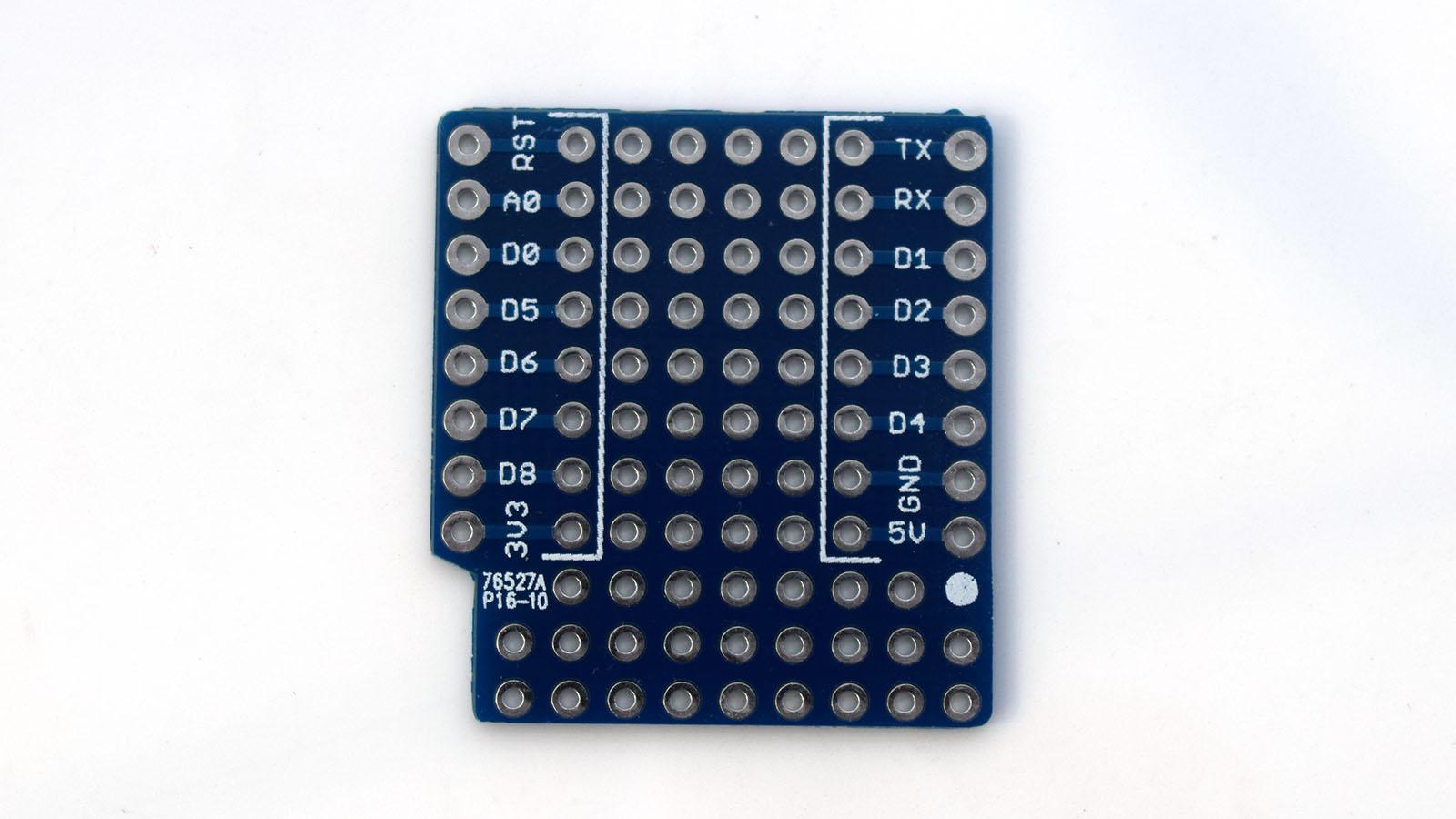
