

CURSO: ROBÓTICA. CONECTANDO CON EL MUNDO FÍSICO

1 Nombre, apellidos y correo electrónico de EducaMadrid del participante

ÍÑIGO MARTÍN MIMBELA.
inigo.martin@educa.madrid.org

2 Título de la Unidad Didáctica

CONTROL PWM DE LED Y MOTOR CON ENTRADA ANALÓGICA DE SENSOR.

3 Curso

3º ESO

4 Descripción del proyecto

Es un proyecto de diseño e interacción en el que se muestra el control PWM (“pulse width modulation”, modulación por ancho de pulsos) de una señal, para controlar la cantidad de energía que se envía a una carga.

Lo aplico secuencialmente, en dos programas del mismo proyecto: primero a un LED para poder variar su intensidad de brillo, y después a un motor de corriente continua para variar la velocidad del mismo según el valor que adquiere el sensor LDR (resistencia dependiente de la luz). La LDR se utiliza como entrada analógica de control, ya que presentará un valor bajo de resistencia ante la presencia de luz, y un valor alto de resistencia en ausencia de luz.

El proyecto se programa en el IDE de Arduino y se controla mediante la tarjeta o placa de control de Arduino Uno. El programa para controlar el motor, cuenta con un bloque para calibrar los valores máximos y mínimos que presenta el sensor (la LDR), y estructuras de control para el manejo de ciclos For y condicionales If/else, además de las variables y funciones digitales y analógicas necesarias. Asimismo el programa ofrece la Lectura Serial del sensor.

5 Competencias que se trabajarán

1. Entender el concepto de modulación PWM.
2. Encender un LED de forma proporcional.
3. Conectar un motor por medio de un transistor.
4. Variar la velocidad de un motor de corriente continua.
5. Escribir entradas y salidas analógicas.
6. Realizar un programa con: ciclos For, arrays o matrices para bucles y, también, con controles de flujo condicionales tipo If/else.
7. Manejar la función map()
8. Controlar y leer una entrada analógica de un sensor (LDR)
9. Configurar e interpretar lecturas de comunicación serial.

6 Objetivos

1. Diseñar y montar circuitos electrónicos de un proyecto tecnológico.
2. Programar los circuitos electrónicos del proyecto.
3. Documentar el prototipo desarrollado.
4. Analizar distintos tipos de circuitos identificando la mejor forma de controlarlos.
5. Desarrollar la innovación y creatividad en la búsqueda de soluciones tecnológicas.

7 Contenidos

1. PWM.
2. Funciones: `digitalWrite()`, `analogRead()`, `analogWrite()`, `Serial.print()`
3. Magnitudes eléctricas básicas. Ley de Ohm. Divisor de voltaje.
4. Máquinas eléctricas básicas: dinamo, motor de corriente continua.
5. Componente electrónicos: LED, LDR, Transistor.
6. Introducción a las máquinas automáticas y robots: automatismos.
7. Sistemas de control.

8 Criterios de evaluación

1. Utiliza adecuadamente los dispositivos electrónicos como fuente de información y para crear contenidos.
2. Representa mediante diagramas de flujo diferentes algoritmos
3. Utiliza con destreza y precisión el IDE de Arduino, con los comandos de control de ejecución: condicionales y bucles.
4. Emplea de manera adecuada variables y listas.
5. Analiza el funcionamiento de un programa a partir de sus instrucciones.
6. Identifica y emplea las entradas y salidas analógicas o digitales de un sistema electrónico.
7. Identifica las posibilidades de interacción con los sensores, y su función como conversores a magnitudes eléctricas de otras variables.
8. Señala las características básicas y la aplicación de componentes, como por ejemplo: LDR, LED, Transistor.
9. Identifica las características básicas de los motores.
10. Realiza el montaje de circuitos electrónicos de acuerdo al diseño de un esquema.

PRIMER PROGRAMA: VARIAR LA INTENSIDAD DE BRILLO DE UN LED.

```
/*  
  Control PWM para variar la intensidad de brillo de un LED.  
*/
```

```
int ledPin = 9; // pin PWM para el LED  
void setup(){} // no es necesario configurar nada  
void loop()  
{  
  for (int i=0; i<=255; i++) // el valor de i asciende  
  {  
    analogWrite(ledPin, i); // se escribe el valor de i en  
    // el PIN de salida del LED  
    delay(100); // pausa durante 100ms  
  }  
  for (int i=255; i>=0; i--) // el valor de i desciende  
  {  
    analogWrite(ledPin, i); // se escribe el valor de i  
    delay(100); // pausa durante 100ms  
  }  
}
```

