
***** 아두이노기초 기말시험. 2018. 12. 13 (목) *****

아두이노 코드를 읽고 쓰면서 하드웨어를 즐기자!



1-2. 다음은 디지털 신호 입력시간을 측정하는 아두이노 코드이다.
 밑줄 친 곳에 알맞은 코드는?

<pre> /* 디지털 신호 입력 시간 측정하기 */ // 2번핀을 스위치 입력으로 설정 const int inputPin = 2; // 현재의 시간을 저장하기 위한 변수 long startTime = 0; // 실제 스위치가 눌린 후 지연되는 시간 long swCountTimer = 0; void setup() { // 스위치 입력을 위하여 2번핀을 입력으로 // 설정하고 풀업시킨다 pinMode(inputPin, [1]_____INPUT_PULLUP); // 시리얼 통신을 설정한다 Serial.begin(9600); } </pre>	<pre> void loop(){ // 스위치 입력이 발생하였을 경우 실행 if(digitalRead(inputPin) == LOW){ // 현재의 시간을 startTime 변수에 넣는다. startTime = millis(); // 스위치가 입력되는 동안 지연시킨다. while(digitalRead(inputPin) == LOW); // swCountTimer 변수에 스위치가 // 눌러진 시간을 넣는다. // 여기까지 측정된 시간에서 앞서 저장 // 한 시간이 스위치가 눌러진 시간이 된다 swCountTimer = [2]__millis() - startTime; // 시리얼 통신으로 값을 출력한다. Serial.print(swCountTimer); Serial.println(" ms"); } } </pre>
---	--

1. 스위치 입력을 위하여 2번핀을 입력으로 설정하고 풀업시키는 설정은?

- | | |
|--------------------|------------------|
| A. INPUT_PULLDOWN | B. INPUT_PULLUP |
| C. OUTPUT_PULLDOWN | D. OUTPUT_PULLUP |

2. 스위치가 눌러진 시간을 측정한다. 밑줄친 곳에 알맞은 코드는?

- | | |
|-------------------------|-------------------------|
| A. startTime | B. startTime - millis() |
| C. millis() - startTime | D. millis() |

3-4. 다음은 포텐쇼미터 입력으로 LED의 밝기를 제어하는 아두이노 코드이다.
 밑줄친 곳에 알맞은 코드는?

<pre> /* 포텐쇼미터 입력으로 LED의 PWM 제어 */ // 0번 아날로그핀을 포텐쇼미터 입력으로 // 설정한다. const int potentiometerPin = 0; //13번 핀에 연결되어 있는 내장 LED를 // 출력으로 사용한다. const int ledPin = 13; // #3 pin is defined to PWM output pin const int pwmOutputPin = 3; void setup() { // 13번 핀을 출력으로 설정한다. 아날로그 // 입력핀은 설정이 불필요하다. pinMode(ledPin, OUTPUT); Serial.begin(9600); } void loop(){ int adcValue; int duty; int pwm; </pre>	<pre> // 포텐쇼미터 값을 읽는다. adcValue = analogRead(potentiometerPin); // 포텐쇼미터 값을 0~100의 범위로 변경한다. duty = map(adcValue, 0, 1023, 0, 100); // LED를 duty 비만큼 점등한다. digitalWrite(ledPin, HIGH); delay(duty); // 나머지 시간에는 소등시킨다. digitalWrite(ledPin, LOW); delay([3]_____100-duty); // pwmOutputPin Led ON pwm = [4]_____map(adcValue, 0, 1023, 0, 255); analogWrite(pwmOutputPin,pwm); // 시리얼 통신으로 ADC 값과 Duty를 출력한다. Serial.print("ADC Value is "); Serial.print(adcValue); Serial.print(". Duty cycle is "); Serial.print(duty); Serial.println(" %"); } </pre>
--	---

3. 내장 LED를 duty 비만큼 점등 후 남은 비율로 소등하는 설정은?

