
2023 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 회로기관, 회로도, 동작 설명 사항을 참조하여 제작하시오.

나. 배포되는 소스는 프로그램 요구사항 ① ~ 요구사항 ⑨까지 동작되게 한 것이다.
이 소스를 완성하여 컴파일, 라이팅하여 프로그램 요구사항 ① ~ 요구사항 ⑨까지
동작시키시오.

※ 단, 자신의 컴파일과 맞지 않는 부분은 수정하여 동작시키시오.

다. 배포된 소스를 참조하여 프로그램 [문제 1] ~ [문제 5]를 프로그램하고 컴파일,
라이팅하여 동작시킨 후 자신의 비번호를 적고, 저장하시오.

C:\WSubway Auto Fee XX.C (XX : 자신의 비번호)

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

2. 과제 개요

가. 이 작품은 지하철 무인 요금 시스템을 모델링해 구현한 것이다.

나. 문제에서 제시하는 ‘정방향’의 방향은 역 순서이다.

다. LED 배치와 정거장명은 다음과 같다.



3. 동작 요구 사항

요구사항 1 : 전원을 인가하거나 SW5(Reset)를 눌렀을 시 다음과 같이 동작되게 하시오.

※ 아래 세부 요구사항 가, 나, 다는 동시에 동작함에 유의하시오.

가. LCD

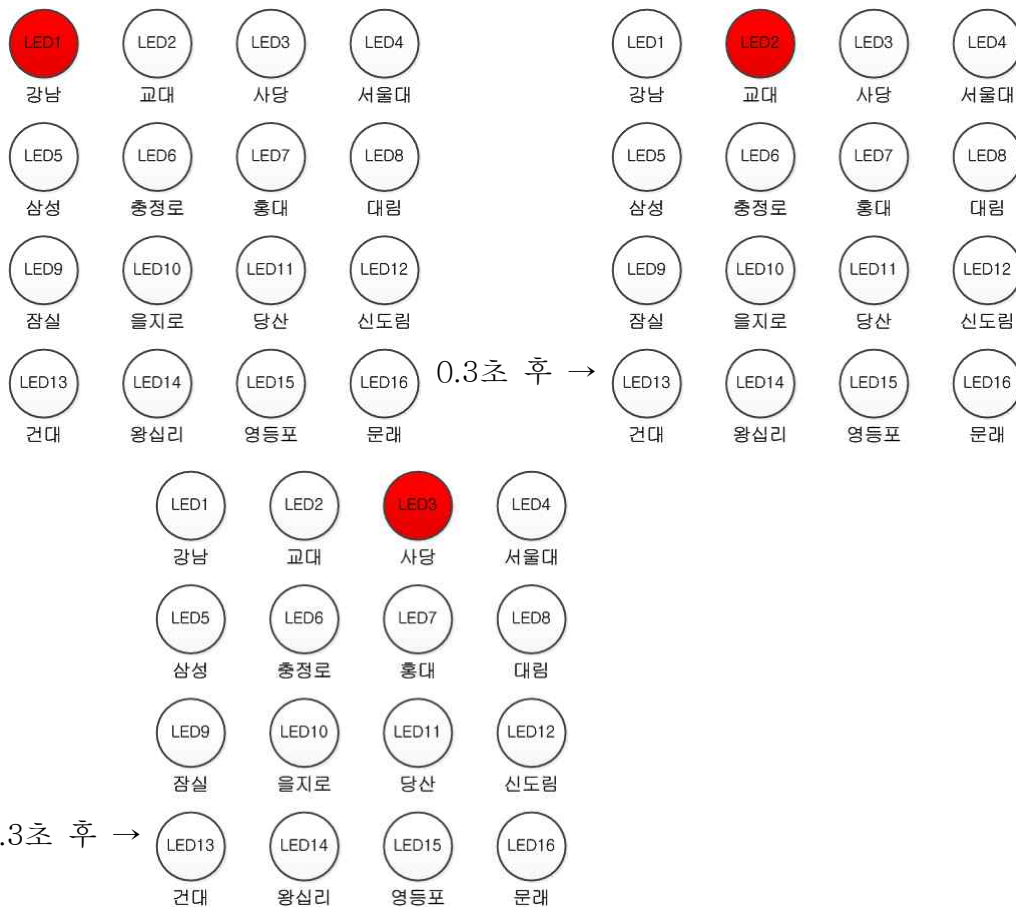
		W	e	l	c	o	m	e		t	o				
G	a	n	g	n	a	m		S	t	a	t	i	o	n	!

[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

(예)



다. FND : 소등되게 하시오.

요구사항 ② : 요구사항 ①이 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. LED: 소등되게 하시오.

다. FND : 소등되게 하시오.

요구사항 ③ : 요구사항 ②의 상태에서 'SW1' 또는 'SW2'을 눌렀을 시 다음과 같이 동작되게 하시오.

가. LCD

(1) SW1을 눌렀을 때

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

(2) SW2를 눌렀을 때

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

나. LED : 소등되게 하시오.

다. FND : 소등되게 하시오.

요구사항 ④ : 요구사항 ③의 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

※ 역번호와 LED의 관계는 다음과 같다.

