

CURSO DE INICIACIÓN A LA ROBÓTICA

TAREA FINAL

Celso Molina Ibáñez

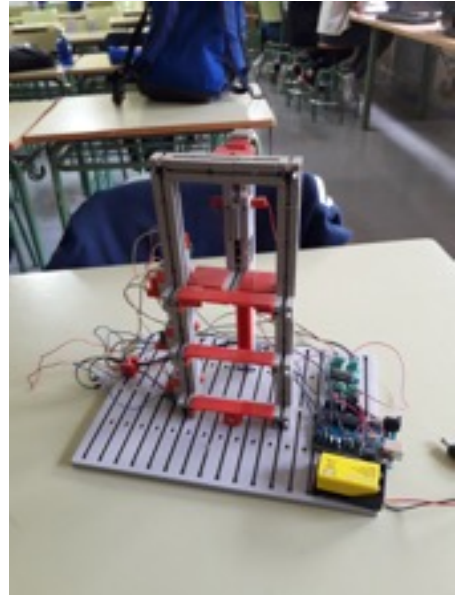
Introducción.

El proyecto que se presenta corresponde a la automatización de un ascensor de tres plantas. La construcción se llevó a cabo con piezas de Fischertechnik de las que se disponía en el centro desde los primeros tiempos de la implantación de la Tecnología en la enseñanza, allá por los años 90.

Esta misma construcción se automatizaba anteriormente por medio de una tarjeta Inves conectada al puerto paralelo y el programa MSWinlogo; sistema que hace tiempo se abandonó por obsoleto y por mal funcionamiento de las tarjetas. Ahora, ésta y otras construcciones, se aprovechan para las demostraciones prácticas con Arduino.

Estructura y Electrónica.

En la imagen de la derecha puede verse el conjunto donde se aprecia la estructura, la pila de 9V para alimentación, la placa Arduino Uno y un driver basado en el L293 para poder hacer funcionar el motor con mayor corriente que la que proporciona la placa de Arduino. El movimiento de la plataforma se lleva a cabo por un motor de corriente continua que se desplaza sobre una cremallera.



En cada planta se sitúa un pulsador, para llamar al ascensor, y un sensor de posición fin de carrera. Estructuralmente son similares, pero la función es distinta. En las dos imágenes de la izquierda pueden apreciarse; primero el que hace la función de pulsador y, después, el que actúa de fin de carrera en la posición de activado por el paso de la superficie que se desplaza haciendo las veces de cabina de ascensor.

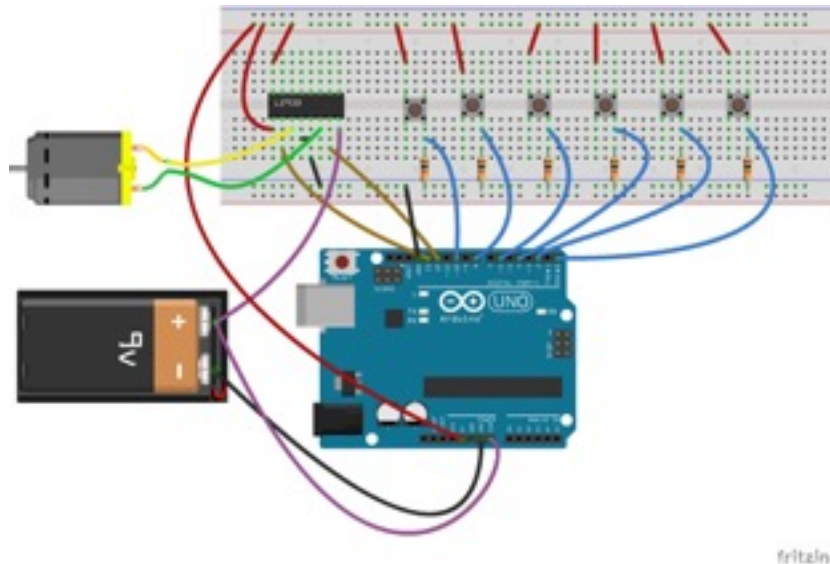


También en la imagen del pulsador, se observan las conexiones para 5v y GND de forma que una de ellas será transmitida al pin de entrada correspondiente según pulsemos o no.

En el esquema eléctrico realizado con Fritzing, se muestra la conexión en pull down para enviar la señal a Arduino. Si se pulsa, el pin de entrada correspondiente queda conectado a 5V (estado alto) y, si por el contrario, el pulsador queda en reposo, queda a tierra. La resistencia evita el cortocircuito

cuando se cierra el circuito.

