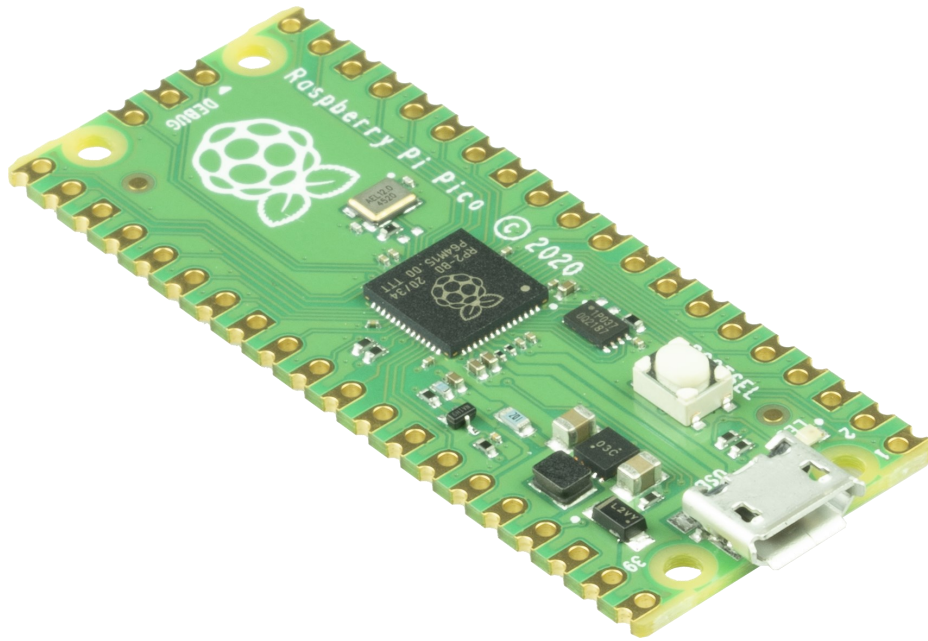


# RASPBERRY PI PICO

## Anfänger Guide

# RASPBERRY PI PICO

## Anfänger Guide



### 1. ALLGEMEINE INFORMATIONEN

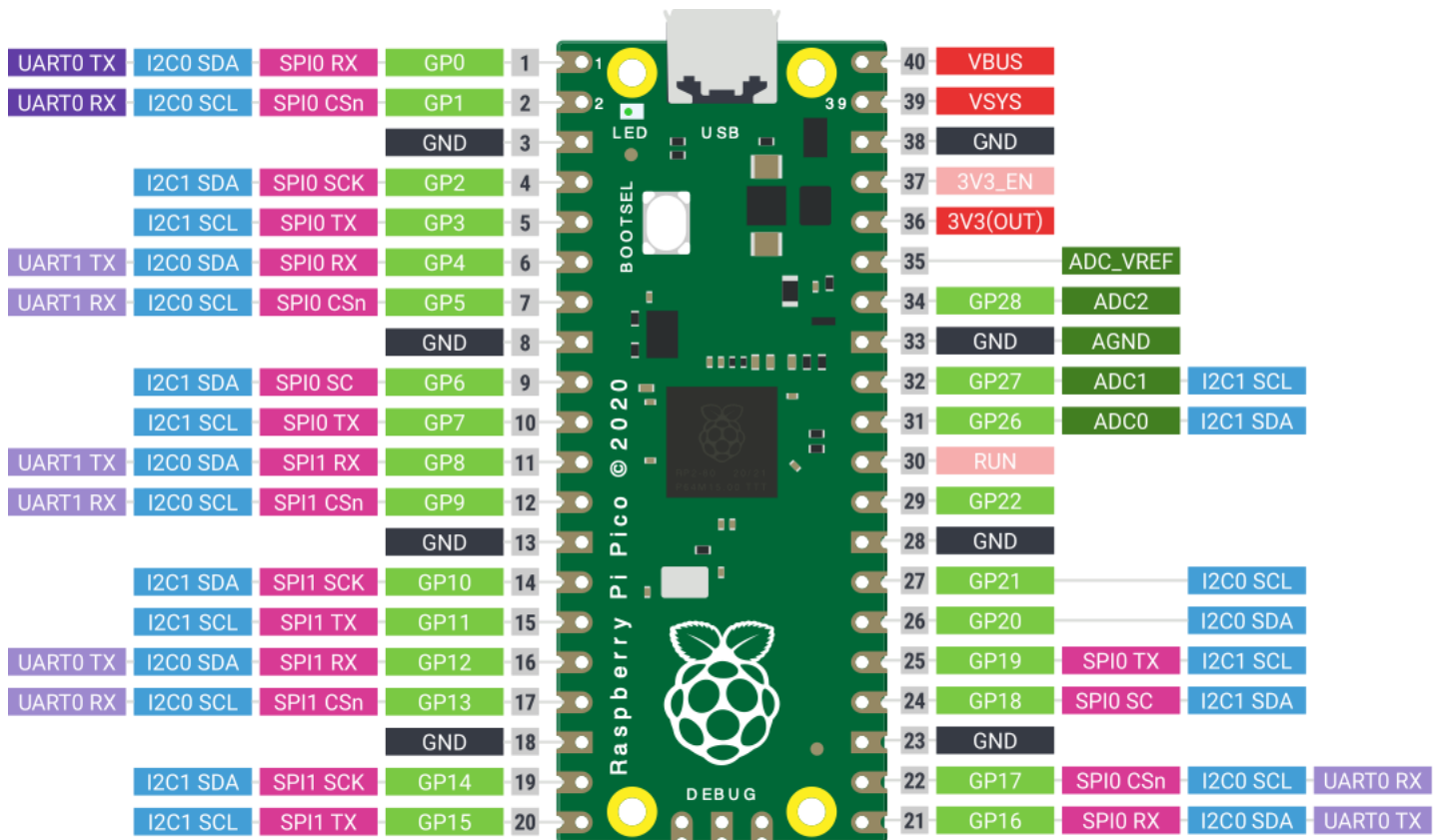
Sehr geehrte\*r Kunde \*in,  
vielen Dank, dass Sie sich für unser Produkt entschieden haben. Im Folgenden zeigen wir Ihnen, was bei der Inbetriebnahme und der Verwendung zu beachten ist.

