Marcin Stępniak

Architektura systemów komputerowych
Laboratorium 9
Symulator SMS32
Urządzenia wejścia i wyjścia

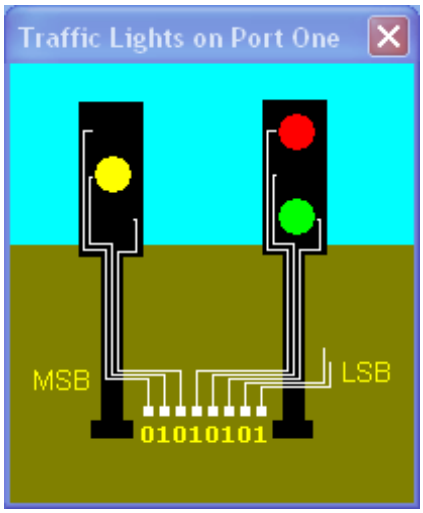
1. Informacje

Symulator SMS32 posiada kilka urządzeń peryferyjnych (wejściowych i wyjściowych) dostępnych pod różnymi portami. Porty oznaczone są odpowiednimi numerami. Instrukcja *IN numer_portu* odczytuje bajt ze wskazanego portu i umieszcza go w rejestrze AL. Do zapisu danych z rejestru AL do portu służy instrukcja *OUT numer_portu*. Tabela 1 zawiera listę urządzeń, które można w ten sposób obsłużyć.

Tabela 1. Dostępne urządzenia i numery ich portów

Numer portu	Urządzenie
00	Dane z klawiatury
01	Światła drogowe
02	Wyświetlacz siedmiosegmentowy
03	Grzałka i termostat
04	Labirynt
05	Silnik krokowy
06	Winda
07	Klawiatura
08	Klawiatura numeryczna

Urządzenia na portach 00, 07 i 08 zostaną omówione na kolejnych laboratoriach.



Rysunek 1. Okno symulatora reprezentujące światła drogowe.

1.1. Światła drogowe

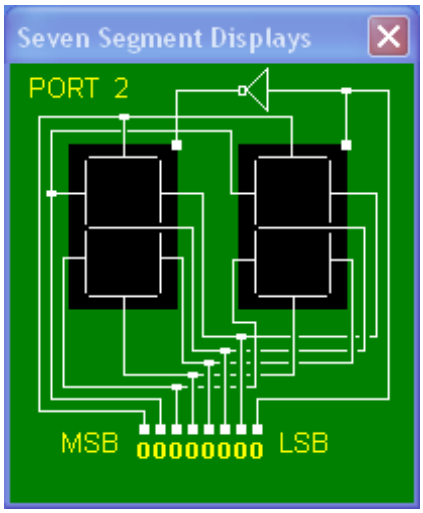
Światła są podłączone do portu 01. Po wysłaniu bajtu do tego portu, tam gdzie znajduje się jedynka, odpowiednie światło zapala się. Na rysunku 1, dane binarne to 01010101. Jeśli przyjrzeć się bliżej widać, że włączone są światła odpowiadające jedynkom w bajcie danych. 01010101 to 55 w systemie szesnastkowym. Listing 1 zawiera program do sterowania światłami.

Listing 1. Program demonstrujący działanie świateł drogowych

```
; ===========================================================================
; ===== 99Tlight.asm ========================================================
; ===== Traffic Lighte on Port 01 ===========================================
Start:
   MOV   AL,55 ; 01010101
   OUT   01 ; Send the data in AL to Port 01 (the traffic lights)

   MOV   AL,AA ; 10101010
   OUT   01 ; Send the data in AL to Port 01 (the traffic lights)

   JMP   Start

   END
; ===========================================================================
```

1.2. Wyświetlacz siedmiosegmentowy

Wyświetlacz siedmiosegmentowy (rys. 2) jest podłączony do portu 02. Po wysłaniu bajtu do tego portu, tam gdzie znajduje się jedynka, odpowiedni segment zapala się. Pierwszy bit z prawej strony określa, która z dwóch grup

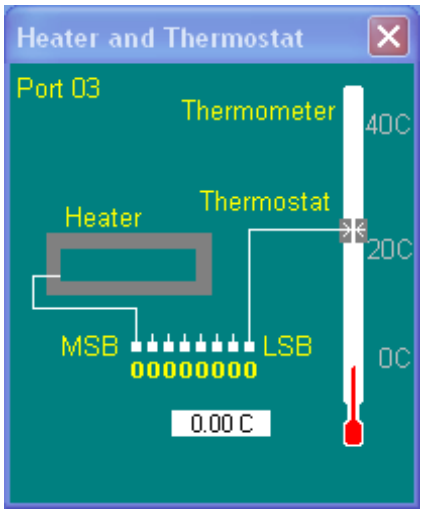


Rysunek 2. Okno symulatora reprezentujące wyświetlacz 7 segmentowy.

będzie aktywna. Jest to prosty przykład multipleksowania. Jeżeli najmniej znaczący bit (LSB) wynosi zero, to segmenty po lewej stronie będą aktywne. Jeżeli bit najmniej znaczący (LSB) ma wartość jeden, to będą aktywne segmenty po prawej stronie. Listing 2 zawiera przykładowy program do sterowania wyświetlaczem.