역 번호	역 명	LED	역 번호	역 명	LED
01	강남	LED1	09	당산	LED11
02	교대	LED2	10	홍대	LED7
03	사당	LED3	11	충정로	LED6
04	서울대	LED4	12	을지로	LED10
05	대림	LED8	13	왕십리	LED14
06	신도림	LED12	14	건대	LED13
07	문래	LED16	15	잠실	LED9
08	영등포	LED15	16	삼성	LED5

(예) SW1(정방향) 또는 SW2(역방향)를 눌러 왕십리역을 선택하면 (LED14 점등)

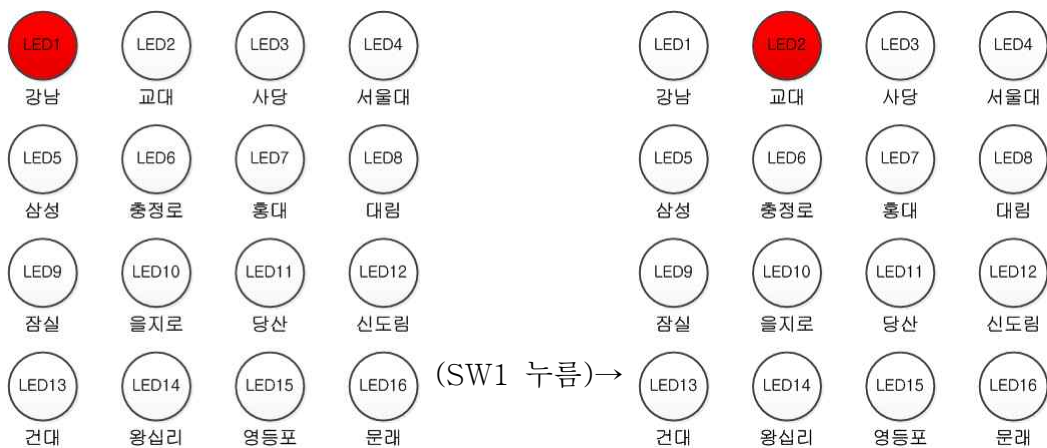
	S	t	a	r	t		S	t	a	t	i	o	n		
S	t	a	t	i	o	n		N	u	m		:		1	4

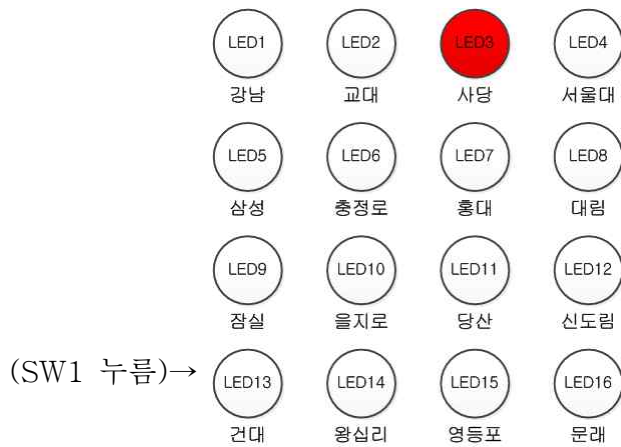
[문제 2] 스위치를 이용하여 LED를 정방향 또는 역방향으로 이동

나. LED

SW1를 1번 누를 때마다 1정거장씩 정방향으로 이동되고, SW2을 1번 누를 때마다 1정거장씩 역방향으로 이동되게 하시오.

(예) 강남(D2)역에서 SW1를 두 번 누르면 D4(사당역)이 선택되어 진다.





다. FND : 소등되게 하시오.

라. 출발역을 선택한 후(해당 역 LED가 점등상태) SW3을 누르면 도착역을 선택하게 하시오. (선택방법은 가, 나 와 같이 선택하시오.)

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

요구사항 5 : 요구사항 4의 상태에서 SW3을 누르면 다음과 같이 동작되게 하시오.

가. LCD

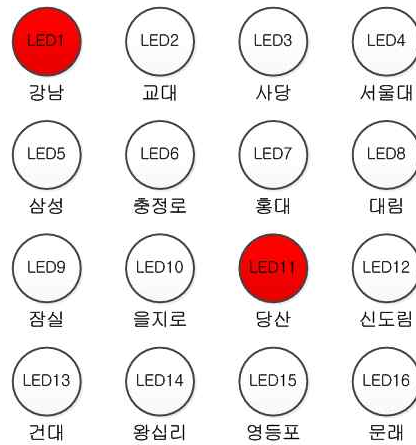
S	t	a	r	t	:	X	X		S	t	a	t	i	o	n
E	n	d			:	Y	Y		S	t	a	t	i	o	n

'XX' 출발역, 'YY' 도착역

(예) SW1 또는 SW2을 이용하여 출발지를 LED1(강남, 역번호 01) 도착지를 LED11 (당산, 역번호 09)을 선택하고 SW3을 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. LED



다. FND : 소등되게 하시오.

라. 출발역과 도착역이 같은 상태에서 SW3을 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

요구사항 6 : 요구사항 5에서 출발역과 도착역이 다른 역으로 설정되면 0.5초 후에 다음과 같이 동작되게 하시오.

[문제 3] 출발역과 도착역 선택 후 최단거리와 요금을 계산하여 LCD에 표시

가. LCD

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 $7+1=8$ 이 된다.

(2) 'YYYY' 는 지불해야 할 요금이다. 계산식은 다음과 같다.

Adult : $YYYY = 500 \times \text{거리}$

Kid : $YYYY = 300 \times \text{거리}$

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

- 거리는 8 이 된다.

- 어른 요금은 $500 \times 8 = 4,000\text{원}$

- 어린이 요금은 $300 \times 8 = 2,400\text{원}$

나. LED : 요구사항 [5]의 상태를 유지하게 하시오.

다. FND : 소등되게 하시오.

요구사항 [7] : 요구사항 [6]이 완료된 후 1초 후에 다음과 같이 동작되게 하시오.

