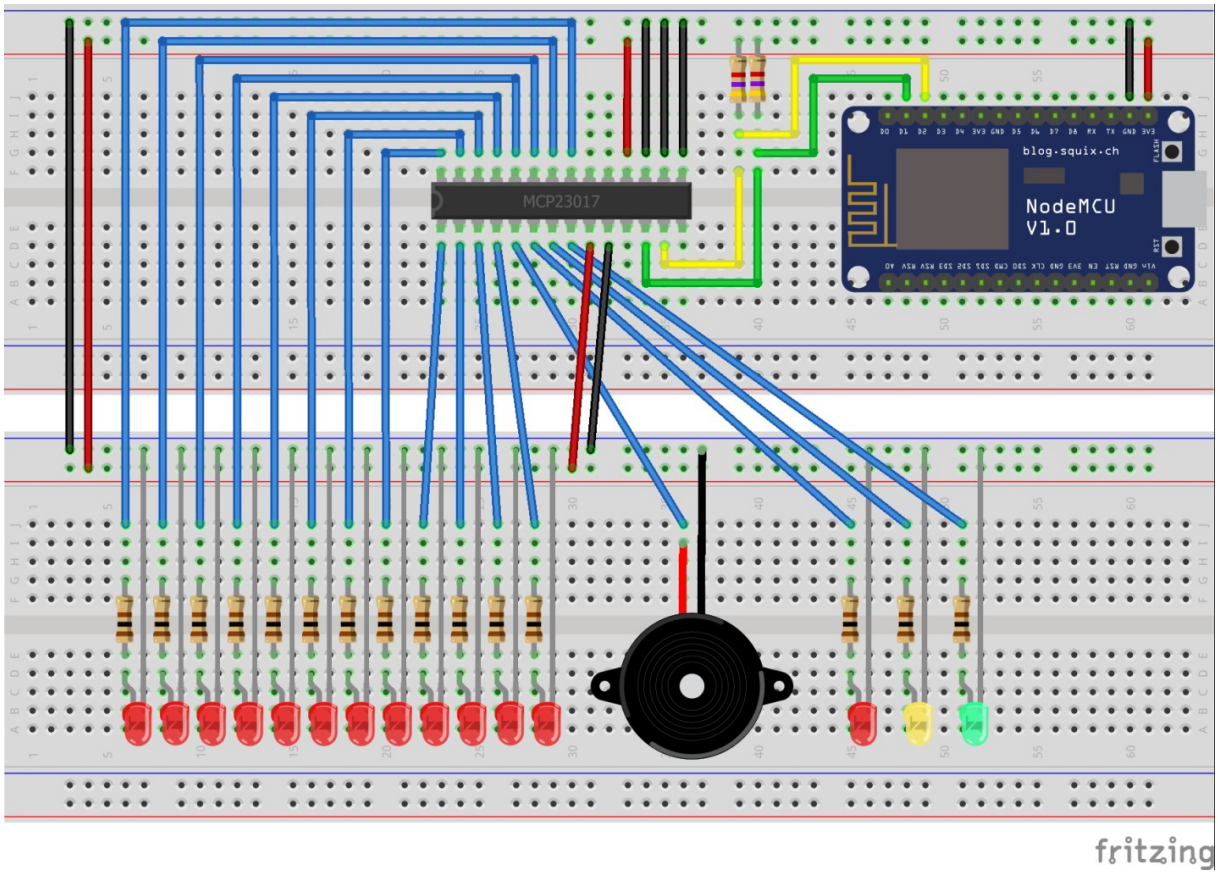
Projekt - Statusanzeige der LupusXT mit dem ESP8266

Schaltbild:



Stückliste (Bestellung bei www. reichelt.de)

Artikel-Nr.	Bezeichnung	WG	Menge	Einzelpreis incl. MwSt.	Gesamtpreis incl. MwSt.
SUMMER CPM 121	Piezosummer	1	1	1,15 €	1,15 €
MCP 23017-E/SP	16bit Input-Outport Port Expander PDIP-28	1	1	1,15 €	1,15 €
DEBO JT ESP8266	Entwicklerboards - NodeMCU ESP8266 WiFi-Modul	0	1	8,99 €	8,99 €
LED 5MM GE	LED, 5 mm, bedrahtet, gelb, 18 mcd, 60Â°	1	1	0,070 €	0,07 €
LED 5MM GN	LED, 5 mm, bedrahtet, grÃ¼n, 18 mcd, 60Â°	1	1	0,060 €	0,06 €
LED 5MM RT	LED, 5 mm, bedrahtet, rot, 4,5 mcd, 60Â°	1	13	0,060 €	0,77 €
METALL 100	Widerstand, Metallschicht, 100 Ohm, 0207, 0,6 W, 1%	1	15	0,049 €	0,73 €
METALL 4,70K	Widerstand, Metallschicht, 4,70 kOhm, 0207, 0,6 W, 1%	1	2	0,082 €	0,16 €
Warenwert					13,09 €
Zuzüglich Versandkosten innerhalb Deutschlands für ca. 0,0 kg					5,60 €

