

V Vorbereitung

Hardware

Es gibt zahlreiche ESP32-Boards. Sie unterscheiden sich hauptsächlich darin, welche Anschlüsse des ESP32 nach außen geführt sind und welche Module zusätzlich auf dem Board montiert sind. Wir werden hier mit der Variante "TTGO T-Display" arbeiten. Sie ist preislich erschwinglich (ab ca. 10 Euro) und bietet ein fest verdrahtetes Farb-Display mit einer Auflösung von 240 x 135 Pixel. Zudem gibt es für dieses Board eine spezielle Micropython-Firmware, welche dieses Display gut unterstützt. Insbesondere bei Stand-Alone-Lösungen hat sich dieses Display als eine wertvolle Ergänzung erwiesen.

TTGO T-Display

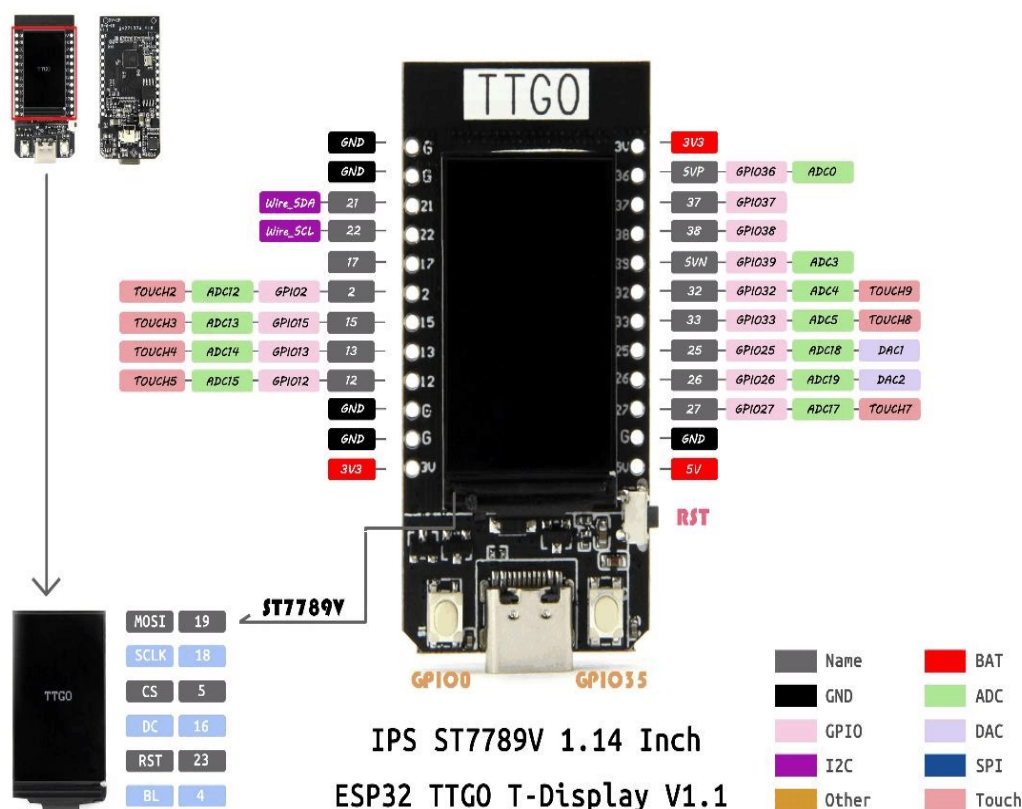


Abb. 1

Für die technischen Daten des Boards und des Displays verweise ich hier auf die Datei `ESP32-Materialien.zip`, siehe <http://www.g-heinrichs.de/wordpress/index.php/informatik/TTGO/>. Was für die Programmierung wichtig ist, werde ich jeweils im entsprechenden Kontext angeben.