가. LCD

(1) 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

[문제 4] 가변저항을 이용한 입력된 지불 요금 표시

① 'XXXX' 는 지불해야 할 요금이 표시되게 하시오.

② 'YYYY'는 VR1의 값에 따라 100원씩 증감하도록 하시오.

③ 'YYYY'값이 'XXXX'값보다 많거나 같으면 'ZZ'에 'OK'라고 표시되며, 그 전에는 공백이 되게 하시오.

(2) 'YYYY' \geq 'XXXX'이면 'ZZ'에 'OK'표시가 되고, SW3을 누르면 다음과 같이 표시되며 'QQQQ'에 거스름돈이 표시되게 하시오. (단, 거스름돈이 없을 경우 (3)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.

(3) (1) 또는 (2)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		

① 'X'의 값은 출발역부터 도착역까지의 거리를 나타내시오.

나. LED : 요구사항 ⑤의 상태를 유지하게 하시오.

다. FND : 소등되게 하시오.

요구사항 ⑧ : SW4를 누르면 다음과 같이 동작되게 하시오.

가. LCD

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

(1) 'X' 는 도착역까지 남은 시간이 표시되게 하시오. 단, 한 정거장당 1분으로 하며 동작상의 1초를 실제시간 1분이라고 가정한다.

(2) 'NNNN' 부분은 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	s

② 역방향일 경우

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

나. LED

출발지의 LED만 표시되고 도착지의 LED는 소등되며, 한 정거장당 1초 간격으로 도착지로 LED의 점멸하며 이동되게 하시오. (현재 역만 점등 함)

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

다. FND

FND1은 출발역 부터 도착역까지 이동한 역의 거리를 출력하고, FND2는 LCD에 표시되는 'X'값이 표시되게 하시오.

라. 도착지에 도착하면 2초 후에 FND와 LED를 소등하고 요구사항 2의 상태로 돌아가게 하시오.

(예) 출발지(LED1)는 강남역이고 도착지는 서울대(LED4)이면

LCD	LED	FND1	FND2
<div> Train Is Going . . - - - > : 3 minutes </div>	<div> <div>LED1 강남</div> <div>LED2 교대</div> <div>LED3 사당</div> <div>LED4 서울대</div> <div>LED5 삼성</div> <div>LED6 충정로</div> <div>LED7 홍대</div> <div>LED8 대림</div> <div>LED9 잠실</div> <div>LED10 을지로</div> <div>LED11 당산</div> <div>LED12 신도림</div> <div>LED13 건대</div> <div>LED14 왕십리</div> <div>LED15 영등포</div> <div>LED16 문래</div> </div>	<div>0</div>	<div>3</div>
↓ 1초 후			
<div> Train Is Going . . - - - > : 2 minutes </div>	<div> <div>LED1 강남</div> <div>LED2 교대</div> <div>LED3 사당</div> <div>LED4 서울대</div> <div>LED5 삼성</div> <div>LED6 충정로</div> <div>LED7 홍대</div> <div>LED8 대림</div> <div>LED9 잠실</div> <div>LED10 을지로</div> <div>LED11 당산</div> <div>LED12 신도림</div> <div>LED13 건대</div> <div>LED14 왕십리</div> <div>LED15 영등포</div> <div>LED16 문래</div> </div>	<div>1</div>	<div>2</div>
↓ 1초 후			
<div> Train Is Going . . - - - > : 1 minutes </div>	<div> <div>LED1 강남</div> <div>LED2 교대</div> <div>LED3 사당</div> <div>LED4 서울대</div> <div>LED5 삼성</div> <div>LED6 충정로</div> <div>LED7 홍대</div> <div>LED8 대림</div> <div>LED9 잠실</div> <div>LED10 을지로</div> <div>LED11 당산</div> <div>LED12 신도림</div> <div>LED13 건대</div> <div>LED14 왕십리</div> <div>LED15 영등포</div> <div>LED16 문래</div> </div>	<div>2</div>	<div>1</div>
↓ 1초 후			
<div> Train Is Arrived Thanks A Lot ! </div>	<div> <div>LED1 강남</div> <div>LED2 교대</div> <div>LED3 사당</div> <div>LED4 서울대</div> <div>LED5 삼성</div> <div>LED6 충정로</div> <div>LED7 홍대</div> <div>LED8 대림</div> <div>LED9 잠실</div> <div>LED10 을지로</div> <div>LED11 당산</div> <div>LED12 신도림</div> <div>LED13 건대</div> <div>LED14 왕십리</div> <div>LED15 영등포</div> <div>LED16 문래</div> </div>	<div>3</div>	<div>0</div>
↓ 2초 후			
FND와 LED를 점등하고 <u>요구사항 2</u> 의 상태로 돌아간다.			

[문제 5] 통신

요구사항 9 : 아래와 같이 통신 속도를 맞추고 출력 양식에 맞춰서 실시간으로 출력하시오.

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

요구사항 8항 실행시 동작하시오.

