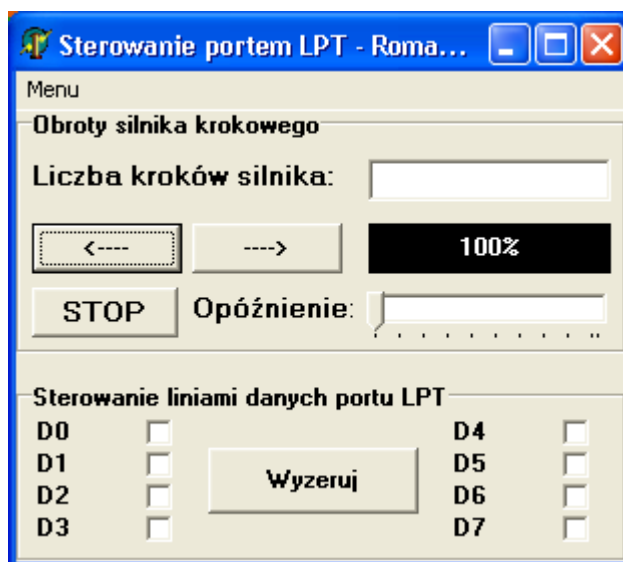
### **Wstęp**

Program „Sterownik LPT” został napisany w celu sterowania urządzeń zewnętrznych. Można nim sterować 8 różnych urządzeń elektronicznych (włącz/wyłącz) lub po zastosowaniu dodatkowych

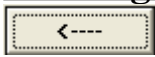
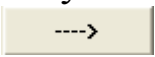
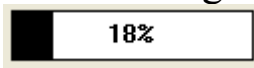
rozszerzeń elektronicznych większej ilości. Za pomocą tego programu możliwe jest również sterowanie silnikiem krokowym, które zostanie opisane w dalszej części tego dokumentu. Program umożliwia również zaprogramowanie pewnej sekwencji ruchów silnika krokowego, co również zostanie opisane później. Do jego realizacji zostało użyte środowisko programistyczne Borland Delphi oraz funkcje WinApi.

### **Główne okno programu**

Główne okno programu umożliwia nam sterowanie silnikiem krokowym, oraz ustawianie poszczególnych linii portu równoległego.



Rysunek 1 - Główne okno programu

**Panel górny** służy do sterowania silnikiem krokowym, przyciski  oraz  odpowiednio służą do startowania obrotów silnika w kierunku przeciwnym do ruchu wskazówek zegara oraz w kierunku zgodnym z ruchem wskazówek zegara. Pasek postępu  odzwierciedla stan obrotów silnika (0% - silnik zaczyna obroty, 100% - silnik skończył obroty). Do wybierania ilości kroków służy pole edycyjne **Liczba kroków silnika:**  (jeśli puste zostanie wykonany jeden pełny obrót – 50 kroków). W trakcie trwania obrotu silnika można go zatrzymać za pomocą przycisku

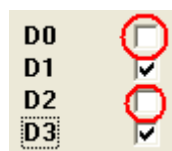
**STOP**. Możliwa jest również regulacja czasu trwania przerwy pomiędzy poszczególnymi krokami, długość opóźnienia ustawiana jest za pomocą suwaka: **Opóźnienie:**

**Panel dolny** służy do ustawiania stanu poszczególnych linii portu LPT. Numer linii odzwierciedlają etykiety od D0 do D7 a ich stan pola (CheckBox). Jeśli pole jest zaznaczone jak na rysunku poniżej to stan linii jest równy „1”.



Rysunek 2 - Pola zaznaczone

Jeśli nie są zaznaczone jak na rysunku 3 to ich stan jest równy „0”.



