
2024 한국폴리텍대학 (다기능과정) IT융합전자회로 설계 및 제작 경진대회

제 3 과제

Project Name : MICOM PROGRAMMING

제한 시간 : 5시간



후원 : 학교법인 한국폴리텍대학

협찬 : 한국폴리텍대학 대구캠퍼스,나인플러스아이티(주)

IT융합전자회로 설계 및 제작 경진대회 과제

과 제 명	Micom Programming	경기시간	5시간
비번호		감독위원확인	(인)

1. 요구사항

가. 지급된 프로그램 설계 회로기판, 회로도, 동작 설명 사항을 참조하여 제작하시오.

나. 배포되는 소스는 프로그램 요구사항 (1)~요구사항 (6)까지 동작되게 한 것이다.
이 소스를 완성하여 컴파일, 라이팅하여 프로그램 요구사항 (1)~요구사항 (6)까지 동작시키시오.

※ 단, 자신의 컴파일과 맞지 않는 부분은 수정하여 동작시키시오.

다. 배포된 소스를 참조하여 프로그램 [문제 1]~[문제 4]를 프로그램하고 컴파일, 라이팅하여 동작시킨 후 자신의 비번호를 적고, 저장하시오.

C:\RefrigeratorXX.C (XX : 자신의 비번호)

라. 배포된 회로와 PCB를 보고 프로그램에 이상이 있는 부분은 수정하시오.

마. 과제 개요

이 작품은 가정에서 사용하는 냉장고 시스템 제어의 일부분을 모델링해 구현한 것이다.

- (1) 온도 설정과 FND를 이용한 현재 온도 표시 기능
- (2) 주변 밝기에 따른 램프 밝기 자동 조절 기능
- (3) 동작 상태에 따른 LCD, LED 표시 기능
- (4) LCD CGRAM을 이용한 특수 문자 표시 기능
- (5) 스위치 및 LED 기능 설명

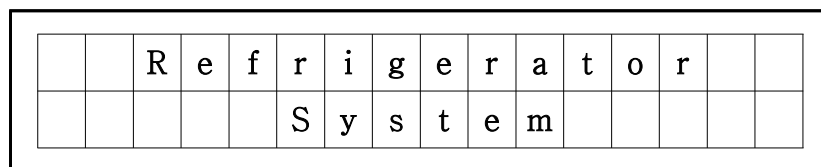
구 분	기 능	비 고
SW1	① 파워 냉장 선택 ② 온도 조절 모드 선택	▷ 냉장모드에서 동작 ▷ 모드 선택 메뉴에서 동작
SW2	냉장고 문 열기, 닫기 동작	▷ 냉장모드에서 동작
SW3	① 온도 조절 스위치(누를 때 1 증가) ② 조도 설정 스위치(누를 때 1 증가)	▷ 냉장 온도 설정에서 동작 ▷ 램프 밝기 조정에서 동작
SW4	① 온도 조절 스위치(누를 때 1 감소) ② 조도 설정 스위치(누를 때 1 감소)	▷ 냉장 온도 설정에서 동작 ▷ 램프 밝기 조정에서 동작
SW5	① 온도 및 조도 설정 메뉴 표시 ② 온도 및 조도 설정 완료	▷ 냉장모드에서 동작 ▷ 온도 및 조도 설정에서 동작
LED1	냉장고 디스플레이 램프용 LED	램프 밝기 조정 설정 이후 동작
LED2	일반 냉장 동작 표시 LED	
LED3	냉장 설정 온도 및 현재 냉장 온도 비교 표시 LED	
LED4	파워 냉장 동작 표시 LED	

바. 동작 요구 사항

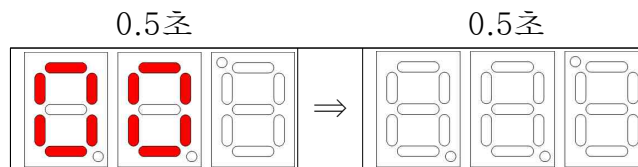
(1) 전원 인가 및 RESET(SW7)을 누른 경우 다음과 같이 동작되게 하시오.

※ 아래 세부 요구사항 (가), (나), (다), (라)는 동시에 동작함에 유의하시오.

(가) LCD : 3초간 아래와 같이 표시되게 하시오.

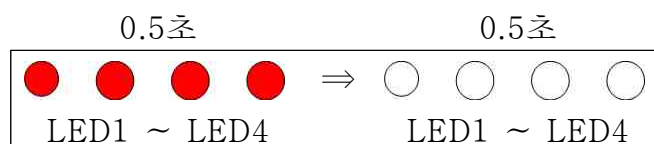


(나) FND : 아래와 같이 0.5초 간격으로 3회 점멸한 후 소등되게 하시오.



(다) 부저 : 0.5초 간격으로 3회 울리게 하시오.

(라) LED : 아래와 같이 0.5초 간격으로 3회 점멸한 후 소등되게 하시오.

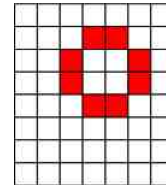


(2) “요구사항 (1),(3),(4),(5),(6)”을 실행 후 다음과 같이 동작되게 하시오.

(가) LCD

M	o	d	e	:	N	o	r			L	u	x	:	0	0
T	e	m	p	:	0	4	°	C		D	o	o	r	:	C

[문제 1] LCD의 CGRAM을 이용하여 LCD에 온도를 나타내는 “℃”의 ‘°’가 표시 되도록 프로그램 하시오.(아래 ①)의 패턴 참조)



① [온도의 "도(°)" LCD 문자 패턴 형태]

- ② 초기 주변 밝기는 00[Lux], “요구사항 (4)” 설정 이후에는 현재 밝기 값을 “설정된 범위 내”에서 실시간으로 표시되게 하시오.
- ③ 초기 온도는 “04”도를 표시되게 하시오.
- ④ 스위치를 이용해 온도를 설정하면, 리셋을 하거나 전원을 다시 켜었을 경우에도 설정 온도가 표시되게 하시오.