Time/m : 2
Distance : 2
Start Station : 01
End Station : 03
Now Station : 01
Money : 0600

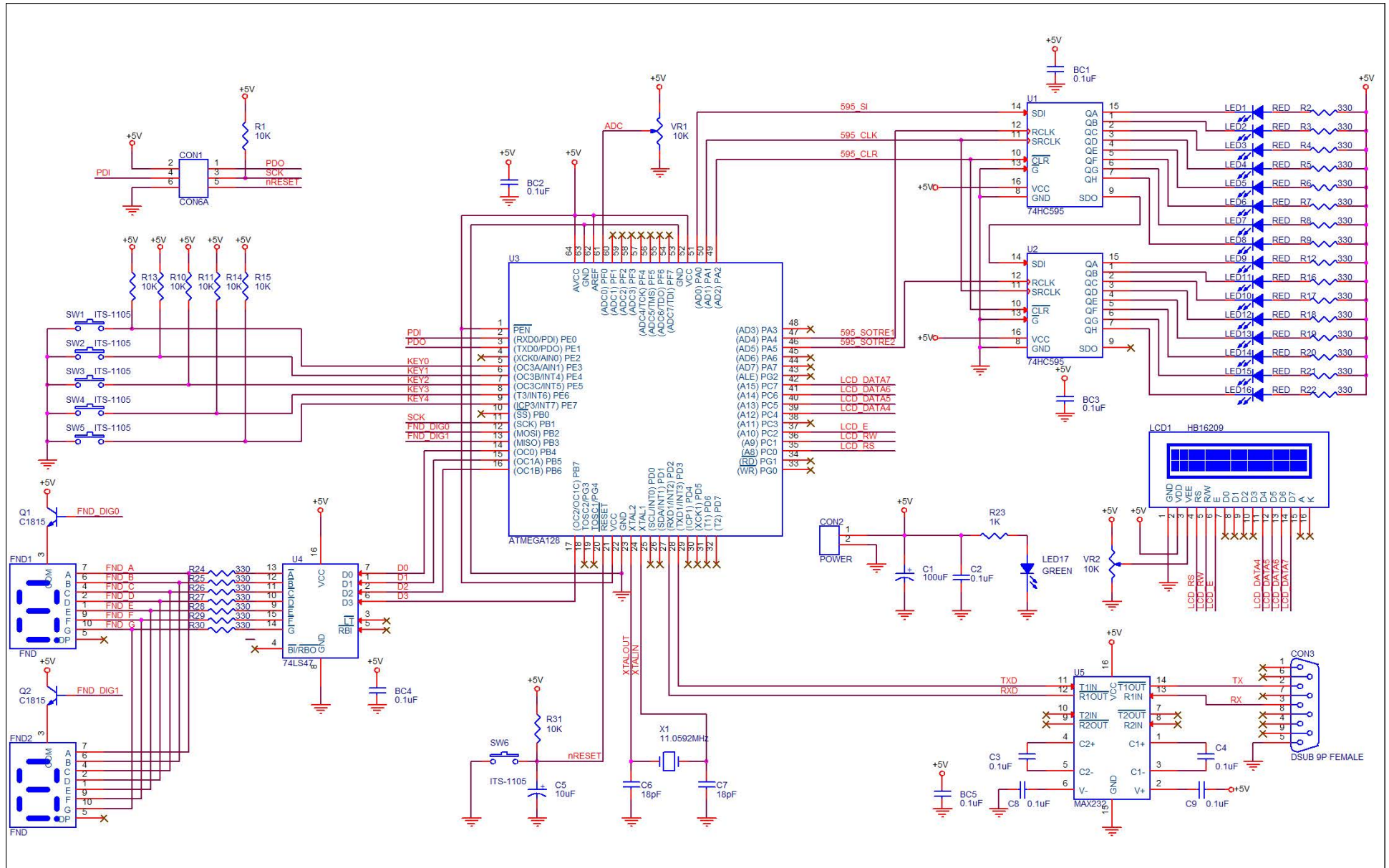
● 음영 처리된 부분은 실시간으로 변동되도록 한다.

출력양식

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

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

4. 회로도



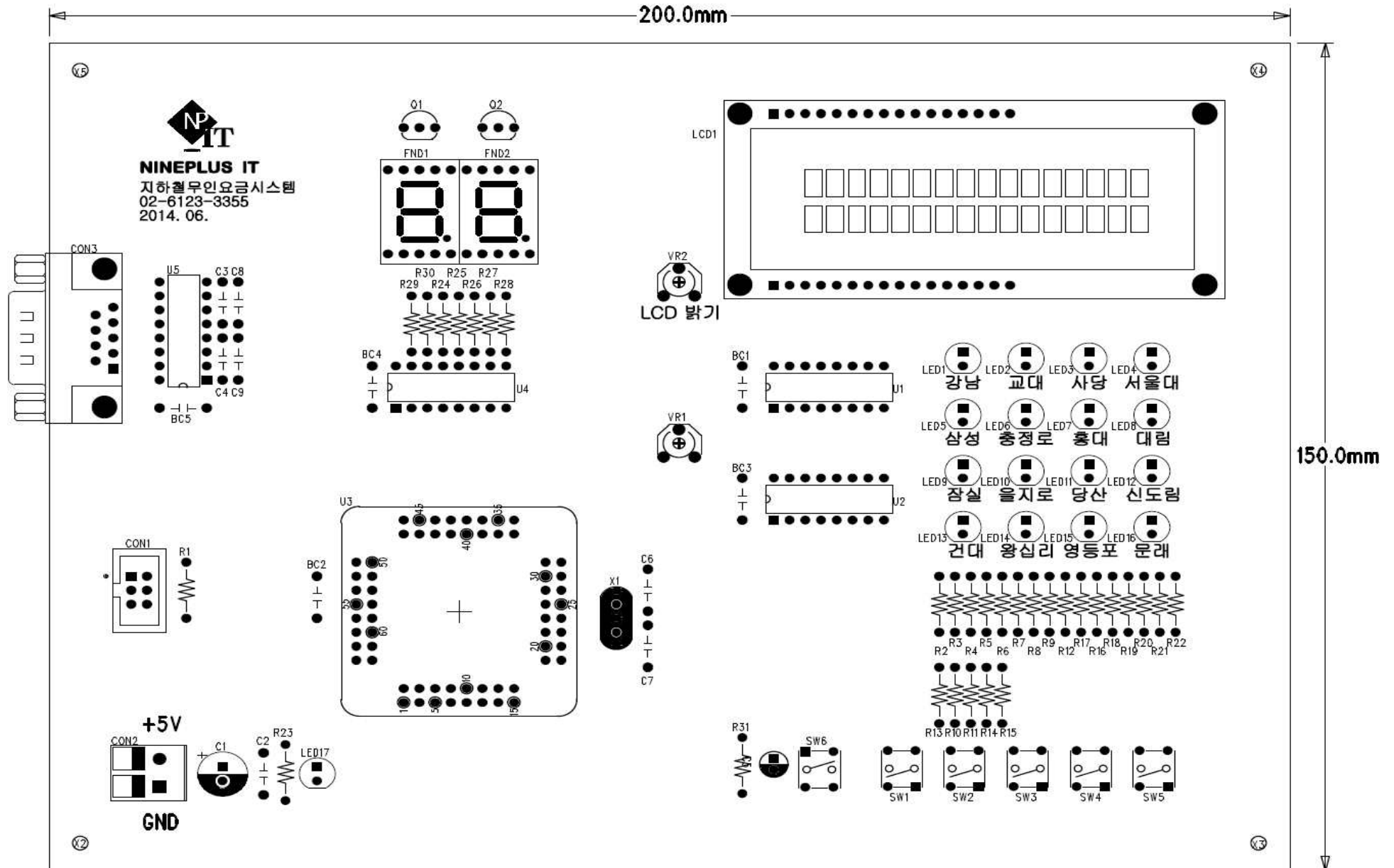
5-1. 지급 재료 목록(1/2)

