CPE301 – SPRING 2019

Design Assignment 2C

Student Name: Minsung Cho

Student #: 2001446442

Student Email: chom3@unlv.nevada.edu

Primary Github address: <https://github.com/cho-minsung>

Directory: <https://github.com/cho-minsung/assignment2c>

1. **COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS**

ATMEGA328PB Arduino Shield Logic Analyzer

PB3(output signal) LED

PB5(LED) Switch

PC1(switch)

1. **INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1/A**

#include <avr/io.h>

#include <avr/interrupt.h>

int main()

{

DDRB |= (1<<PB3);

while(1) {

PORTB |= (1<<PB3);

*\_delay\_ms*(412.5);

PORTB &= (~(1<<PB3));

*\_delay\_ms*(337.5);

}

}

CODE 2A.1

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <util/delay.h>

int main(void)

{

DDRB |= (1 << 5); // the port b becomes an output at pin 2.

PORTB |= (1<<5); // the port b has an output at pin 2.

DDRC &= (0 << 1);

PORTC |= (1 << 1);

while (1) {

if (!(PINC & (1<<PINC1)))

{

PORTB &= ~(1<<5);

*\_delay\_ms*(2000); //delay of 2 seconds.

}

else

PORTB |= (1<<5);

};

return 0;

}

CODE 2A.2

1. **DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A**

#define *F\_CPU* 16000000UL

#include <avr/io.h>

int main()

{

DDRB |= (1<<3); // PB3 set as output

TCCR0A = 0x00; // normal mode

TCCR0B = 0X05; // set prescaler. 1024

while (1) {

TCNT0 = 0;

// 55% duty cycle, 412.5s

for (int i = 0; i <= 24; i++)

{

while (TCNT0 != 255)

{

}

TCNT0 = 0; // reset TCNT

}

PORTB &= ~(1<<3);

TCNT0 = 0;

//45% duty cycle, 337.5s

for (int j = 0; j < 21; j++)

{

while (TCNT0 != 255)

{

}

TCNT0 = 0;

}

PORTB |= (1<<3);

TCNT0 = 0;

}

}

code 2A.1 using Timer 0 – normal mode.

#define *F\_CPU* 16000000UL

#include <avr/io.h>

int main()

{

DDRB |= (1<<5); // PB5 set as output

PORTB |= (1<<5);

DDRC &= ~(1<<1); // PC1 set as input

PORTC &= (1<<1); // pull-up

TCCR0A = 0x00; // normal mode

TCCR0B = 0X05; // prescaler of 1024

TCNT0 = 0; //reset TCNT

while (1) {

while (!(PINC & (1<<PINC1)))

{

PORTB ^= (1<<5);

for (int i = 0; i <= 122; i++)

{

while (TCNT0 != 255)

{

//delay of 2 seconds.

}

TCNT0 = 0;

}

PORTB ^= (1<<5);

TCNT0 = 0;

}

}

}

code 2A.2 using Timer 0 – normal mode.

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <avr/interrupt.h>

int main()

{

DDRB |= (1<<3); // PB3 set as output

TCCR0A = 0x00; // normal mode

TCCR0B = 0X05; // set prescaler. 1024

TIMSK0 = (1<<TOIE0); //timer0 interrupt enabled

sei(); //interrupt enabled

while (1)

{

}

}

ISR(TIMER0\_OVF\_vect)

{

TCNT0 = 0;

TCNT0 = 0;

// 55% duty cycle, 412.5s

for (int i = 0; i <= 24; i++)

{

while (TCNT0 != 255)

{

}

TCNT0 = 0; // reset TCNT

}

PORTB &= ~(1<<3);

TCNT0 = 0;

//45% duty cycle, 337.5s

for (int j = 0; j < 21; j++)

{

while (TCNT0 != 255)

{

}

TCNT0 = 0;

}

PORTB |= (1<<3);

TCNT0 = 0;

}

code 2a.1 using TIMER0\_OVF\_vect interrupt in normal mode

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <avr/interrupt.h>

int main()

