**2.3.9 디지털 입출력 (51p)**

/\*

digital pin test

\*/

int led1=5; //led를 digital 핀 5번에 연결

int led2=6; //led를 digital 핀 6번에 연결

int led3=7; //led를 digital 핀 7번에 연결

void setup()

{

  pinMode(led1,OUTPUT);

  pinMode(led2,OUTPUT); // digital 핀 6번을 출력으로 설정

  pinMode(led3,OUTPUT); // digital 핀 7번을 출력으로 설정

}

void loop()

{

  digitalWrite(led1,HIGH); // led1 on

  digitalWrite(led2,HIGH); // led2 on

  digitalWrite(led3,HIGH); // led3 on

  delay(1000);

  digitalWrite(led1,LOW); // led1 off

  digitalWrite(led2,LOW); // led2 off

  digitalWrite(led3,LOW); // led3 off

  delay(1000);

}

2.3.10 아날로그 입력 & PWM 출력 (53p)

/\*

analog pin test

led=Digital pin D9

potentiometer A0

\*/

int led=9; //led를 핀 9번에 연결

void setup()

{

  pinMode(led,OUTPUT); //digital 핀 9번을 출력으로 작성

}

void loop()

{

  int input = analogRead(A0); //int 변수 input을 선언하고 값을 A0에서 읽은 값으로 초기화

  analogWrite(led,input/4); //가변저항에서 읽어들인 값을 4로 나눈 후 9번판으로 출력

}

77p) 최초 구동시에는 LED가 꺼져있다가 시리얼 모니터로 데이터가 입력되면 시리얼 모니터에 “LED on”이라는 문자열을 출력하고 LED가 켜지게 해라.

/\*

digital pin test

\*/

void setup() { // 초기 설정 함수

  Serial.begin(9600); // 시리얼 속도 9600으로 설정

  pinMode(3, OUTPUT); //생략 가능

  digitalWrite(3,HIGH); // led를 꺼진 상태로 초기화

}

void loop() {

  if(Serial.available()) // 시리얼 모니터를 통해 입력이 있다면,

  {

    Serial.println("LED on"); // led를 켰다는 문자열 출력

    digitalWrite(3,LOW); // led를 켭니다.

  }

}

3.2.1 일반 LED (80p)

int led1 = 6;

int led2 = 7;

void setup() {

  pinMode(led1,OUTPUT); // 6번에 연결된 led1을 출력모드로 설정

  pinMode(led2,OUTPUT); // 7번에 연결된 led2을 출력모드로 설정

  Serial.begin(9600); // 시리얼 통신 속도 9600으로 설정

  Serial.println("시리얼 통신 시작"); //시리얼 통신 시작 문자 출력

}

void loop()

{

  if(Serial.available()) //시리얼 모니터를 통해 입력이 있다면

  {

    char a; //변수 a 선언

    a=Serial.read(); //전송된 문자를 1바이트씩 읽음

    if(a=='1') //전송 받은 문자가 1이라면

    {

      digitalWrite(6,HIGH); //led1 on

      digitalWrite(7,HIGH); //led2 on

      Serial.println("LED ON"); //문자열 출력

    }

    if(a=='0') //전송 받은 문자가 0이라면

    {

      digitalWrite(6,LOW); //led1 off

      digitalWrite(7,LOW); //led2 off

      Serial.println("LED OFF"); // 문자열 출력

    }

  }

}

**3.2.2 RGB LED 모듈**

int LED1 =9;

int LED2 =10;

int LED3 =11;

void setup() {

  pinMode(LED1,OUTPUT);

  pinMode(LED2,OUTPUT);

  pinMode(LED3,OUTPUT);

}

void loop() {

  digitalWrite(LED1,HIGH); //빨간색 LED 1초 동안 내보내기

  delay(1000);

  digitalWrite(LED1,LOW); //빨간색 LED off

  delay(100);

  digitalWrite(LED2,HIGH); //초록색 LED 1초 동안 내보내기

  delay(1000);

  digitalWrite(LED2,LOW); //초록색 LED off

  delay(100);

  digitalWrite(LED3,HIGH); //파란색 LED 1초 동안 내보내기

  delay(1000);

  digitalWrite(LED3,LOW); //파란색 LED off

  delay(100);

  digitalWrite(LED1,HIGH); // 빨간색 LED 1초동안 내보내기

  digitalWrite(LED2,HIGH); // 초록색 LED 1초동안 내보내기

  delay(1000);

  digitalWrite(LED1,LOW); // 빨간색 LED off

  digitalWrite(LED2,LOW); // 초록색 LED off

  delay(100);

  digitalWrite(LED2,HIGH); //초록색 LED 1초 동안 내보내기

  digitalWrite(LED3,HIGH); //파란색 LED 1초 동안 내보내기

  delay(1000);

  digitalWrite(LED2,LOW); // 초록색 LED off

  digitalWrite(LED3,LOW); // 파란색 LED off

  delay(100);

  digitalWrite(LED1,HIGH); // 빨간색 LED 1초동안 내보내기

  digitalWrite(LED2,HIGH); // 초록색 LED 1초동안 내보내기

  digitalWrite(LED3,HIGH); //파란색 LED 1초 동안 내보내기

  delay(1000);

  digitalWrite(LED1,LOW); // 빨간색 LED off

  digitalWrite(LED2,LOW); // 초록색 LED off

  digitalWrite(LED3,LOW); // 파란색 LED off

  delay(100);

