



Arduino Day 2017



MOODLAMP

Lampada RGB WiFi con NodeMCU

Davide Di Gesualdo
Embedded Software Engineer

info@davidedigesualdo.it

Daniele Di Giacomo
Android Software Engineer

daniele.digiacomo@hotmail.it

Componenti del sistema

NodeMCU

ESP8266 + PCB, connettori, passivi, convertitore USB/UART

Power LED RGB

3 canali, anodo in comune

Driver per LED RGB

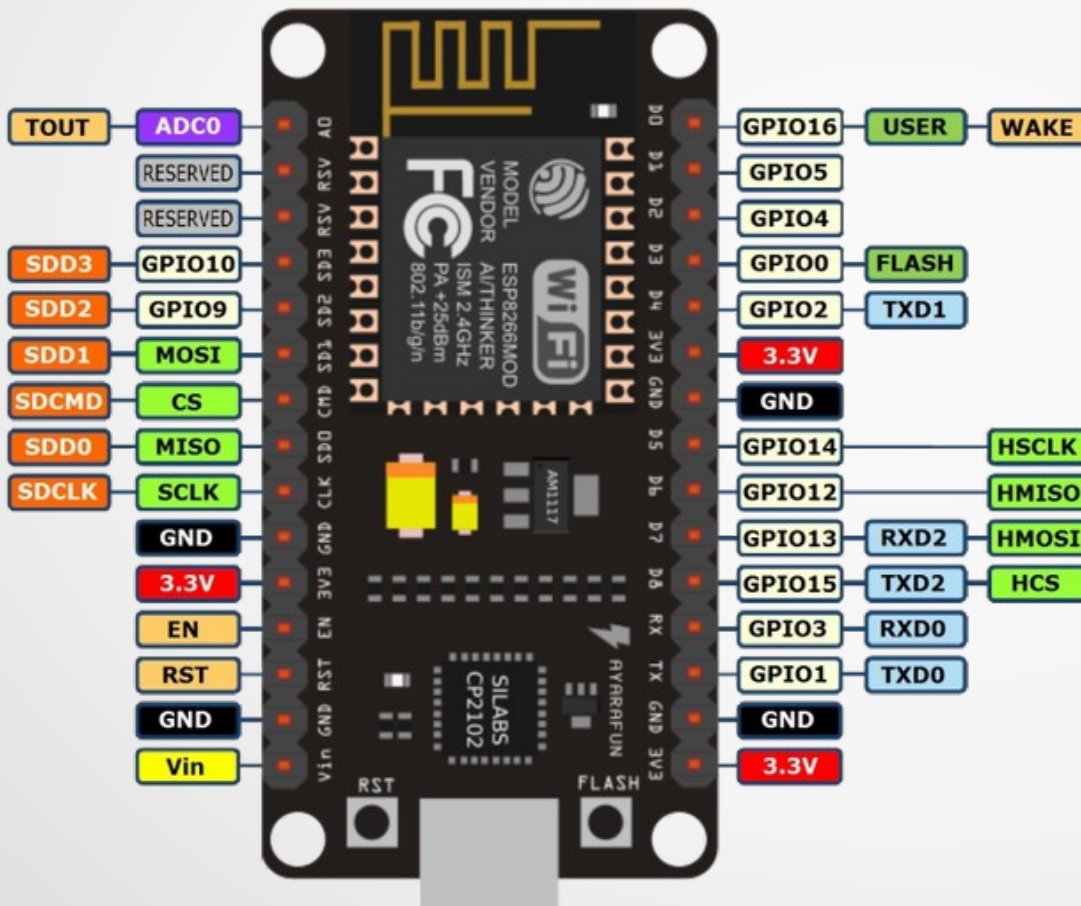
Componenti discreti: BC337, resistenze

App Android

Comunicazione UDP, color picker, messaggi JSON

NodeMCU

Open-source, Interactive, Programmable, Low cost, Simple, Smart, WI-FI enabled