번호	재 료 명	규 격 (치 수)	단위	소요량	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	IC	74HC595	개	2	
8	IC	74LS47	개	1	
9	IC	MAX232	개	1	
10	DSUB Connector	Right Angle(9pin, 암)	개	1	
11	Ceramic Capacitor	0.1uF/50V	개	10	
12	Ceramic Capacitor	18pF/50V	개	2	
13	Electrolytic Capacitor	10uF/16V	개	1	
14	Electrolytic Capacitor	100uF/16V	개	1	
15	Resistor	10KΩ, 1/4W, 5%	개	7	
16	Resistor	1KΩ, 1/4W, 5%	개	1	
17	Resistor	330Ω, 1/4W, 5%	개	23	
18	Variable Resistor	10KΩ	개	2	
19	FND	FND 507	개	2	
20	LED	Red(5φ)	개	16	
21	LED	Green(5φ)	개	1	
22	Buzzer	Buzzer(DC 5V)	개	1	
23	Crystal	11.0592MHz	개	1	
24	TR	2SC1815	개	2	
25	Tact Switch	Tact Switch	개	6	
26	Power Connector	녹색단자 2pin, 5.0mm	개	1	

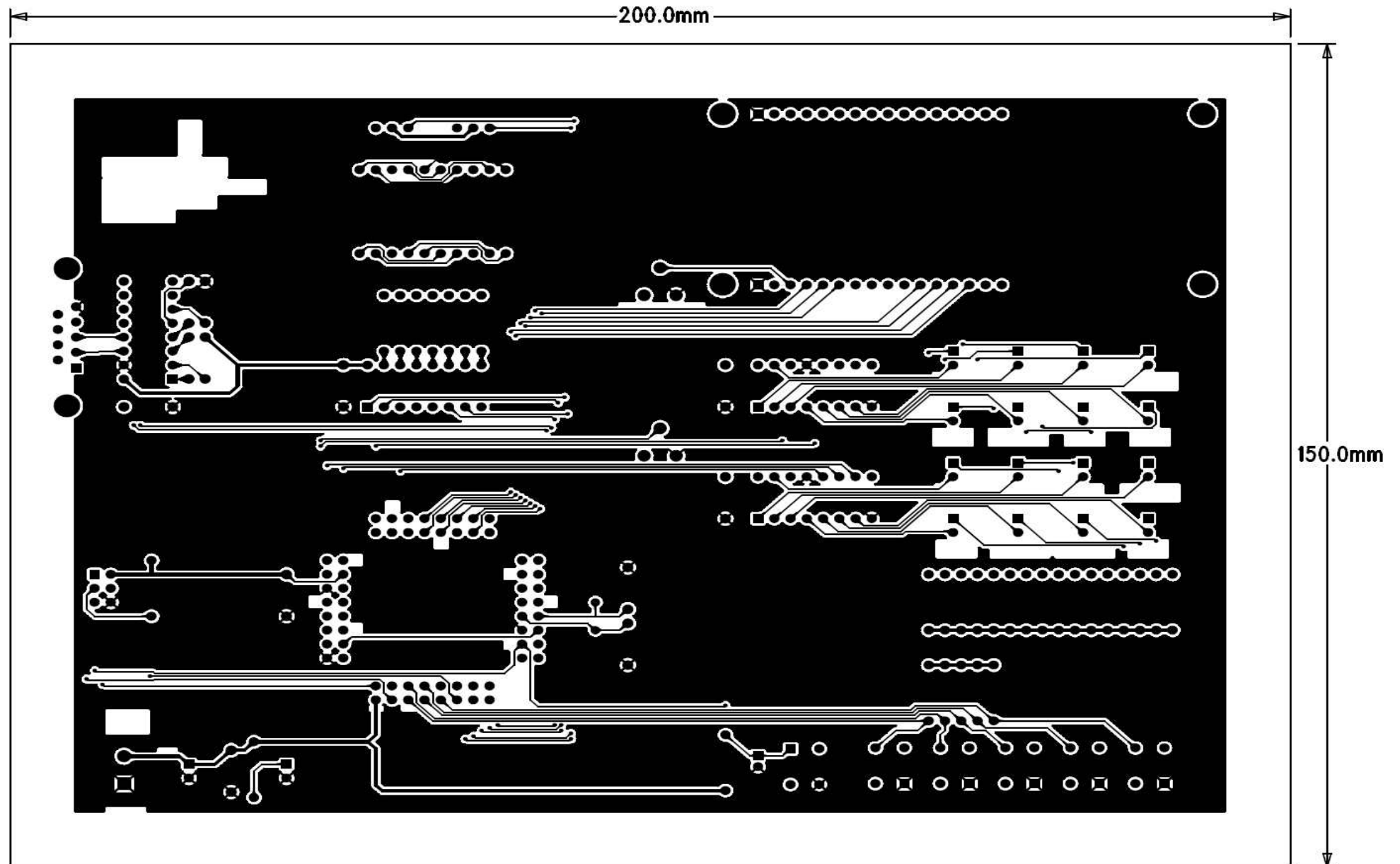
5-2. 지급 재료 목록(2/2)

