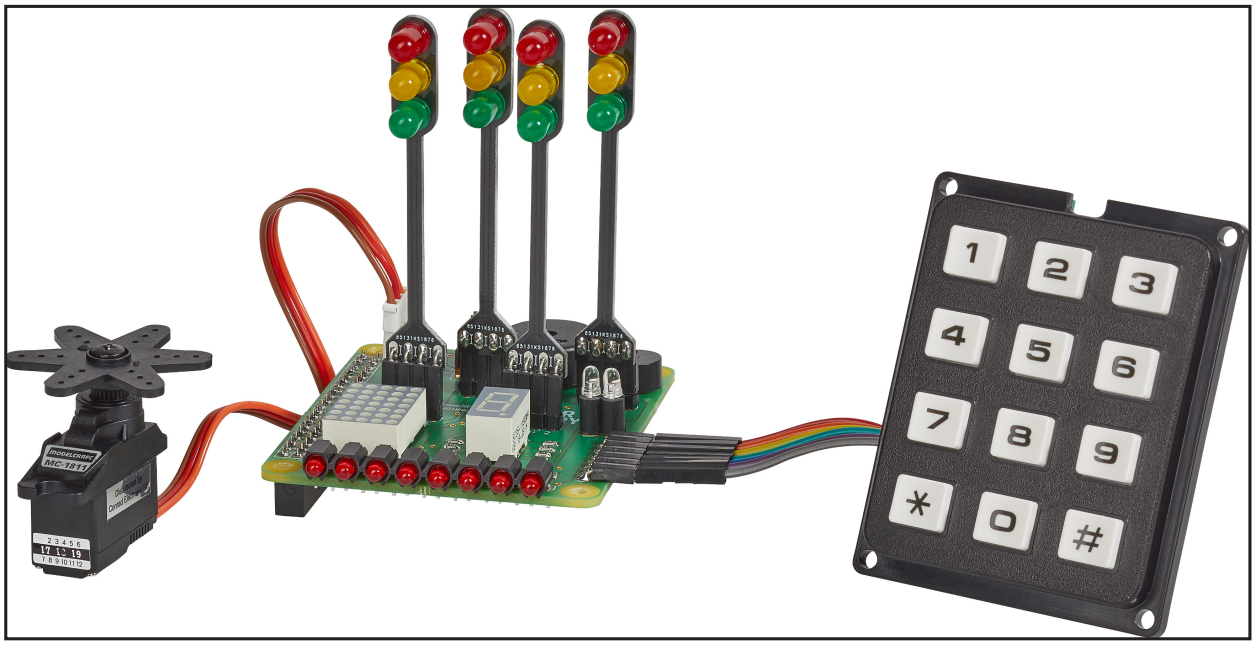
MAKER FACTORY

Leseprobe

Ⓓ Bedienungsanleitung

Raspberry Pi® Lernpaket Programmierung „MF375“

Best.-Nr. 1893842



Inhaltsverzeichnis

D

Seite

1. Einführung

Sehr geehrte Kundin, sehr geehrter Kunde,

wir bedanken uns für den Kauf dieses Produkts.

Dieses Produkt entspricht den gesetzlichen, nationalen und europäischen Anforderungen.

Um diesen Zustand zu erhalten und einen gefahrlosen Betrieb sicherzustellen, müssen Sie als Anwender diese Bedienungsanleitung beachten!

Diese Bedienungsanleitung gehört zu diesem Produkt. Sie enthält wichtige Hinweise zur Inbetriebnahme und Handhabung. Achten Sie hierauf, auch wenn Sie dieses Produkt an Dritte weitergeben.

Heben Sie deshalb diese Bedienungsanleitung zum Nachlesen auf!

Alle enthaltenen Firmennamen und Produktbezeichnungen sind Warenzeichen der jeweiligen Inhaber. Alle Rechte vorbehalten.

Bei technischen Fragen wenden Sie sich bitte an:

Deutschland: www.conrad.de/kontakt

Österreich: www.conrad.at
www.business.conrad.at

Schweiz: www.conrad.ch
www.biz-conrad.ch

2. Symbol-Erklärung

Das Symbol mit dem Ausrufezeichen im Dreieck weist auf wichtige Hinweise in dieser Bedienungsanleitung hin, die unbedingt zu beachten sind.

Das Pfeil-Symbol ist zu finden, wenn Ihnen besondere Tipps und Hinweise zur Bedienung gegeben werden sollen.

3. Bestimmungsgemäße Verwendung

Das Produkt ist eine Aufsteckplatine für den Raspberry Pi® mit 40-poliger Stiftleiste (Aufsteckplatinen werden beim Raspberry Pi® als „HAT“ bezeichnet). Die Platine muss im spannungs-/stromlosen Zustand auf den Raspberry Pi® gesteckt werden.

