
2020 한국폴리텍대학 (다기능과정) IT융합전자회로 설계 및 제작 경진대회

제 3 과제

Project Name : MICOM PROGRAMMING

제한 시간 : 5시간



후원 : 학교법인 한국폴리텍대학

협찬 : 한국폴리텍대학 춘천캠퍼스, 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. 배포된 소스를 참조하여 프로그램 요구조건 【문제1】 ~ 【문제3】 을 프로그램하고, 컴파일, 라이팅하여 동작시킨 후 자신의 선수번호를 적고, 저장하시오.

C:\ Gate_xxx..C , C:\ Gate_xxx.asm (xxx : 자신의 선수번호)

다. 동작요구사항

- ※ 이 작품은 지하철 무인요금 시스템을 실현한 것이다.
- ※ 문제에서 제시하는 ‘정 방향’의 방향은 역 순서대로이다.
- ※ LED 배치와 정거장 명은 다음과 같다.



(1) 전원을 인가하거나 S5를 눌렀을 시 다음과 같이 동작되게 하시오.

(가) LCD

		W	e	l	c	o	m	e		t	o				
G	a	n	g	n	a	m		S	t	a	t	i	o	n	!

(나) LED는 다음 순서대로 0.3초 간격으로 점멸하게 하시오.

LED2→LED3→LED4→LED5→LED9→LED13→LED17→LED16→LED12→
LED8→LED7→LED11→LED15→LED14→LED10→LED6

(예)



(다) FND는 소등되게 하시오.

(2) “요구사항 (1)”이 완료된 후 다음과 같이 동작되게 하시오.

(가) LCD

	S	e	l	e	c	t		B	u	t	t	o	n	!	
S	1	:	A	d	u	l	t		S	2	:	K	i	d	

(나) LED는 소등되게 하시오.

(다) FND는 소등되게 하시오.

(3) “요구사항 (2)”의 상태에서 ‘S1’ 또는 ‘S2’를 눌렀을 시 다음과 같이 동작되게 하시오.

(가) ① LCD(S1 눌렀을시)

Y	o	u		s	e	l	e	c	t		A	d	u	l	t

② LCD(S2 눌렀을시)

Y	o	u		s	e	l	e	c	t		K	i	d		

(나) LED는 소등되게 하시오.

(다) FND는 소등되게 하시오.

[문제 1] 출발역과 도착역 선택 및 요금 확인

(4) “요구사항 (3)”의 상태에서 0.5초 후 다음과 같이 동작되게 하시오.

(가) LCD(초기값은 강남역이다.)

	S	t	a	r	t		s	t	a	t	i	o	n		
S	t	a	t	i	o	n		n	u	m	:		0	1	

※ 역 번호와 LED의 관계는 다음과 같다.

역 번호	역 명	LED	역 번호	역 명	LED
01	강남	LED2	09	당산	LED12
02	교대	LED3	10	홍대	LED8
03	사당	LED4	11	충정로	LED7
04	서울대	LED5	12	을지로	LED11
05	대림	LED9	13	왕십리	LED15
06	신도림	LED13	14	건대	LED14
07	문래	LED17	15	잠실	LED10
08	영등포	LED16	16	삼성	LED6

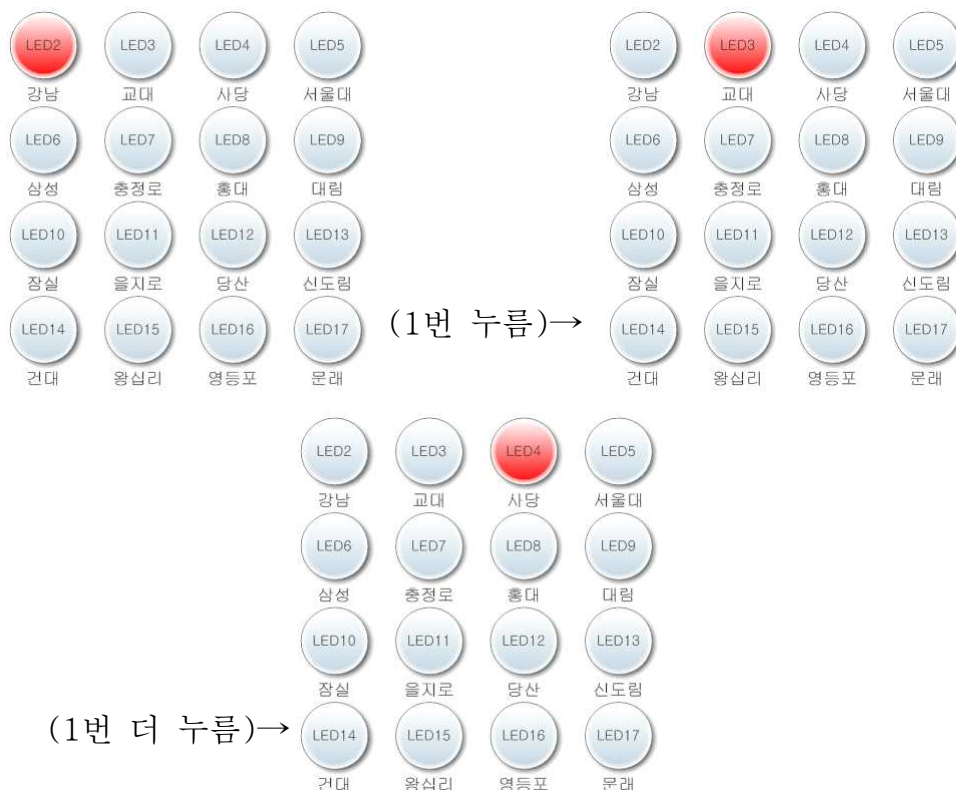
(예) S1(정방향) 또는 S2(역방향)를 눌러 왕십리역을 선택하면(LED15점등)

S	t	a	r	t	S	t	a	t	i	o	n		
S	t	a	t	i	o	n	N	u	m	:		1	3

(나) FND는 소등되게 하시오.