}

**3.2.3 FND**

int seg[] = {2,3,4,5,6,7,8,9};//배열로 D2~D9까지 세그먼트 연결

byte digits[10][8] = //0~9까지 숫자에 대한 LED 값

{

  {1,1,0,0,0,0,0,0 },//0

  {1,1,1,1,1,0,0,1 },//1

  {1,0,1,0,0,1,0,0 },//2

  {1,0,1,1,0,0,0,0 },//3

  {1,0,0,1,1,0,0,1 },//4

  {1,0,0,1,0,0,1,0 },//5

  {1,0,0,0,0,0,1,0 },//6

  {1,1,1,1,1,0,0,0 },//7

  {1,0,0,0,0,0,0,0 },//8

  {1,0,0,1,0,0,0,0 },//9

};

void setup() {

  for(int i=0;i<8;i++) //for 문을 사용해 2~9까지 8개의 핀모드 출력으로 세팅

  {

    pinMode(seg[i], OUTPUT);

  }

}

void loop() {

  for(int i=0;i<10;i++) //이중 for 문을 이용해 led로 숫자 출력

  {

    for(int j=0;j<8;j++)

    {

      digitalWrite(seg[j],digits[i][j]);

    }

    delay(1000); //1초간 딜레이

  }

}

**3.2.4 도트 메트릭스**

int rows[] = {-1,10,11,12,13,14,15,16,17};

int cols[] = {-2,2,3,4,5,6,7,8,9};

void setup()

{

  for(int i = 1; i<9; i++) //도트 메트릭스 행렬 출력으로 세팅

  {

    pinMode(cols[i],OUTPUT);

    pinMode(rows[i],OUTPUT);

  }

}

void loop() {

  //1행엔 켜지는 led가 없으므로 생략

  clear(); //사용자 정의함수 clear 호출, led 초기화

  digitalWrite(rows[2],HIGH); //2행 led on

  digitalWrite(cols[2],LOW);

  digitalWrite(cols[3],LOW);

  digitalWrite(cols[6],LOW);

  digitalWrite(cols[7],LOW);

  delay(2);

  clear();

  digitalWrite(rows[3],HIGH); //3행 led on

  digitalWrite(cols[1],LOW);

  digitalWrite(cols[4],LOW);

  digitalWrite(cols[5],LOW);

  digitalWrite(cols[8],LOW);

  delay(2);

  clear();

  digitalWrite(rows[4],HIGH); //4행 led on

  digitalWrite(cols[1],LOW);

  digitalWrite(cols[8],LOW);

  delay(2);

  clear();

  digitalWrite(rows[5],HIGH); //5행 led on

  digitalWrite(cols[1],LOW);

  digitalWrite(cols[8],LOW);

  delay(2);

  clear();

  digitalWrite(rows[6],HIGH); //6행 led on

  digitalWrite(cols[2],LOW);

  digitalWrite(cols[7],LOW);

  delay(2);

  clear();

  digitalWrite(rows[7],HIGH); //7행 led on

  digitalWrite(cols[3],LOW);

  digitalWrite(cols[6],LOW);

  delay(2);

  clear();

  digitalWrite(rows[8],HIGH); //8행 led on

  digitalWrite(cols[4],LOW);

  digitalWrite(cols[5],LOW);

  delay(2);

}

void clear() //사용자 정의함수 clear 선언

{

  for(int i=1;i<9;i++)//LED 초기화 2중 for문

  {

    digitalWrite(rows[i],LOW);

  }

}

**Quest 8. 93p**

/\* 시리얼 모니터에 1을 입력하면 LED가 한 개 켜지고, FND에 숫자 1을 표시.

시리얼 모니터에 2를 입력하면 LED가 두 개 켜지고, FND에 숫자 2를 표시.

시리얼 모니터에 3을 입력하면 LED가 세 개 켜지고, FND에 숫자 3을 표시.

시리얼 모니터에 0을 입력하면 LED가 모두 꺼지고, FND에 숫자 0을 표시. \*/

int led1 = 12;//led1을 D12에 연결

int led2 = 11;

int led3 = 10;

int seg[] = {2,3,4,5,6,7,8,9};//배열로 D2에서 D9까지 세그먼트 연결

byte digits[4][8] = //1~3까지 숫자에 대한 LED 값

{

  {1,1,0,0,0,0,0,0 },//0

  {1,1,1,1,1,0,0,1 },//1

  {1,0,1,0,0,1,0,0 },//2

  {1,0,1,1,0,0,0,0 },//3

};

void setup()

{

  for(int i = 0; i < 8; i++) //for 문을 사용해 핀 2~9까지 8개의 핀모드 출력으로 세팅

  {

    pinMode(seg[i],OUTPUT);

  }

//초기 세팅

for (int i = 0; i<8; i++) //for 문으로 숫자 0 출력

  {

    digitalWrite(seg[i],digits[0][i]);

  }

pinMode(led1,OUTPUT); //led1 출력세팅

pinMode(led2,OUTPUT);

pinMode(led3,OUTPUT);

digitalWrite(led1,HIGH); //led1 off로 초기화

digitalWrite(led2,HIGH);

digitalWrite(led3,HIGH);

Serial.begin(9600);

Serial.println("시리얼 통신 시작"); //시리얼 통신 시작 문자 출력

}

void loop()//반목 함수 프로그램의 주 내용

{

  char a; //변수 a 선언

  if (Serial.available())//시리얼 모니터를 통해 입력이 있다면

  {

    a=Serial.read(); //전송된 문자를 1바이트씩 읽음

    if(a=='0')//전송받은 문자가 0이라면

    {

     digitalWrite(led1, HIGH);//led1 off

     digitalWrite(led2, HIGH);//led2 off

     digitalWrite(led3, HIGH);//led3 off

     for (int i = 0;i<8; i++) //for 문으로 숫자 0 출력

     {

       digitalWrite(seg[i],digits[0][i]);

