

ESP32 + OLed + MicroPython

Installation, Inbetriebnahme

engelbert gruber

07.10.2020

(Rev.20.10.2020)

2ND CLASS GRADING

INTRODUCTION

Es gibt für den ESP32 MicroPython.

Programmieren kann mit pyCharm, VSCode, uPyCraft und auch in thonny.

Thonny und uPyCraft sind kleine Werkzeuge, VSCode und pyCharm sind volle IDEs.

Ich probiere die kleine Variante mit Thonny.

FRAGEN

A. Ist CircuitPython von adafruit etwas anderes ?

PROCEDURE

1. Thonny installieren.

ACHTUNG: Thonny 3.2.7 hat auf Windows10 ein Problem beim Speichern von Dateien auf dem ESP32, es funktioniert nicht.

Im shell-Fenster steht “<thonny>{ ... } </thonny> ” und die

Datenübertragung geht nicht weiter. Auf MacOSX, Linux und Windows7

funktioniert es (bis jetzt).

Siehe Anhang "Thonny auf Win10".

2. ESP32 anstecken
3. Thonny starten.
4. In Extras - Manage plug-ins esptool installieren.
5. Thonny beenden und starten.
6. Von <https://micropython.org/download/esp32/> die Firmware herunterladen ...
[esp32-idf3-20200902-v1.13.bin](https://micropython.org/download/esp32-idf3-20200902-v1.13.bin)
7. In Extras - Optionen - Interpreter "MicroPython (ESP32)" auswählen.
8. Klick "Open the Dialog for ..."
wähle den Seriellen Port (CP2102 Bridge ...)
wähle die Firmware-Datei aus.
Klick "Install"

Warten bis es fertig kopiert ist. Bei mir ist Thonny dann abgestürzt.

9. Testen. In der Thonny-shell, das Fenster "shell", steht "MicroPython v1.13 on 2020-09-02; ESP32 module with ESP32", das bedeutet alles was wir hier eintippen wird am ESP32 ausgeführt.
10. Test Programm: ein- ausschalten der LED auf dem ESP32.

```
from machine import Pin
p2 = Pin(2, Pin.OUT)      # create output pin on GPIO2
p2.value()                # get value, 0 or 1
p2.on()
p2.off()
```

11. Aber auf unserem ESP32 ist keine zusätzliche LED verbaut.

Probieren wir das OLED-Display.

12. Oled und ESP32 verbinden:

ESP32	OLED
3V3	VCC
GND	GND

22	SCL
21	SDA

13. Library herunterladen:

Für die Library (das python-Modul) dass die Ansteuerung des Displays ermöglicht habe ich drei Möglichkeiten gefunden:

- a. <https://randomnerdtutorials.com/micropython-oled-display-esp32-esp8266/>
Ein kompletter Artikel und hier der Code:
<https://raw.githubusercontent.com/RuiSantosdotme/ESP-MicroPython/master/code/Others/OLED/ssd1306.py>
- b. Adafruit: https://github.com/adafruit/Adafruit_CircuitPython_SSD1306
Ob CircuitPython-Bibliotheken auf MicroPython funktionieren ?
- c. Adafruit für displayIO
https://github.com/adafruit/Adafruit_CircuitPython_DisplayIO_SSD1306/
Heisst displayIO das der Python-Prompt ">>>" auf dem Oled ist ?
- d. <https://github.com/micropython/micropython/blob/master/drivers/display/ssd1306.py>

Beim ersten Versuch habe ich randomnerd verwendet, es funktioniert, der Code gehört zu einem Buch und wurde vor zwei Jahren das letzte Mal geändert.

Adafruit hat eine eigene Python Version CircuitPython, ist im Gegensatz zum randomnerd eine Firma, das bedeutet mehr Support.

“d.” ist ein Treiber direkt bei **micropython** ... ich verwende den.

14. Den github-Link öffnen, den “Raw”-Button klicken und dann unter ssd1306.py speichern.
15. Die Datei ssd1306.py im Thonny öffnen, von “This Computer” und Speichern unter “MicroPython device” mit dem Namen “ssd1306.py” speichern.
16. In der thonny-Shell “import ssd1306” eingeben. Es darf keine Meldung kommen.
17. In thonny eine neue Datei “htl-oled.py” erstellen:

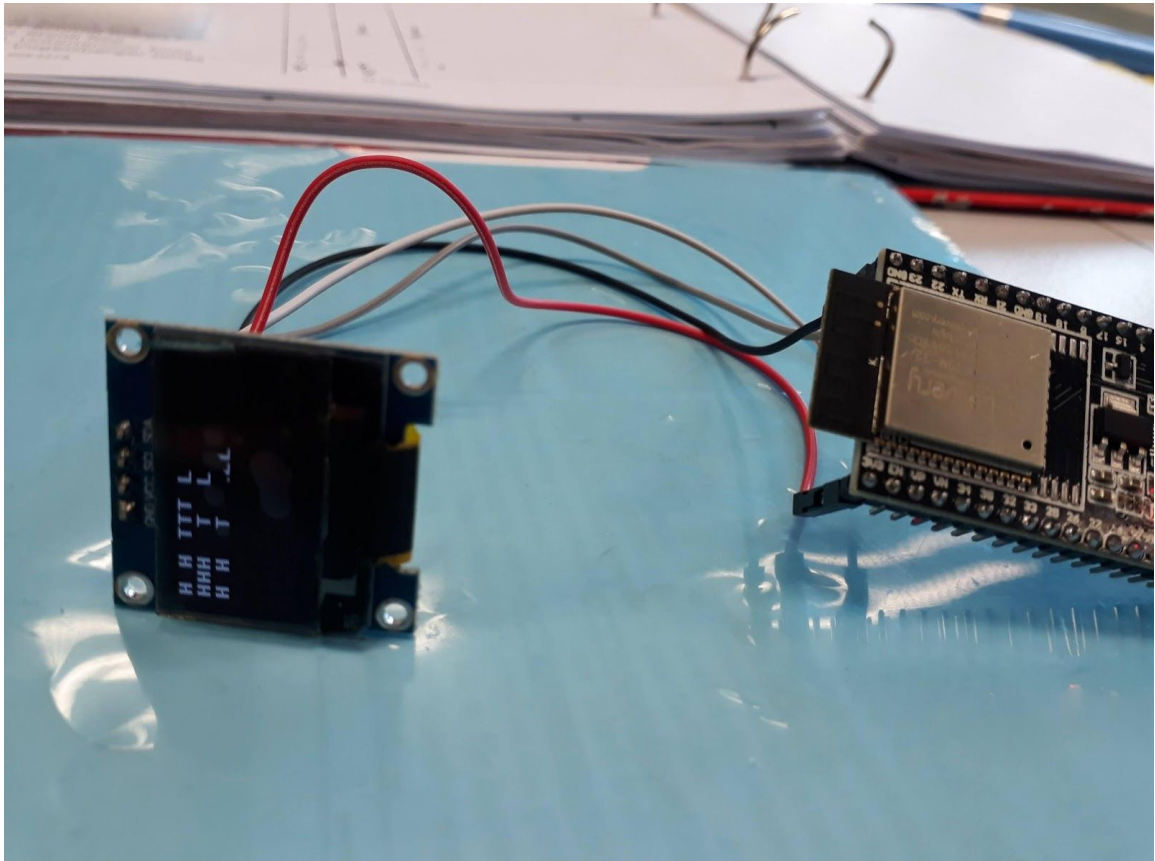
```
from machine import Pin, I2C
import ssd1306
i2c = I2C(-1, scl=Pin(22), sda=Pin(21))
oled_width = 128
oled_height = 64
oled = ssd1306.SSD1306_I2C(oled_width, oled_height, i2c)

oled.text('H H TTT L', 0, 0)
```

```
oled.text('HHH T L', 0, 10)
oled.text('H H T LLL', 0, 20)

oled.show()
```

18. “Ausführen” - “Run current script F5”



OTHER THINGS

WLAN

Folgenden Code zum ESP32 senden, danach am ESP32 `do_connect()` aufrufen.

```
def do_connect():
    import network
    sta_if = network.WLAN(network.STA_IF)
    if not sta_if.isconnected():
        print('connecting to network...')
        sta_if.active(True)
        sta_if.connect('<SID>', '<password>')
        while not sta_if.isconnected():
            pass
    print('network config:', sta_if.ifconfig())
```

Real time clock (RTC)

See machine.RTC

```
from machine import RTC

rtc = RTC()
# set a specific date and time
rtc.datetime((2020, 10, 2, 1, 12, 48, 0, 0))
rtc.datetime() # get date and time
```

Thonny auf Win10

Bei der Vorab/Prerelease Version Thonny 3.3.0b3 funktioniert der Datei-Upload zum esp32.

Solange es noch kein Windows-Installationspaket gibt kann man die beta-Version mit dem Package Installer for Python “pip” installieren.

Unter der Annahme, dass python installiert ist.

1. Ein Kommandofenster “cmd” öffnen.
2. “python -m pip list” eingeben, (python lädt das Modul pip und das Modul listet die installierten Pakete auf.
3. Die beta-Version von thonny installieren, beziehungsweise auf diese upgraden: “python -m pip install --pre --upgrade thonny”.
4. Thonny ist danach womöglich weder im Pfad noch im Menü installiert. Wir starten es analog zu pip: “python -m thonny” (schreiben das in eine Datei thonny.cmd ...

REFERENCES

1. Thonny : <https://thonny.org/>
2. <https://github.com/micropython/micropython>
3. <https://randomnerdtutorials.com/micropython-oled-display-esp32-esp8266/>