[illegible]

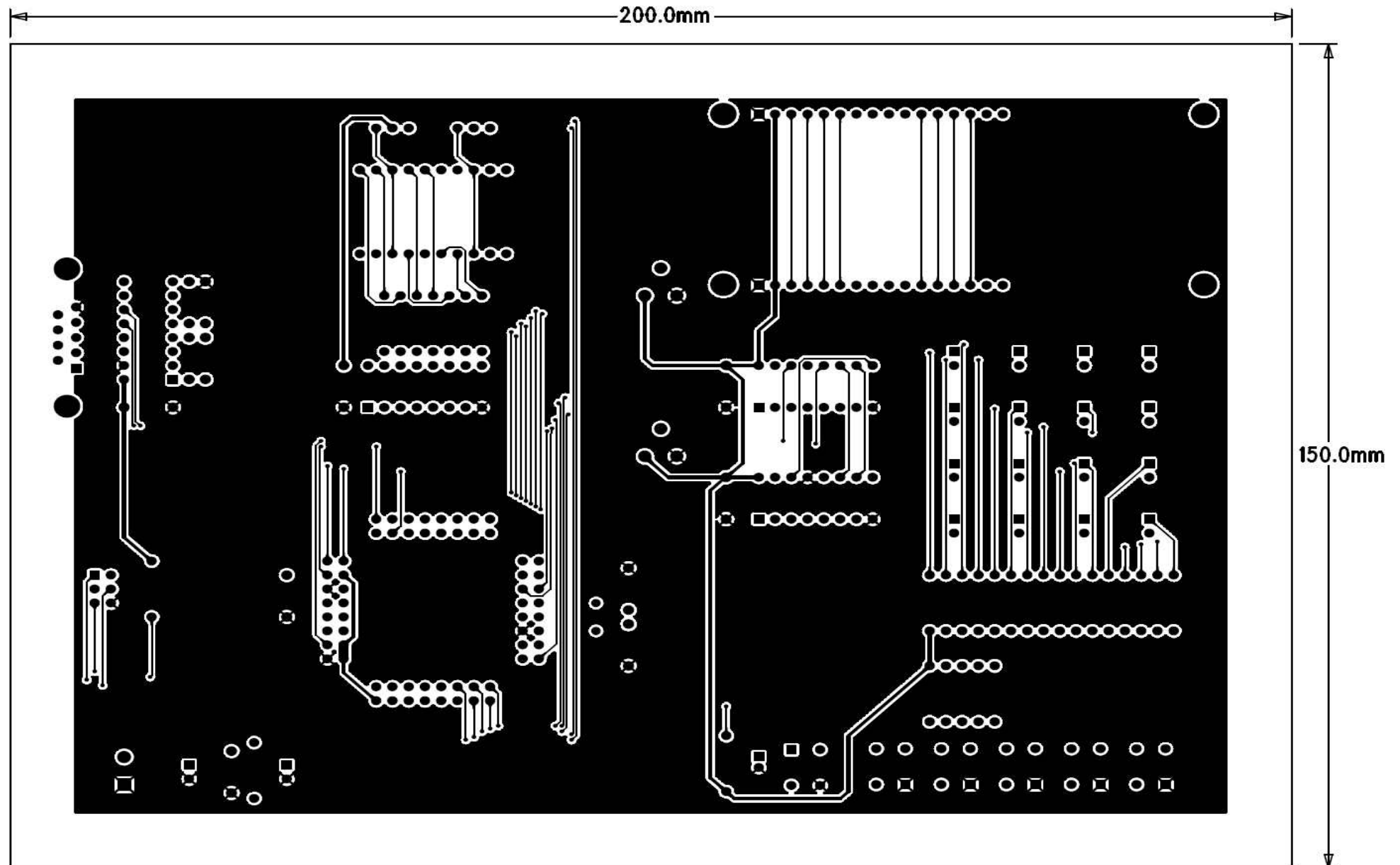
6-1. PCB(Top Silk-Screen)



6-2 PCB(TOP Routing)



6-3. PCB(BOTTOM Routing)



프로그램 정답 소스(AvrStudio)

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

#include <stdio.h>

#define ON          1
#define OFF         0

#define NONE        0
#define ADULT       1
#define KID          2

#define FORWARD     0
#define REWARD     1

// Key Define
#define KEY_PIN      PINE
#define KEY_DDR      DDRE

#define KEY1         1
#define KEY2         2
#define KEY3         3
#define KEY4         4
#define KEY5         5

