Bringe den Clown zum Jonglieren

Verbinde Dich am Smartphone oder Computer mit dem Clown über WLAN.

* SSID = "CoDo-Clown"

* PASSWORT = "12345678"

Es wird angezeigt, dass keine Internetverbindung besteht. Das ist so in Ordnung.

Starte einen Web-Browser (Firefox, MS-Edge…)

Gib die IP Adresse im Eingabefeld ein: **192.168.4.1** (ohne https://)

Der Clown meldet sich mit seiner Eingabeseite:

ESP8266 Access Point zur Steuerung vom Electronic-Clown

Gib den Bällen eine Farbe.

Ball 1 = AUS GELB BLAU GRUEN ROT

Ball 2 = AUS GELB BLAU GRUEN ROT

Clown = START STOP Jonglieren.

Kommando: STOP

Aufrufzähler=1
Günther Ehrenberger Sept2019

Mit den Buttons kann die Farbe der beiden Bälle in seinen Händen ausgewählt werden.

Mit „Start“ beginnt der Clown zu jonglieren, mit „Stop“ beendet der Clown das Jonglieren.

Bauteile:


Spannungsregler 5V - 3,3V AMS1117

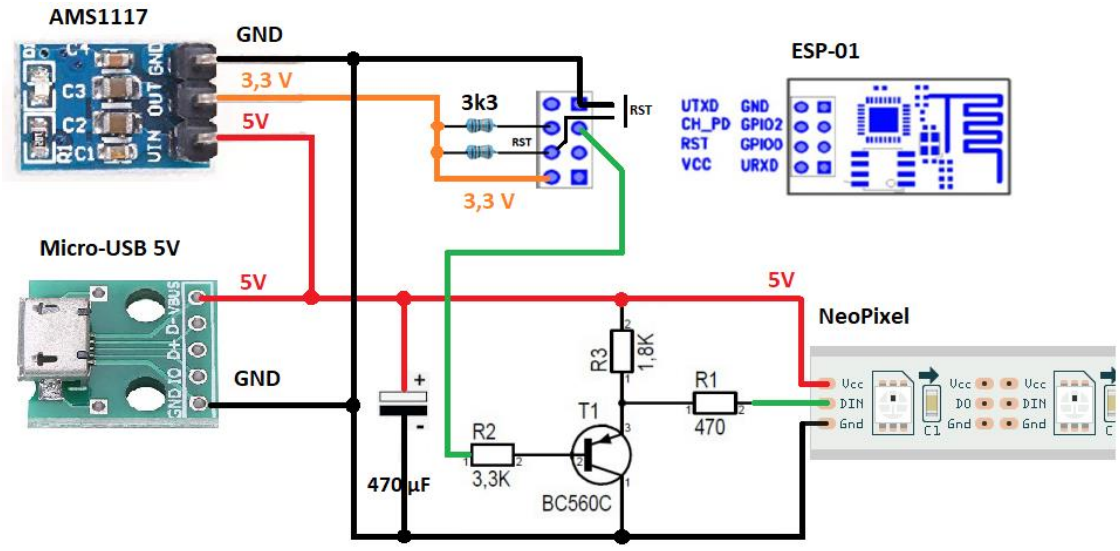

ESP-01

Elko 1x 470 µF


Widerstände 3x 100E, 1x470E, 1x1k8. 2x3k3.

- Clown Bild vom Electronicum: 3x 100 Ohm, 1x Blink-LED, 9x NeoPixel LED WS2812
- Steuerung: ESP-01, 2x 3k3
- Pegelwandler: 1x 3k3, 1x 1k8, 1x 470E, 1x BC560 (PNP), 1x 470µF Elko
- Spannungswandler 5V / 3,3V: AMS1117 und Micro-USB Steckboard.

