



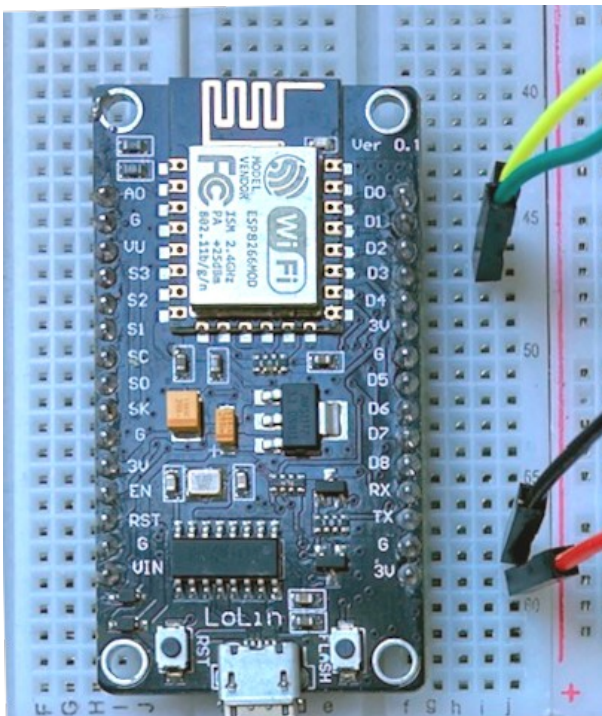
Ziel des Projekts

Mit einem ESP32 oder einem ESP8266 wird mit der Standardbibliothek time.h Datum und Zeit auf einem OLED angezeigt.



