Micropython mit ESP32

Einleitung

Dieses Handbuch begleitet den Videokurs ***Micropython mit ESP32***. Die dazugehörigen Videos sind auf Youtube frei zugänglich. Ich gehe davon aus, dass du die jeweilige Lektion gesehen und nachvollzogen hast.

Mit jeder Lektion wächst das Handbuch um ein Kapitel. Die neuste Version findest du immer im Begleitmaterial zur Lektion. In diesem Begleitmaterial findest du auch alle Source - Codes der Experimente.

Support gibt es ausschliesslich über das Forum.
Auf https://community.hobbyelektroniker.ch/wbb/index.php?board/40-die-lektionen-zu-micropython-mit-esp32/ findest du für jede Lektion einen eigenen Bereich in dem du Fragen stellen kannst.
Selbstverständlich kannst du dort auch auf Fragen anderer Zuschauer antworten.

Auf Micropython wird soweit eingegangen, wie es für das Verständnis der Experimente notwendig ist. Wenn du eine systematischere Einführung möchtest, empfehle ich dir den parallel laufenden Kurs ***Micropython Grundlagen*** zu verfolgen. Auch hier gibt es einen Support - Bereich im Forum: https://community.hobbyelektroniker.ch/wbb/index.php?board/45-die-lektionen-zu-micropython-grundlagen/

Es wird auch zwischendurch immer wieder einmal eine Übungsaufgabe geben. Versuche diese selbstständig zu lösen. Wenn du nicht weiterkommst, suche im Internet die notwendigen Informationen. Wenn das nichts hilft, dann stelle eine Frage im Forum und diskutiere das Problem mit anderen Kursteilnehmern. Erst ganz zum Schluss solltest du dir die Musterlösung anschauen. Diese Lösung ist immer nur ein Vorschlag. Vielleicht ist deine Lösung sogar besser.

Also, dann kann es los gehen. Ich wünsche dir viel Spass!

Lektion 1: Was dich in diesem Kurs erwartet

Wir werden und mit einem etwas weiterentwickelten Mikrokontroller beschäftigen, dem ESP 32. Anders als im Arduino werden wie nicht in der Sprache C oder C++ programmieren, sondern in Micropython. Micropython ist ein auf Mikrokontroller angepasstes Python 3. Du wirst also die Sprache Micropython lernen.

Musst du irgendwelche Voraussetzungen erfüllen?

Der Kurs ist nicht für absolute Mikrokontroller - Neulinge oder Programmieranfänger geeignet. Wenn du aber schon eine Led an einem Arduino zum Leuchten gebracht hast und den Unterschied von Strom und Spannung kennst, solltest du den Kurs mitverfolgen können. Programmierseitig solltest du wissen, was Variablen sind und wozu Funktionen gut sind. Der Arduinokurs auf meinem Youtube - Kanal ist eine sehr gute Voraussetzung. Python oder ESP32 - Kenntnisse sind nicht notwendig. Ein beliebiger Mikrokontroller und eine beliebige Programmiersprache bilden eine gute Grundlage.

Wie weit möchtest du in die Materie eintauchen?

Am meisten lernst du natürlich, wenn du alles was ich zeige nachvollziehst und nicht locker lässt, bevor du es vollständig verstanden hast. Dabei wirst du durch gelegentliche Übungsaufgaben unterstützt. Du wird dann danach in der Lage sein selbstständig ganz andere Projekte zu realisieren.

Teilweise gehe ich aber auf Details ein, die dich vielleicht gar nicht so interessieren. Wir sind hier nicht in der Schule, du kannst also auch Dinge auslassen. Da alle verwendeten Programme zum Download bereit stehen, kannst du sie auch benutzen, ohne deren innere Arbeitsweise vollständig zu verstehen.