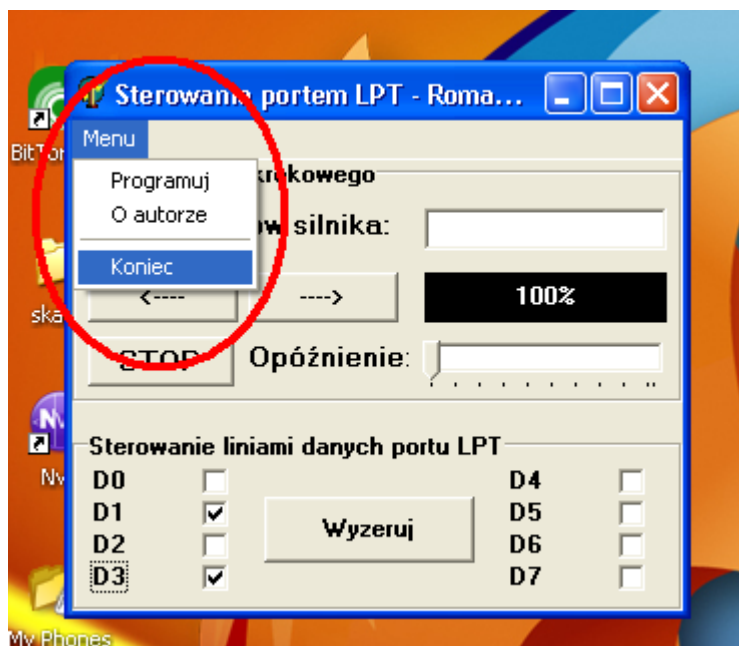
Rysunek 3 - Pola niezaznaczone

Możliwe jest również jednocześnie wyzerowanie wszystkich linii za pomocą przycisku **Wyzeruj** który równocześnie na wszystkie linie podaje sygnał „0”.

### **Menu programu**

Menu programu zostaje wywołane po naciśnięciu na górnym pasku „Menu”, które możesz zobaczyć na rysunku poniżej. Menu to umożliwia wykonać najprostsze funkcje:

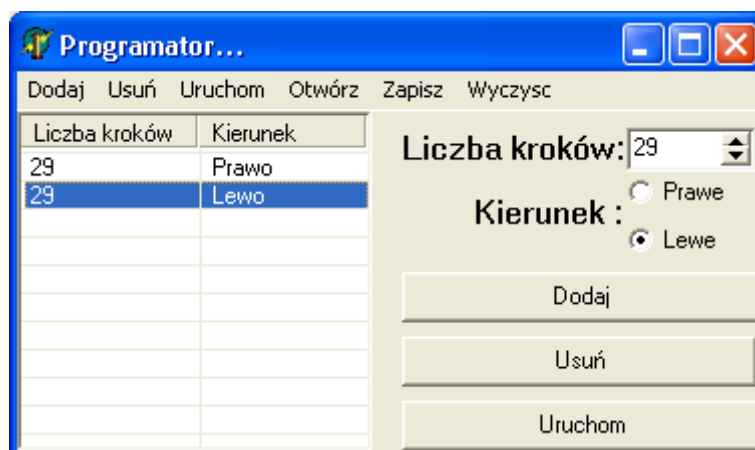
- Zakończyć działanie programu,
- Wyświetlić informacje o autorze,
- Uruchomić okienko programowania ruchów silnika krokowego.



Rysunek 4 - Menu programu

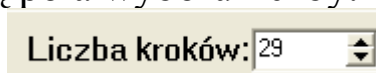
## Programator

Programator służy do programowania kilku następujących sekwencji obrotów silnika krokowego. Aby uruchomić programator kliknij na „Menu” następnie z menu wybierz opcję „Programuj”. Po uruchomieniu pojawi się okienko jak na poniższym rysunku.



Rysunek 5 – Programator


W celu zaprogramowania pewnej sekwencji należy najpierw wybrać ilość kroków za pomocą pola wyboru liczby.




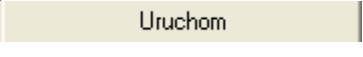
A następnie kierunku obrotów silnika za pomocą pól RadioBox.



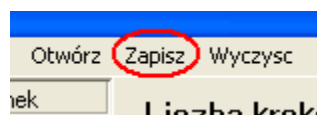
Kierunek :  Prawe  
 Lewe

Po wybraniu liczby kroków oraz kierunku w celu dodania sekwencji do listy kliknij na przycisk  lub „Dodaj” na pasku u góry okna. Ustawiona przez Ciebie sekwencja zostanie dodana do listy.