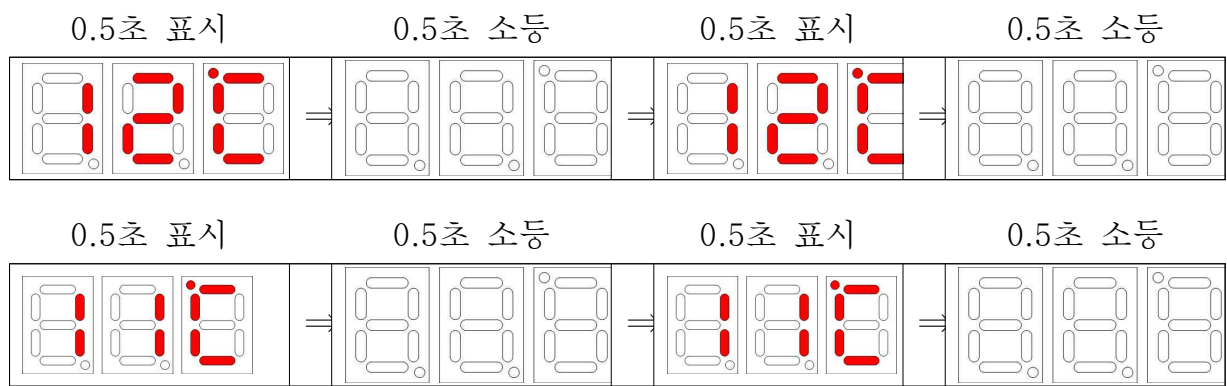
(나) FND : 현재 냉장 온도가 표시되게 하시오.

① 전원 인가 및 리셋을 한 경우 다음과 같이 표시되게 하시오.

㉠ 온도 표시 : 현재온도 표시 후 2초에 1도씩 감소되며 0.5초 마다 점멸되게 하시오.

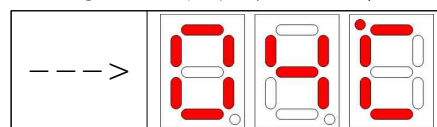
㉡ 설정 온도까지 감소되면 점멸을 멈추게 하시오.

※ 단, 맨 처음의 현재 온도는 12℃, 설정온도는 4℃로 함.



...

설정온도에서 점멸을 멈춤



② “요구사항 (3)”처럼 냉장 온도를 재설정 한 경우에 다음과 같이 동작되게 하시오.

㉠ 현재 온도 값이 설정 값보다 크면 2초에 1도씩 감소되며 0.5초 마다

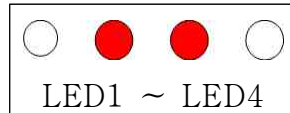
점멸되게 하시오.

㉞ 현재 온도 값이 설정 값보다 작으면 2초에 1도씩 증가되며 0.5초 마다 점멸되게 하시오.

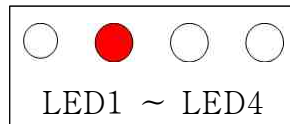
㉟ 설정 온도와 같아지면 점멸을 멈추게 하시오.

(다) LED

① 현재 온도와 설정 온도가 같지 않은 경우 다음과 같이 표시되게 하시오.



② 현재 온도와 설정 온도가 같은 경우 다음과 같이 표시되게 하시오.



(3) 냉장 온도 설정[SW5를 누름]

(가) SW5(Set Mode)를 누른 경우 LCD에 다음과 같이 표시되게 하시오.

				S	e	t	_	M	o	d	e				
P	r	e	s	s		K	e	y		S	W	1	-	2	

(나) SW1을 누른 경우 LCD에 다음 같이 이전 설정 온도가 표시되게 하시오.

S	e	t		T	e	m	p	e	r	a	t	u	r	e	
						0	4	°	C						

(다) 온도 설정

① SW3을 누름 : 온도가 1도씩 증가되게 하시오.

② SW4를 누름 : 온도가 1도씩 감소되게 하시오.

③ 설정 가능 온도 범위 : 00~10도

(설정 범위의 최댓값이 된 경우 SW3을 눌러도 증가하지 않으며, 최솟값이 된 경우 SW4를 눌러도 감소하지 않음)

④ SW5를 누르면 설정을 완료하고, “요구사항 (2)”의 동작을 수행하게 하시오.

(라) FND : “요구사항 (2)”의 현재 온도 값이 표시되게 하시오.

※ 단, 냉장 온도 설정 중에는 현재 온도는 변하지 않음.

(마) LED : “요구사항 (2)”와 같이 동작되게 하시오.

(바) 부저 : 온도 설정 범위를 초과하는 키가 눌리면 0.5초 1회 울리게 하시오.

(4) 램프 밝기 조정 설정[SW5(Set Mode)를 누름]

주변 밝기를 'CDS'의 값을 읽어 10진수 00~99의 100단계로 변환 후 아래 요구 사항대로 동작되게 하시오.

(가) SW5(Set Mode)를 누른 경우 LCD에 다음과 같이 표시되게 하시오.

				S	e	t	_	M	o	d	e				
P	r	e	s	s		K	e	y		S	W	1	-	2	

(나) SW2를 누른 경우 LCD 화면

① 처음 설정하는 경우 다음과 같이 표시되게 하시오.

(최댓값(Max) : 99, 최솟값(Min) : 00)

	S	e	t		L	i	g	h	t		L	u	x		
	M	a	x	:	9	9			M	i	n	:	0	0	

② 이전 설정 값이 있는 경우 저장되어 있는 값이 표시되게 하시오.

※ 단, 전원 인가 및 리셋의 경우에는 초기 값이 표시됨.

(다) 밝기 최댓값, 최솟값 설정하기

① SW3을 한번 누르면 최댓값 숫자가 'CDS'의 상태에 따라 00~99사이로 변하며, SW3을 다시 누르면 고정됨 ('81~98'사이가 되도록 센서 값을 조정하시오.)

② SW4를 한번 누르면 최솟값 숫자가 'CDS'의 상태에 따라 00~99사이로 변하며, SW4를 다시 누르면 고정됨 ('01~19'사이가 되도록 센서 값을 조정하시오.)

③ SW5를 누르면 설정을 완료하고, “요구사항 (2)”의 동작을 수행하며,

[문제 2] “LED1”의 밝기는 아래와 같이 동작되도록 프로그램 하시오.

(타이머 이용)

Lux 값	99~80	79~60	59~40	39~21	20~00
LED 밝기	소등	1단계	2단계	3단계	최대

(라) FND : “요구사항 (2)”의 현재 온도 값이 표시되게 하시오.

※ 단, 냉장 조도 설정 중일 때 현재 온도는 변하지 않음.

