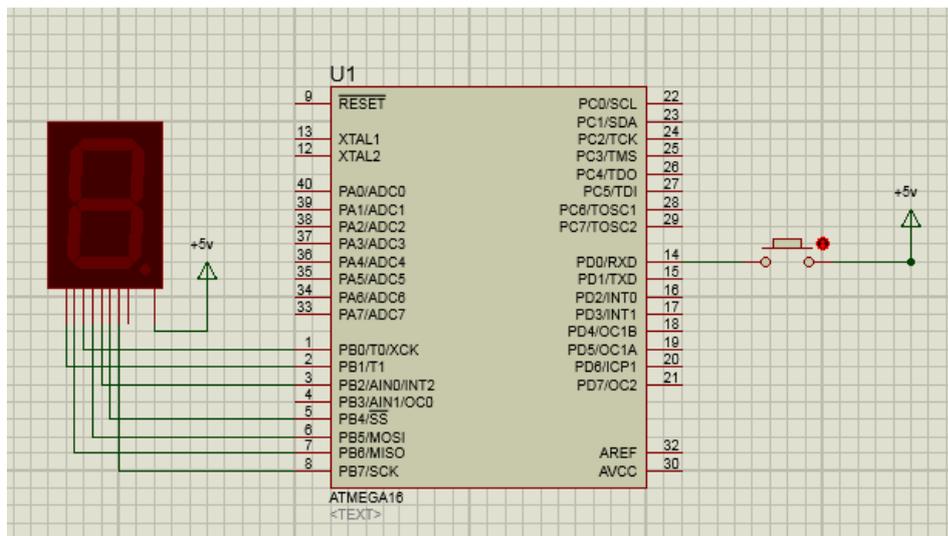


Нужно выполнить программное моделирование микропроцессорной системы. При выполнении задания осуществить выбор микроконтроллера (AVR, PIC), создать программный код (ассемблер, C, C++), запрограммировать микроконтроллер, продемонстрировать работу микропроцессорной системы в программе Proteus CAD Professional.

Произведите вывод символов (0,1,2,3) на семисегментный индикатор в программе Proteus.

Решение

Подключите семисегментный индикатор 7SEG-MPX1-CA к микросхеме АТМЕГА16.



С помощью программы AVR Studio 4 создайте hex-файл с кодом:

```
include "m16def.inc"
```

```
.def temp = r16
.def counter = r17
.def coarse = r18
.def middle = r19
.def fine = r20
```

```
.equ SEG7_PORT= PORTB
.equ SEG7_DDR = DDRB
.equ BUTTON_PORT= PORTD
.equ BUTTON_DDR = DDRD
.equ BUTTON_INPUT = PIND
```

```
.cseg
.org 0
rjmp Reset
```

```
segments: //цифры
.db 0b10001000, 0b10111110, 0b00011001, 0b00011100 // массив [0,1,2,3].
```

```
Reset:
LDI temp,LOW(RAMEND)
OUT SPL,temp
LDI temp,HIGH(RAMEND)
```

```
OUT SPH,temp
```

```
ser temp  
out SEG7_DDR, temp
```

```
ldi temp, 0b1111110  
out BUTTON_DDR, temp
```

```
Reset_counter:  
clr counter
```

```
ldi Zl,low(segments * 2)  
ldi Zh,high(segments * 2)
```

```
Start:  
lpm temp,Z+  
out SEG7_PORT, temp
```

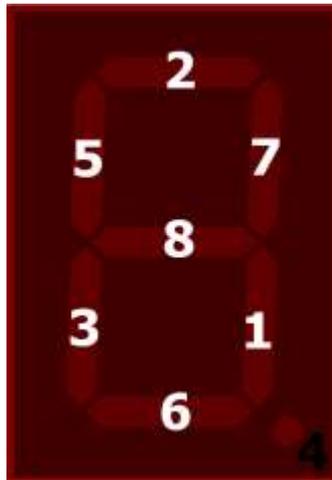
```
loop_while_button_held:  
sbic BUTTON_INPUT,0  
rjmp loop_while_button_held  
rcall delay  
inc counter  
cpi counter, 4 //счётчик массива из цифр 0,1,2,3.  
brne Start
```

```
rjmp Reset_counter
```

```
delay:      ldi    coarse,$02  
delay1:    ldi    middle,$FF  
delay2:    ldi    fine,$FF  
delay3:    dec    fine  
           brne   delay3  
           dec   middle  
           brne   delay2  
           dec   coarse  
           brne   delay1  
           ret
```

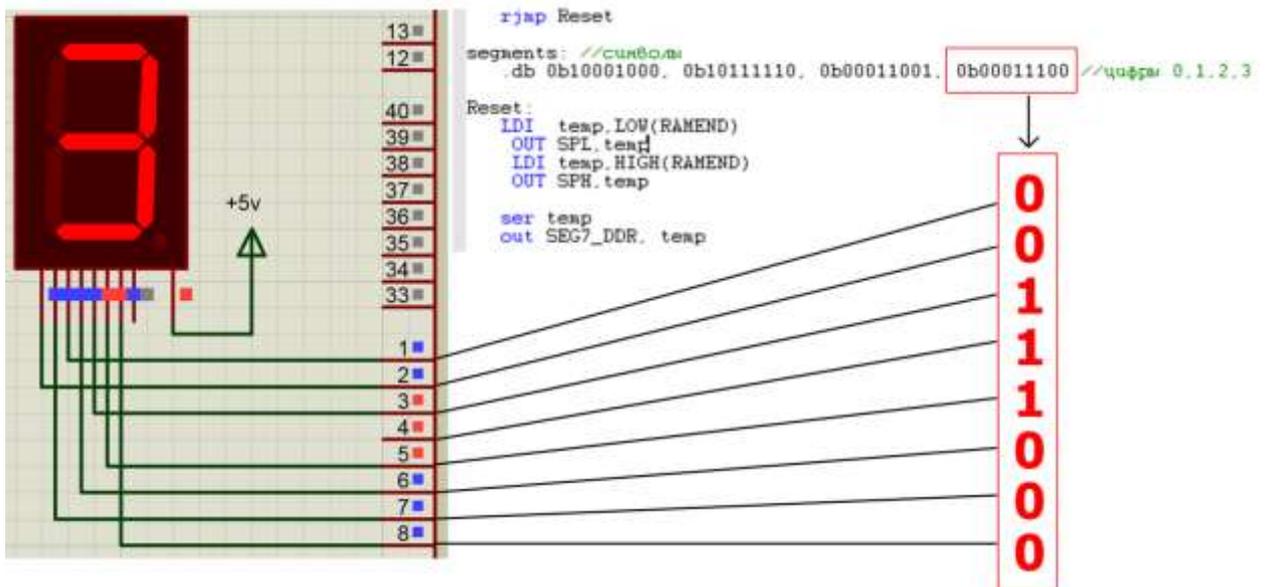
Добавьте полученный HEX-файл в ATMEGA16. Проверьте правильность кода, запустив симуляцию. На дисплее должны сменяться цифры 0,1,2,3.

Бинарная последовательность, соответствующая символам передаётся на portB микросхемы. Выводы микросхемы подключены к сегментам дисплея следующим образом:



*Сигнал с четвёртого вывода отвечает за точку. В данном примере не используется.

Таким образом бинарная последовательность отправляется на дисплей:



Обратите внимание, что первый разряд последовательности отправляется на восьмой вывод, а восьмой на первый.