

Web Server Schaltuhr umschaltbar LAN / WLAN

Passworteinabe

Manuell (Pin25) ist **An**
Schaltuhr ist **Passiv**
Schaltuhr (Pin26) ist **Aus**

[Manuell an \(Pin 25\)](#)-----[Manuell aus \(Pin 25 aus\)](#)

[Schaltuhr aktiv](#)-----[Schaltuhr passiv](#)

Einstellungen der Schaltuhr

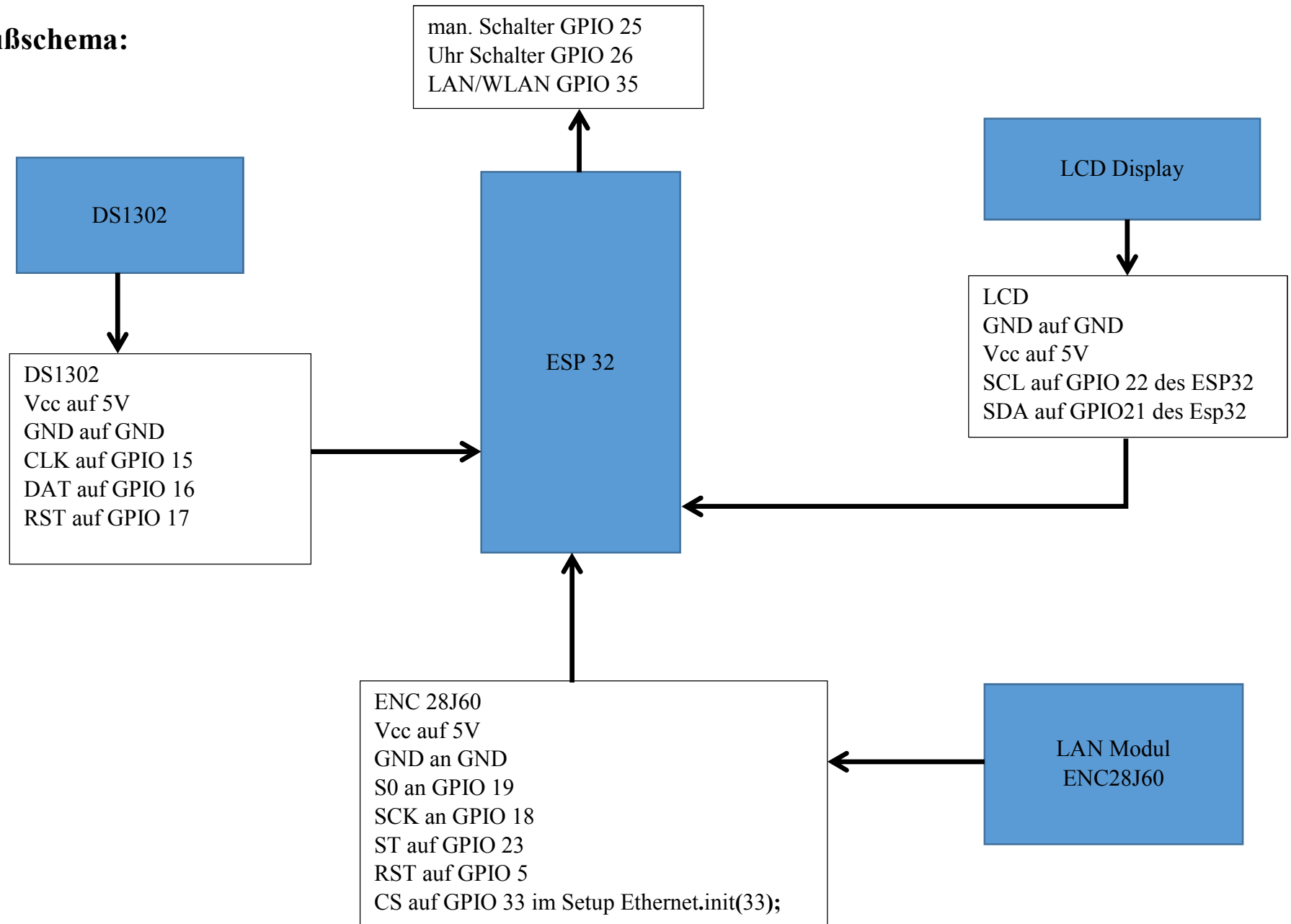
Eingabe AnfStd AnfMin EndStd EndMin jeweils 2 Stellen z.B 13201830 entspricht 13:20 bis 18:30

