Temple University

College of Engineering

Department of Electrical and Computer Engineering (ECE)

Student Lab Report Cover Page

Course Number : 3613

Course Section : 001 / 002

Experiment # : Lab # 7

Student Name (print) : Von Kaukeano

TUid# : 915596703

Date : 10/17/19

Grade : ______ **/100**

TA Name : Sung Choi

Introduction

This lab uses assembly language to operate the IO ports on the Atmel 324PB.Our goal is to control IO and learn the bit operation of the ports on the board. We will also learn and implement assembly instructions such as SBIC and SBIS to continuously poll the input switches. We will be using common techniques such as PWM with the assembly language that executes it and apply it to the IO ports. The lab also shows how to use the seven segment decoder and the pins that represent each segment.

Procedure

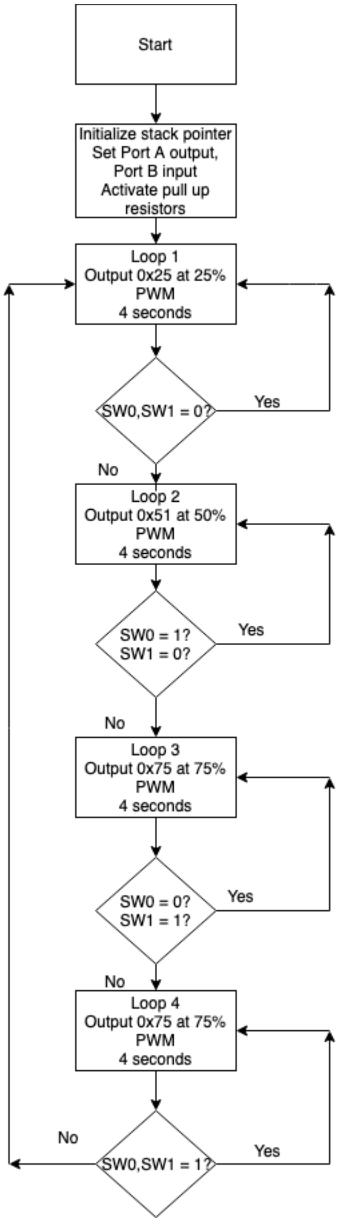
Activity 1

- Read 2 switch inputs from PORTB – PB0 and PB1
- Compare the input combination to the four cases given in Table 1.
- Make the specified outputs from Table 1 through PORTB using 8 LEDs.
    - The LED outputs show the specified duty cycles for each case.
    - Use the time delay subroutine.
- Infinite loop until the input changes.

PORT B Input		**PORT A Output**	
SW1	**SW0**	**PWM**	**LED Value**
0	0	25% duty cycle Period: **4 seconds**	0x25 or 0b00100101
0	1	50% duty cycle Period: **4 seconds**	0x51 or 0b01010001
1	0	75% duty cycle Period: **4 seconds**	0x75 or 0b01110101
1	1	50% duty cycle Period: **1 seconds**	0x51 or 0b01010001

Table 1: Activity 1 Desired Input/Output

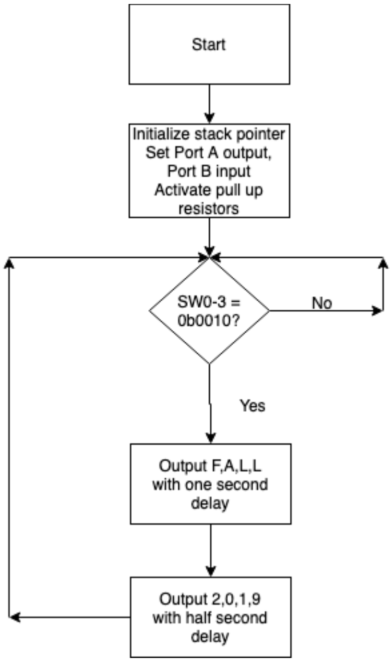
Beginning activity 1, I began by drawing my flowchart. The code needs to continuously check for the input of the switches and activate the PWM based on the logic of the switches. For example, if both switches are low then it will output 25 on the LED's but at a PWM of 25% duty cycle for 4 seconds. To achieve these PWM's we needed to have a two delay subroutines. A subroutine for a full second and one for a half second so the first output would be on for one second and off for three seconds having a 25% duty cycle for four seconds. We then measured the output at PortA0 to see the results.



Activity 2

- Make the outputs on the 7-segment Display – ‘F’,’A’,’L’,’L’,’2’,’0’,’1’,’9’ each letter and each number at a time .
- Fill out table to show each letter and number with the corresponding binary value and segment light on and off.
- Fill out table with result of displaying letter and number on the 7-segment display and time delay.

