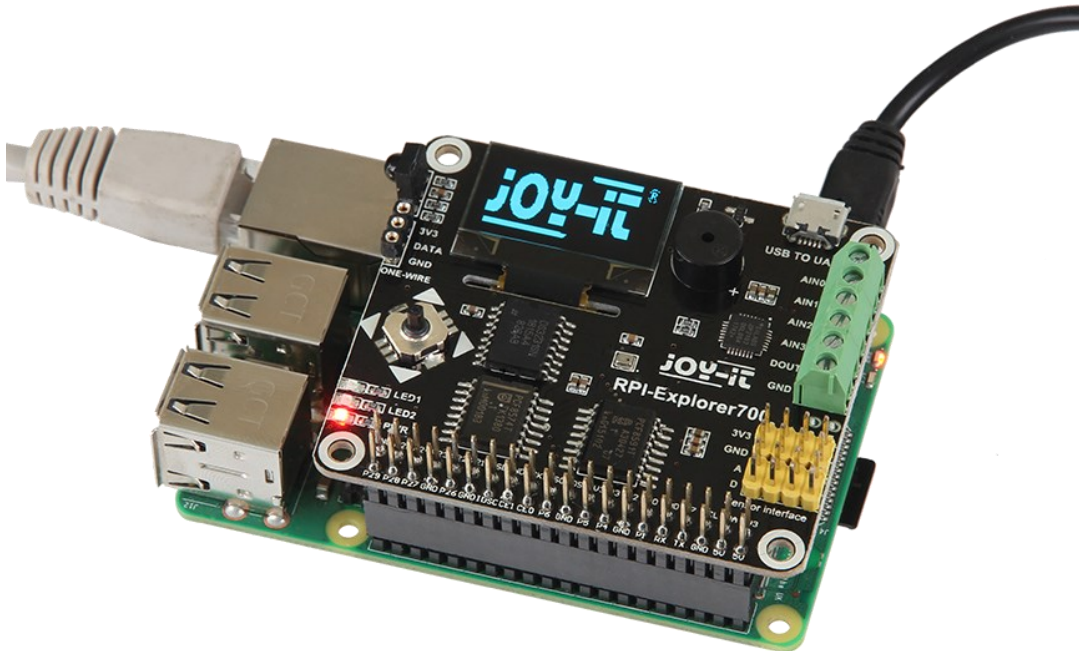


# EXPLORER 700

Erweiterungsplatine

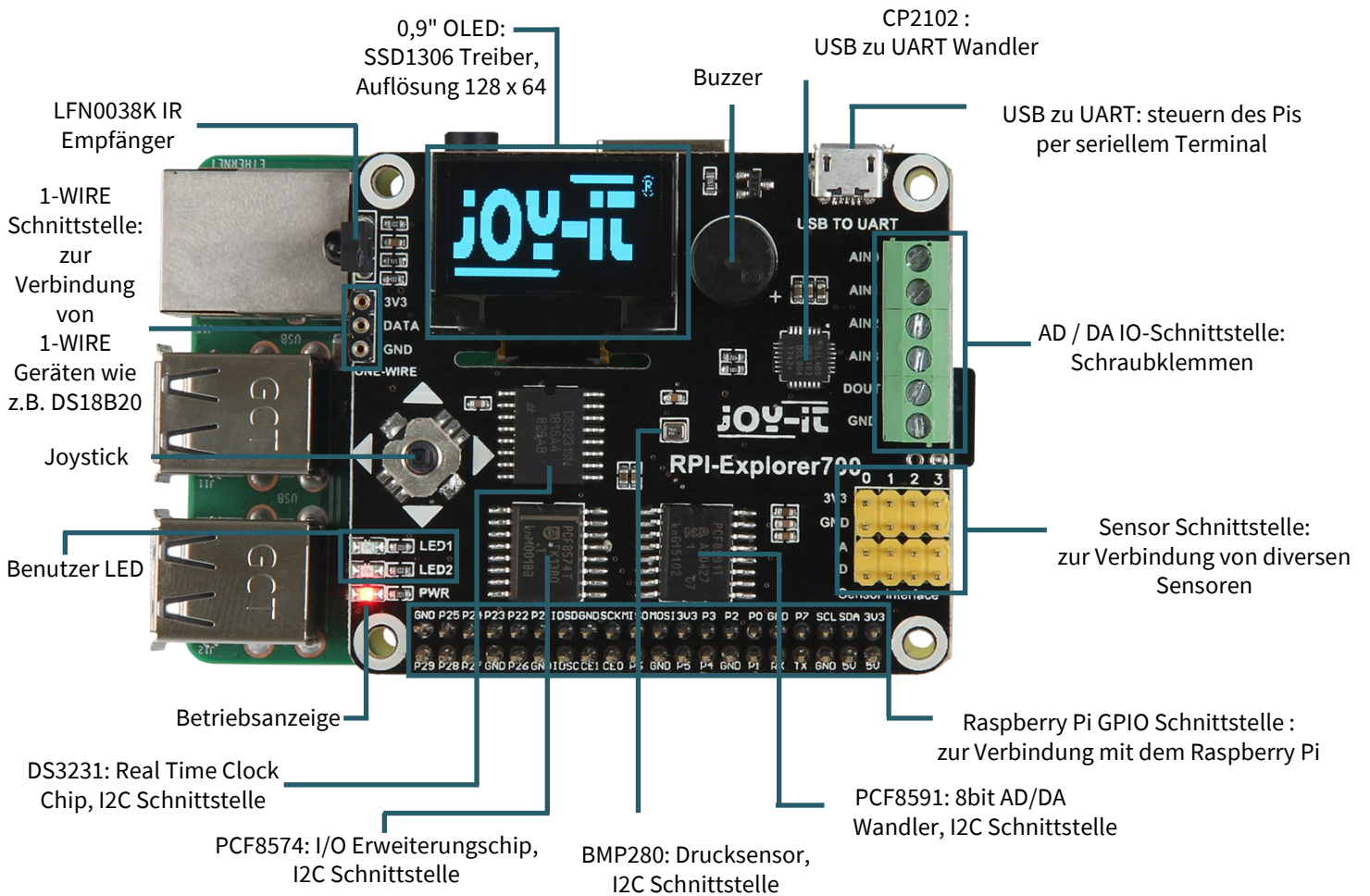


## 1. ALLGEMEINE INFORMATIONEN

Sehr geehrter Kunde,  
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.