SEGUNDO PROGRAMA: VARIAR LA VELOCIDAD DE UN MOTOR SEGÚN EL VALOR QUE ADQUIERE EL SENSOR LDR.

```
/*  
  Control PWM para variar la velocidad de un motor de  
  corriente continua, según el valor que adquiere un sensor LDR.  
  La parte de Calibración está tomada del Tutorial de Arduino.  
*/
```

```
//estas constantes no cambiarán  
const int sensorPin = A0; //pin del sensor  
const int motorPin = 3; //pin PWM del motor  
//variables  
int sensorValue = 0; //valor del sensor
```

```

int sensorMin = 1023; //valor mínimo sensor
int sensorMax = 0; //valor máximo sensor

void setup(){

  Serial.begin(9600);
  pinMode(3, OUTPUT);
  pinMode(13, OUTPUT);
  //encender LED para señalar el comienzo de la calibración
  digitalWrite(13, HIGH);

  //calibrar durante los primeros cinco segundos
  while(millis())<5000){
  sensorValue = analogRead(sensorPin);

  //registrar el valor máximo del sensor
  if(sensorValue>sensorMax){
    sensorMax=sensorValue;
  }

  //registrar el valor mínimo del sensor
  if(sensorValue<sensorMin){
    sensorMin=sensorValue;
  }
  }

  //señalar el final de la calibración
  digitalWrite(13, LOW);
}

void loop(){

  // leer el sensor