Die Platine dient als Grundlage für verschiedene Programmier-Beispiele. Diese sind gedacht für Jugendliche in Schulen oder zu Hause, um spielerisch mit einfachen LEDs das Programmieren zu erlernen.

Die Funktionsweise des Produkts ist in der Bedienungsanleitung anhand von zahlreichen Beispielen genau erklärt. Zur Programmierung dient Python 3. Diese ist bereits bei Raspbian vorinstalliert.

Das gesamte Produkt darf nicht geändert bzw. umgebaut werden! Die Sicherheitshinweise sind unbedingt zu beachten!

Falls Sie das Produkt für andere Zwecke verwenden, als zuvor beschrieben, kann das Produkt beschädigt werden. Außerdem kann eine unsachgemäße Verwendung Gefahren wie z.B. Kurzschluss, Brand, Stromschlag etc. hervorrufen. Lesen Sie sich die Bedienungsanleitung genau durch und bewahren Sie diese auf. Reichen Sie das Produkt nur zusammen mit der Dokumentation an dritte Personen weiter.

Dieses Produkt erfüllt die gesetzlichen, nationalen und europäischen Anforderungen. Alle enthaltenen Firmennamen und Produktbezeichnungen sind Warenzeichen der jeweiligen Inhaber. Alle Rechte vorbehalten.

HDMI ist eine eingetragene Marke der HDMI Licensing L.L.C.

4. Lieferumfang

- Aufsteckplatine mit
  - 8-fach LED-Reihe
  - 5*7 Punkt-Matrix-Anzeige
  - Miniatur-Summer
  - Sieben-Segment-Anzeige
  - 2x Blaulichter
  - Piezo-Signalgeber
  - Schalter
  - Taster
- 4x Abstandshalter
- 4x Ampeln
- Tastenfeld
- Modellbau-Servo
- Bedienungsanleitung mit Programmierbeispielen für Python

Aktuelle Bedienungsanleitungen

Laden Sie aktuelle Bedienungsanleitungen über den Link www.conrad.com/downloads herunter oder scannen Sie den abgebildeten QR-Code. Befolgen Sie die Anweisungen auf der Webseite.

Zum Betrieb sind noch folgende Bestandteile erforderlich, die sich nicht im Lieferumfang befinden:

- Raspberry Pi® mit 40-poliger Stiftleiste (aber keinen „Zero“, dafür passt die Aufsteckplatine von der Größe her nicht)
- MicroSD-Karte (mindestens 8 GByte)
- Stromversorgung, z.B. ein USB-Steckernetzteil und ein MicroUSB-Kabel
- Maus, Tastatur, Monitor mit HDMI-Kabel

Zubehör/Ersatzteile:

MicroSD-Karte fertig mit dem Betriebssystem Noobs 3B+:	Conrad-Bestell-Nr.: 1673078
Tastenfeld als Ersatz:	Conrad-Bestell-Nr.: 709840
Modellbau-Servo als Ersatz:	Conrad-Bestell-Nr.: 275460

→ Passendes Zubehör finden Sie im Internet auf der zugehörigen Produktseite.

5. Sicherheits-und Gefahrenhinweise

Lesen Sie sich die Bedienungsanleitung aufmerksam durch und beachten Sie insbesondere die Sicherheitshinweise. Falls Sie die Sicherheitshinweise und die Angaben zur sachgemäßen Handhabung in dieser Bedienungsanleitung nicht befolgen, übernehmen wir für dadurch resultierende Personen-/Sachschäden keine Haftung. Außerdem erlischt in solchen Fällen die Gewährleistung/Garantie.

Sehr geehrte Kundin, sehr geehrter Kunde,

diese Sicherheitshinweise dienen nicht nur dem Schutz des Produkts, sondern auch Ihrer eigenen Sicherheit und der Sicherheit anderer Personen. Lesen Sie sich deshalb dieses Kapitel sehr aufmerksam durch, bevor Sie das Produkt in Betrieb nehmen!