Uhr 1:

Uhr 2:

Nach Speichern werden alle Einstellungen resetet, Seite muss neu aufgerufen werden!
Systemdatum: 07.07.2020 10:13:47

Anschlußschema:



```
/* LAN und WLAN Variante ESP32 mit ENC28J60
 * Pin Verwendung
 * LCD
 * GND auf GND
 * Vcc auf 5V
 * SCL auf GPIO 22 des ESP32
 * SDA auf GPIO21 des Esp32
 *
 * DS1302
 * Vcc auf 5V
 * GND auf GND
 * CLK auf GPIO 15
 * DAT auf GPIO 16
 * RST auf GPIO 17
 *
 * LED manuell auf GPIO 25
 * LED Uhr auf GPIO 26
 * Speicherbelegung 56% /12% des ESP32
 * GPIO 35 als Achalter für LAN/WLAN
 *
 *ENC 28J60
 *Vcc auf 5V
 *GND an GND
 *S0 an GPIO 19
 *SCK an GPIO 18
 *ST auf GPIO 23
 *RST auf GPIO 5
 *CS auf GPIO 33 im Setup Ethernet.init(33);

 */

#include <stdio.h>
#include <UIPEthernet.h>;
#include <LiquidCrystal_I2C.h>
#include <DS1302.h>
#include <FS.h>
#include <SPIFFS.h>
#include <SPI.h>;
#include <WiFi.h>
```

```

LiquidCrystal_I2C lcd(0x27,20,4); //LCD initialisieren
DS1302 rtc(17,16,15);           //DS1302 initialisieren (RST Pin,DAT Pin,CLK Pin)

byte maclan[] = {0x90,0xA2,0xDA,0x0D,0x4E,0x80};
IPAddress iplan(192,168,1,52); //IP Adresse LAN

IPAddress ipwlan(192,168,1,53); //IP Adresse WLAN
IPAddress gatewaywlan(192,168,1,1); //Gateway WLAN
IPAddress subnetwlan(255,255,255,0); //Subnet WLAN
IPAddress dns wlan(192,168,1,1); //DNS WLAN

EthernetServer server(80);
WiFiServer serverwifi(80);

String buffer = "";
String zustand1 = "aus"; //LED22 Status aus
String zustand2 = "aus"; //LED23 Status aus
int uhr = 0; //Uhr aktiv oder passiv
int merker=0; //für Passwort Abfrage

int anfh1; //Anfangsstunde
int anfmin1; //Anfangsminute
int endh1; //Endstunde
int endmin1; //Endminute

int anfh2; //Anfangsstunde
int anfmin2; //Anfangsminute
int endh2; //Endstunde
int endmin2; //Endminute

String uhr1ges; //Gesamtstring Uhr1 Zb. 13051430 AnfSdt AnfMin EndStd EndMin
String uhr2ges; //Gesamtstring Uhr1 Zb. 18301900 AnfSdt AnfMin EndStd EndMin
String uhr1uhr2; //aus Buffer ausgelesen um in Datei zu speichern
int lcdmerker = 0;
int lanok ; //wird von GPIO 35 bestimmt GPIO35=0 dann LAN GPIO35=1 dann WLAN

void setup() { //-----void setup begin-----
//Serial.begin(115200);

zeitenholen(); //Funktion zeitenholen aus Dateien aufrufen

