Temple University

College of Engineering

Department of Electrical and Computer Engineering (ECE)

Student Lab Report Cover Page

Course Number : 3613

Course Section : 002

Experiment # : Lab #11

Student Name (print) : Robert Bara

TUid# : 915614617

Date : 11/11/2020

Grade : _____ **/100**

TA Name : Sung Choi

ACTIVITIES:

Important: Extra 5 points deduction for each case will be applied for every activities if

- the comments are not enough to explain the code,
- you use screenshot for code

Activity 1

1. Code and Description

(1) Code (Full-Comment)

```
//Activity 1: Robert Bara
//10 microsec squarewave timer 50% duty cycle
.INCLUDE "M324pbDEF.INC"
LDI R16,0xFF
OUT DDRA,R16
BEGIN:LDI R20,-78
OUT TCNT0,R20;load timer0
LDI R20,0b00000101 ;normal mode, prescaled by 1024
OUT TCCR0B,R20 ;Timer0,Normal mode, int clk
AGAIN:IN R20,TIFR0;read TIFR
SBRS R20,0 ;Skip if Bit in Register is set if TOV0 is set skip next inst.
RJMP AGAIN
LDI R20,0x0
OUT TCCR0B,R20;stop Timer0
LDI R20,(1<<TOV0);R20 = 0x01
OUT TIFR0,R20;clear TOV0 flag
EOR R17,R16;toggle A5 of R17
OUT PORTA,R17;toggle PA5
   RJMP BEGIN
```

2. Result

(1) Calculations for timing signal.

```
%Time calculations
%Activity 1:
%divide square wave to get half of the period and divide by the clock's
%period multiplied by the prescaler
Tsig=1; %half the signal because of square wave
CT=0.0625e-6; %clock period
A=(Tsig)/(CT*1024) %must be <255
```

```
A = 15625
```

```
%B=A/1024 %Prescale if it is too big, put into microsoft calc and use hex value for tcc
%subtract from total ticks
Tcnt=65535-B %Plug into microsoft calculator to get hex value
```

```
Tcnt = 49910
```

Figure 1. Time calculations for activity 1

(2) Your result (Data Visualizer reading) to show the signal "on" and "off" times

➔ Required to show the on-pulse length and one-period length in the result)

