

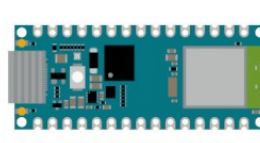
Für diese Anleitung benötigst du einen ESP32 oder einen ESP8266



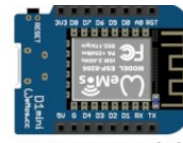
NodeMCU



ESP32-Wroom



Arduino Nano ESP32



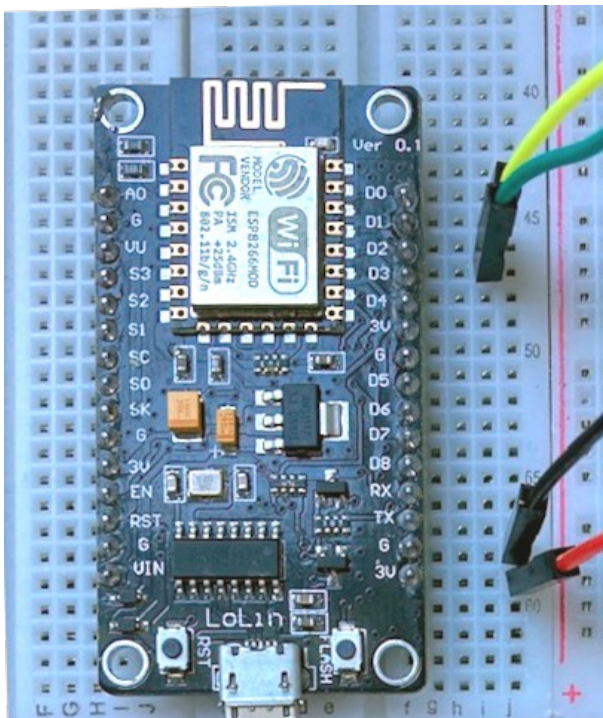
Wemos D1 Mini

Die Standardbibliothek time.h wird verwendet um Datum und Zeit auf einem OLED anzuzeigen.

So sieht es aus:

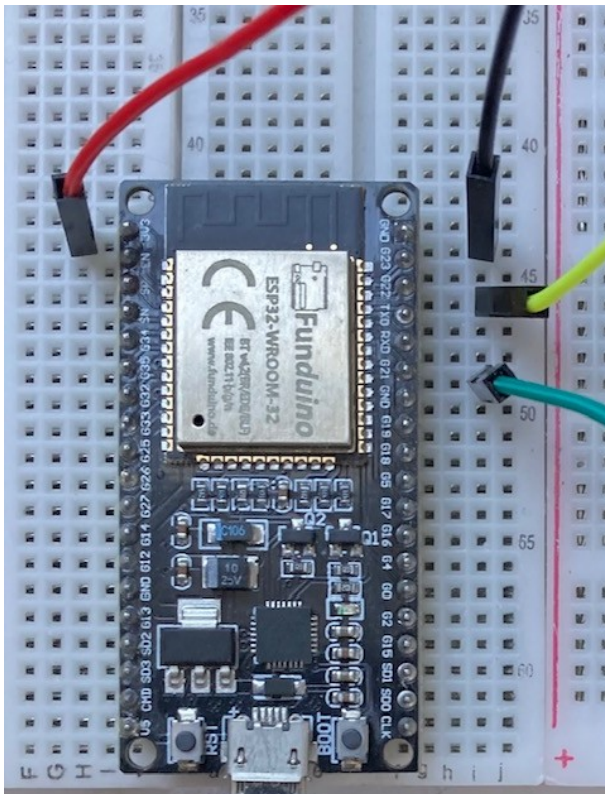


Anschluss an einem NodeMCU



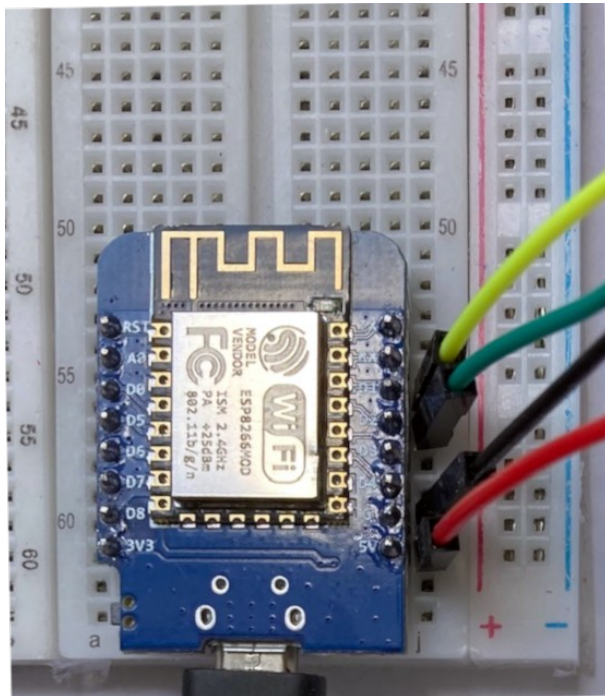
gelb-> SCL (D1)  
grün -> SDA (D2)  
schwarz -> G  
rot -> 3 V