Liczba kroków	Kierunek
29	Prawo
29	Lewo

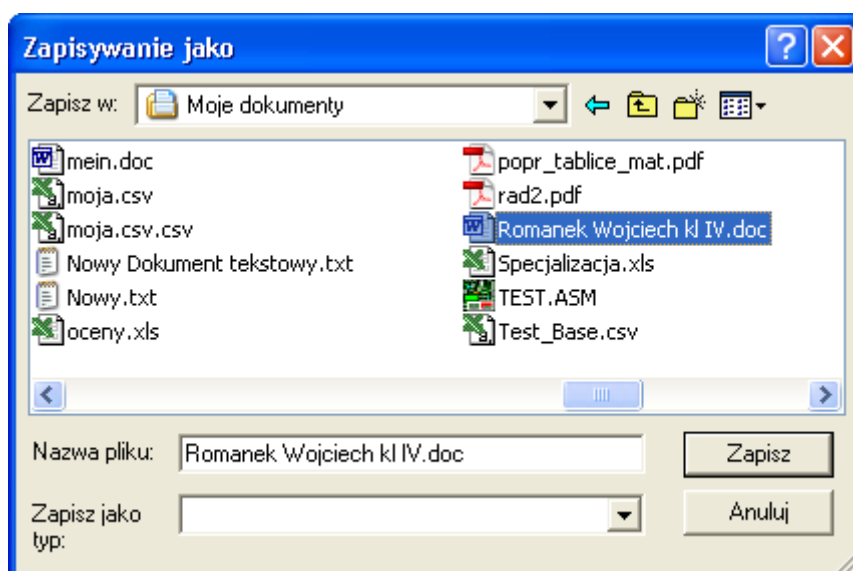
Jeśli podczas dodawania sekwencji pomyliłeś się możesz ją bez problemu usunąć z listy zaznaczając odpowiedni wiersz a następnie naciskając przycisk  lub pozycję „Usuń” na pasku u góry okna. Następnie, jeśli chcesz uruchomić podaną sekwencję należy wcisnąć przycisk  lub pozycję „Uruchom” na pasku u góry okna. Zostanie wystartowany silnik, a poszczególne wiersze w liście kroków będą podświetlane zgodnie z kolejnością ich wykonywania.

Jeśli zaprogramowaną sekwencję chcesz używać częściej możesz ją oczywiście zapisać klikając „Zapisz” na pasku u góry okna.



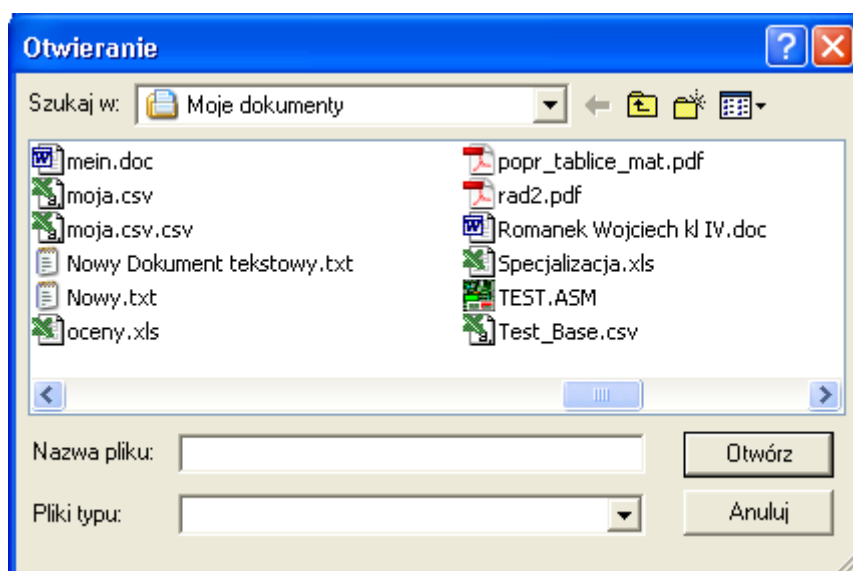
Rysunek 6 – Zapisz

Wyświetli się wówczas okno zapisu plików za pomocą, którego możliwe jest wybranie pliku do zapisania lub nadanie mu nazwy a następnie zapisanie go.