Sollten Sie während der Verwendung unerwartet auf Probleme stoßen, so können Sie uns selbstverständlich gerne kontaktieren.

## 2. ANSCHLÜSSE

### Pinbelegung:

Je nachdem für welches Modell Sie sich entschieden haben, müssen Sie für die einfache Verwendung Stiftleisten auf Ihren Pico auflöten. Dazu benötigen Sie zwei 20 Pin Stiftleisten mit einem Pinabstand von 2,54 mm.



### Stromversorgung:

Der Pico kann über den **Micro-USB** Anschluss mit einer Spannung von  $5\text{ V} \pm 10\%$  versorgt werden.

Alternativ können Sie den Pico auch mit  $5\text{ V} \pm 10\%$  über den **VBUS** Pin oder mit  $1,8\text{ V} - 5,5\text{ V}$  über den **VSYS**-Pin betreiben.

Beachten Sie, dass die IO-Pins nur für ein **Logiklevel von 3,3 V** ausgelegt sind. Das Anschließen von Geräten mit 5 V Logik kann zur Beschädigung des Pico führen.

### 3. MICROPYTHON INSTALLATION

In den meisten Fällen wird der Raspberry Pi Pico mit MicroPython programmiert. MicroPython ist weitgehend kompatibel mit Python 3 und für den Betrieb auf einem Mikrocontroller optimiert.

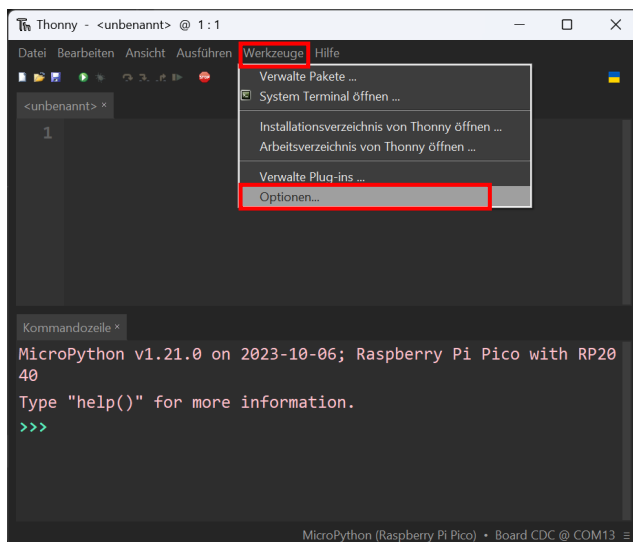
Als Programmierumgebung verwenden wir hier Thonny. Laden Sie sich Thonny unter folgendem Link herunter:

<https://thonny.org/>

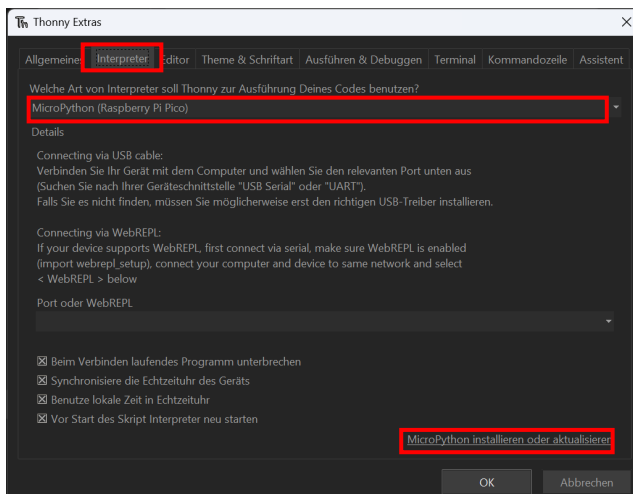
Nach der Installation von Thonny, müssen Sie zunächst die MicroPython Firmware auf Ihren Raspberry Pi Pico spielen. Beachten Sie, dass Sie immer nur eine Firmware auf den Pico aufspielen können. Wenn Sie den Pico mit C++ programmieren möchten, müssen Sie wieder eine andere Firmware aufspielen.

Halten Sie zum Aufspielen der MicroPython Firmware den BOOTSEL Knopf auf dem Pico gedrückt, während sie den Pico über den microUSB Anschluss mit Ihrem PC verbinden.

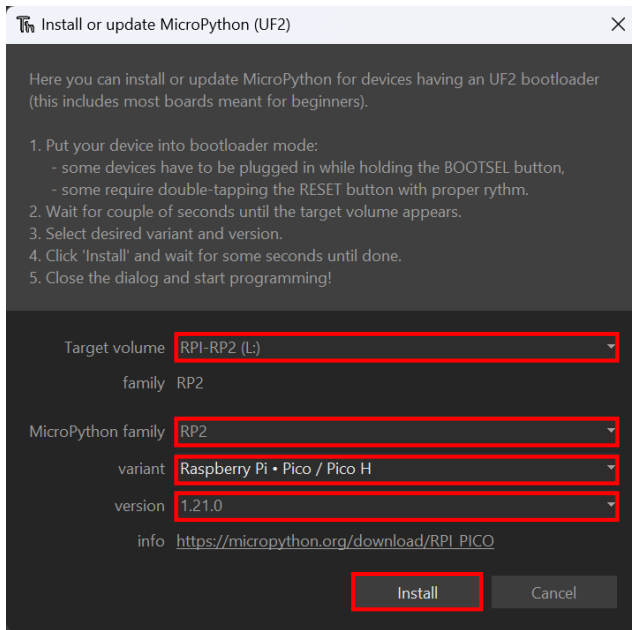
Öffnen Sie nun unter **Werkzeuge** die **Optionen**:



Wählen Sie nun unter **Interpreter** **MicroPython (Raspberry Pi Pico)** aus, klicken Sie anschließend auf **MicroPython installieren oder aktualisieren**.