```

```
sensorValue = analogRead(sensorPin);

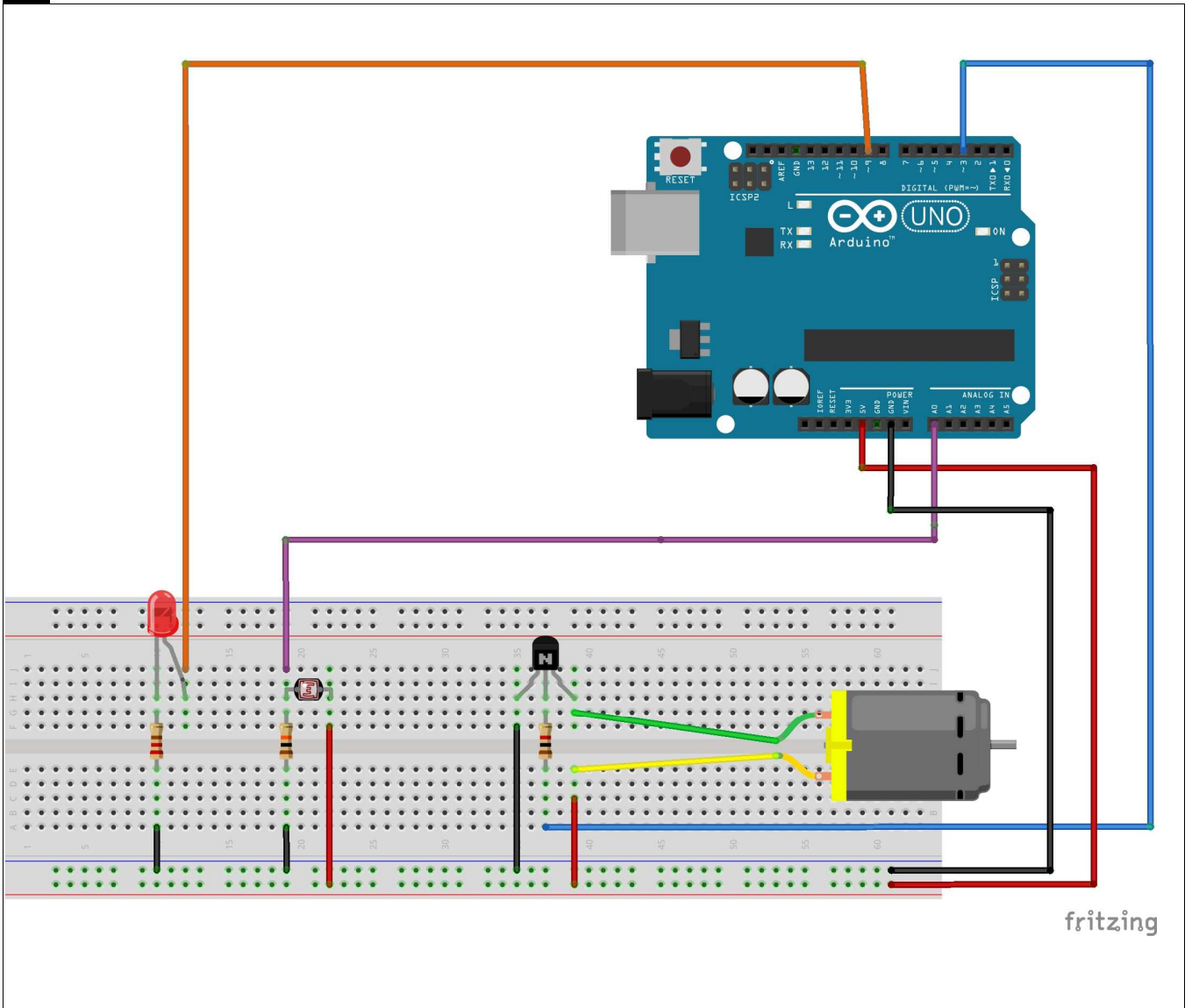
//aplicar la calibración para la lectura del sensor
sensorValue = map(sensorValue, sensorMin, sensorMax, 0, 255);

//en caso de que el valor del sensor esté fuera del rango
//visto durante la calibración
sensorValue = constrain(sensorValue, 0, 255);

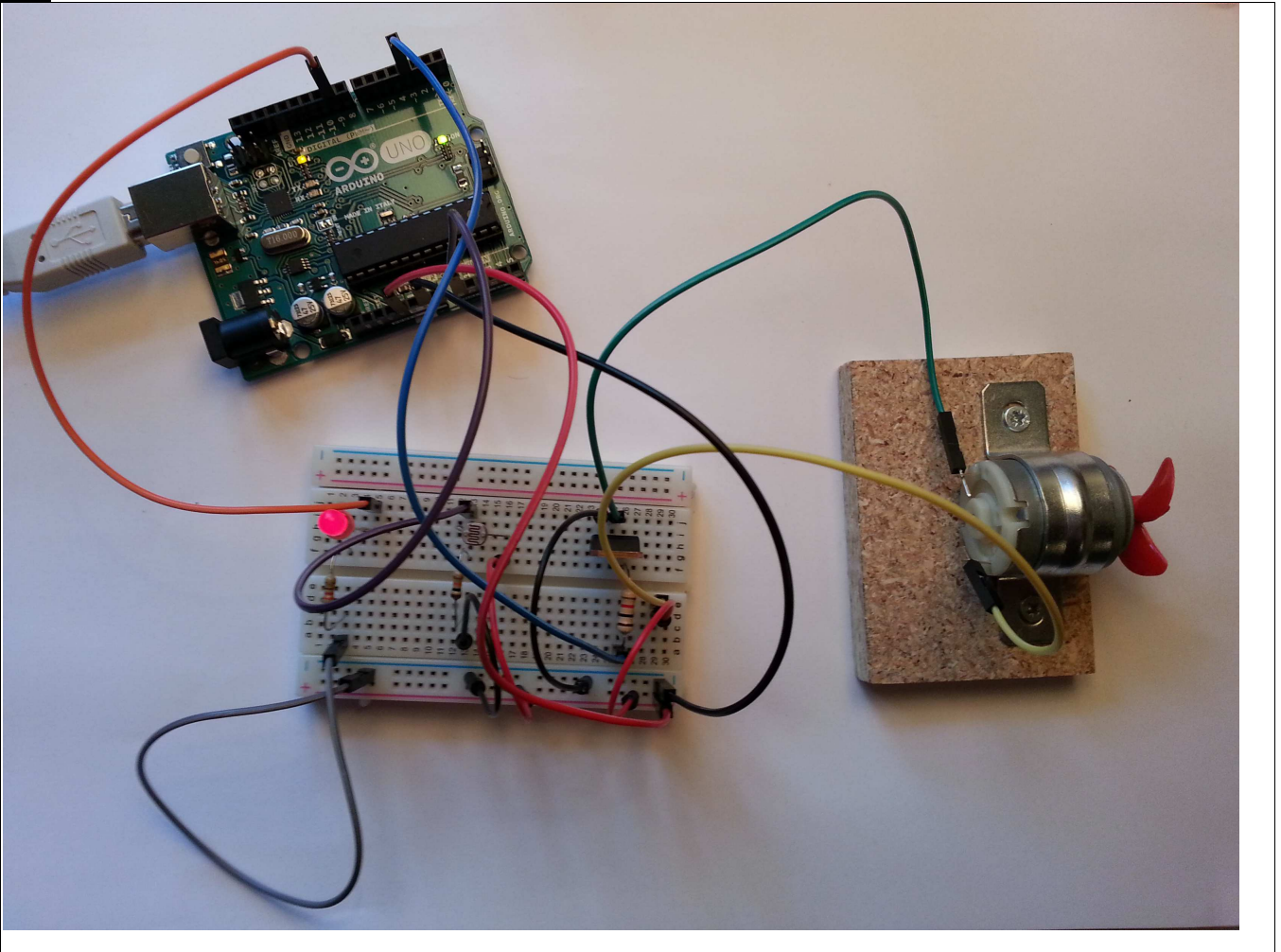
//variación del motor usando el valor de calibración
analogWrite(motorPin, sensorValue);

//mostrar por el puerto serie el valor del sensor
Serial.print("sensor= ");
Serial.println(sensorValue);
delay(100);
}
```

10 Esquema de conexionado



11 Fotografía del montaje



X

Marque con una cruz si acepta que esta unidad didáctica quede recogida bajo una licencia Creative Commons (by – nc – sa) (Reconocimiento, No Comercial, Compartir Igual) para ser compartida por todos los miembros de la comunidad de docentes:
<http://es.creativecommons.org/blog/licencias/>