Dieser Kurs ist auf Praxis ausgelegt. Wir werden Experimente aufbauen und dabei jeweils die dazugehörigen Befehle von Micropython besprechen. Das wird genügen, falls du schon Kenntnisse von Micropython hast oder gar nicht allzu tief in die Details eintauchen möchtest. Für alle Anderen wird parallel dazu auch der Kurs ***Micropython Grundlagen*** angeboten. Hier liegt dann der Schwerpunkt auf der Programmierung.

Wie gehen wir vor?

Alles beginnt mit der Beschaffung der Hardware und der Installation der notwendigen Software. Ich zeige dir die Installation für Mac und Windows. Unter Linux sollte es mehr oder weniger identisch sein.

Danach gibt es einige Grundlagen zum Thema Micropython. Dabei machen wir auch erste Experimente mit unserem ESP32.

Bald schon wird es aber konkret. Wir starten mit dem Projekt 'Wetterstation', das uns den ganzen Kurs begleiten wird. Du lernst, wie die verschiedenen Sensoren abfragt und die gemessenen Werte ausgeben werden. Die Ausgabe wird zuerst auf die Konsole, später aber auf ein kleines Display erfolgen.

Da der ESP32 WLAN-tauglich ist, werden wir auch eine Ausgabe über einen Webserver erstellen. Damit kannst du mit jedem Computer, jedem Handy oder Tablet im WLAN deine Wetterdaten über den Browser abfragen.

Was ist schon heutzutage eine Wetterstation ohne Wettervorhersagen? Also werden wir über das Internet Wettervorhersagen beziehen und diese ebenfalls darstellen. Das ist aber noch nicht alles, einige Ideen werden sicher noch dazukommen.

Mit jedem Schritt wirst du neue Sprachelemente von Mikropython kennenlernen, so dass du am Schluss in der Lage sein wirst, eigene Ideen zu verwirklichen.

Brauchst du exakt die vorgeschlagene Hardware?

Nein, eigentlich nicht. Das hängt aber von deinen Vorkenntnissen ab. Es gibt sehr viele geeignete ESP32 - Module. Wenn du in die notwendige Dokumentation beschaffen kannst und sie auch verstehst, dann ist das kein Problem. Du musst in der Lage sein andere Pins zu verwenden und natürlich dadurch auch die Verdrahtung und die Programme anzupassen. Wenn du hier unsicher bist, dann verwende einfach die vorgeschlagene Hardware.

Bei vollständig anderen Sensoren ist allerdings Vorsicht angebracht. Oft werden diese auf ganz andere Art programmiert und du bist dann auf dich allein gestellt.

Wo bekommst du Unterstützung?

Im Forum selbstverständlich. Die normalen Kommentare auf Youtube eignen sich nicht für ausführlich Antworten auf Fragen.

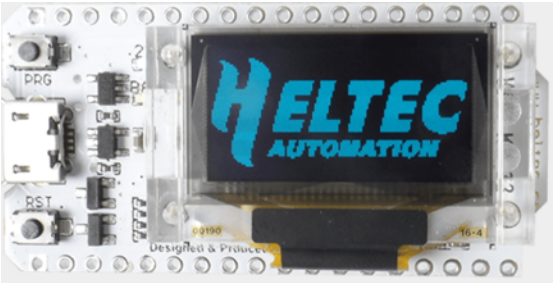
https://community.hobbyelektroniker.ch/wbb/index.php?board/40-die-lektionen-zu-micropython-mit-esp32/

und

https://community.hobbyelektroniker.ch/wbb/index.php?board/45-die-lektionen-zu-micropython-grundlagen/

Zu jeder Lektion wird es einen Bereich geben, in dem Fragen beantwortet werden. Auch deine Antworten sind selbstverständlich immer sehr erwünscht.

Lektion 2: Die Hardware



Heltec WiFi Kit 32

https://heltec.org/project/wifi-kit-32/

Ich empfehle dir eine Bestellung direkt beim Hersteller, Bangood oder Ali-Express. Mit 2 - 3 Wochen Lieferzeit musst du aber rechnen. Bestelle mindestens 2 Stück. So hast du ein Board zum Experimentieren und ein zweites zum fixen Einbau in deine Wetterstation.
Das Board bekommst du vielleicht auch über Amazon oder Ebay. Achte darauf, dass du nicht die LoRa - Variante bestellst. Auf diesem Board sind viele Pins bereits belegt.