- Das Produkt ist kein Spielzeug. Halten Sie es von Kindern und Haustieren fern.
- Lassen Sie das Verpackungsmaterial nicht achtlos liegen. Dieses könnte für Kinder zu einem gefährlichen Spielzeug werden.
- Schützen Sie das Produkt vor extremen Temperaturen, direktem Sonnenlicht, starken Erschütterungen, hoher Feuchtigkeit, Nässe, brennbaren Gasen, Dämpfen und Lösungsmitteln.
- Setzen Sie das Produkt keiner mechanischen Beanspruchung aus.
- Achten Sie darauf, dass Bauteile oder Lötkontakte nicht mit metallischen Teilen in Berührung kommen und somit Kurzschlüsse entstehen. Dabei wird das Produkt beschädigt, Verlust von Gewährleistung/Garantie!
- Wenn kein sicherer Betrieb mehr möglich ist, nehmen Sie das Produkt außer Betrieb und schützen Sie es vor unbeabsichtigter Verwendung. Der sichere Betrieb ist nicht mehr gewährleistet, wenn das Produkt:
  - sichtbare Schäden aufweist,
  - nicht mehr ordnungsgemäß funktioniert,
  - über einen längeren Zeitraum unter ungünstigen Umgebungsbedingungen gelagert wurde oder
  - erheblichen Transportbelastungen ausgesetzt wurde.
- Gehen Sie vorsichtig mit dem Produkt um. Durch Stöße, Schläge oder dem Fall aus bereits geringer Höhe wird es beschädigt.
- Wenden Sie sich an eine Fachkraft, wenn Sie Zweifel über die Arbeitsweise, die Sicherheit oder den Anschluss des Produktes haben. Lassen Sie Wartungs-, Anpassungs- und Reparaturarbeiten ausschließlich von einem Fachmann bzw. einer Fachwerkstatt durchführen.
- Sollten Sie noch Fragen haben, die in dieser Bedienungsanleitung nicht beantwortet werden, wenden Sie sich an unseren technischen Kundendienst oder an andere Fachleute.

6. Inbetriebnahme

Trennen Sie zunächst alle Bestandteile von der Spannungs-/Stromversorgung (z.B. den von Ihnen verwendeten Raspberry Pi).

Montieren Sie zuerst die 4 Abstandshalter an den Raspberry Pi®. Stecken Sie dann die Aufsteckplatine auf die 40polige Stiftleiste und schrauben Sie die Abstandshalter fest.

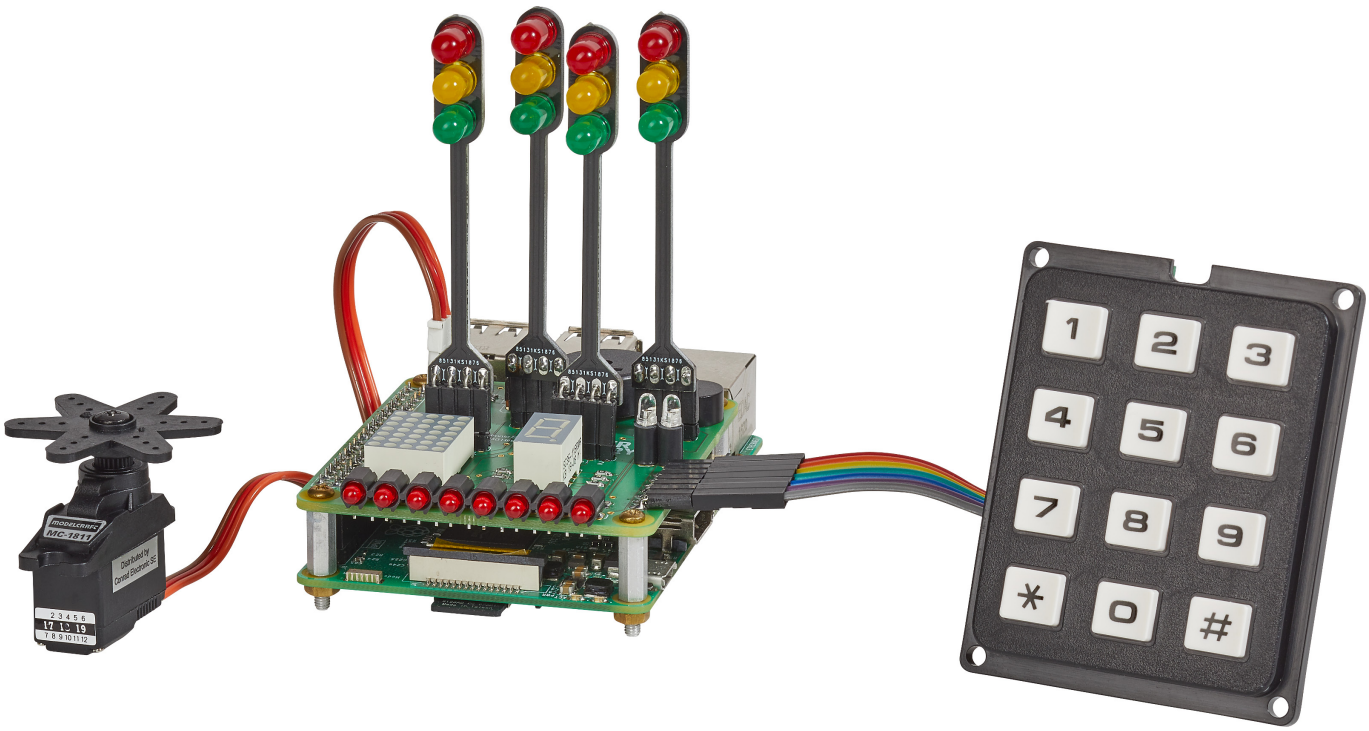
Der Baustein enthält die Halterung für vier Ampeln. Die Ampeln selbst können jederzeit herausgezogen und wieder aufgesteckt werden. Durch den Vorwiderstand und die Dioden sind diese auch verpolungssicher.

