Abu Dhabi University

Microprocessors and Firmware Programming

Lab Report 5
Controlling 7 Segment Display

Author:
Muhammad Obaidullah
1030313

Supervisor:
Dr. Mohammed Assad Ghazal

Section 1

June 27, 2012

Abstract

In this Lab we controlled a seven segment display using a micro-controller. The display counted up every 1 second.

1 Introduction

A seven Segment Display has 8 connection legs. To show each character we have to decide whether to turn ON a particular LED segment or no.

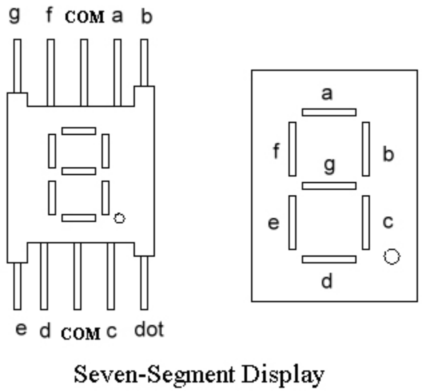
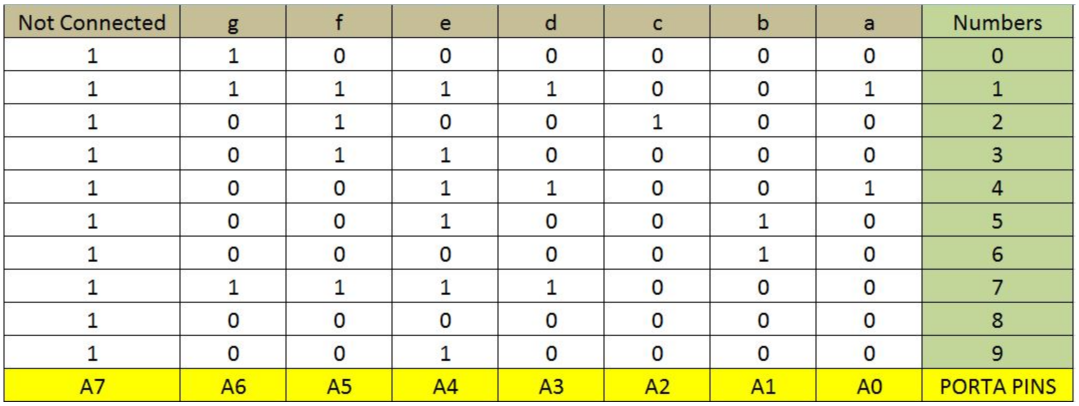


Figure 1: Pin configuration of 7 Segment Display



Not Connected	g	f	e	d	c	b	a	Numbers
1	1	0	0	0	0	0	0	0
1	1	1	1	1	0	0	1	1
1	0	1	0	0	1	0	0	2
1	0	1	1	0	0	0	0	3
1	0	0	1	1	0	0	1	4
1	0	0	1	0	0	1	0	5
1	0	0	0	0	0	1	0	6
1	1	1	1	1	0	0	0	7
1	0	0	0	0	0	0	0	8
1	0	0	1	0	0	0	0	9
A7	A6	A5	A4	A3	A2	A1	A0	PORTA PINS

Figure 2: Chart showing which Segments to turn On

2 Experiment Set-up

The ATMega16 chip was already mounted on a safety bracket. We had to place the bracket with the micro-controller on to the breadboard. Then we connected the micro-processor to each segment and also connected VCC to the common pin. *Figure 3*

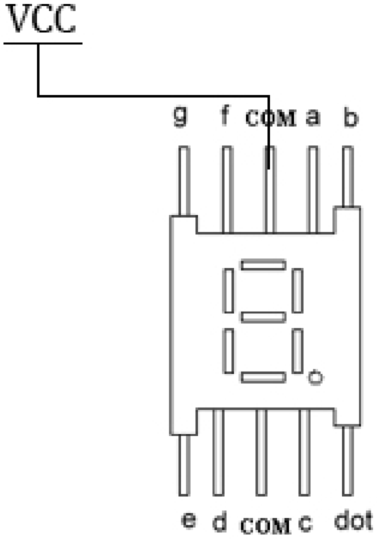


Figure 3: This is how we connect LEDs and the push buttons to the ATMega16

3 List of Equipment used

- ATMega16 micro-controller chip.
- JTAG MKII programmer.
- Wires.
- Breadboard.
- Mounting bracket for micro-controller.
- 7 Segment Display.
- 5V power supply.
- AVR Studio IDE.