Schaltplan:



Software-Installation:

- Lese zuerst die ESP-01 Schnellstartanleitung und die LED-Bilder Anleitung (LED-Array)
- Lade das Arduino-Beispiel „Web-AP-Clown“ mit einem ESP-01 Programmer

Spannungsversorgung:

Laut Datenblatt benötigt eine LED 20mA. Da bei der WS2812 drei verschiedenfarbige LEDs in einem Gehäuse eingebaut sind, benötigen wir 60mA pro Pixel. Bei 9 Stück Pixel sind das maximal 540 mA. Somit wird es mit einem USB Stecker-Netzteil mit 5V und 1A funktionieren.

Programm-Code

```
/*-----------------------------------------------
Accesspoint auf ESP8266
ESP01 zum Steuern des CoderDojo Clowns

Arduino, ESP01, WeMos D1 etc.
Achtung auf Doppelbelegung von TX, LED_BUILTIN, NeoPin

Vorlage ESP_WebServerAP
192.168.4.1

Rev1. 21.Aug.2019 Guenther Ehrenberger
-----------------------------------------------*/
// Neopixel
#include <Adafruit_NeoPixel.h>
//#include <NeoPixelBus.h>
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <ESP8266mDNS.h>

const char* ssid = "CoDo-Clown";
const char* password = "12345678";  // set to "" for open access point w/o passwortd

unsigned long ulReqcount;
bool bo_Start;


// Create an instance of the server on Port 80
WiFiServer server(80);
IPAddress ip(192, 168, 4, 1);      // fix IP of the server
IPAddress gateway(192, 168, 4, 1);  // WiFi router's IP
IPAddress subnet(255, 255, 255, 0);

// Neopixel Config
#define NeoPIN 0         // 0 fuer D1-Mini; 2 für ESP01
#define NUM_LEDS 9
int brightness = 100;
Adafruit_NeoPixel strip = Adafruit_NeoPixel(NUM_LEDS, NeoPIN, NEO_RGB + NEO_KHZ800);

// Clown-Ball-Position
int BALL_1 = 7;
int BALL_2 = 2;

uint32_t myRED = strip.ColorHSV(21845, 255, brightness);
uint32_t myBLUE = strip.ColorHSV(43690,255, brightness);
uint32_t myGREEN = strip.ColorHSV(0,255, brightness);
uint32_t myYELLOW = strip.ColorHSV(10922,255, brightness);
uint32_t myCYAN = strip.ColorHSV(32768,255, brightness);
uint32_t myMAGENTA = strip.ColorHSV(55000,255, brightness);
```

```
void setup()
{
  // start serial
  Serial.begin(115200);
  delay(10);

  // prepare GPIO2 on D1-Mini / GPIO1 on ESP01
  //pinMode(LED_BUILTIN, OUTPUT);
  //digitalWrite(LED_BUILTIN, 1);

  // NeoPixel start
  Serial.println();
  //strip.setBrightness(brightness);  // wenn getPixelColor verwendet wird, dann diese Zeile ausblenden
  strip.begin();
  strip.show();
  strip.fill((0,0,0));
  strip.show();
  bo_Start = false;
  Serial.println("NeoPixel started");
  delay(50);

  // setup globals
  ulReqcount=0;

  // AP mode
  WiFi.mode(WIFI_AP);
  WiFi.softAP(ssid, password);
  delay(2000);
  Serial.println();
  Serial.println("WiFi Connected");
  Serial.print("IP address: ");
  Serial.println(WiFi.softAPIP());
  delay(500);
  server.begin();
  Serial.println("Server Started");

}

// Die Pixelkette mit einer Farbe füllen
void setLedLine(uint32_t color ){
  for(uint16_t i=0; i<NUM_LEDS; i++) { //Fuer jeden NeoPixel
     strip.setPixelColor(i, color);
  }
}
```

```
void loop()
{
  // Check if a client has connected
  WiFiClient client = server.available();
  if (!client)
  {
    //digitalWrite(LED_BUILTIN, LOW);
    delay(150);
    //digitalWrite(LED_BUILTIN, HIGH);
    delay(150);
    if (bo_Start){
      // laufen der NeoPixel um eins
      uint32_t pixColor = strip.getPixelColor(NUM_LEDS-1);  // Farbe vom letzten Led speichern
      for(uint16_t i=(NUM_LEDS-1); i>0; i--) {
        //Serial.println(strip.getPixelColor(i-1));
        strip.setPixelColor(i, strip.getPixelColor(i-1));
      }
      strip.setPixelColor(0, pixColor);  // Farbe vom letzten Led auf das erste Led setzen
      strip.show();
    }
    return;
  }

  // Wait until the client sends some data
  Serial.println("new client");
  unsigned long ultimeout = millis()+250;
  while(!client.available() && (millis()<ultimeout) )
  {
    delay(1);
  }
  if(millis()>ultimeout)
  {
    Serial.println("client connection time-out!");
    return;
  }

  // Read the first line of the request
  String sRequest = client.readStringUntil('\r');
  //Serial.println(sRequest);
  client.flush();

  // stop client, if request is empty
  if(sRequest=="")
  {
    Serial.println("empty request! - stopping client");
    client.stop();
    return;
  }