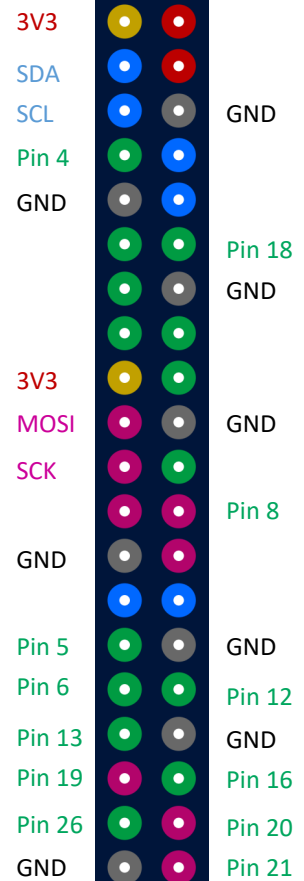
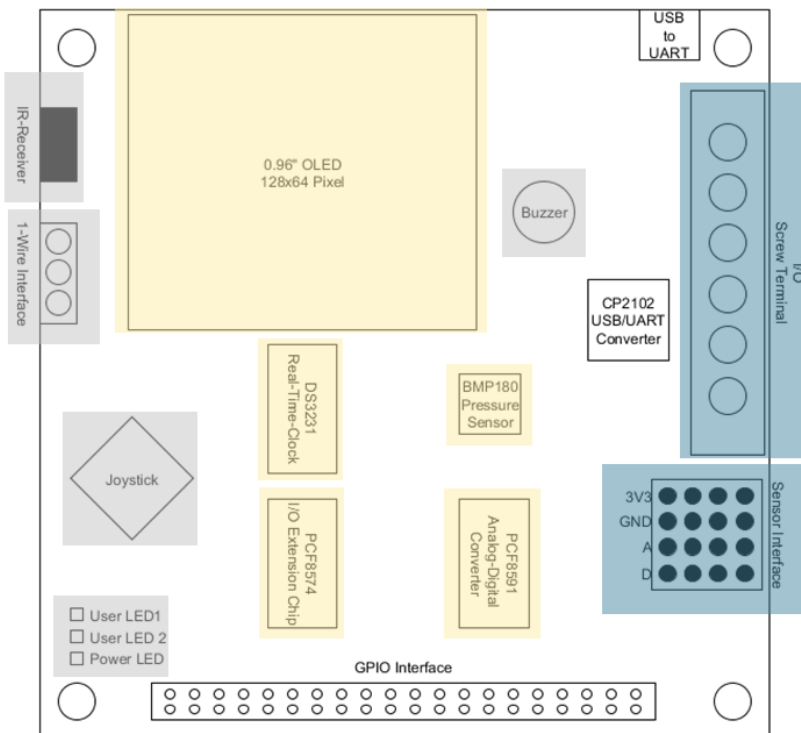
## 2. ANSCHLÜSSE

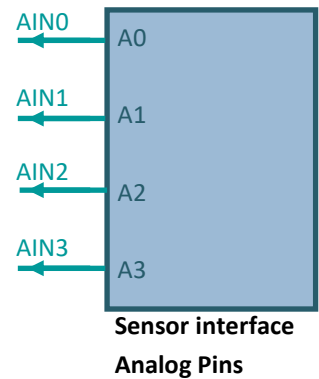
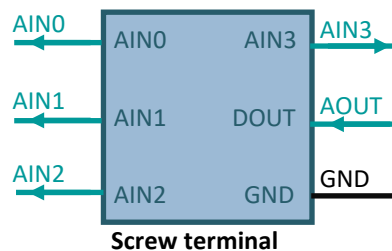
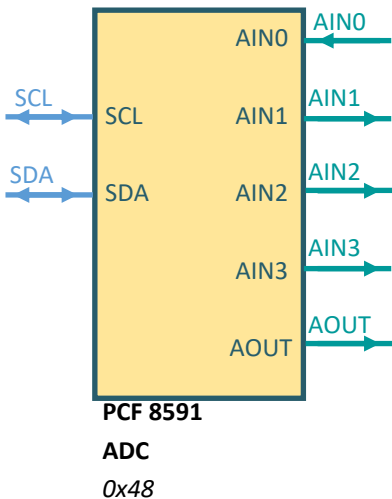
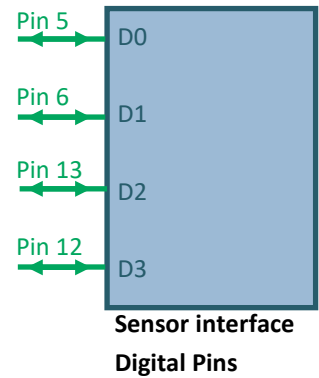
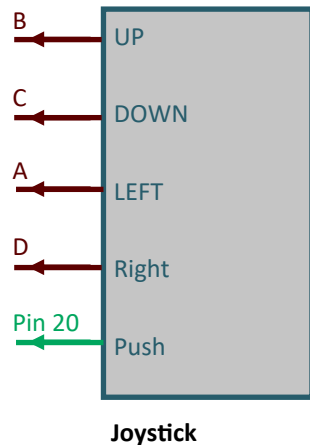
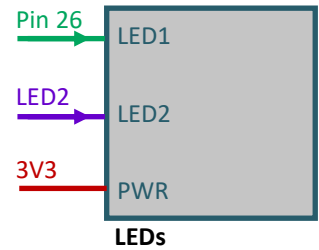
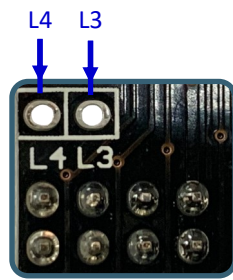
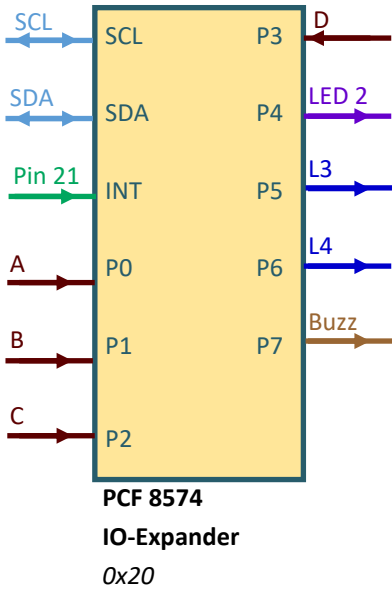
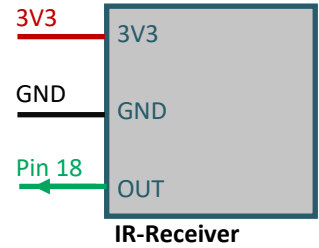
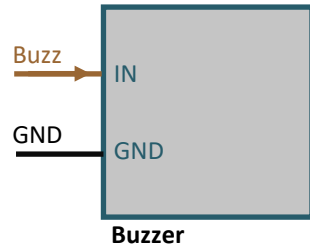
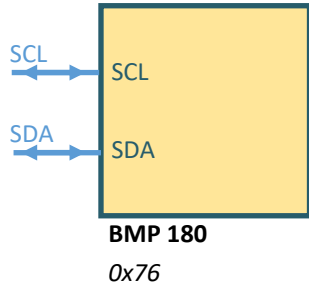
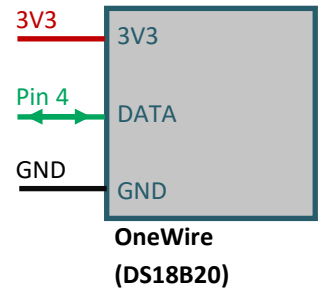
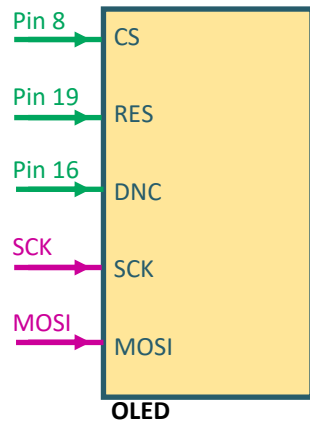
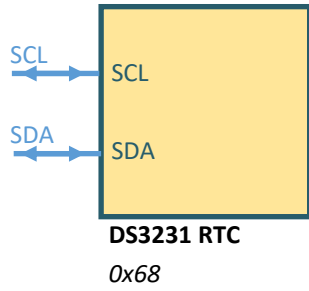