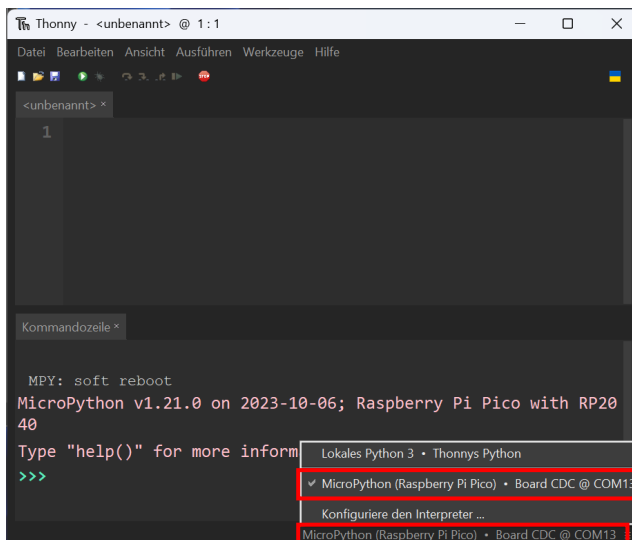


Wählen Sie nun unter **Target volume** Ihren Raspberry Pi aus, unter **MicroPython family** wählen Sie **RP2**, unter **variant** wählen Sie **Raspberry Pi Pico** (achten Sie dabei darauf, dass wenn sie einen Pico mit WLAN benutzen Pico W auswählen müssen). Wählen Sie nun noch die aktuellste Version unter **version** aus und klicken Sie anschließend auf **Install**.



Sobald die Installation abgeschlossen ist, ist Ihr Pico bereit für den Einsatz mit MicroPython.

Sie können nun durch das Klicken auf die untere rechte Ecke von Thonny den zu programmierenden Raspberry Pi Pico auswählen.

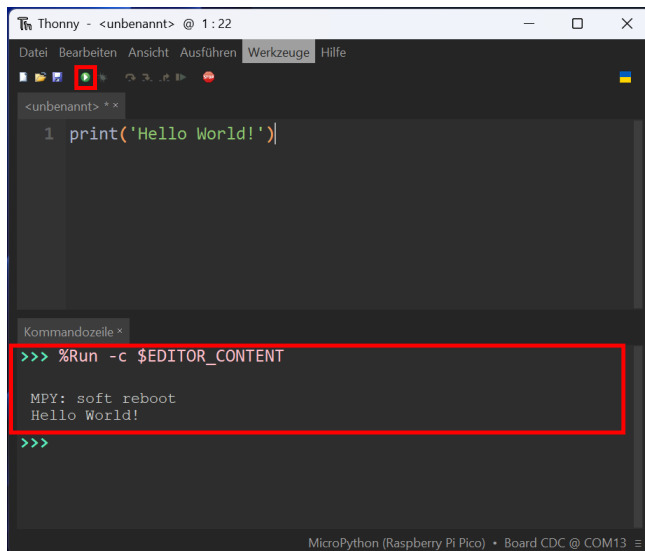


## 4. PROGRAMMIEREN DES PICOS MIT MICROPYTHON

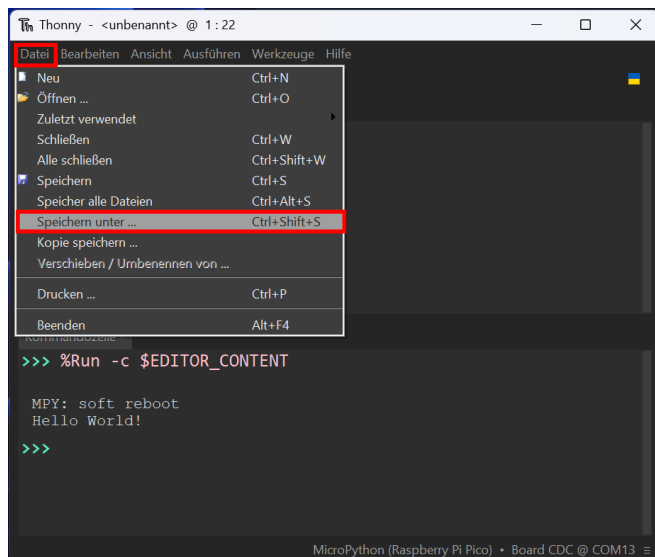
Als einfaches Beispiel, kopieren Sie folgende Zeile in die Thonny IDE:

```
print('Hello World!')
```

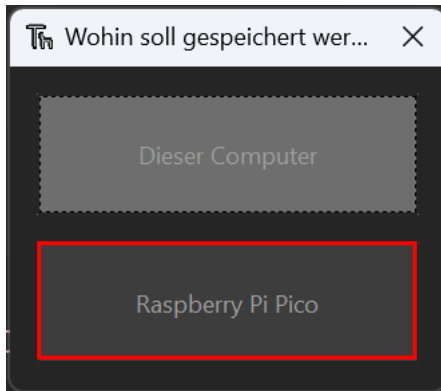
Sie können das Programm durch das Klicken auf das **Play-Symbol** oder drücken der **F5** Taste ausführen. Die Ausgabe des Programms können Sie in der Kommandozeile unter Ihrem Code sehen.



