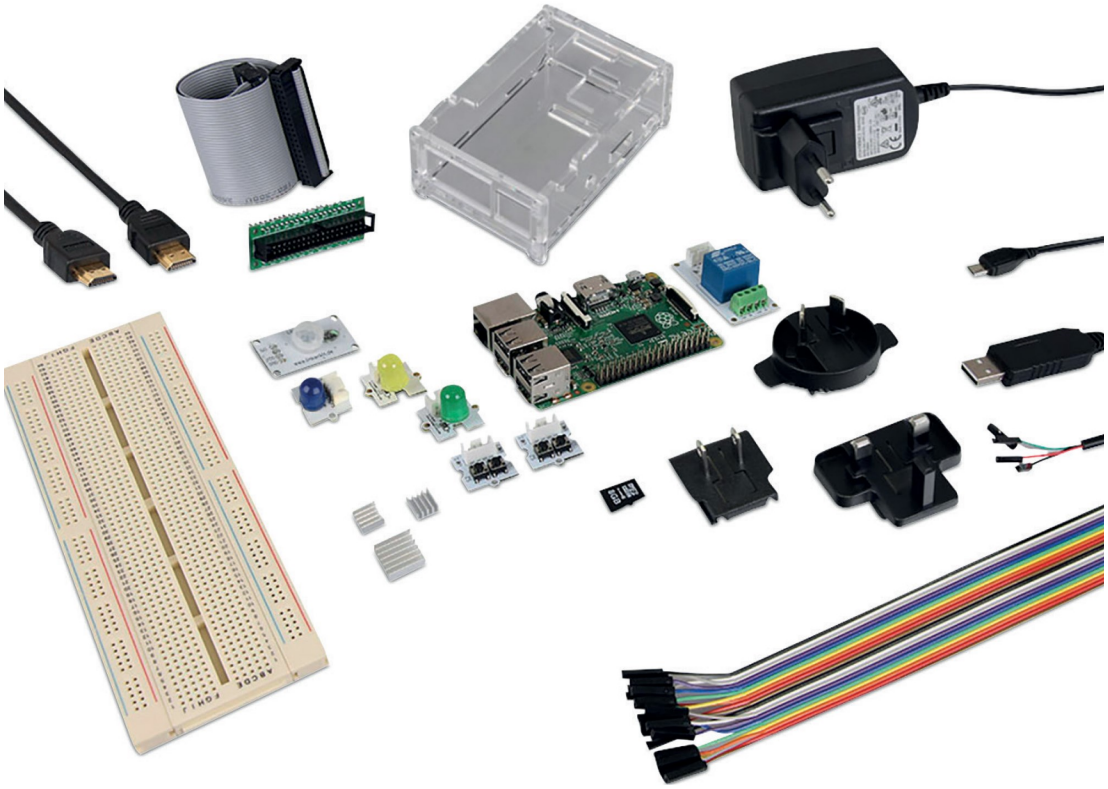
renkforce

RASPBERRY PI 3

Experiment-Set



Sehr geehrter Kunde,

Wir freuen uns, dass Sie mit Ihren Kauf uns Ihr Vertrauen und die Möglichkeit gegeben haben die Vielseitigkeit und Qualität eines renkforce Produktes zu beweisen.
Wir wünschen Ihnen viel Freude mit dem Raspberry Pi 3 Experiment-Set

---

Das Experiment-Set bildet die perfekte Grundlage für Ihren Raspberry. Starten Sie mit diesem umfangreichen Zubehör-Paket Ihre Experimente und Projekte. Lassen Sie Ihrer Kreativität freien Lauf und führen Sie mit den Steckbrückenkabeln, Widerständen, LEDS und Schalter verschiedenste Versuche durch. Eine Verbindung zwischen Raspberry Pi® 3 und der Steckplatine lässt sich mit dem Cobbler-Verbindungskit realisieren. Mit dem USB TTL-Kabel können Sie Ihren Einplatinencomputer sogar mit einem PC verbinden.

Das Gehäuse bietet der Platine den nötigen Schutz und ermöglicht Zugriff auf die Schnittstellen. Die Stromversorgung übernimmt ein Mico-USB Nezteil mit 2000 mA Ausgangsstrom und einer Wechselclip-Funktion für den internationalen Einsatz. Mit dem vorinstallierten Noobs Betriebssystem auf der MicroSD-Karte können Sie sofort losstarten. Eine Verbindung mit einem Monitor oder sogar dem Fernseher kann über das beiliegende HDMI™-Kabel hergestellt werden.

