

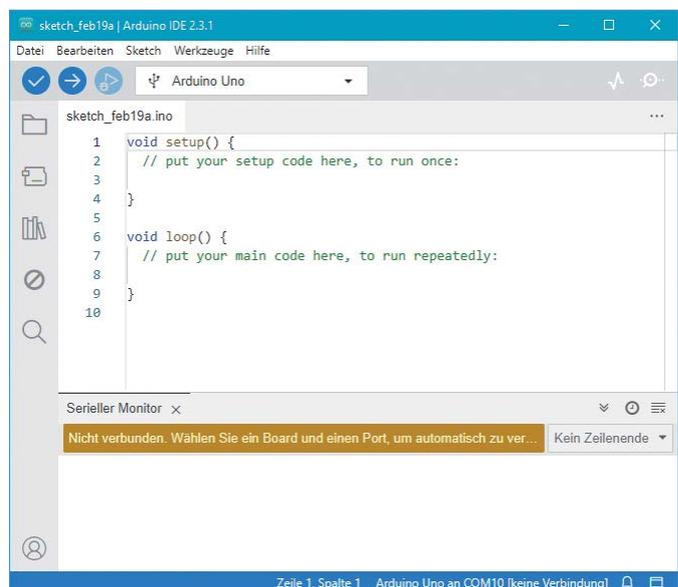
Die passende Entwicklungsumgebung

Anhängig von Ihren Fähigkeiten, Vorlieben und dem benutzten Mikrocontroller stehen unterschiedliche IDEs (Integrated Development Environment) zur Auswahl. So wie in diesem Heft wechselnde Entwicklungsboards benutzt werden, kommen auch mehrere Entwicklungsumgebungen zum Einsatz, um Ihnen zu einem ersten Zugang zu verhelfen.

Verständlicherweise kann keine umfassende Einführung erfolgen, sondern lediglich eine Einführung in die Installation und Nutzung. In den meisten Fällen können Sie auch eine andere Hard- oder Software nutzen, um die Projekte nachzubauen.

ARDUINO-IDE

Häufig werden wir hier im Heft die IDE von und für Arduino nutzen. Auch sie ist Teil des Erfolgs des Arduino Uno, denn sie bietet einen extrem schnellen und leichten Einstieg. Nach der Installation ist alles fix und fertig und es kann losgehen. Keine Treiber, fehlende Bibliotheken oder Zusatzprogramme müssen heruntergeladen und eingerichtet werden.



Die Arduino-IDE ist schlicht und einfach aber einsteigerfreundlich.

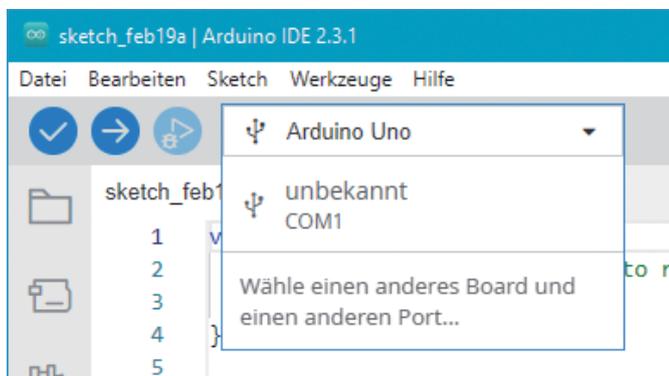
Auch die Programmierung in C wurde vereinfacht und Programme werden oft als *Sketch* bezeichnet und nicht mit dem Suffix `.gcc` oder `.c` sondern `.ino` gespeichert. Die Entwicklungsumgebung baut zwar auf dem freien Gnu C Compiler (GCC) auf, kapselt aber viel Technik in stark vereinfachten Funktionen. So werden Zugriffe auf Ports vereinfacht, PWM- und Analogsignale sind einfach zu nutzen und selbst auf die klassische Funktion `main()` wird verzichtet. Stattdessen gibt es die zwei vordefinierten Funktionen `setup()` und `loop()`.

Der Editor ist eher als rudimentär zu betrachten und bietet viele Funktionen nicht, die ein Profi-System anbietet. Das sorgt aber dafür, dass der Einstieg einfach ist und man nicht von den Möglichkeiten erschlagen wird.