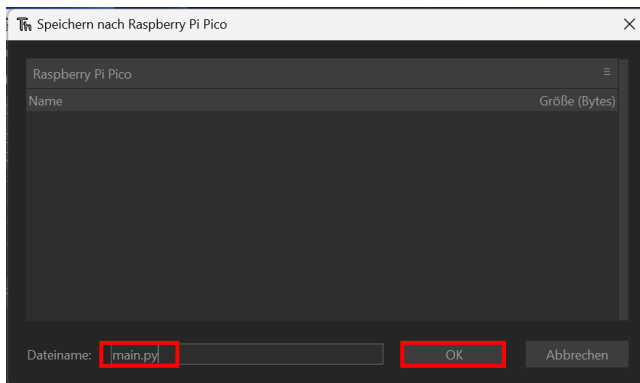
Programme, die automatisch beim Start des Picos ausgeführt werden sollen, müssen als `main.py` Datei direkt auf dem Pico gespeichert werden. Klicken Sie dafür auf **Datei -> Speichern unter...**



Sie werden nun gefragt, ob Sie die Datei auf Ihrem Computer oder direkt auf dem Pico speichern wollen. Wählen Sie hier den **Raspberry Pi Pico** aus.

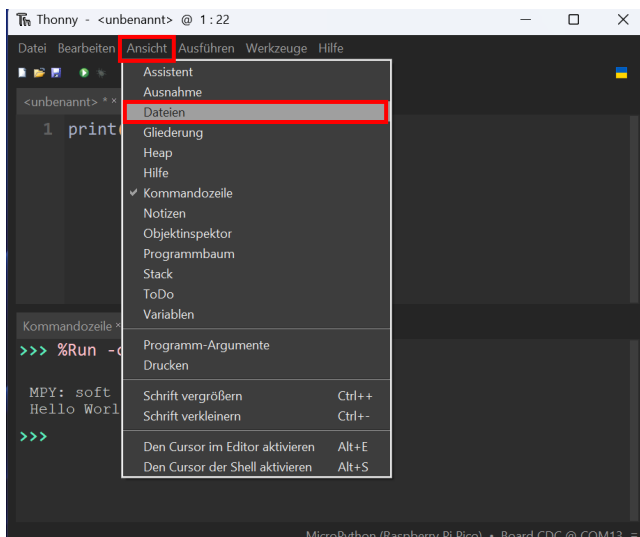


Geben Sie nun `main.py` als Datei-Namen ein und drücken Sie auf **OK**.



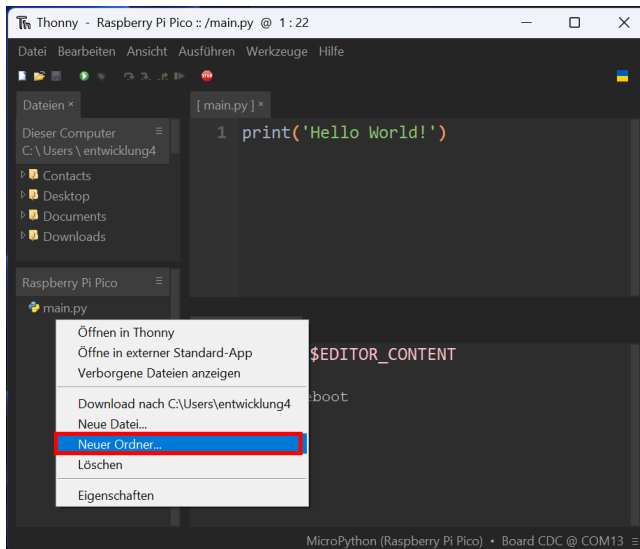
Das Programm wird nun bei jedem Reset des Raspberry Pi Pico automatisch ausgeführt.

Zu einfachen Übersicht welche Dateien sich auf Ihrem Pico befinden, öffnen Sie den Datei Explorer, indem Sie unter **Ansicht** auf **Dateien** klicken.

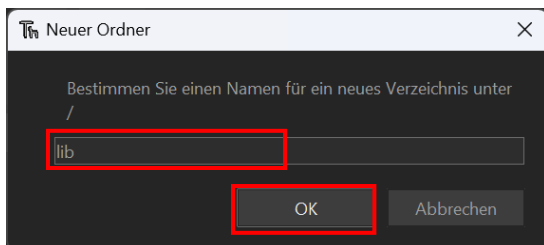


Wenn Sie in Ihrem Programm Bibliotheken verwenden, die nicht bereits in MicroPython integriert sind, müssen die Bibliotheksdateien in einem Ordner mit dem Namen lib auf dem Pico gespeichert werden.

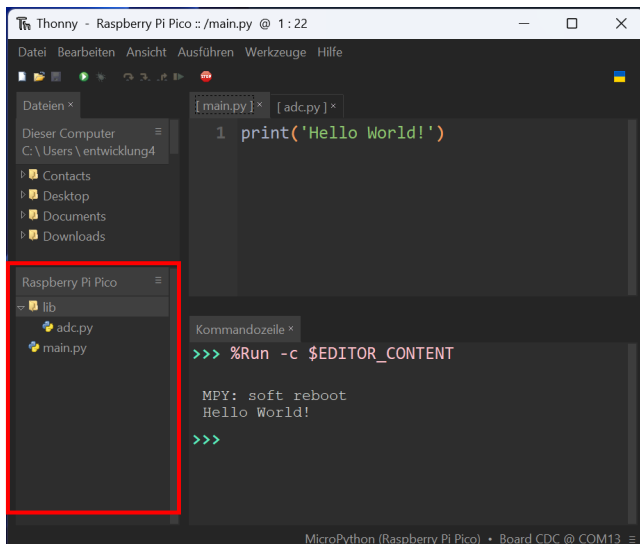
Sie können einen neuen Ordner auf Ihrem Pico erstellen, indem Sie in dem Bereich wo Ihre Dateien vom Pico angezeigt werden, rechts klicken und auf **Neuer Ordner...** gehen.