Die absolute Vielseitigkeit zeichnen das Raspberry Pi® Set desweiteren aus. Setzen Sie es als Mini-PC in Ihrem Wohnzimmer ein und geben Sie Videos in Full HD über den HDMI™-Ausgang wieder oder verbinden Sie ihn mit weiteren Platinen über den GPI0-Port um Steuerungen zu realisieren.

In der folgenden Anleitung haben wir einzelne Beispiele aufgeführt, um Sie in den ersten Schritten mit dem Raspberry Pi 3 Experiment Set zu begleiten und einen reibungslosen Einstieg zu ermöglichen.

Lieferumfang

- Raspberry Pi® 3 Model B 1 GB
- Gehäuse
- Raspberry Pi Micro-USB-Netzteil 2.5 A mit Wechselclips für Europa, Großbritannien, Australien, USA
- Noobs Betriebssystem auf 8 GB MicroSD-Karte
- Kühlkörper-Set
- HDMI™-Kabel
- Cobbler-Verbindungskit
- USB TTL-Kabel
- Steckplatine
- Steckbrücken-Verbindungskabel
- 4x 10 mm LEDs
- 2x Schaltermodule
- Relais
- IR-Bewegungsmelder

Vorbereitung des Raspberry Pi

Um mit den Raspberry Pi die folgenden Beispiele nutzen zu können, muss dieser vorab vorbereitet werden. Im Lieferumfang finden Sie neben dem Raspberry Pi 3 auch ein passendes Gehäuse aus Acryl-Glas, welche eine eigene Anleitung für den Zusammenbau beinhaltet.

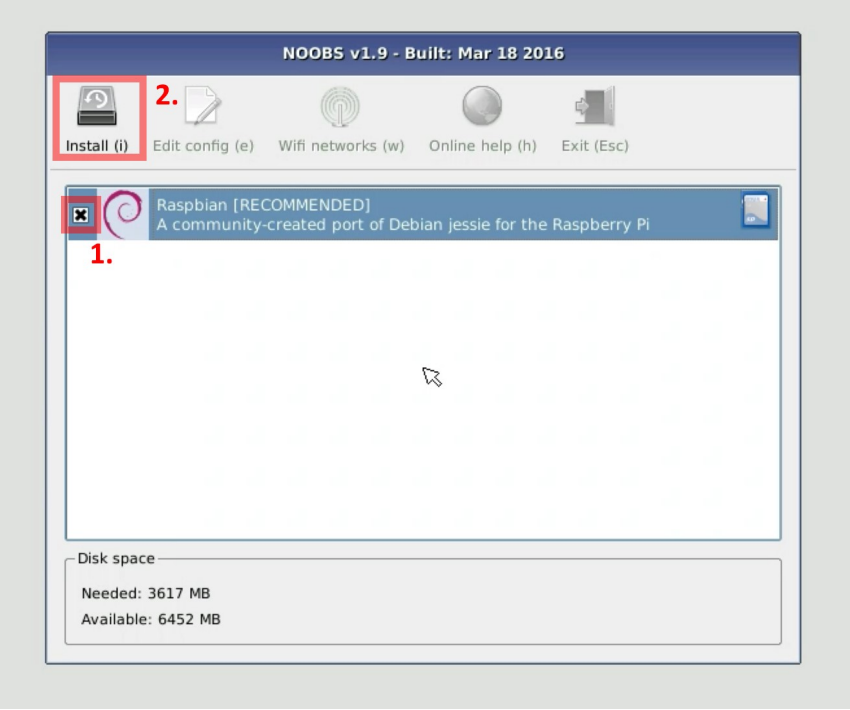
Nachdem Sie das Gehäuse mit eingesetzten Raspberry Pi zusammengebaut haben, setzten Sie die beiliegende microSD-Karte in den entsprechenden Slot an der Unterseite der Platine ein.
Verbinden Sie daraufhin per HDMI-Kabel den Raspberry Pi an einen HDMI-fähigen Bildschirm/Fernseher und schließen Sie an den USB-Buchsen eine handelsübliche Computermaus und Tastatur an (Alternativ: drahtlos Maus-/Tastaturempfänger)
Das in diesem Set enthaltende microUSB Netzteil, hat die Möglichkeit mit verschiedenen Länderadaptern ausgestattet und somit rund um den Globus eingesetzt zu werden. Setzten Sie hier den zu Ihrer Wandsteckdose passenden Adapter auf das Netzteil auf und verbinden Sie das Netzteil auf der einen Seite an die microUSB-Buchse des Raspberry Pi und auf der anderen Seite mit der Wandsteckdose.

