마이크로 컨트롤러 5주차 과제

BUZZER Project No. 2

201921280 김남훈

순서

[SourceCode 2](#_Toc113543576)

[Implementation 3](#_Toc113543577)

# Source Code

#include <avr/io.h>

#define *F\_CPU* 16000000UL

#include <util/delay.h>

#include <avr/interrupt.h>

#define SV 340UL

#define TRIG 6

#define ECHO 7

#define INT4 4

#define BIT4\_LINE2\_DOT58 0x28

#define DISPON\_CUROFF\_BLKOFF 0x0C

#define DISPOFF\_CUROFF\_BLKOFF 0x08

#define INC\_NOSHIFT 0x06

#define DISPCLEAR 0x01

#define CUR1LINE 0x80

#define CUR2LINE 0xC0

#define *NULL* 0x00

#define Go 100

#define Stop 110

void init\_CLCD();

void init\_interrupt();

void CLCD\_cmd(char);

void CLCD\_data(char);

void CLCD\_puts(char\*);

void CLCD\_num\_display(int);

int read\_distance();

int state = Go;

char Distance[] = "Distance: ";

char CLCD\_NUM[] = "000.0";

int rec = 0;

int main(void)

{

DDRA = ((DDRA | (1 <<TRIG)) & ~(1 << ECHO));

DDRC = 0xff;

DDRD = 0xff;

DDRE = 0x00;

init\_interrupt();

init\_CLCD();

while(1)

{

CLCD\_cmd(DISPON\_CUROFF\_BLKOFF);

CLCD\_cmd(CUR1LINE);

if (state == Go)

{

*\_delay\_ms*(100);

rec = read\_distance();

}

CLCD\_puts(Distance);

CLCD\_num\_display(rec);

}

}

void init\_interrupt()

{

EICRB = 0x02;

EIMSK = 0x10;

SREG |= 0x80;

sei();

}

void init\_CLCD()

{

*\_delay\_ms*(50);

PORTC = 0x00;

CLCD\_cmd(BIT4\_LINE2\_DOT58);

CLCD\_cmd(DISPON\_CUROFF\_BLKOFF);

CLCD\_cmd(INC\_NOSHIFT);

CLCD\_cmd(DISPCLEAR);

*\_delay\_ms*(2);

}

ISR(INT4\_vect)

{

*\_delay\_ms*(50);

if ((PINE &= 0x10) != 0x00) return;

EIFR |= 1 << 4;

if (state == Go) state = Stop;

else state = Go;

}

int read\_distance()

{

unsigned int distance = 0;

TCCR1B = 0x03;

PORTA &= ~(1<<TRIG); *\_delay\_us*(10);

PORTA |= (1<<TRIG); *\_delay\_us*(10);

PORTA &= ~(1<<TRIG);

while(!(PINA & (1<<ECHO)));

TCNT1 = 0x0000;

while (PINA & (1<<ECHO));

TCCR1B = 0x00;

distance = (unsigned int)(SV \* (TCNT1 \* 4 / 2) / 1000);

return(distance);

}

void CLCD\_puts(char \*ptr)

{

while(\*ptr != *NULL*) CLCD\_data(\*ptr++);

}

void CLCD\_num\_display(int num)

{

CLCD\_NUM[0] = (num/1000)%10 + 0x30;

CLCD\_NUM[1] = (num/100)%10 + 0x30;

CLCD\_NUM[2] = (num/10)%10 + 0x30;

CLCD\_NUM[3] = '.';

CLCD\_NUM[4] = (num/1)%10 + 0x30;

CLCD\_NUM[5] = *NULL*;

CLCD\_puts(CLCD\_NUM);

}

void CLCD\_data(char data)

{

PORTD = 0x04;

*\_delay\_us*(1);

PORTD = 0x14;

PORTC = data & 0xf0;

PORTD = 0x04;

*\_delay\_us*(2);

PORTD = 0x14;

PORTC = (data << 4) & 0xf0;

PORTD = 0x04;

*\_delay\_ms*(1);

}

void CLCD\_cmd(char cmd)

{

PORTD = 0x00;

*\_delay\_us*(1);

PORTD = 0x10;

PORTC = cmd & 0xf0;

PORTD = 0x00;

*\_delay\_us*(2);

PORTD = 0x10;

