
***** 아두이노기초 중간고사. 2018. 10. 18 (목) *****

아두이노 코드를 읽고 쓰면서 하드웨어를 즐기자!

1. What is the incorrect introduction to Arduino?

- A. Arduino is simple and inexpensive.
- B. Arduino programming is simple and clear.
- C. Arduino supports only Unix platform.
- D. Arduino is based on open source.

2. Now, you have a resistor with four color bands;
Brown(갈), Black(검), Red(빨), Silver.
(Hint: 검갈빨오...)

What is the resistance value of this resistor?

- A. 330 Ω B. 1 kΩ C. 10 kΩ D. 20 kΩ

3. PWM (Pulse Width Modulation) is a technique for getting analog results with digital means. By changing pwm_value in analogWrite(pin, pwm_value) function, we can control the voltage of an LED connected to pin-5.
Which one is the correct usage of analogWrite() to keep an LED 50 % bright by supplying 50 % duty cycle?

- A. analogWrite(5, 0.5)
- B. analogWrite(5, 50)
- C. analogWrite(5, 512)
- D. analogWrite(5, 127)

- 4-5. 다음은 delay를 0.2초로 설정후, 5초 마다 number를 초기화하여 시리얼플로터로 톱니파를 발생시키며 시간은 ms로 계산해서 출력하는 아두이노 코드이다.
밑줄친 곳에 알맞은 코드는?

```
int number = 0;

void setup() {
  Serial.begin(9600); // 9600bps로 시리얼 통신 설정
}

void loop() {
  Serial.print(number); // number 변수값 출력
  Serial.println(" msec");
  // " msec"를 출력 후 줄 바꿈
  delay(200); // 0.2초동안 지연시킨다.

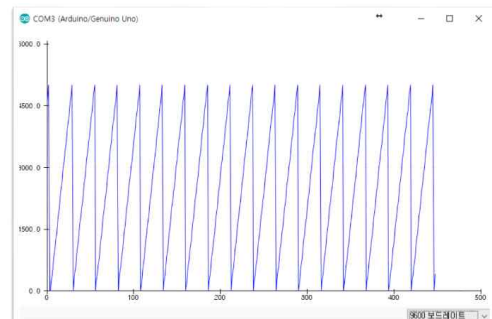
  if (number < 5000) {
    number += [4]_____ 200;
  }else {
    number = [5]_____ 0;
  }
}
```

[4] _____

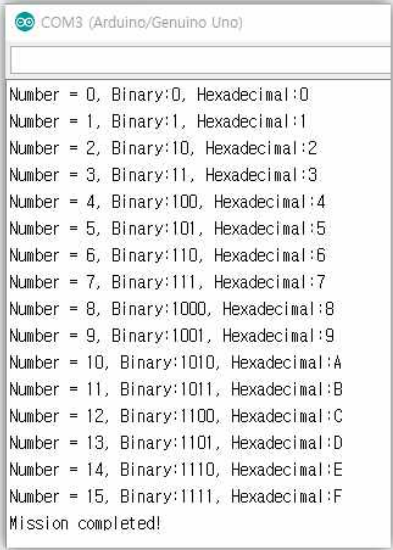
[5] _____

COM3 (Arduino/Genuino Uno)

4600 msec
4800 msec
5000 msec
0 msec
200 msec
400 msec
600 msec
800 msec
1000 msec
1200 msec
1400 msec
1600 msec
1800 msec
2000 msec
2200 msec
2400 msec
2600 msec
2800 msec
3000 msec
3200 msec



6-7. 다음은 0부터 15까지의 숫자를 이진수와 16진수로 출력하고 정지하는 아두이노 코드이다. 밑줄친 곳에 알맞은 숫자는?