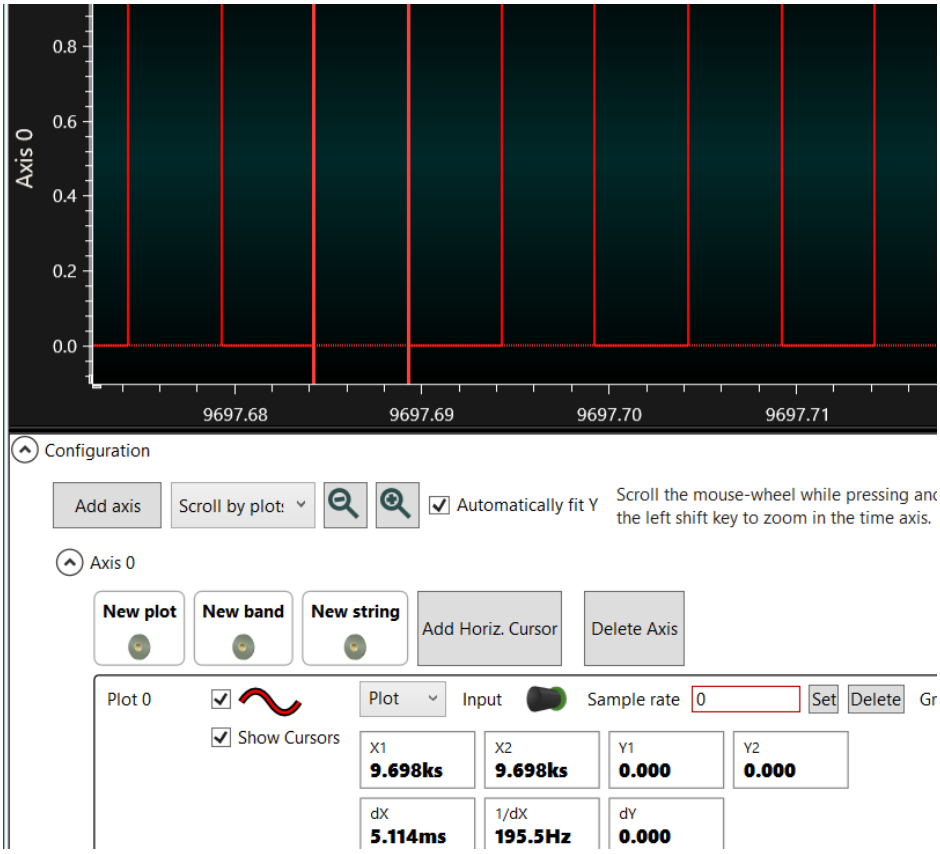


Figure 2. 50% Duty cycle on for 5ms

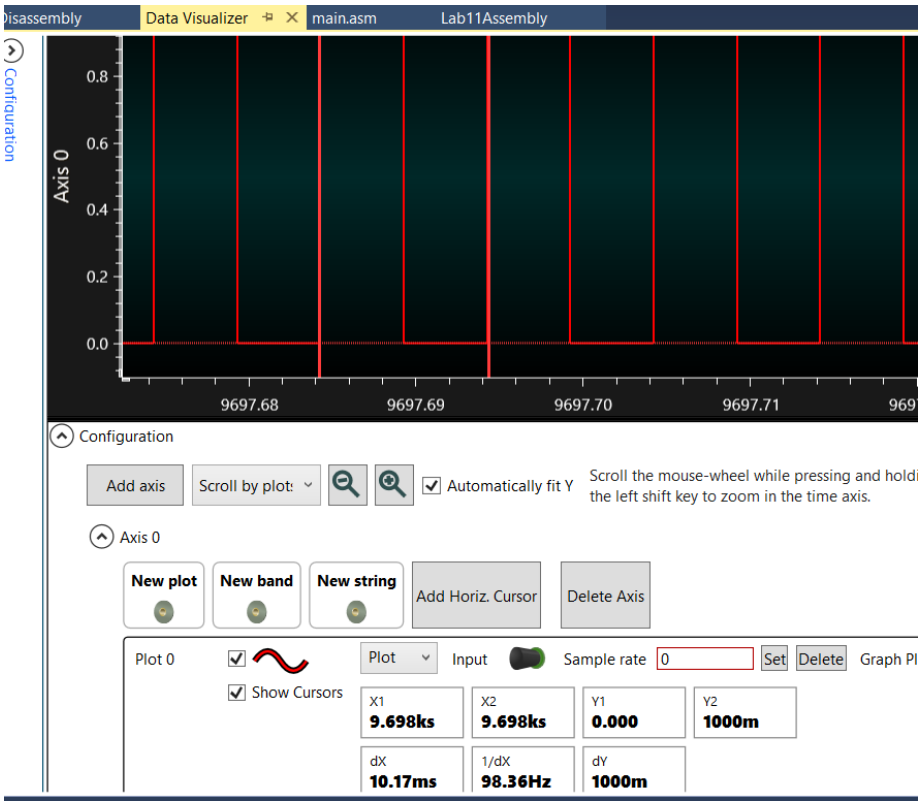


Figure 3. One complete Period, 10ms

Activity 2

1. Code and Description

(1) Code (Full-Comment)

```
// Activity 2
 .INCLUDE "M324pbDEF.INC"
LDI  R16,HIGH(RAMEND)
OUT  SPH,R16
LDI  R16,LOW(RAMEND)
OUT  SPL,R16  ;initialize stack pointer
LDI R16,0XFF
OUT  DDRA,R16 ;PORTA as an output
MAIN: LDI R16,0XFF
OUT PORTA,R16
CALL DELAY1
LDI R16,0X00
OUT PORTA,R16
CALL DELAY2
RJMP MAIN

//on for 2.5ms pulse
DELAY1:LDI  R20,0
OUT  TCNT0,R20  ;TCNT0 = 0
LDI  R20,39
OUT  OCR0B,R20  ;OCR0 = 62 (2.5ms pulse)
```

```
LDI  R20, 0b00000101 ;Timer0, normal mode, prescaler=1024
OUT  TCCR0B,R20  ;Timer0, normal mode, prescaler = 1024
AGAIN1:IN  R20,TIFR0  ;read TIFR
SBRS  R20,OCF0B  ;if OCF0 is set skip next instruction
RJMP  AGAIN1
LDI  R20,0x0
OUT  TCCR0B,R20  ;stop Timer0
LDI  R20,1<<OCF0B
OUT  TIFR0,R20  ;clear OCF0 flag
RET

//off for 7.5ms pulse
DELAY2:LDI  R20,0
OUT  TCNT0,R20  ;TCNT0 = 0
LDI  R20,117
OUT  OCR0B,R20  ;OCR0 = 117 (7.5ms pulse)
LDI  R20,0b00000101 ;Timer0, normal mode, prescaler = 1024
OUT  TCCR0B,R20  ;Timer0,  mode, prescaler = 1024