Geben Sie den Ordner-Namen **lib** ein und klicken Sie auf **OK**.



Sie können nun die verwendete Bibliotheksdatei auf die selbe Art speichern wie zuvor. Jedoch diesmal noch in den Unterordner lib. Das Dateiverzeichnis auf Ihrem Pico sollte dann wie folgt aussehen:





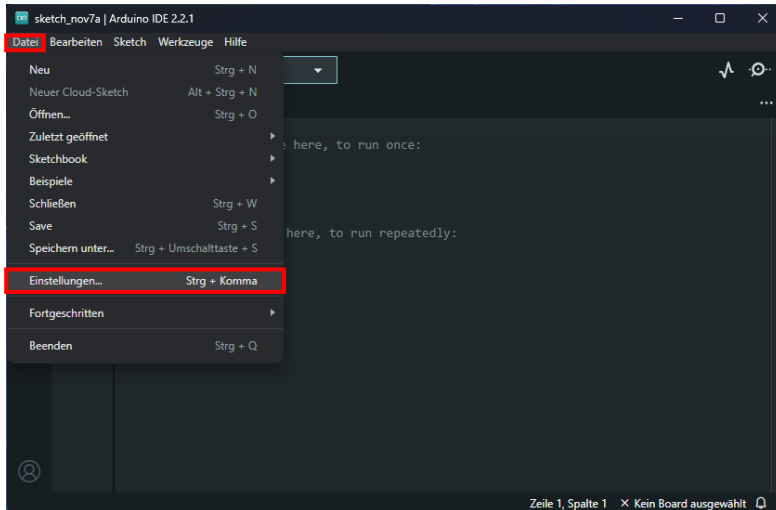
## 5. C++ INSTALLATION

Neben MicroPython können Sie den Raspberry Pi Pico auch in C++ programmieren.

Als Programmierumgebung verwenden wir hier die Arduino IDE. Laden Sie sich diese unter folgendem Link herunter:

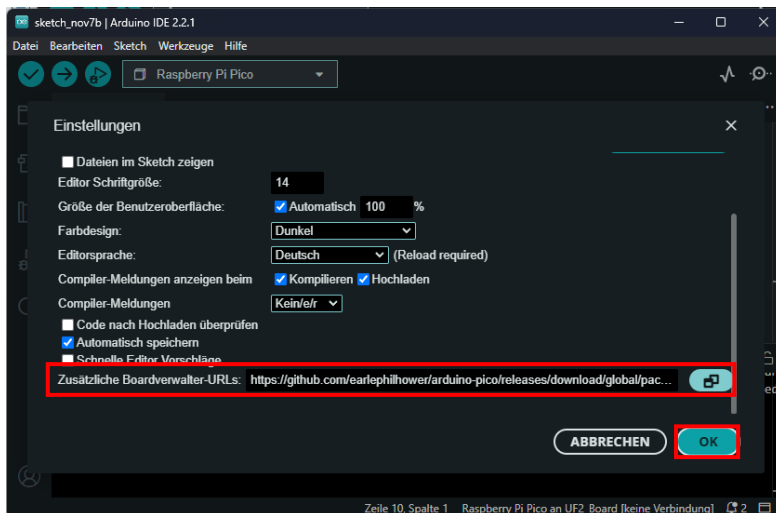
<https://www.arduino.cc/software>

Nach der Installation der Entwicklungsumgebung, müssen Sie zunächst einen zusätzlichen Boardverwalter hinzufügen. Gehen Sie dazu auf **Datei -> Einstellungen**

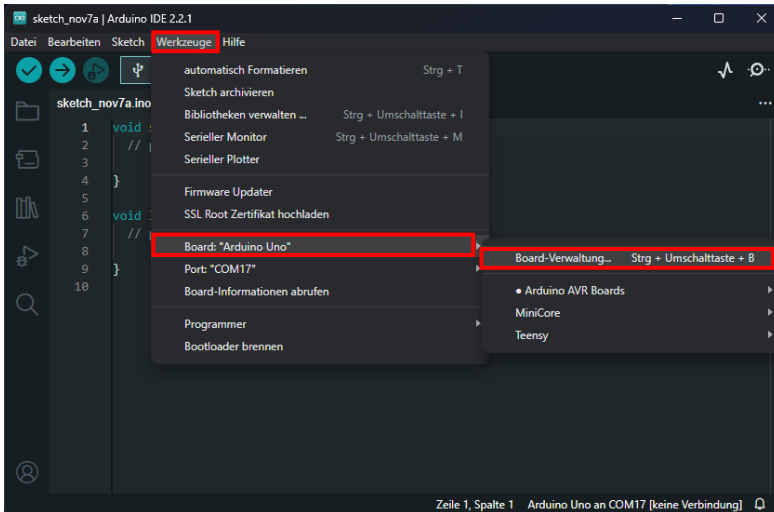


Fügen Sie dort die folgende Boardverwalter-URL hinzu und drücken anschließend auf OK:

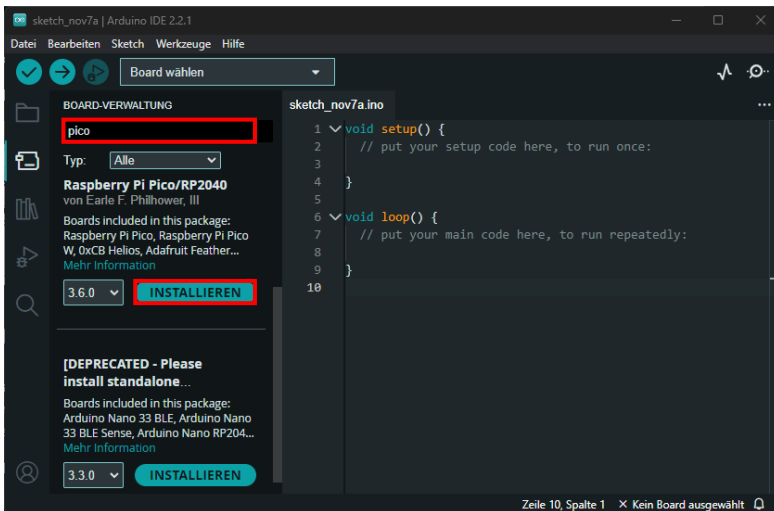
[https://github.com/earlephilhower/arduino-pico/releases/download/global/package\\_rp2040\\_index.json](https://github.com/earlephilhower/arduino-pico/releases/download/global/package_rp2040_index.json)



Nun müssen Sie den neuen Boardverwalter installieren. Gehen Sie dazu auf **Werkzeuge** -> **Board** -> **Board-Verwaltung...**



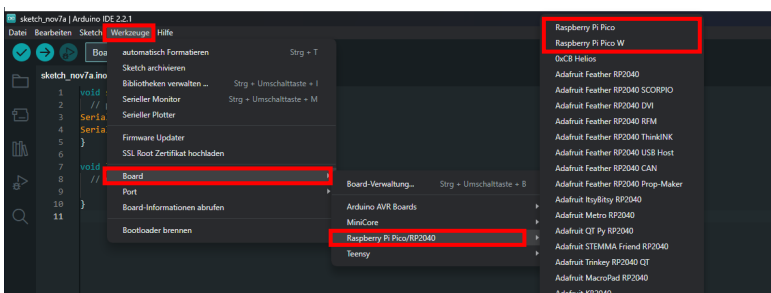
Geben Sie nun in der Suchleiste Pico ein und installieren Sie **Raspberry Pi Pico/RP2040** von Earle F. Philhower, III



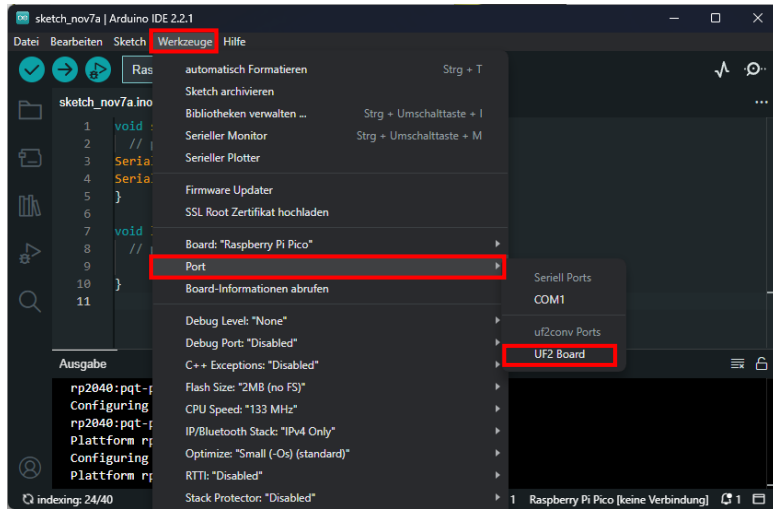
Für die Installation der passenden Firmware auf Ihrem Pico, halten Sie die **BOOTSEL** Taste gedrückt, während Sie den Pico über den microUSB-Port mit Ihrem PC verbinden, um den Pico in den Bootloader-Modus zu versetzen.

Beachten Sie, dass Sie immer nur eine Firmware auf den Pico aufspielen können. Wenn Sie den Pico mit MicroPython programmieren möchten, müssen Sie wieder eine andere Firmware aufspielen.

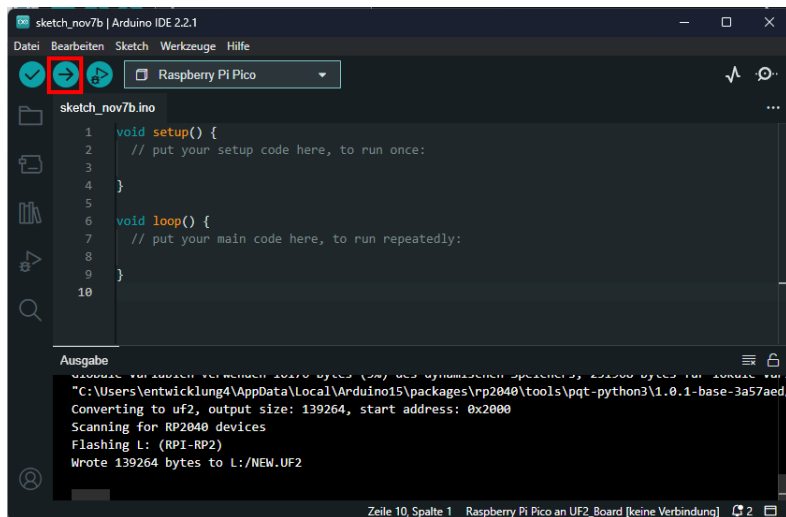
Wählen Sie nun unter **Werkzeuge** -> **Board** -> **Raspberry Pi Pico/RP2040** den **Raspberry Pi Pico** oder den **Raspberry Pi Pico W** aus, je nachdem welches Board Sie besitzen.



Unter Werkzeuge -> Port müssen Sie nun UF2 Board auswählen.



Anschließend können Sie durch aufspielen eines leeren Sketches den Bootloader auf Ihren Pico laden.



Ihr Pico ist nun einsatzbereit für die Verwendung mit der Arduino IDE.