(다) S1을 1번 누를 때마다 1정거장씩 정방향으로 선택되고, S2를 1번 누를 때마다 1정거장씩 역방향으로 선택되게 하시오.

(예) 강남(LED2)역에서 S1을 두번 누르면 사당(LED4)역이 선택되어 진다.



(라) 출발역을 선택한 후(해당 역 LED가 점등상태) S3을 누르면 도착역을 선택하게 하시오. (선택방법은 “요구사항 (4)의 (가), (다)와 같이 선택하시오.”)

LCD)

	E	n	d				S	t	a	t	i	o	n		
S	t	a	t	i	o	n		N	u	m	:		1	3	

(5) “요구사항 (4)”의 상태에서 다시 S3을 누르면 다음과 같이 동작되게 하시오.

(가) LCD

S	t	a	r	t	:	X	X		S	t	a	t	i	o	n
E	n	d			:	Y	Y		S	t	a	t	i	o	n

'XX' 출발역, 'YY' 도착역

(예) S1 또는 S2를 이용하여 출발지를 LED2(강남, 역번호 01) 도착지를 LED12(당산, 역번호 09)를 선택하고 S3을 누르면

S	t	a	r	t	:	0	1		S	t	a	t	i	o	n
E	n	d			:	0	9		S	t	a	t	i	o	n

(나) LED는 출발역과 도착역에 해당되는 LED가 점등되게 하시오.

(예) 출발역이 강남(LED2)이고 도착역이 당산(LED12)이면



(다) FND는 소등되게 하시오.

(라) 출발역과 도착역이 같은 상태에서 S3을 누르면 다음과 같이 되게 하시오.

S	t	a	r	t	:	0	1		S	t	a	t	i	o	n
E	n	d			:	0	1		S	t	a	t	i	o	n

↓ 0.5초 후

				E	r	r	o	r	!	!					
R	e	t	u	r	n		T	o		F	i	r	s	t	!

↓ 0.5초 후

	S	e	l	e	c	t		B	u	t	t	o	n	!	
S	1	:	A	d	u	l	t		S	2	:	K	i	d	

(6) “요구사항 (5)”에서 출발역과 도착역이 다른 역으로 설정되면 0.5초 후에 다음과 같이 동작되게 하시오.

(가) LCD는 다음과 같이 표시되게 하시오.

D	i	s	t	a	n	c	e	:	X						
F	e	e		:		Y	Y	Y	Y		W	o	n		

① 'X' 출발역과 도착역의 최단 거리이다.

거리 계산은 역 번호로 계산한다.

거리 = 출발역과 도착역 사이의 역의 개수 + 1

(예) 출발역이 강남역(01)이고, 도착역이 당산역(09)이면 그 사이의 역의 개수가 7개 이므로 7+1=8이 된다.

② 'YYYY' 는 지불해야 할 요금이다. 계산식은 다음과 같다.

Adult : YYYY= 500 × 거리

Kid : YYYY= 300 × 거리

(예) 출발역이 강남역(01)이고, 도착역이 당산역(09)일 때 요금은?

– 거리는 8이 된다.

– 어른요금은 $500 \times 8 = 4,000$ 원

– 어린이요금은 $300 \times 8 = 2,400$ 원

(나) LED는 “요구사항 (5)”의 상태를 유지하게 하시오.

(다) FND는 소등되게 하시오.

[문제 2]요금 지불 및 운행

(7) “[문제 1]”이 완료된 후 1초 후에 다음과 같이 동작되게 하시오.

(가) LCD는 다음과 같이 표시되게 하시오.

I	n	s	e	r	t		M	o	n	e	y	!	!		
F	e	e	:	X	X	X	X	-	Y	Y	Y	Y		Z	Z

- ① 'XXXX' 는 지불해야 할 요금이 표시되게 하시오.
- ② VR1의 저항 값에 따라 100원씩 증감하도록 하시오.
- ③ ‘YYYY’값이 ‘XXXX’값보다 많거나 같으면 ‘ZZ’에 ‘OK’라고 표시되며, 그 전에는 공백이 되게 하시오.

(나) 'YYYY' >= 'XXXX' 'ZZ'에 'OK' 표시가 되고, S3을 누르면 'QQQQ'에 거스름돈이 표시되게 하시오.(단, 거스름돈이 없을 경우 (다)항으로 넘어간다.)

H	e	r	e		I	s		C	h	a	n	g	e	.	
C	h	a	n	g	e		:		Q	Q	Q	Q			

- ① 'QQQQ'는 지불한 돈에서 요금을 뺀 금액이다.
- (다) ‘(가)’ 또는 ‘(나)’항의 요구사항이 실행되면(요금을 정상적으로 지불하면) 1초 후에 다음과 같이 동작되게 하시오.

		[S	4]		G	o	!						
		D	i	s	t	a	n	c	e		:		X		

- (라) 'X' 의 값은 출발역부터 도착역까지의 거리를 나타내시오.
- (마) LED는 “요구사항 (5)”의 상태를 유지하게 하시오.
- (바) FND는 소등되게 하시오.

(8) S4를 누르면 다음과 같이 동작되게 하시오.

(가) LCD는 다음과 같이 표시되게 하시오.