→ Die Ampeln werden so aufgesteckt, dass diese in Richtung der LED-Reihe zeigen.

Die Ampeln sind für 3,3 V/DC ausgelegt, werden aber auch bei Anlegen von 5 V/DC noch nicht zerstört (der Raspberry Pi hat 2 Versorgungspins, an denen jeweils 5 V/DC anliegen).

Schließen Sie dann das Tastenfeld mit dem beiliegenden Kabel an (siehe Kapitel 14) und danach das Modellbau-Servo (siehe Kapitel 18).

Das nachfolgende Bild zeigt die Platine, die auf einen Raspberry Pi® (nicht im Lieferumfang) aufgesteckt ist:



7. Einrichten des Raspberry Pi

Um einen Raspberry Pi® in Betrieb zu nehmen, müssen Sie ein Betriebssystem herunterladen und auf die MicroSD-Karte kopieren.

Zum Nachprogrammieren der Beispiele benötigen Sie „Python 3“.

Empfohlen wird von unserer Seite her das Betriebssystem „Raspbian“. Dies ist sehr einfach einzurichten und beinhaltet neben Python, Scratch und Java schon sehr viele Module, wie z.B. Browser, Textverarbeitungsprogramme und Zubehör wie den Taschenrechner u.v.m.

→ Welche Programmiersprache man lernt ist eigentlich zweitrangig -wichtiger ist, dass man von Anfang an lernt strukturiert zu programmieren. Jede andere Sprache wie z.B. C oder Java kann leicht gelernt werden, wobei auch hier die Strukturierung und auch die Kommentare sehr wichtig sind. Spätestens wenn andere das Programm weiter entwickeln wollen oder man selbst nach längerer Zeit das Programm wieder „verstehen“ will.

Es gibt schon zahlreiche Beispiele und Internetseiten, wie man das Betriebssystem auf einem Raspberry Pi® einrichtet. Wir empfehlen, dass Sie sich die Software „NOOBS“ herunterladen, damit lässt sich Raspbian sehr einfach installieren.

Alternativ laden Sie sich direkt das Raspbian-Image herunter: https://www.raspberrypi.org/downloads/

Wenn Sie die gezippte Version heruntergeladen haben, dann kopieren Sie die entpackten Dateien (z.B. mit dem Windows Explorer) auf eine MicroSD-Karte mit mindestens 8GByte.

→ Optional ist auch eine 16GByte-MicroSD-Karte fertig mit dem Betriebssystem NOOBS bei Conrad Electronic SE erhältlich; Sie finden diese unter der Bestell-Nr. 1673078.

Anschließend stecken Sie die MicroSD-Karte in einen Raspberry Pi®, schließen Maus, Tastatur und Monitor an und schalten die Spannungs-/Stromversorgung ein. Nach der Installation von Raspbian kann man z.B. die Sprache auf Deutsch einstellen und die Zeitzone auf „Europa/Berlin“ ändern.

Bei Bedarf können Sie noch die Darstellung der Startleiste anpassen.

Für die in dieser Bedienungsanleitung verwendeten Beispiele selbst benötigen Sie keine Internetverbindung. Zum Updaten des Betriebssystems und zur Aktualisierung der Treiber ist jedoch eine Internetverbindung erforderlich.

Die Programme mit den Bibliotheken „RPi.GPIO“ und „GPIO Zero“ funktionieren nach dem Start des Raspberry Pi. Bei „pigpio“ ist das anders: Diese benötigt das Dienstprogramm „pigpiod“, das die pigpio-Bibliothek als Daemon startet. Nach dem Start läuft die pigpio-Bibliothek im Hintergrund und akzeptiert die Befehle.