Sensor BME280 mit I2C (SCL, SDA) für Luftdruck, Feuchte und Temperatur

Die sind sehr verbreitet, du solltest sie an vielen Orten bekommen. Wichtig ist einfach, das es sich um eine I2C - Variante handelt.

Ich werde auch den ***Sensor BME680*** ausprobieren (https://www.aliexpress.com/item/32902672818.html), dieser wird aber nicht Voraussetzung für den Bau der Wetterstation sein.

Ich werde auch Experimente mit ***anderen Displays*** machen. Welche das genau sind, steht noch nicht fest. Dazu werde ich in meiner Bastelkiste graben und hoffen, dass ich etwas Passendes finde. Ob es dann in der Wetterstation verwendet wird, entscheidet sich erst später. Die vollständige Anzeige wird sowieso nur über einen Browser zu sehen sein.

Der Rest (Steckbrett, Anschlussdrähte, USB - Kabel usw.) sollte ja schon in deiner Bastelkiste vorhanden sein.

Andere ESP32 - Board können verwendet werden, wenn du damit zurecht kommst. Eine andere Pin - Belegung wird dann von dir immer Anpassungen an der Verdrahtung und dem Programm verlangen. Sorge auf jeden Fall dafür, dass du vom Hersteller oder Lieferanten die entsprechenden Informationen bekommst.

Beim Sensor BME280 empfehle ich dir dringen, dich daran zu halten. Sonst wirst du auf dich alleine gestellt sein.

Spezifikationen

- ESP32 (240MHz Tensilica LX6 dual-core + 1 ULP, 600 DMIPS)
- 520KB SRAM
- Wi-Fi, dual mode Bluetooth
- 3 x UART, 2 x SPI, 2 x I2C, 1 x I2S
- 29 x general GPIO
- 2 x 12 bit ADC, an 18 Pins
- 2 x 8 bit DAC
- 4 MB (32 MBits) Flash
- 1 x Micro - USB
- USB to UART: CP2102 (Treiber von silabs.com)
- Battery connector: 3.7V Lithium
- Display: OLED 0.96 inch, 128 x 64