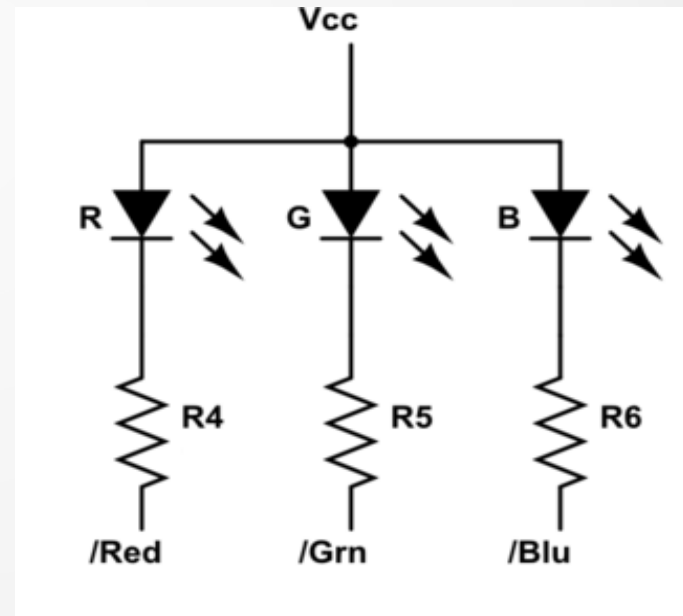


15 GPIO
15 PWM
2 UART
I2C
1wire
SPI
I2S
ADC

ESP8266 core
4MB flash
128kB RAM

RGB power LED

3W power LED
anodo comune



Driver per Power LED

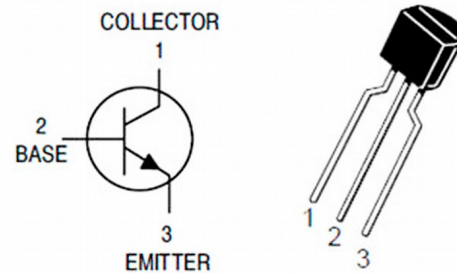
Componenti discreti

BC337

NPN transistor, case TO-92,

3 resistori polarizzazione base

3 resistori polarizzazione carico

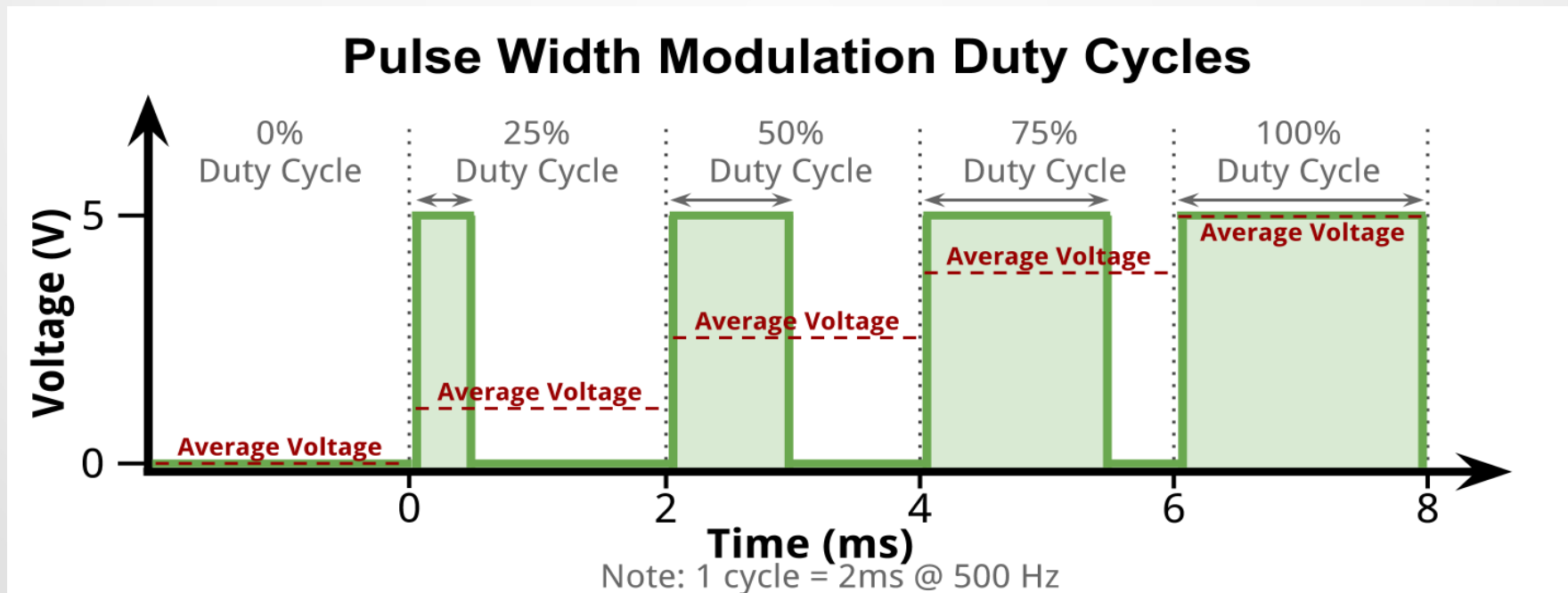


Pilotaggio LED: PWM

PWM: Pulse Width Modulation

Duty cycle variabile

da 0 a 100%, la tensione media varia linearmente



App Android

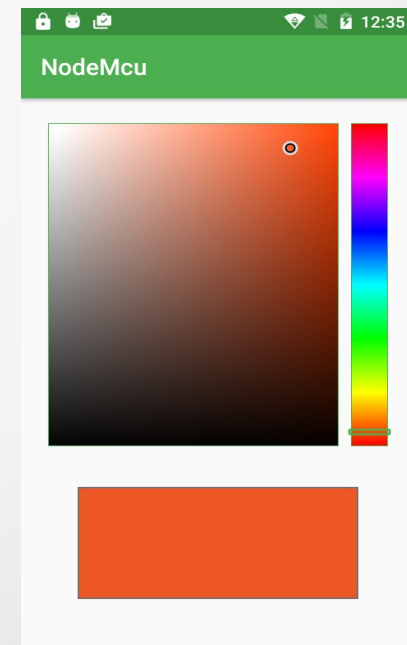
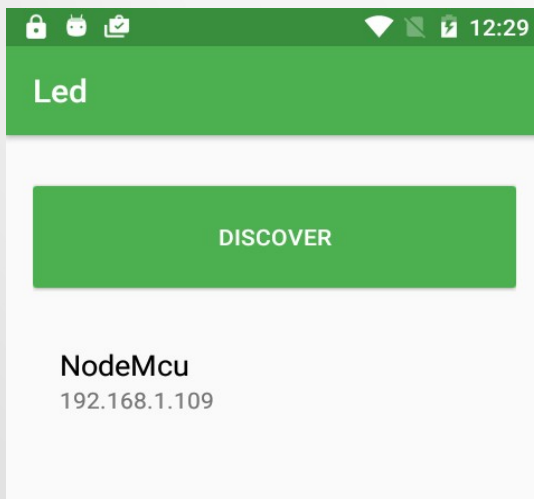
Comunicazione UDP