4 Procedure

4.1 Describing the Inputs and Outputs

Table 1: Inputs, Outputs, and States

States	Inputs	Outputs
Show 0	Timer 0	Segment A
Show 1		Segment B
Show 2		Segment C
Show 3		Segment D
Show 4		Segment E
Show 5		Segment F
Show 6		Segment G
Show 7		Timer 0
Show 8		
Show 9		

4.2 Writing Code.

- Start AVR Studio and click on File/New/New Project.
- Write the following code into the AVR .c file.

```
#include <avr/io.h>
#include <avr/interrupt.h>
int count;
count = 0;
int current;
int numbers[] =
{
0b11000000,        \\0
0b11111001,        \\1
0b10100100,        \\2
0b10110000,        \\3
0b10011001,        \\4
0b10010010,        \\5
0b10000010,        \\6
0b11111000,        \\7
0b10000000,        \\8
0b10010000,        \\9


}

ISR (TIMER0_OVF_VECT)
{
        TCNT0 = 155;
        count++;
        if (count == 10000)
        {
                if (current == 9)
                {
                        current = 0;
                }
                else
                {
                        current++;
                }
        count = 0;

        }


}




int main()
{
TCNT0 = 155;
TIMSK = 0b00000001;
OCR0 = 0x00;
DDRB = 0xFF;
GICR = 0b01000000;
MCUCR = 0b00000011;

        current = 0;
        switch (current)
        {
                case 0: PORTB = numbers[0]; break;
                case 1: PORTB = numbers[1]; break;
                case 2: PORTB = numbers[2]; break;
                case 3:        PORTB = numbers[3]; break;
                case 4:        PORTB = numbers[4]; break;
                case 5:        PORTB = numbers[5]; break;
                case 6:        PORTB = numbers[6]; break;
                case 7:        PORTB = numbers[7]; break;
                case 8:        PORTB = numbers[8]; break;
                case 9:        PORTB = numbers[9]; break;

        }
}
```

4.3 Uploading the code to ATMega16.

- Connect JTAG to the computer through a USB cable and connect the JTAG Pins to the micro-controller.

- connect the Seven segment to the Atmega port B.

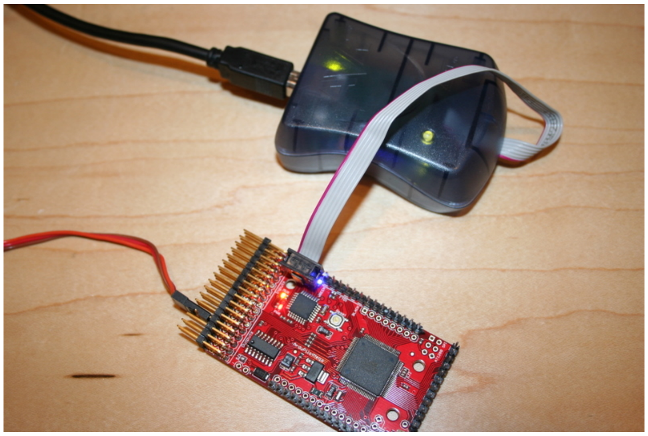


Figure 4: Connecting JTAG MKII to the ATMega 16

- Click build and compile in AVR Studio.
- Run the code.

5 Results and Discussions

At the end of these exercises we got the following results:-

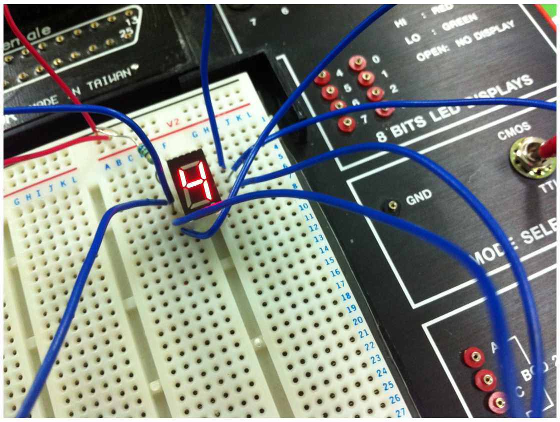


Figure 5: Seven Segment display is showing 4.

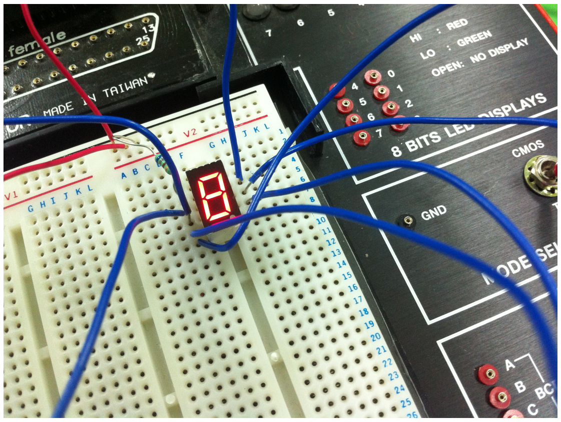


Figure 6: Seven Segment display is showing 8.