Die Schaltpläne

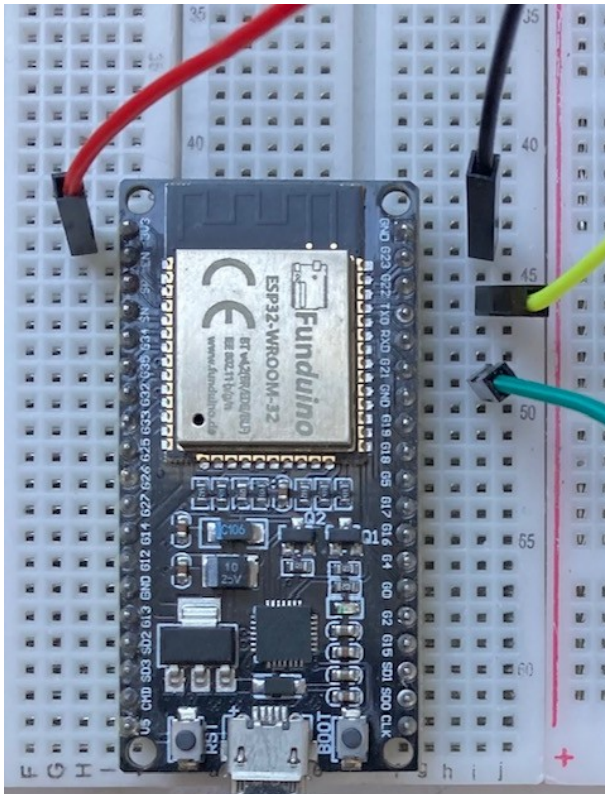
Anschluss an einem NodeMCU



gelb-> SCL (D1)
grün -> SDA (D2)
schwarz -> G
rot -> 3 V

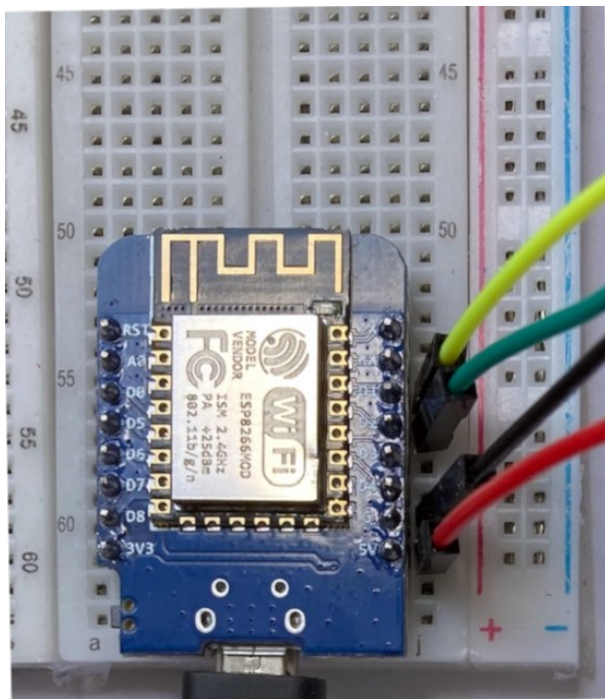


Anschluss an einem ESP32-Wroom



schwarz GND
gelb -> SCL (G22)
grün -> SDA (G21)
rot -> 3,3 V

Anschluss an einem Wemos D1



schwarz GND
gelb -> SCL (D1)
grün -> SDA (D2)
rot -> 5V

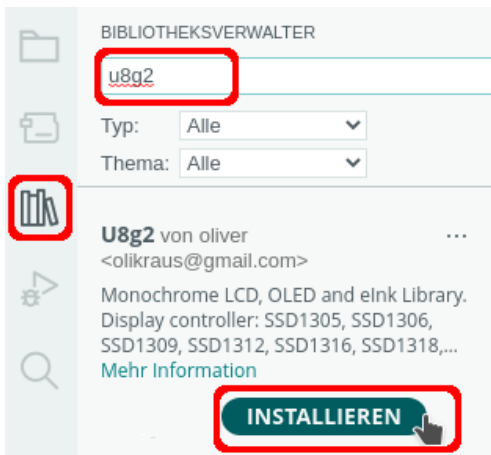


Anschluss an einem Arduino Nano ESP32



schwarz GND
gelb -> SCL (A5)
grün -> SDA (A4)
rot -> 3,3 V

Benötigte Bibliothek





Das Programm

Binde die benötigten Bibliotheken ein und definiere die Variablen. Beachte die Kommentare.

```
#ifdef ESP8266
#include "ESP8266WiFi.h"

#else
#include "WiFi.h"
#endif

#include "time.h"
#include "U8g2lib.h"

// OLED initialisieren
// SSD 1306 Full Buffer
U8G2_SSD1306_128X64_NONAME_F_HW_I2C oled(U8G2_R0);

// SH1106
// U8G2_SH1106_128X64_NONAME_1_HW_I2C oled(U8G2_R0);

// WiFi-Daten
char Router[] = "Router_SSID";
char Passwort[] = "xxxxxxx";

// NTP-Server aus dem Pool
#define Zeitserver "de.pool.ntp.org"

/*
  Liste der Zeitzonen
  https://github.com/nayarsystems/posix_tz_db/blob/master/zones.csv
  Zeitzone CET = Central European Time -1 -> 1 Stunde zurück
  CEST = Central European Summer Time von
  M3 = März, 5.0 = Sonntag 5. Woche, 02 = 2 Uhr
  bis M10 = Oktober, 5.0 = Sonntag 5. Woche 03 = 3 Uhr
*/
#define Zeitzone "CET-1CEST,M3.5.0/02,M10.5.0/03"

// time_t enthält die Anzahl der Sekunden seit dem 1.1.1970 0 Uhr
time_t aktuelleZeit;

/*
  Struktur tm
  tm_hour -> Stunde: 0 bis 23
  tm_min -> Minuten: 0 bis 59
  tm_sec -> Sekunden 0 bis 59
  tm_mday -> Tag 1 bis 31
  tm_wday -> Wochentag (0 = Sonntag, 6 = Samstag)
  tm_mon -> Monat: 0 (Januar) bis 11 (Dezember)
  tm_year -> Jahre seit 1900
  tm_yday -> vergangene Tage seit 1. Januar des Jahres
  tm_isdst -> Wert > 0 = Sommerzeit (dst = daylight saving time)
*/
tm Zeit;
```



```
// Variablen für die Zeit
int Stunden, Minuten, Sekunden;

// beim Start des loop-Teils wird die Zeit einmalig synchronisiert
bool Start = true;

unsigned long Zeitmessung = 0;
```