Arduino Sketch

```
#include <SPI.h>
#include <Wire.h>
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include <Adafruit_MCP23017.h>

const char* ssid = "Your_WLAN_SSID";
const char* password = "Your_WLAN_KEY";
IPAddress ip(192, 168, 1, 100);
IPAddress gateway(192, 168, 1, 254);
IPAddress subnet(255, 255, 255, 0);

unsigned long ulReqcount;
unsigned long ulReconncount;

// Create an instance of the server on Port 80
// WiFiServer server(80);
ESP8266WebServer server(80);

Adafruit_MCP23017 mcp1; // Create MCP 1

void setup() {
  // setup globals
  ulReqcount=0;
  ulReconncount=0;

  // start serial
  Serial.begin(9600);
  delay(1);

  // inital connect
  WiFi.mode(WIFI_STA);
  WiFiStart();

  mcp1.begin(0);      // Start MCP 1 on Hardware address 0x20
  for (int i=0; i<=15; i++) mcp1.pinMode(i, OUTPUT); // Define GPA0-GPB7 on MCP1 as Output

}

void WiFiStart()
{
  ulReconncount++;

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

  WiFi.begin(ssid, password);
  WiFi.config(ip, gateway, subnet);

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

```
  Serial.println("");
  Serial.println("WiFi connected");

  // http://x.x.x.x/Lupus?XT=yyy
  server.on("/Lupus", handleLupus);    //Associate the handler function to the path

  // Start the server
  server.begin();
  Serial.println("Web-Server started");

  // Print the IP address
  Serial.println(WiFi.localIP());
  Serial.println(WiFi.subnetMask());
  Serial.println(WiFi.gatewayIP());

}

void loop() {
  // put your main code here, to run repeatedly:
  server.handleClient();                  //Handling of incoming requests

}

void handleLupus() { //Handler
  String message = "Lupus XT Status-Display";
  if (server.argName(0) == "XT") {
    String XT = server.arg(0);
    message += "\nStatusuebermittlung: " + XT;
    int stat = XT.toInt();
    switch (stat) {
    case (505): //Verzoegerung E+A (LED 12+13+15)
      mcp1.digitalWrite(12, HIGH);
      mcp1.digitalWrite(13, HIGH);
      mcp1.digitalWrite(14, LOW);
      mcp1.digitalWrite(15, HIGH);
      break;
    case (504): //Arm (LED 13)
      mcp1.digitalWrite(12, LOW);
      mcp1.digitalWrite(13, HIGH);
      mcp1.digitalWrite(14, LOW);
      mcp1.digitalWrite(15, LOW);
      break;
    case (501): //Disarm (LED 15)
      mcp1.digitalWrite(12, LOW);
      mcp1.digitalWrite(13, LOW);
      mcp1.digitalWrite(14, LOW);
      mcp1.digitalWrite(15, HIGH);
      break;
    case (506): //Home Arm 1 (LED 13+14)
      mcp1.digitalWrite(12, LOW);
      mcp1.digitalWrite(13, HIGH);
      mcp1.digitalWrite(14, HIGH);
      mcp1.digitalWrite(15, LOW);
      break;
    case (110): //KF G zu (LED 0)
      mcp1.digitalWrite(0, LOW);
```

```
  break;
case (111): //KF G offen (LED 0)
  mcp1.digitalWrite(0, HIGH);
  break;
case (120): //KF K zu (LED 1)
  mcp1.digitalWrite(1, LOW);
  break;
case (121): //KF K offen (LED 1)
  mcp1.digitalWrite(1, HIGH);
  break;
case (130): //GWC zu (LED 2)
  mcp1.digitalWrite(2, LOW);
  break;
case (131): //GWC offen (LED 2)
  mcp1.digitalWrite(2, HIGH);
  break;
case (140): //WZ K zu (LED 3)
  mcp1.digitalWrite(3, LOW);
  break;
case (141): //WZ K offen (LED 3)
  mcp1.digitalWrite(3, HIGH);
  break;
case (150): //WZ G zu (LED 4)
  mcp1.digitalWrite(4, LOW);
  break;
case (151): //WZ G offen (LED 4)
  mcp1.digitalWrite(4, HIGH);
  break;
case (210): //KZ zu (LED 5)
  mcp1.digitalWrite(5, LOW);
  break;
case (211): //KZ offen (LED 5)
  mcp1.digitalWrite(5, HIGH);
  break;
case (220): //BAD zu (LED 6)
  mcp1.digitalWrite(6, LOW);
  break;
case (221): //BAD offen (LED 6)
  mcp1.digitalWrite(6, HIGH);
  break;
case (230): //SZ L zu (LED 7)
  mcp1.digitalWrite(7, LOW);
  break;
case (231): //SZ L offen (LED 7)
  mcp1.digitalWrite(7, HIGH);
  break;
case (240): //SZ R zu (LED 8)
  mcp1.digitalWrite(8, LOW);
  break;
case (241): //SZ R offen (LED 8)
  mcp1.digitalWrite(8, HIGH);
  break;
case (310): //DE zu (LED 9)
  mcp1.digitalWrite(9, LOW);
  break;
case (311): //DE offen (LED 9)
```

```
      mcp1.digitalWrite(9, HIGH);
      break;
    case (320): //DF zu (LED 10)
      mcp1.digitalWrite(10, LOW);
      break;
    case (321): //DF offen (LED 10)
      mcp1.digitalWrite(10, HIGH);
      break;
    case (410): //HT zu (LED 11)
      mcp1.digitalWrite(11, LOW);
      break;
    case (411): //HT offen (LED 11)
      mcp1.digitalWrite(11, HIGH);
      break;
    default:
      message += "\nStatus unbekannt: " + XT;
    }
  }
  else {
    message += "\nDas XT-Argument konnte nicht gefunden werden!";
  }
  server.send(200, "text/plain", message);
  Serial.println(message);

}
```