Wenn Sie sich mit diesem Skript in die Python-Programmierung des ESP32 *einarbeiten* wollen, empfehle ich, sich genau diese Board-Variante anzuschaffen. Dann können Sie bei Bedarf auf die für dieses Board von mir bereitgestellten Programme zurückgreifen (*ESP-Materialien.zip*, s. o.). Natürlich lassen sich die behandelten Projekte auch mit anderen ESP32-Boards realisieren; allerdings kann es in dem einen oder anderen Fall erforderlich sein, Anpassungen vorzunehmen. Meist wird es darum gehen, andere Anschlüsse zu benutzen; bei dem Einsatz eines anderen Displays können die nötigen Änderungen aber auch schon gravierender sein.

Für manche Projekte werden Sie einige weitere Teile benötigen: einen LM75A-Tempersensor, 2 LEDs (rot oder grün) und 2 Vorwiderstände (jeweils zwischen 220 Ω und 470 Ω), ein Breadboard, etwa zehn Verbindungskabel (männlich-männlich) und – last but not least – ein USB-Kabel zum Anschluss des Boards an Ihren Rechner. Auf der ESP32-Seite muss dieses Kabel einen Stecker vom Typ C besitzen.

Bitte beachten Sie: Häufig werden die ESP32-Boards nicht fertig montiert geliefert; es bleibt dem Käufer überlassen, die beiden Steckerleisten selbst an das ESP32-Board zu löten. Mit anderen Worten: Sie benötigen also noch einen Elektronik-Lötkolben und müssen damit umgehen können, oder sie kennen jemanden, der... ☺

Software

Zur Programmierung des ESP32 werden wir hier die Thonny-IDE benutzen. Sie wird auf der Webseite www.thonny.org vorgestellt. Dort findet man direkt rechts neben der Überschrift Links zum kostenlosen Download (Windows, LINUX und Mac).

Es gibt eine Reihe von Installationsanleitungen zur Thonny-IDE im Internet. Die meisten sind in Englisch verfasst. Die Installation des Programms ist eigentlich recht einfach, aber bei der Konfiguration des Programms und auch bei der Installation des Treibers für die USB-Schnittstelle des ESP32-Boards kann der eine oder andere Hinweis gute Dienste leisten. Deswegen möchte ich hier eine möglichst detaillierte Anleitung geben. Diese bezieht sich auf die Windows-Version.

1. Schritt: Thonny installieren

Wir laden das Installationsprogramm `thonny-3.2.7.exe` herunter und starten es. Es erscheint das Formular aus Abb. 2. Dort klicken wir auf die Schaltfläche **Next>** und starten damit die Installation. Wie üblich müssen wir die Lizenz-Vereinbarung akzeptieren, bevor wir fortfahren können (vgl. Abb. 3).

Anschließend bestätigen wir den vorgeschlagen Installationsordner oder ändern ihn ab. Im nächsten Schritt aktivieren wir das Kontrollkästchen zur Er-



Abb. 2

zeugung eines Thonny-Icons auf dem Desktop. Das Icon können wir bei Bedarf später entfernen oder in einen anderen Ordner verschieben. Wir klicken wieder auf **Next** und bekommen die Mitteilung "Fertig zum Installieren", vgl. Abb. 6. Wenn wir wollen, können wir an dieser Stelle über die Schaltfläche **Back** noch Korrekturen vornehmen. Die Installation beginnt, nachdem wir die Schaltfläche **Install** betätigt haben. Sie dauert eine Weile; ihr Ende wird mit dem Fenster aus Abb. 7 angezeigt. Hier betätigen wir die Schaltfläche **Finish**.