Beginning activity 2 I created a flowchart to get my ideas out. This code also needs to continuously check the input for switch to be 0b0010 which maps to SW2 being active for the code to execute. I made directives for each of the values to display on the seven segment display and mapped them to their binary values. I made a loop to check the input on the switches. Once the correct input is implemented I then made another loop to output each value. In order to get a one second delay I placed a delay before output of the value and then output zero value to clear the bits for every character displayed. The same for the numbers except it was done at a half second delay.



Results

<u>Activity 1</u>

Below are the screenshots of the PWM with their respective inputs. I did a print screen and pasted in paint and cut off the duty cycle percentage that was at the bottom left corner.

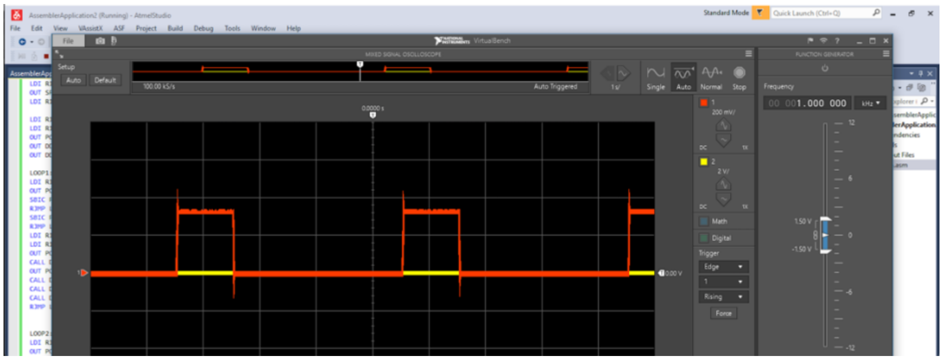
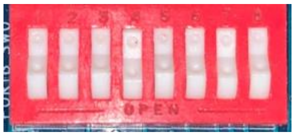
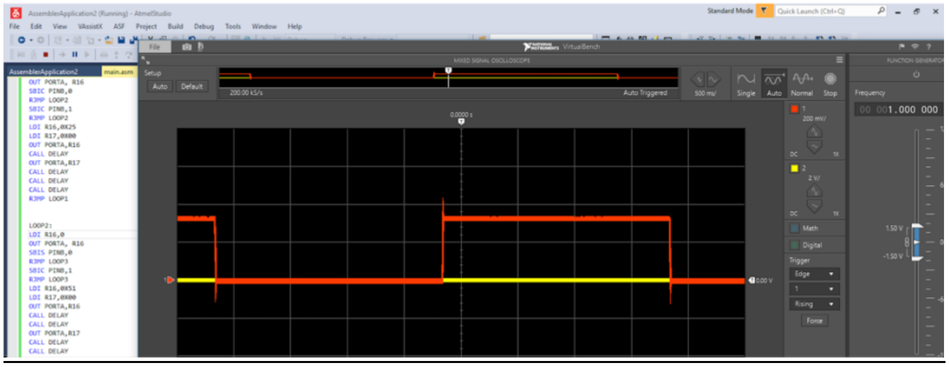


Figure 1: PWM 25% at 4 seconds

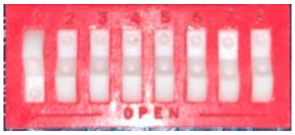


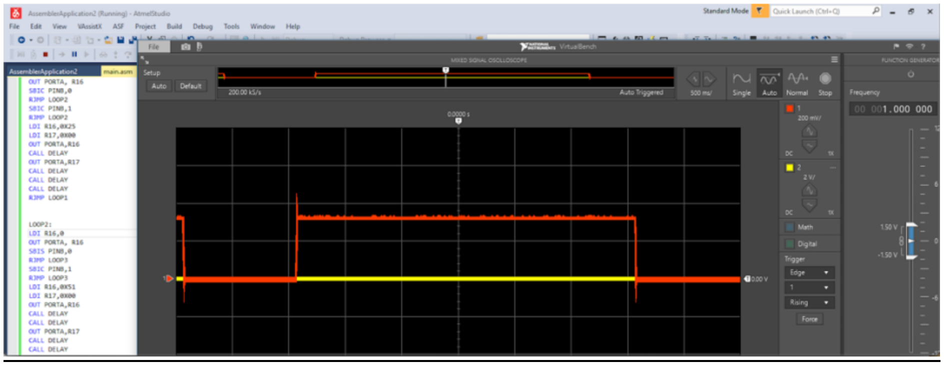


```
OUT PORTA, R16
SBIC PINB,0
RJMP LOOP2
SBIC PINB,1
RJMP LOOP2
LDI R16,0X25
LDI R17,0X00
OUT PORTA,R16
CALL DELAY
OUT PORTA,R17
CALL DELAY
CALL DELAY
CALL DELAY
RJMP LOOP1


LOOP2:
LDI R16,0
OUT PORTA, R16
SBIS PINB,0
RJMP LOOP3
SBIC PINB,1
RJMP LOOP3
LDI R16,0X51
LDI R17,0X00
OUT PORTA,R16
CALL DELAY
CALL DELAY
OUT PORTA,R17
CALL DELAY
CALL DELAY
```