// Lcd Define
#define LCD_PORT     PORTC
#define LCD_PIN      PINC
#define LCD_DDR      DDRC

#define LCD_RS       0x01
#define LCD_RW       0x02
#define LCD_E        0x04

#define DDRAM        0x80

// FND Define
#define FND_PORT     PORTB
#define FND_DDR      DDRB
#define FND_DIG0     0x04
#define FND_DIG1     0x08

// Matrix LED Define
#define MLED_PORT     PORTA
#define MLED_DDR      DDRA
#define MLED_DATA     0x01
#define MLED_SHIFT    0x02
#define MLED_CLEAR    0x04
#define MLED_LATCH1   0x10
#define MLED_LATCH2   0x20

// Serial Baudrate Define
#define B4800         143
#define B9600         71
#define B19200        35
#define B38400        17
#define B57600        11
#define B115200        5

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

```

// Global Variable Define
volatile unsigned char key_flag;
volatile char fnd_buf[2], fnd_flag;
volatile int dig;
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;
int fee_tbl[2] = { 300, 500 }, fee, money;

void timer0_init(void)
{
    TCCR0 = 0x07;
    TCNT0 = 256 - 11;
    TIMSK = 0x01;
}

SIGNAL(TIMER0_OVF_vect)
{
    TCNT0 = 256 - 11;

    if( fnd_flag ) {
        FND_PORT = 0x00;
        FND_PORT = (fnd_buf[dig] << 4) | (FND_DIG0 << dig);

        dig ^= 0x01;
    }
}

void fnd_off(void)
{
    fnd_flag = OFF;
    FND_PORT = 0x00;
}

void matrix_led(int data)
{
    int tmp = ~data;
    int i;

    MLED_PORT |= MLED_CLEAR;
    MLED_PORT &= ~(MLED_LATCH1 | MLED_LATCH2);
    for( i = 0; i < 16; i++ ) {
        if( tmp & 0x8000 ) MLED_PORT |= MLED_DATA;
        else MLED_PORT &= ~MLED_DATA;

        MLED_PORT |= MLED_SHIFT;
        tmp <<= 1;
        MLED_PORT &= ~MLED_SHIFT;
    }

    MLED_PORT |= (MLED_LATCH1 | MLED_LATCH2);
    MLED_PORT &= ~(MLED_LATCH1 | MLED_LATCH2);
}

void matrix_alloff(void)
{
    int i;

    MLED_PORT |= MLED_CLEAR;
    MLED_PORT &= ~(MLED_LATCH1 | MLED_LATCH2);
    for( i = 0; i < 16; i++ ) {
        MLED_PORT |= MLED_DATA;

        MLED_PORT |= MLED_SHIFT;
        MLED_PORT &= ~MLED_SHIFT;
    }
}

```

```

    MLED_PORT |= (MLED_LATCH1 | MLED_LATCH2);
    MLED_PORT &= ~(MLED_LATCH1 | MLED_LATCH2);
}

```

```

void matrix_allon(void)
{
    MLED_PORT &= ~MLED_CLEAR;
    MLED_PORT |= (MLED_LATCH1 | MLED_LATCH2);
    MLED_PORT &= ~(MLED_LATCH1 | MLED_LATCH2);
}

```

```

char getkey(char prev_key)
{
    char key = 0, old_key;

    key_flag = 0;

    old_key = ~(KEY_PIN >> 3) & 0x1F;
    if( old_key ) {
        _delay_ms(50);
        key = ~(KEY_PIN >> 3) & 0x1F;
        if( key == old_key ) {
            if( key == 0x01 )    key = KEY1;
            else if( key == 0x02 ) key = KEY2;
            else if( key == 0x04 ) key = KEY3;
            else if( key == 0x08 ) key = KEY4;
            else if( key == 0x10 ) key = KEY5;

            if( key != prev_key )    key_flag = 1;
        }
    }

    return key;
}

```

```

void lcd_bussycheck(void)
{
    LCD_DDR = 0x0F;

    LCD_PORT = 0x00;
    LCD_PORT |= LCD_RW;
    LCD_PORT |= LCD_E;
    while( LCD_PIN & 0x80 );
    LCD_PORT &= ~LCD_E;

    LCD_DDR = 0xFF;
}

```

```

unsigned char lcd_command_read(void)
{
    char cmd;

    LCD_DDR = 0x0F;
    LCD_PORT = 0x00;

    LCD_PORT |= LCD_RW;

    LCD_PORT |= LCD_E;
    asm("nop");
    cmd = LCD_PIN & 0xF0;
    LCD_PORT &= ~LCD_E;

    LCD_PORT |= LCD_E;
    asm("nop");
    cmd |= LCD_PIN >> 4;
    LCD_PORT &= ~LCD_E;

    return cmd;
}