### Schematische Übersicht

Alle hier angegebenen Pins beziehen sich auf die GPIO/BCM-Pins des Raspberry Pi.

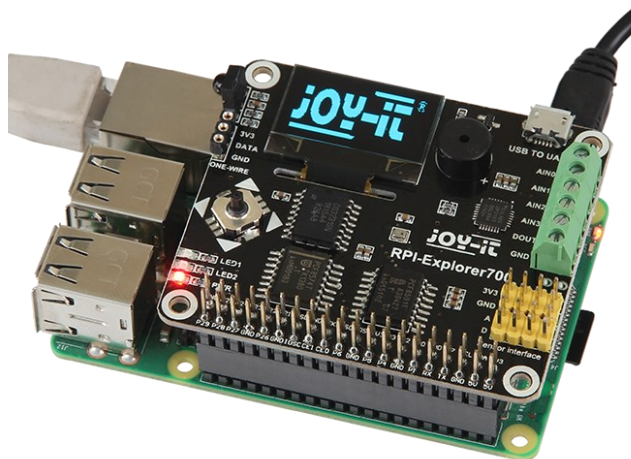
- ICs
- Anschlüsse
- Module





### 3. VERBINDEN DES ERWEITERUNGSBOARDS

Das Explorer 700 Erweiterungsboard unterstützt das Raspberry Pi A+, B+, 2, 3 Model B und 4. Vor dem Ausführen der Beispielprogramme sollten Sie sicherstellen, dass der Explorer 700 mit Ihrem Raspberry Pi verbunden ist und dieser mit Strom versorgt wird.



### 4. VERFÜGBARER DOWNLOAD

Wir stellen Ihnen ein vorgefertigtes Image zur Verfügung, welches Ihnen das installieren und einstellen von Bibliotheken erspart. Sie können sich das Image [hier](#) herunterladen.

Um das Image auf Ihre microSD-Karte zu installieren, können Sie [Win32DiskImager](#) verwenden. Hier wählen Sie die zu beschreibende microSD-Karte aus und das heruntergeladene Image. Nun klicken Sie auf **Write** um diese SD-Karte zu beschreiben.

Wenn Sie unser Image nicht verwenden, können Sie unsere Beispielcodes [hier](#) herunterladen, was Ihnen den Zugriff auf diese Beispiele erleichtert. Beachten Sie jedoch dann die nächsten zwei Kapitel.

Diese Kapitel können bei der Verwendung unseres Images übersprungen werden.

### 5. INSTALLATION DER ZUSATZBIBLIOTHEKEN

Um die Beispiele zu benutzen, die wir anbieten, sind bestimmte Bibliotheken notwendig die manuell installiert werden sollten, falls Sie nicht unser Image verwenden.

Python Bibliotheken für Raspbian (enthält Raspberry Pi, GPIO und spidev Installationspakete. Siehe: <https://pypi.python.org/pypi/RPi.GPIO> <https://pypi.python.org/pypi/spidev> ) erhalten Sie per **wget** Kommando.

Bitte beachten Sie, dass Ihr Raspberry Pi mit dem Netzwerk verbunden ist wenn Sie das Kommando zum installieren der Bibliothek benutzen. Vor der Installation können Sie folgendes Kommando benutzen um die neueste Version ihrer Software zu erhalten.

```
sudo apt-get update
sudo apt-get upgrade
```

Geben Sie nun das folgende Kommando ein, um das Python-dev Paket zu installieren:

```
sudo apt-get install python3-dev
```

Geben Sie nun die folgenden Kommandos ein, zur Installation der Bibliothek smbus (I2C Schnittstellen Funktionen):

```
sudo apt-get install python3-smbus
sudo pip3 install smbus2
```

Geben Sie folgendes Kommando ein, um die serielle Bibliothek zu installieren, welche die UART Schnittstellen Funktionen enthält:

```
sudo apt-get install python3-serial
```

Für unsere Beispielcodes benötigen Sie die [RPi.GPIO](#) und die [spidev](#) Bibliotheken.

Die RPi.GPIO Bibliothek installieren Sie mit dem folgenden Befehl:

```
sudo apt-get install python3-rpi.gpio
```

Die spidev Bibliothek mit diesem Befehl:

```
sudo pip3 install spidev
```

Um das OLED nutzen zu können müssen Sie dazu diese Bibliothek installieren:

```
sudo apt-get install python3-pil
```

Desweiteren benötigen Sie für den BMP280 zwei Bibliotheken von Adafruit [CircuitPython BMP280](#) und [CircuitPython Bundle](#), welche beide unter der MIT Lizenz ([BMP280](#) und [Bundle](#)) veröffentlicht wurden. Sie installieren diese wie folgt:

```
sudo apt-get install git
git clone https://github.com/adafruit/Adafruit_CircuitPython_BMP280.git
sudo pip3 install adafruit-circuitpython-bmp280
git clone https://github.com/adafruit/Adafruit_CircuitPython_Bundle.git
sudo pip3 install adafruit-circuitpython-lis3dh
```

Für die Installation der ADC Bibliothek geben Sie folgende Befehle ein:

```
git clone https://github.com/joy-it/PCF8591-python3-lib.git
cd ~/PCF8591-python3-lib/
sudo python3 setup.py install
```

## 6. KONFIGURATION DER SCHNITTSTELLEN

Vor dem Ausführen der API Codes, die wir zur Verfügung stellen, sollten Sie die entsprechenden Treiber der Schnittstellen aktivieren.