- A. 1023 - duty B. duty - 100
 C. duty - 1023 D. 100 - duty

4. 포텐쇼미터 입력으로 LED의 밝기를 제어한다. 밑줄 친 곳에 알맞은 코드는?

- A. map(adcValue, 0, 1023, 0, 100) B. map(adcValue, 0, 1023, 0, 255)
 C. map(duty, 0, 1023, 0, 100) D. map(duty, 0, 1023, 0, 255)

5-6. 다음은 빛 입력을 측정하여 LED를 점등/소등하는 아두이노 코드이다.

밑줄 친 곳에 알맞은 코드는?

<pre> /* 빛 입력을 측정하여 LED 점등/소등 */ // I2C 통신 라이브러리 설정 #include <Wire.h> // I2C LCD 라이브러리 설정 #include <LiquidCrystal_I2C.h> // LCD I2C address 설정 LiquidCrystal_I2C lcd(0x3f,16,2); // 0번 아날로그핀을 CdS 셀 입력으로 설정 const int CdSPin = 0; const int ledPin = 13; // 빛 입력 문턱값 설정 const int threshold = 512; void setup() { lcd.init(); // LCD 설정 lcd.backlight(); // 백라이트를 켜다. // 메시지를 표시한다. lcd.setCursor(0,1); lcd.print("CdS Cell Test"); delay(3000); // 3초동안 메시지를 표시 lcd.clear(); lcd.setCursor(0,0); lcd.print("ADC : "); lcd.setCursor(0,1); lcd.print("Illuminance:"); lcd.setCursor(15,1); lcd.print("%"); pinMode(ledPin,OUTPUT); } </pre>	<pre> void loop(){ int adcValue; int illuminance; // 현재의 밝기. 0~100% // CdS cell을 통하여 입력되는 전압을 읽는다. adcValue = analogRead(CdSPin); // 아날로그 입력 값을 0~100의 범위로 변경한다. illuminance = [5]_____map(adcValue, 0, 1023, 100, 0); // 전에 표시했던 내용을 지우고 // LCD에 ADC 값과 밝기를 출력한다. // 전에 표시했던 내용을 지운다. lcd.setCursor(9,0); lcd.print(" "); // ADC 값을 표시한다 lcd.setCursor(9,0); lcd.print(adcValue); // 전에 표시했던 내용을 지운다. lcd.setCursor(13,1); lcd.print(" "); lcd.setCursor(12,1); lcd.print(illuminance); // 밝기를 표시한다 if([6]_____adcValue <= threshold) digitalWrite(ledPin, LOW); else digitalWrite(ledPin, HIGH); delay(1000); } </pre>
---	---

5. CdS(조도 센서) 입력 값을 CdS의 특성을 반영해서 0~100의 범위로 지정하는 설정은?

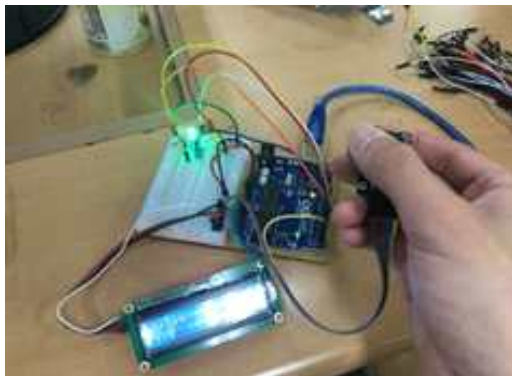
- | | |
|--|-----------------------------------|
| A. <u>map(adcValue, 0, 1023, 100, 0)</u> | B. map(adcValue, 0, 1023, 0, 100) |
| C. map(adcValue, 0, 100, 1023, 0) | D. map(adcValue, 0, 100, 100, 0) |

6. 측정되는 빛 입력값이 지정된 빛 입력 문턱값보다 작으면 LED가 꺼지고, 문턱값보다 크면 LED가 켜진다. 밑줄친 곳에 알맞은 코드는?