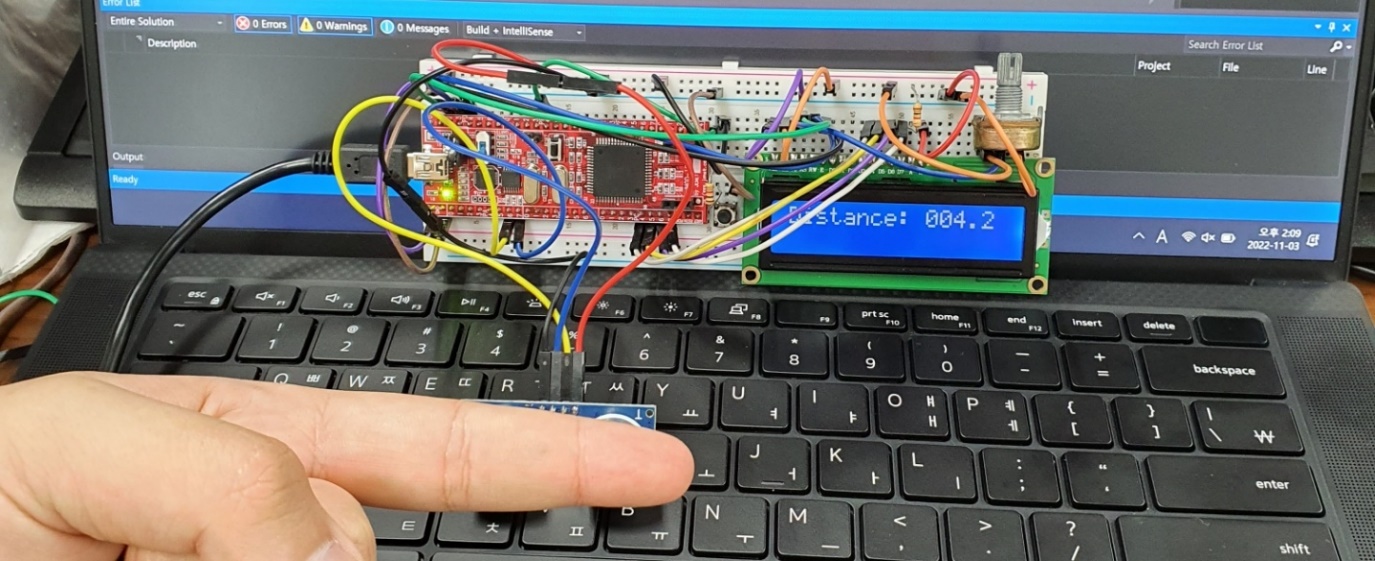
PORTC = (cmd << 4) & 0xf0;

PORTD = 0x00;

*\_delay\_ms*(1);

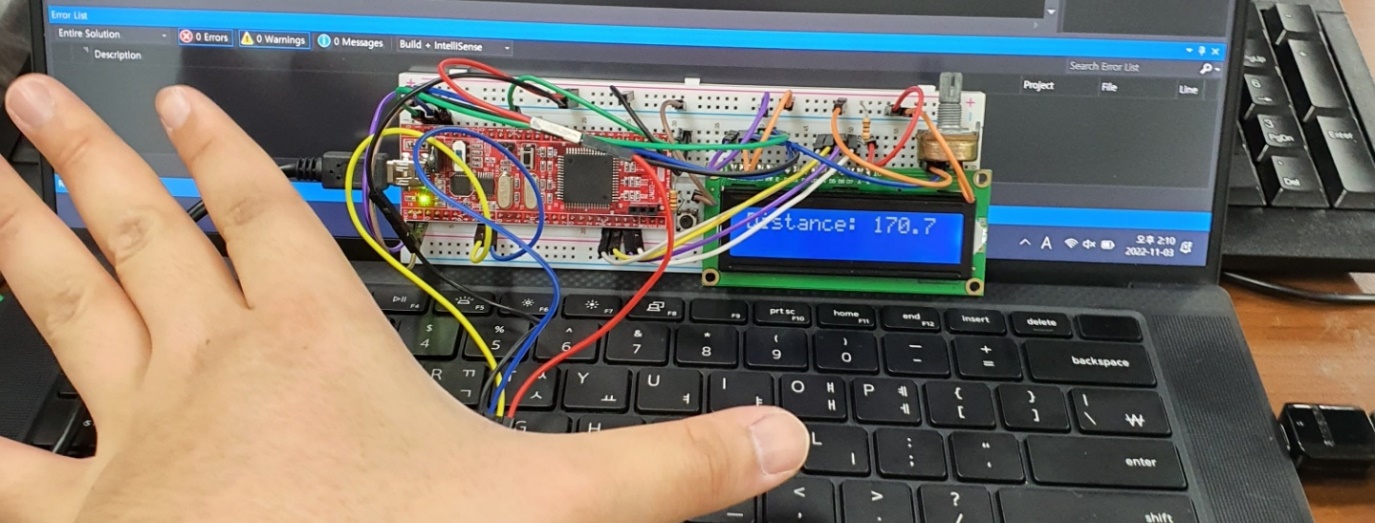
}

# Implementation





[아래 사진은 스위치를 눌러 측정을 중단한 상태]



\*물체가 아무리 가까워도, 측정상태가 STOP이기 때문에 측정값 업데이트가 되지 않는다.\*