     }

    }

    if(a=='1') //전송받은 문자가 1이라면

    {

     digitalWrite(led1, LOW);//led1 on

     digitalWrite(led2, HIGH);//led2 off

     digitalWrite(led3, HIGH);//led3 off

     for(int i = 0; i < 8; i++) //for 문으로 숫자 1 출력

     {

       digitalWrite(seg[i],digits[1][i]);

     }

    }

    if(a=='2') //전송받은 문자가 2라면

    {

     digitalWrite(led1, LOW);//led1 on

     digitalWrite(led2, LOW);//led2 on

     digitalWrite(led3, HIGH);//led3 off

     for(int i = 0; i < 8; i++) // for 문으로 숫자 2 출력

     {

       digitalWrite(seg[i],digits[2][i]);

     }

    }

    if(a=='3') //전송받은 문자가 3이라면

    {

     digitalWrite(led1, LOW);//led1 on

     digitalWrite(led2, LOW);//led2 on

     digitalWrite(led3, LOW);//led3 on

     for(int i = 0; i < 8; i++) // for 문으로 숫자 3출력

     {

       digitalWrite(seg[i],digits[3][i]);

     }

    }

  }

}

**3.3.1 택트 스위치 99p**

int sw = 8; //스위치를 digital 핀 8번에 연결

int led = 7; //led를 digital 핀 7 번에 연결

void setup() {

  pinMode(sw, INPUT); //digital 핀 8 번을 입력으로 설정

  pinMode(led, OUTPUT); //digital 핀 7 번을 출력으로 설정

}

void loop() {

  if (digitalRead(sw) ==0) //8번핀의 입력 값을 읽어서 입력값이 0이면

  {

    digitalWrite(led,LOW); //led off

  }

  else //8번핀의 입력 값을 읽어서 입력값이 1이면

  {

    digitalWrite(led, HIGH); //led on

  }

  delay(100); //시간 지연 0.1초

}

**3.3.2 볼 스위치**

int led = 6; //D6에 led 연결

int ball = 7; //D7에 볼 스위치 연결

void setup()

{

  pinMode(led,OUTPUT); //led 출력 모드로 설정

  pinMode(ball,INPUT\_PULLUP); // ball switch는 입력 모드 설정, 내부 풀업저항 사용

}

void loop()

{

  if (digitalRead(ball)) // 평소에는 led off

  {

    digitalWrite(led,HIGH);

  }

  else // ball switch가 기울어지면 led on

  {

    digitalWrite(led,LOW);

  }

}

**3.3.3 4X4 키 패드 103p**

#include <Keypad.h> // keypad 라이브러리 포함

const byte ROWS = 4; // 행의 개수

const byte COLS = 4; // 열의 개수

char myKeys[ROWS][COLS] = //배열로 각 버튼을 숫자나 문자로 대응

{

  {'0','1','2','3'},

  {'4','5','6','7'},

  {'8','9','a','b'},

  {'c','d','e','f'}

};

byte rowPins[ROWS] = {5,4,3,2}; // 행이 연결된 아두이노 핀 번호

byte colPins[COLS] = {8,9,10,11}; // 열이 연결된 아두이노 핀 번호

//keypad 객체 인스턴스화

Keypad customKeypad = Keypad(makeKeymap(myKeys), rowPins, colPins, ROWS, COLS);

void setup()

{

  Serial.begin(9600); //통신속도 9600으로 시리얼 통신 시작

}

void loop()

{

  //getKey()는 키패드에서 입력된 값이 있을 경우 반환합니다.

  char customKey = customKeypad.getKey();

  if(customKey) // 입력된 값이 있으면 출력

  {

    Serial.println(customKey);

  }

}

**Quest 9. 105p**

**스위치 두 개와 RGB LED 모듈로 구성된 회로를 구성함. 스위치 1을 누르면 누를 때마다 RGB LED가 ‘빨 노 초 하 파 자’ 순서로 변하고 스위치 2를 누르면 누를 때마다 RGB LED 가 ‘자 파 하 초 노 빨’ 순서로 변함.**

int LED1 = 9; // Red를 9번에 연결

int LED2 = 10; // Green을 핀 10번에 연결

int LED3 = 11; // BLUE를 vls 11qjsdp dusruf

int sw1 = 3; //스위치1을 디지털 핀 3번에 연결

int sw2 = 2; //스위치2을 디지털 핀 2번에 연결

int key\_count = 0 ; // 스위치 눌린 횟수 초기화

void setup()

{

  pinMode(LED1,OUTPUT); // LED 출력 설정

  pinMode(LED2,OUTPUT); // LED 출력 설정

  pinMode(LED3,OUTPUT); // LED 출력 설정

  pinMode(sw1,INPUT); // 스위치 입력 설정

  pinMode(sw2,INPUT); // 스위치 입력 설정

  digitalWrite(LED1,HIGH); // 빨강 LED 켜짐으로 초기설정

  digitalWrite(LED2,LOW); // 초록 LED 꺼짐으로 초기설정

  digitalWrite(LED3,LOW); // 파랑 LED 꺼짐으로 초기설정

}

void loop()

