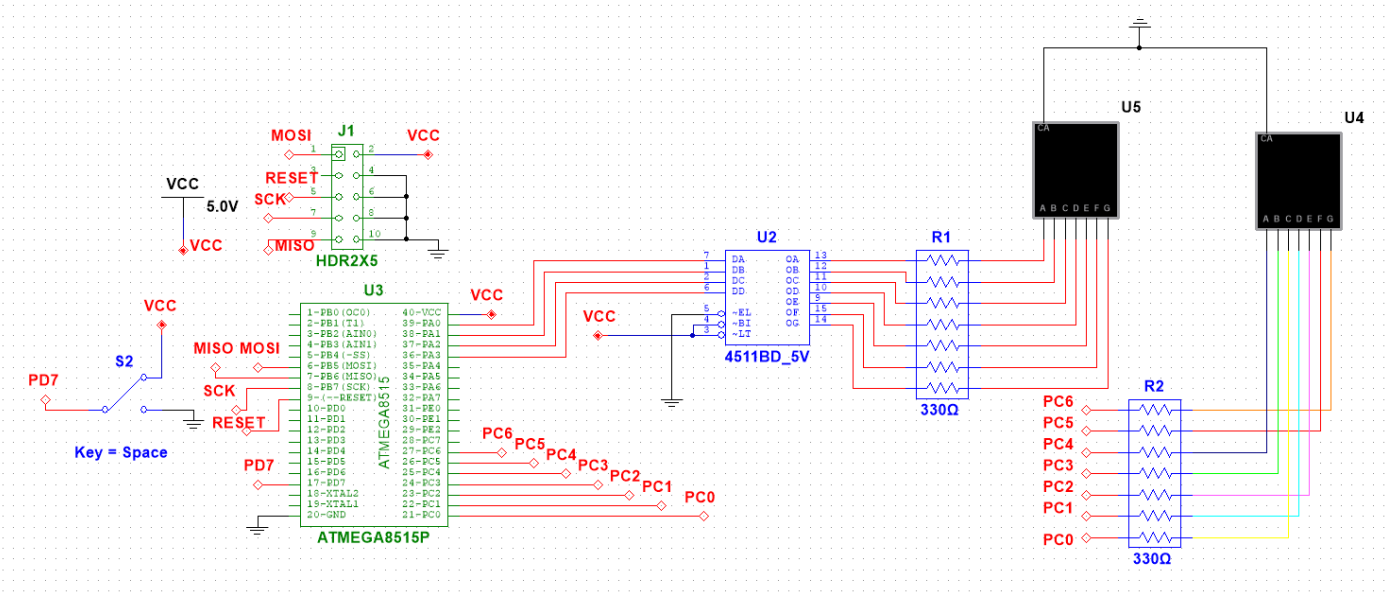
Laboratory 6 – Loop Applications

Alexandre Ciugulan
Department of Electronics Engineering, Dawson College
EMBEDDED SYSTEM PROGRAMMING
Professor Nick Markou
March 25, 2024

1.0 – Purpose

In this lab, we will be using the C language and the AVR concurrently to control external hardware. We will drive a 7-segment display to showcase numeric values, enabling time-related control applications. The lab is divided into four tasks. The first task is to create a counter that counts up to nine on PORT A using a for loop. The second task involves using a combination of the switch/case and do/while constructs to count up to five on PORT C. The third task combines the previous two tasks to repetitively count from 00 to 59. Finally, the fourth task modifies the third task to reset both counters to 00 when the most significant bit of PORT D is high, by adding an if statement.

2.0 – Circuit Diagrams



3.0 – Program Code

Task 1:

```
#include <avr/io.h>
#include <avr/io.h>
#include <util/delay.h>
#define F_CPU 1000000


int main(void)
{
	// Variable Init
	int ledf;
	DDRA = 0xFF;

    while (1)
    {
		// for loop that counts from 0-9
		for(ledf = 0; ledf <= 9; ledf++)
		{
			PORTA = ledf;
			_delay_ms(1000); /* 1 second interval delay */

		}
    }
}
```

Task 2:

```
#include <avr/io.h>
#include <util/delay.h>
#define F_CPU 1000000

int main(void)
{
	// Variable Init
	int led = 0;
	DDRC = 0xFF;

	/* loop that counts from 0-5 with switch/case */
	while (1)
	{
		do
		{
			switch(led)
			{
				case 0 : PORTC = 0x3F; /*0*/
				_delay_ms(1000);
				break;
				case 1 : PORTC = 0x09; /*1*/
				_delay_ms(1000);
				break;
				case 2 : PORTC = 0x5E; /*2*/
				_delay_ms(1000);
				break;
				case 3 : PORTC = 0x5B; /*3*/
				_delay_ms(1000);
				break;
				case 4 : PORTC = 0x69; /*4*/
				_delay_ms(1000);
				break;
				case 5 : PORTC = 0x73; /*5*/
				_delay_ms(1000);
				break;
			}
			led ++;
		} while (led <= 5);
		//Restarts the counter
		led = 0;
	}
}
```

