

# CURSO: ROBÓTICA. CONECTANDO CON EL MUNDO FÍSICO

## 1 Nombre, apellidos y correo electrónico de EducaMadrid del participante

Elena Ribón Villamediana <eribonvillamediana@educa.madrid.org>

José Antonio Cobo Arroyo <jcoboarroyo@educa.madrid.org>

## 2 Título de la Unidad Didáctica

Automatización de un ascensor.

## 3 Curso

3º ESO – 4º ESO

## 4 Descripción del proyecto

Se pretende automatizar el funcionamiento de un ascensor con dos plantas. Proyecto que se puede realizar con madera en el taller de tecnología.

Tenemos dos pulsadores de llamada y dos leds amarillos (1 por planta respectivamente). Al pulsar el botón de una planta, se encenderá el led del Arduino (pin 13), y el ascensor deberá ir a esa planta, salvo que ya se encuentre en esa misma planta, que no hará nada.

Para simular el movimiento del ascensor se utilizarán dos motores de C.C. y dos leds de control, uno verde para indicar que el ascensor está subiendo y uno rojo para indicar que está bajando. Se utilizarán dos pulsadores, imitando dos finales de carrera, para detener el ascensor en cada planta. Cuando esto suceda, ha de encenderse el led amarillo de la planta correspondiente y apagarse los demás.

Como mejoras del presente circuito, se puede introducir un circuito con inversor de giro y utilizar un solo motor. También se puede añadir un potenciómetro para regular la velocidad de giro del motor. Adicionalmente se puede conectar el puerto serie del ordenador, o bien para controlar los motores con el teclado, o para realizar lecturas de las variables como velocidad del motor, estado de los leds y pulsadores, número de planta del ascensor, etc.

Este circuito también se puede aplicar al control de dos motores, con sentidos de giro opuestos, y controlados por dos pulsadores cada uno, funcionamiento y parada.

## 5 Competencias que se trabajarán

Se trabajarán las siguientes competencias básicas:

1. Competencia en comunicación lingüística.
3. Competencia en el conocimiento y la interacción con el mundo físico.
4. Tratamiento de la información y competencia digital.
7. Competencia para aprender a aprender.
8. Autonomía e iniciativa personal.

## 6 Objetivos

1. Montar y conectar correctamente un led a la controladora Arduino Uno.
2. Montar y conectar correctamente un pulsador a Arduino.
3. Montar y conectar correctamente un motor a Arduino.
4. Aplicar un transistor bipolar npn en "modo interruptor" y conectarlo correctamente a Arduino.
5. Orden y limpieza en el montaje de un circuito en placa protoboard y controladora Arduino.
6. Programar correctamente la automatización de un ascensor de dos plantas controlado por un motor, dos pulsadores,
7. Documentar el código de manera limpia y clara para reutilización por terceras personas.

## 7 Contenidos

- Placa controladora Arduino Uno, morfología y conexiones.
- Led, aplicaciones y modos de conexión a Arduino.
- Pulsadores, aplicaciones y modos de conexión a Arduino.
- Motores de CC, aplicaciones y modos de conexión a Arduino.
- Transistores bipolares, aplicaciones y modos de funcionamiento.
- Programación en C++. Instrucciones básicas: pinMode(), digitalWrite(), digitalRead(), do(), while(), if(), else(), switch(), operadores (=, !=, ==, >=, <=).
- Documentación de un programa. Sintaxis y normas de estilo.

## 8 Criterios de evaluación

1. Monta y conecta correctamente un led a controladora Arduino.
2. Monta y conecta correctamente un pulsador a Arduino.
3. Monta y conecta correctamente un motor a Arduino.
4. Monta y conecta correctamente un motor de CC mediante un transistor npn a Arduino.
5. Realiza el montaje del circuito con orden y claridad.
6. Programa correctamente la automatización de un ascensor, de manera que compila y el sistema funciona según lo previsto.
7. Documenta el código de manera limpia y clara.