Zur Verfügung stehende Pins

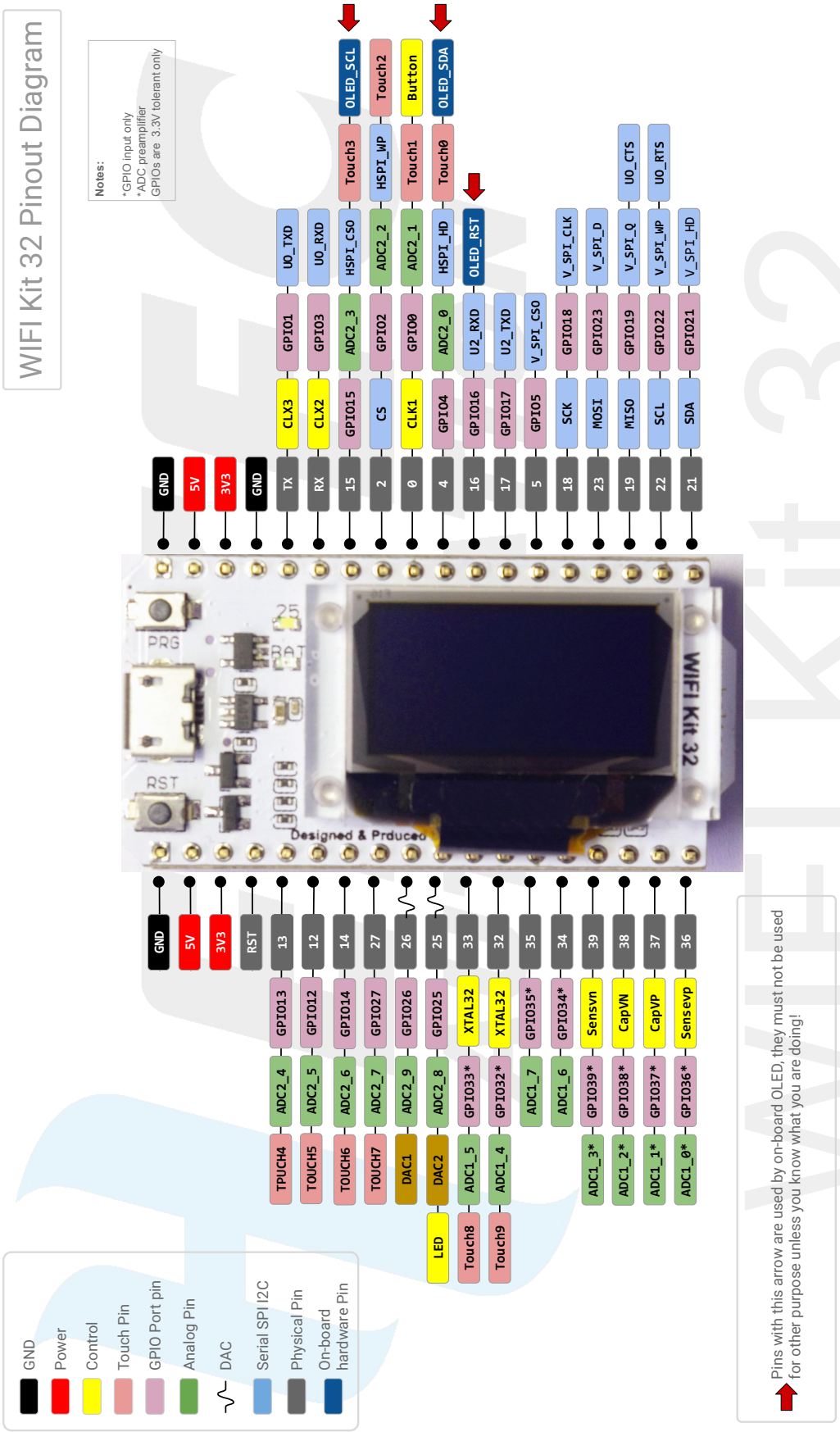
Diese Pins werden beziehen sich auf das Heltec - Board. Falls du ein anderes Board verwendest, musst du die Liste anpassen.

Achtung: Alle Pins nur 3.3V, sie sind nicht 5V tolerant !!!

GPIO 0	Eingebauter Button	
GPIO 2	frei	
GPIO 4	OLED	SDA
GPIO 5	frei	
GPIO 12	frei	
GPIO 13	frei	
GPIO 14	frei	
GPIO 15	OLED	SCL
GPIO 16	OLED	muss HIGH gesetzt werden
GPIO 17	frei	
GPIO 18	frei	SCK (SPI)
GPIO 19	frei	MISO (SPI)
GPIO 21	frei	SDA (I2C)
GPIO 22	frei	SCL (I2C)
GPIO 23	frei	MOSI (SPI)
GPIO 25	Eingebaute LED	
GPIO 26	frei	
GPIO 27	frei	
GPI 34	frei	nur Input, kein PULL_UP/PULL_DOWN
GPI 35	frei	nur Input, kein PULL_UP/PULL_DOWN
GPI 36	???	nur Input, kein PULL_UP/PULL_DOWN
GPI 37	???	nur Input, kein PULL_UP/PULL_DOWN
GPI 38	???	nur Input, kein PULL_UP/PULL_DOWN
GPI 39	???	nur Input, kein PULL_UP/PULL_DOWN