```

```

lcd.init();           //LCD initialisieren
lcd.setBacklight(HIGH); //LCD Hintergrund beleuchtung an

rtc.writeProtect(false); //DS1302 Schreibschutz aus
rtc.halt(false);       //DS1302 stoppen aus

//Eingabe aktuelles Datum und Zeit und Wochentag nach 1. Start auskommentieren !!!
//kSunday = So kMonday =Mo kTuesday=Die kWednesday=Mi kThursday=Do kFriday=Fr kSaturday=Sa
//Time t(2020,07,06,12,23,00, Time:: kMonday); // Datum u. Zeit Festlegen
//rtc.time(t); //im Chip speichern *****

pinMode(25, OUTPUT); // Pins auf Output setzen Schalter manuell
pinMode(26, OUTPUT); // Pins auf Output setzen Schalter Uhr

pinMode(35, INPUT); // Schalten LAN WLAN Pin35 an = WLAN

digitalWrite(25, LOW); // Pin 25 ausschalten Schalter manuell
digitalWrite(26, LOW); // Pin 25 ausschalten Schalter Uhr

//Serial.println("1 für WLAN 0 für LAN");
//Serial.println(digitalRead(35));

if (digitalRead(35)==LOW){
  lanok = 0; //LAN
}

if (digitalRead(35)==HIGH){
  lanok = 1; //WLAN
}

if (lanok == 0){ //LAN an GPIO 34 = 0
  Ethernet.init(33);
  Ethernet.begin(macvlan,iplan);

  Serial.print("IP LAN :");
  Serial.println(Ethernet.localIP());
  Serial.print("Gateway LAN: ");
  Serial.println(Ethernet.gatewayIP());
}

```

```

Serial.print("Subnet LAN: ");
Serial.println(Ethernet.subnetMask());
Serial.print("Dns LAN:");
Serial.println(Ethernet.dnsServerIP());

server.begin();
}

if (lanok == 1){ //WLAN an GPIO 34 = 1
WiFi.config(ipwlan,gatewaywlan,subnetwlan,dnswlan);
WiFi.begin("SSID","Passwort"); //WLAN SSID,WLAN Pw festlegen
while (WiFi.status() != WL_CONNECTED){
  delay(1000);
}
Serial.println("Verbunden mit Netzwerk");
Serial.print("IP WLAN :");
Serial.println(WiFi.localIP());
Serial.print("Gateway WLAN: ");
Serial.println(WiFi.gatewayIP());
Serial.print("Subnet WLAN: ");
Serial.println(WiFi.subnetMask());
Serial.print("MAC WLAN: ");
Serial.println(WiFi.macAddress());
Serial.print("SSI: ");
Serial.println(WiFi.RSSI());

serverwifi.begin();
}

} //-----void setup ende-----

void zeitenspeichern(){ //-----void zeitenspeichern beginn-----
String uhr1ges1; //zur Kontrolle
String uhr2ges2; //zur Kontrolle

int pos1;
int pos2;
pos1 = uhr1uhr2.indexOf("uhr1="); //position "uhr1=" im string holen
pos2 = uhr1uhr2.indexOf("uhr2="); //position "uhr2=" im string holen

```

```

//Kontrolle
uhr1ges1 = uhr1uhr2.substring(pos1 + 5,pos1 + 13); //uhr1uhr2 ist der String aus Puffer beim Anklicken von Speichern
uhr2ges2 = uhr1uhr2.substring(pos2 + 5,pos2 + 13); //uhr1uhr2 ist der String aus Puffer beim Anklicken von Speichern

//speichern in Datei

if(SPIFFS.begin()){
  //Serial.println("SPIFFS initialisierung OK");
}
else{
  //Serial.println("SPIFFS initialisierung Fehler");
}

File x = SPIFFS.open("/uhr1.txt","w");
x.print(uhr1ges1);
x.close(); //Datei schließen

File y = SPIFFS.open("/uhr2.txt","w");
y.print(uhr2ges2);
y.close(); //Datei schließen

delay(500);
ESP.restart(); //Neustart des ESP damit neue Zeiten aus Datei geladen werden
}//-----Void zeitenspeichern ende-----

void zeitenholen(){ //-----void aus Datei-----Beginn-----
int i;
if(SPIFFS.begin()){
  //Serial.println("SPIFFS Initialisierung OK");
}
else{
  //Serial.println("SPIFFS Initialisierung Fehler");
}
//Lesen der Datei
File f = SPIFFS.open("/uhr1.txt","r");
for(i=0;i<f.size();i++){
  uhr1ges+= uhr1ges+(char)f.read();
}
f.close(); //Datei schließen