## Anschluss an einem ESP32-Wroom



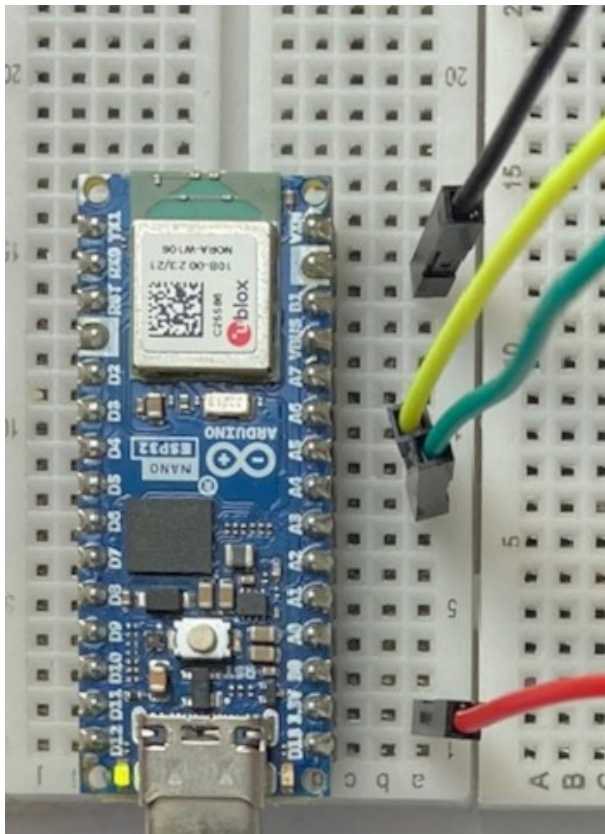
schwarz GND  
gelb -> SCL (G22)  
grün -> SDA (G21)  
rot -> 3,3 V

## Anschluss an einem Wemos D1



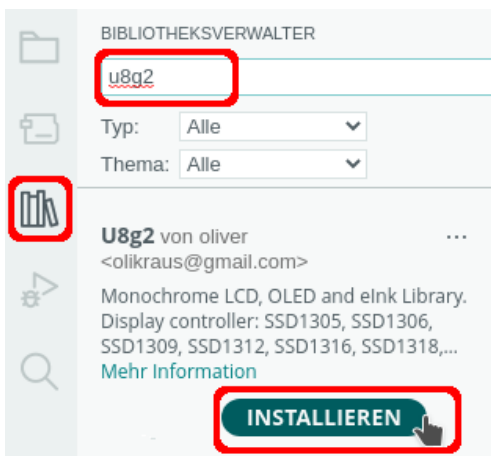
schwarz GND  
gelb -> SCL (D1)  
grün -> SDA (D2)  
rot -> 5V

## Anschluss an einem Arduino Nano ESP32



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

## Benötigte Bibliothek:



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

```
#include "WiFi.h"
#include "time.h"
#include "U8g2lib.h"

/*
  Typbezeichnung mit Bildschirmgröße in Pixeln
  1 = page buffer mode, F = full screen buffer mode
  Hardware I2C
  Name des OLEDs
  Rotation R0 (keine)
*/
```

```
// Parameter 0,96 Zoll SSD1306
U8G2_SSD1306_128X64_NONAME_1_HW_I2C oled(U8G2_R0, U8X8_PIN_NONE);

// Parameter 1,3 Zoll SH1106
// U8G2_SH1106_128X64_NONAME_1_HW_I2C oled(U8G2_R0, U8X8_PIN_NONE);

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;

// Zeit bis zur nächsten Messung
static unsigned long GesicherteStartZeit = 0;
unsigned long Startzeit;

// Sekunden Intervall, kann angepasst werden
int Intervall = 10000;
```

Der setup-Teil. Beachte die Kommentare.

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

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

  WiFi.begin(Router, Passwort);

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

  // SSID des Routers anzeigen
  Serial.println();
  Serial.print("Verbunden mit ");
  Serial.println(WiFi.SSID());

  // IP anzeigen
  Serial.print("IP: ");
  Serial.println(WiFi.localIP());

  // OLED starten
  oled.begin();
}
```

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

```
void loop()
{
  // Startzeit setzen
  Startzeit = millis();

  // wenn das festgelegte Intervall erreicht ist
  if (Startzeit - GesicherteStartZeit > Intervall)
  {
    ZeitAnzeigen();

    // Startzeit zurücksetzen
    GesicherteStartZeit = Startzeit;
  }
}
```



## Die Funktion ZeitAnzeigen()

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

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

    // Zeichenfarbe weiß
    oled.setDrawColor(1);

    // horizontale Schrift
    oled.setFontDirection(0);
    oled.firstPage();

    do
    {
        // 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);
}
while (oled.nextPage());

// Serielle Ausgabe mit Namen des Wochentages
switch (Zeit.tm_wday)
{
    case 0:
        Serial.print("Sonntag ");
        break;

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

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

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

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

    case 5:
        Serial.print("Freitag ");
        break;
```

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

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

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

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

    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.print(Zeit.tm_sec);
    Serial.println();
    Serial.print("Tage seit 1. Januar: ");
    Serial.print(Zeit.tm_yday);
    Serial.println();

    // Normalzeit/Sommerzeit
    if(Zeit.tm_isdst > 0) Serial.println("MESZ = Mitteleuropäische Sommerzeit");
    else Serial.println("MEZ = Mitteleuropäische Zeit");
}
```