- | | |
|-----------------------------|------------------------------------|
| A. illuminance <= threshold | <u>B. adcValue <= threshold</u> |
| C. illuminance >= threshold | D. adcValue >= threshold |

7-8. 다음은 조이스틱 입력을 이용해서 LED로 방향을 표시하는 아두이노 코드이다.
 밑줄 친 곳에 알맞은 코드는?

<pre> /* 조이스틱 입력으로 LED를 이용한 방향 표시 */ #include <Wire.h> #include <LiquidCrystal_I2C.h> LiquidCrystal_I2C lcd(0x27,16,2); // 0번 아날로그핀을 X 축 입력으로 설정 const int xAxisPin = 0; // 1번 아날로그핀을 Y 축 입력으로 설정 const int yAxisPin = 1; // 2번 디지털 입력 핀을 Z 축 입력으로 설정 const int zAxisPin =2; // LED pin 설정 const int wShoot = 3; const int rStop = 5; const int bGo = 6; const int yRight = 9; const int yLeft = 10; void setup() { pinMode(zAxisPin,INPUT_PULLUP); lcd.init(); // LCD 설정 lcd.backlight(); // 백라이트를 켜다. lcd.setCursor(0,1); lcd.print("Using Joystick"); // 3초동안 메시지를 표시한다. delay(3000); // 모든 메시지를 삭제한 뒤 // X축 Y축 문자를 출력한다. lcd.clear(); lcd.setCursor(0,0); lcd.print("X:"); lcd.setCursor(0,1); lcd.print("Y:"); lcd.setCursor(15,1); pinMode(wShoot,OUTPUT); pinMode(rStop,OUTPUT); pinMode(bGo,OUTPUT); pinMode(yRight,OUTPUT); pinMode(yLeft,OUTPUT); } </pre>	<pre> void loop(){ // X, Y, Z 축 값을 읽는다. int xValue = [7.A]_____analogRead(xAxisPin); int yValue = [7.B]_____analogRead(yAxisPin); int zValue = [7.C]_____digitalRead(zAxisPin); // 그래프를 그리기 위해서 X, Y 값을 조절한다. int xDisplay = map(xValue,0,1023,6,15); int yDisplay = map(yValue,0,1023,6,15); if(xValue>600){ digitalWrite(bGo,HIGH); }else if(xValue<500){ digitalWrite(rStop,HIGH); }else if(yValue>600){ digitalWrite(yRight,HIGH); }else if(yValue<500){ digitalWrite(yLeft,HIGH); } else{ digitalWrite(bGo,LOW); digitalWrite(rStop,LOW); digitalWrite(yRight,LOW); digitalWrite(yLeft,LOW); } // 첫 째 줄에 전에 표시했던 내용을 지운다. lcd.setCursor(2,0); lcd.print(" "); // 14칸 공백 // X 축의 ADC 값을 출력한다. lcd.setCursor(2,0); lcd.print(xValue); // 조이스틱의 X 값에 따라 그래프를 출력한다 lcd.setCursor(xDisplay,0); lcd.print(" "); } </pre>
---	--



```
// 둘째 줄에 전에 표시했던 내용을 지운다.
lcd.setCursor(2,1);
lcd.print("          "); // 14칸 공백
// Y 축의 ADC 값을 출력한다.
lcd.setCursor(2,1);
lcd.print(yValue);
// 조이스틱의 Y 값에 따라 그래프를 출력한다
lcd.setCursor(yDisplay,1);
lcd.print("|");

// Z 방향으로 눌렀을 때 백라이트를 점멸하고
// LED 점등.
if(zValue == [8]____LOW){
    lcd.noBacklight();
    digitalWrite(wShoot,HIGH);
}
else{
    digitalWrite(wShoot,LOW);
    lcd.backlight();
}

delay(100);
}
```

7. Joystick의 세 방향 출력 값들을 읽는 설정은 (A, B, C 순서대로 무엇인가)?

- A. analogRead, analogRead, analogRead B. digitalRead, digitalRead, analogRead
C. digitalRead, digitalRead, digitalRead D. analogRead, analogRead, digitalRead

8. Z 방향으로 눌렀을 때 백라이트를 점멸하고, LED가 점등된다.
밑줄친 곳에 알맞은 코드는?

- A. LOW B. HIGH
C. xDisplay D. yDisplay

9-10. 다음은 포텐쇼미터 입력으로 속도를 제어하는 스텝모터 구동 아두이노 코드이다.
 밑줄 친 곳에 알맞은 코드는?