Um diese Datei automatisch beim Booten der Raspberry Pi zu starten, muss im LX-Terminal folgender Befehl eingegeben werden:

sudo systemctl enable pigpiod

Danach kann man einen Neustart machen oder man startet das Programm sofort mit dem Befehl:

sudo systemctl start pigpiod

8. Erklärung der GPIOs

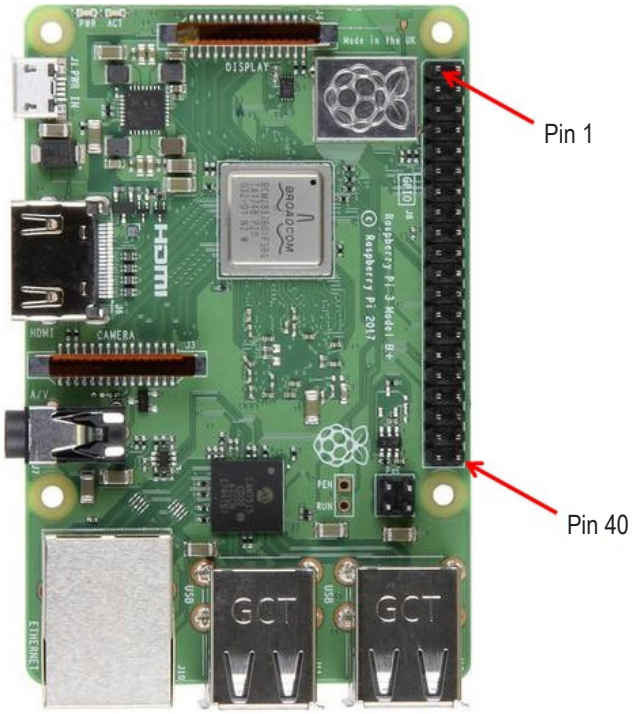
Auf dem Raspberry Pi® befindet sich eine 40polige Stiftleiste. Die Platine wird auf die diese Stiftleiste des Raspberry Pi® aufgesteckt.

An diesen 40 Pins findet man eine Betriebsspannung von 3,3 V/DC, 5 V/DC, einige GND-Pins und verschiedene programmierbare Pins, die so genannten GPIOs.

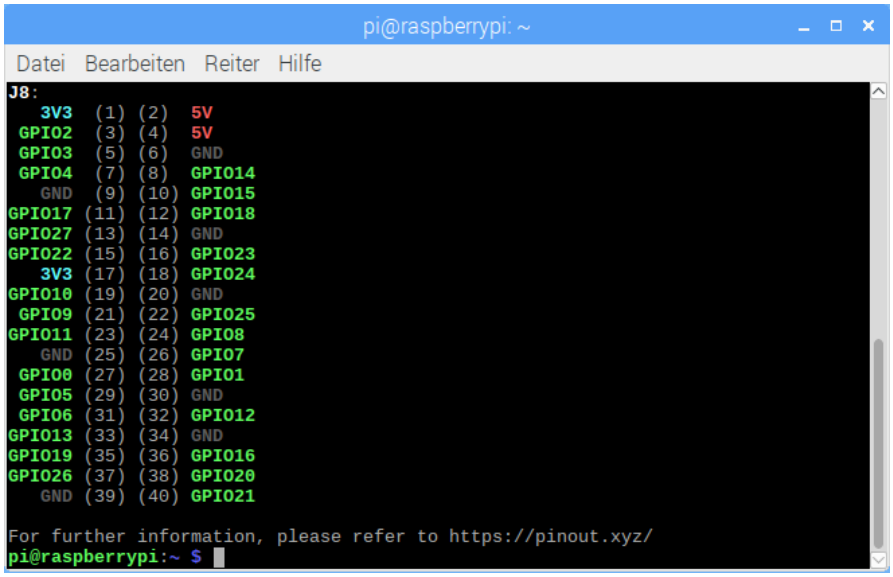
„GPIO“ steht für „General Purpose Input Output“ (= Ein-/Ausgänge zur allgemeinen Verwendung) und 26 von diesen Ports können als Eingang oder als Ausgang geschaltet werden. Diese Ports vom Controller können als Ausgänge auf High gesetzt werden mit einer Spannung von 3,3 V/DC oder als Low mit 0 V/DC. Diese Ports können aber auch als Eingang eingelesen werden und liefern „High“ oder „Low“ je nach Zustand.