## 9 Código del programa de Arduino

```
// Establecemos los números de pin:

const int motorsube = 3; // número de pin de activación del motor en sentido de subida.
const int motorbaja = 4; // número de pin de activación del motor en sentido de bajada.

const int ledP1 = 5; // número de pin del led de la primera planta.
const int ledP2 = 6; // número de pin del led de la segunda planta.

const int pinbotonP1 = 7; // número de pin del botón de la primera planta.
const int pinbotonP2 = 8; // número de pin del botón de la segunda planta.

const int pinFDC1 = 9; // número de pin del final de carrera (FDC) de la primera planta.
const int pinFDC2 = 10; // número de pin del final de carrera (FDC) de la segunda planta.

const int ledverde = 11; // número de pin del led verde.
const int ledrojo = 12; // número de pin del led rojo.
const int ledplaca = 13; // número de pin del led de la placa.

// Establecemos las variables:

int botonP1 = 0; // variable para leer el estado del botón de la planta 1.
int b1 = 0; // Contador para saber si se ha pulsado el botón de la planta 1.
int botonP2 = 0; // variable para leer el estado del botón de la planta 2.
int b2 = 0; // Contador para saber si se ha pulsado el botón de la planta 2.
int fdc1 = 0; // variable para leer el estado del final de carrera de la planta 1.
int fdc2 = 0; // variable para leer el estado del final de carrera de la planta 2.
int planta = 2; // variable para la lectura de la planta del ascensor.

// Inicializamos los pines
void setup() {

  // inicializamos pin digital 3 cómo una salida.
