

```
#include <ESP8266WiFi.h>

const char* ssid = "";
const char* password = "";

short rc_pin=3;
int ledPin=2;
WiFiServer server(80);

void setup() {
  Serial.begin(115200);
  delay(10);

  pinMode(rc_pin, OUTPUT); //definiere rc_Pin als Ausgang (schliesslich wollen wir senden)
  pinMode(ledPin, OUTPUT);

  // Connect to WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");

  // Start the server
```

```
server.begin();  
Serial.println("Server started");  
  
// Print the IP address  
Serial.print("Use this URL to connect: ");  
Serial.print("http://");  
Serial.print(WiFi.localIP());  
Serial.println("/");  
  
}  
  
void loop() {  
  
    digitalWrite(ledPin, HIGH);  
    delay(2000);  
    digitalWrite(ledPin, LOW);  
    delay(2000);  
  
    // Check if a client has connected  
    WiFiClient client = server.available();  
    if (!client) {  
        return;  
    }  
  
    // Wait until the client sends some data  
    Serial.println("new client");  
    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

int value = LOW;
if (request.indexOf("/LED=ON") != -1) {
    sendCode("111110100010");
    digitalWrite(ledPin,HIGH);
}
if (request.indexOf("/LED=OFF") != -1) {
    sendCode("111110100001");
    digitalWrite(ledPin,LOW);
}

// Set ledPin according to the request
//digitalWrite(ledPin, value);

// Return the response
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println("");
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("<a href=\"/LED=ON\"><button>Turn On </button></a>");
client.println("<a href=\"/LED=OFF\"><button>Turn Off </button></a><br />");
client.println("</html>");
```

```
delay(1);
Serial.println("Client disconnected");
Serial.println("");
```

```
}
```

```
boolean sendCode(char code[]){ //empfangen den Code in Form eines Char[]
```

```
for(short z = 0; z<7; z++){ //wiederhole den Code 7x
```

```
for(short i = 0; i<12; i++){ //ein Code besteht aus 12bits
```

```
sendByte(code[i]);
```

```
}
```

```
sendByte('x'); //da der code immer mit x/sync abschliesst, brauchen wir den nicht im code und  
haengen es autisch immer hinten ran.
```

```
}
```

```
return true;
```

```
}
```

```
void sendByte(char i) { //Diese Funktion soll 0,1 oder x senden koennen. Wir speichern die  
gewuenschte Ausgabe in der Variabel i
```

```
switch(i){ //nun gucken wir was i ist
```

```
case '0':{ //Der Code fuer '0'
```

```
digitalWrite(rc_pin,HIGH);
```

```
wait(1); //da die Pausen x*350us lang sind, machen wir daraus eine Funktion
```

```
digitalWrite(rc_pin,LOW);
```

```
wait(3);
```

```
digitalWrite(rc_pin,HIGH);
```

```
wait(3);
```

```
digitalWrite(rc_pin,LOW);
```

```
wait(1);
```

```
return;
```

```
}  
case '1':{ //Der Code fuer '1'  
    digitalWrite(rc_pin,HIGH);  
    wait(1);  
    digitalWrite(rc_pin,LOW);  
    wait(3);  
    digitalWrite(rc_pin,HIGH);  
    wait(1);  
    digitalWrite(rc_pin,LOW);  
    wait(3);  
    return;  
}  
case 'x':{ //Der Code fuer x(sync)  
    digitalWrite(rc_pin,HIGH);  
    wait(1);  
    digitalWrite(rc_pin,LOW);  
    wait(31);  
}  
  
}  
}  
void wait(int x)  
{  
    delayMicroseconds(x*350); //warte x*350us  
}
```