Task 3:

```
#include <avr/io.h>
#define F_CPU 1000000
#include <util/delay.h>


int main(void)
{
	// Variable Init
	int led = 0;
	int ledf;
	DDRC = 0xFF;
	DDRA = 0xFF;

	/* loop that counts from 0-5 with switch/case */
	while (1)
	{
		do
		{
			switch(led)
			{
				case 0 : PORTC = 0x3F; /*0*/
				_delay_ms(1000);
				break;
				case 1 : PORTC = 0x09; /*1*/
				_delay_ms(1000);
				break;
				case 2 : PORTC = 0x5E; /*2*/
				_delay_ms(1000);
				break;
				case 3 : PORTC = 0x5B; /*3*/
				_delay_ms(1000);
				break;
				case 4 : PORTC = 0x69; /*4*/
				_delay_ms(1000);
				break;
				case 5 : PORTC = 0x73; /*5*/
				_delay_ms(1000);
				break;
			}
			// for loop that counts from 0-9
				for(ledf = 0; ledf <= 9; ledf++)
					{
						PORTA = ledf;
						_delay_ms(1000);
					}
				led++;
		} while (led <= 5);
		// Restarts the counter
		led = 0;
	}
}
```

Task 4:

```
#include <avr/io.h>
#define F_CPU 1000000
#include <util/delay.h>

int main(void)
{
	// Variable Init
	int led = 0;
	int ledf;
	DDRC = 0xFF;
	DDRA = 0xFF;
	DDRD = 0xFF;

	while (1)
	{
			do
			{
				/* loop that counts from 0-5 with switch/case */
				switch(led)
				{
					case 0 : PORTC = 0x3F; /*0*/
					_delay_ms(1000);
					break;
					case 1 : PORTC = 0x09; /*1*/
					_delay_ms(1000);
					break;
					case 2 : PORTC = 0x5E; /*2*/
					_delay_ms(1000);
					break;
					case 3 : PORTC = 0x5B; /*3*/
					_delay_ms(1000);
					break;
					case 4 : PORTC = 0x69; /*4*/
					_delay_ms(1000);
					break;
					case 5 : PORTC = 0x73; /*5*/
					_delay_ms(1000);
					break;
				}
				// for loop that counts from 0-9
				for(ledf = 0; ledf <= 9; ledf++)
				{
				//Check if the MSB is HIGH, If it is HIGH it resets both counters
					if (PIND & (1 << PIND7))
					{
						ledf = 0x00;
						led = 0;
						PORTC = 0x3F;
					}
					PORTA = ledf;
					_delay_ms(1000);

				}
				led++;
			} while (led <= 5);
			// Restarts the counter
			led = 0;
		}
}
```

4.0 – Results

Task 1:

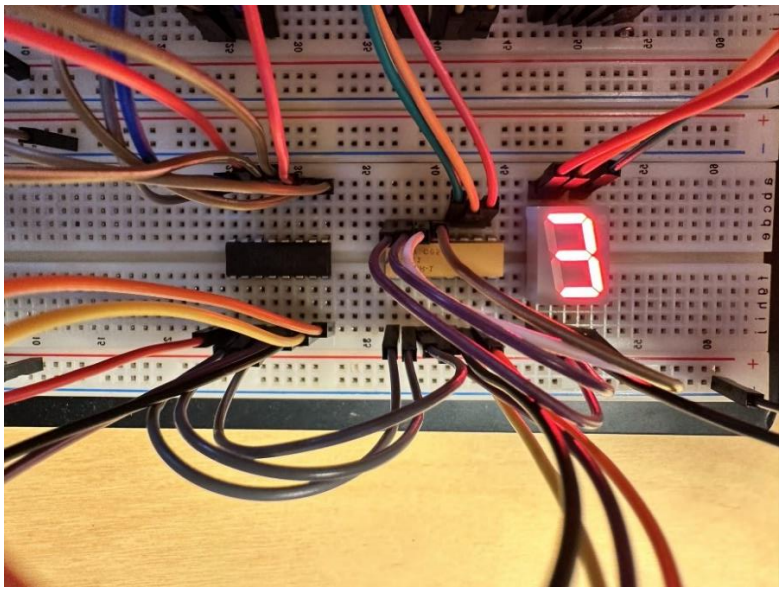


Figure 1: Showing the for loop.