```

```

//Lesen der Datei
File m = SPIFFS.open("/uhr2.txt","r");
for(i=0;i<m.size();i++){
    uhr2ges= uhr2ges+(char)m.read();
}
m.close(); //Datei schließen

} //-----void Zeit holen ende -----

void zeitschaltuhr(){ //-----void zritschaltuhr Beginn-----
Time t= rtc.time(); // Zeit vom Chip abrufen
//Berechnung akt Stunde
float aktstunde =(t.min / 60.0)+ t.hr;

//Uhr1
int resultat1 = 0; //Zustand Uhr1
int resultat2 = 0; //Zustand Uhr2

    anfh1=(uhr1ges.substring(0,2)).toInt(); //aus ges. String der Datei z.B. 13201430 std und min für Anfang und Ende
auslesen
    anfmin1 =(uhr1ges.substring(2,4)).toInt();
    endh1=(uhr1ges.substring(4,6)).toInt();
    endmin1=(uhr1ges.substring(6,8)).toInt();

float anfstunde1 = (anfmin1 / 60.0) + anfh1; //Berechnung Anfangsstunde1
float endstunde1 = (endmin1 / 60.0) + endh1; //Berechnung Endstunde1

if(aktstunde >= anfstunde1 && aktstunde <= endstunde1){ //unter Anfangszeit
    resultat1 = 1;
}
else{
    resultat1 = 0;
}

//Uhr2

    anfh2=(uhr2ges.substring(0,2)).toInt();
    anfmin2 =(uhr2ges.substring(2,4)).toInt();
    endh2=(uhr2ges.substring(4,6)).toInt();
    endmin2=(uhr2ges.substring(6,8)).toInt();

```



```

float anfstunde2 = (anfmin2 / 60.0) + anfh2; //Berechnung Anfangsstunde1
float endstunde2 = (endmin2 / 60.0) + endh2; //Berechnung Endstunde1

if(aktstunde >= anfstunde2 && aktstunde <= endstunde2){ //unter Anfangszeit
    resultat2 = 1;
}
else{
    resultat2 = 0;
}

//Auswertung
if (uhr == 1){
    lcd.setCursor(0,1);
    lcd.print ("Schaltuhr aktiv ");

    if(resultat1 == 1 or resultat2 == 1){
        digitalWrite(26,HIGH); // Uhr Pin 26 an
    }

    else{
        digitalWrite(26,LOW); // Uhr Pin 26 aus
    }
}
else{
    lcd.setCursor(0,1);
    lcd.print ("Schaltuhr inaktiv ");
    digitalWrite(26,LOW); // Uhr Pin 26 aus
}

} //-----void zeitschaltuhr ende-----

void loop(){ //-----void loop Beginn-----

Time t = rtc.time(); // Zeit vom Chip abrufen
char buf[50];
sprintf(buf,sizeof(buf),"%02d.%02d.%04d %02d:%02d:%02d",t.date,t.mon,t.yr,t.hr,t.min,t.sec); //Formatierung Datum und
Zeit
//lcd.clear(); //LCD löschen

```

```

lcd.setCursor(0,0); //LCD 2.Stelle 1. Zeile
lcd.print(buf);    //Ausgabe auf LCD akt Datum unmd Zeit formatiert

if (lanok == 1){ //WLAN an
  lcd.setCursor(16,2);
  lcd.print(WiFi.RSSI());
}

if (lanok == 0){ //LAN an
  lcd.setCursor(16,2);
  lcd.print("LAN");
}

zeitschaltuhr(); //void zeitschaltuhr aufrufen

//.....für LAN GPIO 35 aus.....
if (lanok == 0){
  EthernetClient client = server.available();

if (client) {
  boolean currentLineIsBlank = true;
  while (client.connected()) {
    if (client.available()) {
      char c = client.read();
      buffer+=c;
      if (c == '\n' && currentLineIsBlank) {
        // http response header
        client.println("HTTP/1.1 200 OK");
        client.println("Content-Type: text/html");
        client.println("Connection: close");
        client.println();
        client.println("<!DOCTYPE HTML>");
        client.println("<html>");
        client.println("<body bgcolor=\"d0d0f0\">");

        if(merker!=0){ //Passwort OK
          client.println("<meta http-equiv=\"refresh\" content=\"7\">"); //meta refresh tag hinzufügen, welcher alle 7
          Sekunden den Browser refresht.
        }
      }
    }
  }
}