### **Aktivierung der I2C - Funktion**

Geben Sie folgendes Kommando ein, um Ihren Raspberry Pi zu konfigurieren:

```
sudo raspi-config
```

Wählen Sie **Interfacing Options** → **I2C** → **yes**, zum Starten des Treibers. Danach müssen Sie ihr Konfigurationsdatei modifizieren. Geben Sie folgendes Kommando ein um die Konfigurationsdatei zu öffnen:

```
sudo nano /etc/modules
```

Fügen Sie der Konfigurationsdatei folgende beiden Zeilen hinzu:

```
i2c-bcm2708  
i2c-dev
```

Drücken Sie die Tastenkombination **STRG + C** zum Verlassen und geben Sie ein **Y** ein um die Einstellungen zu speichern.

### **Aktivierung der seriellen Funktionen**

Der serielle Port des Raspberry Pis ist standardmäßig auf das Debuggen per Konsolenmodus eingestellt. Wenn Sie den seriellen Port als normalen IO benutzen wollen, müssen Sie die Einstellungen des Raspberry Pis verändern. Wenn das Debuggen per Konsolenfunktion deaktiviert ist, können Sie nicht auf die Raspberry Pi Platine per seriellen Port zugreifen. Wenn Sie den Raspberry Pi steuern wollen, müssen Sie das Debuggen per Konsolenfunktion des seriellen Ports wieder aktivieren.

```
sudo raspi-config
```

Wählen Sie in **Interfacing Options** → **Serial** aus. Durch das Auswählen der Option **No** kann das Debuggen per Konsolenfunktion abgeschaltet werden. Danach kann der serielle Port für die serielle Kommunikation benutzt werden. Mit dem Wählen von Option **Yes** kann das Debuggen per Konsolenfunktion wieder aktiviert werden.

Außerdem müssen Sie das 1-Wire Interface aktivieren. Wählen Sie dazu

**Interfacing Options** → **1-Wire** und aktivieren die Funktion mit **Yes**.

Sie sollten nun den Raspberry Pi neustarten, damit die Einstellungen übernommen werden.

**Notiz:**     **Der serielle Port des Raspberry Pi 3 B ist nicht verfügbar, da Pin 14 und 15 mit dem eigenen Bluetooth verbunden sind.**

Um trotzdem die seriellen Funktionen nutzen zu können, müssen Sie die SPI-Funktion aktivieren. Dabei starten Sie die SPI-Funktion und geben folgenden Befehl ein:

```
sudo raspi-config
```

Nun müssen Sie nur noch in den **Interfacing Options** → **SPI** → **Yes** auswählen, um die seriellen Funktionen nutzen zu können.

## 7. ANALOG DIGITAL CONVERTER BEISPIELE

Das Explorerboard besitzt 4 Analoge Eingänge die durch den ADC PCF8591 in Digitale Signale umgewandelt werden können.

Zum Starten des Beispiel Programms, geben Sie folgende Befehle ein:

```
cd ~/PCF8591-python3-lib/examples/  
sudo python3 read.py
```

Außerdem besitzt das Board einen Analogen Ausgang, über den ein Analoges Signal ausgegeben werden kann.

Zum Starten des Beispiel Programms, geben Sie folgende Befehle ein:

```
cd ~/PCF8591-python3-lib/examples/  
sudo python3 write.py
```

## 8. ÄNDERN DES LED1 STATUS

Das Explorerboard besitzt mehrere LEDs, welche mit unseren Beispielprogrammen angespielt werden können. LED1 kann dabei direkt verwendet werden.

Öffnen Sie dafür die Linux Konsole und geben Sie folgende Kommandos ein:

```
cd ~/Desktop/Explorer-700/LED1/  
sudo python3 led.py
```

Dieser Beispielcode bringt LED1 nun zum blinken.

Drücken Sie die Tasten **STRG + C** um das Programm zu beenden.

Für LED1 gib es eine weiter Code, welcher die LED zum pulsieren bringt.

```
sudo python3 pwm.py
```

Dies geschieht durch das schrittweise verändern von der Helligkeit.

Drücken Sie die Tasten **STRG + C** um das Programm zu beenden.

## 9. DRUCKFUNKTION DES JOYSTICKS

Der Joystick besitzt eine Druckfunktion neben den üblichen Bewegungsrichtungen. Unser Beispielcode zeigt Ihnen an, wenn diese Druckfunktion genutzt wird. Öffnen Sie dafür die Linux Konsole und geben Sie folgende Kommandos ein:

```
cd ~/Desktop/Explorer-700/  
sudo python3 press-joystick.py
```

Beim Drücken des Joysticks wird die Konsole folgendes anzeigen:

```
JOYSTICK PRESSED  
JOYSTICK PRESSED  
JOYSTICK PRESSED
```

## 10. I/O-EXPANDER - PCF8574

Die I/O Expander PCF8574 steuert zudem die LED2, den Buzzer und den Joystick. Diese werden dadurch über das PCF8574 angesteuert und nicht direkt.

Öffnen Sie die Konsole und geben Sie die folgenden Kommandos ein:

```
cd ~/Desktop/Explorer-700/  
sudo python3 led2.py
```

Mit diesem Programm blinkt nun die LED2. Drücken Sie die Tasten **STRG + C** um das Programm zu beenden.

```
sudo python3 joystick.py
```

Dieses Programm vereint die Funktionen des Explorerboards, welche über das PCF8574 gesteuert werden. Denn wenn der Joystick in eine Richtung gedrückt wird, ertönt der Buzzer, LED2 leuchtet auf und die Konsole gibt die gedrückte Richtung aus.

```
up  
left  
down  
right
```

Drücken Sie die Tasten **STRG + C** um das Programm zu beenden.

## 11. BAROMETER - BMP280

Dieser Beispielcode basiert auf zwei Bibliotheken von Adafruit (**Kapitel 5**). Dieser Sensor kann die Temperatur und die Luftdruck messen und kann dadurch auch die Höhe ausgeben.

**Beachten Sie, dass die Temperatur durch die Abhitze des Raspberry Pis beeinflusst werden kann.**

**Um eine korrekte Höhe ausrechnen zu können, muss im Skript die Meeresluftdruck auf den Luftdruck des momentanen Standorts geändert werden.**

```
cd ~/Desktop/Explorer-700/BMP280/  
sudo python3 bmp280.py
```

Drücken Sie die Tasten **STRG + C** um das Programm zu beenden.