Task 2 :

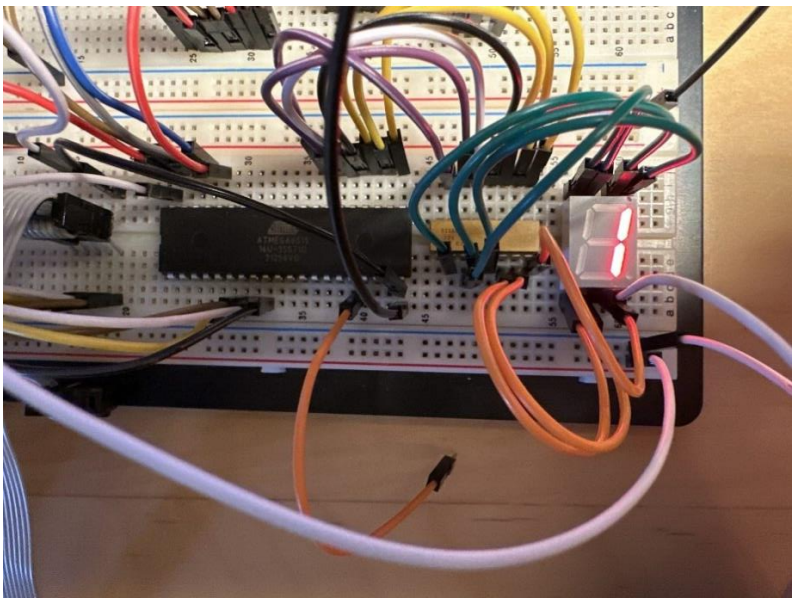


Figure 2: Switch/Case counter.

Task 3:

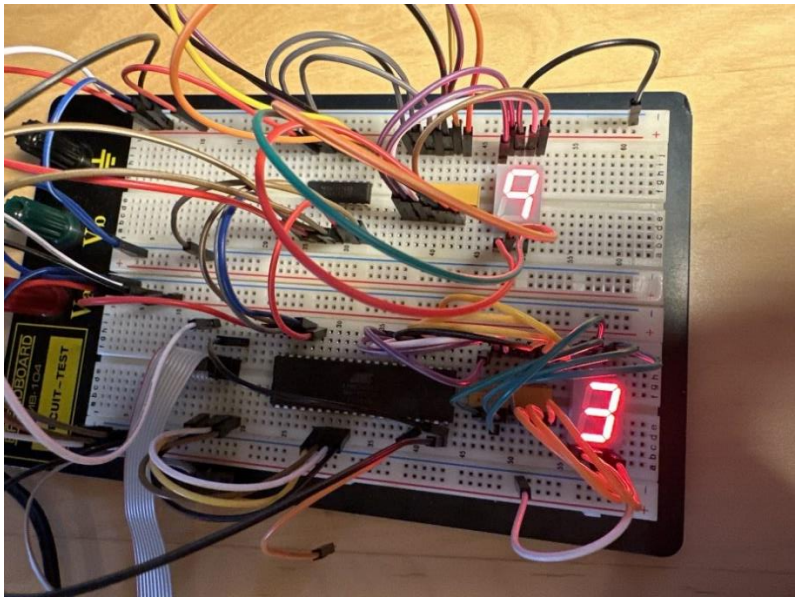


Figure 3: Both counters in action

Task 4:

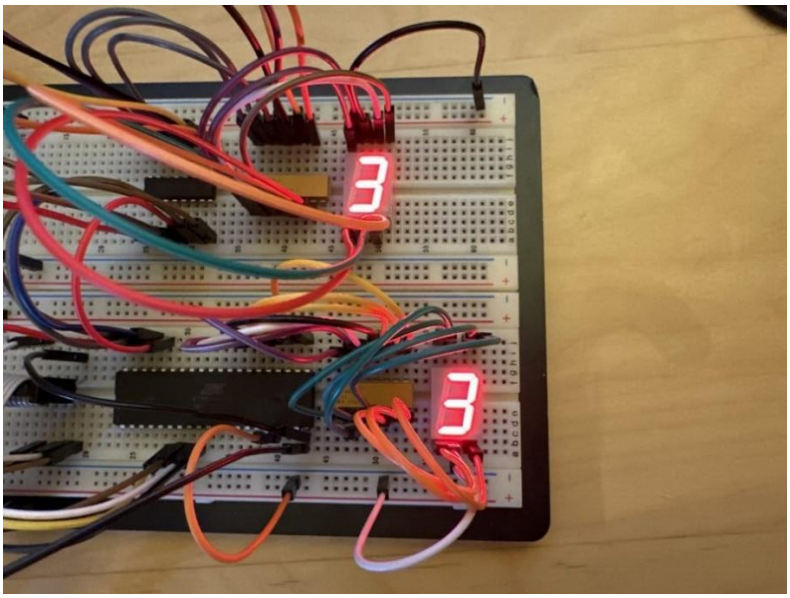


Figure 4: When the MSB is low from PIND7

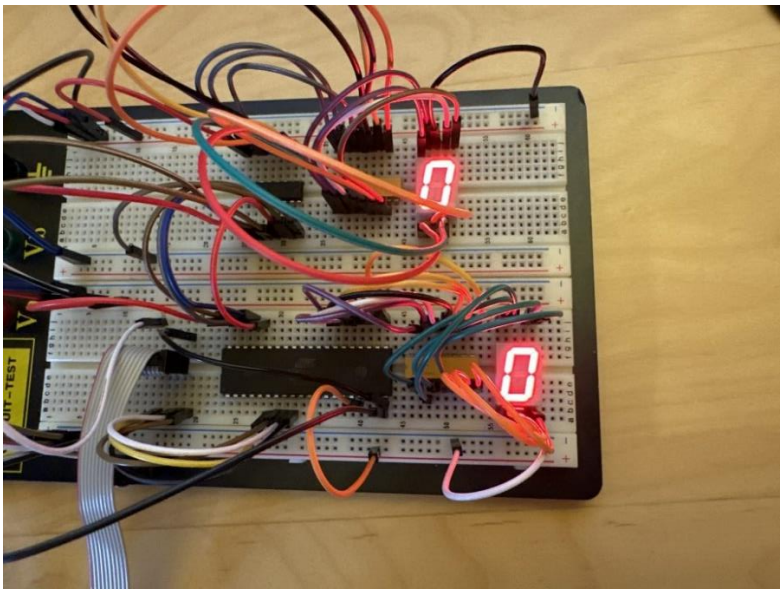


Figure 5: When the MSB is high from PIND7 (resets both counters)

5.0 – Discussion

The first task was to simply create a for loop that counts up to 9. My train of thought was to declare DDRA as outputs and to declare my variable, which is “ledf”. Since the counter starts at 0, “ledf” needs to be initialized to 0 and then count up to 9, so I simply incremented “ledf”. In the for loop, I had to assign PORTA to my previously declared variable because the outputs of DDRA or PORTA are the inputs of the 4511 IC, which is what I was expecting.

For the second task, we had to implement a switch/case inside a do/while loop counter that counts from 0 to 5. I thought that each case in the switch/case represents a sequence of LEDs. I initialized my variable “led” and declared DDRC as outputs. The challenging part was figuring out what HEX values I had to assign to PORTC since I had wired them up in an unorthodox way. I had to determine which pin was associated with each segment from the most significant to the least significant and then convert it from binary into HEX code because each pin from PORTA represented a high or low bit. Aside from that, I had to end my do/while loop by setting it to stop when it was less than or equal to 5, which represents each case statement in the switch/case. In order to reset this counter, I simply added “led = 0” after the while statement, or else it would be stuck at 5 and the first counter would continue counting indefinitely without the second one changing.

For the third task, we had to combine both counters to count from 00 to 59 at one-second intervals. My initial thought was to place it inside the do/while loop, which ended up working because the goal was to combine both counters and have them run indefinitely. Initially, I used the same variable for both counters, but this proved me wrong as it would only count up to 5. I also placed the for loop before the switch, which didn't work because it would start counting to 9 before starting the second counter, not taking into account that C is a procedural language. Therefore, I decided to place the for loop after the switch/case.

Finally, the last task was to add an if statement that detected if the most significant bit of PORTD was high, in which case it would reset both counters. This was the most challenging part for me. Initially, I added the if statement outside the do/while loop, which ended up having no effect because it did not check the state of each counter. To fix this, I simply added it inside the for loop. Continuing, I added a variable "msb" and assigned it to PORTD. I used the "AND" or "&" operator with different HEX values to isolate the most significant bit, which was PIND7. After not finding the required HEX value, I decided to change my method of isolating the bit. With a bit of research, I found that PIND & (1 << PIND7) worked for me. This tells me whether PIND7 is high or low, and if the pin is low, it resets both counters. In essence, 1 << PIND7 shifts the value of 1 to the left by the number of bits specified by PIND7, which is 7. This equates to 128 or 10000000 in binary. PIND just specifies the register that holds the input value of the PORT D pins. Furthermore, I used the "&" bitwise operator to isolate the bit since it works like an AND gate (I need both inputs to be 1 for the output to be 1, otherwise it is 0, as derived from its truth table). However, I realized after I wrote my code that I could have used the HEX code 0x80, which is equivalent to 128 and also isolates the most significant bit.