T	r	a	i	n		I	s		G	o	i	n	g	.	.
N	N	N	N	N		:		X		M	i	n	u	t	e

- ① 'X' 는 도착역까지 남은 시간이 표시되게 하시오. 단, 한 정거장당 1분으로 하며 동작상의 1초를 실제시간 1분이라고 가정한다.
- (나) LED는 출발지의 LED만 표시되고 도착지의 LED는 소등되며, 한 정거장당 1초 간격으로 도착지로 LED가 점멸하며 이동되게 하시오. (현재 역만 점등함)

LED이동 방향은 거리가 짧은 쪽으로 이동하고 같은 거리일 경우 정방향으로 이동되게 하시오.

(다) FND1에는 출발역 부터 도착역까지 이동한 역의 거리를 출력하고, FND2는 LCD에 표시되는 'X'값이 표시되게 하시오.

(라) 'NNNN' 부분은 LCD에는 다음과 같이 표시하시오.

① 정방향일 경우

T	r	a	i	n		I	s		G	o	i	n	g	.	.
-	-	-	-	>		:		X		m	i	n	u	t	e

② 역방향일 경우

T	r	a	i	n		I	s		G	o	i	n	g	.	.
<	-	-	-	-		:		X		m	i	n	u	t	e

(마) 도착지에 도착하면 2초 후에 FND와 LED를 소등하고 “요구사항 (2)”의 상태로 돌아가게 하시오.

(예) 출발지는 강남(LED2)역이고 도착지는 서울대(LED5)이면

LCD	LED	FND1	FND2
<div> <div>T r a i n I s G o i n g . .</div> <div>- - - - > : 3 M i n u t e</div> </div>	<div> <div>LED2 LED3 LED4 LED5</div> <div>강남 교대 사당 서울대</div> <div>LED6 LED7 LED8 LED9</div> <div>삼성 충정로 홍대 대림</div> <div>LED10 LED11 LED12 LED13</div> <div>잠실 을지로 당산 신도림</div> <div>LED14 LED15 LED16 LED17</div> <div>건대 왕십리 영등포 문래</div> </div>	<div>0</div>	<div>3</div>
↓ 1초 후			
<div> <div>T r a i n I s G o i n g . .</div> <div>- - - - > : 2 M i n u t e</div> </div>	<div> <div>LED2 LED3 LED4 LED5</div> <div>강남 교대 사당 서울대</div> <div>LED6 LED7 LED8 LED9</div> <div>삼성 충정로 홍대 대림</div> <div>LED10 LED11 LED12 LED13</div> <div>잠실 을지로 당산 신도림</div> <div>LED14 LED15 LED16 LED17</div> <div>건대 왕십리 영등포 문래</div> </div>	<div>1</div>	<div>2</div>
↓ 1초 후			
<div> <div>T r a i n I s G o i n g . .</div> <div>- - - - > : 1 M i n u t e</div> </div>	<div> <div>LED2 LED3 LED4 LED5</div> <div>강남 교대 사당 서울대</div> <div>LED6 LED7 LED8 LED9</div> <div>삼성 충정로 홍대 대림</div> <div>LED10 LED11 LED12 LED13</div> <div>잠실 을지로 당산 신도림</div> <div>LED14 LED15 LED16 LED17</div> <div>건대 왕십리 영등포 문래</div> </div>	<div>2</div>	<div>1</div>
↓ 1초 후			
<div> <div>T r a i n I s A r r i v e d</div> <div>Th a n k s A L o t !</div> </div>	<div> <div>LED2 LED3 LED4 LED5</div> <div>강남 교대 사당 서울대</div> <div>LED6 LED7 LED8 LED9</div> <div>삼성 충정로 홍대 대림</div> <div>LED10 LED11 LED12 LED13</div> <div>잠실 을지로 당산 신도림</div> <div>LED14 LED15 LED16 LED17</div> <div>건대 왕십리 영등포 문래</div> </div>	<div>3</div>	<div>0</div>
↓ 2초 후			
FND와 LED를 점등하고 “요구사항 (2)”의 상태로 돌아간다.			

[문제3] 통신

(9) 아래와 같이 통신 속도를 맞추어 출력 양식에 맞춰 실시간으로 출력하시오.

(통신 속도 : 9600bps, 데이터:8bit, 패리티: 없음, 정지비트:1Bit)

-Time/m : -Distance : 요구사항 (8)항 실행시 동작하시오.

Start Station :

End Station :

Now Station :

Money :

☐ 음영 처리된 부분은 실시간으로 변동되도록 한다.

출력 양식)

이름	출력	출력 예
Time/m	남은 시간	2
Distance	남은 역	2
Start Station	출발역	01
End Station	도착역	03
Now Station	현재 역	01
Money	지불할 돈	0600