AGAIN2:IN  R20,TIFR0  ;read TIFR
SBRS  R20,OCF0B  ;if OCF0 is set skip next instruction
RJMP  AGAIN2
LDI  R20,0x0
OUT  TCCR0B,R20  ;stop Timer0
LDI  R20,1<<OCF0B
OUT  TIFR0,R20  ;clear OCF0 flag
  RET
```

2. Result

(1) Calculations for timing signal.

```
%Time calculations
%Activity 2:
%When it is on for 2.5ms
Tsig=2.5e-3;
CT=0.0625e-6; %clock period
A=(Tsig)/(CT) %must be <255
```

```
A = 40000
```

```
B=A/1024 %Prescale if it is too big, put into microsoft calc and use hex value for tcc
```

```
B = 39.0625
```

```
%subtract from total ticks
Tcnt=256-B %Plug into microsoft calculator to get hex value
```

```
Tcnt = 216.9375
```

```
%When it is on for 7.5ms
Tsigoff=7.5e-3;
A2=(Tsigoff)/(CT) %must be <255
```

```
A2 = 120000
```

```
B2=A2/1024 %Prescale if it is too big, put into microsoft calc and use hex value for tcc
```

```
B2 = 117.1875
```

```
%subtract from total ticks
Tcnt_off=256-B2 %Plug into microsoft calculator to get hex value
```

```
Tcnt_off = 138.8125
```

Figure 4. Time calculations for activity 2

(2) Your result (Data Visualizer reading) to show the signal “on” and “off” times.

➔ Required to show the on-pulse length and one-period length in the result)

