/\*\*

\* ProLight v0.1

\* -------------

\* Der Versuch einer RGB-Led-Steuerung mit ESP8266

\*

\* Grundlegend:

\* - Namensgebung:

\* - speicherung im SPIFFS

\*

\* Bestandteile-Wlan:

\* - Wlan-Verbindung

\* - Wlan-AP zum einbinden

\* - mDNS zum auffinden im Wlan

\* - ansteuern per http-request

\*

\* Bestandteile-Led:

\* - Ein-/Ausschalten

\* - Stufenlose, statische LED-Beleuchtung

\* - Dimmer

\* - Random-Wave-Ausgabe mit Zeiteinstellung

\* - Random-Blitzer-Ausgabe mit 2-facher Zeiteinstellung (Ein-delay und Aus-Delay)

\*

\* -------------

\* Autor: Andreas Bring (andreas@abring.de)

\* -------------

\*

\* Versinierung:

\* v 0. 1. 0 - Grundstruktur angelegt, WiFi, mDNS, SPIFFS etc.

\* v 0. 1. 1 - Erste Steuerelemente per http-request:

\* - /setConfig (ssid (String), password (String), extention (String))

\* - /setColor (red (int (0 - 1023)), green (int (0 - 1023), blue(int (0 - 1023)))

\*/

// Die includes

#include <ESP8266WiFi.h> // https://github.com/bportaluri/WiFiEsp Version 2.2.2

#include <ESP8266mDNS.h> // https://arduino-esp8266.readthedocs.io/en/latest/ Version 2.5.0

#include <WiFiClient.h> // https://arduino-esp8266.readthedocs.io/en/latest/ Version 2.5.0

#include <ESP8266WebServer.h> // https://github.com/esp8266/Arduino/tree/master/libraries/ESP8266WebServer/

#include <FS.h> // https://arduino-esp8266.readthedocs.io/en/latest/ Version 2.5.0

#include <WebSocketsServer.h> // https://github.com/Links2004/arduinoWebSockets Version 2.1.1

#include <math.h> // just in case

#include <ESP8266HTTPUpdateServer.h>

// definition der Ausgänge

#define ledR 12 // Rot liegt an Ausgang 12

#define ledG 14 // Grün liegt an Ausgang 4

#define ledB 4 // Blau liegt an Ausgang 14

// define debug-Mode

static const bool debug **=** **true;** // prüfen wir oder prüfen wir nicht, das ist hier die Frage

// definition der globalen Variablen

String deviceName **=** "ProLight"**;** // setze den Namen

String deviceVersion **=** "0.1.2"**;** // setze die Versionsnummer

String apPassword **=** "vierzigzwei"**;** // das Passwort für den AccessPoint

const char**\*** username **=** "admin"**;** // Webupdate: Username

const char**\*** update\_path **=** "/firmware"**;** // Adresse zum Updaten http://deineipadresse/update\_path

bool powerOn **=** **false;** // standartmäßig Ausgeschaltet

unsigned int dimmer **=** 1024**;** // standartmäßig auf 100% (0 = 0%, 1024 = 100%)

const byte stat **=** 0**;** // statische Beleuchtung

const byte wave **=** 1**;** // radom-Wave-Ausgabe

const byte stro **=** 2**;** // random-Blitz-Ausgabe

byte currentLedMode **=** stat**;** // standartmäßig setze den Modus auf statisch

unsigned long delayLoop **=** 25**;** // standartmäßig 25 ms Pause im Loop

unsigned long delayStat **=** 200**;** // standartmäßig 200 ms Pause im statischen-Betrieb

unsigned long delayWave **=** 1000**;** // standartmäßig 1000 ms = 1 s Pause im wave-Betrieb

unsigned long delayStroOn **=** 100**;** // standartmäßig 100 ms Pause im einschalten des strobo-Betriebs

unsigned long delayStroOff **=** 100**;** // standartmäßig 100 ms Pause im einschalten des strobo-Betriebs

unsigned long statEndMillis **=** 0**;** // standartmäßig 0 ms bis zum nächsten modeLoop

unsigned long waveEndMillis **=** 0**;** // standartmäßig 0 ms bis zum nächsten modeLoop

unsigned long stroEndMillis **=** 0**;** // standartmäßig 0 ms bis zum nächsten modeLoop

bool randomColor **=** **false;** // standartmäßig wird keine neue Farbe gesetzt

unsigned int color**[**3**]** **=** **{**244**,** 164**,** 96**};** // standartmäßig setze die Ffarbe auf "sandy brown"

bool modeChange **=** **false;** // wir befinden uns nicht in einem ModeChange

//Lade Bibliotheken

ESP8266WebServer server**(**80**);** // lade den WebServer an Port 80

MDNSResponder mdns**;** // lade den mDNS-Responder

ESP8266HTTPUpdateServer httpUpdater**;** // Konfiguriert den WebUpserver als httpUpdater