```

```
  // get path; end of path is either space or ?
  // Syntax is e.g. GET /?pin=MOTOR1STOP HTTP/1.1
  String sPath="",sParam="", sCmd="";
  String sGetstart="GET ";
  int iStart,iEndSpace,iEndQuest;
  iStart = sRequest.indexOf(sGetstart);
  if (iStart>=0)
  {
    iStart+=+sGetstart.length();
    iEndSpace = sRequest.indexOf(" ",iStart);
    iEndQuest = sRequest.indexOf("?",iStart);

    // are there parameters?
    if(iEndSpace>0)
    {
      if(iEndQuest>0)
      {
        // there are parameters
        sPath  = sRequest.substring(iStart,iEndQuest);
        sParam = sRequest.substring(iEndQuest,iEndSpace);
      }
      else
      {
        // NO parameters
        sPath  = sRequest.substring(iStart,iEndSpace);
      }
    }
  }

  ////////////////////////////////////////////////////////////////////////////
  // output parameters to serial, you may connect e.g. an Arduino and react on it
  ////////////////////////////////////////////////////////////////////////////
  if(sParam.length()>0)
  {
    int iEqu=sParam.indexOf("=");
    if(iEqu>=0)
    {
      sCmd = sParam.substring(iEqu+1,sParam.length());
      Serial.println(sCmd);
    }
  }


  ////////////////////////////
  // format the html response
  ////////////////////////////
  String sResponse,sHeader;

  ////////////////////////////
  // 404 for non-matching path
  ////////////////////////////
```

```
  if(sPath!="/")
  {
    sResponse="<html><head><title>404 Not Found</title></head><body><h1>Not Found</h1><p>The requested URL was not found on this server.</p></body></html>";

    sHeader  = "HTTP/1.1 404 Not found\r\n";
    sHeader += "Content-Length: ";
    sHeader += sResponse.length();
    sHeader += "\r\n";
    sHeader += "Content-Type: text/html\r\n";
    sHeader += "Connection: close\r\n";
    sHeader += "\r\n";
  }
```

```
  //////////////////////
  // format the html page
  //////////////////////
  else
  {
    ulReqcount++;
    sResponse  = "<html><head><title>Electronic Clown</title></head><body>";
    sResponse += "<font color=\"#000000\"><body bgcolor=\"#d0d0f0\">";
    sResponse += "<meta name=\"viewport\" content=\"width=device-width, initial-scale=1.0, user-scalable=yes\">";
    sResponse += "<h1>ESP8266 Access Point zur Steuerung vom CoderDojo<br />Electronic-Clown</h1>";
    sResponse += "Gib den B&auml;llen eine Farbe.<BR>";
    sResponse += "<BR>";
    //sResponse += "<FONT SIZE=+1>";
    sResponse += "<p>Ball 1 = <a href=\"?arg=BALL_1_OFF\"><button>AUS</button></a> <a href=\"?arg=BALL_1_YELLOW\"><button>GELB</button>"
    sResponse += "</a> <a href=\"?arg=BALL_1_BLUE\"><button>BLAU</button></a> <a href=\"?arg=BALL_1_GREEN\"><button>GRUEN</button>"
    sResponse += "</a> <a href=\"?arg=BALL_1_RED\"><button>ROT</button></a></p>";
    sResponse += "<p>Ball 2 = <a href=\"?arg=BALL_2_OFF\"><button>AUS</button></a> <a href=\"?arg=BALL_2_YELLOW\"><button>GELB</button>"
    sResponse += "</a> <a href=\"?arg=BALL_2_BLUE\"><button>BLAU</button></a> <a href=\"?arg=BALL_2_GREEN\"><button>GRUEN</button>"