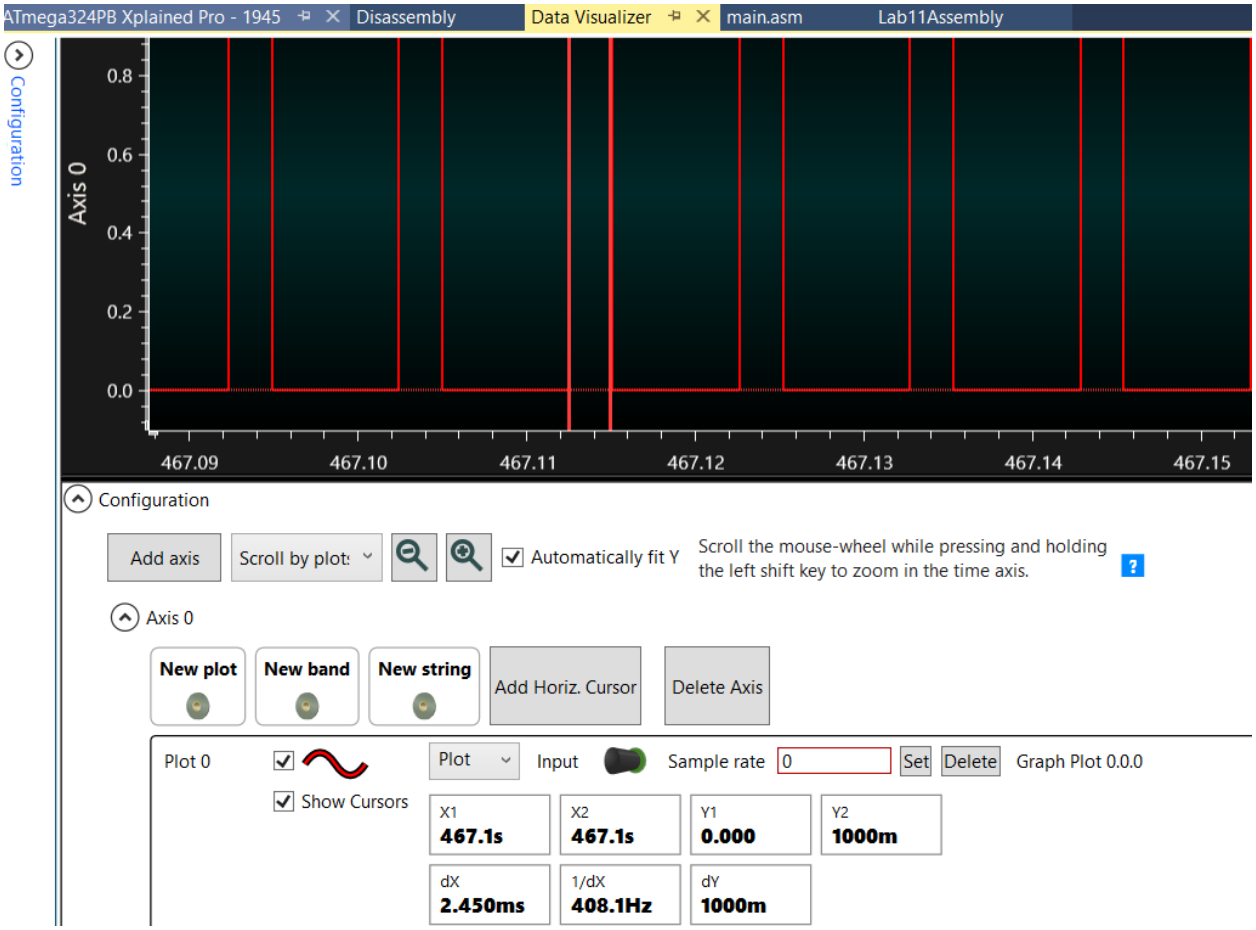


Figure 5. Time calculations for 2.5ms on

Figure 6. Activity 2 Full period

Activity 3

1. Code and Description

(1) Code (Full-Comment)

```
// Activity 3, CTC mode
 .INCLUDE "M324pbDEF.INC"
LDI   R16,HIGH(RAMEND)
OUT   SPH,R16
LDI   R16,LOW(RAMEND)
```

```
OUT  SPL,R16  ;initialize stack pointer
LDI R16,0XFF
OUT  DDRA,R16 ;PORTA as an output
MAIN: LDI R16,0XFF
OUT PORTA,R16
CALL DELAY1
LDI R16,0X00
OUT PORTA,R16
CALL DELAY2
RJMP MAIN

//on for 7.5ms pulse
DELAY1:LDI  R20,0
OUT  TCNT0,R20  ;TCNT0 = 0
LDI  R20,117
OUT  OCR0B,R20  ;OCR0 = 117 (7.5ms pulse)
LDI  R20,0b00001101 ;Timer0, CTC mode, prescaler = 1024
OUT  TCCR0B,R20  ;Timer0, CTC mode, prescaler = 1024
AGAIN1:IN  R20,TIFR0  ;read TIFR
SBRS  R20,OCF0B  ;if OCF0 is set skip next instruction
RJMP  AGAIN1
LDI  R20,0x0
OUT  TCCR0B,R20  ;stop Timer0
LDI  R20,1<<OCF0B
OUT  TIFR0,R20  ;clear OCF0 flag
RET

//off for 2.5ms pulse
DELAY2:LDI  R20,0
OUT  TCNT0,R20  ;TCNT0 = 0
LDI  R20,39
OUT  OCR0B,R20  ;OCR0 = 39 (2.5ms pulse)
LDI  R20, 0b00001101 ;Timer0, CTC mode, prescaler=1024
OUT  TCCR0B,R20  ;Timer0, CTC mode, prescaler = 1024
AGAIN2:IN  R20,TIFR0  ;read TIFR
SBRS  R20,OCF0B  ;if OCF0 is set skip next instruction
RJMP  AGAIN2
LDI  R20,0x0
OUT  TCCR0B,R20  ;stop Timer0
LDI  R20,1<<OCF0B
OUT  TIFR0,R20  ;clear OCF0 flag
  RET
```

2. Result

(1) Calculations for timing signal.

The calculations for activity 2 are the same as activity 3 with the exception that the values loaded are flipped between the two delay subroutines. I used the same matlab code I generated for activity 2.

(2) Your result (Data Visualizer reading) to show the signal “on” and “off” times.

➔ Required to show the on-pulse length and one-period length in the result)