{

  if(digitalRead(sw1) == 1) // 스위치1의 입력 값을 읽어서 입력값이 1이면

  {

    key\_count++; //키 카운드 1 증가

  }

  if(digitalRead(sw2) == 1) // 스위치2의 입력 값을 읽어서 입력값이 1이면

  {

    key\_count--; //키 카운드 1 감소

  }

  if(key\_count<0)key\_count = 5; // 스위치 눌린 횟수가 0보다 작을 때 5로

  if(key\_count>5)key\_count = 0; // 스위치 눌린 횟수가 5보다 클 때 0으로

  rainbow(key\_count); // 해당 키 카운트에 해당하는 led를 켜는 사용자 함수

  delay(200); //짧은 시간 지연

}

void rainbow(int kc)

{

  if(kc ==0) // 빨강

  {

    digitalWrite(LED1,HIGH);

    digitalWrite(LED2,LOW);

    digitalWrite(LED3,LOW);

  }

  else if(kc == 1) //노랑

  {

    digitalWrite(LED1,HIGH);

    digitalWrite(LED2,HIGH);

    digitalWrite(LED3,LOW);

  }

  else if(kc == 2) // 초록

  {

    digitalWrite(LED1,LOW);

    digitalWrite(LED2,HIGH);

    digitalWrite(LED3,LOW);

  }

  else if(kc == 3) // 하늘

  {

    digitalWrite(LED1,LOW);

    digitalWrite(LED2,HIGH);

    digitalWrite(LED3,HIGH);

  }

  else if(kc == 4) // 파랑

  {

    digitalWrite(LED1,LOW);

    digitalWrite(LED2,LOW);

    digitalWrite(LED3,HIGH);

  }

  else if(kc == 5) // 자주

  {

    digitalWrite(LED1,HIGH);

    digitalWrite(LED2,LOW);

    digitalWrite(LED3,HIGH);

  }

}

**3.4.1 가변저항 110p**

/\*

potentiometer\_value

가변저항으로 값을 시리얼 모니터로 출력하는 예제

\*/

#define max\_R 10000 // 가변저항의 총 저항값 정의

void setup()

{

  Serial.begin(9600); // 9600bps로 시리얼 통신 시작

  Serial.println("Serial Com start"); // 시리얼 통신 시작 문자 출력

}

void loop()

{

  int val = analogRead(A0); // 아날로그 A0에서 값을 읽어서 변수 var에 저장. 전압에 따라 0~1023의 범위

  int resistor = map(val,0,1023,0,max\_R); // map 함수를 사용해서 아날로그값을 저항값으로 변환

  Serial.print("Resistor value");

  Serial.print(resistor); //저항값 출력

  Serial.println("ohm");

  delay(200);

}

**3.4.2 조이스틱 113p**

/\*

joystick 값을 시리얼 모니터로 출력하는 예제

\*/

int sw = 7; //조이스틱 푸쉬 스위치를 D7번에 연결

void setup()

{

  Serial.begin(9600); //통신속도 9600bps로 시리얼 통신 시작

  pinMode(sw, INPUT\_PULLUP); //스위치를 입력으로 세팅하고 내부 풀업저항을 사용함

  delay(1000); //1초 후에 시작

}

void loop()

{

  Serial.print("X축:");

  Serial.print(analogRead(A0)); //X축 값 출력

  Serial.print("\t");

  Serial.print("Y축:");

  Serial.print(analogRead(A1)); //Y축 값 출력

  Serial.print("\t");

  Serial.print("스위치:");

  Serial.print(analogRead(sw)); //스위치 값 출력

  delay(100);

}

**Quest 10. 115p**

**LED 4개를 마름모 모양으로 배치한다. 조이스틱을 기울이면 기울인 방향에 해당하는 LED 가 들어오도록 회로를 구성해서 구동해라.**

int LED\_R = 2; // 오른쪽 LED를 D2에 연결

int LED\_D = 3; // 아래쪽 LED를 D3에 연결

int LED\_U = 4; // 위쪽 LED를 D4에 연결

int LED\_L = 5; // 왼쪽 LED를 D5에 연결

void setup()

{

  pinMode(LED\_R, OUTPUT); //오른쪽 LED 출력 설정

  pinMode(LED\_D, OUTPUT); //아래쪽 LED 출력 설정

  pinMode(LED\_U, OUTPUT); //위쪽 LED 출력 설정

  pinMode(LED\_L, OUTPUT); //왼쪽 LED 출력 설정

  digitalWrite(LED\_R, HIGH); //오른쪽 LED 꺼짐으로 초기설정

  digitalWrite(LED\_D, HIGH); //아래쪽 LED 꺼짐으로 초기설정

  digitalWrite(LED\_U, HIGH); //위쪽 LED 꺼짐으로 초기설정

  digitalWrite(LED\_L, HIGH); //왼쪽 LED 꺼짐으로 초기설정

}

void loop()

{

  int val\_ud = analogRead(A0); // 조이스틱 상하 값 변수에 저장

  int val\_rl = analogRead(A1); // 조이스틱 좌우 값 변수에 저장

  if (val\_ud < 256)

  {

    digitalWrite(LED\_U, LOW); // 위쪽 LED 켜짐

    digitalWrite(LED\_D, HIGH); // 아래쪽 LED 꺼짐

  }

  else if(val\_ud>768)

  {

    digitalWrite(LED\_U,HIGH); //위쪽 LED 꺼짐

    digitalWrite(LED\_D,LOW); //아래쪽 LED 켜짐

  }

  else

  {

    digitalWrite(LED\_U,HIGH); //위쪽 LED 꺼짐

    digitalWrite(LED\_D,HIGH); //아래쪽 LED 꺼짐

  }

  if(val\_rl<256)