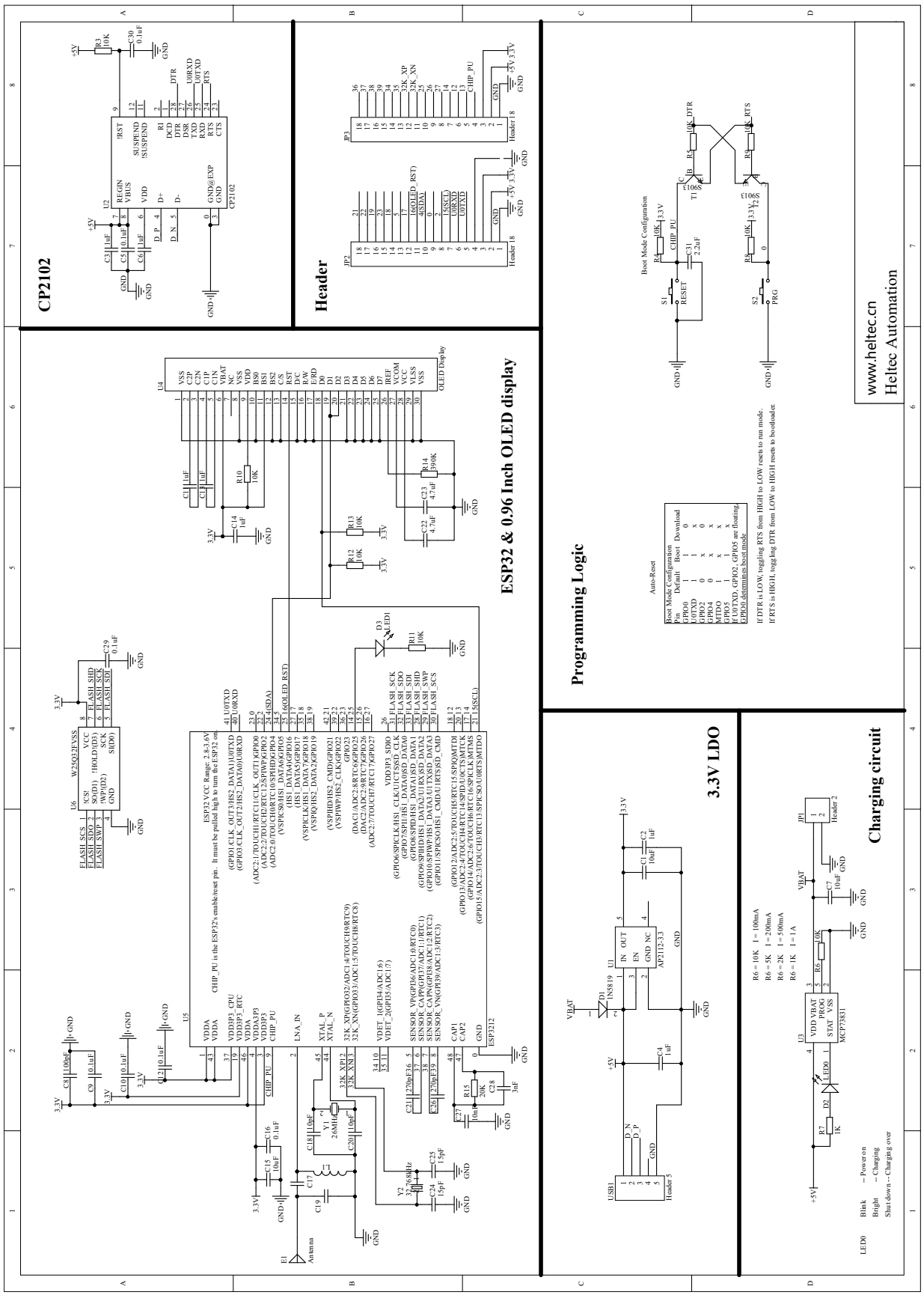
Pinout

https://github.com/Heltec-Aaron-Lee/WiFi_Kit_series/blob/master/PinoutDiagram/WIFI%20Kit%2032.pdf