  pinMode(motorsube, OUTPUT);
  // inicializamos pin digital 4 cómo una salida.
  pinMode(motorbaja, OUTPUT);

  // inicializamos pin digital 5 cómo una salida.
  pinMode(ledP1, OUTPUT);
  // inicializamos pin digital 6 cómo una salida.
  pinMode(ledP2, OUTPUT);

  // inicializamos pin digital 7 cómo una entrada.
  pinMode(pinbotonP1, INPUT);
  // inicializamos pin digital 8 cómo una entrada.
  pinMode(pinbotonP2, INPUT);

  // inicializamos pin digital 9 cómo una entrada.
  pinMode(pinFDC1, INPUT);
  // inicializamos pin digital 10 cómo una entrada.
  pinMode(pinFDC2, INPUT);

  // inicializamos pin digital 11 cómo una salida.
  pinMode(ledverde, OUTPUT);
  // inicializamos pin digital 12 cómo una salida.
  pinMode(ledrojo, OUTPUT);
  // inicializamos pin digital 13 cómo una salida.
  pinMode(ledplaca, OUTPUT);
}
```

```

// the loop function runs over and over again forever
void loop() {

// Lectura de llamada de los botones de planta b1 - planta1, b2 - planta2.

if (b1 == 0) {
  botonP1 = digitalRead(pinbotonP1);
  if (botonP1 == HIGH) {
    digitalWrite(ledplaca, HIGH); // Enciende el led de llamada de planta.
    b1 = 1;
  }
}

if (b2 == 0) {
  botonP2 = digitalRead(pinbotonP2);
  if (botonP2 == HIGH) {
    digitalWrite(ledplaca, HIGH); // Enciende el led de llamada de planta.
    b2 = 1;
  }
}

// Comprueba si el botón de la planta 1 está pulsado.
// Si lo está, comprueba si el ascensor está en la planta 2:

if (b1 == 1) {

  if (planta == 2) {
    while(fdc1 != HIGH){
      digitalWrite(motorsube, LOW);