  {

    digitalWrite(LED\_R,HIGH); //오른쪽 LED 꺼짐

    digitalWrite(LED\_L,LOW); //왼쪽 LED 켜짐

  }

  else if(val\_rl>768)

  {

    digitalWrite(LED\_R,HIGH); //오른쪽 LED 꺼짐

    digitalWrite(LED\_L,LOW); //왼쪽 LED 켜짐

  }

  else

  {

    digitalWrite(LED\_R,HIGH); // 오른쪽 LED 꺼짐

    digitalWrite(LED\_L,HIGH); //왼쪽 LED 꺼짐

  }

  delay(200); // 짧은 시간지연

}

**3.5.1 온도센서 119p**

/\*

LM35 온도센서 출력 예제

\*/

void setup()

{

  Serial.begin(9600); //통신속도 9600으로 통신 시작

}

void loop()

{

  int temp = analogRead(A0); //온도 값을 읽고 변수에 저장

  int value = 5.0\*temp\*100.0/1024; // 센서값 섭씨온도로 변환 (LM35 계산 공식에 의함)

  Serial.print(value);

  Serial.println("C");

  delay(1000);

}

**3.5.2 온습도센서 121p**

#include "DHT.h" // DHT 라이브러리 호출

#define DHTPIN 5 // 온습도 센서가 5번에 연결

#define DHTTYPE DHT11 // DHT11 온습도 센서 사용

DHT dht(DHTPIN, DHTTYPE); // DHT 설정(5,DHT11)

void setup(){

  Serial.begin(9600); // 통신속도 9600으로 통신 시작

  Serial.println("DHT11 test!"); // 문자 출력

}

void loop()

{

  int h = dht.readHumidity(); // 습도값을 h에 저장

  int t = dht.readTemperature(); // 온도값을 t에 저장

  Serial.print("Humidity:"); //문자열 출력

  Serial.print(h); // 습도값 출력

  Serial.print("%");

  Serial.print("Temperature:");

  Serial.print(t); // 온도값 출력

  Serial.print("C");

  delay(2000);

}

**Quest 11. 123p**

**온도를 0.5초 간격으로 시리얼 모니터에 출력한다. 30도 초과일 때 LED에 빨간불, 20도 미만일 때 파란불, 20~30도일 때 초록불이 들어오게 회로를 구성하여 구동해라.**

int LED1 = 8; //Red를 8번에 연결

int LED2 = 9; //Green을 9번에 연결

int LED3 = 10; //Blue를 핀 10번에 연결

void setup(){

  Serial.begin(9600); //통신속도 9600으로 통신 시작

  pinMode(LED1, OUTPUT); // LED출력 설정

  pinMode(LED2, OUTPUT); // LED 출력 설정

  pinMode(LED3, OUTPUT); // LED 출력 설정

  digitalWrite(LED1, LOW); //빨강 LED 꺼짐으로 초기설정

  digitalWrite(LED2, LOW); //초록 LED 꺼짐으로 초기설정

  digitalWrite(LED3, LOW); //파랑 LED 꺼짐으로 초기설정

}

void loop()

{

  int temp = analogRead(A0); // 온도센서 값을 변수에 저장

  int cels = 5.0\*temp\*100.0/1024; // 섭씨온도로 변환하여 변수에 저장

  if(cels>30) // 온도가 30도 초과일 때

  {

    digitalWrite(LED1, HIGH); // 빨강 LED 켜짐으로 초기설정

    digitalWrite(LED2, LOW); //초록 LED 꺼짐으로 초기설정

    digitalWrite(LED3, LOW); //파랑 LED 꺼짐으로 초기설정

  }

  else if(cels<20) //온도가 20도 미만일 때

  {

    digitalWrite(LED1, LOW); //빨강 LED 꺼짐으로 초기설정

    digitalWrite(LED2, LOW); //초록 LED 꺼짐으로 초기설정

    digitalWrite(LED3, HIGH); //파랑 LED 켜짐으로 초기설정

  }

  else// 온도가 20~30도 사이일 때

  {

    digitalWrite(LED1, LOW); //빨강 LED 꺼짐으로 초기설정

    digitalWrite(LED2, HIGH); //초록 LED 켜짐으로 초기설정

    digitalWrite(LED3, LOW); //파랑 LED 꺼짐으로 초기설정

  }

  Serial.print(cels);

  Serial.println("C");

  delay(500);

}

**3.6.1 조도센서 127p**

/\*

조도센서 LED 예제

어두우면(아날로그값 중간 이하) led의 불이 켜지는 예제

\*/

int led = 12; // 12번핀 led를 변수 led에 대입

void setup()

{

  pinMode(led, OUTPUT); // led를 출력모드로 설정

  Serial.begin(9600); // 속도 9600으로 통신 시작

}

void loop()

{

  int val = analogRead(A1); // A1 아날로그 값을 읽어서 val에 저장

  Serial.print(val); // val 값을 출력

  if(val<512) // 아날로그 값이 중간 미만이라면 LED를 켜라

  {

    digitalWrite(led, LOW);

  }

  else // 아날로그 값이 중간 이상이라면 LED를 꺼라

  {

    digitalWrite(led, HIGH);

  }

  delay(100);

}

**3.6.2 불꽃감지 센서 130p**

/\*

불꽃센서로 테스트 예제

\*/

int led1 = 5; // led1 5번에 연결(빨강 LED)

int led2 = 6;// led2 6번에 연결(파랑 LED)

int val;

void setup()