Figure 2: PWM 50% at 4 seconds

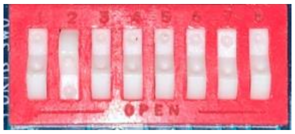


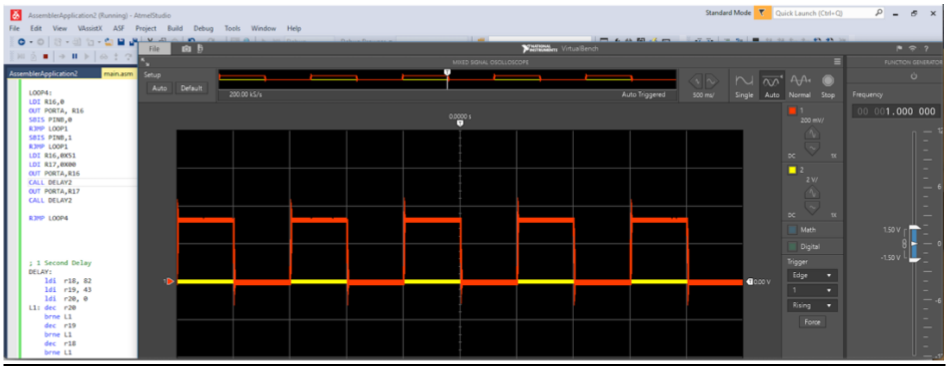


```
OUT PORTA, R16
SBIC PINB,0
RJMP LOOP2
SBIC PINB,1
RJMP LOOP2
LDI R16,0X25
LDI R17,0X00
OUT PORTA,R16
CALL DELAY
OUT PORTA,R17
CALL DELAY
CALL DELAY
CALL DELAY
RJMP LOOP1


LOOP2:
LDI R16,0
OUT PORTA, R16
SBIS PINB,0
RJMP LOOP3
SBIC PINB,1
RJMP LOOP3
LDI R16,0X51
LDI R17,0X00
OUT PORTA,R16
CALL DELAY
CALL DELAY
OUT PORTA,R17
CALL DELAY
CALL DELAY
```

Figure 3: PWM 75% at 4 seconds





```
LOOP4:
LDI R16,0
OUT PORTA, R16
SBIS PINB,0
RJMP LOOP1
SBIS PINB,1
RJMP LOOP1
LDI R16,0X51
LDI R17,0X00
OUT PORTA,R16
CALL DELAY2
OUT PORTA,R17
CALL DELAY2

RJMP LOOP4



; 1 Second Delay
DELAY:
    ldi  r18, 82
    ldi  r19, 43
    ldi  r20, 0
L1: dec  r20
    brne L1
    dec  r19
    brne L1
    dec  r18
    brne L1
```

Figure 4: PWM 50% at 1 second

https://drive.google.com/open?id=1Q4VfbGoD_frZDqVBZ-rKxg6TUvLNOIJ2

Activity 2

Port Bits	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Segments	DP	G	F	E	D	C	B	A
Letter ‘F’	off	on	on	on	off	off	off	On
Binary (CC)	0	1	1	1	0	0	0	1
Letter ‘A’	off	off	on	on	off	on	on	on
Binary (CC)	0	0	1	1	0	1	1	1
Letter ‘L’	off	off	on	on	on	off	off	off
Binary (CC)	0	0	1	1	1	0	0	0
Letter ‘L’	off	off	on	on	on	off	off	off
Binary (CC)	0	0	1	1	1	0	0	0
Letter ‘2’	off	on	off	on	on	off	on	on
Binary (CC)	0	1	0	1	1	0	1	1
Letter ‘0’	off	off	on	on	on	on	on	on
Binary (CC)	0	0	1	1	1	1	1	1
Letter ‘1’	off	off	off	off	off	on	on	off
Binary (CC)	0	0	0	0	0	1	1	0
Letter ‘9’	off	on	on	off	off	on	on	on
Binary (CC)	0	1	1	0	0	1	1	1