Der setup-Teil. Beachte die Kommentare.

```
void setup()
{
    Serial.begin(9600);

    // Zeitzone: Parameter für die zu ermittelnde Zeit
    configTzTime(Zeitzone, Zeitserver);

    WiFi.mode(WIFI_STA);

    // WiFi starten
    WiFi.begin(Router, Passwort);

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

    while (WiFi.status() != WL_CONNECTED) {
        delay(200);
        Serial.print(".");
    }

    Serial.println();
    Serial.print("Verbunden mit ");
    Serial.println(Router);
    Serial.print("IP über DHCP: ");
    Serial.println(WiFi.localIP());

    // Zeit holen
    time(&aktuelleZeit);

    // localtime_r -> Zeit in die lokale Zeitzone setzen
    localtime_r(&aktuelleZeit, &Zeit);

    // beim Start entspricht das Datum der Unixtime: 1.1.1970
    // Datum/Kalender sollen erst angezeigt werden, wenn das Datum korrekt ist
    String Jahr = String(Zeit.tm_year + 1900);
    int Zaehler = 0;

    // String Jahr nach "1970" durchsuchen
    int Suche = Jahr.indexOf("1970");

    Serial.println("-----");
    Serial.println("Datum und Zeit holen (maximal 90 Sekunden)...");

    // solange die Suche nicht erfolgreich ist
    while (Suche != -1)
    {
```



```
// aktuelle Zeit holen
time(&aktuelleZeit);

// localtime_r -> Zeit in die lokale Zeitzone setzen
localtime_r(&aktuelleZeit, &Zeit);
Jahr = String(Zeit.tm_year + 1900);

// String Jahr nach "1970" durchsuchen
Suche = Jahr.indexOf("1970");

// Zeit in Stunden, Minuten und Sekunden
Stunden = int(Zeit.tm_hour), Minuten = int(Zeit.tm_min), Sekunden = int(Zeit.tm_sec);
delay(1000);
Zaehler++;

if (Zaehler >= 90)
{
    Serial.println();
    Serial.print("Datum und Zeit konnte innerhalb von " + String(Zaehler));
    Serial.println(" Sekunden nicht geholt werden");
    Serial.println("Programm wird beendet");

    // Programm beenden
    while (1);
}

Serial.print(".");
}

Serial.println();

// Datum/Zeit erfolgreich synchronisiert
if (Suche == -1)
{
    Serial.println("-----");
    Serial.println("Datum/Zeit erfolgreich synchronisiert ...");

    if (Zeit.tm_mday < 10) Serial.print("0");
    Serial.print(Zeit.tm_mday);
    Serial.print(".");

    // Monat: führende 0 ergänzen
    if (Zeit.tm_mon < 9) Serial.print("0");

    // Zählung beginnt mit 0 -> +1
    Serial.print(Zeit.tm_mon + 1);
    Serial.print(".");

    // Anzahl Jahre seit 1900
    Serial.println(Zeit.tm_year + 1900);

    if (Zeit.tm_hour < 10) Serial.print("0");
    Serial.print(Zeit.tm_hour);
    Serial.print(":");

    if (Zeit.tm_min < 10) Serial.print("0");
```




```
Serial.print(Zeit.tm_min);
Serial.print(":");

if (Zeit.tm_sec < 10) Serial.print("0");
Serial.println(Zeit.tm_sec);
Serial.println("-----");
}

// Zeit in Stunden, Minuten und Sekunden
Stunden = Zeit.tm_hour, Minuten = Zeit.tm_min, Sekunden = Zeit.tm_sec;

// OLED starten
oled.begin();

// Zeichenfarbe weiß
oled.setDrawColor(1);

Zeitmessung = millis() + 1000;
}
```

Der loop-Teil. Die Darstellung auf dem OLED übernimmt die Funktion ZeitAnzeigen()

```
void loop()
{
  // Start = true
  // -> Zeit einmalig synchronisieren
  if (Start)
  {
    Start = false;
    ZeitAnzeigen();
  }

  // Sekunden weiter zählen
  if (Zeitmessung < millis())
  {
    Zeitmessung += 1000;
    Sekunden++;

    if (Sekunden == 60)
    {
      ZeitAnzeigen();
    }
  }
}
```



Die Funktion ZeitAnzeigen()

```
void ZeitAnzeigen()
{
    // aktuelle Zeit lesen
    time(&aktuelleZeit);

    // localtime_r -> Zeit in die lokale Zeitzone setzen
    localtime_r(&aktuelleZeit, &Zeit);

    // Zeit in Stunden, Minuten und Sekunden
    Stunden = Zeit.tm_hour, Minuten = Zeit.tm_min, Sekunden = Zeit.tm_sec;

    // Zeichenfarbe weiß
    oled.setDrawColor(1);

    // horizontale Schrift
    oled.setFontDirection(0);

    // Wochentag anzeigen
    oled.setCursor(2, 15);
    oled.setFont(u8g2_font_helvB12_tf);

    switch (Zeit.tm_wday)
    {
        case 0:
            oled.print("Sonntag ");
            break;

        case 1:
            oled.print("Montag ");
            break;

        case 2:
            oled.print("Dienstag ");
            break;

        case 3:
            oled.print("Mittwoch ");
            break;

        case 4:
            oled.print("Donnerstag ");
            break;

        case 5:
            oled.print("Freitag ");
            break;

        case 6:
            oled.print("Samstag ");
            break;
    }
}
```




```
// Datum anzeigen
oled.setCursor(2, 33);

// Tag: führende 0 ergänzen
if (Zeit.tm_mday < 10) oled.print("0");
oled.print(Zeit.tm_mday);
oled.print(".");

// Monat: führende 0 ergänzen
if (Zeit.tm_mon < 10) oled.print("0");
oled.print(Zeit.tm_mon + 1);
oled.print(".");

// Anzahl Jahre seit 1900
oled.print(Zeit.tm_year + 1900);

// horizontale Linie
oled.drawLine(1, 36, oled.getDisplayWidth());

// Zeit anzeigen
oled.setFont(u8g2_font_helvR24_tf);
oled.setCursor(2, 64);

if (Zeit.tm_hour < 10) oled.print("0");
oled.print(Zeit.tm_hour);
oled.print(":");

if (Zeit.tm_min < 10) oled.print("0");
oled.print(Zeit.tm_min);
oled.sendBuffer();
oled.clearBuffer();
}
```