<pre>// start number int number = 0; // 문자열 두가지를 설정한다. String stringValue=("Binary:", "Hexadecimal:"); // array void setup() { // 9600bps로 시리얼 통신 설정 Serial.begin(9600); } void loop() { // 'char Value'를 출력하고 문자열과 숫자를 변수 유형별로 출력한다. Serial.print("Number = "); Serial.print(number); Serial.print(", "); Serial.print(stringValue[0]); // stringValue 중 첫 번째 문자열 출력 Serial.print(number, BIN); // 2진수 형태로 출력 Serial.print(", "); Serial.print(stringValue[1]); // stringValue 중 첫 번째 문자열 출력 Serial.print(number, HEX); // 16진수 형태로 출력 // 줄바꿈 Serial.println(); number++; // number 1 증가 if(number > [6]_____ 15) { Serial.print("Mission completed!"); delay(1000); exit([7]_____ 0); } delay(1000); // 1초동안 지연시킨다. }</pre>	 <pre>COM3 (Arduino/Genuino Uno) Number = 0, Binary:0, Hexadecimal:0 Number = 1, Binary:1, Hexadecimal:1 Number = 2, Binary:10, Hexadecimal:2 Number = 3, Binary:11, Hexadecimal:3 Number = 4, Binary:100, Hexadecimal:4 Number = 5, Binary:101, Hexadecimal:5 Number = 6, Binary:110, Hexadecimal:6 Number = 7, Binary:111, Hexadecimal:7 Number = 8, Binary:1000, Hexadecimal:8 Number = 9, Binary:1001, Hexadecimal:9 Number = 10, Binary:1010, Hexadecimal:A Number = 11, Binary:1011, Hexadecimal:B Number = 12, Binary:1100, Hexadecimal:C Number = 13, Binary:1101, Hexadecimal:D Number = 14, Binary:1110, Hexadecimal:E Number = 15, Binary:1111, Hexadecimal:F Mission completed!</pre>
--	---

6. 16진수를 출력한 후 종료하기 위해 여기에 들어갈 수는? --- ()

7. loop()를 정상 종료하기 위하여 여기에 들어갈 수는? --- ()

- 8-9. 다음은 시리얼 통신으로 입력 받은 1~9의 숫자에 대하여 LCD의 백라이트가 입력된 숫자만큼 점멸하고 점멸 횟수를 표시하는 아두이노 코드이다.
 밑줄친 곳에 알맞은 코드는?

<pre>// I2C 통신 라이브러리 설정 #include <Wire.h> // I2C LCD 라이브러리 설정 #include <LiquidCrystal_I2C.h> // LCD I2C address 설정 PCF8574A:0x3F LiquidCrystal_I2C lcd(0x3F,16,2); int blinkNumber = 0; void setup() { // 9600 bps로 시리얼 통신 설정 Serial.begin(9600); lcd.init(); // LCD 설정 lcd.clear(); // LCD를 모두 지운다. lcd.backlight(); // 백라이트를 켜다. // Arduino LCD, Welcome 표시 lcd.setCursor(0,0); lcd.print("Arduino LCD"); delay(3000); lcd.setCursor(0,1); lcd.print("welcome"); delay(250); // Open Serial Monitor, Type to display 표시 lcd.clear(); lcd.setCursor(0,0); lcd.print("Open Serial Mntr"); lcd.setCursor(0,1); lcd.print("Type to display"); }</pre>	<pre>void loop() { // 시리얼 통신 수신 값이 있을 때 if (Serial.available()) { char val= Serial.read(); if(isDigit(val)){ blinkNumber = (val - [8]_____'0'); // blinkNumber에는 실제 숫자가 저장된다. } delay(100); // 모두 삭제 lcd.clear(); // 커서를 좌측 상단으로 lcd.setCursor(0,0); // "Message from PC" 출력 lcd.print("Message from PC"); // 커서를 두 번째 줄로 lcd.setCursor([9]_____, 0,1); lcd.print(blinkNumber); for(char i=0;i < blinkNumber;i++){ lcd.noBacklight(); delay(250); lcd.backlight(); delay(250); } // LCD에 PC에서 전송된 데이터를 출력 while (Serial.available()>0) { lcd.write(Serial.read()); } } }</pre>
--	--

8. 직렬 통신으로 전송된 문자로 된 숫자를
 실제 숫자로 변환하기 위한 코드는? --- ()
9. 커서를 LCD의 아랫 줄 처음으로 보내는 코드는? --- ()