(마) LED : LED1 이외의 LED는 “요구사항 (2)”와 같이 동작되게 하시오.

(5) 파워 냉장 모드

파워 냉장 모드는 SW1(파워모드)을 누르면 선택되고, 현재 온도가 설정 온도와 같아지면 자동으로 일반 냉장 모드로 전환된다.

[문제 3]

(가) 현재 온도가 설정 온도보다 큰 경우 다음과 같이 동작되도록 프로그램 하시오.

① LCD

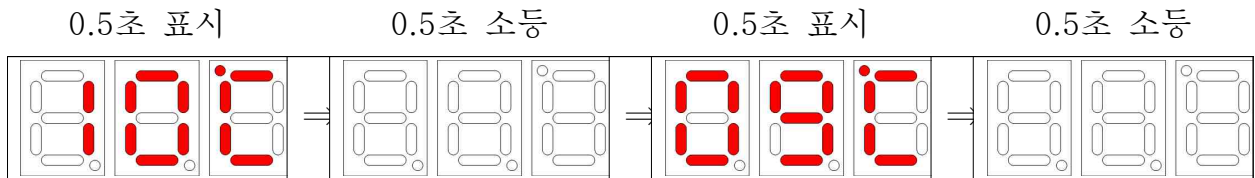
M	o	d	e	:	P	o	w			L	u	x	:	0	0
T	e	m	p	:	0	4	°	C		D	o	o	r	:	C

② 조도(Lux), 설정 온도, 문의 개폐 상태는 이전 처리 상태를 유지하도록 하시오.

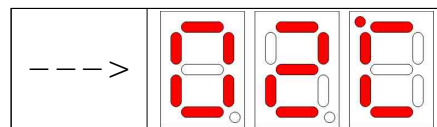
③ FND : 현재 냉장 온도가 표시되게 하시오.

㉓ 1초에 1도씩 감소되며 0.5초 마다 점멸되게 하시오.

㉔ 설정 온도까지 감소되면 점멸을 멈추게 하시오.

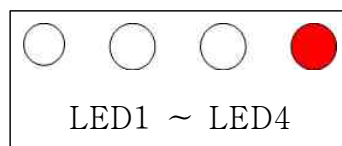


설정온도에서 점멸을 멈춤



(현재 온도 10도, 설정 온도가 02도인 경우)

④ LED : 파워 냉장 모드 동안 아래 그림과 같이 동작



(나) 현재온도가 설정온도보다 작거나 같은 경우 다음과 같이 동작되게 하시오.

① LCD : 다음과 같은 동작을 3회 반복한 후 다음을 수행하게 하시오.

	N	o	t		P	o	w	e	r		M	o	d	e	
P	l	e	a	s	e		S	e	t		T	e	m	p	!

↓ 0.5초 후

↓ 0.5초 후

- ② FND, LED : 이전 상태 값이 유지되게 하시오.
- ③ 부저 : 0.5초 간격으로 3회 반복하여 울리게 하시오.
- ④ 3초 후 요구 사항 ②와 같이 동작되게 하시오.

(6) 문의 개폐 모드

"SW2(Door)"를 한번 누르면 냉장고 문이 열리며, 다시 누르면 냉장고 문이 닫힌다. 단 파워 냉장 모드인 경우에는 문을 열수 없다.

(가) 냉장고 문 열기

- ① LCD : 다음과 같이 동작되게 하시오.

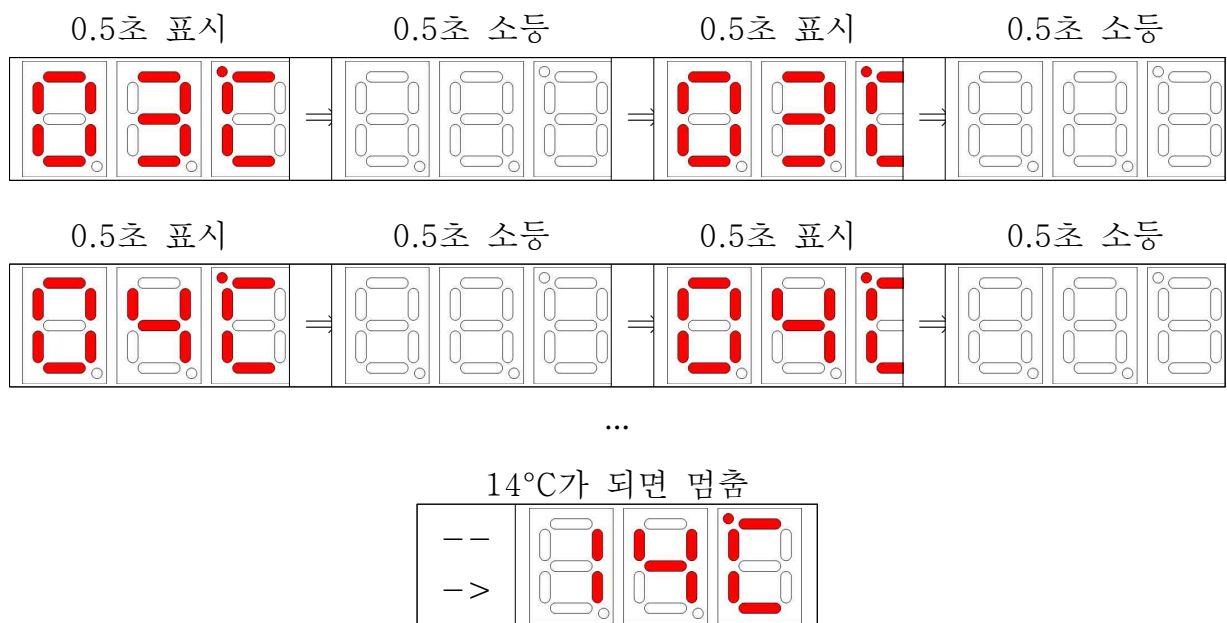
			D	o	o	r		O	p	e	n				
	T	i	m	e	:	0	0		U	s	e	:	0	0	

- “Time:00”은 현재 냉장고를 열고 있는 시간으로 1초마다 1씩 증가한다. 시간 범위는 “00~99”초이다.
- “Use:00”은 냉장고를 열은 횟수를 나타내는 것으로, 냉장고를 열 때 마다 1씩 증가한다. (이 값은 리셋 및 전원인가 시에도 유지되어야 한다.)