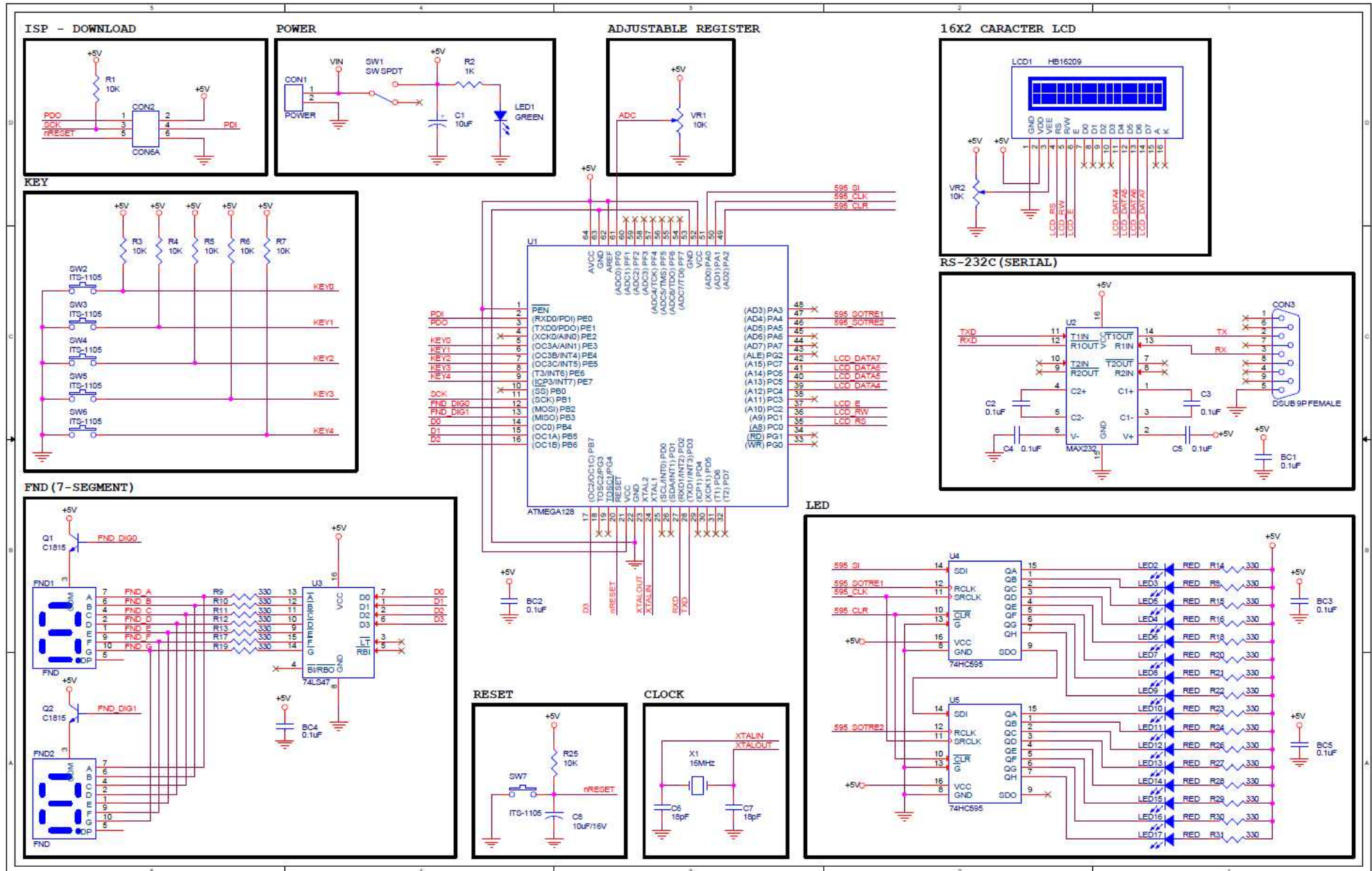
(현재 출발역은 01번 역이고 도착역은 03번 역으로 설정되어 있을 때.)

2. 유의사항

가. 안전사고에 유의하시오.

나. 심사위원의 지시에 따라 작업을 진행하시오.