Table 2: 7-Segment Display Operation

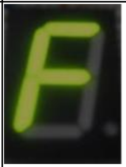
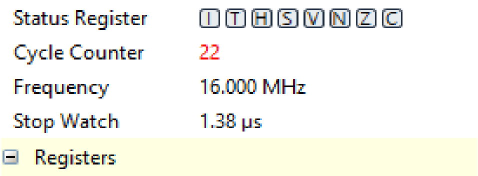
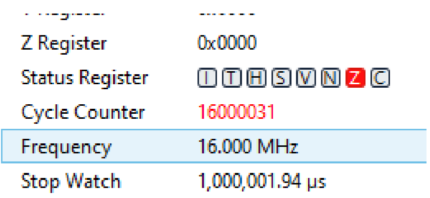
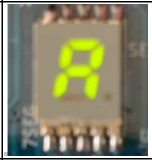
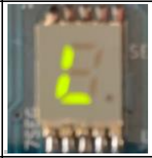
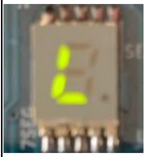
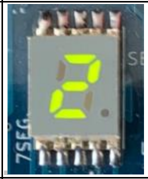
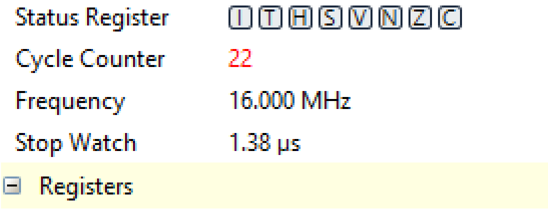
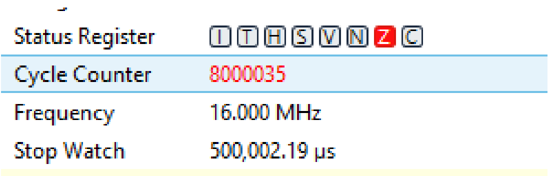
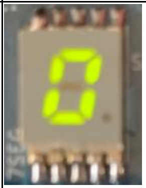
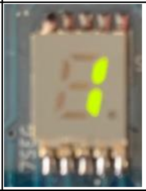
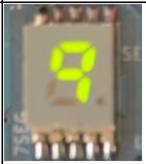
Output	Output PORTA	7-Segment Display view	**Stopwatch view** (only show one delay execution for letter and one delay execution for number)
F	**01110001**		**For Letter,** **Stopwatch reading initial:**  Status Register I T H S V N Z C Cycle Counter 22 Frequency 16.000 MHz Stop Watch 1.38 µs ⊟ Registers **Stopwatch reading final:**  Z Register 0x0000 Status Register I T H S V N Z C Cycle Counter 16000031 Frequency 16.000 MHz Stop Watch 1,000,001.94 µs Use **one of the letters simulation** time delay 1 second
A	**01110111**		
L	**00111000**		
L	**00111000**		
2	**01011011**		**For Number:** **Stopwatch reading initial:**  Status Register I T H S V N Z C Cycle Counter 22 Frequency 16.000 MHz Stop Watch 1.38 µs ⊟ Registers **Stopwatch reading final:**  Status Register I T H S V N Z C Cycle Counter 8000035 Frequency 16.000 MHz Stop Watch 500,002.19 µs Use **one of the numbers simulation** time delay 0.5 second
0	**00111111**		
1	**00000110**		
9	**01100111**		

Table 3:7-Segment Display Result and Time Delay

https://drive.google.com/open?id=1ZfELhRm_lj6oEZbQLsN_B_XSDmIjnXy5

Discussion

In Activity 1 we displayed values at a specific PWM. On the board the correct LED’s light up to the corresponding numbers. For example, 0x25 is 0b00100101 which the LED’s light up in that order. Secondly, the screenshots of the PWM on the oscilloscope show that at each input the PWM is at the correct percentage. By using one, one second delay and outputting the display then outputting zero to clear the bits ensure a constant one second delay.

In Activity 2 we displayed ‘F’,’A’,’L’,’L’,’2’,’0’,’1’,’9’. To verify this we output it onto the ATMEL board. We mapped the binary values to the seven segment decoder and placed a one second time delay between each letter and half second between the numbers. This can be seen by the stop watch timer values and in the video the numbers move through quicker than the letters.

Conclusion

This lab was a great way to understand how the IO ports on a microcontroller work. The implementation of the seven segment decoder was interesting in learning because in digital we did a similar activity but were common cathode. The PWM was a also an important topic in learning because I have used PWM on a raspberry pi and never had to used delayed in order to create the signal. The delays are what caused the duty cycle. Finally the assembly code such as SBIS and SBIC that polled the input bits are a significant in real world applications that I enjoy learning about.