## 12. REAL - TIME CLOCK - DS3231

Sie können die Verbindung mit der RTC mit dem folgenden Befehl überprüfen:

```
sudo i2cdetect -y 1
```

Nun müsste Ihnen eine Adresse angezeigt werden mit welcher Sie die RTC ansteuern können. **Bei uns ist es 0x68. Wenn diese bei Ihnen abweicht, müssen Sie diese im Codebeispiel ändern.**

Sie können diese RTC zum einen in Ihr System als Real Time Clock einbetten oder nur in einem Code verwenden. Dabei ist zu beachten das eine Real Time Clock nur vollständig funktionsfähig ist, wenn eine Batterie in der entsprechenden Batteriehalterung angebracht wurde.

### Im System einbetten

Zunächst bearbeiten Sie die *modules* Datei. Dies machen Sie mit dem folgenden Befehl:

```
sudo nano /etc/modules
```

Fügen Sie dort diese Zeile am Ende der Datei an:

```
rtc-ds3231
```

Sie können mit **STRG + O** die Datei speichern und mit **STRG + X** den Editor verlassen. Nun müssen Sie die *config.txt* bearbeiten. Diese öffnen Sie mit dem folgenden Befehl:

```
sudo nano /boot/config.txt
```

Dort fügen Sie diese Zeile am Ende der Datei an:

```
dt-overlay=i2c-rtc,ds3231
```

Sie können mit **STRG + O** wieder die Datei speichern und mit **STRG + X** den Editor verlassen. Installieren Sie nun den Treiber für die RTC um Sie in das System einzubetten. Der Treiber von DS1307 (eine andere RTC) wird von der DS3231 mit verwendet.

```
sudo modprobe rtc-ds1307
```

Bearbeiten Sie nun die folgende Datei damit die Systemuhrzeit mit der Uhrzeit der RTC überschrieben wird.

```
sudo nano /etc/rc.local
```

Über dem Befehl **exit 0** fügen Sie dort die folgenden drei Zeilen hinzu:

```
echo ds3231 0x68 > /sys/class/i2c-adapter/i2c-1/new_device  
hwclock --hctosys  
date  
exit 0
```

Schließen Sie die Datei wieder mit **STRG + O** und **STRG + X** und starten Sie den Raspberry Pi neu. Nun wird die Uhrzeit des Systems nach jedem Neustart durch die RTC neu gesetzt.

**Beachten Sie, dass eine CR1220 Batterie im Batteriehalter eingelegt sein muss, damit die Real Time Clock die richtige Uhrzeit beibehalten**

Jetzt können Sie die RTC als diese auch im System ansteuern. Deswegen können Sie nun die folgenden Befehle verwenden:

```
sudo hwclock -r
```

Mit diesem Befehl können Sie die Uhrzeit der RTC ausgeben lassen.

```
sudo hwclock -w
```

Mit diesem Befehl wird die RTC auf die Uhrzeit des Systems angepasst

### **Im Code verwenden**

Wenn Sie Ihre RTC im Code verwenden wollen, müssen Sie die Treiber der RTC deinstallieren oder die RTC nicht im System einbetten, denn der Treiber der RTC blockt diese für weitere Prozesse. Den Treiber deinstallieren Sie mit:

```
sudo rmdir rtc-ds1307
```

Diesen Befehl müssen Sie nur ausführen, wenn Sie die RTC ins System eingepflegt haben. Sie sollten dann auch die hinzugefügten Zeilen in *rc.local* entfernen um falsche Änderungen im System zu vermeiden. Wir verwenden in unserem Codebeispiel die Bibliothek [RTC SDL DS3231](#), welche von [SwitchDoc Labs](#) veröffentlicht wurde. Um den Code auszuführen müssen Sie zunächst in einen Unterordner wechseln. Dies machen Sie mit dem folgenden Befehl:

```
cd ~/Desktop/Explorer-700/RTC_SDL_DS3231
```

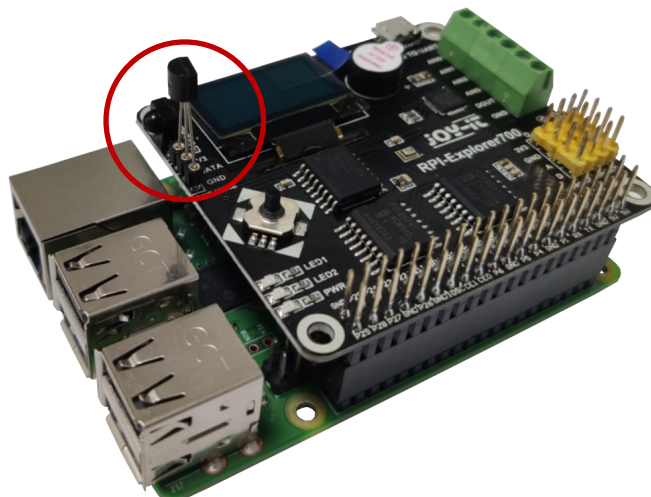
Dort können Sie unseren Code ausführen, welcher die RTC auf die Systemuhrzeit setzt und jede Sekunde die Uhrzeit und das Datum ausgibt. Diesen Code können Sie mit dem folgenden Befehl ausführen:

```
sudo python3 ds3231.py
```

Drücken Sie die Tasten **STRG + C** um das Programm zu beenden.

## **13. TEMPERATURSENSOR - DS18B20**

Der Temperatursensor, welcher im Lieferumfang enthalten ist, wird separat angeschlossen und ist nicht auf der Platine verbaut. Hierbei ist zu beachten, dass die gerade Seite des Sensors zum Display zeigt und die runde Seite Richtung USB- und Ethernet-Ports des Raspberry Pi.



Für das Beispiel öffnen Sie die Konsole und geben Sie die folgenden Kommandos ein:

```
cd ~/Desktop/Explorer-700
sudo python3 ds18b20.py
```

Die Konsole wird daraufhin die Temperatur in Celsius anzeigen.

```
24.687 *C
24.750 *C
24.812 *C
```

Drücken Sie die Tasten **STRG + C** um das Programm zu beenden.

## 14. IRM - INFRAROT REFLEKTIERENDER SENSOR

Auf der Platine befindet sich ein Infrarot-Empfänger, welcher Infrarotlicht in elektrische Signale umwandelt. Diese Signale enthalten Infrarotwerte, welche mittels diesem Quellcode-Beispiel durch die Konsole ausgegeben werden.

```
cd ~/Desktop/Explorer-700
sudo python3 irm.py
```

Hierzu ist eine Infrarot-Fernbedienung von Nöten, welche auf den Infrarot-Empfänger zielt, diese ist jedoch nicht im Lieferumfang enthalten.

```
IRM test start:
Get the key: 0x0c
Get the key: 0x18cd
Get the key: 0x5e
```

Drücken Sie die Tasten **STRG + C** um das Programm zu beenden.