- Successful operation of seven segment display.
- If we use timer0, which is a hardware clock, it keeps the micro-controller less busy and allows it to perform other functions.
- For enabling interrupts we have to call a function sei()
- For using interrupts we have to include a header file “avr/interrupt.h”.
- TCCR0 stands for Timer Counter Control Register, which controls the timer clock and also enables it.
- Every digit incremented after 1 second.

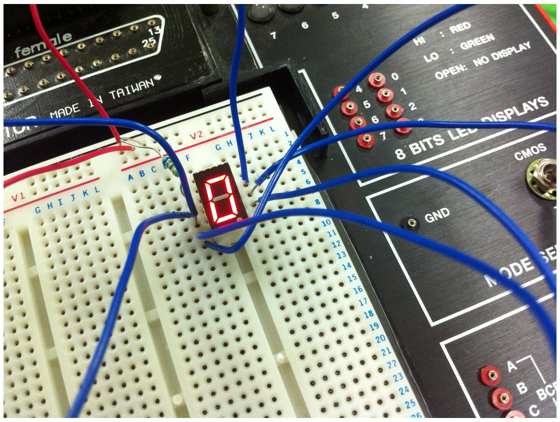


Figure 7: Seven Segment display is showing 0.

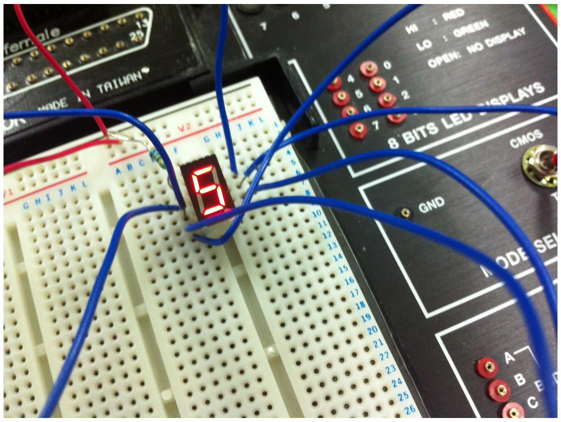


Figure 8: Seven Segment display is showing 5.

6 Conclusion

- All the segments have to be connected to the Port B because, we have set the Port B as output and not the other ports.
- If a pin is low, ie. grounded, it would light up.
- The ATMEGA16 has 3 hardware timers for performing several operations.

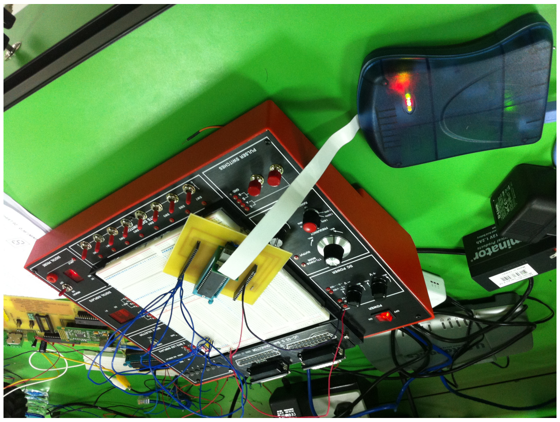


Figure 9: East-West is green while North-South is Red.

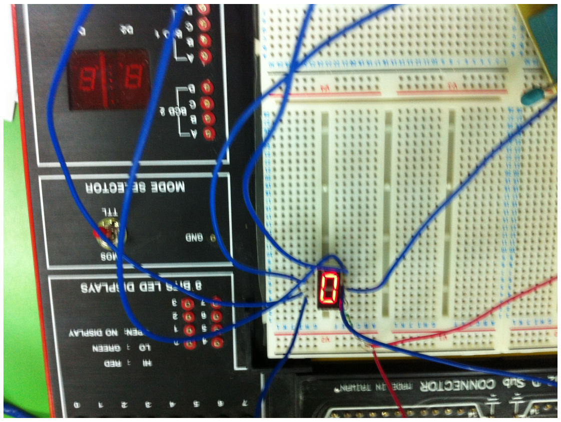


Figure 10: East-West is in warning state after being Green for 30 seconds