{

  pinMode(led1,OUTPUT); // led 출력설정

  pinMode(led2,OUTPUT);

  pinMode(A0, INPUT); // 불꽃 감지센서 입력설정

  Serial.begin(9600); //통신속도 9600으로 통신 시작

}

void loop()

{

  val = analogRead(A0); //불꽃 감지센서에서 값을 읽음

  Serial.println(val); // 불꽃 감지센서 입력값 출력

  //불꽃 감지센서 입력값이 800 이상이면 (불꽃을 감지하면)

  //led 1 on, led 2 off

  if(val>=800) // 아날로그값 800은 조정 가능

  {

    digitalWrite(led1,HIGH);

    digitalWrite(led2,LOW);

  }

  else // 불꽃 감지센서 입력값이 800 이하면 반대

  {

    digitalWrite(led1,LOW);

    digitalWrite(led2,HIGH);

  }

}

**3.6.3 IR 리모컨 134p**

/\*

리모컨으로 적외선 센서에 값을 넣고, led를 제어

\*/

#include <IRremote.h>

int RECV\_PIN = 11; // IR 수신기 D11에 연결

int LED1 = 7; // LED1은 7번에 연결

int LED2 = 8; //LED2는 8번에 연결

IRrecv irrecv(RECV\_PIN); //IR 리모컨 사용을 위한 클래스 생성

decode\_results decodedSignal; // IR 리모컨으로부터 받은 데이터

void setup()

{

  Serial.begin(9600); // 시리얼 통신 시작

  irrecv.enableIRIn(); // 리모컨 수신 시작

  pinMode(LED1,OUTPUT); //LED 출력모드로 설정

  pinMode(LED2,OUTPUT);

}

void loop()

{

  if(irrecv.decode(&decodedSignal)==true) // IR 수신값이 있는지 판단

  {

    Serial.println(decodedSignal.value);

    if(decodedSignal.value == 0xFFA25D) // 리모컨의 ch-버튼이 눌리면 LED 1 ON

    {

      digitalWrite(LED1, HIGH);

    }

    if(decodedSignal.value == 0xFFE21D) // 리모컨의 ch+버튼이 눌리면 LED 1 OFF

    {

      digitalWrite(LED1, LOW);

    }

    if(decodedSignal.value == 0xFF22DD) // 리모컨의 prev버튼이 눌리면 LED 2 ON

    {

      digitalWrite(LED2, HIGH);

    }

     if(decodedSignal.value == 0xFFC23D) // 리모컨의 play버튼이 눌리면 LED 2 OFF

    {

      digitalWrite(LED2, LOW);

    }

    irrecv.resume(); // IR 다음 데이터 수신

  }

}

**Quest 12 136p**

**조도센서의 값을 0.5초마다 시리얼 모니터에 출력하고, 감지되는 밝기가 밝을수록 LED도 밝아지는 회로를 구성하고 구동하시오.**

int led = 11; //LED를 11번핀에 연결

void setup()

{

  pinMode(led,OUTPUT); // LED를 출력모드로 설정

  Serial.begin(9600); // 속도 9600으로 통신 시작

}

void loop()

{

  int val = analogRead(A0); //A0 아날로그 값을 읽어서 val에 저장

  Serial.println(val); //val 값을 출력

  //조도센서의 val 아날로그 값의 범위는 0~1023, 밝을수록 저항이 낮아져서 아날로그 값은 낮아짐.

  //led 의 pwm 값의 범위는 0~225, pwm이 높을수록 밝아지기 때문에, map 함수를 사용하여 변환.

  int brightness = map(val,0,1023,255,0);

  analogWrite(led,brightness);

  delay(500);

}

**3.7.1 사운드 센서 140p**

/\*

소리 감지 모듈을 사용해서 led를 제어하는 예제

\*/

int led = 9;

void setup()

{

  pinMode(led,OUTPUT); // led를 출력으로 설정

  Serial.begin(9600); // 통신속도 9600bps로 시리얼 통신 시작

}

void loop()

{

  int value = analogRead(A2); //소리감지 센서값을 읽어서 변수 value에 저장

  //map() 함수 사용 map(value, fromLow, fromHigh, toLow, toHigh)

  //읽어들인 값으로 LED의 밝기를 구하고, LED의 밝기를 새로고침 합니다.

  int adjust = map (value,0,1024,0,255);

  analogWrite(led,adjust);

  Serial.println(value);//센서값을 시리얼 모니터로 출력

  delay(100);

}

**3.7.2 버저 143p**

int buzzer = 5; //버저를 핀 5번에 연결

int sw = 4; //스위치를 핀 4번에 연결

//'도레미파솔라시도' 음계 주파수 배열로 선언

int melody[]={262, 294, 330, 349, 392, 440, 494, 523};

void setup()

{

  pinMode(buzzer,OUTPUT); // 핀 5번을 출력 설정

  pinMode(sw,INPUT\_PULLUP); //핀 4번을 내부풀업으로 입력설정

}

void loop()

{

  if(digitalRead(sw)==0)

  {

    for(int a=0; a<8; a++)

    {

      tone(buzzer,melody[a],400); //0.4초동안 음계 출력

      delay(500); //0.5초 대기

    }

  }

  else

  {

    noTone(5); //피에조 버저 off

  }

}

**Quest 13 145p**

**상어가족 첫 소절을 무한히 연주하는 회로를 구성하고 구동하시오.**

int buzzer = 5; //버저를 핀 5에 연결

//쉼표

int rest = 0;

//계이름을 순서대로 배열로 선언

int melody[] = {294,330,392,392,392,392,392,392,392,294,330,392,rest,

392,392,392,392,392,294,330,392,rest,

392,392,392,392,392,392,392,370,370,rest};