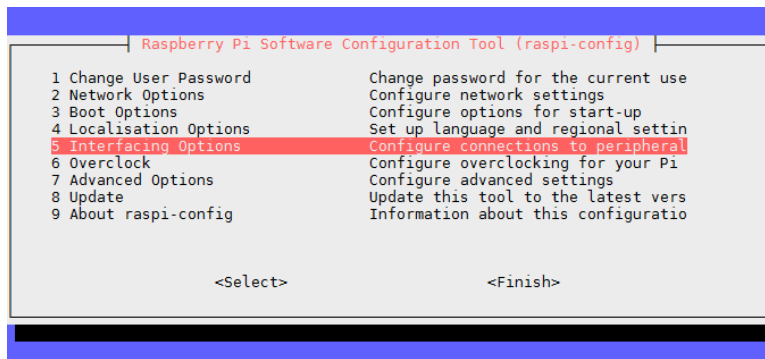
## 15. UART - ÜBERTRAGUNG SERIELLER DATEN

Die serielle Schnittstelle des Raspberry Pi ist standardmäßig deaktiviert. Für dieses Programm, muss jedoch diese Funktion aktiviert sein damit der Raspberry Pi nicht über den seriellen Port kommuniziert. Um den Beispielcode verwenden zu können, muss zunächst die serielle Funktion in der Konfiguration des Raspberry Pis geändert werden.

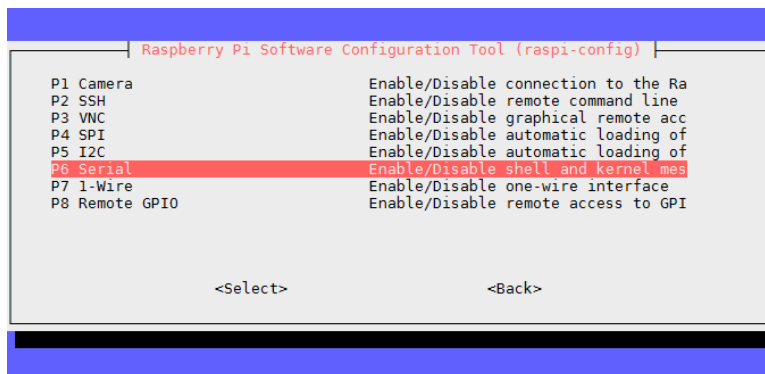
Geben Sie dafür zunächst diesen Befehl in die Konsole ein:

```
sudo raspi-config
```

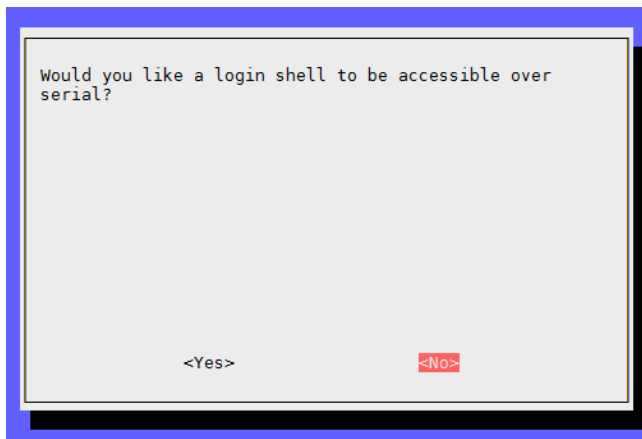
Navigieren Sie nun, mit den Pfeiltasten Ihrer Tastatur zu dem Menüpunkt **5 Interfacing Options**.



Danach wählen Sie den **Punkt P6 Serial** aus.



Die Frage **Would you like a login shell to be accessible over serial?** beantworten Sie mit **No**.

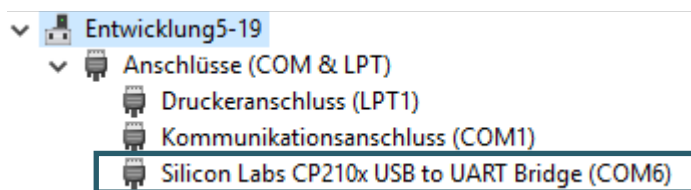
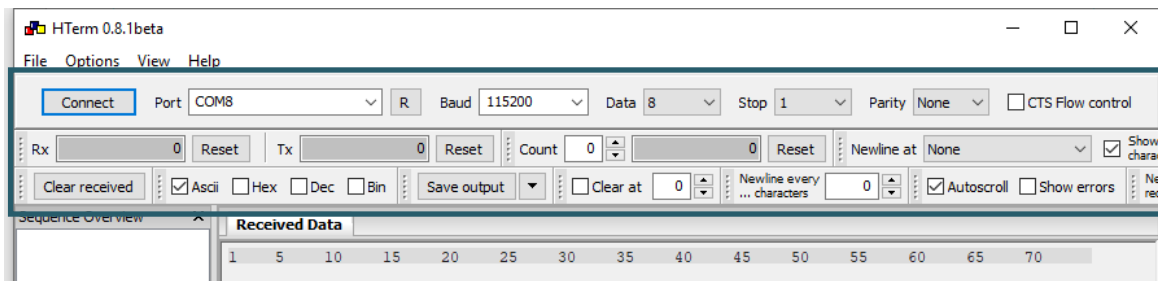


Die Frage **Would you like the serial port hardware to be enabled?** beantworten Sie mit **Yes**.



Die USB to UART Schnittstelle Ihres Raspberry Pi ermöglicht Ihren Pi mit einem Computer zu verbinden und sie miteinander kommunizieren zu lassen, ohne eine Verbindung zum Internet.  
In diesem Beispiel nutzen wir HTerm. Laden Sie sich das Programm [hier](#) herunter und installieren Sie es.

