

ภาคผนวก ก

โปรแกรมซอร์สโค้ดภาษาซีของไอซีไมโครคอนโทรลเลอร์

### โปรแกรมซอร์สโค้ดภาษาซีของไอซีไมโครคอนโทรลเลอร์

```
#include <16F877A.h>

#define adc=8

#FUSES NOWDT           //No Watch Dog Timer

#FUSES HS              //High speed Osc (> 4mhz for PCM/PCH) (>10mhz for PCD)

#FUSES NOPUT          //No Power Up Timer

#FUSES NOPROTECT       //Code not protected from reading

#FUSES NOBROWNOUT      //No brownout reset

#FUSES NOLVP           //No low voltage prgming, B3(PIC16) or B5(PIC18) used for I/O

#FUSES NOCPD           //No EE protection

#FUSES NOWRT           //Program memory not write protected

#FUSES NODEBUG         //No Debug mode for ICD

#use delay(clock=10000000)

#use rs232(baud=9600,parity=N,xmit=PIN_C6,rcv=PIN_C7)

/*****

/* INCLUDE LIBRARYS                                     */

*****/

#include <sht15.c>

#include <lcd.c>

/*****

/* TEMPERATURE ADRESS                                     */

*****/

#define TEMP_ADDR0  0x01

#define TEMP_ADDR1  0x02

#define TEMP_ADDR2  0x03

#define TEMP_ADDR3  0x04

/*****

/* DEFINE BUTTON AND RELAY                               */
```

```

/*****/

#define BTTN_SET()  input(PIN_B0)

#define BTTN_UP()   input(PIN_B1)

#define BTTN_DW()   input(PIN_B2)

#define RELAY(x)    output_bit(PIN_B7,x)

#define ENABLE      1

#define DISABLE     0

int1 Display_State = ENABLE;

int time_out;

float temp,ref_temp;

/*****/

/* FUNCTION PROTOTYPES */

/*****/

void setting(void);

void display(void);

void control(void);

float read_temp(void);

void write_temp(float temp);

/*****/

/* FUNCTION : main() */

/*****/

void main (void)
{
    /*****/

    /* Set i/o port */

    /*****/

    port_b_pullups(true);

    /*****/

```

```

/* Set timer1 */
/*****/

setup_timer_1(T1_INTERNAL|T1_DIV_BY_8);
set_timer1(34286);
/*****/

/* Enable interrupt */
/*****/

enable_interrupts(GLOBAL);
enable_interrupts(INT_TIMER1);
/*****/

/* Set temperature in eeprom */
/*****/

if(read_eeprom(0x00) == 0xFF){
    write_eeprom(0x00,0x00);
    write_temp(14);
}
/*****/

/* Read Reference Temperature in eeprom */
/*****/

ref_temp = read_temp();
lcd_init();
while(true){
    setting();
    display();
    control();
}
}

/*****/

/* FUNCTION : timer1_isr() */

```

```

/* Interrupt every 100 ms */
/*****

#INT_TIMER1

void timer1_isr(void)
{
    static unsigned int time = 0;
    set_timer1(34286);
    if(++time >= 10){
        time = 0;
        Display_State = ENABLE;
        time_out++;
    }
}

*****/

/* FUNCTION : setting() */
/*****

void setting(void)
{
    int state = 0;
    float temp_buf = 0, update_temp = -1;
    unsigned int16 t_btn_up = 0, t_btn_dw = 0;
    static unsigned int32 t_btn_up_dw = 0;

    if(!BTTN_SET()){
        delay_ms(10);
        if(!BTTN_SET()){
            RELAY(0);
            time_out = 0;
            temp_buf = ref_temp;

```

```

lcd_gotoxy(1,1);printf(lcd_putc,"  Setting  ");
while(time_out < 5){
    if(update_temp != temp_buf){
        update_temp = temp_buf;
        lcd_gotoxy(1,2);printf(lcd_putc," Temp : %f C    ",temp_buf);
    }
    /***/
    /* PRESS BUTTON "UP" */
    /***/
    if(!BTTN_SET() && state!=0){
        delay_ms(10);
        if(!BTTN_SET()){
            lcd_gotoxy(1,1); printf(lcd_putc," Saving...  ");
            lcd_gotoxy(1,2); printf(lcd_putc," Temperature  ");
            ref_temp = temp_buf;
            write_temp(ref_temp);
            while(!BTTN_SET());delay_ms(10);
            goto exit_setting;
        }
    }
    if(BTTN_SET()){
        state = 1;
    }
    /***/
    /* PRESS BUTTON "UP" */
    /***/
    if(!BTTN_UP()){
        if(t_btn_up==30000){
            if(temp_buf<100) temp_buf = temp_buf + 0.05;
            delay_ms(10);