      digitalWrite(motorbaja, HIGH); // el ascensor baja
      digitalWrite(ledrojo, HIGH); // Enciende el led rojo de bajada del ascensor.
      fdc1 = digitalRead(pinFDC1); // Lectura del estado del el Final de Carrera de la planta 1.
    }
    digitalWrite(ledrojo, LOW);
    digitalWrite(motorbaja, LOW); // Para el motor de bajada.
    digitalWrite(ledplaca, LOW); // Apaga el led de llamada.
    planta = 1;
    b1 = 0;
    fdc1 = 0;
  }
}

// Comprueba si el botón de la planta 2 está pulsado.
// Si lo está, comprueba si el ascensor está en la planta 1:

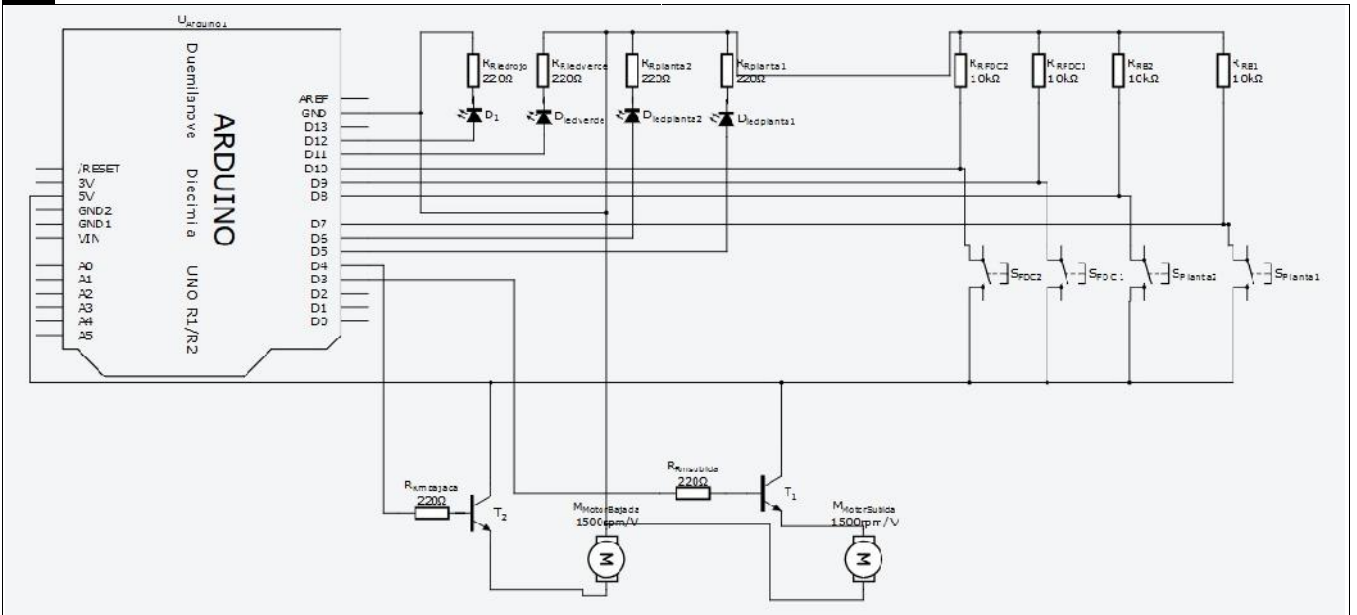
if (b2 == 1) {

  if (planta == 1) {
    while(fdc2 != HIGH){
      digitalWrite(motorsube, HIGH); // el ascensor sube.
      digitalWrite(motorbaja, LOW);
      digitalWrite(ledverde, HIGH); // Enciende el led verde de subida del ascensor.
      fdc2 = digitalRead(pinFDC2); // Lectura del estado del el Final de Carrera de la planta 2.
    }
    digitalWrite(ledverde, LOW);
    digitalWrite(motorsube, LOW); // Para el motor de bajada.
    digitalWrite(ledplaca, LOW); // Apaga el led de llamada.
    planta = 2;
    b2 = 0;
    fdc2 = 0;
  }
}

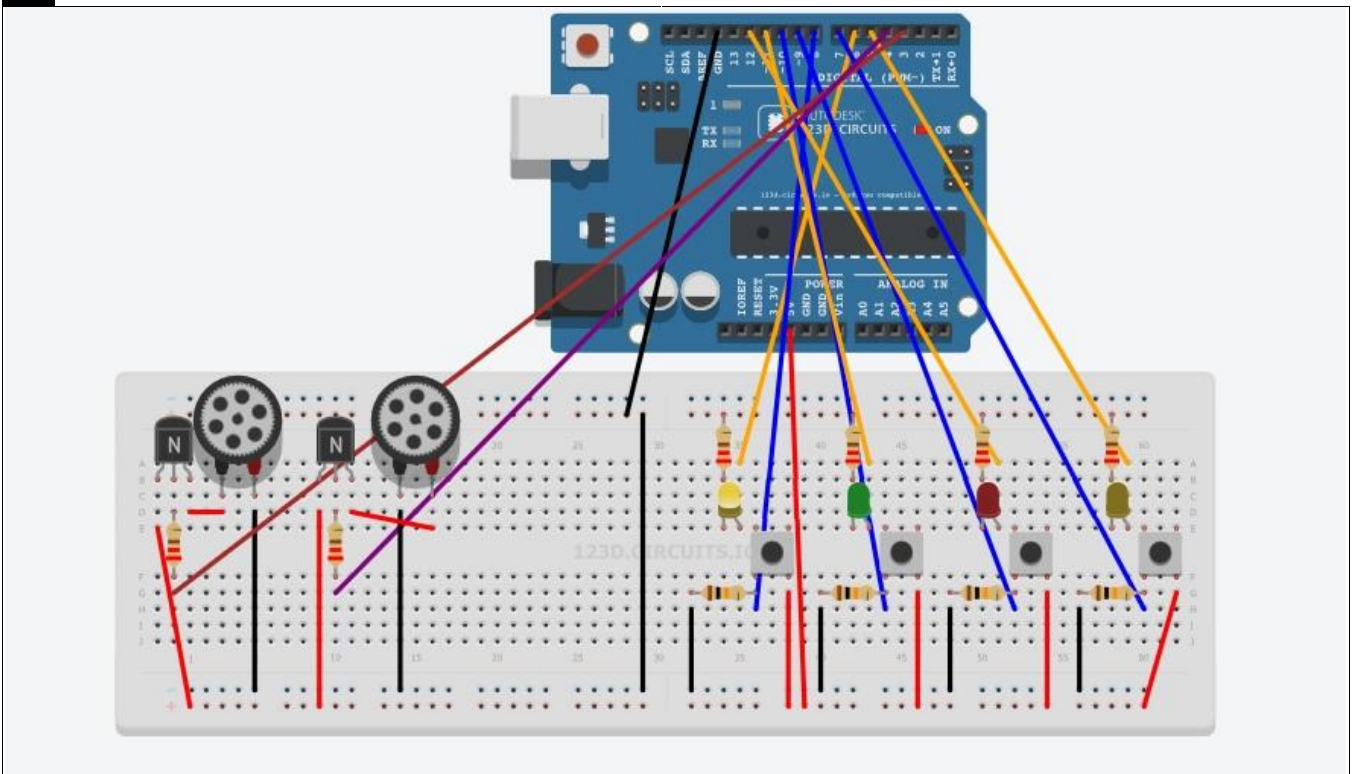
// Código para encender y/o apagar los leds de las plantas:
switch (planta) {
case 1:
  // enciende el led de la planta uno, apaga los leds de las otras plantas:
  digitalWrite(ledP1, HIGH);
  digitalWrite(ledP2, LOW);
  break;
case 2:
  // enciende el led de la planta dos, apaga los leds de las otras plantas:
  digitalWrite(ledP1, LOW);
  digitalWrite(ledP2, HIGH);
  break;
}
}