Übersicht der GPIOs:

GPIO	Pin	Pin	GPIO
3,3 V	1	2	5 V
GPIO 2 (SDA1)	3	4	5 V
GPIO 3 (SCL1)	5	6	GND
GPIO 4 (CLK0)	7	8	GPIO 14 (TXD0)
GND	9	10	GPIO 15 (RXD0)
GPIO 17	11	12	GPIO 18 (PCM)
GPIO 27	13	14	GND
GPIO 22	15	16	GPIO 23
3,3 V	17	18	GPIO 24
GPIO 10 (MOSI)	19	20	GND
GPIO 9 (MISO)	21	22	GPIO 25
GPIO 11 (SCLK)	23	24	GPIO 8 (CE0)
GND	25	26	GPIO 7 (CE1)
(GPIO 0) ID_SD	27	28	(GPIO 1) ID_SC
GPIO 5	29	30	GND
GPIO 6	31	32	GPIO 12
GPIO 13	33	34	GND
GPIO 19	35	36	GPIO 16
GPIO 26	37	38	GPIO 20
GND	39	40	GPIO 21



→ Eine Übersicht der GPIOs sieht man auch, wenn man im LX-Terminal „pinout“ eingibt.



Die GPIOs kann man mit verschiedenen Software-Tools wie z.B. Python programmieren (in dieser Bedienungsanleitung verwendet).

Fast alle GPIOs besitzen einen internen Pullup- bzw. Pulldown-Widerstand, den man in seinem Programm ein- und ausschalten kann. Mit diesen intern zuschaltbaren Widerständen kann der Eingangspegel auf einen definierten Zustand geschaltet werden, so dass ein für den Prozessor eindeutiger Zustand vorliegt (Pull-down = Pin über Widerstand auf GND/Masse; Pull-Up: Pin über Widerstand auf 3,3 V/DC = High).

⟶ Mit den Widerständen wird außerdem der Strom begrenzt. Ohne einen solchen Widerstand könnte ein zu hoher Strom fließen und die Bauteile beschädigen. Deswegen sind auf der Platine absichtlich externe Widerstände integriert.

Ansteuerung der GPIOs

Die Ansteuerung der GPIOs über Python kann über verschiedene Arten von Bibliotheken erfolgen. Nachfolgend die möglichen Bibliotheken zur Ansteuerung der GPIOs über Python:

„wiringPi“

Diese Bibliothek ist schon etwas älter und wird bei unseren Beispielen nicht verwendet.

„RPi.GPIO“

Die Bibliothek „RPi.GPIO“ ist eine sehr robuste Bibliothek, die gerne von Anfängern benutzt wird.

„GPIO Zero“

Diese Bibliothek ist intuitiv und optimiert den Python-Code, d.h. der Code wird dadurch sehr kurz und ist deshalb auch leicht lesbar.

„pigpio“

Diese Bibliothek schaltet alle GPIOs über die Hardware und ist daher geeignet für zeitkritische Programme. Und sie besitzt die Fähigkeit, die GPIO-Pins sogar über das Netzwerk zu steuern. pigpio kann manuell installiert werden mit dem Befehl: **sudo apt install pigpio**

Vergessen Sie nicht, das Programm „pigpiod“ zu starten (siehe Kapitel 7).

⟶ Die Beispiele in dieser Bedienungsanleitung sind in Python 3 geschrieben und getestet. Es ist möglich, dass manche Programmbeispiele auch mit anderen Versionen von Python funktionieren, dies kann aber natürlich nicht garantiert werden.

Kopieren Sie sich am besten unter „Entwicklung“ das Icon von „Python 3 (IDLE)“ auf den Desktop. Neben „IDLE“ ist bei NOOBS seit kurzer Zeit auch die neue Version der „Thonny Python IDE“ enthalten, die mit dem „Step“-Befehl den Code sogar Schritt für Schritt durchgehen kann.

Im Kapitel 20. a) ist das Programm „Alles_aus.py“. Vor Beginn jeder Übung sollte dieses Programm immer einmal aufgerufen werden, um alle Ports in einen definierten Zustand (Input) zu bringen.

Andernfalls müssten in jedem Programm jedes Mal alle Pins definiert werden.

⟶ Alle Python-Programme in dieser Anleitung könnte man natürlich ganz leicht kopieren, in den Editor einfügen und dann starten. Dabei geht jedoch der Lerneffekt verloren, beim Programmieren sauber zu arbeiten und Fehler zu vermeiden (bzw. auftretende Fehler selbst zu finden und zu beseitigen). Wir empfehlen, mindestens zwei bis drei Programme selbst einzugeben, um sich mit dem System vertraut zu machen.

Sehen Sie die nachfolgenden Programme außerdem nur als eine der möglichen Lösungen an. Nachdem Sie die Beispielprogramme durchgearbeitet und ihre Funktionen verstanden haben, können Sie diese abändern oder miteinander kombinieren, um die Möglichkeiten der Aufsteckplatine in Verbindung mit einem Raspberry Pi® zu erkunden. Optimal ist es, wenn Sie eigene Lösungen suchen, anstatt vorgefertigte Programme zu verwenden.