```

```

        time_out = 0;
    }
else
    t_btn_up++;
}
if(BTTN_UP()){
    if((t_btn_up>1000)&&(t_btn_up<30000)){
        if(temp_buf<100) temp_buf = temp_buf + 0.05;
        time_out = 0;
    }
    t_btn_up = 0;
}

/*****

/* PRESS BUTTON "DW"          */

*****/

if(!BTTN_DW()){
    if(t_btn_dw==30000){
        if(temp_buf>-50) temp_buf = temp_buf - 0.05;
        delay_ms(10);
        time_out = 0;
    }
else
    t_btn_dw++;
}
if(BTTN_DW()){
    if((t_btn_dw>1000)&&(t_btn_dw<30000)){
        if(temp_buf>-50) temp_buf = temp_buf - 0.05;
        time_out = 0;
    }
    t_btn_dw = 0;
}

```

```

        }
    }
}

exit_setting:

    /**
     * Press button "UP" + "DW" for set default temperature
     */
    /**
     *
     */
    if(!BTTN_UP() && !BTTN_DW()){
        if(t_btn_up_dw == 20000){
            lcd_gotoxy(1,1); printf(lcd_putc, " Set default of ");
            lcd_gotoxy(1,2); printf(lcd_putc, " temperature. ");
            write_temp(14);
            ref_temp = 14;
            delay_ms(2000);
            while(!BTTN_UP() && !BTTN_DW());
        }
        else
            t_btn_up_dw++;
    }
    if(BTTN_UP() || BTTN_DW()){
        t_btn_up_dw = 0;
    }
}

    /**
     * FUNCTION : display()
     */
    /**
     *
     */
    void display(void)
    {

```

```

float fRh_lin;

float fRh_true;

long IValue_rh;

long IValue_temp;

int R;

if(Display_State == ENABLE){

    Display_State = DISABLE;

    SHTConReset();

    delay_ms(11);

    /***/

    /* Read temperature */

    /***/

    SHTStart();          //@1 start transmission

    R=SHTWrite(MEASURE_TEMP);    //@2 measure temperature

    if(R==1){

        lcd_gotoxy(1,1); printf(lcd_putc," The sensor has ");

        lcd_gotoxy(1,2); printf(lcd_putc," a problem! ");

        }

    else

        {

            IValue_temp = SHTRead();

            temp = (D1+(D2*IValue_temp)); // temperature calculation

            delay_ms(12);          // delay 11ms before next command

            /***/

            /* Read humidity */

            /***/

            SHTStart();          // start transmission

            SHTWrite(MEASURE_HUMI);    // measure relative humidity

            IValue_rh = SHTRead();

```

```

/*****/

/* Relative humidity calculation */

/*****/

fRh_lin = (C1+(C2*IValue_rh)+(C3*IValue_rh*IValue_rh));
fRh_true = (((temp-25)*(T1+(T2*IValue_rh)))+fRh_lin);

/*****/

/* Show temperature on lcd */

/*****/

lcd_gotoxy(1,1); printf(lcd_putc," Set : %3.2f C    ",ref_temp);
lcd_gotoxy(1,2); printf(lcd_putc," temp: %3.2f C    ",temp);
}
}

}

/*****/

/* FUNCTION : control() */

/*****/

void control(void)
{
    if(temp>ref_temp) RELAY(1);
    else RELAY(0);
}

/*****/

/* FUNCTION : display() */

/*****/

float read_temp(void)
{
    float temp;

```

```

*((unsigned int8 *)&temp) + 0) = read_eeprom(TEMP_ADDR0);
*((unsigned int8 *)&temp) + 1) = read_eeprom(TEMP_ADDR1);
*((unsigned int8 *)&temp) + 2) = read_eeprom(TEMP_ADDR2);
*((unsigned int8 *)&temp) + 3) = read_eeprom(TEMP_ADDR3);
return(temp);
}

/*****

/* FUNCTION : display() */

*****/

void write_temp(float temp)
{
    write_eeprom(TEMP_ADDR0,*((unsigned int8 *)&temp)+0));
    write_eeprom(TEMP_ADDR1,*((unsigned int8 *)&temp)+1));
    write_eeprom(TEMP_ADDR2,*((unsigned int8 *)&temp)+2));
    write_eeprom(TEMP_ADDR3,*((unsigned int8 *)&temp)+3));
}

```