3. 회로도



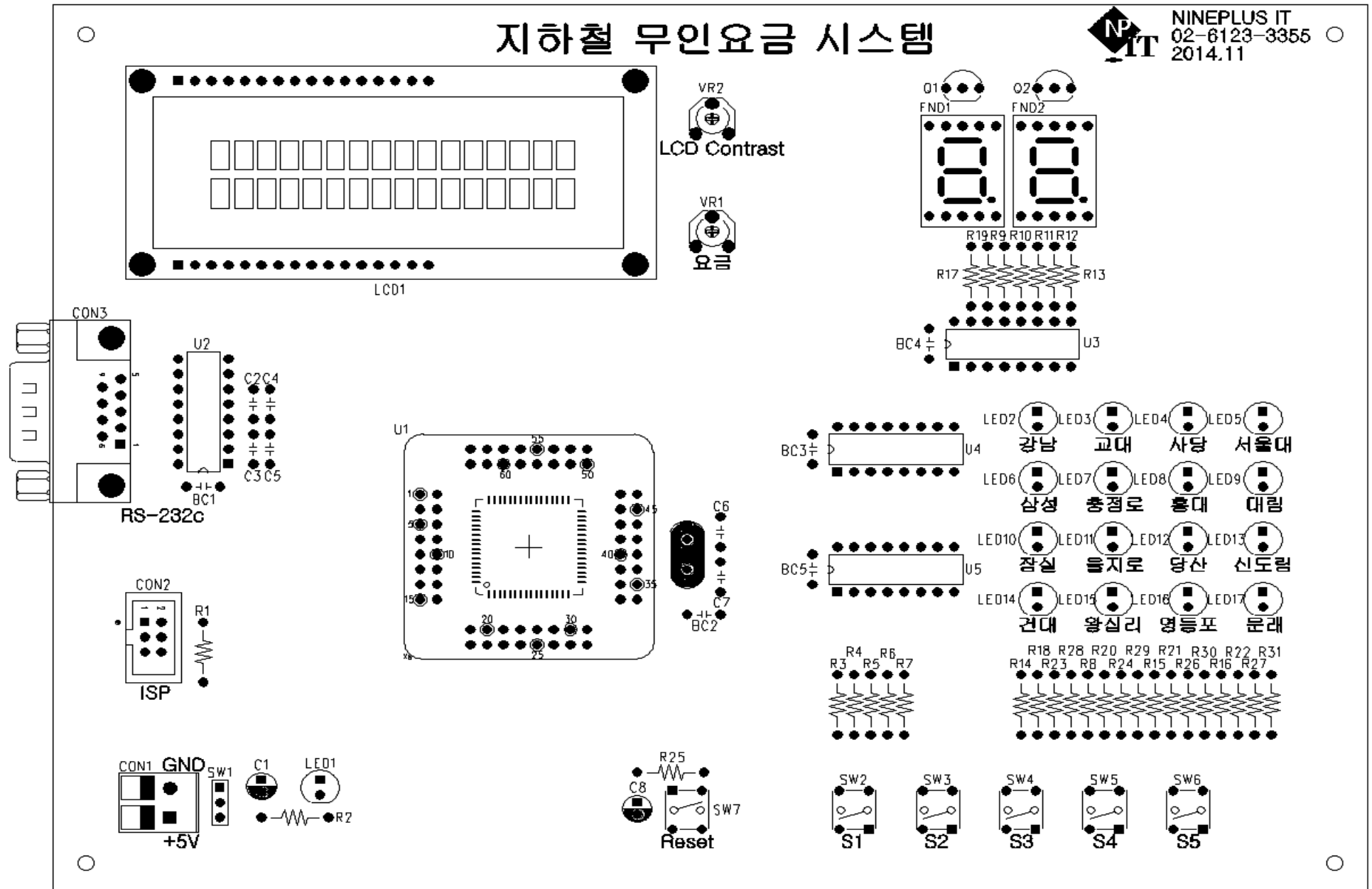
4-1. 지급 재료 목록

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	ISP Connector	Dual Header 3x2, 수	개	1	
5	Power Connector	녹색단자 2pin, 5.0mm	개	1	
6	Slide Switch	MSL-1C2P	개	1	
7	Crystal	16MHz	개	1	
8	세라믹 Capacitor	0.1uF	개	9	
9	세라믹 Capacitor	18pF	개	2	
10	전해 Capacitor	10uF/16V	개	2	
11	저항	330Ω	개	23	
12	저항	1KΩ	개	1	
13	저항	10KΩ	개	7	
14	가변저항	10KΩ	개	2	
15	LED	Red(5φ)	개	16	
16	LED	Green(5φ)	개	1	
17	FND	FND 507	개	2	
18	LCD	HB16209 5V	개	1	
19	LCD Connector	Single Connector(14pin, 수)	개	1	
20	LCD Connector	Single Connector(14pin, 암)	개	1	
21	TR	C1815	개	2	