```

```

client.println("<br><br>");
client.print("<center>");
client.print("<h2>Passworteinhabe</h2>");
client.print("<form method=\"GET\">");
client.print("<p><input type=\"password\" name=\"pswd\" size=\"20\"></p>");
client.print("<p><input type=\"submit\" value=\"OK\"></p>");
client.print("</form>");
client.print("<br><br>");

if(merker==0){
    break;        //Passwort falsch
}

if (zustand1=="an") {
    client.print("Manuell (Pin25) ist <font color = 'green'> An </font><br>");
    digitalWrite(25,HIGH);
    lcd.setCursor(0,3);
    lcd.print("manuell Pin25 an ");
    //merker =1;
}
if (zustand1=="aus") {
    client.print("Manuell (Pin25) ist <font color = 'red'> Aus </font><br>");
    digitalWrite(25,LOW);
    lcd.setCursor(0,3);
    lcd.print("manuell Pin25 aus");
    //merker=1;
}

if (zustand2=="an") {
    client.print("Schaltuhr ist <font color = 'green'> Aktiv </font><br>");
    uhr = 1; //Uhr aktiv setzen
    //merker =1;
}
if (zustand2=="aus") {
    client.print("Schaltuhr ist <font color = 'red'> Passiv </font><br>");
    uhr = 0; //Uhr passiv setzen
    //merker=1;
}

```

```

if (digitalRead(26)== HIGH) {
    client.print("Schaltuhr (Pin26) ist <font color = 'green'> An </font><br>");
}
if (digitalRead(26)== LOW) {
    client.print("Schaltuhr (Pin26) ist <font color = 'red'> Aus </font><br>");
}

client.println("<br><br>");
client.print("<a href=\"an25\" target=\"_self\">Manuell an (Pin 25)</a>-----");
client.print("<a href=\"aus25\" target=\"_self\">Manuell aus (Pin 25 aus</a>");
client.println("<br><br>");
client.print("<a href=\"an14\" target=\"_self\">Schaltuhr aktiv</a>-----");
client.print("<a href=\"aus14\" target=\"_self\">Schaltuhr passiv</a>");
client.println("<br><br>");
//.....Einstellungen anzeigen..auf HTML...Beginn.....
client.print("<center>");
client.print("<h2>Einstellungen der Schaltuhr</h2>");
client.print("<form name=\"f\" method=\"GET\">");
// Textfeld für Uhr1
client.print("Eingabe AnfStd AnfMin EndStd EndMin jeweils 2 Stellen z.B 13201830 entspricht 13:20 bis
18:30");
client.print("<p>Uhr 1:&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type=\"text\" name = \"uhr1\" size=\"7\" maxlength=\"8\" value
=\"");
client.print(uhr1ges); //Variable die im Textfeld stehen soll
client.print("</p>");
// Textfeld für Uhr2
client.print("<p>Uhr 2:&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type=\"text\" name = \"uhr2\" size=\"7\" maxlength=\"8\" value
=\"");
client.print(uhr2ges); //Variable die im Textfeld stehen soll
client.print("</p>");
client.print("<p><input type=\"submit\" value=\"Speichern\"></p>");
client.print("Nach Speichern werden alle Einstellungen resetet, Seite muss neu aufgerufen werden!");
client.print("</form>");
client.print("Systemdatum: ");
client.print (buf);
//.....Einstellungen auf HTML Ende.....

client.print("</center>");
client.println("</body>");
client.println("</html>");