{

DDRB |= (1<<5); // PB5 set as output

PORTB |= (1<<5);

DDRC &= ~(1<<1); // PC1 set as input

PORTC &= (1<<1); // pull-up

TCCR0A = 0x00; // normal mode

TCCR0B = 0X05; // prescaler of 1024

TIMSK0 = (1<<TOIE0); //timer 0 overflow interrupt

sei(); //interrupt enabled

while (1)

{

}

}

ISR(TIMER0\_OVF\_vect)

{

TCNT0 = 0; //reset TCNT

while (1) {

while (!(PINC & (1<<PINC1)))

{

PORTB ^= (1<<5);

for (int i = 0; i <= 122; i++)

{

while (TCNT0 != 255)

{

//delay of 2 seconds.

}

TCNT0 = 0;

}

PORTB ^= (1<<5);

TCNT0 = 0;

}

}

}

code 2.a.2 using TIMER0\_OVF\_vect interrupt in normal mode

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <avr/interrupt.h>

int main()

{

DDRB |= (1<<3); // PB3 set as output

TCCR0A |= (1<<WGM01); // CTC mode

TCCR0B |= (1<<CS02) | (1<<CS00); // set prescaler. 1024

TIMSK0 = (1<<OCIE0A); //timer0 Compare enabled

OCR0A = 249; //top is set for 0.75s of period

sei(); //interrupt enabled

while (1)

{

}

}

ISR(TIMER0\_COMPA\_vect)

{

TCNT0 = 0;

TCNT0 = 0;

// 55% duty cycle, 412.5s

for (int i = 0; i <= 24; i++)

{

while (!(TIFR0 & (1<<OCF0A)))

{

}

TCNT0 = 0; // reset TCNT

TIFR0 |= (1<<OCF0A); //clear the flag

}

PORTB &= ~(1<<3);

TCNT0 = 0;

//45% duty cycle, 337.5s

for (int j = 0; j < 21; j++)

{

while (!(TIFR0 & (1<<OCF0A)))

{

}

TCNT0 = 0;

TIFR0 |= (1<<OCF0A);

}

PORTB |= (1<<3);

TCNT0 = 0;

}

code 2a.1 using TIMER0\_COMPA\_vect interrupt in CTC mode

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <avr/interrupt.h>

int main()

{

DDRB |= (1<<5); // PB5 set as output

PORTB |= (1<<5);

DDRC &= ~(1<<1); // PC1 set as input

PORTC &= (1<<1); // pull-up

OCR0A = 249; //top set

TCCR0A |= (1<<WGM01); // CTC mode

TCCR0B |= (1<<CS02)|(1<<CS00); // prescaler of 1024

TIMSK0 = (1<<OCIE0A); //timer compare enabled

sei(); //interrupt enabled

while (1)

{

}

}

ISR(TIMER0\_COMPA\_vect)

{

TCNT0 = 0; //reset TCNT

while (!(PINC & (1<<PINC1)))

{

PORTB ^= (1<<5);

for (int i = 0; i <= 122; i++)

{

while (!(TIFR0 & (1<<OCF0A)))

{

//delay of 2 seconds.

}

TCNT0 = 0;

TIFR0 |= (1<<OCF0A);

}

PORTB ^= (1<<5);