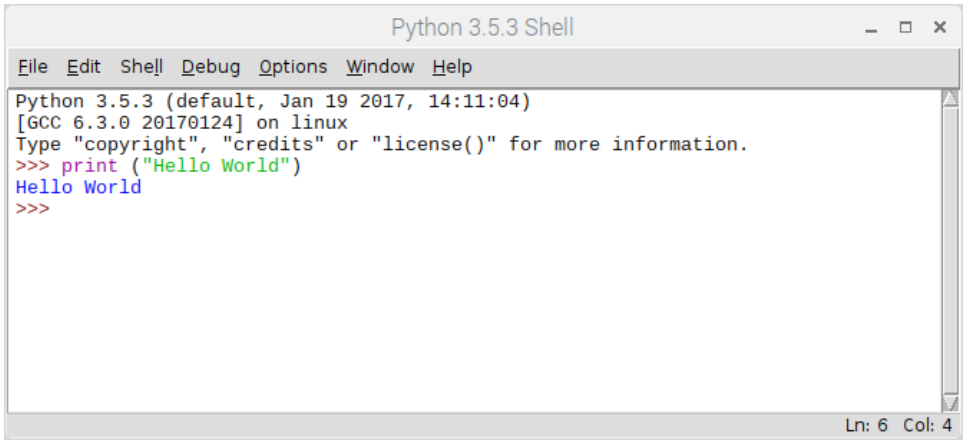
9. Beispiel-Programme: Kreuzung mit RPi.GPIO

Starten Sie Python 3 mit einem Doppelklick auf das Symbol am Desktop. Daraufhin öffnet sich das Shell-Entwicklungsfenster. Führen Sie einen kurzen Test durch, indem Sie das bekannte „Hello World“ ausgeben lassen.

Geben Sie ein:

```
print ("Hello World")
```

Sobald die Eingabe mit der Enter-Taste abgeschlossen wird, erscheint das „Hello World“ als Ausgabe im Fenster.



Python besitzt auch ein Hilfesystem. Geben Sie ein:

```
print (
```

Sobald die Klammer eingegeben wird, erscheint die Erklärung zu dem Befehl. Dies funktioniert auch bei anderen Befehlen.

Gewöhnen Sie sich von Anfang an daran, mit dem Editor zu arbeiten. Öffnen Sie eine neue Datei über „File / New File“. Nun erscheint ein neues Fenster zur Eingabe des Codes.

⟶ Die neu angelegte Datei besitzt beim Abspeichern die Endung „.py“ (für Python) und wird normalerweise unter „/home/pi“ im Hauptverzeichnis abgelegt.

a) Programm „Blinklichter“

Eine Ampel hat drei Leuchtdioden (rot, gelb und grün).

⟶ Wie eine LED funktioniert, ist in Kapitel 11. a) erklärt.

In dem ersten Beispiel wollen wir eine LED der Ampel ein- und ausschalten. Im Programm werden zuerst die Deklarationen festgelegt, also z.B. die benötigten Bibliotheken importiert und die Variablen festgelegt.

In unserem ersten Beispiel wollen wir die Ausgänge mit „RPi.GPIO“ ansteuern:

```
import RPi.GPIO as GPIO
```

Es gibt hier zwei Arten, wie man die Ausgänge bezeichnet: „BCM“, wenn die GPIOs verwendet werden; „BOARD“, wenn die Pin-Nummern verwendet werden sollen. Wir wollen die GPIOs verwenden:

```
GPIO.setmode(GPIO.BCM)
```

Um am Anfang nicht zu sehr verwirrt zu werden, wollen wir die Warnungen ausschalten:

```
GPIO.setwarnings(False)
```

Mit dem nächsten Befehl importieren wir die Bibliothek „time“, die z.B. für Warteschleifen oder Zeitmessungen verwendet werden kann:

```
import time
```

Zum Programmieren der Ampel müssen wir erst mal wissen wo die LEDs angeschlossen sind:

	Rot	Gelb	Grün
Ampel 1	GPIO17	GPIO27	GPIO22
Ampel 2	GPIO11	GPIO9	GPIO10
Ampel 3	GPIO25	GPIO8	GPIO7
Ampel 4	GPIO18	GPIO15	GPIO14

Zum Schalten der roten LED von Ampel 1 an Pin 11 (= GPIO 17) gibt es nun zwei Möglichkeiten:

```
GPIO.setmode (GPIO.BOARD)
GPIO.setup (11, GPIO.OUT)
GPIO.output (11, GPIO.HIGH)
```

oder:

```
GPIO.setmode (GPIO.BCM)
GPIO.setup (17, GPIO.OUT)
GPIO.output (17, GPIO.HIGH)
```

Erklärung:

Mit dem Befehl **GPIO.setup()** wird einem Port zugewiesen, ob dieser ein Ausgang (Output) oder ein Eingang (Input) ist.

