دانشگاه کردستان
University of Kurdistan
زانکۆی کوردستان

Microprocessor Lab

Simulate traffic light using ATMEAG64 with MPX counter

Professor:

Dr. Hooshmand Ghasri

Parsa Zahedi

Required tools:

ATMEGA 64 micro controller, 1 traffic light modules, 1 MPX Seven Segment common cathode.

How to:

Designing the schematic was very easy. Add the required tools and connect them to ATMEGA 64 microcontroller as shown in figure 1.

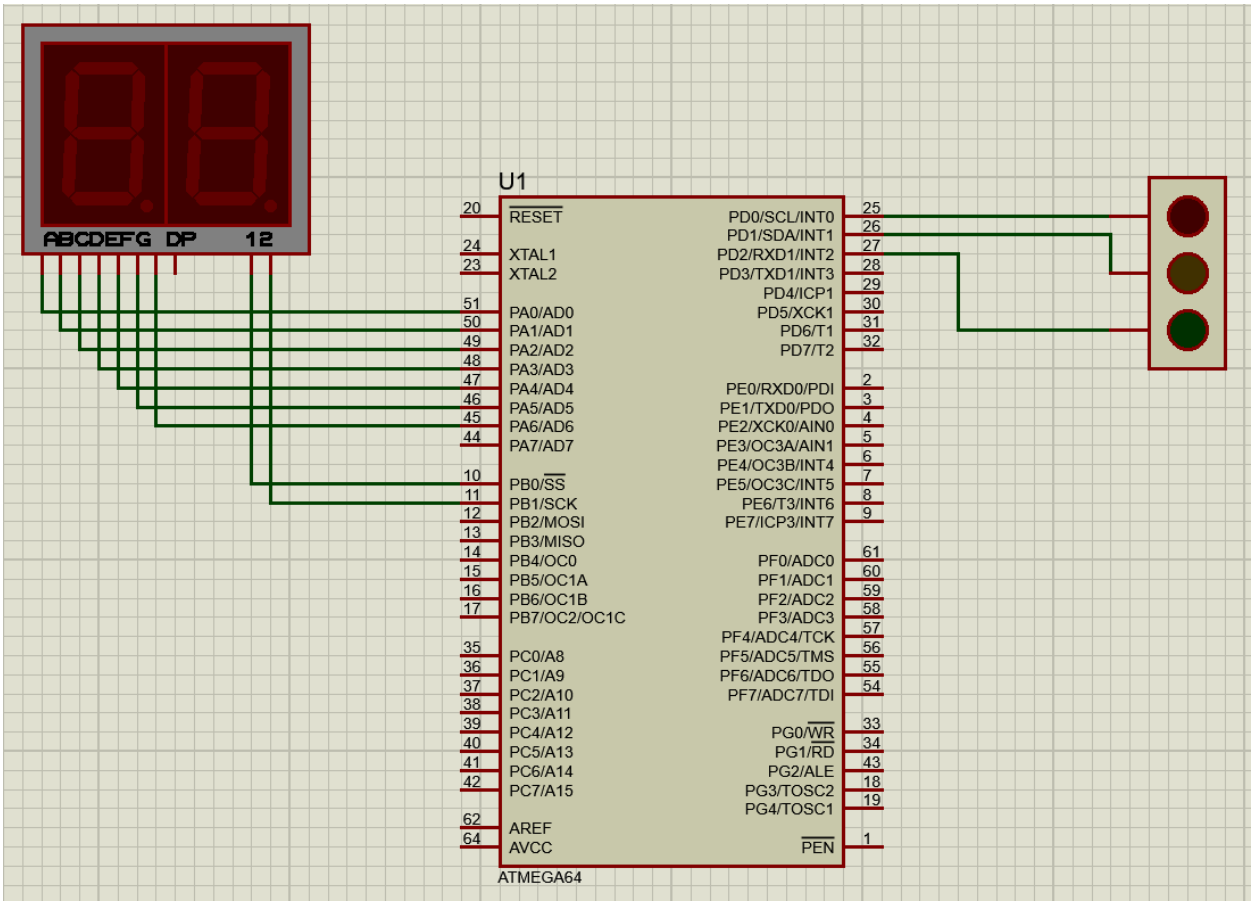

Figure 1- Schematic of Exercise 2

To start the project, I completely ignored the traffic light and only focused on the 7-segment counter. This is called breaking a huge problem into smaller ones. The first problem is making a counter that counts from 99 to 0. My first solution was a nested loop but the problem with that is you can only display the “tens” or the “ones” you cannot display both or switch between so fast that naked human eye won’t notice. Then I had the idea of getting the two digit number and separating “ones” and “tens” using / and % operators in CPP.

```c
#include <avr/io.h>
#include <util/delay.h>

unsigned char display[] = {0X3F, 0X06, 0X5B, 0X4F, 0X66, 0X6D, 0X47D, 0X07, 0X7F, 0X6F};

int main(void)
{
	int ones, tens = 0;
	DDRA = 0XFF;
	DDRB = 0XFF;
	DDRD = 0XFF;
	while (1)
	{
		PORTD = 0b00000001;//RED
		for (int i = 99; i >= 0; i--)
		{
			ones = i % 10;
			tens = i / 10;


			for (int j = 0; j <50; j++)
			{
				PORTB = 0b00000010;
				PORTA = display[tens];//T

				_delay_ms(10);

				PORTB = 0b00000001;
				PORTA = display[ones];//O

				_delay_ms(10);
			}
		}

		PORTD = 0b00000100;//GREEN
		for (int i = 30; i >= 0; i--)
		{
			ones = i % 10;
			tens = i / 10;


			for (int j = 0; j <50; j++)
			{
				PORTB = 0b00000010;
				PORTA = display[tens];//T

				_delay_ms(10);

				PORTB = 0b00000001;
				PORTA = display[ones];//O

				_delay_ms(10);
			}
		}
		PORTD = 0b00000010;//YELLOW
		for(int i = 0; i <10; i++)
		{

			for (int j = 0; j <50; j++)
			{
				PORTB = 0b00000010;
				PORTA = display[0];

				_delay_ms(10);
```

```
65.
66.                                     PORTB = 0b00000001;
67.                                     PORTA = display[0];
68.
69.                                     _delay_ms(10);