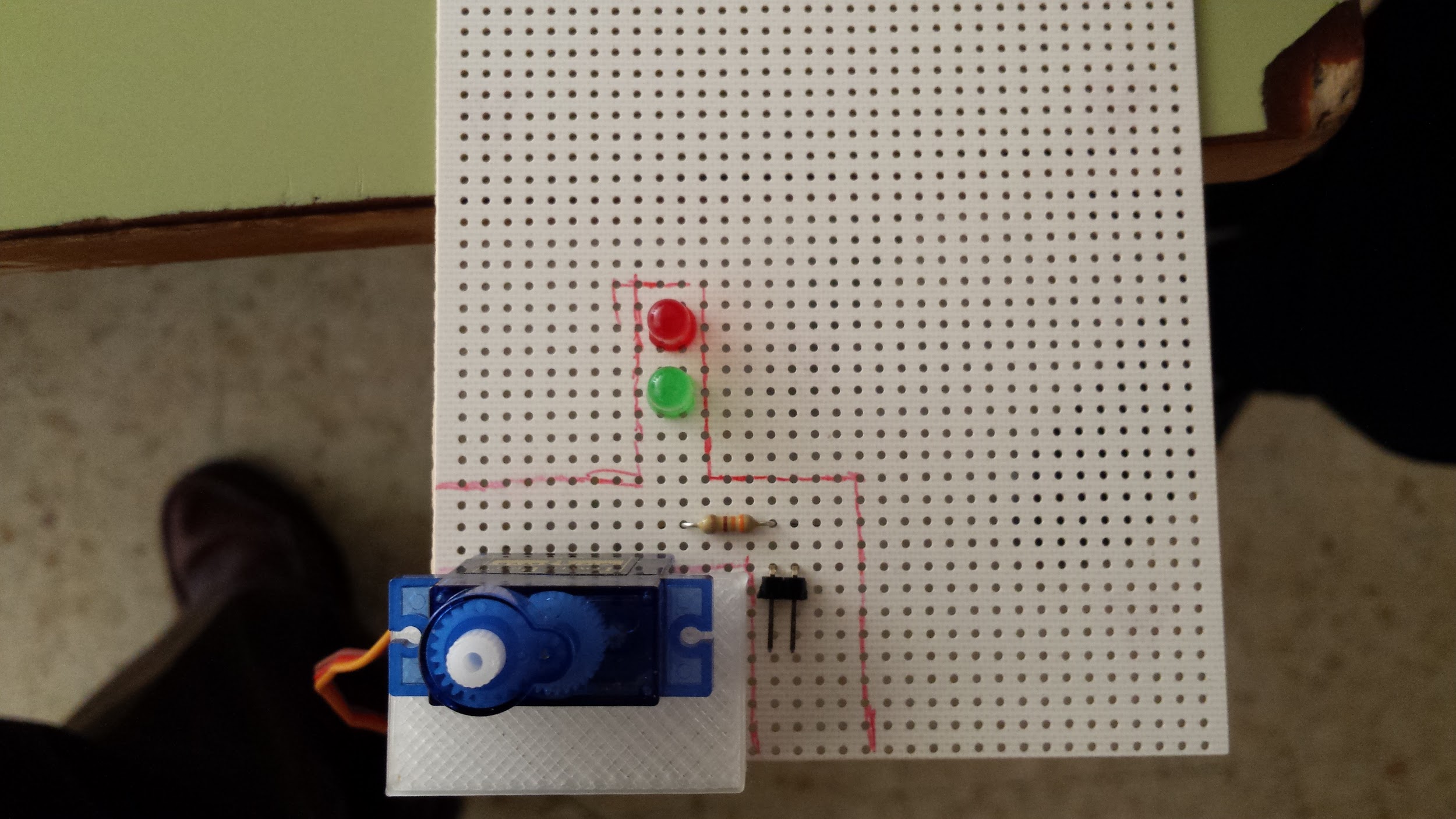
# Maqueta pas amb barrera