- ② FND

냉장고가 열려 있는 동안 현재 온도는 2초에 1도씩 증가하며, 열려 있는 시간에 따라 최대 14도 까지 증가되도록 하시오.

(현재 온도가 03도인 경우)



- ③ LED : “요구사항 (2)”와 같이 동작되게 하시오.

[문제 4] 냉장고 문이 열려 있는 경우 부저는 아래와 같이 동작되도록 프로그램 하시오.

- ④ 부저 : 문이 열려 있는 시간이 10초 이상부터 아래 표와 같이 울리도록 하시오.

문이 열려 있는 시간	부저 울림
0~10	울리지 않음
11~20	2초 마다 0.5초 울림
21~30	1초 마다 0.5초 울림
30~	계속 울림

(나) 냉장고 문 닫기 : “요구사항 (2)”의 동작을 수행하게 하시오.

마. 배포된 PCB에 오류가 있으면 수정하여 동작시키시오.

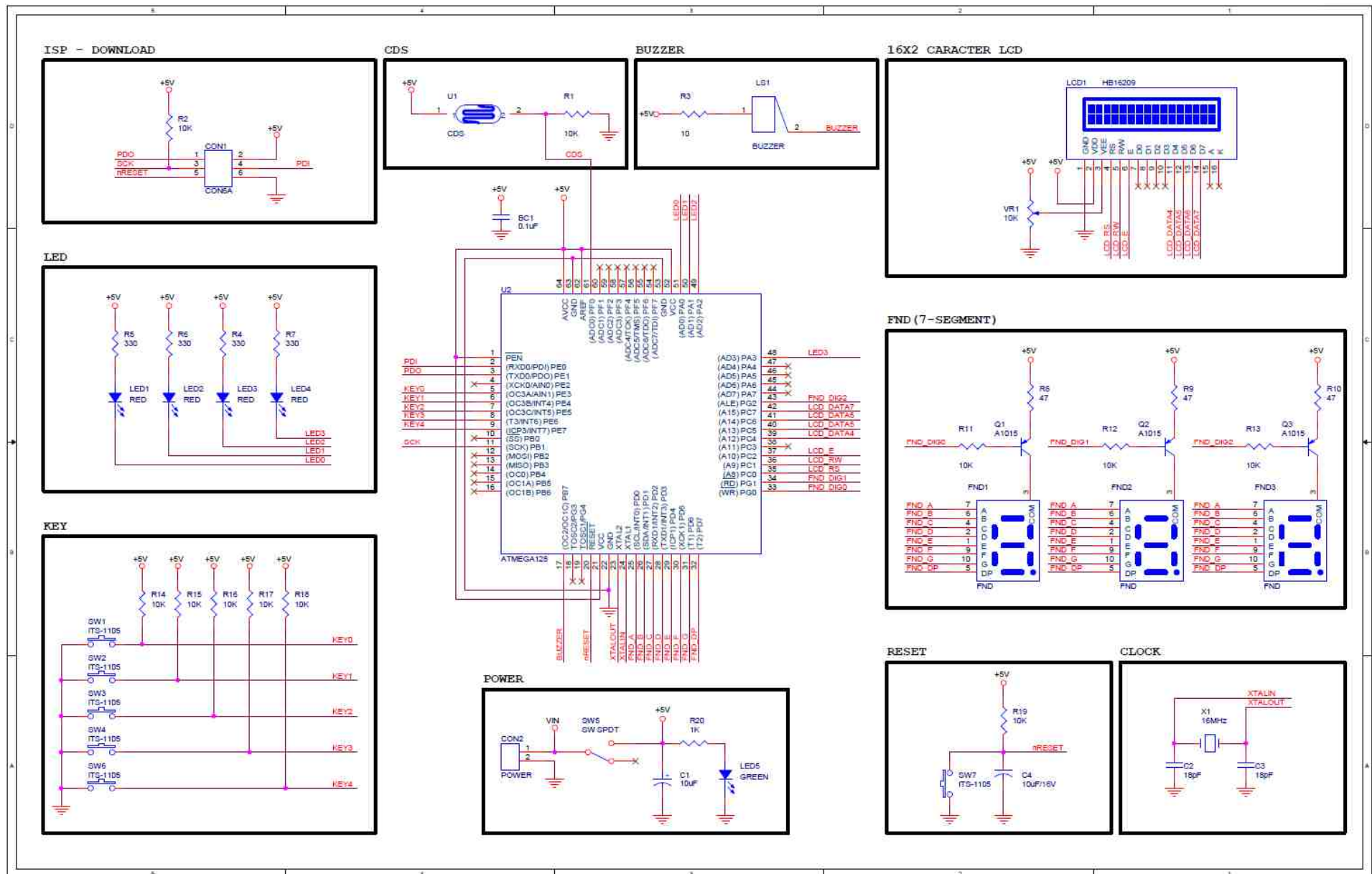
바. 작업이 완료되면 심사위원에게 동작 검사를 신청하시오.

2. 유의사항

가. 안전사고에 유의하시오.

나. 심사의원 및 집행위원의 지시에 따라 작업하시오.