<pre> /* 포텐쇼미터로 속도를 제어하는 스텝모터 구동 */ // 스텝 모터 신호핀 설정 int motorPin1 = 8; int motorPin2 = 9; int motorPin3 = 10; int motorPin4 = 11; // 포텐쇼미터 핀 설정 int potentiometerPin = 0; // stop 구간 폭 설정 int stopRange = 100; // 모터 속도 관련 변수 설정 int motorSpeed; // 스텝 사이의 지연시간 // 으로서 4500~1000의 범위를 갖는다. int motorSpeedPercent; // 속도를 0~100%로 나타낸다. // 스텝 모터의 스텝 설정 //0~7은 동작신호, 8번째는 모터정지 신호 int steps[] = {B1000, B1100, B0100, B0110, B0010, B0011, B0001, B1001, B0000}; void setup() { //모터 신호핀을 출력으로 설정 pinMode(motorPin1, OUTPUT); pinMode(motorPin2, OUTPUT); pinMode(motorPin3, OUTPUT); pinMode(motorPin4, OUTPUT); // 시리얼 통신 설정 Serial.begin(9600); } </pre>	<pre> void loop(){ // 포텐쇼미터 값을 읽어옴 int potentiometer = analogRead(potentiometerPin); // CW로 회전시 모터를 CW방향으로 회전시킨다. if(potentiometer >= [9]___512+(stopRange/2)){ //모터의 속도를 계산한다. motorSpeed=map(potentiometer,[9]___512+(stopRange/2), 1023,4500,1000); // 모터의 속도를 백분율로 변환시킨다. motorSpeedPercent=map(motorSpeed,4500,1000, 1,100); // 시리얼 통신 메시지를 출력한다. Serial.print("CW Motor Speed: "); Serial.print(motorSpeedPercent); Serial.println("%"); // CW로 회전시킨다. clockwise(); } // CCW로 회전시 모터를 CW방향으로 회전시킨다. else if(potentiometer <= [9]___512-(stopRange/2)){ //모터의 속도를 계산한다. motorSpeed = map(potentiometer,[9]___512-(stopRange/2), 0,4500,1000); // 모터의 속도를 백분율로 변환시킨다. motorSpeedPercent = map(motorSpeed, 4500,1000, 1,100); // 시리얼 통신 메시지를 출력한다. Serial.print("CCW Motor Speed: "); Serial.print(motorSpeedPercent); Serial.println("%"); // CCW로 회전시킨다. counterClockwise(); } // 중간에 위치했을 경우 정지시킨다. else{ Serial.println("Motor Stop"); motorStop(); } } </pre>
---	---

```

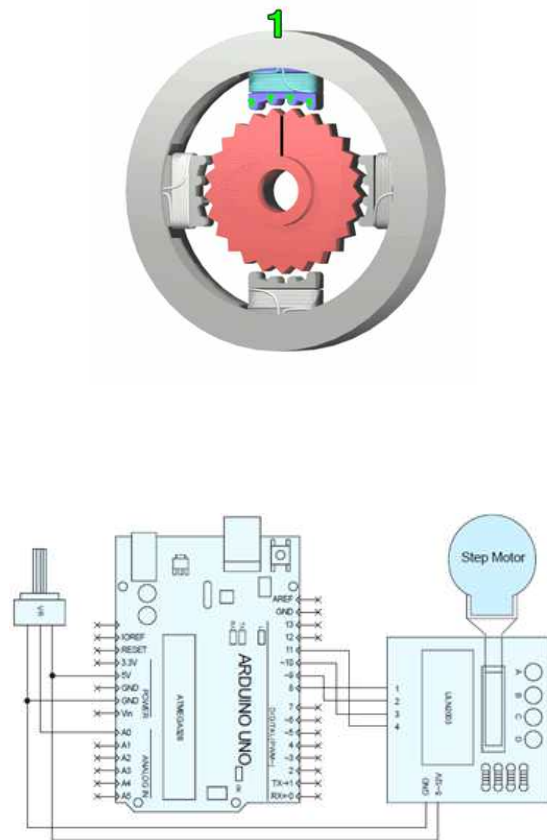
void counterClockwise(){
    // 0~7 번째 신호를 순차적으로 출력시킨다.
    for(int i = 0; i < 8; i++)
    {
        motorSignalOutput(i);
        delayMicroseconds(motorSpeed);
    }
}

void clockwise(){
    // 7~0 번째 신호를 순차적으로 출력시킨다.
    for(int i = 7; i >= 0; i--)
    {
        motorSignalOutput(i);
        delayMicroseconds(motorSpeed);
    }
}

void motorStop(){
    // 정지신호를 출력시킨다.
    motorSignalOutput([10]__8);
}

void motorSignalOutput(int out)
{
    // out 변수에 해당하는 모터 시그날을 출력
    digitalWrite(motorPin1, bitRead(steps[out], 0));
    digitalWrite(motorPin2, bitRead(steps[out], 1));
    digitalWrite(motorPin3, bitRead(steps[out], 2));
    digitalWrite(motorPin4, bitRead(steps[out], 3));
}