//연주 유지시간을 배열로 선언

int damper[] = {400,400,200,200,200,100,300,100,200,200,200,100,30,

200,100,300,100,200,200,200,100,30,

200,100,300,100,200,200,200,200,200,3000};

void setup()

{

  pinMode(buzzer,OUTPUT); //핀 5번을 출력 설정

}

void loop()

{

  for(int a=0; a<32; a++)

  {

    tone(5,melody[a],damper[a]); //각 음계와 유지시간으로 버저 연주

    delay(damper[a]+100); //유지시간+100ms만큼 대기하여 연주 박자 유지

  }

}

**3.8.1 LCD 151p**

/\*

led 시리얼 테스트

\*/

#include <Wire.h> // l2C 통신을 위해 Wire 라이브러리 사용

#include <LiquidCrystal\_I2C.h> //LiquidCRystal\_l2c 라이브러리 사용

//객체이름을 lcd로 선언 주소값을 0x27로 설정, 16 문자 2줄 디스플레이

//만약 lcd가 출력이 안된다면 주소값을 0x3F로 수정

LiquidCrystal\_I2C lcd(0x27, 16, 2);

void setup()

{

  lcd.begin(16,2); //lcd 시작

  lcd.backlight(); //lcd 백라이트 on

  Serial.begin(9600); //속도 9600bps로 시리얼 통신시작

  Serial.println("LCD 통신시작"); // 시리얼 통신 시작 문자 출력

}

void loop()

{

  if(Serial.available()) // 시리얼 통신으로 문자열이 도착하면

  {

    delay(10); //메시지가 모두 도착하도록 잠시 대기

    lcd.clear();  //lcd 내용 삭제 (초기화)

    //받은 문자들을 시리얼 모니터와 lcd에 출력

    while (Serial.available()>0)

    {

      char val = Serial.read(); // 받은 문자열을 val에 저장

      if(val>0)

      {

        Serial.print(val); //val 값을 시리얼 모니터에 출력

        lcd.print(val); //val 값을 lcd에 출력

      }

      else

      {

        Serial.print("16자 이내로 입력해주세요");

        lcd.print("plz 16char below");

      }

    }

  }

}

**3.8.2 서보모터 154p**

#include <Servo.h> //서보모터 라이브러리 사용

Servo myservo; // 서보모터 객체를 myservo로 생성

int potentiometer = A0; //가변저항 A0에 연결

int val; // val 변수 선언

void setup()

{

  myservo.attach(9); //서보모터 9번핀에 연결

}

void loop()

{

  val = analogRead(potentiometer); //가변저항에서 읽은 아날로그 값을 val에 저장

  //map 함수를 이용해 아날로그값 0~1023을 0~180으로 조정한 후 val 에 저장

  val = map (val,0,1012,0,180);

  myservo.write(val); //val 값에 따라 서보모터 이동

  delay(15); //서보모터가 움직일 시간 15ms 딜레이

}

**3.8.3 스테퍼모터 157p**

/\*

스테퍼모터와 모터드라이버는 각각 28BYJ-48과 UNL2003을 사용

\*/

#include <Stepper.h> // 스테퍼모터 라이브러리 사용

const int steps = 2048; // 모터의 스텝수 2048 = 1바퀴, 1024 = 반바퀴

//myStepper 객체 생성(스텝 수, lN4,lN3,lN2,lN1)

//lN1~4까지 각각 아두이노 D8~D11번에 연결

Stepper myStepper(steps,11,9,10,8);

void setup(){

  myStepper.setSpeed(15); // 스테퍼모토 속도 (rpm) 설정, 스텝 수와 연결되어 있음

  Serial.begin(9600); // 9600bps로 시리얼통신 시작

}

void loop()

{

  Serial.println("시계방향");

  myStepper.step(steps); // 시계방향으로 스텝 수만큼 이동

  delay(500); //모터 정지 후 잠시 대기

  Serial.println("시계반대방향");

  myStepper.step(-steps); // 반시계방향으로 스텝 수만큼 이동

  delay(500); //모터 정지 후 잠시 대기

}

**Quest 14**

**스테퍼모터와 map 함수를 사용해서 한 바퀴를 60등분하여 시계 초침처럼 1초에 6도 이동 후 대기하는 회로를 구성하고 구동하시오.**

#include <Stepper.h> //스테퍼모터 라이브러리 사용

const int steps = 2048; //모터의 스텝수 2048 =1바퀴

//myStepper 객체 생성(스텝 수, lN4, lN3,lN2, lN1)

//lN1~4까지 각각 아두이노 D8~D11번에 연결

Stepper myStepper(steps,11,9,10,8);

void setup()

{

  myStepper.setSpeed(10); // 스테퍼모터 속도 (rpm)설정, 스텝 수와 연결되어 있음

}

void loop()

{

  int one\_sec = map(1,0,60,0,steps);

  myStepper.step(one\_sec); // 시계방향으로 스텝 수만큼 이동, 10rpm은 6도 이동시 0.1초 소요

  delay(900); // 0.9초 대기

}

**Final**

1. **리모컨의 1번부터 9번까지 눌렀을 때 버저에서 각각 [도,미,솔샾, 라,시,높은도,높은레,높은레샾,높은미]의 음의 0.1초간 출력된다.**
2. **LCD에 현재 출력되고 있는 음을 [Do, Mi, Sol#, La, Si, High Do, High Re, High Re#, High Mi] 의 문자열로 출력한다.**
3. **989895764, 1245, 2356, 989895764, 1245, 2654 순서대로 리모컨을 눌러 연주한다.**