3. 회로도



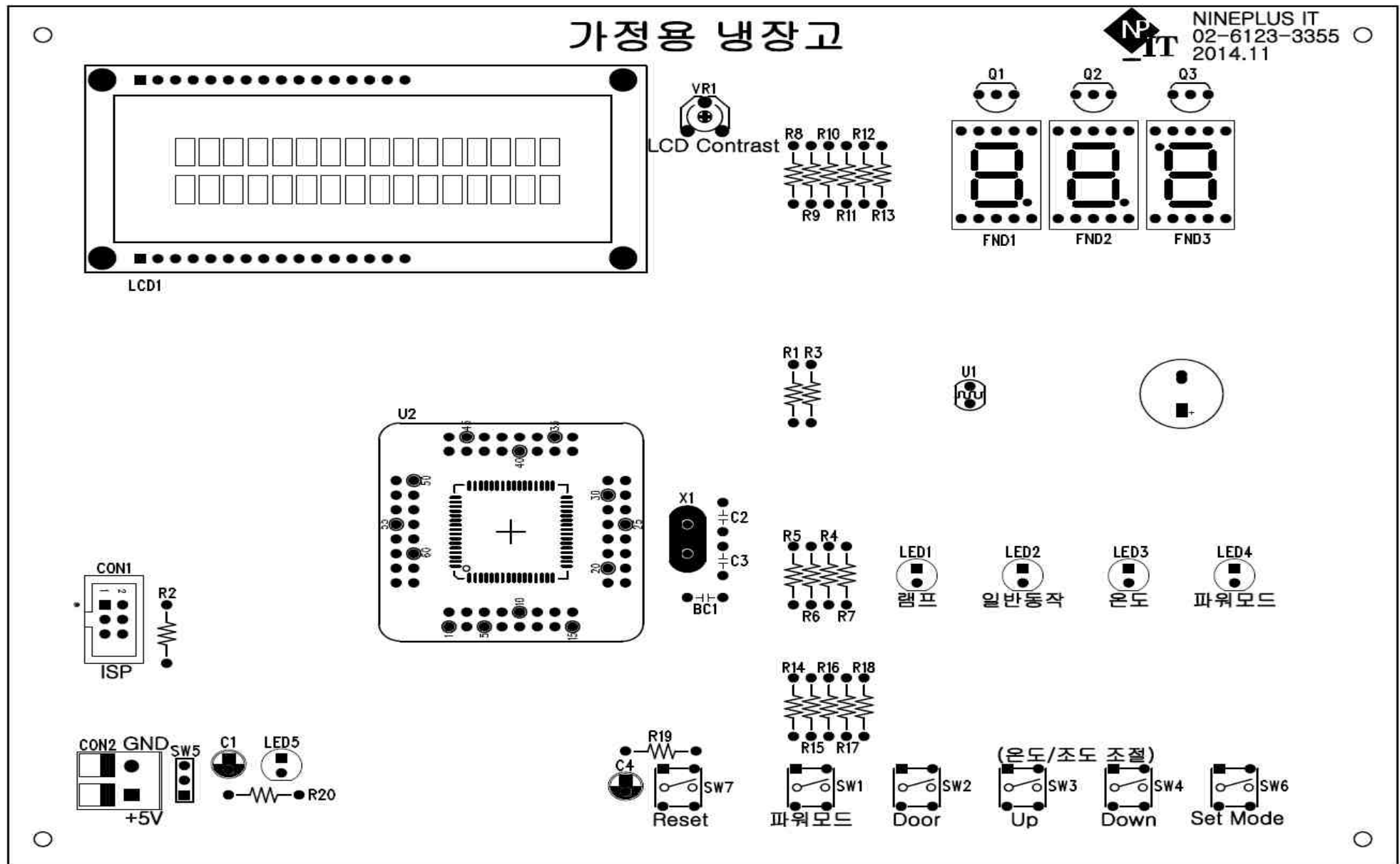
4-1. 지급 재료 목록

일련 번호	재 료 명	규 격(치수)	단위	수량	비 고
1	IC	ATMEGA128	개	1	
2	CPU Module Connector	Dual Connector(8x2, 수)	개	4	
3	CPU Module Connector	Dual Connector(8x2, 암)	개	4	
4	LCD	HB16209	개	1	
5	LCD Connector	Single Connector(14pin, 수)	개	1	
6	LCD Connector	Single Connector(14pin, 암)	개	1	
7	세라믹 Capacitor	0.1uF	개	1	
8	세라믹 Capacitor	18pF	개	2	
9	전해 Capacitor	10uF/16V	개	2	
10	저항	10Ω	개	1	
11	저항	47Ω	개	3	
12	저항	330Ω	개	4	
13	저항	1KΩ	개	1	
14	저항	10KΩ	개	11	
15	Variable Resistor	10KΩ	개	1	
16	FND	FND 507	개	3	
17	LED	Red(5φ)	개	4	
18	LED	Green(5φ)	개	1	
19	Buzzer	Buzzer(DC 5V)	개	1	
20	Crystal	16MHz	개	1	
21	TR	A1015	개	3	
22	Tact Switch	Tact Switch	개	6	
23	Slide Switch	MSL-1C2P	개	1	
24	CDS	소형 CDS	개	1	
25	Power Connector	녹색단자 2pin, 5.0mm	개	1	