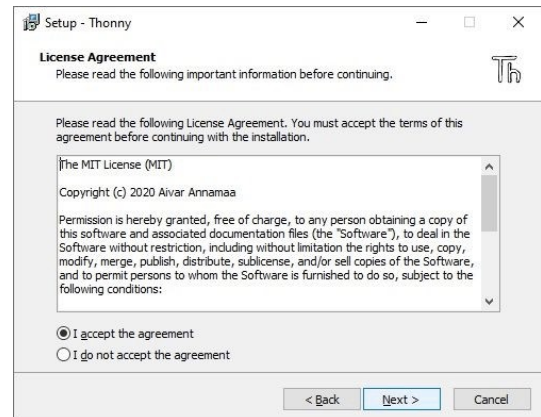


Abb. 3

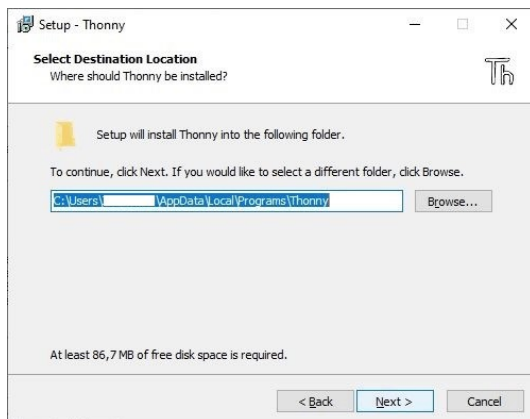


Abb. 4

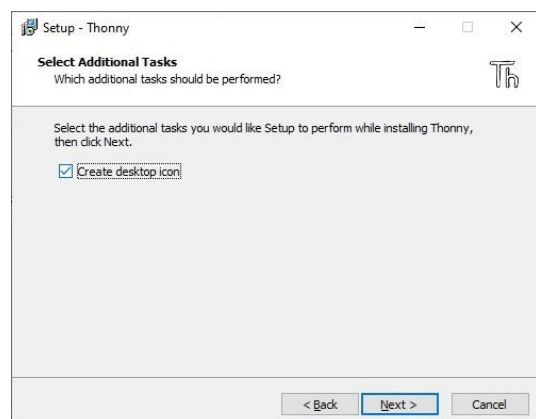


Abb. 5

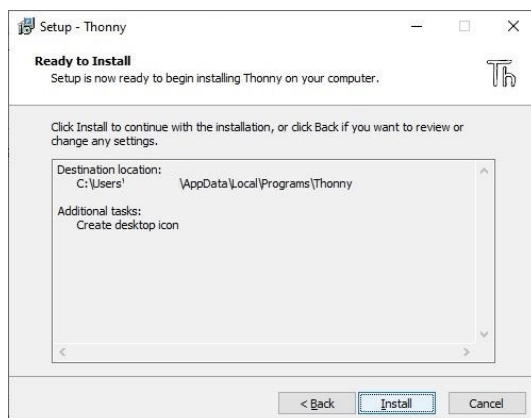


Abb. 6

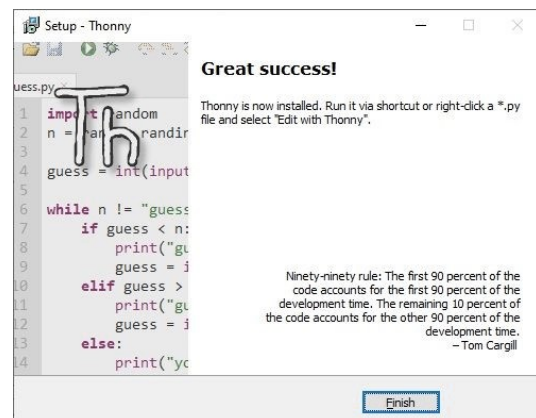


Abb. 7

2. Schritt: Thonny konfigurieren

Wir starten die Thonny-IDE. Zunächst erscheint das Fenster aus Abb. 8. Hier haben Sie die Möglichkeit, die Sprache einzustellen. Die Sprache Deutsch finden Sie in der Auswahlliste oben an erster Stelle. Sie werden aber sehen, dass in dieser ALPHA-Sprachversion die Übersetzung ins Deutsche leider noch nicht vollständig ist.