```

```

break;
}

if (c == '\n') {
buffer = "";
currentLineIsBlank = true;
}
else if (c == '\r') {

if(buffer.indexOf("pswd=Passwort") != -1){ //Passwortabfrage Passwort festlegen
merker=1; //Passwort OK
}

if(buffer.indexOf("uhr1=") >= 0 and (buffer.indexOf("uhr2=" >= 0))){ //Speichern angeklickt für Zeiten
speichern
uhr1uhr2 = buffer;
zeitenspeichern(); //void zeitenspeichern aufrufen
merker=1;
}

if(buffer.indexOf("an25")>=0){ //Pin25 manuell an
zustand1="an";
merker=1;
}

if(buffer.indexOf("aus25")>=0){ //Pin25 manuell aus
zustand1="aus";
merker=1;
}
if(buffer.indexOf("an14")>=0){ //Pin14 Schaltuhr aktiv
zustand2="an";
merker=1;
}

if(buffer.indexOf("aus14")>=0){ //Pin14 Schaltuhr passiv
zustand2="aus";
merker=1;
}

buffer = "";

```

```

    }
  }
}
// Gibt dem Webbrowser Zeit, die Daten zu erhalten.
delay(100);
// SchlieÙe Connection:
client.stop();
merker=0;
}
}
//.....für WLAN GPIO 35 an.....

if (lanok == 1){
  WiFiClient client = serverwifi.available();

if (client) {
  boolean currentLineIsBlank = true;
  while (client.connected()) {
    if (client.available()) {
      char c = client.read();
      buffer+=c;
      if (c == '\n' && currentLineIsBlank) {
        // http response header
        client.println("HTTP/1.1 200 OK");
        client.println("Content-Type: text/html");
        client.println("Connection: close");
        client.println();
        client.println("<!DOCTYPE HTML>");
        client.println("<html>");
        client.println("<body bgcolor=\"d0d0f0\">");

        if(merker!=0){ //Passwort OK
          client.println("<meta http-equiv=\"refresh\" content=\"7\">"); //meta refresh tag hinzufügen, welcher alle 7
          Sekunden den Browser refresht.
        }

        client.println("<br><br>");
        client.print("<center>");
        client.print("<h2>Passworteinhabe</h2>");
        client.print("<form method=\"GET\">");

```

```

client.print("<p><input type=\"password\" name=\"pswd\" size=\"20\"></p>");
client.print("<p><input type=\"submit\" value=\"OK\"></p>");
client.print("</form>");
client.print("<br><br>");

if(merker==0){ //Passwort falsch
    break;
}

if (zustand1=="an") {
    client.print("Manuell (Pin25) ist <font color = 'green'> An </font><br>");
    digitalWrite(25,HIGH);
    lcd.setCursor(0,3);
    lcd.print("manuell Pin25 an ");
    //merker =1;
}
if (zustand1=="aus") {
    client.print("Manuell (Pin25) ist <font color = 'red'> Aus </font><br>");
    digitalWrite(25,LOW);
    lcd.setCursor(0,3);
    lcd.print("manuell Pin25 aus");
    //merker=1;
}

if (zustand2=="an") {
    client.print("Schaltuhr ist <font color = 'green'> Aktiv </font><br>");
    uhr = 1; //Uhr aktiv setzen
    //merker =1;
}
if (zustand2=="aus") {
    client.print("Schaltuhr ist <font color = 'red'> Passiv </font><br>");
    uhr = 0; //Uhr passiv setzen
    //merker=1;
}

if (digitalRead(26)== HIGH) {
    client.print("Schaltuhr (Pin26) ist <font color = 'green'> An </font><br>");
}
if (digitalRead(26)== LOW) {
    client.print("Schaltuhr (Pin26) ist <font color = 'red'> Aus </font><br>");
}