10-12. 다음은 4개의 단색 LED를 순서대로 디밍(dimming)시키는 아두이노 코드이다.
밑줄친 곳에 알맞은 내용은?

```
int ledR = 3;    // LED connected to digital pin 3
int ledG = 5;
int ledB = 6;
int ledY = [10]____; // 가능한 PWM pin number?

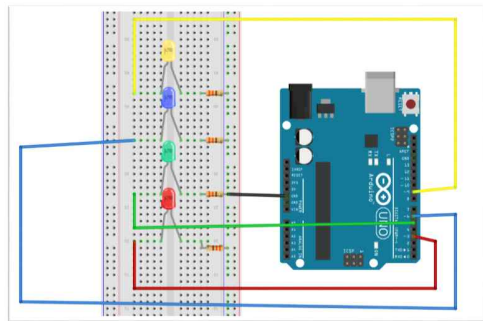
int dimTime = 20;

void setup() {
  // nothing happens in setup
}

void loop() {
  dimLed(ledR); // fade ledR
  dimLed(ledG); // fade ledG
  dimLed(ledB); // fade ledB
  dimLed(ledY); // fade ledY
}

void dimLed(int led) {
  // fade in from min to max in increments of 5
  points:
  for(int fadeValue = 0 ; fadeValue <= 255;
  fadeValue +=5) {
    // sets the value (range from 0 to 255):
    analogWrite([11]____ led , fadeValue);
    // wait for 20 milliseconds to see dimming
    delay(dimTime);
  }

  // fade out from max to min in increments of 5
  // points:
  for(int fadeValue = [12]____ 255 ; fadeValue >=
  0; fadeValue -=5) {
    analogWrite([11]____ led , fadeValue);
    // wait for 20 milliseconds to see dimming
    delay(dimTime);
  }
}
```



[10] _____

[11] _____

[12] _____

13-14. 다음은 8개의 LED로 구성된 FND에 0에서 9까지의 수를 1초에 하나씩
 순서대로 출력하는 아두이노 코드이다.
 밑줄친 곳에 알맞은 내용은?

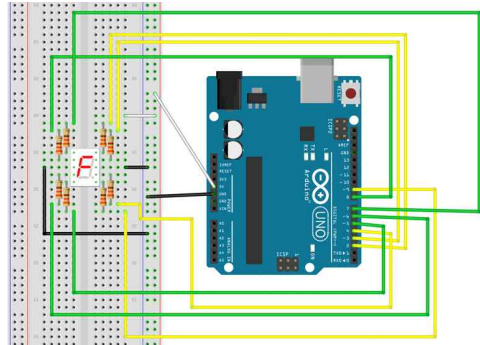
```
// 0~9까지 LED 표시를 위한 상수 설정
const byte number[10] = {
//dot g f e d c b a
  B00111111,    //0
  B00000110,    //1
  B01011011,    //2
  B01001111,    //3
  B01100110,    //4
  B01101101,    //5
  B01111101,    //6
  B00000111,    //7
  B01111111,    //8
  B01101111,    //9
};

void setup()
{ // 2~9번 핀을 a b c d e f g dot 의 순서로 사용한다
  // 2~9번핀을 출력으로 초기화 시킨다.
  for(int i = 2; i <= 9; ++i){
    pinMode(i,OUTPUT);
  };
  digitalWrite(9,LOW); // 점은 표시하지 않는다
}

void loop()
{ // k값을 0~9로 변화시킨다.
  for(int k = 0; k <= 9; ++k){
    fndDisplay(k); // k값을 출력한다
    delay(1000);
  };
}

// LED 점등
void fndDisplay(int displayValue){
  // bitValue 변수를 선언한다.
  boolean bitValue;

  for(int i=2; i<=9; ++i){
    // 2~9번핀에 모두 소등시킨다
    digitalWrite(i, [13]_____ LOW);
  };
  for(int i=0; i<=7; ++i){
    // number 상수의 하나의 비트값을 읽는다
    bitValue = bitRead(number[displayValue], i);
    // 앞서 읽은 비트값을 2~9번핀에 출력시킨다
    digitalWrite([14]_____ i+2, bitValue);
  };
}
```