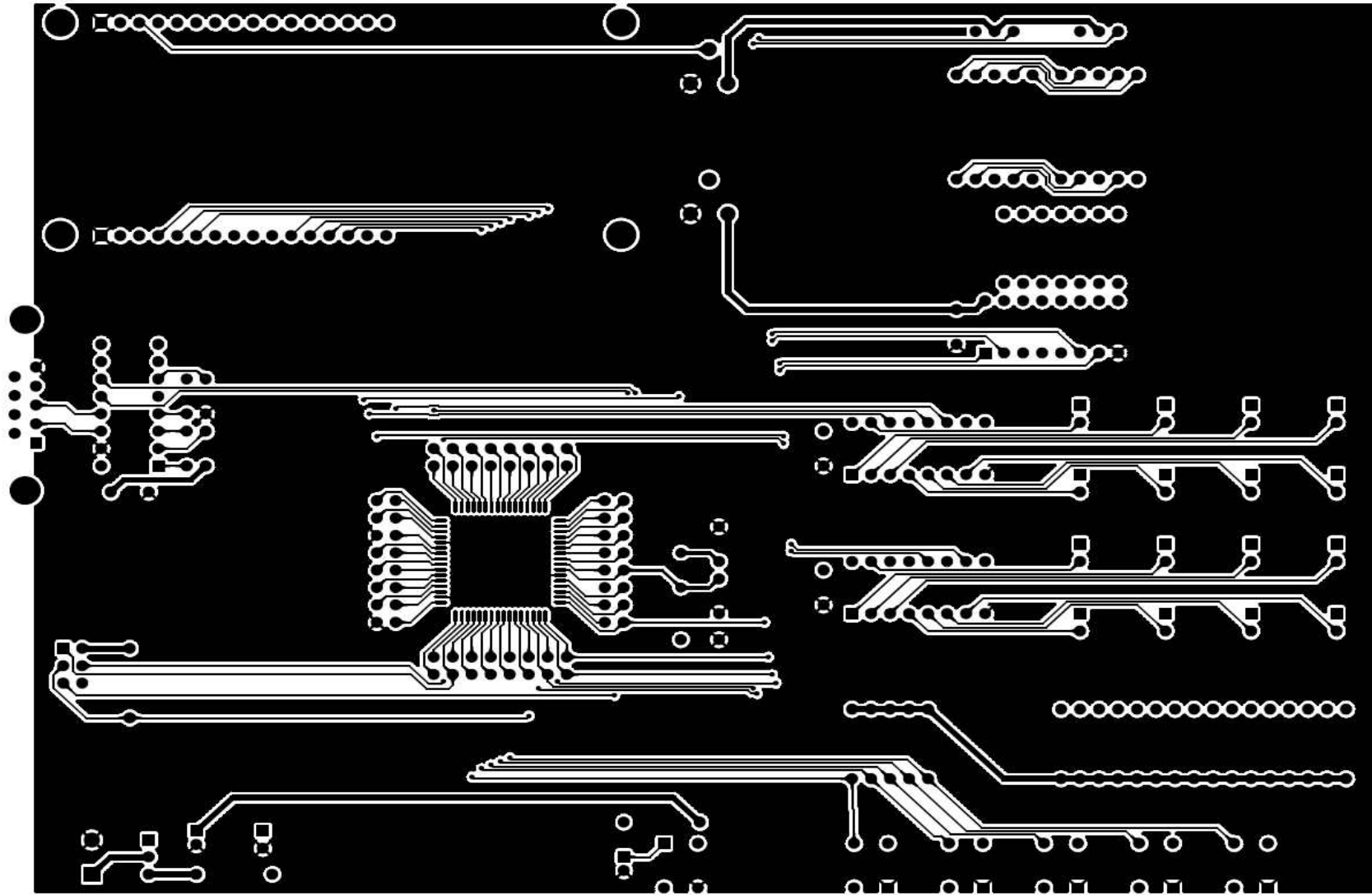
4-2. 지급 재료 목록

일련 번호	재 료 명	규 격(치수)	단위	수량	비 고
22	IC	MAX232	개	1	
23	IC	74LS47	개	1	
24	IC	74HC595	개	2	
25	IC Socket	16PIN(DIP)	개	4	
26	DSUB Connector	DSUB 9P Female	개	1	
27	Tact Switch	Tact Switch	개	6	
28	서포트	3 \varnothing x5mm	개	4	
29	너트	3 \varnothing	개	4	
30	PCB	Bare PCB	장	1	
31	PCB	CPU Module PCB	장	1	
32					
33					
34					
35					
36					
37					
38					
39					
40					
41					

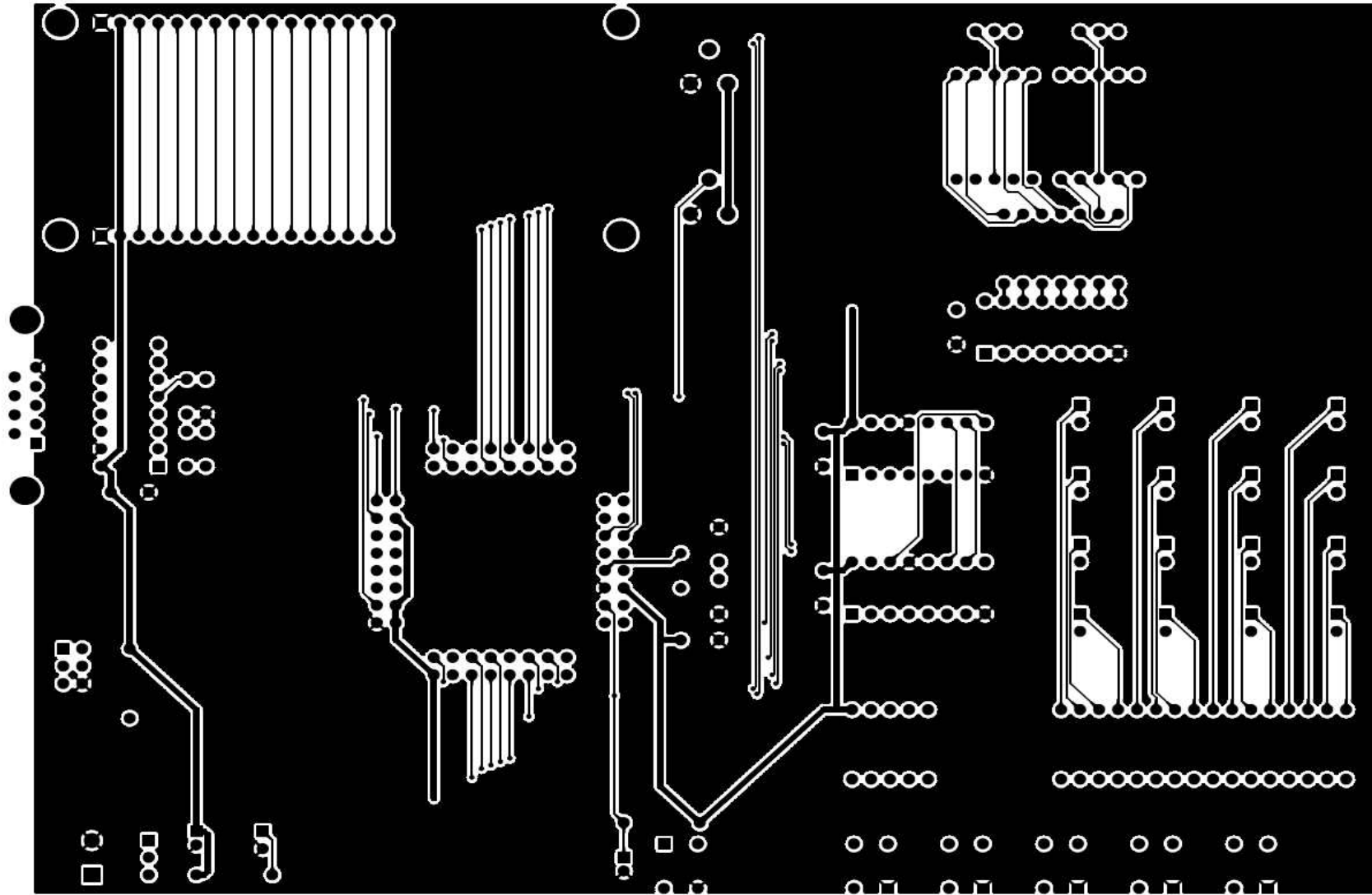
5-1. PCB(Top Silk-Screen)