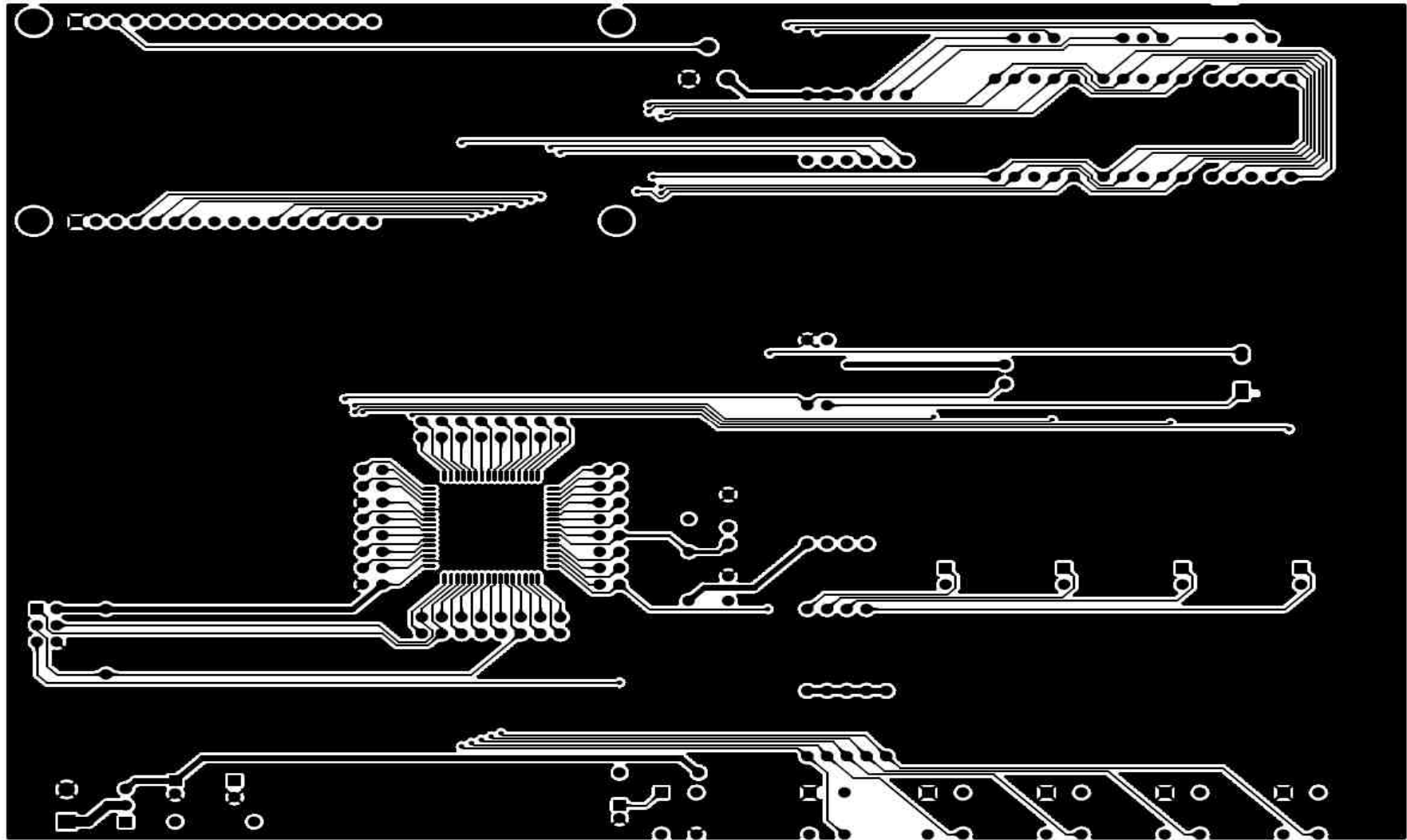
4-2. 지급 재료 목록

[illegible]

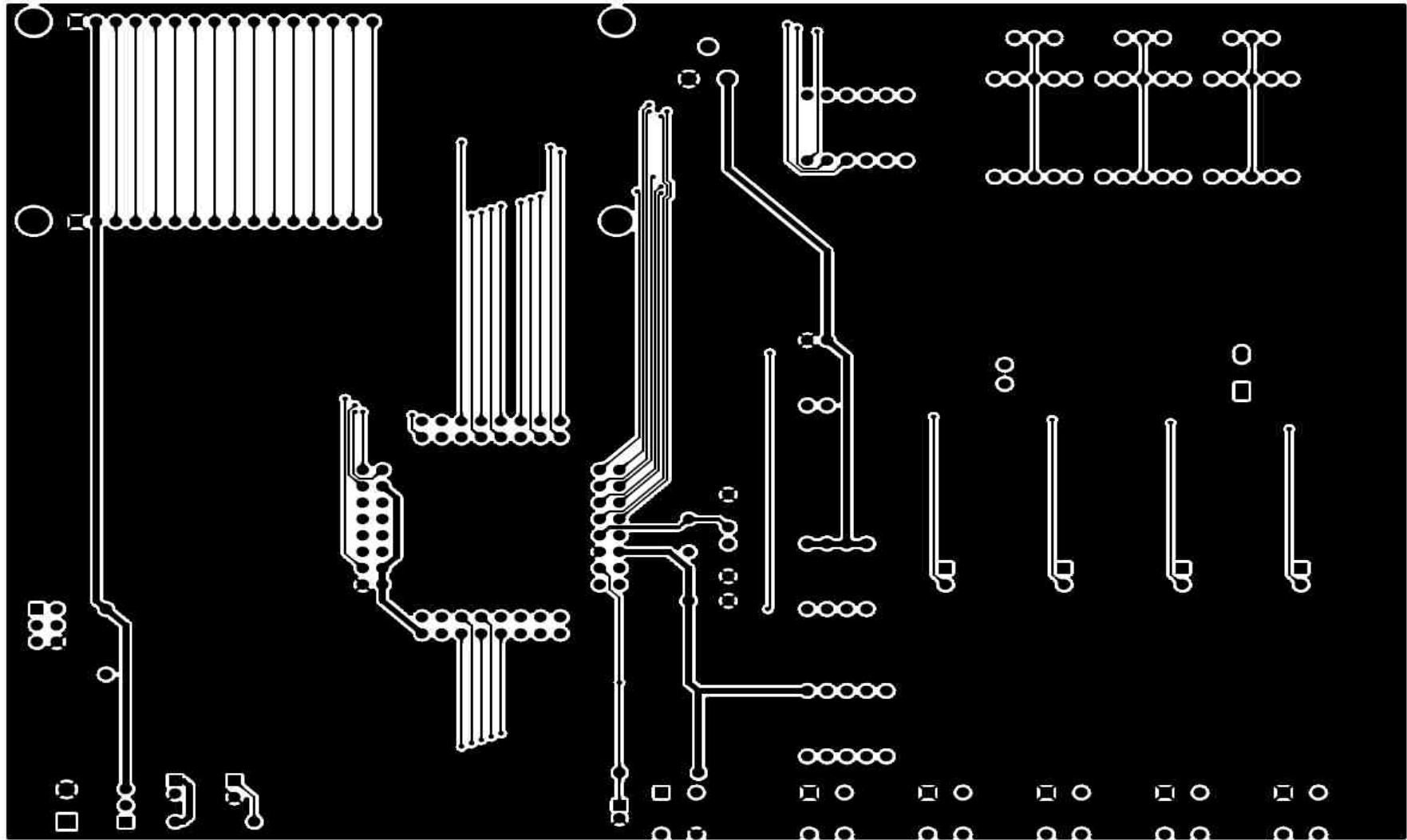
5-1. PCB(Top Silk-Screen)



5-2 PCB(TOP Routing)