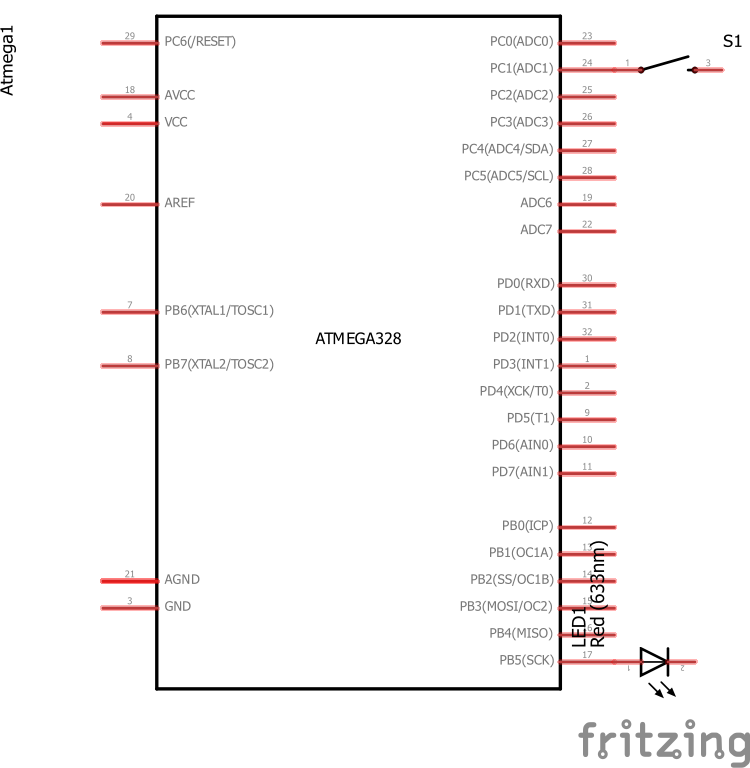
TCNT0 = 0;

}

}

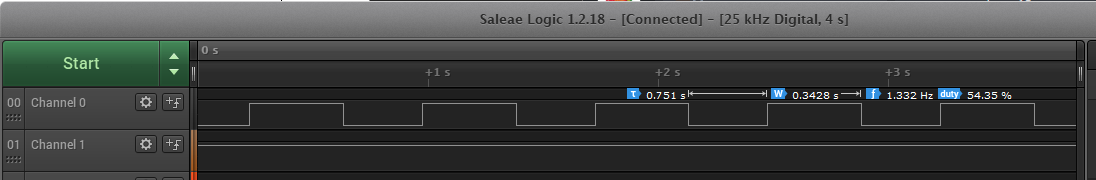
code 2a.2 using TIMER0\_COMPA\_vect interrupt in CTC mode

1. **SCHEMATICS**

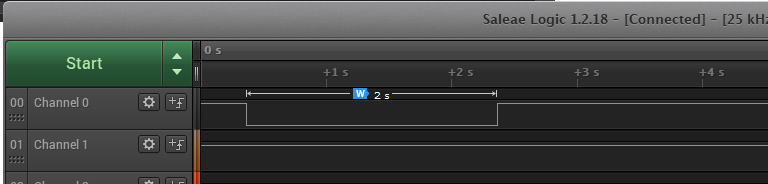


schematic of 2a.2 setup.

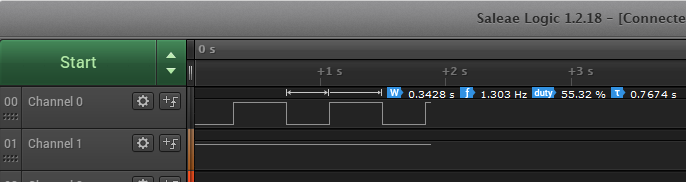
1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**



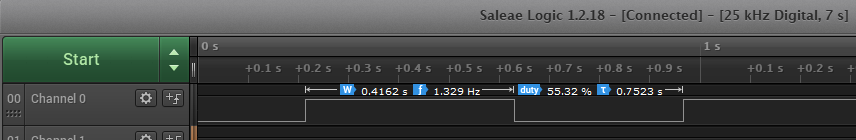
output of the code 2C.1.1 (code 2a.1 using Timer0 – normal mode)



output of the code 2C.1.2 (code 2a.2 using Timer0 – normal mode)