    sResponse += "</a> <a href=\"?arg=BALL_2_RED\"><button>ROT</button></a></p>";
    sResponse += "<p>Clown  = <a href=\"?arg=START\"><button>START</button></a> <a href=\"?arg=STOP\"><button>STOP</button></a>   Jonglieren.</p>";

    //////////////////////
    // react on parameters
    //////////////////////
    if (sCmd.length()>0)
    {
      // write received command to html page
      sResponse += "Kommando:  " + sCmd + "<BR>";
      bo_Start = false;

      // switch GPIO
      if(sCmd.indexOf("BALL_1_OFF")>=0)
      {
        //setLedLine(strip.Color(0,0,0));  //  RGB Farbe Schwarz
        strip.setPixelColor(BALL_1, strip.Color(0,0,0));
        strip.show();
      }
      else if(sCmd.indexOf("BALL_1_YELLOW")>=0)
      {
        strip.setPixelColor(BALL_1, myYELLOW );
        strip.show();
      }
      else if(sCmd.indexOf("BALL_1_BLUE")>=0)
      {
        strip.setPixelColor(BALL_1, myBLUE);
        strip.show();
      }
      else if(sCmd.indexOf("BALL_1_GREEN")>=0)
      {
        strip.setPixelColor(BALL_1, myGREEN);
        strip.show();
      }
```

```
     else if(sCmd.indexOf("BALL_1_RED")>=0)
      {
        // setLedLine(strip.Color(0,255,0));  //  RGB Farbe ROT
        strip.setPixelColor(BALL_1, myRED);
        strip.show();
      }
      else if(sCmd.indexOf("BALL_2_OFF")>=0)
      {
        strip.setPixelColor(BALL_2, strip.Color(0,0,0));
        strip.show();
       // digitalWrite(LED_BUILTIN, 1);
      }
      else if(sCmd.indexOf("BALL_2_YELLOW")>=0)
      {
        strip.setPixelColor(BALL_2, myYELLOW);
        strip.show();
      }
      else if(sCmd.indexOf("BALL_2_BLUE")>=0)
      {
        strip.setPixelColor(BALL_2, myBLUE);
        strip.show();
      }
      else if(sCmd.indexOf("BALL_2_GREEN")>=0)
      {
        strip.setPixelColor(BALL_2, myGREEN);
        strip.show();
      }
      else if(sCmd.indexOf("BALL_2_RED")>=0)
      {
        strip.setPixelColor(BALL_2, myRED);
        strip.show();
      }
      else if(sCmd.indexOf("START")>=0)
      {
        bo_Start = true;
        strip.show();
      }
      else if(sCmd.indexOf("STOP")>=0)
      {
        bo_Start = false;
        //setLedLine(strip.Color(0,0,0));  //  RGB Farbe Schwarz
        strip.fill((0,0,0));
        strip.show();
      }
    }
```

```

    sResponse += "<FONT SIZE=-2>";
    sResponse += "<BR>Aufrufz&auml;hler=";
    sResponse += ulReqcount;
    sResponse += "<BR>";
    sResponse += "G&uuml;nther Ehrenberger Sept2019<BR>";
    sResponse += "</body></html>";

    sHeader  = "HTTP/1.1 200 OK\r\n";
    sHeader += "Content-Length: ";
    sHeader += sResponse.length();
    sHeader += "\r\n";
    sHeader += "Content-Type: text/html\r\n";
    sHeader += "Connection: close\r\n";
    sHeader += "\r\n";
  }

  // Send the response to the client
  client.print(sHeader);
  client.print(sResponse);

  // and stop the client
  client.stop();
  Serial.println("Client disonnected");
}

```