Nach Anklicken der Schaltfläche *Let's go* öffnet sich das Fenster aus Abb. 9. Hier erkennen Sie zwei Fenster: Das obere Fenster (**Editor**) dient zur Eingabe und Bearbeitung von Python-Programmen. Das untere Fenster (**Shell** bzw. **Terminal**) hat mehrere Funktionen: So können hier z. B. Status-Meldungen vom Python-System angezeigt werden oder auch Ein- und Ausgaben eines Python-Programms erfolgen. In diesem Fall wird uns hier mitgeteilt, dass die Thonny-IDE hier auf die Python-Version 3.7.7 zurückgreift. Dies ist eine Version von Python für PCs, nicht für Microcontroller.

Wir wollen hier mit Micropython arbeiten; wesentlich ist dabei, dass der Micropython-Interpreter nicht auf dem PC, sondern auf dem Microcontroller selbst liegen muss; dadurch können Python-Programme auch stand-alone auf dem ESP32 laufen.

Dazu müssen bei der Thonny-IDE einige Einstellungen vorgenommen werden; dies wollen wir jetzt schrittweise durchführen.

A: Treiber für das ESP32-Board installieren

Damit Thonny mit dem ESP32-Board kommunizieren kann, muss auf dem PC ein passender Treiber installiert sein. Um dies zu testen, schließen wird das Board an den PC an. Wir öffnen den Gerätemanager und kontrollieren, ob sich in der Rubrik **Anschlüsse COM & LPT** ein Gerät mit der Bezeichnung **CP210x** befindet. Ist das so, können Sie die folgenden Anweisungen überspringen und direkt mit Schritt B weiter machen.

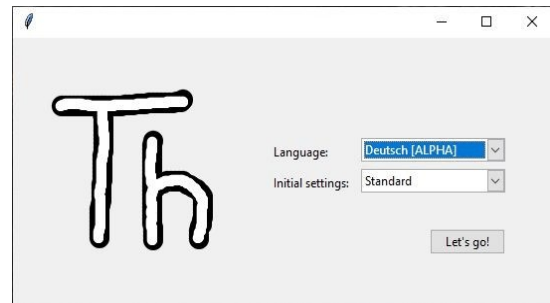


Abb. 8

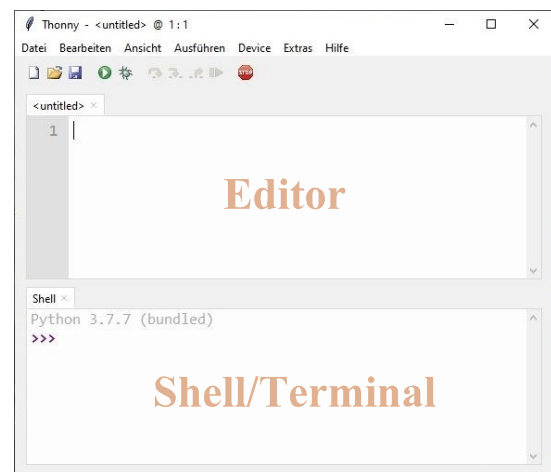


Abb. 9

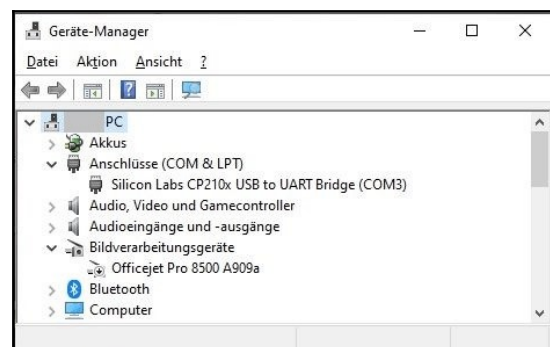


Abb. 10

Entfernen Sie zunächst wieder die Verbindung zwischen PC und ESP32-Modul.

Laden Sie nun den passenden Treiber herunter. Sie finden ihn unter:

<https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers>
oder in dem bereits erwähnten Materialien-Ordner.

Download Software

The CP210x Manufacturing DLL and Runtime DLL have been updated and must be used with v6.0 and later of the CP210x Windows VCP Driver. Application Note Software downloads affected are AN144SW.zip, AN205SW.zip and AN223SW.zip. If you are using a 5.x driver and need support you can download archived Application Note Software.