Por otra parte, el driver L293 dispone de cuatro entradas y cuatro salidas, de las que solamente se emplean dos. Conectando el pin 1 a estado alto, habilitamos las salidas 1 (pin 3) y 2 (pin 6); su acción se

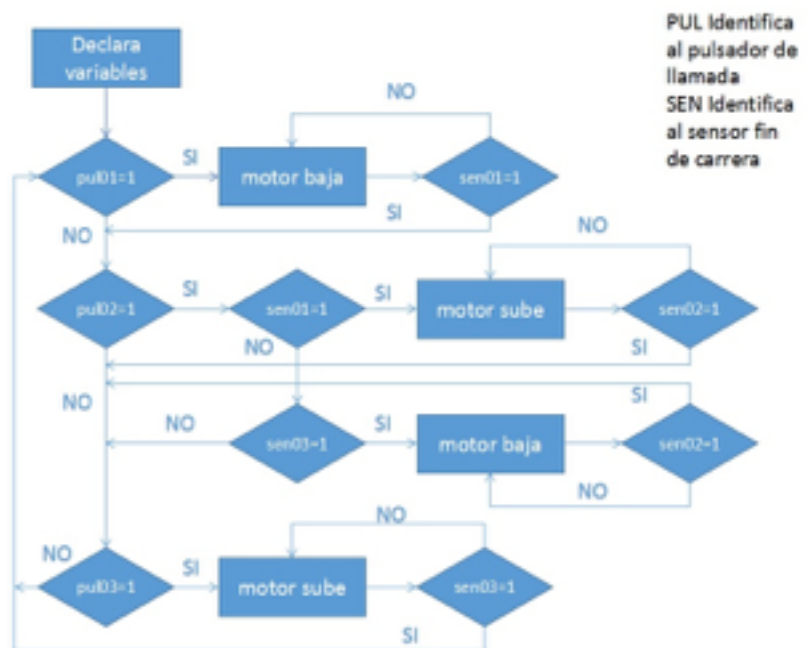


corresponde con las entradas 1 (pin 2) y 2 (pin 7). Si éstas están en estado alto, el driver coloca en la salida la tensión que dispongamos para el motor (pin 8); si no, dará una salida de 0v.

Por último indicar que, pese a que la alimentación de la tarjeta parece en el esquema mostrado como pinchada en el pin Vin, realmente se hace a través del jack (positivo en el centro).

Programa de control.-

El diagrama de flujo se muestra a continuación. En honor a la verdad, lo he hecho a posteriori; es decir, ya había realizado el sketch en Arduino IDE y probado su funcionamiento. Cuando se lea el listado de instrucciones, podrá verse que la acción del motor, tanto para subir como para bajar, se ha configurado con una función a la que se pasa como argumento el sensor hasta el que tiene que llegar. Dichos movimientos se realizan colocando los pines correspondientes en estado alto o bajo siguiendo el siguiente esquema: para subir, pin 12 = HIGH y pin 13 =



LOW; para bajar se obra al contrario y, para parar el motor, se disponen los dos en estado bajo

Listado. -

```
/* Programa para practica con el ascensor.
```

```
Tenemos tres pisos
```

```
cada uno de ellos con sensor de fin de carrera y
```

```
pulsador de llamada. Además un motor con cambio
```

```
de sentido, mueve la cabina
```

```
*/
```

```
int pul01 = 0; // Pulsadores de llamada
```

```
int pul02 = 2;
```

```
int pul03 = 4;
```

```
int sens01 = 6; // Finales de carrera
```

```
int sens02 = 8;
```

```
int sens03 = 10;
```

```
int motorA = 12; // Al driver del motor
```

```
int motorB = 13;
```

```
void sube(int sensor);
```

```
void baja(int sensor);
```

```
void setup() {
```

```
  pinMode(pul01, INPUT);
```

```
  pinMode(pul02, INPUT);
```

```
  pinMode(pul03, INPUT);
```

```
  pinMode(sens01, INPUT);
```

```
  pinMode(sens02, INPUT);
```

```
  pinMode(sens03, INPUT);
```

```

pinMode(motorA, OUTPUT);
pinMode(motorB, OUTPUT);
}

void loop() {
  if (digitalRead(pul01))
  {
    baja(sens01);
  }
  if (digitalRead(pul02))
  {
    if (digitalRead(sens01))
    {
      sube(sens02);
    }
    if (digitalRead(sens03))
    {
      baja(sens02);
    }
  }
  if (digitalRead(pul03))
  {
    sube(sens03);
  }
}

void sube(int sensor) {
  while (!(digitalRead(sensor)))
  {
    digitalWrite(motorA, HIGH); //A=1 y B=0 sube
    digitalWrite(motorB, LOW);
  }
  digitalWrite(motorA, LOW); // A=0 y B=0 parado
}

```

```

void baja(int sensor) {
  while (!(digitalRead(sensor)))
  {
    digitalWrite(motorB, HIGH); //A=0 y B=1 baja
    digitalWrite(motorA, LOW);
  }
  digitalWrite(motorB, LOW); // A=0 y B=0 parado

```

MBLOCK