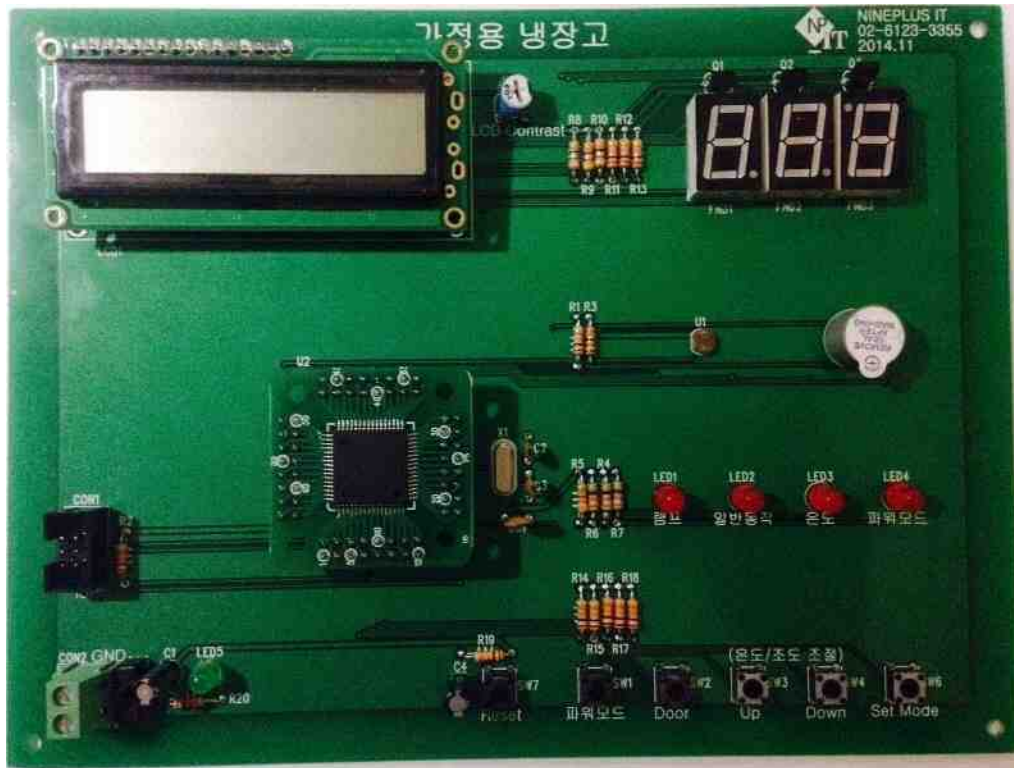


5-3. PCB(BOTTOM Routing)



6. 배포용 소스

1. 전체 사진



2. C Source : main.c

```
#include <avr/io.h>
#include <avr/interrupt.h>
#include <avr/eeprom.h>
#include <util/delay.h>

#include <stdio.h>
#include <string.h>

#include "buzzer.h"
#include "cds.h"
#include "fnd.h"
#include "key.h"
#include "lcd.h"
#include "led.h"

#define EEP_FLAG (unsigned char *)0x01
#define EEP_TEMP (unsigned char *)0x02
#define EEP_DOOR (unsigned char *)0x03

enum {
    NORMAL_MODE,
    POWER_MODE
};

enum {
    CLOSE,
    OPEN
}
```



```

};

extern volatile char key_flag;

volatile char disp_flag;
volatile unsigned char mode;
volatile unsigned char cur_temp, set_temp;
volatile unsigned char door_count, door_state, door_time;
volatile unsigned int door_tick;
volatile unsigned char set_lux, max_lux, min_lux, cur_lux;
volatile unsigned char led_bright;
volatile unsigned char down_flag, down_tick, down_time;

void timer0_init(void)
{
    TCCR0 = 0x05;
    TCNT0 = 256 - 125;
    TIMSK |= 0x01;
}

ISR(TIMER0_OVF_vect)
{
    static unsigned char tick0;

    TCNT0 = 256 - 125;

    tick0++;

    if( tick0 < led_bright )          LED_OUT &= ~LED1;
    else if( tick0 >= led_bright && tick0 < 10 ) LED_OUT |= LED1;
    else if( tick0 == 10 )             tick0 = 0;

    if( door_state ) {
        door_tick++;
        //////////////////////////////////////
        // [문제 4] 냉장고 문이 열려 있는 경우 부저는 아래와 같이 동작되도록 프로그램
        //      하시오.
        //////////////////////////////////////
        if( (door_tick % 500) == 0 ) {

        }

        if( (door_tick % 1000) == 0 ) {

        }

        if( door_tick == 2000 ) {
            door_tick = 0;

```

```

        if( cur_temp < 14 )cur_temp++;
        else {
            disp_flag = 1;
            TIMSK |= 0x40;
        }
    }
}

void timer1_init(void)
{
    TCCR1A = 0x00;
    TCCR1B = 0x03;
    TCCR1C = 0x00;
    TCNT1 = 65536 - 25000;
    TIMSK |= 0x04;
}

ISR(TIMER1_OVF_vect)
{
    TCNT1 = 65536 - 25000;

    if( (cur_temp != set_temp) && !door_state ) {
        if( !down_flag ) {
            down_flag = 1;
            down_tick = 0;
        }
        else {
            down_tick++;
            if( down_tick % 5 == 0 ) disp_flag ^= 0x01;

            if( down_tick >= down_time ) {
                down_tick = 0;
                if( cur_temp > set_temp ) cur_temp--;
                else if( cur_temp < set_temp ) cur_temp++;
            }
        }
    }
    else if( (cur_temp == set_temp) && !door_state ) {
        disp_flag = 1;
        TIMSK |= 0x40;
    }
}

void mcu_init(void)
{
    buzzer_init();
    cds_init();
    fnd_init();
    key_init();
    led_init();
    lcd_init();
}

void variable_init(void)

```

```

{
    if( eeprom_read_byte(EEP_FLAG) == 107 ) {
        set_temp = eeprom_read_byte(EEP_TEMP);
        door_count = eeprom_read_byte(EEP_DOOR);
    }
    else {
        set_temp = 4;
        door_count = 0;
        eeprom_write_byte(EEP_FLAG, 107);
        eeprom_write_byte(EEP_TEMP, 4);
        eeprom_write_byte(EEP_DOOR, 0);
    }

    disp_flag = 0;

    mode = NORMAL_MODE;

    cur_temp = 12;

    set_lux = 0;
    max_lux = 99;
    min_lux = 0;
    cur_lux = 0;

    down_time = 20;

    door_state = CLOSE;
}

void start_refrigerator(void)
{
    lcd_gotoxy(0, 0);
    lcd_string(" Refrigerator ");
    lcd_gotoxy(0, 1);
    lcd_string("      System      ");

    for( int i = 0; i < 3; i++ ) {
        SEG_OUT = 0xC0;
        DIG_OUT = 0x04;
        LED_OUT = 0x00;
        _delay_ms(500);

        SEG_OUT = 0xFF;
        DIG_OUT = 0x07;
        LED_OUT = 0x0F;
        _delay_ms(500);
    }

    disp_flag = 1;
}

/////////////////////////////////////////////////////////////////
// [문제 1] LCD의 CGRAM을 이용하여 LCD에 온도를 나타내는 “℃”의 ‘°’가 표시
//           되도록 프로그램 하시오.
/////////////////////////////////////////////////////////////////
void lcd_cgram(void)