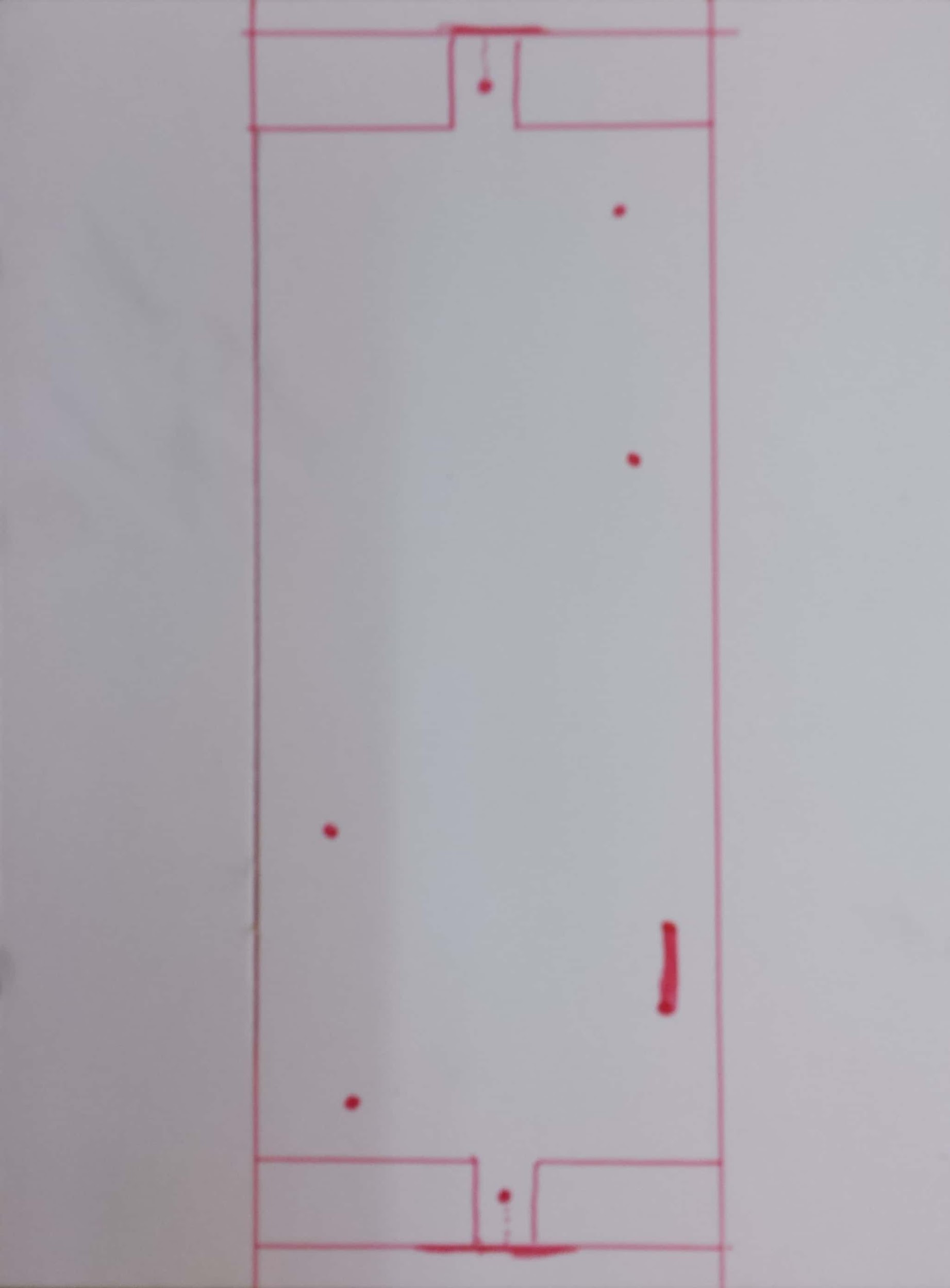
## Connexions D1 mini



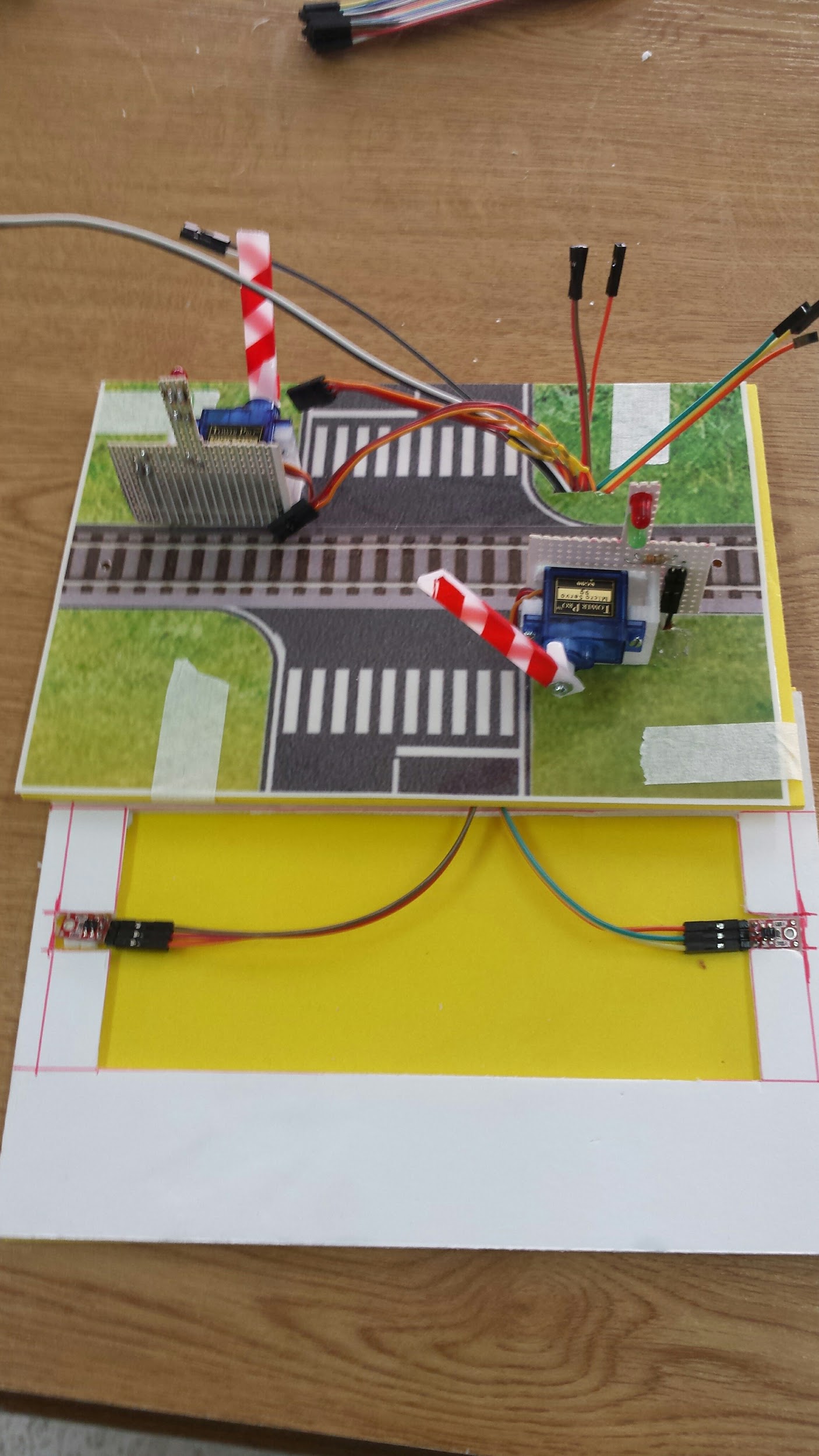
|  |  |  |
| --- | --- | --- |
| D1 mini | Shield / mòdul | Notes |
| D0 |  |  |
| D1 / SCL | Sensor esquerra |  |
| D2 / SDA | Sensors dreta |  |
| D7 | Semàfors | 1 verd 0 vermell |
| D8 | Semàfors | 0 verd 1 vermell |
| D5 | Brunzidor |  |
| D6 |  |  |
| D3 | Servo inf |  |
| D4 | Servo sup |  |
| A0 |  |  |