Figure 7. On for 7.5ms

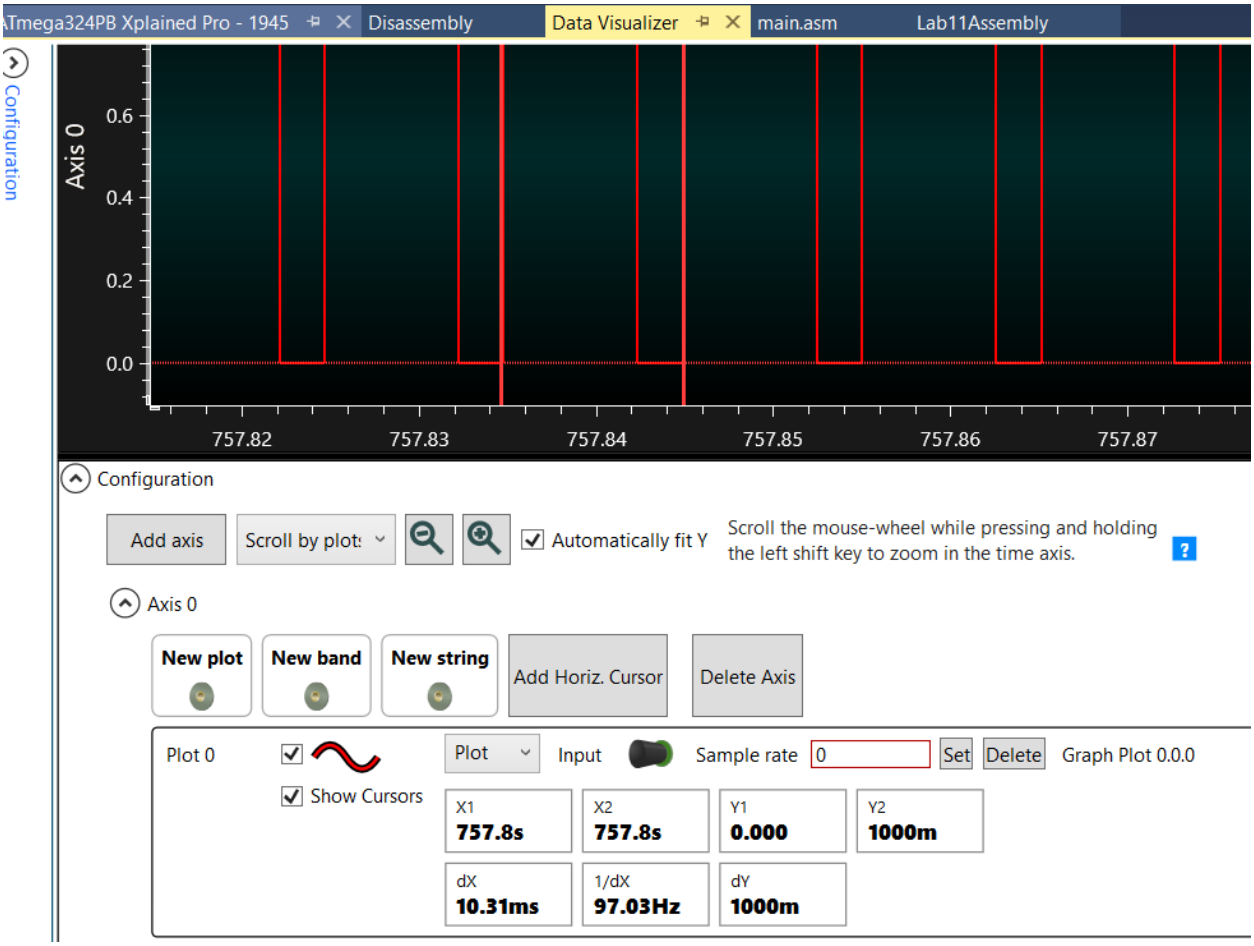


Figure 8. One complete period

Activity 4

1. Code and flowchart

(1) Code (Full-Comment)