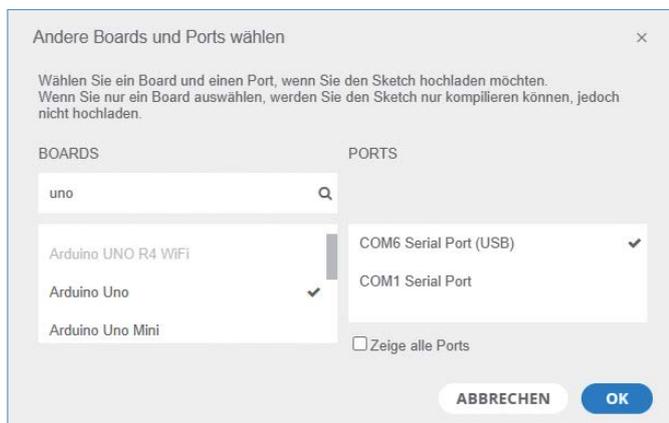


Zum unschätzbaren Schatz wird die Arduino-Welt, beim Blick auf die Vielfalt an vorgefertigten Bibliotheken. Mit der Einführung haben sich Firmen wie Adafruit oder SparkFun daran gemacht, Sensoren und andere praktische Schaltungen auf kleinen Modulen zu verkaufen, die bequem mit Jumperkabeln verbunden werden können. Dazu gibt es dann stets Bibliotheken, die den Zugriff so stark vereinfachen, dass nur wenige einfache Befehle notwendig sind. Inzwischen unterstützt die IDE nicht nur Arduinos, sondern kann auch genutzt werden, um viele andere Mikrocontroller zu programmieren. Die Bibliotheken sind oft so gestaltet worden, dass sie universell nutzbar sind, oder die Entwicklergemeinschaft hat speziell angepasste Versionen veröffentlicht.

- Über die Auswahl in der Symbolleiste am oberen Fensterrand können Sie einstellen, für welchen Controller der Programmcode übersetzt (kompiliert) werden soll.



- Klicken Sie auf **WÄHLE EINEN ANDERES BOARD...**, um die folgende Auswahl zu öffnen.
- Wählen Sie das passende Board und an welchem virtuellen COM-Port es angeschlossen ist, denn meistens erkennt die IDE das nicht automatisch.

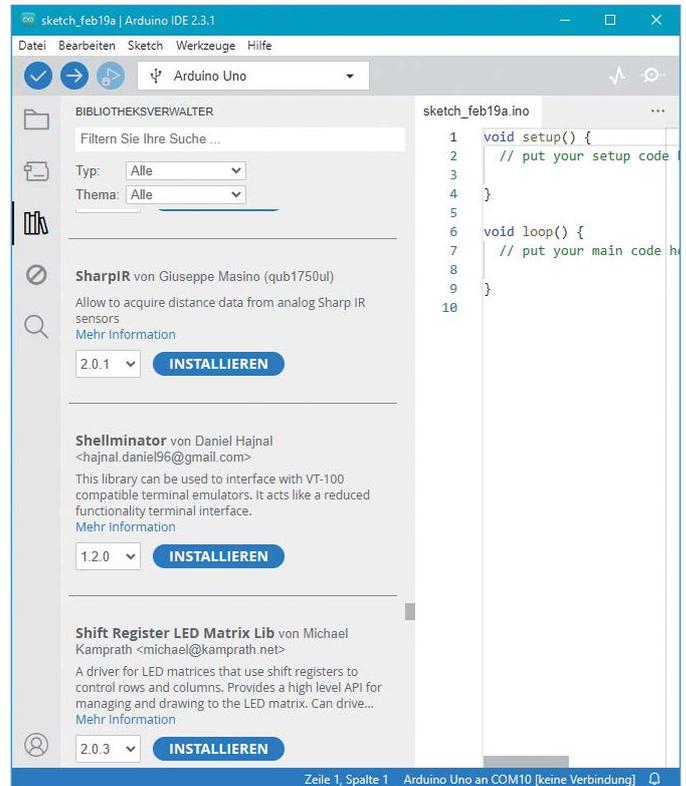


- Steht Ihr Board nicht zur Auswahl, dann können Sie es eventuell nachträglich installieren, in dem Sie **WERKZEUGE|BOARD...|BOARD-VERWALTUNG** öffnen.

Libraries bzw. Bibliotheken die in Ihrem Programm eingebunden werden, müssen oft bei der ersten Nutzung installiert werden. Die wichtigsten sind zwar schon von Anfang an dabei aber für spe-

zielle Sensoren oder Displays gibt es oft eine Auswahl mehrere Anbieter.

- Wählen Sie **WERKZEUGE|BIBLIOTHEKEN VERWALTEN**, so dass sich der angedockte Bereich mit der Bibliotheksverwaltung einblendet.