Listing 2. Program demonstrujący działanie wyświetlacza siedmiosegmentowego

```
; ======================================================
; ===== 99sevseg.asm ===================================
; ===== Seven Segment Displays Port 02 =================
Start:
   MOV   AL,FA ; 1111 1010
   OUT   02 ; Send the data in AL to Port 02

   MOV   AL,0  ; 0000 0000
   OUT   02 ; Send the data in AL to Port 02

   MOV   AL,FB ; 1111 1011
   OUT   02 ; Send the data in AL to Port 02

   MOV   AL,1  ; 0000 0001
   OUT   02 ; Send the data in AL to Port 02

   JMP   Start

   END
; ======================================================
```



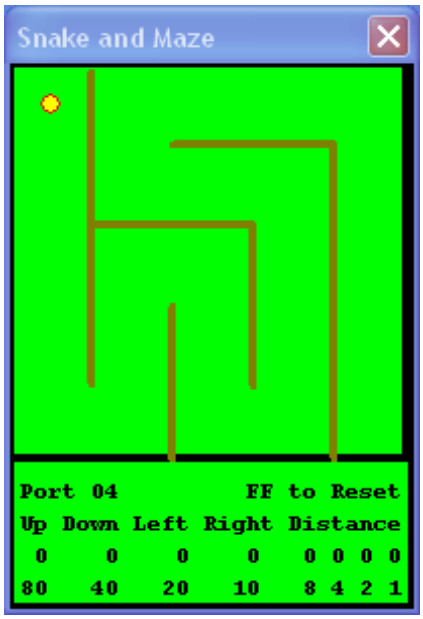
Rysunek 3. Okno symulatora reprezentujące grzałkę i termostat.

1.3. Grzałka i termostat

Układ grzałki i termostatu (rys. 3) jest podłączony do portu 03. Aby wyłączyć grzałkę należy wysłać 00 do portu 3, a $80_{(16)}$ w celu jej włączenia. Wczytanie danych z portu 03 pozwala sprawdzić stan termostatu. Najmniej znaczący bit zostanie ustawiony, jeżeli temperatura przekroczy określony poziom. Listing 3 zawiera przykładowy program do sterowania grzałką w zależności od temperatury. Jest on jednak niekompletny, bo nie działa w sposób ciągły. Kliknięcie na termometr pozwala na ustawienie temperatury, co może być przydatne przy testowaniu programów korzystających z tego urządzenia.

Listing 3. Program demonstrujący działanie termostatu

```
; ===== Heater and Thermostst on Port 03 ======================
; ===== 99Heater.asm ==========================================
   MOV   AL,0  ; Code to turn the heater off
   OUT   03 ; Send code to the heater

   IN 03 ; Input from Port 03
   AND   AL,1  ; Mask off left seven bits
   JZ Cold  ; If the result is zero, turn the heater on
   HALT     ; Quit

Cold:
   MOV   AL,80 ; Code to turn the heater on
   OUT   03 ; Send code to the heater

   END
; ============================================================
```



Rysunek 4. Okno symulatora reprezentujące labirynt i poruszającego się w nim węża.

1.4. Labirynt

Dane przesyłane do portu 04 kontrolują ruch węża. Cztery lewe (najbardziej znaczące) bity kontrolują kierunek ruchu węża:

— 80 - Góra
— 40 - Dół
— 20 - Lewo
— 10 - Prawo

Cztery prawe (najmniej znaczące) bity kontrolują dystans jaki wąż ma przebyć, np. 4F oznacza przesunięcie węża w dół na dystansie 15 (4 oznacza dół, a $F_{(16)} = 15_{(10)}$).

Listing 4 zawiera program resetujący pozycję węża i wykonujący nim 4 ruchy.

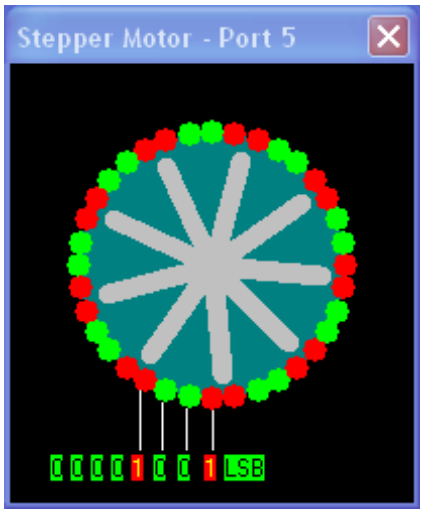
Listing 4. Program sterujący wężem w labiryncie

```
; ========================================================
; ===== 99snake.asm ======================================
; ===== Snake and Maze ===================================

Start:
   MOV   AL,FF ; Special code to reset the snake.
   OUT   04 ; Send AL to port 04 to control the snake.

   MOV   AL,4F ; 4 means DOWN.  F means 15.
```

```
   OUT    04 ; Send 4F to the snake
   OUT    04 ; Send 4F to the snake
   OUT    04 ; Send 4F to the snake
   OUT    04 ; Send 4F to the snake

   JMP    Start

   END
; ============================================================
```

1.5. Silnik krokowy

Silnik krokowy pozwala na precyzyjne sterowanie jego ruchem. Silnik w symulacji (rys. 5) jest sterowany przez aktywowanie czterech magnesów po kolei. Jest możliwe, aby sterować silnikiem w półkrokach zasilając odpowiednie pary magnesów. Jeżeli magnesy są zasilane w niewłaściwej kolejności, silnik skarży się wydając dźwięk z głośnika komputera. Listing 5 zawiera przykładowy program sterujący silnikiem krokowym (sterowanie pełnokrokowe jak i półkrokowe).