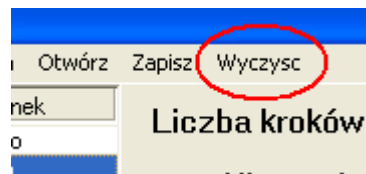
Rysunek 7 – Zapisywanie

Plik sterownika nie ma żadnego rozszerzenia (ta opcja jest nowa i w niedługim czasie zostanie zmieniona). Oczywiście każdy plik już zapisany można następnie otworzyć i odtworzyć zaprogramowaną sekwencję. W tym celu klikasz na „Otwórz” na pasku u góry okna Programatora wybierasz wcześniej zapisany plik w oknie takim jak użyte do zapisywania, po załadowaniu pliku klikasz „Uruchom” i silnik zostaje uruchomiony.



Rysunek 8 – Otwieranie

Jeszcze jedną opcją, o której nie wspomniałem jest czyszczenie listy kroków. Wywołujemy ją klikając na „Wyczyść” na pasku u góry Programatora. Wywołanie funkcji wywoła usunięcie wszystkich wierszy z listy kroków.



Rysunek 9 – Wyczyść

## Silnik krokowy - unipolarny

**Silnik krokowy** – silnik elektryczny, w którym impulsowe zasilanie prądem elektrycznym powoduje, że jego wirnik nie obraca się ruchem ciągłym, lecz wykonuje za każdym razem ruch obrotowy o ściśle ustalonym kącie. Dzięki temu, kąt obrotu wirnika jest ściśle zależny od liczby dostarczonych impulsów prądowych, a prędkość kątowna wirnika jest dokładnie równa częstotliwości impulsów pomnożonej przez wartość kąta obrotu wirnika w jednym cyklu pracy silnika.

Kąt obrotu wirnika pod wpływem działania jednego impulsu może mieć różną wartość, zależnie od budowy silnika – jest to zwykle wartość od kilku do kilkudziesięciu stopni. Silniki krokowe, zależnie od przeznaczenia są przystosowane do wykonywania od ułamków obrotu na sekundę do nawet kilku tysięcy obrotów na sekundę.

Są trzy podstawowe typy silników krokowych pod względem budowy:

- silnik z magnesem trwałym,
- silnik o zmiennej reluktancji,
- silnik hybrydowy.

Silniki krokowe są stosowane wszędzie tam, gdzie kluczowe znaczenie ma możliwość precyzyjnego sterowania ruchem.



Rysunek 10 - Użyty silnik

## Sterowanie portem LPT z użyciem Delphi i WinApi

Na początku tworzymy zmienną najlepiej globalną hLPT, w której zapiszemy uchwyt naszego portu LPT.

Za pomocą funkcji CreateFile przypisujemy do hLPT uchwyt naszego portu, dzięki niemu możemy teraz wysłać na port dane. Do wysyłania danych używamy funkcji WriteFile dane, które chcemy wysłać zapisujemy do zmiennej data a następnie wywołujemy powyższą funkcji. Oto przykładowy kod:

```
var
  hLPT:HWND;
  ret:dword;
  data:integer; // tu będą zapisywane dane wysyłane na port
begin
  hLPT:=CreateFile('LPT1',GENERIC_WRITE,0,nil,OPEN_EXISTING,0,0);
  data:=1;      //00000001 binarnie, czyli sygn. "1" na pinie zerowym
  (D0)
  WriteFile(hLPT,data,1,ret,nil);
  data:=8; // 00001000 binarnie, czyli sygn. "1" na pinie trzecim (D3)
  WriteFile(hLPT,data,1,ret,nil);
  data:=0; // czyli wyzerowanie portu
  WriteFile(hLPT,data,1,ret,nil);
  CloseHandle(hLPT); //zamknięcie portu
end;
```