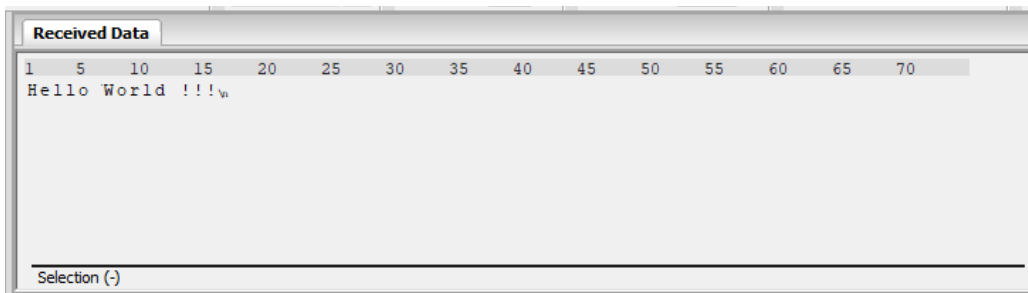
Öffnen Sie zunächst HTerm und übernehmen Sie die Einstellungen, wie Sie in der unteren Abbildung zu sehen ist. Dabei ist zu beachten, dass Sie die richtige Baudrate (nämlich 115200) und den Port richtig einstellen. Jedoch kann der Port bei Ihnen abweichen, was Sie im Geräte-Manager überprüfen sollten.



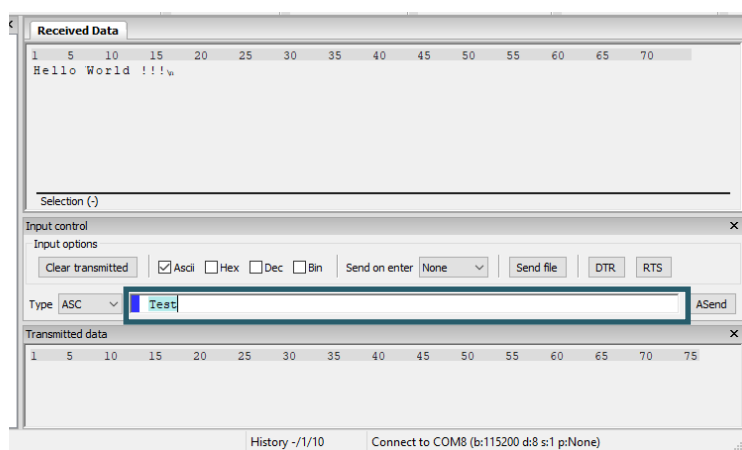
Nun können Sie mit einem Klick auf Connect die Verbindung herstellen. Als nächstes müssen Sie das Beispielprogramm auf Ihrem Raspberry Pi starten. Dazu geben Sie in der Konsole das folgende ein:

```
cd ~/Desktop/Explorer-700  
sudo python3 uart.py
```

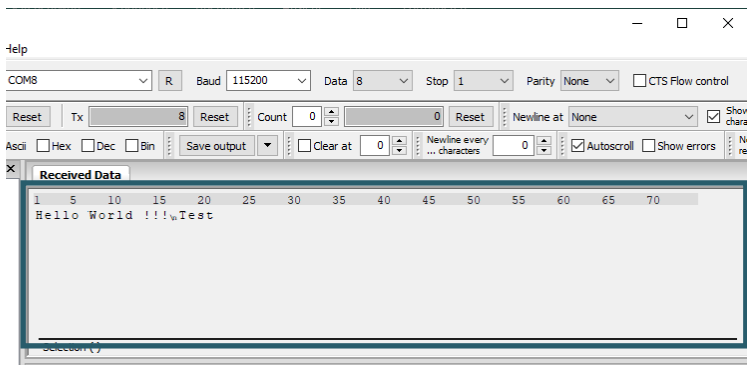
In HTerm wird Ihnen daraufhin das folgende angezeigt:



Nun können Sie auch Texte zu Ihrem Raspberry Pi senden. Das Beispielprogramm sorgt dafür, dass der geschickte Text vom Raspberry Pi wieder zurück zum Computer geleitet wird. Sie können diese Texte im Textfeld eingeben und mit **ASend** abschicken.



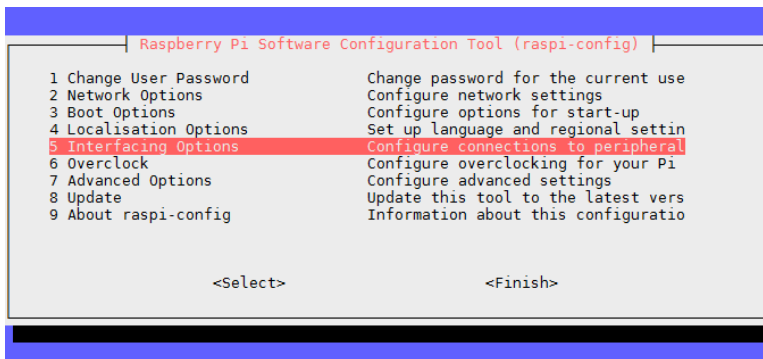
Daraufhin wird Ihr Computer das folgende erhalten:



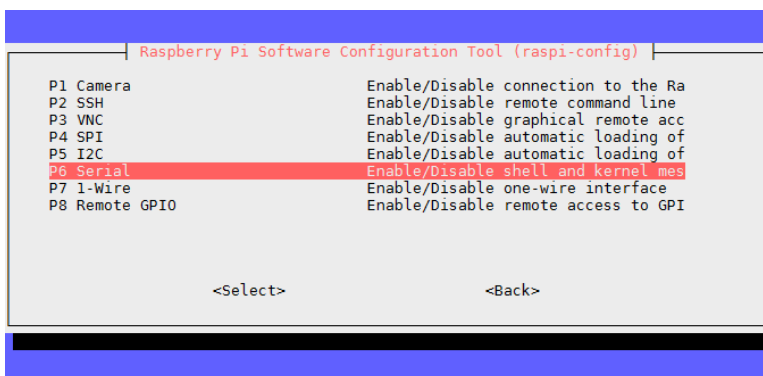
Sie können auch die Terminalkonsole über die UART-Verbindung öffnen, anstatt via eines externen Monitors oder über das Internet. Hierfür können Sie PuTTY, MobaXterm oder andere verbinden. Sie müssen hierfür, jedoch andere Einstellungen vornehmen als zuvor. Geben Sie dafür zunächst diesen Befehl in die Konsole ein:

```
sudo raspi-config
```

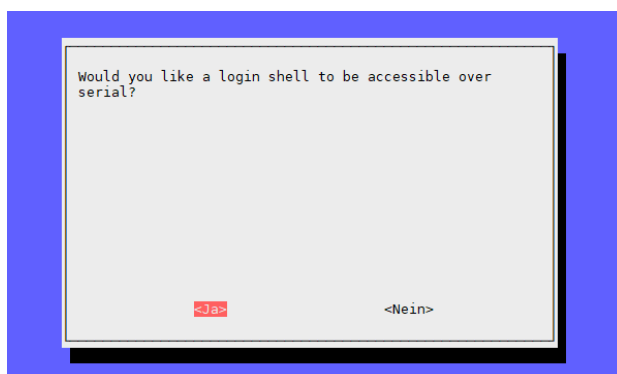
Navigieren Sie nun, mit den Pfeiltasten Ihrer Tastatur in den Menüpunkt **5 Interfacing Options**.