Messaggi JSON

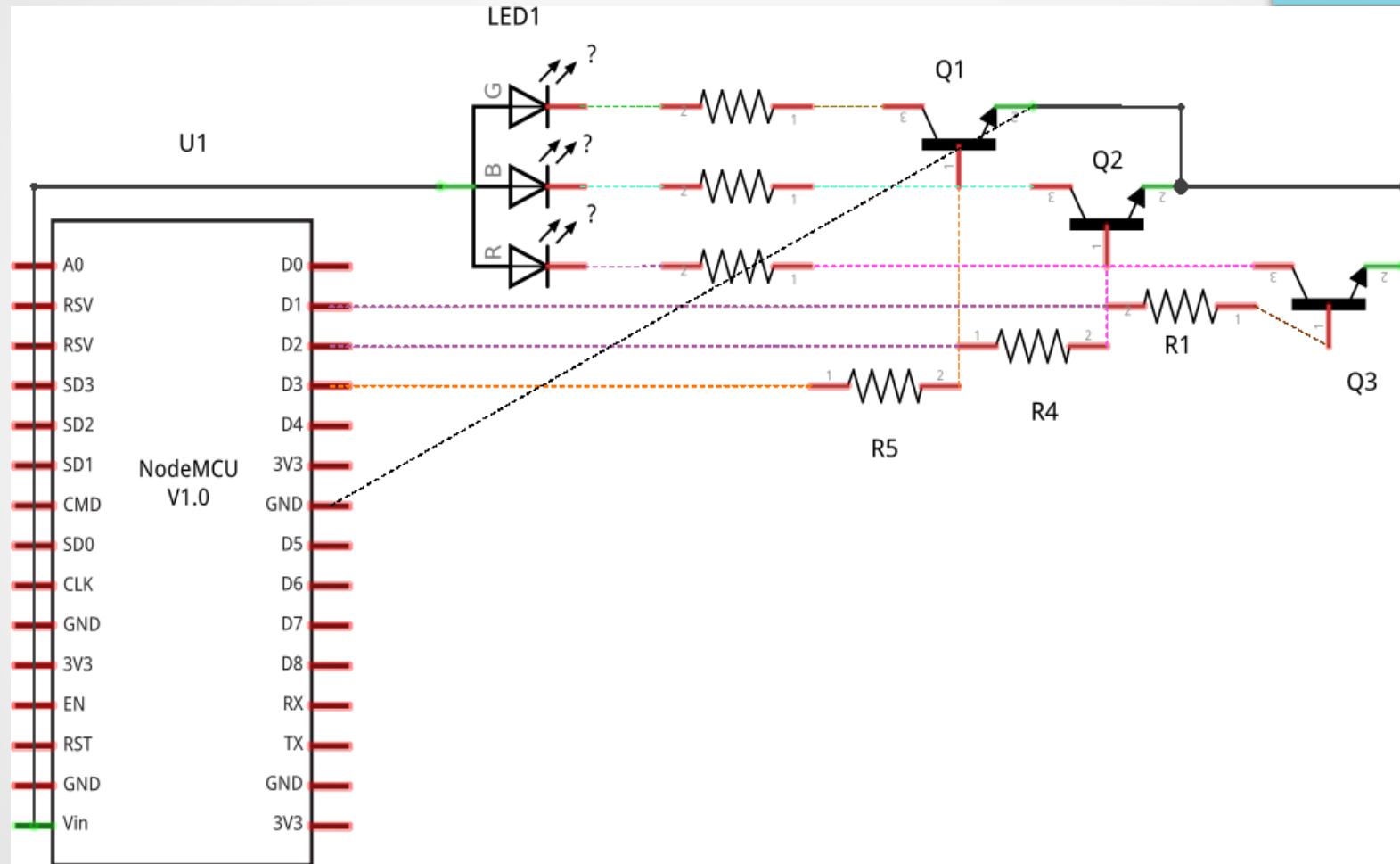
discover: {'command': 'whois'}

debug: {'command': 'debug', 'value': 1}