## Sekwencja sterująca silnikiem

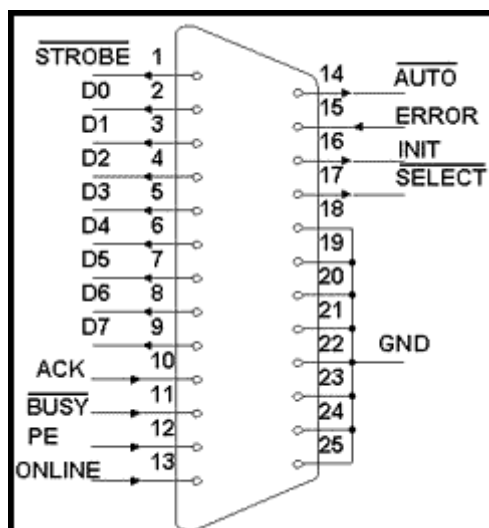
### *Połączenie z portem LPT*

Silnik unipolarny ma zazwyczaj 5 wyprowadzeń jedno masa (wspólne wszystkich cewek) oraz 4 zakończenia poszczególnych cewek. W celu poruszenia silnikiem krokowym unipolarnym należy na masę podać 0 lub połączyć ją z masą portu LPT, a na poszczególne cewki (zgodnie z kolejnością ich ułożenia) podawać stan wysoki.

Połączenie silnika z portem LPT:

- GND – GND (piny od 18 do 25)
- W1 – D0
- W2 – D1
- W3 – D2
- W4 – D3

W1 – W 4 – Wyprowadzenie poszczególnych cewek w silniku krokowym.



Rysunek 11 - Wyprowadzenia portu LPT

## Sterowanie silnikiem

Jak widać w powyższym opisie do sterowania silnika potrzebujemy 4 bity (od D0 do D3) oraz masę, tak więc po podłączeniu musimy podać sekwencję słów 4 bitowych jak niżej:

BINARNIE	DZIESIĘTNIE
0001	1
0010	2
0100	4
1000	8

W Delphi wyglądać to będzie następująco:

```
data:=1; //00000001 binarnie, czyli sygn. "1" na pinie zerowym (D0)
WriteFile(hLPT,data,1,ret,nil);
data:=2; //00000010 binarnie, czyli sygn. "1" na pinie pierwszym(D1)
WriteFile(hLPT,data,1,ret,nil);
data:=4; //00000100 binarnie, czyli sygn. "1" na pinie drugim (D2)
WriteFile(hLPT,data,1,ret,nil);
data:=8; //00001000 binarnie, czyli sygn. "1" na pinie trzecim (D3)
WriteFile(hLPT,data,1,ret,nil);
```

Po wykonaniu takiej sekwencji silnik wykona jeden krok, w celu wykonania większej ilości kroków wystarczy tą sekwencję uruchomić w pętli. Aby wykonać krok wstecz należy wykonać tą sekwencję od tyłu.

## Spis ilustracji

Rysunek 1 - Główne okno programu.....	4
Rysunek 2 - Pola zaznaczone.....	5
Rysunek 3 - Pola niezaznaczone.....	5
Rysunek 4 - Menu programu.....	6
Rysunek 5 – Programator.....	6
Rysunek 6 – Zapisz.....	7

Rysunek 7 – Zapisywanie.....	8
Rysunek 8 – Otwieranie.....	8
Rysunek 9 – Wyczyść.....	9
Rysunek 10 - Użyty silnik.....	10
Rysunek 11 - Wyprowadzenia portu LPT.....	11

## **Informacje o autorze**

Autor projektu: [Wojciech Romanek](http://www.wromanek.info) kl. IV „d” ZSE w Rzeszowie,  
<http://www.wromanek.info>

Promotor :       mgr inż. Zbigniew Niedbała

Rok szkolny: 2006/2007

Aktualną wersję tego dokumentu oraz pliki projektu znajdziesz na mojej stronie domowej: <http://www.wromanek.info>, jeśli masz pytania to pisz: [wojtekkk87@o2.pl](mailto:wojtekkk87@o2.pl)