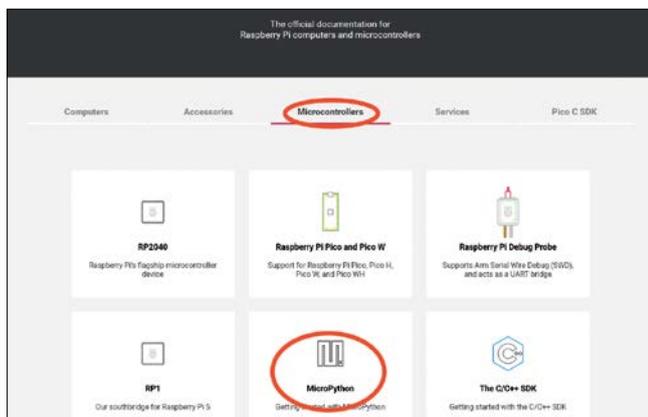
- Bei **Typ** können Sie Filtern welche Bibliotheken aufgelistet und gesucht werden sollen.
- Um eine neue Bibliothek zu installieren, geben Sie deren Namen in das Suchfeld ein.
- Klicken Sie bei den Ergebnissen auf **MEHR INFORMATIONEN**, wenn Sie die Webseite zur Bibliothek ansehen wollen.
- Manchmal ist es notwendig, eine bestimmte Version zu installieren — meistens ist aber die neueste die richtige Wahl. Wählen Sie die Version links neben der Schaltfläche **INSTALLIEREN** aus und klicken Sie auf **INSTALLIEREN**.

Mit dem Symbol eines nach rechts weisendem Pfeil am oberen Fensterrand kompilieren Sie Ihr Programm und übertragen es auf den angeschlossenen Mikrocontroller, wenn keine Fehler auftreten. In der Symbolleiste finden Sie oben rechts zwei Schaltflächen, um den Plotter und die serielle Ausgabe zu öffnen.

THONNY UND MICROPYTHON

MicroPython baut auf der Programmiersprache Python auf und besteht aus einer Laufzeitumgebung, die auf dem Mikrocontroller installiert wird und dort dann die Python-Befehle ausführt. Benutzen Sie einen Raspberry Pi und die OS Standard-Installation, dann ist Python und Thonny bereits installiert. Auf einem Entwicklungs-PC mit Windows muss Python installiert werden und auf dem Mikrocontroller in jedem Fall MicroPython.

1. Beginnen Sie damit, Python 3 (<https://www.python.org/>) auf dem PC zu installieren.
2. Installieren Sie die Thonny IDE (<https://thonny.org/>). Wenn Sie nicht wissen, für welche Version Sie sich entscheiden sollen, benutzen Sie den 32-Bit-Installer.
3. Drücken Sie auf dem Pi Pico die kleine Taste BOOTSEL und verbinden Sie das Board bei gedrückter Taste über den USB-Anschluss mit Ihrem Computer. Anschließend können Sie die Taste loslassen.
4. Öffnen Sie den Datei-Explorer und wechseln Sie auf das neu eingerichtete Laufwerk RPI-RP2, das der Pi Pico bereitstellt.
5. Dort finden Sie zwei Dateien. Klicken Sie doppelt auf **INDEX . HTM**, um die Datei im Browser als Webseite zu betrachten.

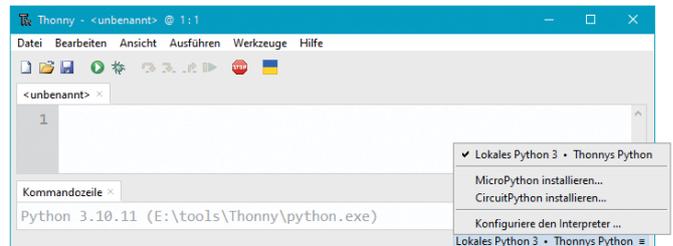


6. Wechseln Sie in die Rubrik **MICROCONTROLLERS** und klicken Sie auf **MICROPYTHON**.
7. Auf der sich öffnenden Seite werden zwei Downloads angeboten: für den Pi Pico ohne WLAN und den mit. Klicken Sie mit der rechten Maustaste auf den zu Ihrem Board passenden Link und wählen Sie aus dem Kontextmenü den Eintrag **ZIEL SPEICHERN UNTER** (oder ähnlich). Wählen Sie als Ziel das Laufwerk des Raspi aus. Nach dem Download wird der Raspi automatisch vom System abgemeldet.