File fsUploadFile**;** // Setzt die Variable File UploadFile

/\*\*

\* ------------------------------------------------------ Das Setup ... ------------------------------------------------------

\* Debug-Modus

\* OUTPUTs

\*/

void setup**()** **{**

//starte debug, wenn gewünscht

**if** **(**debug**)** **{**

Serial**.**begin**(**115200**);**

Serial**.**println**();**

Serial**.**println**();**

Serial**.**println**(**"Debug-Mode gestartet:"**);**

Serial**.**println**(**"---------------------"**);**

Serial**.**println**();**

**}**

//define OUTPUTs

**if** **(**debug**)** Serial**.**println**(**"Setze OUTPUTs"**);**

pinMode**(**ledR**,** OUTPUT**);**

pinMode**(**ledG**,** OUTPUT**);**

pinMode**(**ledB**,** OUTPUT**);**

pinMode**(**LED\_BUILTIN**,** OUTPUT**);**

//starte File-System SPIFFS

**if** **(**debug**)** Serial**.**println**(**"Starte SPIFFS"**);**

SPIFFS**.**begin**();**

//lade Config, falls möglich

**if** **(**debug**)** Serial**.**println**(**"Lade Config"**);**

File file**;**

String ssid **=** ""**;**

String password **=** ""**;**

String deviseNameExtention **=** ""**;**

file **=** SPIFFS**.**open**(**"/config/ssid.txt"**,** "r"**);**

**if** **(**file**)** **{**

ssid **=** file**.**readString**();**

**if** **(**debug**)** Serial**.**println**(**"SSID geladen: " **+** ssid**);**

**}** **else** **{**

**if** **(**debug**)** Serial**.**println**(**"SSID nicht geladen!"**);**

**}**

file**.**close**();**

file **=** SPIFFS**.**open**(**"/config/password.txt"**,** "r"**);**

**if** **(**file**)** **{**

password **=** file**.**readString**();**

**if** **(**debug**)** Serial**.**println**(**"Passwort geladen: " **+** password**);**

**}** **else** **{**

**if** **(**debug**)** Serial**.**println**(**"Passwort nicht geladen!"**);**

**}**

file**.**close**();**

file **=** SPIFFS**.**open**(**"/config/extention.txt"**,** "r"**);**

**if** **(**file**)** **{**

deviceName **+=** "\_" **+** file**.**readString**();**

**if** **(**debug**)** Serial**.**println**(**"Namensextention geladen!"**);**

**}** **else** **{**

deviceName **+=** "\_" **+** WiFi**.**macAddress**();**

**if** **(**debug**)** Serial**.**println**(**"Namensextention nicht geladen!"**);**

**}**

file**.**close**();**

**if** **(**debug**)** Serial**.**println**(**"Der Name lautet: " **+** deviceName**);**

//starte Wlan

**if** **(**debug**)** **{**

Serial**.**println**(**"Starte WLan"**);**

Serial**.**println**(**"---------------------"**);**

Serial**.**print**(**"Verbinde "**);**

**}**

IPAddress ipAddress**;**

WiFi**.**begin**(**ssid**.**c\_str**(),** password**.**c\_str**());**

byte count **=** 0**;**

**while(**WiFi**.**status**()** **!=** WL\_CONNECTED **&&** count **<** 25**)** **{**

**if** **(**debug**)** Serial**.**print**(**"."**);**

digitalWrite**(**LED\_BUILTIN**,** LOW**);**

delay**(**10**);**

digitalWrite**(**LED\_BUILTIN**,** HIGH**);**

delay**(**490**);**

count**++;**

**}**

Serial**.**println**();**

**if** **(**WiFi**.**status**()** **==** WL\_CONNECTED**)** **{**

digitalWrite**(**LED\_BUILTIN**,** LOW**);**

delay**(**10**);**

digitalWrite**(**LED\_BUILTIN**,** HIGH**);**

delay**(**240**);**

digitalWrite**(**LED\_BUILTIN**,** LOW**);**

delay**(**10**);**

digitalWrite**(**LED\_BUILTIN**,** HIGH**);**

ipAddress **=** WiFi**.**localIP**();**

**if** **(**debug**)** **{**

Serial**.**println**(**"Mit Netzwerk verbunden: " **+** ssid**);**

Serial**.**print**(**"Signalstaerke: "**);**

Serial**.**print**(**WiFi**.**RSSI**());**

Serial**.**println**(**" dBm"**);**

Serial**.**print**(**"IP-Adresse: "**);**

Serial**.**println**(**WiFi**.**localIP**());**

**}**

**}** **else** **{**

digitalWrite**(**LED\_BUILTIN**,** LOW**);**

delay**(**1000**);**

digitalWrite**(**LED\_BUILTIN**,** HIGH**);**

**if** **(**debug**)** **{**

Serial**.**println**(**"Netzwerkverbindung nicht möglich."**);**

Serial**.**println**(**"Starte AccessPoint: " **+** deviceName**);**

Serial**.**println**(**"Password : " **+** apPassword**);**

**}**

**if** **(**debug**)** Serial**.**println**(**"Starte AP."**);**

bool wifiEstablished **=** WiFi**.**softAP**(**deviceName**.**c\_str**(),** apPassword**.**c\_str**());**

