/\*\*\*\*\*\*\*\*UNIVERSITY OF MANCHESTER SCHOOL OF EE&E\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*FOURTH YEAR PROJECT: ROBOT ORCHESTRA\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*Test code for ESP8266 Wifi Module

\* Written by: Andrei Buruiana

\* email: andrei.buruiana27@gmail.com

\*/

/\*documentation link: http://arduino-esp8266.readthedocs.io/en/latest/index.html

/\*CONNECTIONS:

ESP8266 ARDUINO

RX RXD

VCC 3V3

GPIO\_0 GND (puts the esp module into flash mode)

RESET

GPIO\_2 could connect to LED

CH\_PD 3V3 (if connected to GND, smaller current->low LED brightness)

GND GND

TX TXD

RES -> connect to GND (Arduino will

work as a USB to serial connector)

\*/

#include <ESP8266WiFi.h>

const char\* ssid = "bobo";

const char\* password = "bobobobo";

//const char\* ssid = "SKY05CBE";

//const char\* password = "XWWRADUV";

//const char\* ssid = "BTHub5-FPFK";

//const char\* password = "bad675c7e4";

int LED\_pin = 2; // GPIO2 of ESP8266

WiFiServer server(80);//Service Port

void setup() {

Serial.begin(115200);

delay(10);

pinMode(LED\_pin, OUTPUT);

digitalWrite(LED\_pin, LOW);

String ssid\_search;

uint8\_t encryptionType;

int32\_t RSSI;

uint8\_t\* BSSID;

int32\_t channel;

bool isHidden;

//scan for available networks

int networks = WiFi.scanNetworks();

for (int i = 0; i < networks; i++){

WiFi.getNetworkInfo(i, ssid\_search, encryptionType, RSSI, BSSID, channel, isHidden);

Serial.printf("%d: %s, Ch:%d (%ddBm) \n", i + 1, ssid\_search.c\_str(), channel, RSSI);

}

Serial.println();

Serial.println();

Serial.print("Connecting to ");

Serial.println(ssid);

//connect to WiFi

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(200);

Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected");

// Start the server

server.begin();

Serial.println("Server started");

// Print WiFi module IP address

Serial.print("Access the WiFi module at the address: ");

Serial.print("http://");

Serial.print(WiFi.localIP());

Serial.println("/");

}

void loop() {

// Check if a client has connected

WiFiClient client = server.available(); //returns server object if available

if (!client) {

return;

}

// Wait until the client sends some data

Serial.println("new request");

while(!client.available()){

delay(1);

}

// Read the first line of the request

String request = client.readStringUntil('\r');

Serial.println(request);

client.flush();

// Match the request on the LED

int value = LOW;

if (request.indexOf("/LED=ON") != -1) {

digitalWrite(LED\_pin, HIGH);

value = HIGH;

}

if (request.indexOf("/LED=OFF") != -1){

digitalWrite(LED\_pin, LOW);

value = LOW;

}

if (request.indexOf("/LED=BLINK") != -1){

for(int k=0; k<10; k++){

digitalWrite(LED\_pin, HIGH);

delay(100);

digitalWrite(LED\_pin, LOW);

delay(100);

}

}

//Print on thr webpage

client.println("HTTP/1.1 200 OK");

client.println("Content-Type: text/html");

client.println(""); // do not forget this one

client.println("<!DOCTYPE HTML>");

client.println("<html>");

client.print("Led pin is now: ");

if(value == HIGH) {

client.print("On");

} else {

client.print("Off");

}

client.println("<br><br>");

client.println("Turn the LED <a href=\"/LED=ON\">ON</a><br>");

client.println("Turn the LED <a href=\"/LED=OFF\">OFF</a><br>");

client.println("</html>");

client.println("Turn the LED <a href=\"/LED=BLINK\">BLINKING</a><br>");

client.println("</html>");

delay(1);

Serial.println("request solved");

Serial.println("");

}