마이크로 컨트롤1 10주차 과제

THERMAL SENSOR Project No. 2

201921280 김남훈

순서

[SourceCode 2](#_Toc113543576)

[Implementation 3](#_Toc113543577)

# SourceCode

#define *F\_CPU* 16000000UL

#define F\_SCK 40000UL

#include <avr/io.h>

#include <util/delay.h>

#define LM75A\_ADDR 0x90

#define LM75A\_TEMP\_REG 0

void init\_twi\_port();

int read\_twi\_2byte\_nopreset(char reg);

void display\_FND(int value);

void tone\_buzzer(void);

int buzzer\_onoff = 0;

int main(){

int i, temperature;

init\_twi\_port();

while (1){

for (i=0; i<30; i++){

if (i == 0) temperature = read\_twi\_2byte\_nopreset(LM75A\_TEMP\_REG);

display\_FND(temperature);

}

}

}

void init\_twi\_port(){

DDRC = 0xff;

DDRG = 0xff;

TWSR = TWSR & 0xfc;

TWBR = (*F\_CPU*/F\_SCK - 16) / 2;

}

int read\_twi\_2byte\_nopreset(char reg){

char high\_byte, low\_byte;

TWCR = (1 << TWINT) | (1<<TWSTA) | (1<<TWEN);

while (((TWCR & (1 << TWINT)) == 0x00) || (TWSR & 0xf8) != 0x08) ;

TWDR = LM75A\_ADDR | 0;

TWCR = (1 << TWINT) | (1 << TWEN);

while (((TWCR & (1 << TWINT)) == 0x00) || (TWSR & 0xf8) != 0x18) ;

TWDR = reg;

TWCR = (1 << TWINT) | (1 << TWEN);

while (((TWCR & (1 << TWINT)) == 0x00) || (TWSR & 0xf8) != 0x28) ;

TWCR = (1 << TWINT) | (1<<TWSTA) | (1<<TWEN);

while (((TWCR & (1 << TWINT)) == 0x00) || (TWSR & 0xf8) != 0x10) ;

TWDR = LM75A\_ADDR | 1;

TWCR = (1 << TWINT) | (1 << TWEN);

while (((TWCR & (1 << TWINT)) == 0x00) || (TWSR & 0xf8) != 0x40) ;

TWCR = (1 << TWINT) | (1 << TWEN | 1 << TWEA);

while(((TWCR & (1 << TWINT)) == 0x00) || (TWSR & 0xf8) != 0x50) ;

high\_byte = TWDR;

TWCR = (1 << TWINT) | (1 << TWEN);

while(((TWCR & (1 << TWINT)) == 0x00) || (TWSR & 0xf8) != 0x58) ;

low\_byte = TWDR;

TWCR = (1 << TWINT) | (1 << TWSTO) | (1 << TWEN);

while ((TWCR & (1 << TWSTO))) ;

return((high\_byte<<8) | low\_byte);

}

void display\_FND(int value){

char digit[12] = {0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7c, 0x07, 0x7f, 0x67, 0x00, 0x40};

char fnd\_sel[4] = {0x01, 0x02, 0x04, 0x08};

int value\_int, value\_deci, num[4], i;

if ((value & 0x8000) != 0x8000)

num[3] = 10;

else{

num[3] = 11;

value = (~value)+1;

}

value\_int = (value & 0x7F00) >> 8;

value\_deci = (value & 0x0080);

num[2] = (value\_int / 10) % 10;

num[1] = value\_int % 10;

num[0] = (value\_deci == 0x80) ? 5 : 0;

if(value\_int >= 30 || value\_int < 20) buzzer\_onoff = !buzzer\_onoff;

tone\_buzzer();

for(i=0; i<4; i++){

PORTC = digit[num[i]]; PORTG = fnd\_sel[i];

if (i==1) PORTC |= 0x80;

if (i%2) *\_delay\_ms*(2);

else *\_delay\_ms*(3);

}

}

void tone\_buzzer(void)

{

DDRB |= 0x10;

if (buzzer\_onoff)

{

PORTB |= 1 << 4;

*\_delay\_us*(200);

}

else PORTB &= ~(1 << 4);

}

# Implementation