**if** **(**wifiEstablished**)** **{**

ipAddress **=** WiFi**.**softAPIP**();**

**if** **(**debug**)** **{**

Serial**.**print**(**"IP-Adresse : "**);**

Serial**.**println**(**ipAddress**);**

**}**

**}** **else** **{**

**if** **(**debug**)** **{**

Serial**.**println**(**"AccessPoint konnte nicht gestartet werden!"**);**

Serial**.**println**(**"Versuchts nochmal, ich gehe schlafen."**);**

Serial**.**println**(**"Gute Nacht !!!"**);**

**}**

ESP**.**deepSleep**(**0**);**

**}**

**}**

//starte mDNS

**if** **(**debug**)** Serial**.**println**(**"Starte mDNS"**);**

**if** **(**mdns**.**begin**(**deviceName**.**c\_str**(),** ipAddress**))** **{**

mdns**.**addService**(**"http"**,** "tcp"**,** 80**);**

**if** **(**debug**)** Serial**.**println**(**"mDNS aktiv an: " **+** deviceName**);**

**}** **else** **{**

**if** **(**debug**)** **{**

Serial**.**println**(**"mDNS konnte nicht gestartet werden!"**);**

Serial**.**println**(**"Versuchts nochmal, ich gehe schlafen."**);**

Serial**.**println**(**"Gute Nacht !!!"**);**

**}**

ESP**.**deepSleep**(**0**);**

**}**

//starte Webserver

**if** **(**debug**)** Serial**.**println**(**"Starte Webserver"**);**

server**.**onNotFound**(**handleNotFound**);**

server**.**on**(**"/setColor"**,** setColor**);**

server**.**on**(**"/setConfig"**,** setConfig**);**

server**.**on**(**"/doRestart"**,** doRestart**);**

server**.**begin**();**

**if** **(**debug**)** Serial**.**println**(**"Webserver aktiv."**);**

//starte Update-Service

**if** **(**debug**)** Serial**.**println**(**"Starte Update-Service"**);**

httpUpdater**.**setup**(&**server**,** update\_path**,** username**,** apPassword**.**c\_str**());** // Webupdate starten

**if** **(**debug**)** **{**

Serial**.**print**(**"Update-Service aktiv: http://"**);**

Serial**.**print**(**ipAddress**);**

Serial**.**println**(**update\_path**);**

**}**

//geschafft ...

**if** **(**debug**)** **{**

Serial**.**println**(**"---------------------"**);**

Serial**.**println**(**"Geschafft !!!"**);**

Serial**.**println**(**"Alle Dienste gestartet..."**);**

Serial**.**println**(**"---------------------"**);**

Serial**.**println**();**

**}**

**}**

/\*\*

\* ------------------------------------------------------ Der Main-Loop ------------------------------------------------------

\*/

void loop**()** **{**

unsigned long loopStartMillis **=** millis**();**

server**.**handleClient**();**

**switch** **(**currentLedMode**)** **{**

**case** stat**:**

**if** **(**modeChange **||** loopStartMillis **<** statEndMillis**)** **{**

statEndMillis **=** loopStartMillis **+** delayStat**;**

**}**

writeColor**(**color**[**0**],** color**[**1**],** color**[**2**]);**

**break;**

**case** wave**:**

**if** **(**modeChange **||** loopStartMillis **<** waveEndMillis**)** **{**

waveEndMillis **=** loopStartMillis **+** delayWave**;**

**}**

**break;**

**case** stro**:**

**if** **(**modeChange **||** loopStartMillis **<** stroEndMillis**)** **{**

stroEndMillis **=** loopStartMillis **+** delayStroOn **+** delayStroOff**;**

**}**

**break;**

**}**

**}**

/\*\*

\* ------------------------------------------------------ Die Serveranfragen -------------------------------------------------