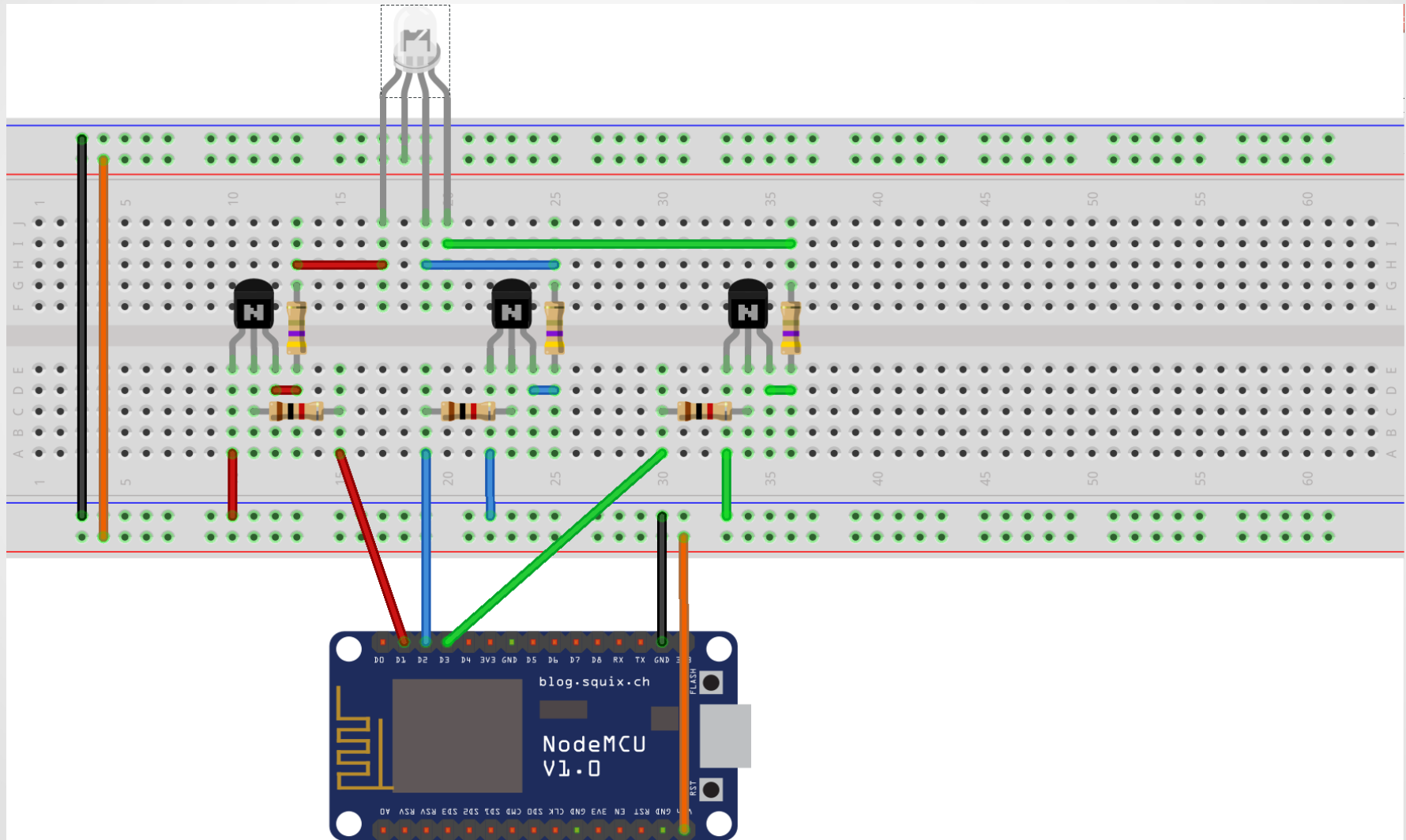
colors: {'command': 'colors', 'r': 50,
'g': 124, 'b': 220}