Rysunek 5. Okno symulatora reprezentujące silnik krokowy.

Listing 5. Program demonstrujący działanie silnika krokowego

```
; ================================
; ===== 99Step.asm ===============
; ===== Stepper Motor ============
   mov    al,1  out    05
   mov    al,2  out    05
   mov    al,4  out    05
   mov    al,8  out    05
```

```
   mov   al,9  out   05
   mov   al,1  out   05
   mov   al,3  out   05
   mov   al,2  out   05
   mov   al,6  out   05
   mov   al,4  out   05
   mov   al,c  out   05
   mov   al,8  out   05
   mov   al,9  out   05
   mov   al,1  out   05

   end
; ================================
```

1.6. Winda

Urządzenie windy (rys. 6) pozwala na sterowanie przyciskami windy, silnikiem i odczytywanie danych z czujników na górze i dole szybu.

Sygnały wejściowe

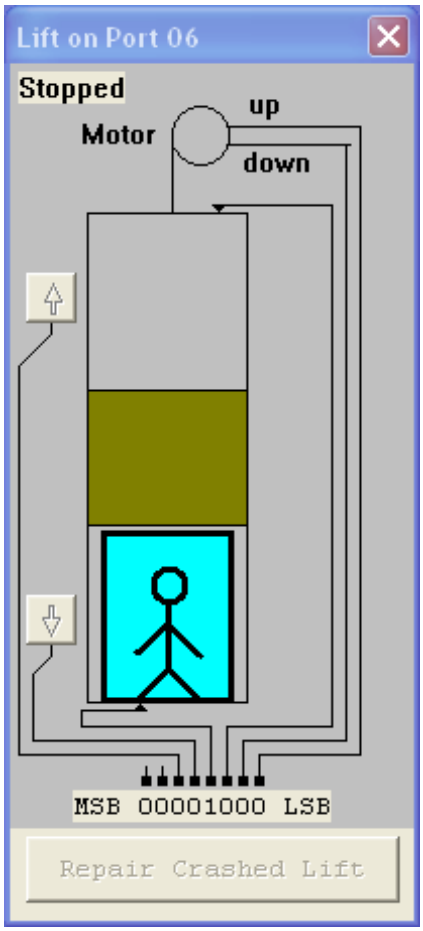
Bity 8 i 7 są nieużywane. Bit 6 jest podłączony do górnego przycisku przywołania. Bit 5 jest podłączony do dolnego przycisku przywołania. Jeśli te przyciski zostaną kliknięte myszą, odpowiednie bity zostaną ustawione. Bit 4 jest podłączony do czujnika, który wykrywa kabinę windy i ustawia bit w stan wysoki, gdy kabina osiągnie dno szybu. Bit 3 przechodzi w stan wysoki, gdy klatka windy osiągnie szczyt szybu.

Wyjścia

Ustawienie bitu 2 steruje silnikiem w taki sposób, aby kabina opadała. Ustawienie bitu 1 włącza silnik powodując, że kabina jedzie w górę.

1.7. Dodatkowe informacje

1. http://www.softwareforeducation.com/sms32v50/sms32v50_manual/400-Peripherals.htm
2. https://pl.wikipedia.org/wiki/Silnik_krokowy
3. https://pl.wikipedia.org/wiki/Termostat



Rysunek 6. Okno symulatora reprezentujące windę.

2. Zadania

2.1. Zadanie 1

Wykorzystując urządzenie świateł drogowych należy zasymulować rzeczywistą sekwencję zmiany świateł (tylko kolejne zmiany kolorów, bez opóźnień).

2.2. Zadanie 2

Wyświetlić na wyświetlaczu siedmiosegmentowym dzień swoich urodzin (tylko dzień miesiąca, bez numeru miesiąca). Jeżeli jest to liczba jednocyfrowa, to należy na pierwszej pozycji wstawić 0.

2.3. Zadanie 3

Napisać program, który wykorzystuje termostat i grzałkę w taki sposób, aby temperatura utrzymywała się na stałym poziomie.

2.4. Zadanie 4

Napisać program, który doprowadzi węża do końca labiryntu bez kolizji ze ścianami.

2.5. Zadanie 5

Napisać program sterujący silnikiem krokowym w sposób ciągły. Silnik ma obracać się w kierunku przeciwnym do ruchu wskazówek zegara. Można wykorzystać pełne kroki lub/i półkroki.