```

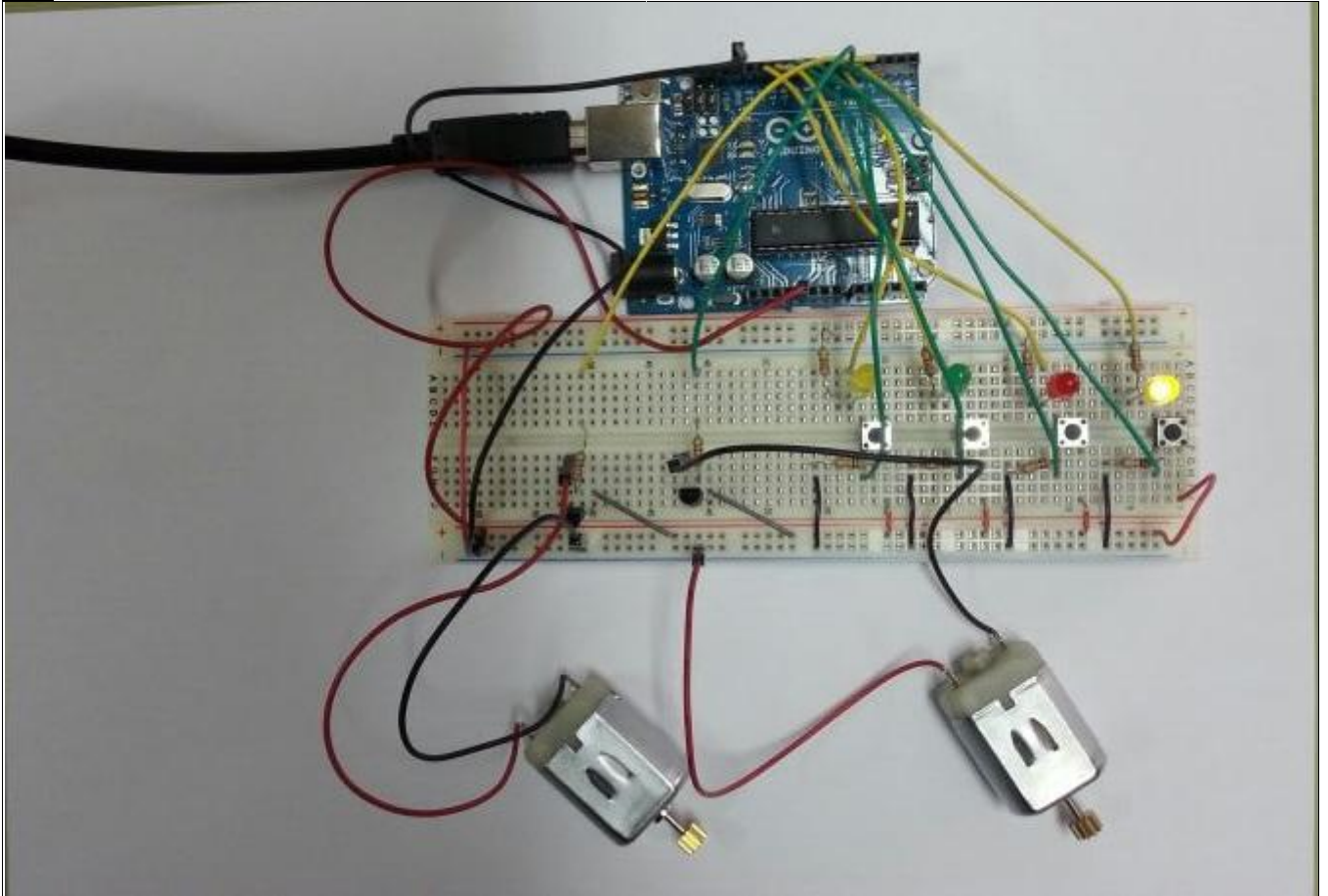
## 10 Esquema de conexionado



## 11 Simulación del montaje



## 12 Foto 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/>