6.0 – Conclusion

In conclusion, I was able to complete all the tasks. The most time-consuming part was figuring out the HEX code for all tasks (mostly task 2), which was due to unclear wiring on my part. This could have been avoided by conducting more thorough testing of my circuit at home. The second challenge was determining where certain parts of the code needed to be placed. I figured this out the hard way because my circuit/code would simply not work. I overcame this by extensively testing my code and conducting research.

7.0 – Questions

1. Describe how you would change the code in step 3 so that it would be able to either count up or count down. Explain.

You would have to reverse switch case statements like so:

```
switch (led)
{
        case 5: PORTC = 0x73; /*5*/
        _delay_ms(1000);
        break;
        case 4: PORTC = 0x69; /*4*/
        _delay_ms(1000);
        break;
        case 3: PORTC = 0x5B; /*3*/
        _delay_ms(1000);
        break;
        case 2: PORTC = 0x5E; /*2*/
        _delay_ms(1000);
        break;
        case 1: PORTC = 0x09; /*1*/
        _delay_ms(1000);
        break;
        case 0: PORTC = 0x3F; /*0*/
        _delay_ms(1000);
        break;
}
```

Then change the led initialization from 0 to 5 because you would want it to start at 5 instead of 0 like so :

```
int led = 5;
```

Afterwards to reverse the actually sequence you have to change the while condition to while (led >= 0); because you want the loop to stop when the case 0 condition has been reached.

Lastly, we can add the if statement from part 4 and add an else. If PIND7 is high, we would count in normally or else it just counts in reverse. The thing is that my counter only changes direction when the first count sequence is done. For example, for it to count normally I would have to wait 59 to 00 sequence to finish and vice-versa. This is because C is procedural and needs to finish what it is doing before it can do anything else, and it goes from top to bottom.

```c
/*
 * Lab6Question.c
 *
 * Created: 3/11/2024 12:05:10 PM
 * Author : 2237458
 */


#include <avr/io.h>
#define F_CPU 1000000
#include <util/delay.h>

int main(void)
{
        // Variable Init
        int led = 5;
        int ledf;
        DDRC = 0xFF;
        DDRA = 0xFF;

while (1)
{
        if (PIND & (1<<PIND7))
        {
            do
                {
                    /* loop that counts from 0-5 with switch/case */
                    switch(led)
                    {
                        case 0 : PORTC = 0x3F; /*0*/
                        _delay_ms(1000);
                        break;
                        case 1 : PORTC = 0x09; /*1*/
                        _delay_ms(1000);
                        break;
                        case 2 : PORTC = 0x5E; /*2*/
                        _delay_ms(1000);
                        break;
                        case 3 : PORTC = 0x5B; /*3*/
                        _delay_ms(1000);
                        break;
                        case 4 : PORTC = 0x69; /*4*/
                        _delay_ms(1000);
                        break;
                        case 5 : PORTC = 0x73; /*5*/
                        _delay_ms(1000);
                        break;
                    }
                    // for loop that counts from 0-9
                    for(ledf = 0; ledf <= 9; ledf++)
                    {
                        PORTA = ledf;
                        _delay_ms(1000);

                    }
                    led++;
                } while (led <= 5);
                // Restarts the counter
                led = 0;
        }
        else
        {
                do
                    {
                    switch (led)
                    {
                        case 5: PORTC = 0x73; /*5*/
                            _delay_ms(1000);
                            break;
```

```
                    case 4: PORTC = 0x69; /*4*/
                        _delay_ms(1000);
                        break;
                    case 3: PORTC = 0x5B; /*3*/
                        _delay_ms(1000);
                        break;
                    case 2: PORTC = 0x5E; /*2*/
                        _delay_ms(1000);
                        break;
                    case 1: PORTC = 0x09; /*1*/
                        _delay_ms(1000);
                        break;
                    case 0: PORTC = 0x3F; /*0*/
                        _delay_ms(1000);
                        break;
                }
                    // for loop that counts from 0-9
                    for (ledf = 9; ledf >= 0; ledf--)
                    {
                        PORTA = ledf;
                        _delay_ms(1000);
                    }
                    led--;
                } while (led >= 0);
            // Restarts the counter
            led = 5;
        }


    }

}
```