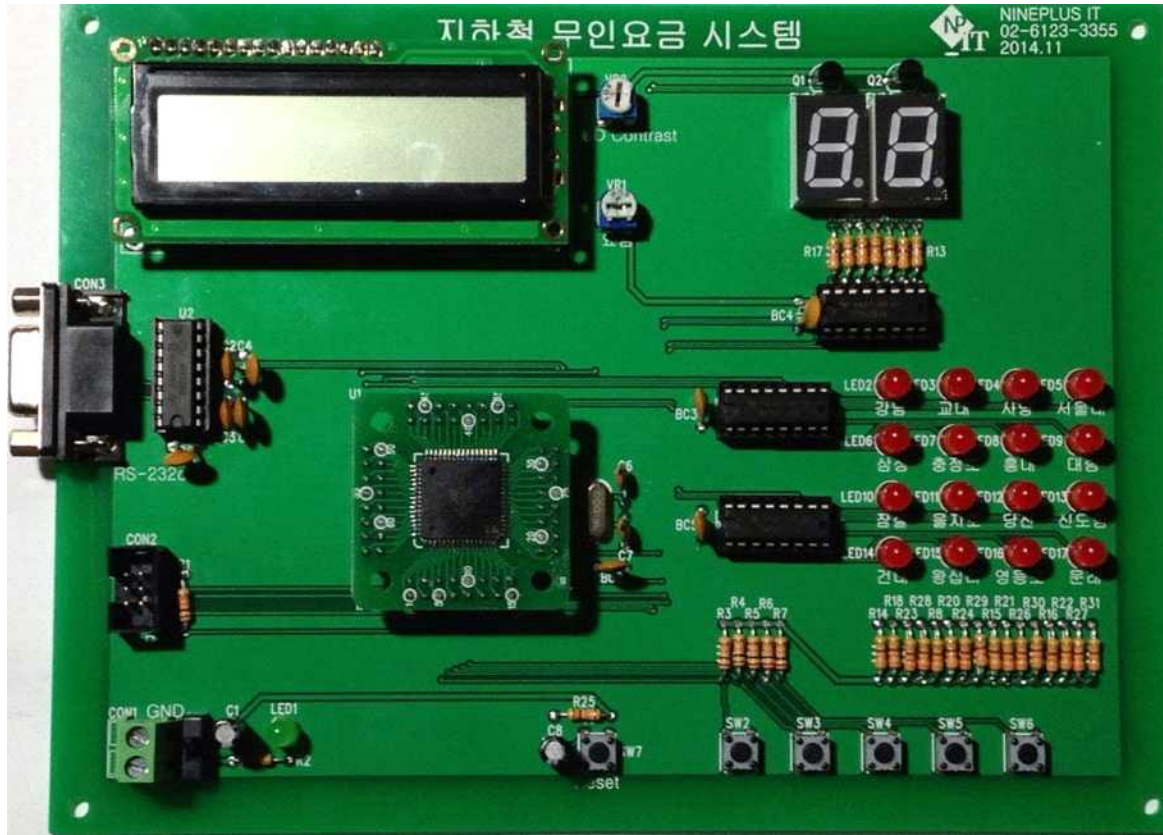
5-2. PCB(TOP Routing)



5-3. PCB(BOTTOM Routing)



6. 완성된 전체사진



7. 배포용 소스(C Source : main.c)

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

#include <stdio.h>

#include "key.h"
#include "fnd.h"
#include "var.h"
#include "lcd.h"
#include "serial.h"
#include "led.h"

#define ON      1
#define OFF    0

#define FORWARD    0
#define REWARD    1

enum {
    NONE,
    ADULT,
    KID
};
```

```

enum {
    SELECT_TICKET_TYPE,
    SELECT_START_STATION,
    SELECT_END_STATION,
    CALCULATE_FEE,
    TRAIN_GO
};

extern volatile char key_flag;
extern volatile char fnd_flag;
extern volatile unsigned char fnd_buf[2];
extern volatile int money;

unsigned int led_order[16] = {
    0x0001, 0x0002, 0x0004, 0x0008, 0x0080, 0x0800, 0x8000, 0x4000,
    0x0400, 0x0040, 0x0020, 0x0200, 0x2000, 0x1000, 0x0100, 0x0010
};
volatile char state;
volatile char ticket_type, direction;
volatile int start_station, end_station, distance;
volatile int fee;
int fee_tbl[2] = { 300, 500 };

void mcu_init(void)
{
    key_init();
    fnd_init();
    var_init();
    lcd_init();
    serial_init(B9600);
    matrix_init();
}

void variable_init(void)
{
    fnd_off();

    ticket_type = NONE;
    start_station = NONE;
    end_station = NONE;

    distance = 0;
    fee = 0;
}

void main_screen(void)
{
    fnd_off();
    matrix_alloff();

    lcd_gotoxy(0, 0);
    lcd_string(" Select Button! ");
    lcd_gotoxy(0, 1);
    lcd_string("S1:Adult  S2:Kid");
}

void start_screen(void)

```

```

{
    int i;

    lcd_string(" Welcome To ");
    lcd_gotoxy(0, 1);
    lcd_string("Gangnam Station!");

    //////////////////////////////////////
    // [문제 1] LED의 순차적 점멸
    // 나. LED : 다음 순서대로 0.3초 간격으로 점멸하게 하시오.
    // LED1→LED2→LED3→LED4→LED8→LED12→LED16→LED15
    // →LED11→LED7→LED6→LED10→LED14→LED13→LED9→LED5
    //////////////////////////////////////
    for( i = 0; i < 16; i++ ) {

    }

    main_screen();
}

void select_station(char keyin)
{
    char key = keyin;
    int station;

    _delay_ms(500);

    start_station = 1;
    station = start_station;

    distance = 0;
    fee = 0;

    lcd_gotoxy(0, 0);
    lcd_string(" Start Station ");
    lcd_gotoxy(0, 1);
    lcd_string("Station Num : ");

    while( 1 ) {
        key = getkey(key);
        if( key_flag ) {
            switch( key ) {
                //////////////////////////////////////
                // [문제 2] 스위치를 이용하여 LED를 정방향 또는 역방향으로 이동
                // 나. LED
                // SW1를 1번 누를 때마다 1정거장씩 정방향으로 이동되고,
                // SW2을 1번 누를 때마다 1정거장씩 역방향으로 이동되게 하시오.
                //////////////////////////////////////
                case KEY_1 :

                    break;

                case KEY_2 :

```

```

        break;

    case KEY_3 :
        if( state == SELECT_START_STATION ) {
            start_station = station;
            end_station = 1;
            station = end_station;

            lcd_gotoxy(0, 0);
            lcd_string(" End    Station  ");

            state = SELECT_END_STATION;
        }
        else if( state == SELECT_END_STATION ) {
            end_station = station;

            lcd_gotoxy(0, 0);
            printf("Start:%02d Station", start_station);
            lcd_gotoxy(0, 1);
            printf("End    :%02d Station", end_station);

            matrix_led(led_order[start_station - 1] | led_order[end_station - 1]);