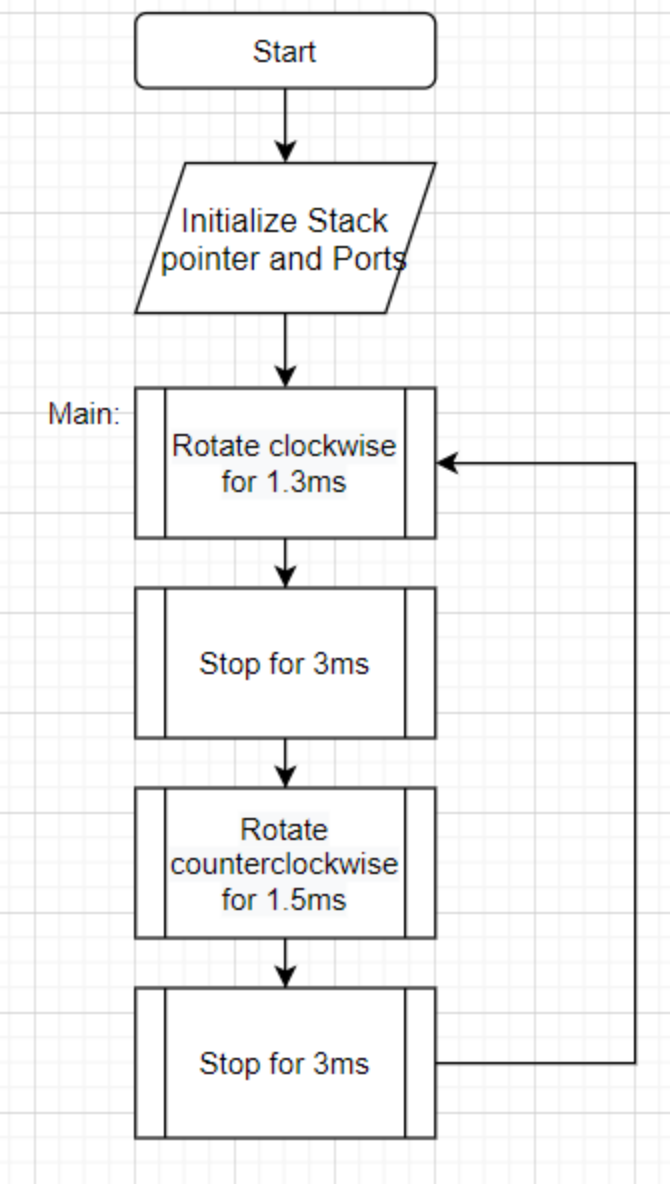
```
//Robert Bara Activity 4
.INCLUDE "M324pbDEF.INC"
#define F_CPU 16000000UL
LDI R16,HIGH(RAMEND);init stack pointer
OUT SPH,R16
LDI R16,LOW(RAMEND)
OUT SPL,R16
SBI DDRD,2;PD2 as an output
LDI R30,150

//Rotate clockwise for 3 seconds
```

```
ClockwiseRot:
SBI PORTD,2;PD2 = 1
RCALL DELAY_1_3ms //1.3ms timer
CBI PORTD,2;PD2 = 0
RCALL DELAY_20ms
DEC R30
BRNE ClockwiseRot    ;restart if r30 if not equal to 0
LDI R30,150

//Stopping for 3 sec
Stop1:
SBI PORTD,2;PD2 = 1
RCALL DELAY_1_5ms //1.5ms timer
CBI PORTD,2;PD2 = 0
RCALL DELAY_20ms
DEC R30
BRNE Stop1           ;restart at stop1
LDI R30,150

//Rotate counterclockwise for 3 sec
CCW:
SBI PORTD,2;PD2 = 1
RCALL DELAY_1_7ms
CBI PORTD,2;PD2 = 0
RCALL DELAY_20ms
DEC R30
BRNE CCW
LDI R30,150

//Stopping for 3 seconds
Stop2:
SBI PORTD,2;PD2 = 1
RCALL DELAY_1_5ms
CBI PORTD,2;PD2 = 0
RCALL DELAY_20ms
DEC R30
BRNE Stop2

//Clockwise 1.3ms delay
DELAY_1_3ms:
LDI R20,0xAE
STS 0X0085,R20
;OUTTCNT1H,R20;TEMP = 0xD8
LDI R20,0xBF
STS 0X0084,R20
;OUTTCNT1L,R20;TCNT1L = 0xF0, TCNT1H = TEMP
LDI R20,0x0
sts 0x0080,r20
;OUTTCCR1A,R20;WGM11:10=00
LDI R20,0x1
sts 0x0081,r20
;OUTTCCR1B,R20;WGM13:12=00,CS=CLK
AGAIN1:
lds R20,0X0036
;IN R20,TIFR1;read TIFR1
SBRS R20,TOV1;if OCF1A is set skip next instruction
RJMP AGAIN1
LDI R20,1<<TOV1
```

```
STS 0X0036,R20
;OUTTIFR,R20;clear TOV1 flag
LDI R19,0
sts 0x0080,r19
;OUTTCCR1B,R19;stop timer
sts 0x0081,r19
;OUTTCCR1A,R19;
RET

//Counter clockwise
DELAY_1_7ms:
LDI R20,0x95
STS 0X0085,R20