```

```

{

}

void init_screen(void)
{
    lcd_gotoxy(0, 0);
    printf("Mode:Nor  Lux:%02d", cur_lux);
    lcd_gotoxy(0, 1);
    printf("Temp:%02d C Door:C", set_temp);
    lcd_gotoxy(7, 1);
    lcd_data_write(0x00);
}

void check_condition(void)
{
    if( mode == NORMAL_MODE ) {
        if( cur_temp != set_temp ) led_light(LED2 | LED3);
        else led_light(LED2);
    }
    else {
        if( cur_temp == set_temp ) {
            down_time = 20;

            lcd_gotoxy(0, 0);
            lcd_string("Mode:Nor  Lux:00");

            LED_OUT |= LED4;

            mode = NORMAL_MODE;
        }
    }

    if( set_lux && !door_state ) {
        cur_lux = (unsigned char)(cds_read() / 10.23);

        lcd_gotoxy(14, 0);
        printf("%02d", cur_lux);

        //////////////////////////////////////
        // [문제 2] "LED1"의 밝기는 아래와 같이 동작되도록 프로그램 하시오.
        //      (타이머 이용)
        //////////////////////////////////////

    }

    if( door_state ) {
        lcd_gotoxy(6, 1);
        printf("%02d", door_time);
    }

```

```

    }
}

void set_temperature(void)
{
    lcd_gotoxy(6, 1);
    printf("%02d", set_temp);
}

void set_luxlight(void)
{
    lcd_gotoxy(5, 1);
    printf("%02d", max_lux);
    lcd_gotoxy(13, 1);
    printf("%02d", min_lux);
}

unsigned char set_mode(unsigned char keyin)
{
    char key = keyin;
    char set_mode = 0, lux, lux_flag = 0;
    enum {
        NONE,
        SET_TEMP,
        SET_LUX
    };

    lcd_gotoxy(0, 0);
    lcd_string("    Set_Mode    ");
    lcd_gotoxy(0, 1);
    lcd_string("Press Key SW1-2 ");

    while( 1 ) {
        key = getkey(key);
        if( key_flag ) {
            switch( key ) {
                case KEY_POWER :
                    if( set_mode == NONE ) {
                        lcd_gotoxy(0, 0);
                        lcd_string("Set Temperature ");
                        lcd_gotoxy(0, 1);
                        printf("      %02d C      ", set_temp);
                        lcd_gotoxy(8, 1);
                        lcd_data_write(0);

                        set_mode = SET_TEMP;
                        TIMSK &= ~0x04;
                    }
                    break;

                case KEY_DOOR :
                    if( set_mode == NONE ) {
                        lcd_gotoxy(0, 0);
                        lcd_string(" Set Light Lux  ");
                        lcd_gotoxy(0, 1);
                        printf(" Max:%02d  Min:%02d ", max_lux, min_lux);

```

```

        set_mode = SET_LUX;
    }
    break;

case KEY_UP :
    if( set_mode == SET_TEMP ) {
        if( set_temp < 10 ) set_temp++;
        else buzzer_on();
    }
    else if( set_mode == SET_LUX ) {
        if( lux_flag == 99 ) lux_flag = 0;
        else lux_flag = 99;
    }
    break;

case KEY_DOWN :
    if( set_mode == SET_TEMP ) {
        if( set_temp > 0 ) set_temp--;
        else buzzer_on();
    }
    else if( set_mode == SET_LUX ) {
        if( lux_flag == 1 ) lux_flag = 0;
        else lux_flag = 1;
    }
    break;

case KEY_SET :
    if( set_mode == SET_TEMP ) {
        TIMSK |= 0x04;
        eeprom_write_byte(EEP_FLAG, 107);
        eeprom_write_byte(EEP_TEMP, set_temp);
    }
    else if( set_mode == SET_LUX ) set_lux = 1;
    set_mode = 0;
    return key;
}

}

if( set_mode == SET_TEMP ) set_temperature();
else if( set_mode == SET_LUX ) {
    lux = (unsigned char)(cds_read() / 10.23);
    if( lux_flag == 99 ) max_lux = lux;
    else if( lux_flag == 1 ) min_lux = lux;
    set_luxlight();
}

if( cur_temp != set_temp ) led_light(LED2 | LED3);
else led_light(LED2);
}

}

void door_condition(void)
{
    door_state ^= 0x01;

```

```

if( door_state == OPEN ) {
    if( door_count < 99 ) door_count++;
    eeprom_write_byte(EEP_FLAG, 107);
    eeprom_write_byte(EEP_DOOR, door_count);

    lcd_gotoxy(0, 0);
    lcd_string(" Door Open ");
    lcd_gotoxy(0, 1);
    printf(" Time:00 Use:%02d ", door_count);

    door_time = 0;
    door_tick = 0;
}
else {
    BUZZER_OUT |= 1 << BUZZER;
    init_screen();
}
}

void power_mode(void)
{
    if( cur_temp <= set_temp ) {
        for( int i = 0; i < 3; i++ ) {
            lcd_gotoxy(0, 0);
            lcd_string(" Not Power Mode ");
            lcd_gotoxy(0, 1);
            lcd_string("Please Set Temp!");

            buzzer_on();

            lcd_command_write(0x01);
            _delay_ms(500);
        }

        init_screen();
    }
    else {
        //////////////////////////////////////
        // [문제 3]
        // (가) 현재 온도가 설정 온도보다 큰 경우 다음과 같이 동작되도록 프로그램
        // 하시오.
        //////////////////////////////////////

        mode = POWER_MODE;
    }
}

```

```

int main(void)
{
    unsigned char key = 0;

    mcu_init();
    variable_init();
    start_refrigerator();

    sei();

    timer0_init();
    timer1_init();
    fdevopen((void *)lcd_data_write, 0);

    lcd_cgram();
    init_screen();

    while( 1 ) {
        key = getkey(key);
        if( key_flag ) {
            switch( key ) {
                case KEY_POWER :
                    power_mode();
                    break;

                case KEY_DOOR :
                    door_condition();
                    break;

                case KEY_SET :
                    key = set_mode(key);
                    init_screen();
                    break;
            }
        }

        check_condition();
    }

    return 0;
}

```