            _delay_ms(500);

            if( start_station == end_station ) {
                lcd_gotoxy(0, 0);
                lcd_string("    Error!!    ");
                lcd_gotoxy(0, 1);
                lcd_string("Return To First!");
                _delay_ms(500);

                start_station = NONE;
                end_station = NONE;

                main_screen();

                state = SELECT_TICKET_TYPE;
            }
            else {
////////////////////////////////////
//  [문제 3] 출발역과 도착역 선택 후 최단거리와 요금을 계산하여 LCD에 표시
//      가. LCD
//      거리 : 출발역과 도착역 간의 최단 거리를 계산하여 표시
//      요금 : Adult = 500 X 거리, Kid = 300 X 거리
////////////////////////////////////
                distance = end_station - start_station;
                if( distance < 0 ) {

```

```

        }

```

```

        else {
            if( distance > 8 ) {

                }
            else {

                }
        }

        lcd_gotoxy(0, 0);

        lcd_gotoxy(0, 1);

        _delay_ms(1000);

        lcd_gotoxy(0, 0);
        lcd_string("Insert Money!! ");
        lcd_gotoxy(0, 1);
        printf("Fee:%04d-0000   ", fee);

        state = CALCULATE_FEE;
    }

    return;
}
break;

case KEY_5 :
    wdt_enable(WDTO_30MS);
    while( 1 );
    break;
}
}

matrix_led(led_order[station - 1]);
lcd_gotoxy(14, 1);
printf("%02d", station);
}
}

void train_go(void)
{
    int i, pos = start_station - 1;
    char str[30];

    lcd_gotoxy(0, 0);
    lcd_string("Train Is Going..");

    fnd_flag = ON;
    for( i = 0; i <= distance; i++ ) {
        matrix_led(led_order[pos]);

        if( i < distance ) {

```

```

        lcd_gotoxy(0, 1);
        if( direction == FORWARD ) printf("---> : %d Minutes", distance - i);
        else printf("<--- : %d Minutes", distance - i);
    }
    else {
        lcd_gotoxy(0, 0);
        lcd_string("Train Is Arrived");
        lcd_gotoxy(0, 1);
        lcd_string(" Thanks A Lot! ");
    }

    fnd_buf[0] = i;
    fnd_buf[1] = distance - i;

    //////////////////////////////////////
    //  [문제 5] 통신
    //      출력 양식에 맞춰서 실시간으로 PC 터미널에 출력하시오.
    //      이동시간, 남은거리, 출발역, 도착역, 현재역, 요금을 터미널에 출력
    //////////////////////////////////////
    serial_transmit('\f');

```

```

        if( direction == FORWARD ) {
            if( pos < 15 ) pos++;
            else pos = 0;
        }
        else {
            if( pos ) pos--;
            else pos = 15;
        }

        _delay_ms(1000);
    }
    _delay_ms(1000);

    ticket_type = NONE;
    start_station = NONE;
    end_station = NONE;

    distance = 0;
    fee = 0;

    main_screen();

    state = SELECT_TICKET_TYPE;
}

```

```

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

    mcu_init();
    variable_init();

    sei();

    fdevopen((void *)lcd_data_write, 0);

    start_screen();
    state = SELECT_TICKET_TYPE;

    var_start();

    while( 1 ) {
        key = getkey(key);
        if( key_flag ) {
            switch( key ) {
                case KEY_1 :
                case KEY_2 :
                    if( state == SELECT_TICKET_TYPE ) {
                        state = SELECT_START_STATION;

                        lcd_command_write(0x01);
                        lcd_gotoxy(0, 0);
                        if( key == KEY_1 )        lcd_string("You Select Adult");
                        else if( key == KEY_2 )    lcd_string("You Select Kid  ");

                        ticket_type = key;
                        select_station(key);
                    }
                    break;

                case KEY_3 :
                    if( state == CALCULATE_FEE ) {
                        if( money >= fee ) {
                            if( money > fee ) {
                                lcd_gotoxy(0, 0);
                                lcd_string("Here Is Change. ");
                                lcd_gotoxy(0, 1);
                                printf("Change : %04d  ", money - fee);
                            }
                            _delay_ms(1000);

                            lcd_gotoxy(0, 0);
                            lcd_string(" [S4] Go!!  ");
                            lcd_gotoxy(0, 1);
                            printf(" Distance : %d  ", distance);

                            state = TRAIN_GO;
                        }
                    }
                    break;
            }
        }
    }
}

```



```

        case KEY_4 :
            if( state == TRAIN_GO ) train_go();
            break;

        case KEY_5 :
            wdt_enable(WDTO_30MS);
            while( 1 );
            break;
    }
}

/////////////////////////////////////////////////////////////////
//  [문제 4] 가변저항을 이용하여 지불 요금 표시
//      1. VR1의 ADC 입력값을 지불 금액으로 LCD에 표시 |
//      2. 입력된 금액을 요금과 비교하여 크거나 같으면 'OK'라고 표시하고
//      그 전에는 공백이 되게 하시오.
/////////////////////////////////////////////////////////////////
    if( state == CALCULATE_FEE ) {

        if( money >= fee ) {

            }

        }

    }

return 0;

```