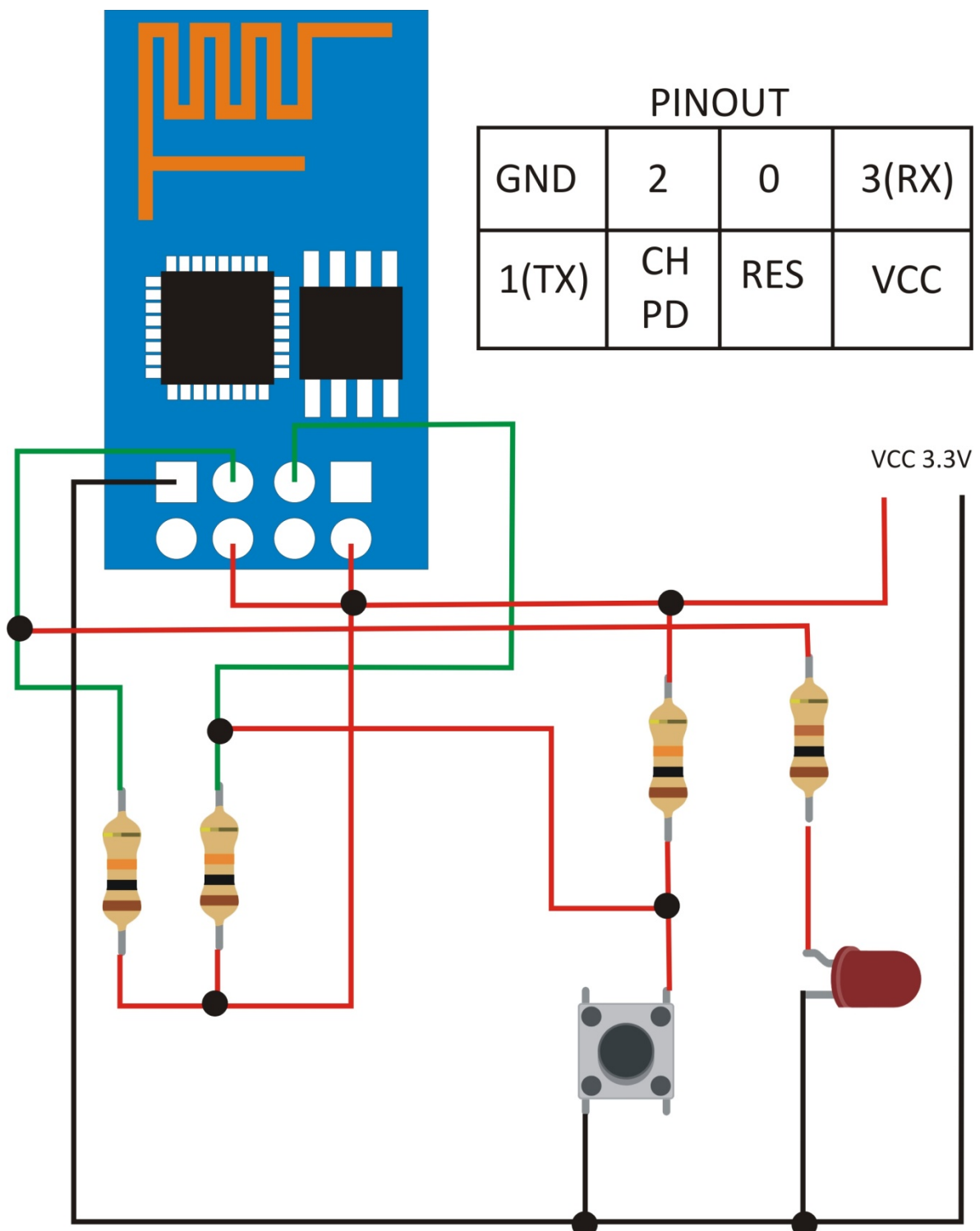


## WEB SERVER CON PULSANTE E NOTIFICA

Un esempio emblematico del funzionamento di un web server in una rete locale (LAN). Alla pressione del pulsante si accenderà il led (processo avviato). Se un client ne farà richiesta (digitando <http://esp8266.local/>) il server risponderà con un messaggio. Dal client sarà anche possibile accendere lo stesso led digitando <http://esp8266.local/on> e spegnerlo con <http://esp8266.local/off>.

IMPORTANTE: Nello schema di funzionamento si sono aggiunte due resistenze di pull up da 10k tra pin 2 e pin 0 verso Vcc. Senza di queste il funzionamento della scheda potrebbe risultare instabile, specialmente per applicazioni su breadboard.



## Sketch per l'IDE di Arduino

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <ESP8266mDNS.h>
int led=2;
int puls=0;

//per una maggiore stabilità di funzionamento, collegare tra il pin 2 e
//Vcc e tra il pin 0 e Vcc due resistenze da 10k.

int flag=0;
int statopuls=0;

ESP8266WebServer server(80);
//crea l'oggetto server sulla porta classica (80)

void setup()
{
  Serial.begin(115200);
  pinMode(led,OUTPUT);
  pinMode(puls,INPUT);
  //Serial.begin(115200);

  WiFi.mode(WIFI_STA);//per collegarsi alla rete locale
  WiFi.begin("*****", "*****");//inserire SSID e password di rete

  // aspetta la connessione
  while (WiFi.status() != WL_CONNECTED) //finchè non si connette....
  {
    delay(500);
    Serial.print(".");
    //stampa un punto (lo vedremo nel caso avessimo a disposizione un
    //serial //monitor)
  }
  Serial.println(WiFi.localIP());//si può stampare l'IP della scheda

  // se non si conosce l'IP della scheda o non si ha a disposizione
  //un serial monitor (come nel caso dell'ESP01)

  MDNS.begin("esp8266");
  //per dare l'indirizzo http://esp8266.local/
  //come parametro forniamo un alias del nome della scheda in questo
  //caso esp8266
```

```

server.on("/on", ledon);
//http://esp8266.local/on nel caso si digiti /on chiamerà la
//subroutine ledon()
  server.on("/off", ledoff);
//http://esp8266.local/off nel caso si digiti /off chiamerà la
//subroutine ledoff()

  server.on("/", ledstato);
//http://esp8266.local/ nel caso si digiti solo / chiamerà la
//subroutine ledstato()

  server.begin(); //avvia il server
}

void loop()
{
  server.handleClient();
  //controlla periodicamente se ci sono richieste di un client
  MDNS.update();//aggiorna il nome alias
  statopuls=digitalRead(puls);

if (statopuls==LOW)
  {
  while(statopuls==LOW)
  {
  statopuls=digitalRead(puls);
  }
  flag=!flag;//cambia lo stato della flag

  delay(200);
  }

if (flag==1) /analizza lo stato della flag
  {
  digitalWrite(led,LOW);
  }
else
  {
  digitalWrite(led,HIGH);
  }

}

void ledon()
// la funzione che viene chiamata in caso si digiti /on
{
  digitalWrite(led,HIGH); //
  server.send(200, "text/plain", "led on");
  //tre parametri: 200 corrisponde a una pagina,

```

```
    // "text/plain" a testo semplice, "led on" è il messaggio flag=0;
}
void ledoff()
// la funzione che viene chiamata in caso si digiti /off
{
    digitalWrite(led,LOW);
    server.send(200, "text/plain", "led off");
    flag=1;
}

void ledstato()
//la funzione che viene chiamata in caso si digiti solo /
if (flag==0)
{
    server.send(200, "text/plain", "Avvio processo");
}
else
{
    server.send(200, "text/plain", "Stop processo");
}
}
```