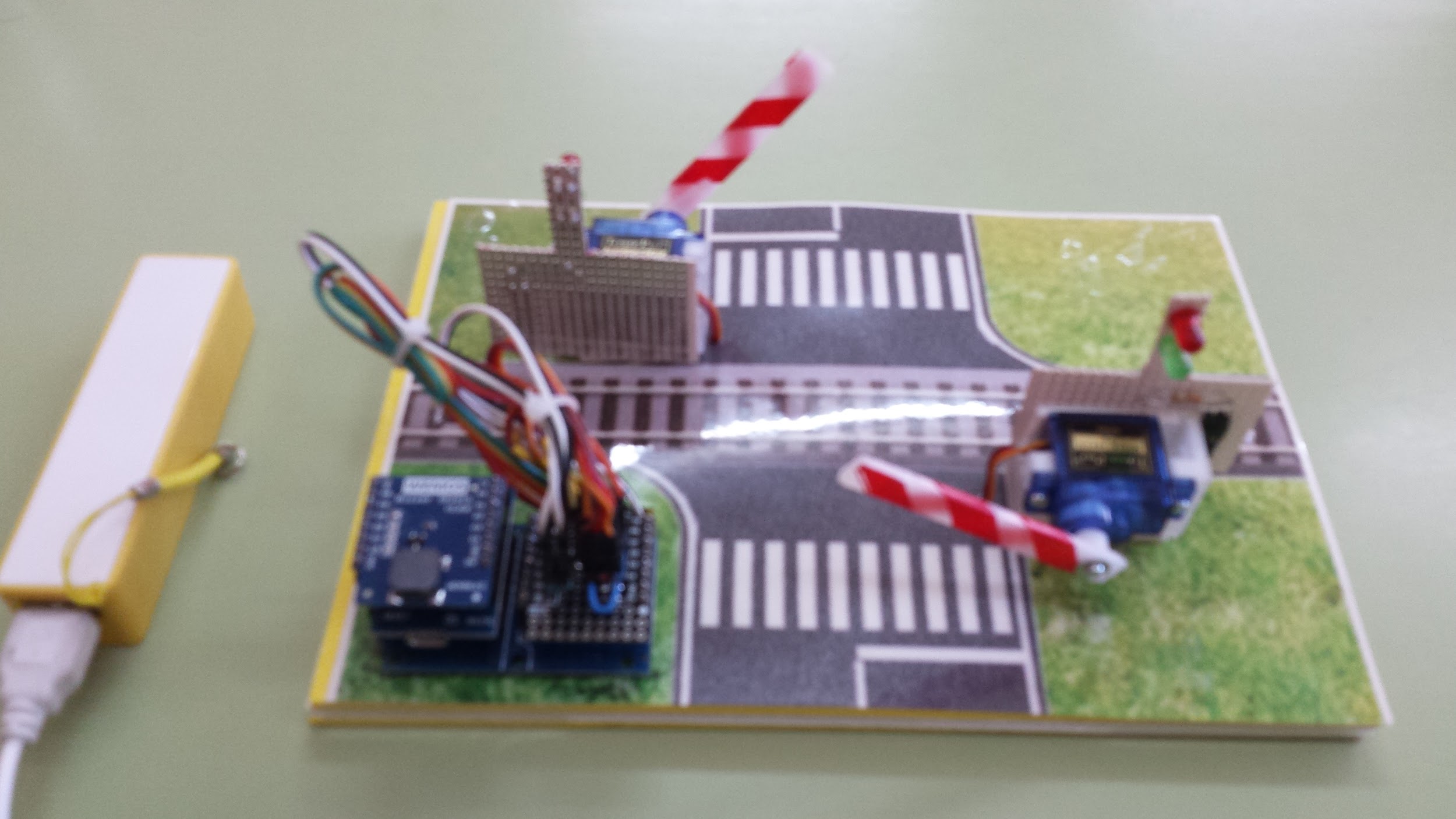
## Maqueta

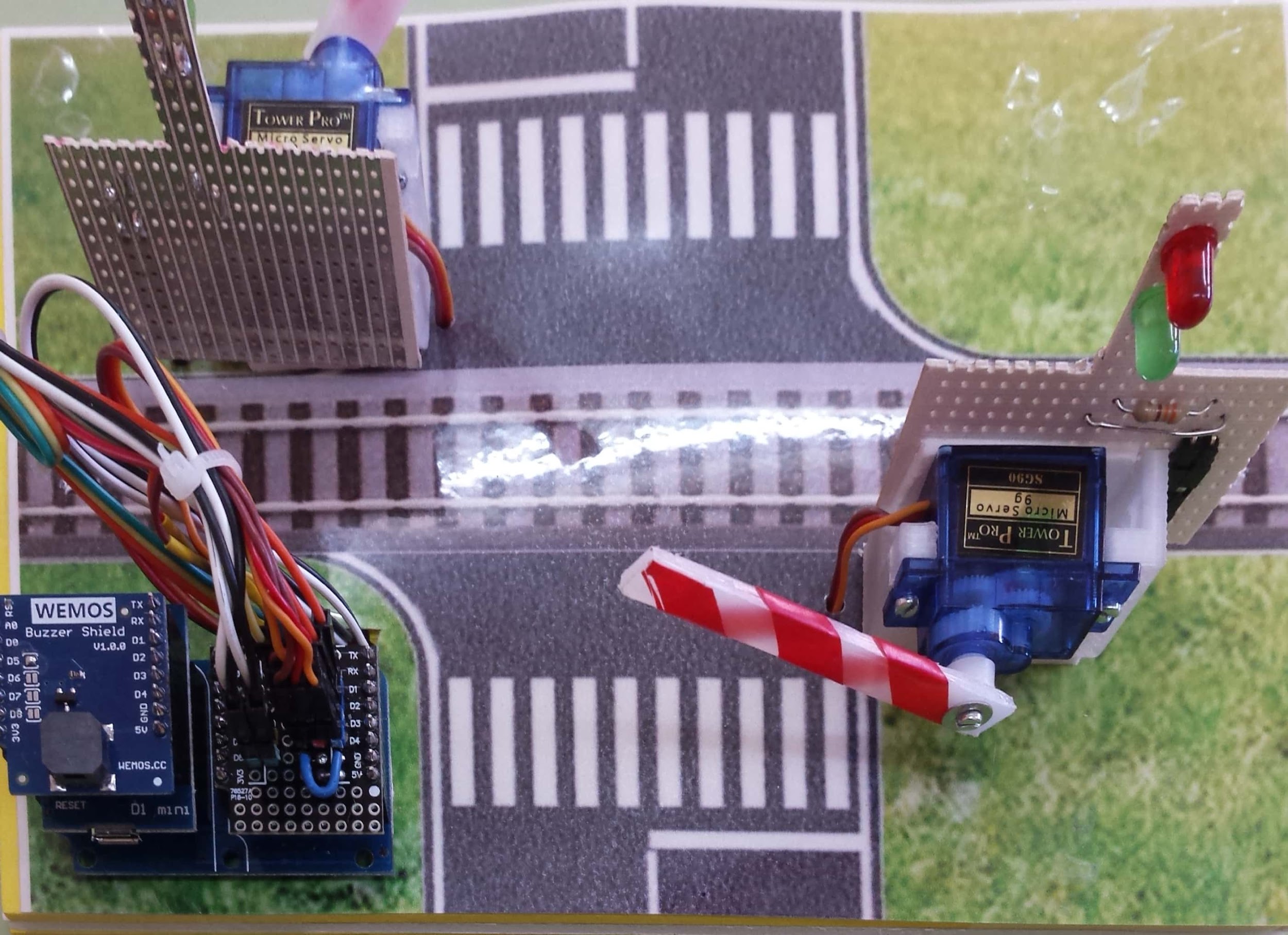


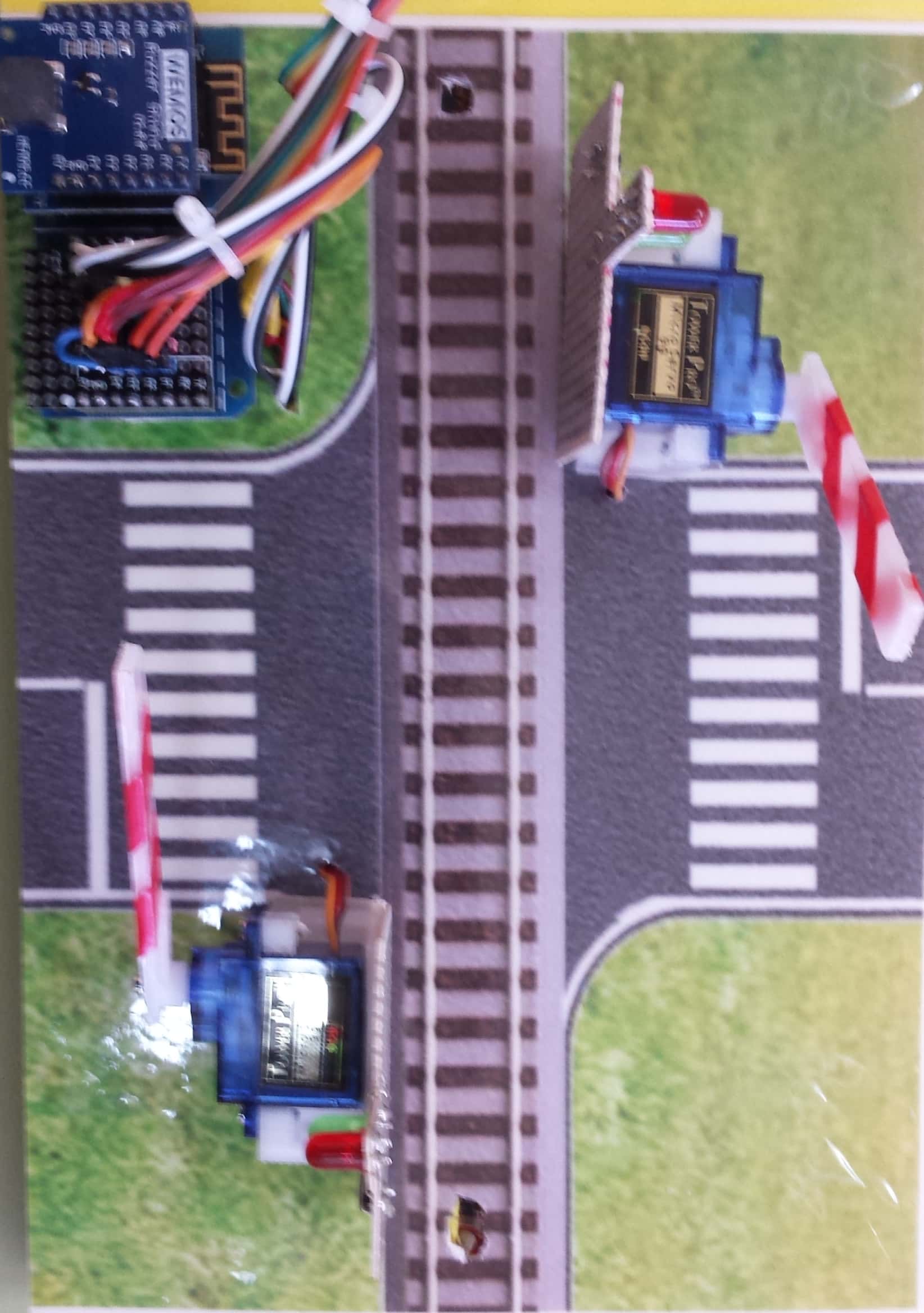












## Firmware

#include <ESP8266WiFi.h>  
#include <WiFiClient.h>   
#include <ESP8266WebServer.h>  
#include <ESP8266mDNS.h>  
  
  
/\* Set these to your desired credentials. \*/  
const char \*ssid = "ESPap";  
const char \*password = "trenwemosD1";  
const char\* host = "maqueta";  
  
  
  
ESP8266WebServer server(80);  
  
/\* Just a little test message. Go to http://192.168.4.1 in a web browser  
 \* connected to this access point to see it.  
 \*/  
  
const int sensorE = D1;  
const int sensorD = D2;  
int estatE = 0;  
int estatD = 0;  
  
const int buzzerPin = D5;  
const long interval = 500; // pause for 500 useconds  
  
#include <Servo.h>  
Servo servo1; // create servo object to control a servo  
Servo servo2; // create servo object to control a servo  
  
void pageOk(){  
 server.send(200, "text/html", "<h1>Comanda ok</h1>");  
}  
  
void sensE(){  
 if (digitalRead(sensorE) == LOW)   
 server.send(200, "text/html", "1");  
 else  
 server.send(200, "text/html", "0");  
}  
void sensD(){  
 if (digitalRead(sensorD) == LOW)   
 server.send(200, "text/html", "1");  
 else  
 server.send(200, "text/html", "0");  
}  
void histE(){  
 if (estatE == 1)   
 server.send(200, "text/html", "1");  
 else  
 server.send(200, "text/html", "0");  
 estatE=0;  
}  
void histD(){  
 if (estatD ==1)   
 server.send(200, "text/html", "1");  
 else  
 server.send(200, "text/html", "0");  
 estatD=0;  
}  
  