[Legacy OS software and driver package download links and support information >](#)

Download for Windows 10 Universal (v10.1.8)

Note: The latest version of the Universal Driver can be automatically installed from Windows Update.

Platform	Software	Release Notes
 Windows 10 Universal	Download VCP (2.3 MB)	Download VCP Revision History

Abb. 11

Mit Hilfe der rechten Maustaste laden Sie die Software in einen temporären Ordner. Hier entpacken Sie die herunter geladene zip-Datei.

Starten Sie nun das Treiber-Installations-Programm; wählen Sie die exe-Datei aus, die zu Ihrem Betriebssystem passt.

Nach dieser Installation können Sie das ESP32-Board wieder an Ihren Rechner anschließen. Nun sollte das CP210x-Gerät im Gerätemanager (Abb. 10) auftauchen.

B: Download der Micropython-Firmware

Laden Sie über die Webseite https://github.com/russhughes/st7789_mpy/tree/master/firmware die Datei `firmware.bin` herunter oder benutzen Sie die entsprechende Datei aus dem ESP32-Materialien-Ordner. Diese benötigen wir für den nächsten Schritt.

C: Thonny-IDE einrichten

Wählen Sie im Menüpunkt **Extras** den Punkt **Optionen**. Es erscheint das Formular "Thonny options"; hier wählen Sie die Lasche "Interpreter", vgl. Abb. 12. Wählen Sie zunächst als Interpreter/Gerät aus: MicroPython (ESP32), so wie es in der Abbildung zu sehen ist.

Klicken Sie nun auf die Schaltfläche *Open the dialog for installing...* Es erscheint jetzt ein neues Fenster (vgl. unterer Bereich von Abb. 12). Wählen Sie hier Ihren Anschluss (Port) aus und fügen Sie die den Dateinamen (inkl. Pfad) der Firmware aus Schritt 2 ein; am einfachsten geht das über die Schaltfläche *Browse...* **Ganz wichtig: Aktivieren Sie unbedingt das Kontrollkästchen zum Löschen des Flash-Speichers!**

Wenn Sie nun die Schaltfläche *Install* anklicken, wird die Firmware auf den ESP32 übertragen; dies nimmt etwas Zeit in Anspruch. Nach Abschluss der Firmware-Installation können wir das Optionen-Fenster schließen.