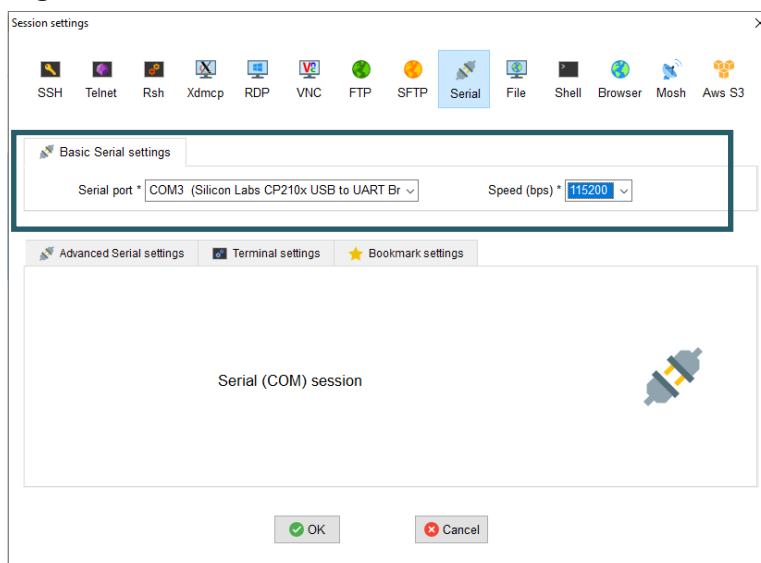
Danach wählen Sie den **Punkt P6 Serial** aus.



Die Frage **Would you like a login shell to be accessible over serial?** beantworten Sie mit **Yes**.



Nach einem Neustart können Sie nun Ihren Raspberry Pi an Ihrem PC anschließen. Nun öffnen Sie MobaXterm, welches Sie sich [hier](#) herunterladen können und stellen eine Verbindung mit den folgenden Einstellungen her:



Hierbei ist zu beachten, dass Sie als Verbindungsart **Serial** und den richtigen Port (sichtbar im Geräte-Manager) auswählen als auch die **Baudrate** auf **115200** setzen.

Nun wird sich die Konsole öffnen und Sie müssen sich auf Ihrem Raspberry Pi einloggen. In unserem Image ist der Benutzername **pi** und das Passwort **raspberrypi**.

**Achtung**, dabei werden nicht die Tastenschläge bei der Passwordeingabe angezeigt.

```
Raspbian GNU/Linux 10 raspberrypi ttyS0
raspberrypi login: pi
Passwort: █
```

Nachdem Sie das Login erfolgreich gemeistert haben, öffnet sich nun die Konsole und Sie können das Terminal des Raspberry Pi genauso nutzen, wie mit SSH nur ohne eine Internetverbindung zu benötigen.

```
Raspbian GNU/Linux 10 raspberrypi ttyS0
raspberrypi login: pi
Passwort:
Letzte Anmeldung: Donnerstag, den 24. Oktober 2019, 15:56:29 CEST auf tty1
Linux raspberrypi 4.19.66-v7+ #1253 SMP Thu Aug 15 11:49:46 BST 2019 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.

pi@raspberrypi:~$ █
```

## 16. OLED - DISPLAY

Um diese Codes ausführen zu können, stellen Sie sicher, dass **I2C** und **SPI** aktiviert sind. In **Kapitel 6** sehen Sie, wie Sie diese aktivieren. Es muss auch die PIL-Bibliothek installiert sein. Wie sie diese installieren sehen Sie in **Kapitel 5**.

Unsere Beispielcodes können verschiedene Ausgaben auf diesem OLED erzeugen. Diese unterscheiden sich in wesentlichen Aspekten untereinander.

```
cd ~/Desktop/Explorer-700/OLED
sudo python3 dispchar.py
```

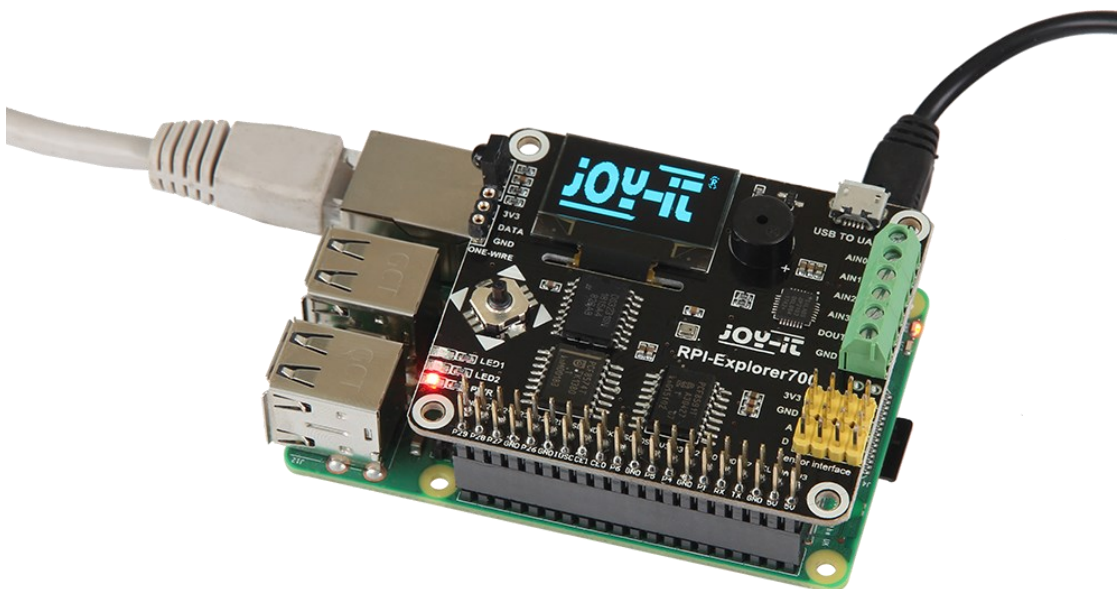
Dieses Skript gibt einen Text auf dem Display aus, welcher auch im Quellcode verändert werden kann.

```
sudo python3 image.py
```

Dieses Skript stellt ein Bild auf dem OLED dar. Dieses kann auch personalisiert werden.

```
sudo python3 animate.py
```

Das letzte Skript erzeugt eine Animation auf Ihrem OLED.





## 17. 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 Alttakkumulatoren, 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 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.

## 18. 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 98469 – 66 (10 - 17 Uhr)

Für weitere Informationen besuchen Sie unsere Website:

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