void apagat(){  
 digitalWrite(D7, LOW);  
 digitalWrite(D8, LOW);  
 pageOk();  
}  
void verd(){  
 digitalWrite(D7, HIGH);  
 digitalWrite(D8, LOW);  
 pageOk();  
}  
void vermell(){  
 digitalWrite(D7, LOW);  
 digitalWrite(D8, HIGH);  
 pageOk();  
}  
void pujab1(){  
 servo1.write(45); // tell servo to go to position V  
 pageOk();  
}  
void baixab1(){  
 servo1.write(135); // tell servo to go to position H  
 pageOk();  
}  
void pujab2(){  
 servo2.write(45); // tell servo to go to position V  
 pageOk();  
}  
void baixab2(){  
 servo2.write(135); // tell servo to go to position H  
 pageOk();  
}  
  
void campana(){  
 for(int i=0;i<1000;i++){  
 digitalWrite(buzzerPin, HIGH); // turn on relay with voltage HIGH  
 delayMicroseconds(interval); // pause  
 digitalWrite(buzzerPin, LOW); // turn off relay with voltage LOW  
 delayMicroseconds(interval); // pause  
 pageOk();  
 }  
}  
  
void handleNotFound() {  
 String message = "File Not Found\n\n";  
 message += "URI: ";  
 message += server.uri();  
 message += "\nMethod: ";  
 message += ( server.method() == HTTP\_GET ) ? "GET" : "POST";  
 message += "\nArguments: ";  
 message += server.args();  
 message += "\n";  
  
 for ( uint8\_t i = 0; i < server.args(); i++ ) {  
 message += " " + server.argName ( i ) + ": " + server.arg ( i ) + "\n";  
 }  
  
 server.send ( 404, "text/plain", message );  
}  
  
void handleRoot() {  
String menu = "<h1>Maqueta pas a nivell</h1></br>";  
menu += "<h2>Comandes:</h2></br>";  
menu += "<a href='/off'>off</a></br>";  
menu += "<a href='/green'>green</a></br>";  
menu += "<a href='/red'>red</a></br>";  
menu += "<a href='/b1up'>b1up</a></br>";  
menu += "<a href='/b1dn'>b1dn</a></br>";  
menu += "<a href='/b2up'>b2up</a></br>";  
menu += "<a href='/b2dn'>b2dn</a></br>";  
menu += "<a href='/bell'>bell</a></br>";  
menu += "<a href='/sensorE'>sensorE</a></br>";  
menu += "<a href='/sensorD'>sensorD</a></br>";  
menu += "<a href='/canviE'>canviE</a></br>";  
menu += "<a href='/canviD'>canviD</a></br>";  
  
 server.send(200, "text/html", menu);  
}  
  
  
void setup() {  
 pinMode(D1, INPUT);  
 pinMode(D2, INPUT);  
 pinMode(buzzerPin, OUTPUT);  
 pinMode(D7, OUTPUT);  
 pinMode(D8, OUTPUT);  
 servo1.attach(D3); // attaches the servo on pin D3  
 servo2.attach(D4); // attaches the servo on pin D4  
  
 delay(1000);  
 Serial.begin(115200);  
 Serial.println();  
 Serial.print("Configuring access point...");  
 /\* You can remove the password parameter if you want the AP to be open. \*/  
 WiFi.softAP(ssid, password);  
  
 IPAddress myIP = WiFi.softAPIP();  
 Serial.print("AP IP address: ");  
 Serial.println(myIP);  
  
 MDNS.begin(host);  
  
 server.on("/", handleRoot);  
 server.on ( "/off", apagat );  
 server.on ( "/green", verd );  
 server.on ( "/red", vermell );  
 server.on ( "/b1up", pujab1 );  
 server.on ( "/b1dn", baixab1 );  
 server.on ( "/b2up", pujab2 );  
 server.on ( "/b2dn", baixab2 );  
 server.on ( "/bell", campana );  
 server.on ( "/sensorE", sensE );  
 server.on ( "/sensorD", sensD );  
 server.on ( "/canviE", histE );  
 server.on ( "/canviD", histD );  
 server.onNotFound ( handleNotFound );  
   
 server.begin();  
 Serial.println("HTTP server started");  
  
  
}  
void loop() {  
 if (digitalRead(sensorE) == LOW)   
 estatE=1;  
 if (digitalRead(sensorD) == LOW)   
 estatD=1;  
  
 server.handleClient();  
}