```




9. 시계방향(CW), 반시계방향(CCW)으로 회전시키는 기준이 되는 값은 ?

- A. 0 B. 50
C. 128 D. 512

10. 스텝모터를 정지시키기 위한 매개변수의 값은? --- (8)

11-12. 다음은 적외선 리모컨 코드 읽는 아두이노 코드이다.
 밑줄 친 곳에 알맞은 코드는?

<pre> /* 적외선 리모컨 코드 읽기 */ // 적외선 리모컨 라이브러리를 불러온다. #include <IRremote.h> // 적외선 수신부가 연결될 핀을 설정한다. int irPin = 11; // 적외선 수신부가 연결된 핀을 리모컨 // 수신 핀으로 설정한다. IRrecv irrecv(irPin); // 수신된 신호의 결과를 results 변수로 // 설정한다. decode_results results; void setup() { // 시리얼 통신을 설정한다. Serial.begin(9600); // 적외선 리모컨 수신을 시작한다. irrecv.enableIRIn(); // 13번 핀에 연결된 LED를 리모컨 수신시 // 점멸시킨다. irrecv.blink13(true); } </pre>	<pre> void loop() { // 수신된 코드가 있을 때 실행한다. if ([11]_____irrecv.decode(&results)){ // 0xFFFFFFFF 값을 제외하고 출력한다. if(results.value != 0xFFFFFFFF){ // 수신된 값을 16진수 형태로 출력한다. Serial.print("Received Code is "); Serial.println([12]_____results.value, HEX); } // 다음 수신을 위해서 준비한다. irrecv.resume(); } } </pre> 
---	---

11. 수신된 코드가 있을 경우, 코드를 판독하는 조건은?

- | | |
|---------------------------------------|---------------------------|
| A. <u>irrecv.decode(&results)</u> | B. irrecv.decode(results) |
| C. irrecv.encode(&results) | D. irrecv.encode(results) |

12. 수신된 값을 16진수 형태로 출력한다. 밑줄 친 곳에 알맞은 코드는?

- | | |
|------------------------------|-----------------------|
| A. results.code, HEX | B. results.code, 0xF |
| C. <u>results.value, HEX</u> | D. results.value, 0xF |

13-14. 다음은 피에조 부저를 이용하여 도, 레, 미를 출력하는 아두이노 코드이다.
 밑줄 친 곳에 알맞은 코드는?