#include <Wire.h> // l2C 통신을 위해 Wire 라이브러리 사용

#include <LiquidCrystal\_l2C.h>

#include <lRremote.h>

int RECV\_PIN = 9; // lR 수신기 D9에 연결

int buzzer = 10; // 버저를 핀 10번에 연결

int melody[]={262, 330, 415, 440, 494, 523, 587, 622, 659};

//객체이름을 lcd로 선언 l2C 주소값을 0x27로 설정, 16문자 2줄 디스플레이

//만약 lcd가 출력이 안되면 주소값을 0x3F로 수정

LiquidCrystal\_l2C lcd(0x27,16,2);

lRrecv irrecv(RECV\_PIN); // IR 리모컨 사용을 위한 클래스 생성

decode\_results decodedSignal; //IR 리모컨으로부터 받은 데이터

void setup()

{

  lcd.begin(); //lcd 시작

  lcd.backlight(); //lcd 백사이트 on

  irrecv.enableRln(); //리모컨 수신 시작

}

void loop()

{

  if(irrecv.decode(&decodedSignal)==true)//lR 수신값이 있는지 판단,

  {

    if(decodedSignal.value == 0xFF30CF) //1번이 눌렸을 때

    {

      tone(buzzer, melody[0],100);

      lcd.clear(); // lcd 초기화

      lcd.print("Do");

    }

    if(decodedSignal.value == 0xFF18E7) //2번이 눌렸을 때

    {

      tone(buzzer, melody[1],100);

      lcd.clear(); // lcd 초기화

      lcd.print("Mi");

    }

    if(decodedSignal.value == 0xFF7A85) //3번이 눌렸을 때

    {

      tone(buzzer, melody[2],100);

      lcd.clear(); // lcd 초기화

      lcd.print("Sol#");

    }

    if(decodedSignal.value == 0xFF10EF) //4번이 눌렸을 때

    {

      tone(buzzer, melody[3],100);

      lcd.clear(); // lcd 초기화

      lcd.print("La");

    }

    if(decodedSignal.value == 0xFF38C7) //5번이 눌렸을 때

    {

      tone(buzzer, melody[4],100);

      lcd.clear(); // lcd 초기화

      lcd.print("Si");

    }

    if(decodedSignal.value == 0xFF5AA5) //6번이 눌렸을 때

    {

      tone(buzzer, melody[5],100);

      lcd.clear(); // lcd 초기화

      lcd.print("High Do");

    }

    if(decodedSignal.value == 0xFF42BD) //7번이 눌렸을 때

    {

      tone(buzzer, melody[6],100);

      lcd.clear(); // lcd 초기화

      lcd.print("High Re");

    }

    if(decodedSignal.value == 0xFF4AB5) //8번이 눌렸을 때

    {

      tone(buzzer, melody[7],100);

      lcd.clear(); // lcd 초기화

      lcd.print("High Re#");

    }

      if(decodedSignal.value == 0xFF52AD) //9번이 눌렸을 때

    {

      tone(buzzer, melody[8],100);

      lcd.clear(); // lcd 초기화

      lcd.print("High Mi");

    }

    delay(100);//0.1초 대기

    irrecv.resume(); //IR 다음 데이터 수신

  }

}

**Quest 15**

1. **온도를 측정하여 일정 시간마다 시리얼 모니터에 출력한다.**
2. **출력하는 시간 간격은 포텐셔미터로 조정하며 최소 0.5초에서 최대 2초 간격으로 조절 가능하다.**
3. **초기에는 LED가 켜져있고 이 때는 시리얼 모니터에 섭씨로 온도가 출력된다.**
4. **스위치를 누르고 있을 때는 LED가 꺼지며 시리얼 모니터에 화씨로 온도가 출력된다.**
5. **LM 35 센서값 섭씨 변환 공식 : 5.0 x 센서값 x 100.0 / 1024**
6. **화씨온도 구하는 공식 : (섭씨온도 x 9/5) + 32**

int led = 8; //led를 8번핀에 연결

int sw = 9; //스위치를 9번핀에 연결

void setup()

{

  Serial.begin(9600); //통신속도 9600으로 통신 시작

  pinMode(led,OUTPUT);// LED를 출력모드로 설정

  pinMode(sw,INPUT);// 스위치를 입력모드로 설정

  digitalWrite(led,LOW);//LED를 켜진 상태로 초기허ㅣ

}

void loop()

{

  int temp = analogRead(A0); //온도 값을 읽고 변수에 저장

  int temp\_c = 5.0\*temp\*100.0/1024; //센서값 섭씨온도로 변환

  int temp\_f = (temp\_c\*9/5)+32; //섭씨온도 화씨온도로 변환

  int key\_scan = digitalRead(sw); //스위치 값을 읽고 변수에 저장

  int term = analogRead(A1); //가변저항 값을 읽고 변수에 저장

  term = map(term,0,1023,500,2000); //가변저항 값을 0.5초에서 2초 범위로 변환

  if(key\_scan ==1)//스위치가 눌리지 않았을 때

  {

    digitalWrite(led,LOW); //LED를 킨다

    Serial.print(temp\_c);//섭씨온도 출력

    Serial.println("C");

  }

  else //스위치가 눌리고 있을 때

  {

    digitalWrite(led,HIGH); //LED를 끈다

    Serial.print(temp\_f); //화씨온도 출력

    Serial.println("F");

  }

  delay(term); //가변저항에 의해 조절된 출력시간 간격

}