\*/

void handleNotFound**()** **{**

**if** **(**debug**)** Serial**.**print**(**"/handleNotFound: nf"**);**

server**.**send**(**200**,** "text/plain"**,** "nf"**);**

**}**

void setColor**()** **{**

**if** **(**debug**)** Serial**.**print**(**"/setColor:"**);**

**if** **(**server**.**hasArg**(**"red"**))** **{**

color**[**0**]** **=** server**.**arg**(**"red"**).**toInt**();**

**if** **(**debug**)** **{**

Serial**.**print**(**" red: "**);**

Serial**.**print**(**color**[**0**]);**

**}**

**}**

**if** **(**server**.**hasArg**(**" green"**))** **{**

color**[**1**]** **=** server**.**arg**(**"green"**).**toInt**();**

**if** **(**debug**)** **{**

Serial**.**print**(**"green: "**);**

Serial**.**print**(**color**[**1**]);**

**}**

**}**

**if** **(**server**.**hasArg**(**" blue"**))** **{**

color**[**2**]** **=** server**.**arg**(**"blue"**).**toInt**();**

**if** **(**debug**)** **{**

Serial**.**print**(**"blue: "**);**

Serial**.**print**(**color**[**2**]);**

**}**

**}**

server**.**send**(**200**,** "text/plain"**,** "ok"**);**

**if** **(**debug**)** Serial**.**println**(**" ok"**);**

**}**

void setConfig**()** **{**

**if** **(**debug**)** Serial**.**print**(**"/setConfig: "**);**

**if** **(**server**.**hasArg**(**"username"**)** **&&**

server**.**hasArg**(**"password"**)** **&&**

server**.**arg**(**"username"**)** **==** username **&&**

server**.**arg**(**"password"**)** **==** apPassword**)** **{**

**if** **(**server**.**hasArg**(**"ssid"**))** **{**

**if** **(**writeToFile**(**"ssid.txt"**,** server**.**arg**(**"ssid"**)))** **{**

**if** **(**debug**)** Serial**.**println**(**"SSID written."**);**

**}** **else** **{**

**if** **(**debug**)** Serial**.**println**(**"SSID writing failed."**);**

**}**

**}**

**if** **(**server**.**hasArg**(**"password"**))** **{**

**if** **(**writeToFile**(**"password.txt"**,** server**.**arg**(**"password"**)))** **{**

**if** **(**debug**)** Serial**.**println**(**"Password written."**);**

**}** **else** **{**

**if** **(**debug**)** Serial**.**println**(**"Password writing failed."**);**

**}**

**}**

**if** **(**server**.**hasArg**(**"extention"**))** **{**

**if** **(**writeToFile**(**"extention.txt"**,** server**.**arg**(**"extention"**)))** **{**

**if** **(**debug**)** Serial**.**println**(**"Extention written."**);**

**}** **else** **{**

**if** **(**debug**)** Serial**.**println**(**"Extention writing failed."**);**

**}**

**}**

server**.**send**(**200**,** "text/plain"**,** "ok"**);**

**if** **(**debug**)** Serial**.**println**(**"ok"**);**

**}** **else** **{**

server**.**send**(**200**,** "text/plain"**,** "np"**);**

**if** **(**debug**)** Serial**.**println**(**"np"**);**

**}**

**}**

void doRestart**()** **{**

**if** **(**debug**)** Serial**.**print**(**"/doRestart: "**);**

**if** **(**server**.**hasArg**(**"username"**)** **&&**

server**.**hasArg**(**"password"**)** **&&**

server**.**arg**(**"username"**)** **==** username **&&**

server**.**arg**(**"password"**)** **==** apPassword**)** **{**

server**.**send**(**200**,** "text/plain"**,** "ok"**);**

**if** **(**debug**)** Serial**.**println**(**"ok"**);**

ESP**.**restart**();**

**}** **else** **{**

server**.**send**(**200**,** "text/plain"**,** "np"**);**

**if** **(**debug**)** Serial**.**println**(**"np"**);**

**}**

**}**

bool writeToFile**(**String filename**,** String data**)** **{**

File file **=** SPIFFS**.**open**(**"/config/" **+** filename**,** "w"**);**

**if** **(!**file**)** **{**

**return** **false;**

**}**

file**.**print**(**data**);**

file**.**flush**();**

file**.**close**();**

**return** **true;**

**}**

/\*\*

\* ------------------------------------------------------ Die LED-Sachen -----------------------------------------------------

\*/

void writeColor**(**unsigned int cRed**,** unsigned int cGreen**,** unsigned int cBlue**)** **{**

unsigned int red **=** round**((**cRed **\*** dimmer**)** **/** 1023.0f**);**

unsigned int green **=** round**((**cGreen **\*** dimmer**)** **/** 1023.0f**);**

unsigned int blue **=** round**((**cBlue **\*** dimmer**)** **/** 1023.0f**);**

analogWrite**(**ledR**,** red**);**

analogWrite**(**ledG**,** green**);**

analogWrite**(**ledB**,** blue**);**

**}**