<pre> /* 피에조 부저를 이용한 소리 출력: Do, Re, Mi */ int swPin1 = 2; int swPin2 = 3; int swPin3 = 4; int buzzerPin = 9; int beats[] = {1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1}; int tempo = 200; void setup() { Serial.begin(9600); pinMode(swPin1, INPUT_PULLUP); pinMode(swPin2, INPUT_PULLUP); pinMode(swPin3, INPUT_PULLUP); pinMode(buzzerPin, OUTPUT); } int frequency(char note){ int i; int notes = 8; char names[] = { 'c', 'd', 'e', 'f', 'g', 'a', 'b', 'C' }; int frequencies[] = {262, 294, 330, 349, 392, 440, 494, 523}; for (i = 0; i < notes; i++){ if (names[i] == note) return([13]_____frequencies[i]); }; }; return(0); } </pre>	<pre> void loop() { int duration; int swVal1 = digitalRead(swPin1); int swVal2 = digitalRead(swPin2); int swVal3 = digitalRead(swPin3); char note = ' '; if(swVal1 == LOW) note = 'c'; if(swVal2 == LOW) note = 'd'; if(swVal3 == LOW) note = 'e'; duration = tempo; if (note == ' ') delay(duration); else{ // tone 명령어를 통하여 부저 핀으로 // 사각파를 출력한다 [14]_____tone(buzzerPin, frequency(note), duration); delay(duration); } delay(tempo / 10); Serial.println(note); } </pre>
--	---

13. frequency 함수가 반환하는 값은? --- (frequency[i])

14. 누른 스위치에 할당된 음을 출력하는 코드는?

- | | |
|--------------------------------------|--|
| A. tone(buzzerPin, frequency(tempo)) | B. tone(buzzerPin, frequency(tempo), duration) |
| C. tone(buzzerPin, frequency(note)) | D. tone(buzzerPin, frequency(note), duration) |

15. 다음은 초음파 거리센서를 이용하여 거리를 측정하는 아두이노 코드이다. 밑줄 친 곳에 알맞은 코드는?

```

/*
  초음파 거리센서를 이용한 거리 측정
*/
// 트리거 핀과 에코 핀 번호를 설정한다.
const char trigPin = 13;
const char echoPin = 12;
// 펄스 폭과 거리 변수 설정
int pulseWidth;
int distance;
int distanceOld;

void setup() {
  // 시리얼 통신 설정
  Serial.begin (9600);
  // 트리거 핀은 출력으로,
  // 에코핀은 입력으로 설정
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  // 트리거 핀의 초기값을 LOW로 한다
  digitalWrite(trigPin, LOW);
}

void loop(){
  // 10us의 트리거 신호를 HC-SR04로 내보낸다.
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Echo 펄스 폭을 측정하여
  // pulseWidth 변수에 저장한다.
  pulseWidth = pulseIn([15]____echoPin, HIGH);
  // 거리를 계산한다.
  distance = pulseWidth / 58;

  // 감지거리인 2~200cm 범위의 거리값만 사용한다.
  if(distance <= 200 || distance >= 2){
    // 이전의 거리값과 비교하여 변화가 있을 경우에만
    // 시리얼 통신으로 전송한다.
    if(distance != distanceOld){
      Serial.print(distance);
      Serial.println(" cm");
    };
  };
  distanceOld = distance;
  delay(100);
}

```

15. Echo 펄스의 폭을 측정하는 설정은?

- A. trigPin, HIGH B. trigPin, LOW
C. echoPin, HIGH D. echoPin, LOW

16[Bonus]. Now, you have a resistor with four color bands; Brown(갈), Black(검), Orange(오), Silver.

What is the resistance value of this resistor?

- A. 330 Ω B. 1 k Ω C. 10 k Ω D. 20 k Ω