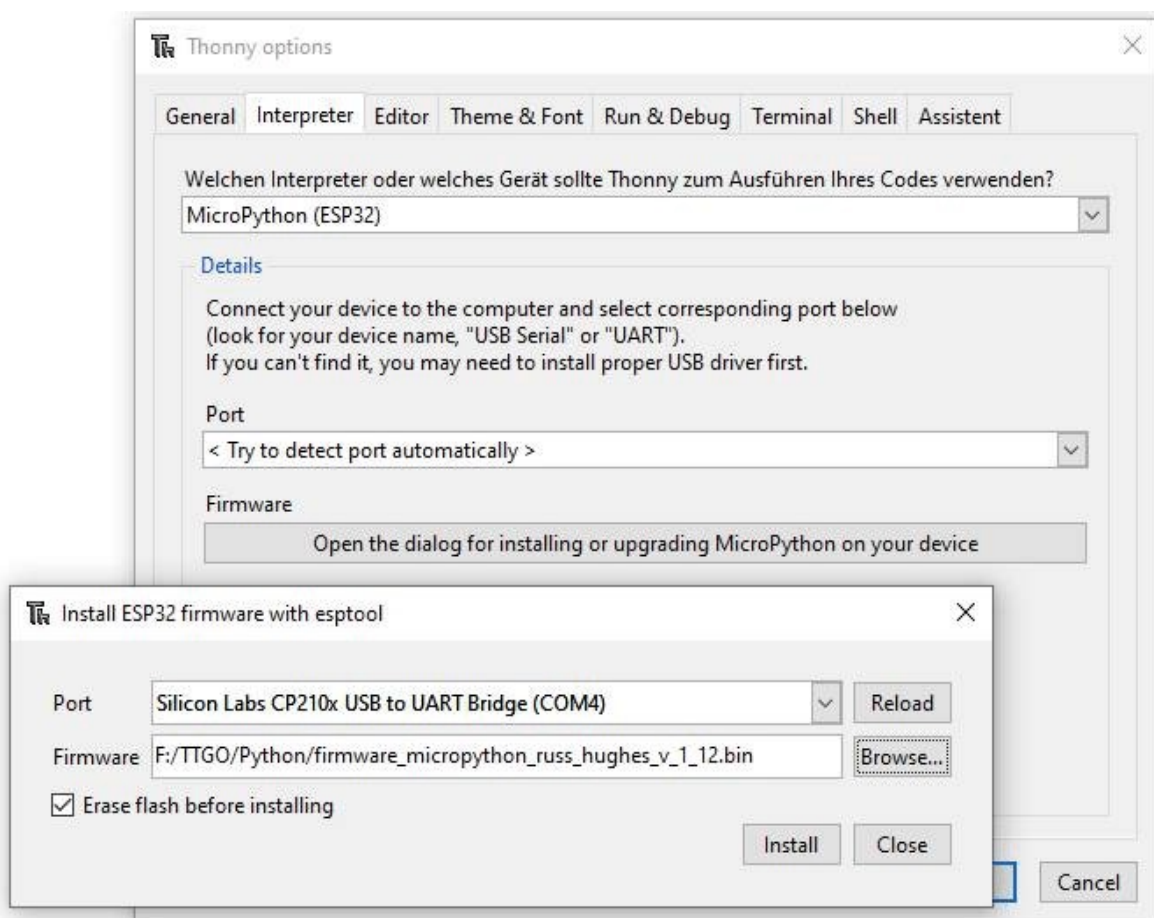
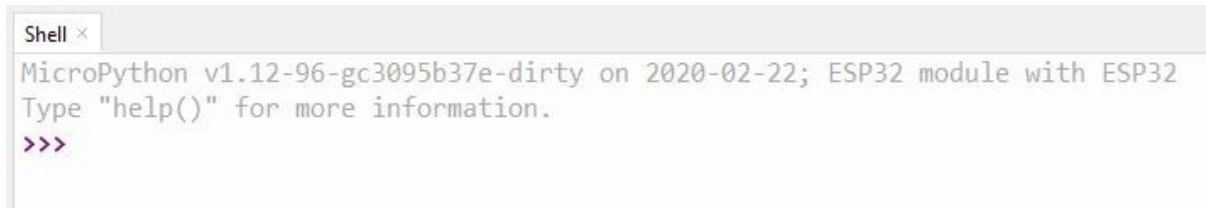


Abb. 12

D: Endkontrolle

Wir lassen das ESP32-Modul angeschlossen und beenden die Thonny-IDE. Anschließend öffnen wir die Thonny-IDE wieder; jetzt zeigt die Shell eine **MicroPython-Version** für den ESP32 an.



```
Shell x
MicroPython v1.12-96-gc3095b37e-dirty on 2020-02-22; ESP32 module with ESP32
Type "help()" for more information.
>>>
```

Abb. 13

Folgen Sie einmal dem Vorschlag in der zweiten Zeile und tippen Sie hinter das **Python-Prompt-Zeichen** >>> den Befehl

```
help()
```

ein; schließen Sie die Eingabe mit der Enter-Taste ab. Als bald erscheinen im Terminal ein paar Informationen zum Umgang mit Micropython – ein weiteres Zeichen dafür, dass die Thonny-IDE und die Firmware auf dem ESP32 korrekt arbeiten. Damit ist die Thonny-Konfiguration abgeschlossen.

Sollten noch irgendwelche Probleme bestehen, finden Sie vielleicht hier eine Lösung:

<https://randomnerdtutorials.com/getting-started-thonny-micropython-python-ide-esp32-esp8266/>