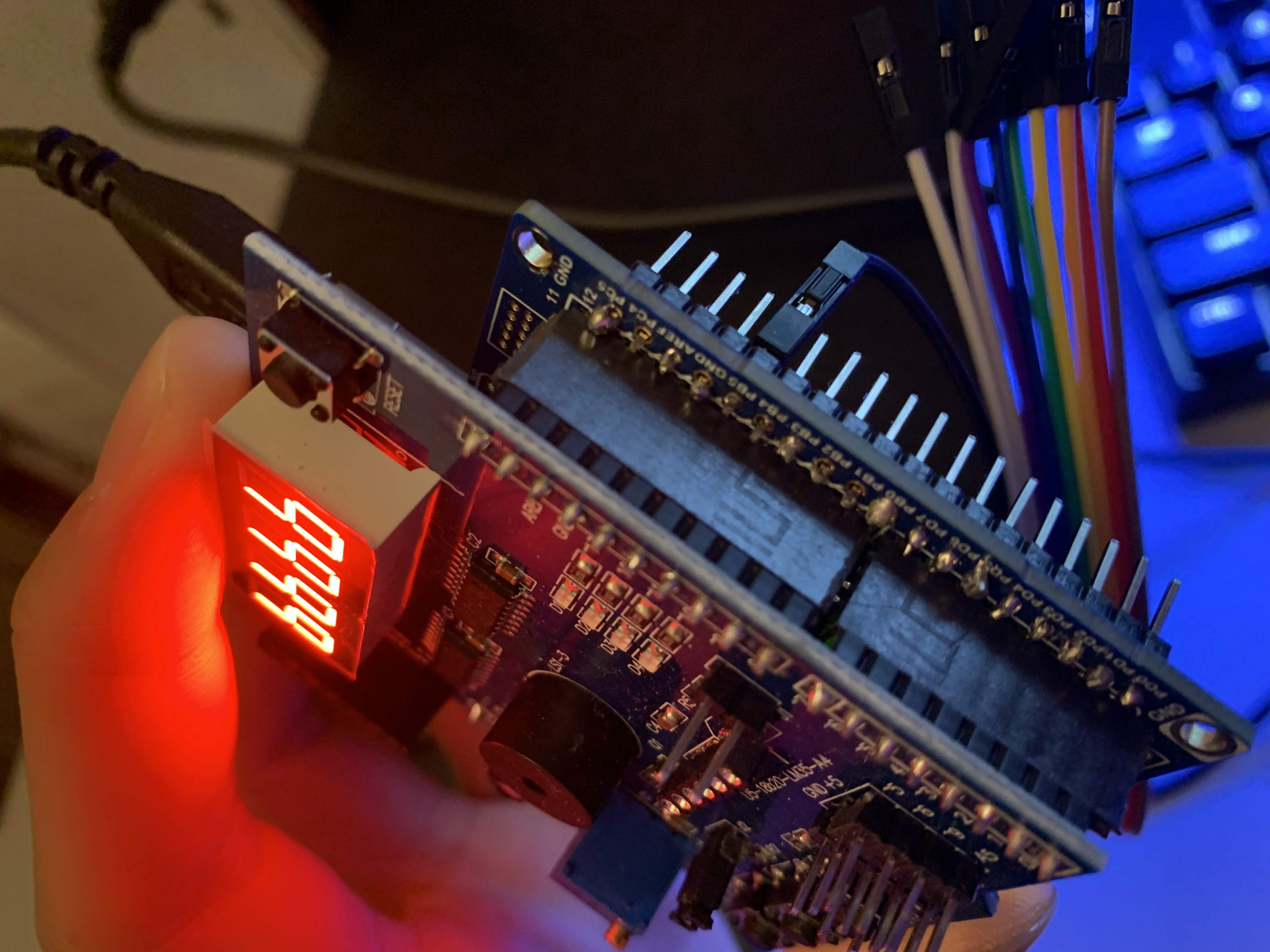


output of the code 2C.2.1 (code 2a.1 using TIMER0\_OVF\_vect interrupt in normal mode)



output of the code 2C.3.1 (code 2a.1 using TIMER0\_COMPA\_vect interrupt in CTC mode)

1. **SCREENSHOT OF EACH DEMO (BOARD SETUP)**



1. **VIDEO LINKS OF EACH DEMO**

https://youtu.be/xTaiCf3gtwU

1. **GITHUB LINK OF THIS DA**

<https://github.com/cho-minsung/assignment2c>

**Student Academic Misconduct Policy**

<http://studentconduct.unlv.edu/misconduct/policy.html>

“This assignment submission is my own, original work”.

Minsung Cho