;OUTTCNT1H,R20;TEMP = 0xD8
LDI R20,0xBF
STS 0X0084,R20

;OUTTCNT1L,R20;TCNT1L = 0xF0, TCNT1H = TEMP
LDI R20,0x0
sts 0x0080,r20
;OUTTCCR1A,R20;WGM11:10=00
LDI R20,0x1
sts 0x0081,r20
;OUTTCCR1B,R20;WGM13:12=00,CS=CLK
AGAIN2:
lds R20,0X0036
;IN R20,TIFR1;read TIFR1
SBRS R20,TOV1;if OCF1A is set skip next instruction
RJMP AGAIN2
LDI R20,1<<TOV1
STS 0X0036,R20
;OUTTIFR,R20;clear TOV1 flag
LDI R19,0
sts 0x0080,r19
;OUTTCCR1B,R19;stop timer
sts 0x0081,r19
;OUTTCCR1A,R19;
RET

//stop 1.5ms
DELAY_1_5ms:
LDI R20,0xA2
STS 0X0085,R20
;OUTTCNT1H,R20;TEMP = 0xD8
LDI R20,0xA3
STS 0X0084,R20
;OUTTCNT1L,R20;TCNT1L = 0xF0, TCNT1H = TEMP
LDI R20,0x0
sts 0x0080,r20
;OUTTCCR1A,R20;WGM11:10=00
LDI R20,0x1
sts 0x0081,r20
;OUTTCCR1B,R20;WGM13:12=00,CS=CLK
AGAIN3:
lds R20,0X0036
;IN R20,TIFR1;read TIFR1
SBRS R20,TOV1;if OCF1A is set skip next instruction
RJMP AGAIN3
```

```
LDI R20,1<<TOV1
STS 0X0036,R20
;OUTTIFR,R20;clear TOV1 flag
LDI R19,0
sts 0x0080,r19
;OUTTCCR1B,R19;stop timer
sts 0x0081,r19
;OUTTCCR1A,R19;
RET

DELAY_20ms:
LDI R20,0xEC
STS 0X0085,R20
;OUTTCNT1H,R20;TEMP = 0xD8
LDI R20,0x77
STS 0X0084,R20
;OUTTCNT1L,R20;TCNT1L = 0xF0, TCNT1H = TEMP
LDI R20,0x0
sts 0x0080,r20
;OUTTCCR1A,R20;WGM11:10=00
LDI R20,0x03
sts 0x0081,r20
;OUTTCCR1B,R20;WGM13:12=00,CS=CLK
AGAIN4:
lds R20,0X0036
;IN R20,TIFR1;read TIFR1
SBRS R20,TOV1;if OCF1A is set skip next instruction
RJMP AGAIN4
LDI R20,1<<TOV1
STS 0X0036,R20
;OUTTIFR,R20;clear TOV1 flag
LDI R19,0
sts 0x0080,r19
;OUTTCCR1B,R19;stop timer
sts 0x0081,r19
;OUTTCCR1A,R19;
  RET
```