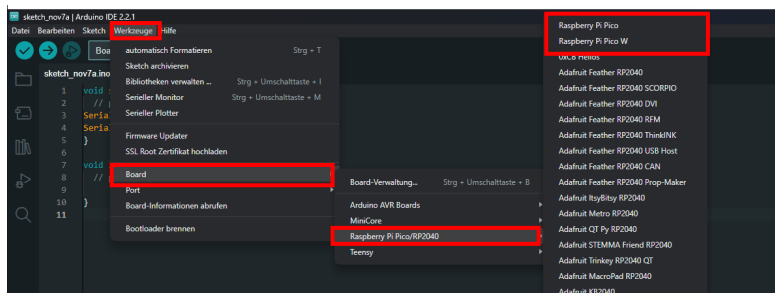
## 5. PROGRAMMIEREN DES PICOS MIT C++

Als einfaches Beispiel, kopieren Sie folgende Zeilen in die Arduino IDE:

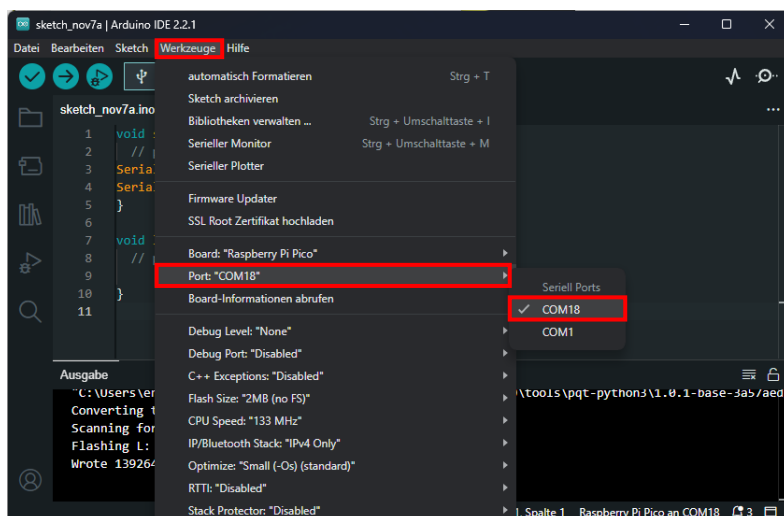
```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
  }
  Serial.println("Hello World!");
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

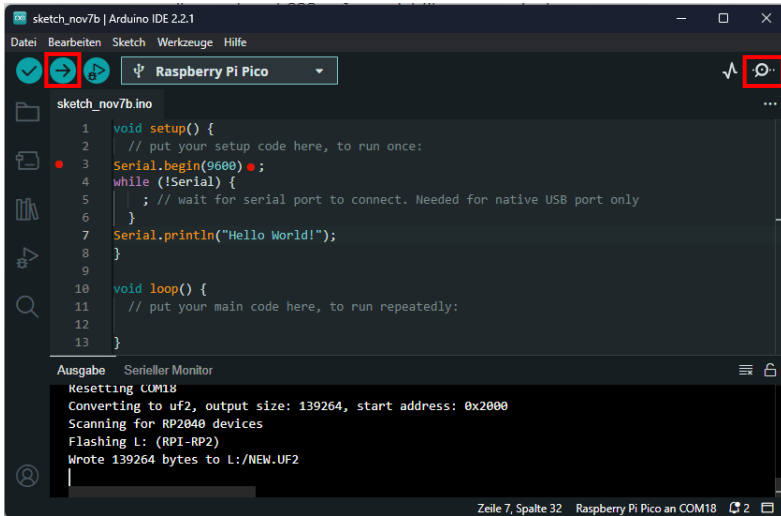
Bevor Sie das Beispiel auf Ihren Pico aufspielen können, müssen Sie zunächst sicherstellen, dass das richtige Board ausgewählt ist. Wählen Sie dazu unter **Werkzeuge** -> **Board** -> **Raspberry Pi Pico/ RP2040** den **Raspberry Pi Pico** oder den **Raspberry Pi Pico W** aus, je nachdem welches Board Sie besitzen.



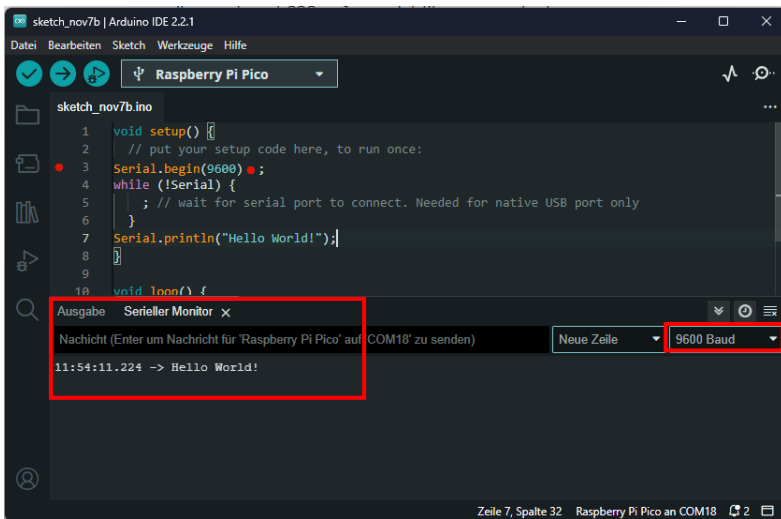
Danach muss noch der korrekte Port unter **Werkzeuge** -> **Port** ausgewählt werden. Sie können über den Gerätemanager Ihres PCs den richtigen Port herausfinden.



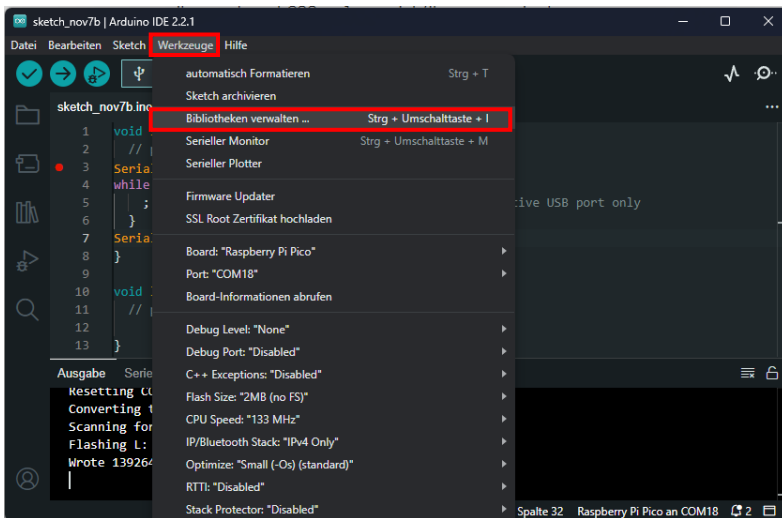
Klicken Sie anschließend auf Hochladen.  
Und öffnen Sie den seriellen Monitor.



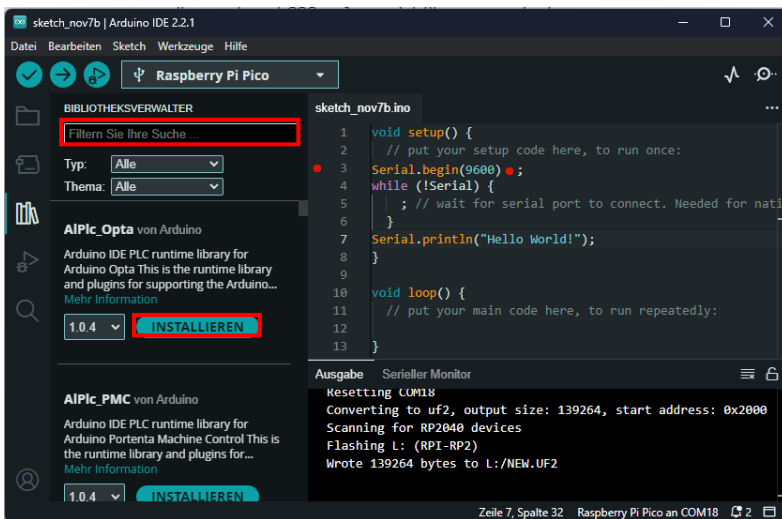
Im seriellen Monitor können Sie nun die Ausgabe des Programms sehen.  
Achten Sie darauf, dass Sie die richtige Baudrate eingestellt haben, in  
diesem Beispiel 9600 Baud.



Wenn Sie externe Bibliotheken hinzufügen möchten, klicken Sie auf **Werkzeuge** -> **Bibliotheken verwalten...**



Nun öffnet sich auf der linken Seite der Bibliotheksverwalter. Dort können Sie über das Suchfeld nach der gewünschten Bibliothek suchen, und dies durch klicken auf **Installieren** hinzufügen.



### 3. SONSTIGE INFORMATIONEN

Unsere Informations- und Rücknahmepflichten nach dem Elektroggesetz (ElektroG)



