/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CÓDIGO 1:

Código para encender y apagar un led conectado al pin D4 con un delay de 1.5 segundos

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#define pulso D4

void setup(){

 pinMode(D4,OUTPUT);

}

void loop(){

 digitalWrite(pulso,HIGH);

 delay(1500);

 digitalWrite(pulso,LOW);

 delay(1500);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

 CÓDIGO 2:

 Controla un LED RGB vía http. El dispositivo se conecta a una red WIFI y se comporta

 como un servidor WEB que puede recibir peticiones a través de URLs y realizar

 acciones.

 URLs para activar acciones en el ESP8266:

 [http://x.x.x.x/LED=AZUL](http://x.x.x.x/LED%3DAZUL)

 http://x.x.x.x/LED=ROJO

 http://x.x.x.x/LED=VERDE

 [http://x.x.x.x/LED=OFF](http://x.x.x.x/LED%3DOFF)

 [http://x.x.x.x/LED=EFECTO1](http://x.x.x.x/LED%3DEFECTO1)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <ESP8266WiFi.h>

#define rojo D6

#define verde D7

#define azul D5

#define max 255

int i=0;

String ssid = "INFINITUM7M84\_2.4";

String password = "kzbnzdgNK7";

WiFiServer server(80); //objeto de la clase WiFiServer

int estado = 0;

void setup() {

 // Definimos los pins

 pinMode(rojo, OUTPUT); // Red: D6

 pinMode(verde, OUTPUT); // Green: D7

 pinMode(azul, OUTPUT); // Blue: D5

 // Inicia Serial

 Serial.begin(115200);

 Serial.println("\n");

 color(0,0,0);

 // Conexión WIFI

 WiFi.begin(ssid, password);

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

 color(max,0,0);

 delay(250);

 color(0,0,0);

 delay(250);

 Serial.print(".");

 }

 color(0,max,0);

 delay(3000);

 color(0,0,0);

 Serial.println("");

 Serial.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

 Serial.print("Conectado a la red WiFi: ");

 Serial.println(WiFi.SSID());

 Serial.print("IP: ");

 Serial.println(WiFi.localIP());

 Serial.print("macAdress: ");

 Serial.println(WiFi.macAddress());

 Serial.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

 server.begin(); //begin() levantamos el servidor

}

void color (int R, int G, int B) {

 analogWrite(rojo, max-R);

 analogWrite(verde, max-G);

 analogWrite(azul, max-B);

}

void loop() {

 WiFiClient client = server.available(); //objeto de la clase WiFiClient

 // avalaible() detecta un cliente nuevo del objeto de la clase WifiServer

 if(!client){

 return;

 }

 Serial.println("Nuevo cliente...");

 while(!client.available()){ //espera a un cliente diponible

 delay(1);

 }

 String peticion = client.readStringUntil('\r'); //lee la peticion del cliente

 Serial.println(peticion);

 client.flush(); //limpia la peticion del cliente

 if(peticion.indexOf("LED=AZUL") != -1)

 {

 color(0,0,max);

 }

 if(peticion.indexOf("LED=ROJO") != -1)

 {

 color(max,0,0);

 }

 if(peticion.indexOf("LED=VERDE") != -1)

 {

 color(0,max,0);

 }

 if(peticion.indexOf("LED=OFF") != -1)

 {

 color(0,0,0);

 }

 if(peticion.indexOf("LED=EFECTO1") != -1)

 {

 for(i=1;i<=5;i++){

 delay(150);

 color(0,max,0);

 delay(150);

 color(0,0,0);

 }

 for(i=1;i<=5;i++){

 delay(150);

 color(max,0,0);

 delay(150);

 color(0,0,0);

 }

 for(i=1;i<=5;i++){

 delay(150);

 color(0,0,max);

 delay(150);

 color(0,0,0);

 }

 }

 if(peticion.indexOf("LED=EFECTO2") != -1)

 {

 for(i=1;i<=5;i++){

 color(0,max,0);

 delay(350);

 color(0,0,0);

 delay(150);

 color(max,0,0);

 delay(350);

 color(0,0,0);

 delay(150);

 color(0,0,max);

 delay(350);

 color(0,0,0);

 delay(150);

 }

 }

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

 client.println("");

 client.println("");

 client.println("");

 client.println("");

 delay(10);

 Serial.println("Peticion finalizada");

 Serial.println("");

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CÓDIGO 3:

Código para encender y apagar un dispositivo vía http

URLs:

http://x.x.x.x/LED=ON

[http://x.x.x.x/LED=OFF](http://x.x.x.x/LED%3DOFF)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <ESP8266WiFi.h>

#define azul D5

int i=0;

String ssid = "INFINITUM7M84\_2.4";

String password = "kzbnzdgNK7";

WiFiServer server(80); //objeto de la clase WiFiServer

int estado = 0;

void setup() {

 // Definimos los pins

 pinMode(azul, OUTPUT); // Blue: D5

 // Inicia Serial

 Serial.begin(115200);

 Serial.println("\n");

 // Conexión WIFI

 WiFi.begin(ssid, password);

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

 Serial.print(".");

 }

 Serial.println("");

 Serial.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

 Serial.print("Conectado a la red WiFi: ");

 Serial.println(WiFi.SSID());

 Serial.print("IP: ");

 Serial.println(WiFi.localIP());

 Serial.print("macAdress: ");

 Serial.println(WiFi.macAddress());

 Serial.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

 server.begin(); //begin() levantamos el servidor

}

void loop() {

 WiFiClient client = server.available(); //objeto de la clase WiFiClient

 // avalaible() detecta un cliente nuevo del objeto de la clase WifiServer

 if(!client){

 return;

 }

 Serial.println("Nuevo cliente...");

 while(!client.available()){ //espera a un cliente diponible

 delay(1);

 }

 String peticion = client.readStringUntil('\r'); //lee la peticion del cliente

 Serial.println(peticion);

 client.flush(); //limpia la peticion del cliente

 if(peticion.indexOf("LED=ON") != -1)

 {

 digitalWrite(azul,HIGH);

 }

 if(peticion.indexOf("LED=OFF") != -1)

 {

 digitalWrite(azul,LOW);

 }

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

 client.println("");

 client.println("");

 client.println("");

 client.println("");

 delay(10);

 Serial.println("Peticion finalizada");

 Serial.println("");

}