```

```

void lcd_command_write(unsigned char cmd)
{
    lcd_bussycheck();
}

```

```

    LCD_PORT = 0x00;

    LCD_PORT |= cmd & 0xF0;
    LCD_PORT |= LCD_E;
    LCD_PORT &= ~LCD_E;

    LCD_PORT &= 0x0F;
    LCD_PORT |= cmd << 4;
    LCD_PORT |= LCD_E;
    LCD_PORT &= ~LCD_E;

    _delay_ms(2);
}

unsigned char lcd_data_read(void)
{
    char data;

    LCD_DDR = 0x0F;
    LCD_PORT = 0x00;

    LCD_PORT |= LCD_RS;
    LCD_PORT |= LCD_RW;

    LCD_PORT |= LCD_E;
    asm("nop");
    data = LCD_PIN & 0xF0;
    LCD_PORT &= ~LCD_E;

    LCD_PORT |= LCD_E;
    asm("nop");
    data |= LCD_PIN >> 4;
    LCD_PORT &= ~LCD_E;

    return data;
}

void lcd_data_write(unsigned char data)
{
    lcd_bussycheck();

    LCD_PORT = 0x00;

    LCD_PORT |= LCD_RS;

    LCD_PORT |= data & 0xF0;
    LCD_PORT |= LCD_E;
    LCD_PORT &= ~LCD_E;

    LCD_PORT &= 0x0F;
    LCD_PORT |= data << 4;
    LCD_PORT |= LCD_E;
    LCD_PORT &= ~LCD_E;

    _delay_us(50);
}

void lcd_string(char *str)
{
    while( *str ) lcd_data_write(*str++);
}

void lcd_gotoxy(char x, char y)
{
    lcd_command_write(DDRAM | (0x40 * y) | x);
}

void lcd_init(void)
{
    LCD_DDR = 0xFF;

    lcd_command_write(0x20);
}

```

```

    _delay_ms(10);
    lcd_command_write(0x20);
    _delay_ms(10);
    lcd_command_write(0x20);
    _delay_ms(10);
    lcd_command_write(0x28);
    lcd_command_write(0x08);
    lcd_command_write(0x01);
    lcd_command_write(0x06);
    _delay_ms(10);
    lcd_command_write(0x0C);
}

void uart_transmit(unsigned char data)
{
    while( !( UCSR1A & (1 << UDRE1)) );

    UDR1 = data;
}

void uart_string(char *str)
{
    while( *str ) uart_transmit(*str++);
}

unsigned char uart_receive(void)
{
    while ( !(UCSR1A & (1 << RXC)) );

    return UDR1;
}

void uart_init(unsigned char baudrate)
{
    UCSR1A = 0b00000000;
    UCSR1B = 0b10011000;
    UCSR1C = 0b00000110;
    UBRR1H = 0x00;
    UBRR1L = baudrate;
}

void adc_init(void)
{
    ADCSRA = 0b10001111;
    ADMUX = 0b00000000;
}

void adc_start(void)
{
    ADCSRA |= 0x40;
}

SIGNAL(ADC_vect)
{
    money = (int)(((1023 - ADC) / 20.46) * 100);
    ADCSRA |= 0x40;
}

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++ ) {
        matrix_led(led_order[i]);
        _delay_ms(300);
    }

    main_screen();
}

void mcu_init(void)
{
    KEY_DDR = 0x00;
    FND_DDR = 0xFC;
    FND_PORT = 0xF0;
    MLED_DDR = 0xFF;
    MLED_PORT = 0x00;

    adc_init();
    lcd_init();
    uart_init(B9600);
    timer0_init();

    sei();
}

void variable_init(void)
{
    key_flag = OFF;

    dig = 0;
    fnd_off();

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

    distance = 0;
    fee = 0;
}

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 KEY1 :
            station++;
            if( station == 17 )station = 1;
            break;

        case KEY2 :
            station--;
            if( station == 0 ) station = 16;
            break;

        case KEY3 :
            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 ) {
                    distance *= (-1);
                    if( distance < 8 ) {
                        direction = REWARD;
                    }
                }
                else {
                    direction = FORWARD;
                    distance = 16 - distance;
                }
            }
        }
}

```

```

        else {
            if( distance > 8 ) {
                direction = REWARD;
                distance = 16 - distance;
            }
            else {
                direction = FORWARD;
            }
        }

        lcd_gotoxy(0, 0);
        printf("Distance:%d      ", distance);

        fee = fee_tbl[ticket_type % 2] * distance;
        lcd_gotoxy(0, 1);
        printf("Fee : %04d Won   ", fee);

        _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 KEY5 :
    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
/////////////////////////////////////////////////////////////////
    uart_transmit('Wf');
    sprintf(str, "Time/m          : %dWrWn", distance - i);
    uart_string(str);
    sprintf(str, "Distance       : %dWrWn", i);
    uart_string(str);
    sprintf(str, "Start Station : %02dWrWn", start_station);
    uart_string(str);
    sprintf(str, "End Station  : %02dWrWn", end_station);
    uart_string(str);
    sprintf(str, "Now Station  : %02dWrWn", pos + 1);
    uart_string(str);
    sprintf(str, "Money        : %04dWrWn", fee);
    uart_string(str);

    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)
{
    char key = 0xFF;

    mcu_init();
    variable_init();
    matrix_alloff();

    adc_start();

    fdevopen((void *)lcd_data_write, 0);

    start_screen();

    state = SELECT_TICKET_TYPE;

    while( 1 ) {
        key = getkey(key);
        if( key_flag ) {
            switch( key ) {
                case KEY1 :
                case KEY2 :
                    if( state == SELECT_TICKET_TYPE ) {
                        state = SELECT_START_STATION;

                        lcd_command_write(0x01);
                        lcd_gotoxy(0, 0);
                        if( key == KEY1 )    lcd_string("You Select Adult");
                        else if( key == KEY2 ) lcd_string("You Select Kid ");
                    }
                }
            }
        }
    }
}

```

```

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

case KEY3 :
    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 KEY4 :
    if( state == TRAIN_GO ) train_go();
    break;

case KEY5 :
    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 ) {
    lcd_gotoxy(9, 1);
    printf("%04d   ", money);

    if( money >= fee ) {
        lcd_gotoxy(14, 1);
        lcd_string("OK");
    }
}

}

return 0;
}

```