[13] _____

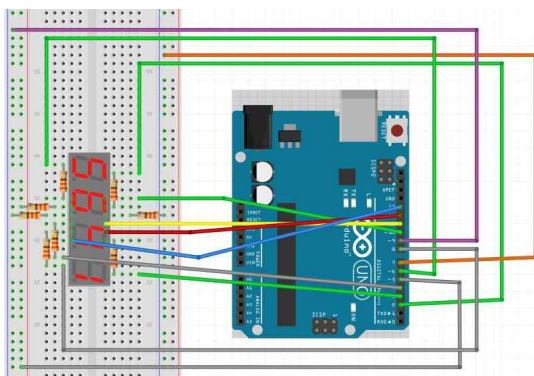
[14] _____

15-17. 다음은 4-digit FND에 'XXX1', 'XX2X', 'X3XX', '4XXX' 의 표시가 1초 간격으로 반복하는 아두이노 코드이다. (X:는 꺼짐을 나타낸다)
 밑줄친 곳에 알맞은 내용은?

```
// 0~9까지 LED 표시를 위한 상수
const byte number[10] = {
    //dot gfedcba
    B00111111,    //0
    B00000110,    //1
    B01011011,    //2
    B01001111,    //3
    B01100110,    //4
    B01101101,    //5
    B01111101,    //6
    B00000111,    //7
    B01111111,    //8
    B01101111,    //9
};
// 표시할 숫자 변수
int count = 1;
int fnd_index = 13; // FND4 -> 숫자 1부터 시작

void setup()
{
    // 2~9번 핀을 a b c d e f g dot 의 순서로 사용한다.
    // 10~13번 핀을 Digit 1~4 의 순서로 사용한다.
    for(int i = 2; i <= 13; ++i){
        pinMode(i,OUTPUT); //2~13번핀을 출력으로 설정한다.
    };

    // 4 digit와 연결된 10~13번핀에 모두 소등시킨다
    for(int i=10; i<=13; ++i){
        digitalWrite(i, [15]_____ HIGH);
    };
}
```



```
void loop()
{
    // count 변수값을 FND에 출력한다.
    // 우측 FND에 1을 출력
    digitalWrite(fnd_index, LOW);
    fndDisplay(count);
    delay(1000);
    // 출력 후 FND 소등
    digitalWrite(fnd_index, HIGH);
    // FND를 변경하며 2, 3, 4 출력
    if(count >= 4){
        count = 1;
        fnd_index = [16]_____ 13;
    }
    else {
        count++;
        [17]_____ fnd_index--;
    }

    delay(100);
}

// LED 켜는 루틴
void fndDisplay(int displayValue){
    // bitValue 변수를 선언한다.
    boolean bitValue;

    // 2~9번핀에 모두 LOW 신호를 줘서 소등시킨다.
    for(int i=2; i<=9; ++i){
        digitalWrite(i, LOW);
    };

    for(int i=0; i<=7; ++i){
        // number 상수의 하나의 비트값을 읽는다.
        bitValue = bitRead(number[displayValue],i);
        // 앞서 읽은 비트값을 2~9번핀에 출력시킨다.
        digitalWrite(i+2, bitValue);
    };
}
```

18-20. 다음은 8X8 dot Matrix 하나에 두 개의 하트 이미지를 번갈아 그리는 아두이노 코드이다. 밑줄친 곳에 알맞은 내용은?

```
#include "LedControlMS.h"

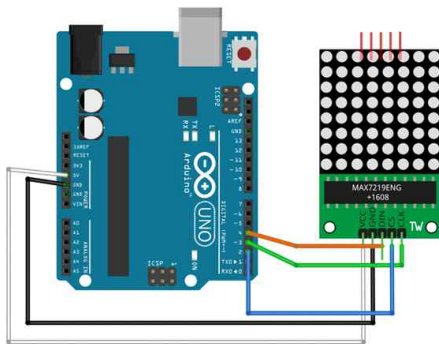
#define NBR_MTX [18]_____1

LedControl lc=LedControl(4,3,2, NBR_MTX);//

unsigned long delayTime=1000;
// Delay between Frames

// Put values in arrays
byte heart1a[] =
{
    B00100100, // First frame of heart1a
    B01011010,
    B10000001,
    B10000001,
    B10000001,
    B01000010,
    B00100100,
    B00011000
};

byte heart1b[] =
{
    B00000000, // Second frame of heart1b
    B01100110,
    B11111111,
    B11111111,
    B11111111,
    B01111110,
    B00111100,
    B00011000
};
```



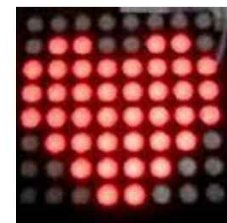
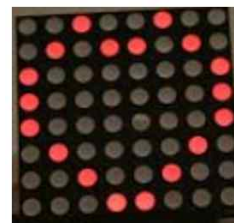
```
void setup()
{
    lc.shutdown(0, [19]_____false); //Wake up displays
    lc.setIntensity(0,5); // Set intensity levels
    lc.clearDisplay(0); // Clear Displays
}

// Take values in Arrays and Display them
void heart1()
{
    for (int i = 0; i < 8; i++)
    {
        lc.[20]_____setRow(0, i, heart1a[i]);
    }
}

void heart2()
{
    for (int i = 0; i < 8; i++)
    {
        lc.[20]_____setRow(0, i, heart1b[i]);
    }
}

void loop()
{
    // Put #1 frame on the first Display
    heart1();
    delay(delayTime);

    // Put #2 frame on the first Display
    heart2();
    delay(delayTime);
}
```



[20.보기] setPxI, setLed, setDot, setCol, setRow