```

```

}

client.println("<br><br>");
client.print("<a href=\"an25\" target=\"_self\">Manuell an (Pin 25)</a>-----");
client.print("<a href=\"aus25\" target=\"_self\">Manuell aus (Pin 25 aus</a>");
client.println("<br><br>");
client.print("<a href=\"an14\" target=\"_self\">Schaltuhr aktiv</a>-----");
client.print("<a href=\"aus14\" target=\"_self\">Schaltuhr passiv</a>");
client.println("<br><br>");
//.....Einstellungen anzeigen..auf HTML...Beginn.....
client.print("<center>");
client.print("<h2>Einstellungen der Schaltuhr</h2>");
client.print("<form name=\"f\" method=\"GET\">");
// Textfeld für Uhr1
client.print("Eingabe AnfStd AnfMin EndStd EndMin jeweils 2 Stellen z.B 13201830 entspricht 13:20 bis
18:30");
client.print("<p>Uhr 1:&nbsp;&nbsp;&nbsp;<input type=\"text\" name = \"uhr1\" size=\"7\" maxlength=\"8\" value
=\"\"");
client.print(uhr1ges); //Variable die im Textfeld stehen soll
client.print("</p>");
// Textfeld für Uhr2
client.print("<p>Uhr 2:&nbsp;&nbsp;&nbsp;<input type=\"text\" name = \"uhr2\" size=\"7\" maxlength=\"8\" value
=\"\"");
client.print(uhr2ges); //Variable die im Textfeld stehen soll
client.print("</p>");
client.print("<p><input type=\"submit\" value=\"Speichern\"></p>");
client.print("Nach Speichern werden alle Einstellungen resetet, Seite muss neu aufgerufen werden!");
client.print("</form>");
client.print("Systemdatum: ");
client.print (buf);
//.....Einstellungen auf HTML Ende.....

client.print("</center>");
client.println("</body>");
client.println("</html>");
break;
}

if (c == '\n') {
buffer = "";

```



```

currentLineIsBlank = true;
}
else if (c == '\r') {

if(buffer.indexOf("pswd=Passwort") != -1){ //Passwortabfrage
    merker=1; //Passwort OK
}

speichern if(buffer.indexOf("uhr1=") >= 0 and (buffer.indexOf("uhr2=" >= 0))){ //Speichern angeklickt für Zeiten
    uhr1uhr2 = buffer;
    zeitenspeichern(); //void zeitenspeichern aufrufen
    merker=1;
}

if(buffer.indexOf("an25")>=0){ //Pin25 manuell an
    zustand1="an";
    merker=1;
}

if(buffer.indexOf("aus25")>=0){ //Pin25 manuell aus
    zustand1="aus";
    merker=1;
}

if(buffer.indexOf("an14")>=0){ //Pin14 Schaltuhr aktiv
    zustand2="an";
    merker=1;
}

if(buffer.indexOf("aus14")>=0){ //Pin14 Schaltuhr passiv
    zustand2="aus";
    merker=1;
}

    buffer = "";
}
}
}
// Gibt dem Webbrowser Zeit, die Daten zu erhalten.
delay(100);

```

```

    // SchlieÙe Connection:
    client.stop();
    merker=0;
}
}
//.....Varianten Ende.....

//Display aktualisieren
if (digitalRead(25)==HIGH){ //manueller Pin
    lcd.setCursor(0,3);
    lcd.print("manuell Pin25 an ");
}
else{
    lcd.setCursor(0,3);
    lcd.print("manuell Pin25 aus ");
}

if (digitalRead(26)==HIGH){ //Uhr Pin
    lcd.setCursor(0,2);
    lcd.print("Uhr Pin26 an ");
}
else{
    lcd.setCursor(0,2);
    lcd.print("Uhr Pin26 aus");
}
//Display aktual. Ende

Wire.begin();
Wire.beginTransmission (0x27); //Prüfen ob LCD an ist Adresse 0x27

if ((Wire.endTransmission() == 0) and (lcdmerker > 100)){ // LCD ist an
    //Serial.println("LCD ein");
    lcd.init(); //LCD neu initialisieren
    lcdmerker = 0;
}

lcdmerker = lcdmerker + 1;

if (lcdmerker > 110){
    lcdmerker =0;
}

```

```
}  
  
//Serial.println(lcdmerker);  
delay(50); //Zeiten zwischen 50 und 100  
  
} //-----void loop ende -----
```