Mit **GPIO.setup (17, GPIO.OUT)** legen wir demnach GPIO17 als Ausgang fest. Anschließend können wir mit dem Befehl **GPIO.output()** den Ausgang auf HIGH (1) oder LOW (0) setzen:

Auf HIGH setzen:

```
GPIO.output (17, GPIO.HIGH)
```

oder

```
GPIO.output (17, 1)
```

Auf LOW setzen:

```
GPIO.output (17, GPIO.LOW)
```

oder

```
GPIO.output (17, 0)
```

Nun wollen wir die rote LED der ersten Ampel ein- und wieder ausschalten.

Wichtiger Hinweis!

In Python müssen Befehle, die in Kontrollstrukturen wie Abfragen oder Schleifen organisiert sind, immer eingerückt werden (mit Tab-Stopps oder mit Leerzeichen, nicht gemischt verwenden). Wenn dies nicht eingehalten wird, dann läuft das Programm nicht mehr (oder nicht mehr richtig) und entsprechende Fehlermeldungen werden ausgegeben.

Der Vorteil beim Einrücken ist, dass das Programm übersichtlicher wird, außerdem erkennt man auf einen Blick fehlerhafte Kontroll-strukturen.

Gewöhnen Sie sich an, bestimmte Bereiche des Programms mit Kommentaren zu versehen. Dies erleichtert es, Programme auch nach längerer Zeit noch zu verstehen, was genau passiert. Kommentare beginnen mit einem Doppelkreuz (Hash) #.

Hier die ersten Erkenntnisse:

- Blöcke muss man Einrücken
- Kommentar beginnen mit einem Doppelkreuz (Hash) #

So sieht unser erstes Programm aus:

Programm „Ampel_1.py“:

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
import time

# Hier startet das Programm
GPIO.setup(17, GPIO.OUT)
while True:
    GPIO.output (17,1)
    time.sleep (1)
    GPIO.output (17,0)
    time.sleep (1)
```

Geben Sie das Programm im Editor ein und speichern Sie es ab (z.B. als „Ampel_1“).

Erklärung der Funktion:

- Nachdem der GPIO17 als Ausgang definiert wird, läuft das Programm endlos in der „while true“-Schleife, da diese ja immer „wahr“ ist und nie falsch werden kann.
- Mit dem Befehl **GPIO.output (17,1)** wird der Ausgang eingeschaltet.
- Bei der Funktion **time.sleep()** wartet das Programm für die in Klammern eingegebene Zeit in Sekunden.
- Danach wird der Ausgang und damit die rote LED wieder ausgeschaltet und erneut die angegebene Zeit gewartet.
- Das Programm springt am Ende zurück an die Stelle zum Einschalten des GPIO17: **GPIO.output (17,1)**.

Starten Sie das Programm mit der F5-Taste. Jetzt blinkt die rote LED der Ampel 1.

Zum Abbruch aktivieren wir wieder den Editor IDLE und drücken die Tastenkombination „Strg+C“.

Ein Programme kann man auch direkt starten im „LX Terminal“. Dieses Beispiel starten wir mit dem Befehl:

```
python3 Ampel_1.py
```

Auch hier können Sie das Programm ab mit der Tastenkombination „Strg+C“ abbrechen (Taste „Strg“ halten und Taste „C“ drücken). Die Taste „Strg“ (= Steuerung) entspricht der Taste „Ctrl“ (= Control). Eventuelle Fehlermeldungen interessieren uns hier noch nicht.

Starten und Beenden von Programmen:

- Programme startet man unter „Run / Run Module“ oder einfach mit Drücken der F5-Taste
- Programme kann man auch direkt im „LX Terminal“ starten
- Programme werden abgebrochen mit der Tastenkombination „Strg+C“ (bzw. „Ctrl+C“)

Wenn wir im obigen Beispiel anstatt der roten LED die gelbe LED (angeschlossen an GPIO27) verwenden, dann macht unser kleines Programm schon einen professionellen Eindruck: Wie bei einer in der Nacht ausgeschalteten Ampel sieht man nur das gelbe Blinklicht.

Aufgabe:

Anstelle der roten LED soll die gelbe LED der Ampel 1 blinken. Alle nötigen Informationen sind oben bereits zu finden!