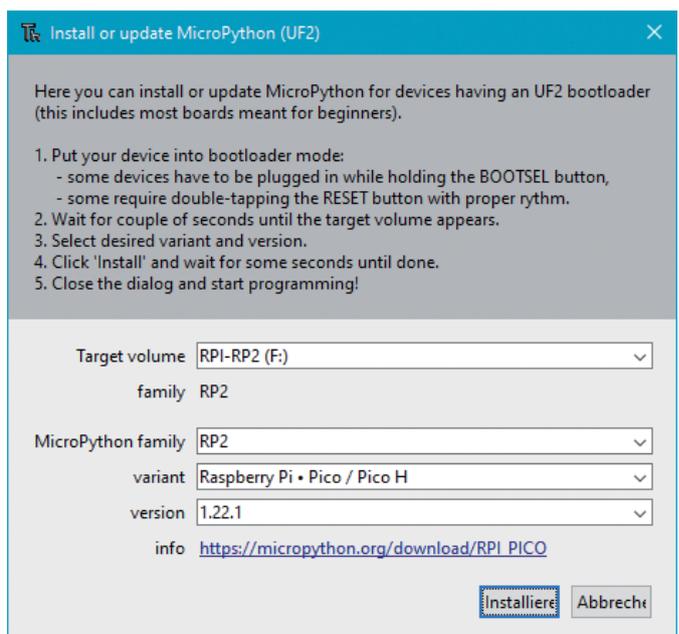


8. Trennen Sie den Raspberry vom USB-Port und verbinden Sie ihn bei gedrückter Taste **BOOTSEL** erneut, um wieder auf das Laufwerk des Raspi zugreifen zu können.

9. Starten Sie Thonny. Beim ersten Start können Sie die Sprache für die Benutzeroberfläche einstellen. Wir verwenden **DEUTSCH**.
10. Klicken Sie unten rechts am Fensterrand auf **LOKALES PYTHON...** und wählen Sie im Kontextmenü den Eintrag **MICROPYTHON INSTALLIEREN**.



11. Im nachfolgenden Dialogfenster wählen Sie ggf. das Laufwerk des Raspberry (sollte bereits ausgewählt sein) und wählen Sie bei **VARIANT** die passende Version für den Pi mit oder ohne WLAN.

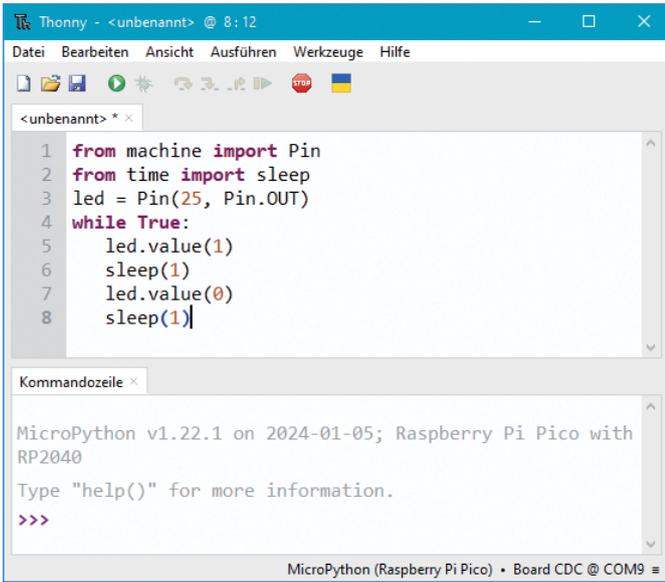


12. Nach dem Download und der Installation wird der Pi wieder vom System abgemeldet.

Jetzt haben Sie das System und den Raspi eingerichtet. Probieren Sie ein einfaches Beispiel, mit dem die LED auf dem Board zum Blinken gebracht wird:

1. Schließen Sie den Raspi Pi Pico per USB an Ihren PC an aber drücken Sie **nicht** die **BOTSEL**-Taste dabei. So wird kein Laufwerk erzeugt aber Thonny kann auf den Pi zugreifen.
2. Starten Sie Thonny. Am unteren rechten Fensterrand muss **MICROPYTHON...** angezeigt werden und im Bereich Kommandozeile sehen Sie eine Statusausgabe des Pi.