Appendix

Activity 1

```
LDI R16,LOW(RAMEND)
OUT SPL,R16
LDI R17,HIGH(RAMEND)

LDI R16,$00
LDI R17,$FF
OUT PORTB , R17
OUT DDRA,R17
OUT DDRB,R16

LOOP1:
LDI R16,0
OUT PORTA, R16
SBIC PINB,0
RJMP LOOP2
SBIC PINB,1
RJMP LOOP2
LDI R16,0X25
LDI R17,0X00
OUT PORTA,R16
CALL DELAY
OUT PORTA,R17
CALL DELAY
CALL DELAY
CALL DELAY
RJMP LOOP1


LOOP2:
LDI R16,0
OUT PORTA, R16
SBIS PINB,0
RJMP LOOP3
SBIC PINB,1
RJMP LOOP3
LDI R16,0X51
LDI R17,0X00
OUT PORTA,R16
CALL DELAY
CALL DELAY
OUT PORTA,R17
CALL DELAY
CALL DELAY

RJMP LOOP2

LOOP3:
LDI R16,0
OUT PORTA, R16
SBIC PINB,0
RJMP LOOP4
SBIS PINB,1
RJMP LOOP4
LDI R16,0X75
LDI R17,0X00
OUT PORTA,R16
CALL DELAY
CALL DELAY
CALL DELAY
OUT PORTA,R17
CALL DELAY

RJMP LOOP3

LOOP4:
LDI R16,0
OUT PORTA, R16
SBIS PINB,0
RJMP LOOP1
SBIS PINB,1
RJMP LOOP1
LDI R16,0X51
LDI R17,0X00
OUT PORTA,R16
CALL DELAY2
OUT PORTA,R17
CALL DELAY2

RJMP LOOP4

; 1 Second Delay
DELAY:
    ldi  r18, 82
    ldi  r19, 43
    ldi  r20, 0
L1: dec  r20
    brne L1
    dec  r19
    brne L1
    dec  r18
    brne L1
    lpm
    nop
       RET


; HALF SECOND DELAY
DELAY2:
    ldi  r18, 41
    ldi  r19, 150
    ldi  r20, 128
L2: dec  r20
    brne L1
    dec  r19
    brne L1
    dec  r18
    brne L2
       RET
```

Activity 2

```
.EQU ZERO = 0B00111111
.EQU ONE = 0B00000110
.EQU TWO = 0B01011011
.EQU NINE = 0B01100111
.EQU F = 0B01110001
.EQU A = 0B01110111
.EQU L = 0B00111000


LDI R16,LOW(RAMEND)
OUT SPL,R16
LDI R17,HIGH(RAMEND)

LDI R16,$00
LDI R17,$FF
```

```
OUT PORTB , R17
OUT DDRA,R17
OUT DDRB,R16


LOOP:
LDI R16,0
OUT PORTA, R16
SBIC PINB,0
RJMP LOOP
SBIC PINB,1
RJMP LOOP
SBIS PINB,2
RJMP LOOP
SBIC PINB,3
RJMP LOOP

RJMP LOOP1

LOOP1:
LDI R16,F
LDI R17,0
OUT PORTA,R16
CALL DELAY
OUT PORTA,R17


LDI R16,A
OUT PORTA,R16
CALL DELAY
OUT PORTA,R17

LDI R16,L
OUT PORTA,R16
CALL DELAY
OUT PORTA,R17

LDI R16,L
OUT PORTA,R16
CALL DELAY2
OUT PORTA,R17

LDI R16,TWO
OUT PORTA,R16
CALL DELAY2
OUT PORTA,R17

LDI R16,ZERO
OUT PORTA,R16
CALL DELAY2
OUT PORTA,R17

LDI R16,ONE
OUT PORTA,R16
```

```
CALL DELAY2
OUT PORTA,R17

LDI R16,NINE
OUT PORTA,R16
CALL DELAY2
OUT PORTA,R17

RJMP LOOP

; 1 Second Delay
DELAY:
    ldi  r18, 82
    ldi  r19, 43
    ldi  r20, 0
L1: dec  r20
    brne L1
    dec  r19
    brne L1
    dec  r18
    brne L1
    lpm
    nop
       RET


; HALF SECOND DELAY
DELAY2:
    ldi  r18, 41
    ldi  r19, 150
    ldi  r20, 128
L2: dec  r20
    brne L1
    dec  r19
    brne L1
    dec  r18
    brne L2
       RET
```