Das Schema

https://github.com/Heltec-Aaron-Lee/WiFi_Kit_series/blob/master/SchematicDiagram/WIFI_Kit_32_Schematic_diagram.PDF



Lektion 3: Die Software

Die Entwicklungsumgebung

Wir werden mit der Thonny - IDE arbeiten. Sie ist sehr einfach aufgebaut und lässt sich leicht installieren. Falls du die entsprechenden Kenntnisse hast, kannst du selbstverständlich auch grosse Entwicklungsumgebungen wie PlatformIO oder PyCharm verwenden. Das ist aber gar nicht so einfach, wie es auf den ersten Blick aussieht.

Ich verwende im Kurs Thonny (https://thonny.org/) und zwar in der ***Version 3.1.2***. Zum Zeitpunkt des Kursstartes war die Version 3.2.0 im Beta - Stadium und noch nicht stabil genug. Das ist schade, da die Version 3.2 einige interessante Neuerungen kennt. Einige Funktionen fallen aber weg, so dass sich die Arbeitsweise ändern wird.

Die Version 3.1.2 kann für Windows und Mac hier heruntergeladen werden:

Windows: https://github.com/thonny/thonny/releases/download/v3.1.2/thonny-3.1.2.exe
Mac: https://github.com/thonny/thonny/releases/download/v3.1.2/thonny-3.1.2.dmg

Weitere Informationen auf Github: https://github.com/thonny/thonny/releases/tag/v3.1.2

Installation

1. Treiber für USB-Port herunterladen und installieren:
https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers

2. Download Micropython für ESP32:
https://micropython.org/download

3. Thonny herunterladen und installieren:
https://thonny.org oder die Links im vorherigen Abschnitt.

4. ESP32 - Board anschliessen

5. Thonny starten

6. Tools / Manage plug-ins...
- Suchen nach esptool, installieren
- Suchen nach thonny-esp, installieren

7. Thonny beenden und neu starten

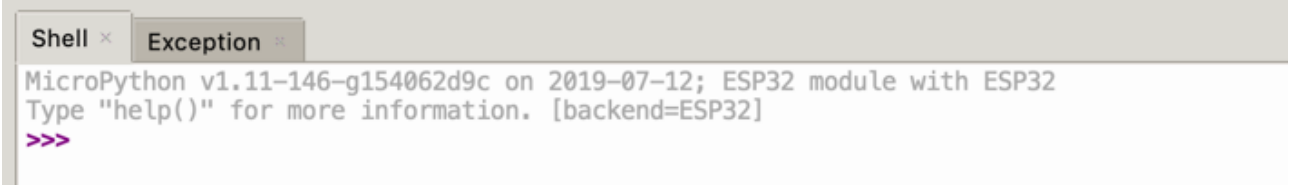
8. Preferences / Interpreter
- MicroPython on ESP32
- Port: CP2102

9. Device / Erase ESP8266/ESP32 flash ausführen

10. Device / Install Micropython to ESP8266/ESP32 ausführen

11. Run / Stop/Restart backend ausführen

Micropython sollte sich jetzt melden.



Erste Schritte

Jetzt wollen wir aber wissen, ob es funktioniert. Also schreiben wir unser erstes Python - Programm und führen es aus. Das Programm muss noch näher betrachtet werden. Das ist dann aber ein Thema für die nächste Lektion.

blink.py

```
# Ein einfaches Blink - Programm
import time
from machine import Pin

interne_led = Pin(25, Pin.OUT)

print("blink.py gestartet")
while True:
    interne_led.on()
    print("ein")
    time.sleep_ms(500)  # 500 ms
    interne_led.off()
    print('aus')  # auch diese Schreibweise ist möglich
    time.sleep(0.5)     # 0.5 Sekunden
```

Dieses Programm läuft in einer Endlosschleife und muss abgebrochen werden! Der Abbruch kann mit ***Stop/ Restart backend*** oder mit ***ctrl d*** durchgeführt werden.