Schematico



Breadboard



Snippet codice (1)

```
#include <ESP8266WiFi.h>
#include <ESP8266WiFiMulti.h>
#include <WiFiUdp.h>
#include <ArduinoJson.h>

#define LED_RED          5 // D1
#define LED_GREEN       0 // D3
#define LED_BLUE        4 // D2

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

WiFiUDP udpSocket;
unsigned int localPort = 11111;

char packetBuffer[255];

int red = 0;
int green = 0;
int blue = 0;

short debug = 0;
```

Snippet codice (2)

```
void setup() {
  pinMode(LED_RED, OUTPUT);
  pinMode(LED_GREEN, OUTPUT);
  pinMode(LED_BLUE, OUTPUT);
  pinMode(LED_BUILTIN, OUTPUT);
  digitalWrite(LED_RED, LOW);
  digitalWrite(LED_GREEN, LOW);
  digitalWrite(LED_BLUE, LOW);
  digitalWrite(LED_BUILTIN, LOW);

  Serial.begin(115200);
  Serial.println();
  Serial.print("Configuring access point...");

  WiFi.begin(ssid, password);

  while(WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
    digitalWrite(LED_BUILTIN, !digitalRead(LED_BUILTIN));
  }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());

  digitalWrite(LED_BUILTIN, HIGH);

  udpSocket.begin(localPort);
}
```

Snippet codice (3)

```
void loop() {
  char JSONMessage[255];

  int packetSize = udpSocket.parsePacket();

  if (packetSize) {
    IPAddress remoteIp = udpSocket.remoteIP();

    int len = udpSocket.read(packetBuffer, 255);
    if (len > 0) {
      packetBuffer[len] = 0;
    }

    strcpy(JSONMessage, packetBuffer);

    if(debug) {
      Serial.print("Received packet of size ");
      Serial.println(packetSize);
      Serial.print("From ");
      Serial.print(remoteIp);
      Serial.print(", port ");
      Serial.println(udpSocket.remotePort());
      Serial.println("Contents:");
      Serial.println(packetBuffer);
    }

    parsePacket(JSONMessage);
  }
}
```

Snippet codice (4)

```
void commandWhois(JsonObject &json) {
    StaticJsonBuffer<200> jsonBuffer;
    JsonObject& root = jsonBuffer.createObject();
    root["command"] = "whois";
    root["name"] = "MOODLAMP";
    root["r"] = red;
    root["g"] = green;
    root["b"] = blue;

    String toSend;
    root.printTo(toSend);

    udpSocket.beginPacket(udpSocket.remoteIP(), localPort);
    udpSocket.print(toSend);
    udpSocket.endPacket();

    digitalWrite(LED_BUILTIN, LOW);
    delay(500);
    digitalWrite(LED_BUILTIN, HIGH);
}
```

```
analogWrite(LED_RED, 1023*red/255);
analogWrite(LED_GREEN, 1023*green/255);
analogWrite(LED_BLUE, 1023*blue/255);
```



**Grazie per
l'attenzione!**