70.                             }
71.                             for (int j = 0; j <50; j++)
72.                             {
73.                                     PORTB = 0b00000010;
74.                                     PORTA = 0X00;
75.
76.                                     _delay_ms(10);
77.
78.                                     PORTB = 0b00000001;
79.                                     PORTA = 0X00;
80.
81.                                     _delay_ms(10);
82.                             }
83.                     }
84.             }
85. }
86.
```

Code analysis:

Obviously first of all ATMEGA pins are initialized and set then display array for 7-segment is written and then we'll use "ones" and "tens" strategy that I explained above to create a counter. The next problem is 7-segment blinking and to solve that we need to quickly switch the select line on the MPX. As shown in figure 2 with a yellow square.

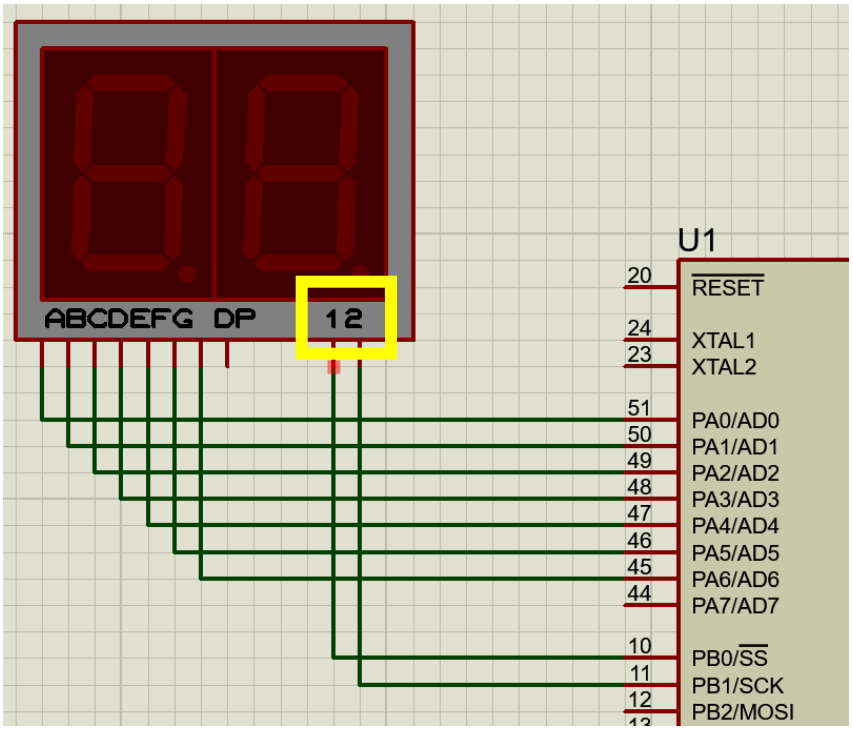

Figure 2- Select line for ones and tens

We control this select line using PORTB on the microcontroller.

```
15. for (int i = 99; i >= 0; i--)
16.                 {
17.                           ones = i % 10;
18.                           tens = i / 10;
19.
20.
21.                           for (int j = 0; j <50; j++)
22.                           {
23.                                     PORTB = 0b00000010;
24.                                     PORTA = display[tens];//T
25.
26.                                     _delay_ms(10);
27.
28.                                     PORTB = 0b00000001;
29.                                     PORTA = display[ones];//O
30.
31.                                     _delay_ms(10);
32.                           }
33.                 }
```

This code snippet was the main challenge in the exercise. To create a counter using MPX 7-Segment that doesn't blink (at least to naked human eye). We also have to make sure it takes one second for the counter to change that is why our second for loop repeats exactly 50 times and in each of those times select line toggles with a 10ms delay from showing "tens" to "ones".

- **A video file is attached that shows our program working!**