(2)Flowchart



2. Result

(1) Calculations for timing signal

I referred to the chart from Dr H. PowerPoint for the calculations.

Pulse width	calculation	# of ticks	Normal mode count	TCNT1H:TCNT1L
1.3msec	(1.3msec/.0625µsec)	20,800	65535-20800=44735	$AE:$BF
1.7msec	(1.7msec/.0625µsec)	27,200	65535-27200=38335	$95:$BF
1.5msec	1.5msec/.0625µsec)	24000	65535-24000=41635	$A2:$A3
20msec	(20msec/.0625µsec)	320,000	Too big for 16-bit timer	Na
Prescale by 64	(20msec/.0625µsec)/64	5000	65535-5000=60535	$EC:$77

(2) Data Visualizer reading for each servo operation – clockwise, stop, and counterclockwise

➔ Required to show the on-pulse length and off-pulse length in the result)

I was having issues with my data visualizer, and unfortunately going home for break and being swarmed with work this past week led to poor time management, so I did not have time to fully figure it out. I will take the point deduction.

(3) YouTube video of motor motions

https://youtu.be/9ado74B7LbA

ECE3613 Processor System Laboratory Rubric

Lab #: 11 Section: 001 / 002
Name: _____

Activity	Contents	Full Points	Earned Points	Comment
1	Code	10		• Code with full comments
	Result	10		• Calculations for timing signal (5) • **Screenshots** showing your results (5)
2	Code	10		• Code with full comments
	Result	10		• Calculations for timing signal (5) • **Screenshots** showing your results (5)
3	Code	10		• Code with full comments
	Result	10		• Calculations for timing signal (5) • **Screenshots** showing your results (5)
4	Code	10		• Code with full comments
	Result	20		• Calculations for timing signal (5) and Data visualizer reading (15)
	Video	10		• Video for the servo in motion
Total		100		