3. Geben Sie oberen Fensterbereich den Python-Code ein (achten Sie auf die Einrückungen einiger Zeilen):

```
from machine import Pin
from time import sleep
led = Pin(25, Pin.OUT)
while True:
    led.value(1)
    sleep(1)
    led.value(0)
    sleep(1)
```

4. Klicken Sie in der Symbolleiste auf das Diskettensymbol zum Speichern oder wählen Sie DATEI|SPEICHERN. In folgenden Fenster klicken Sie auf die Schaltfläche RASPBERRY PI PICO, um die Datei auf dem Pi zu speichern. Danach vergeben Sie einen Dateinamen für das Programm mit dem Suffix .pi — beispielsweise blink.pi.

AUTOSTART

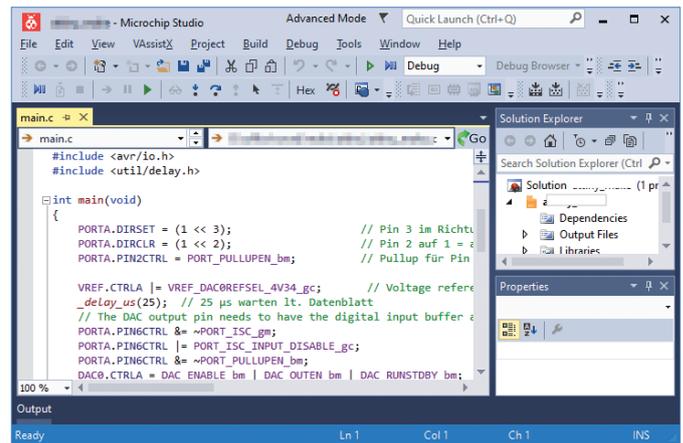
Die auf dem Pi gespeicherten Programme starten nicht automatisch, wenn das Board mit Strom versorgt wird. Wollen Sie ein Programm automatisch starten lassen, müssen Sie es main.py nennen.

5. Klicken Sie auf das grüne Play-Symbol oder benutzen Sie AUSFÜHREN|AKTUELLES SKRIPT AUSFÜHREN, um das Programm auf dem Pi zu starten, woraufhin die LED anfängt zu blinken.
6. Bevor Sie das Programm ändern, müssen Sie die Ausführung im Editor mit dem Stop-Symbol anhalten.

WEITERE IDES

Es gibt ein breites Spektrum an Entwicklungsumgebungen von einfach bis professionell und von kostenlos bis teuer. Neben den einfachen IDEs sind es vor allem individuelle Vorlieben, die Entwickler sich für eine Software entscheiden lassen. Hinzu kommen speziell benötigte Funktionen oder eine gute Integrationsmöglichkeit in die Entwicklung im Team etc. Im Rahmen dieses Specials können nur einige kurz angerissen werden.

Da ist zum Beispiel das Microchip Studio, welches früher Atmel Studio hieß: Dies ist die kostenlose Entwicklersoftware vom Hersteller der Atmel Mikrocontroller wie dem ATmega328 auf dem Arduino Uno. In dieser wird direkt mit GCC gearbeitet, was bedeutet, dass viele Annehmlichkeiten der Arduino-Welt erst einmal nicht verfügbar sind. Dafür ist man etwas näher an der Hardware dran und nutzt vor allem die Vorteile des konfigurierbaren Compilers. Das macht sich allein dann schon bemerkbar, wenn nicht bei jeder kleinen Codeänderung das ganze Projekt mit allen Abhängigkeiten minutenlang neu übersetzt wird.



Das Microchip Studio kann für Atmel-Familie genutzt werden.

Wenn Sie die Arduino-IDE installiert haben, dann können Sie auch in Microchip Studio wie in der Arduino-IDE programmieren und die Libraries nutzen, haben aber einen deutlich besseren Editor mit Projektverwaltung zur Hand.

Ein Mischmasch aus IDE und Framework für verschiedene Softwareentwicklungswerkzeugen stellt PlatformIO dar. Es vereint als Werkzeug diverse Editoren und IDEs unter einer Haube — hauptsächlich zur architekturübergreifenden Entwicklung für Embedded-Systeme. Als Editor kann beispielsweise Atom oder Eclipse eingebunden werden. Oft wird aber das bevorzugte Visual Studio Code benutzt, das sich auch als Einzellösung großer Beliebtheit erfreut. Auch die Arduino-IDE kann integriert werden, so dass alle Vorteile der einfachen Programmierung in einem deutlich leistungsfähigeren Editor nutzbar sind. Die erste Einrichtung und Nutzung ist für Einsteiger nicht ganz einfach, da zwischen Editor und IDE unterschieden und (oft unauffällig) gewechselt wird bzw. unterschiedliche Menüs benutzt werden müssen.