**Symbol auf Elektro- und Elektronikgeräten:**

Diese durchgestrichene Mülltonne bedeutet, dass Elektro- und Elektronikgeräte **nicht** in den Hausmüll gehören. Sie müssen die Altgeräte an einer Erfassungsstelle abgeben. Vor der Abgabe haben Sie Altbatterien und Altakkumulatoren, die nicht vom Altgerät umschlossen sind, von diesem zu trennen.

#### **Rückgabemöglichkeiten:**

Als Endnutzer können Sie beim Kauf eines neuen Gerätes, Ihr Altgerät (das im Wesentlichen die gleiche Funktion wie das bei uns erworbene neue erfüllt) kostenlos zur Entsorgung abgeben. Kleingeräte, bei denen keine äußere Abmessungen größer als 25 cm sind können unabhängig vom Kauf eines Neugerätes in haushaltsüblichen Mengen abgeben werden.

#### **Möglichkeit Rückgabe an unserem Firmenstandort während der Öffnungszeiten:**

SIMAC Electronics GmbH, Pascalstr. 8, D-47506 Neukirchen-Vluyn

#### **Möglichkeit Rückgabe in Ihrer Nähe:**

Wir senden Ihnen eine Paketmarke zu, mit der Sie das Gerät kostenlos an uns zurücksenden können. Hierzu wenden Sie sich bitte per E-Mail an [Service@joy-it.net](mailto:Service@joy-it.net) oder per Telefon an uns.

#### **Informationen zur Verpackung:**

Verpacken Sie Ihr Altgerät bitte transportsicher, sollten Sie kein geeignetes Verpackungsmaterial haben oder kein eigenes nutzen möchten kontaktieren Sie uns, wir lassen Ihnen dann eine geeignete Verpackung zukommen.

### 4. SUPPORT

Wir sind auch nach dem Kauf für Sie da. Sollten noch Fragen offen bleiben oder Probleme auftauchen, stehen wir Ihnen auch per E-Mail, Telefon und Ticket-Supportsystem zur Seite.

E-Mail: [service@joy-it.net](mailto:service@joy-it.net)

Ticket-System: <http://support.joy-it.net>

Telefon: +49 (0)2845 9360-50 (Mo - Do: 09:00 - 17:00 Uhr,  
Fr: 09:00 - 14:30 Uhr)

Für weitere Informationen besuchen Sie unsere Website:

[www.joy-it.net](http://www.joy-it.net)