Der Raspberry Pi 3 sollte nun starten und ein Bild auf dem HDMI-Bildschirm ausgeben.

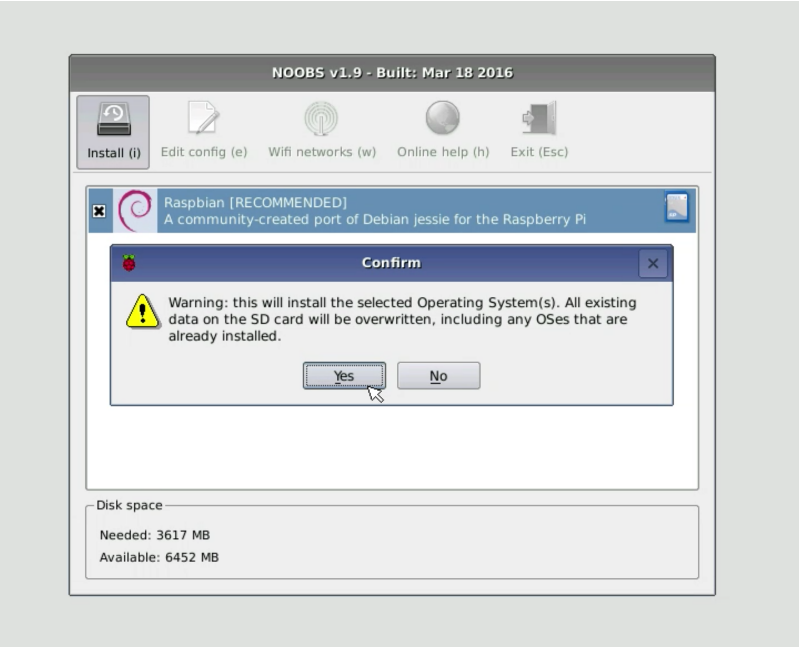
Vorbereitung des Betriebsystemes

Die beiliegende micro-SD Karte beinhaltet das Betriebssystem-Installationsprogramm „NOOBS“, welches erlaubt mehrere verschiedene Betriebssysteme auf dem Raspberry Pi zu nutzen—unter anderem auch als Office-PC, als Programmierumgebung oder als Multimediacenter.
Sollten Sie den Raspberry Pi zusätzlich mit einem Netzwerkkabel an das Internet angeschlossen haben, sollten Sie eine aktuelle Auswahl an den installierbaren Betriebsystemen angezeigt bekommen.

Darunter befindet sich auch das Betriebsystem „Raspbian“, welches eine für den Raspberry Pi angepasste Version der Linux-Distribution Debian ist (diese wird Ihnen auch ohne Online-Zugang angezeigt, da dieses auf dieser SD-Karte schon vorbereitet ist).
Für die folgenden Beispiele empfehlen wir auf Raspbian zu setzen und zeigen im folgenden auf, wie dieses zu installieren ist:

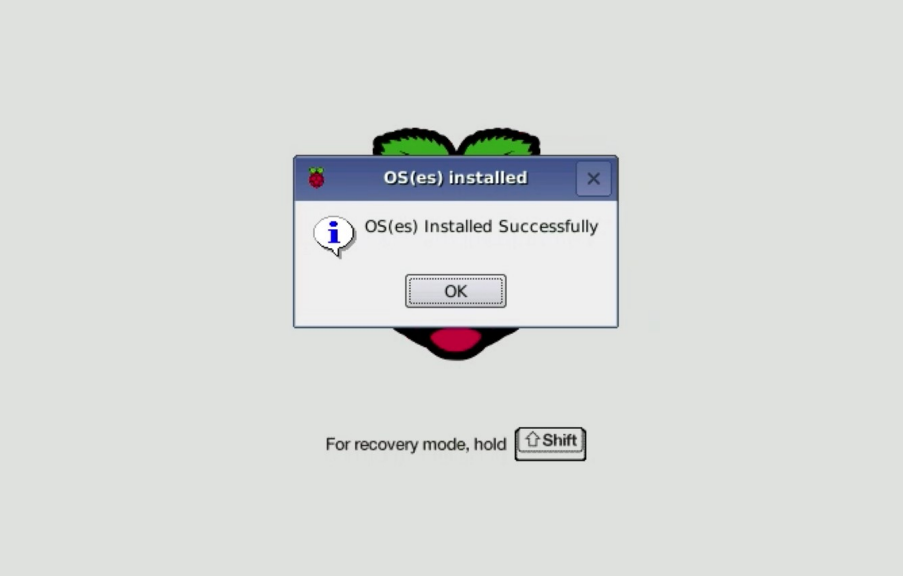


Wählen Sie zu allererst das Betriebssystem aus, indem Sie das Kreuz in das vorgesehene Kästchen mit einem Linksklick setzen—danach starten Sie die Installation mit einem Drücken auf den Punkt „Install (i)“



Die darauffolgende Meldung bestätigen Sie mit einen Drücken auf „Yes“

Danach wird die Installation gestartet; diese dauert in der Regel 10-15 Minuten.

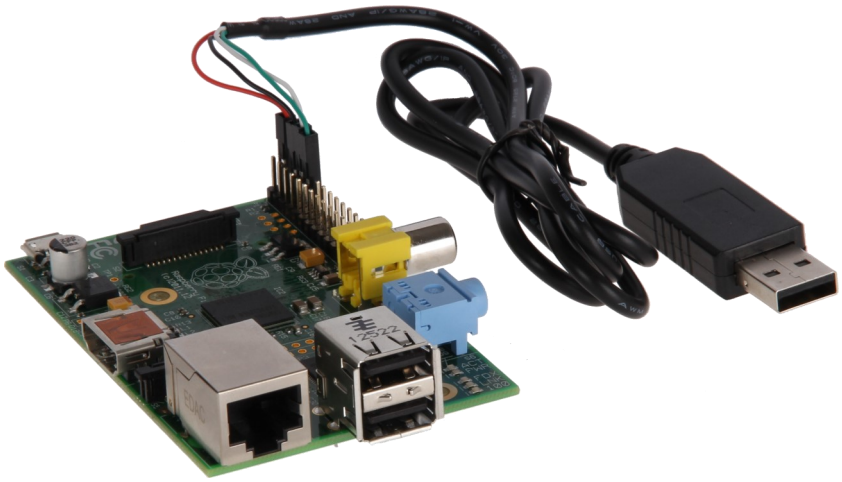


Nach der Installation können wir diese mit einem Drücken auf OK beenden.

Danach bootet der Raspberry Pi in das Betriebssystem Raspbian.

Kommunikation mit dem PC über die serielle Schnittstelle

In diesem Set enthalten finden Sie ein USB-TTL Adapterkabel. Dieses ermöglicht Ihnen mit diversen Mikrokontroller– und Einplatinencomputersystemen zu kommunizieren—sei es die Überprüfung von Messwerten, Eingabe von Parametern oder die Steuerung des Betriebssystems über eine Terminalkonsole. Letztere Möglichkeit ist im Betriebsystem Raspbian des Raspberry Pi enthalten.

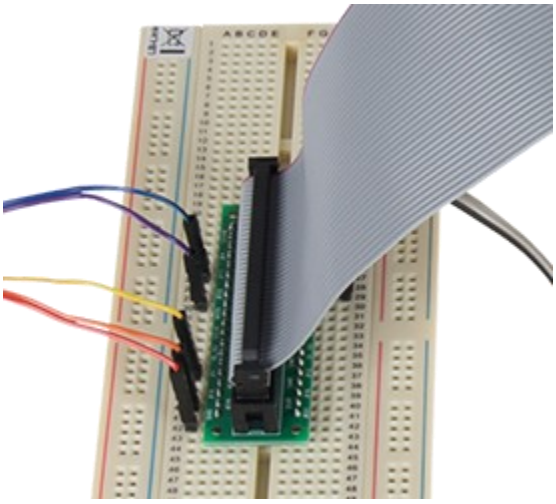


Über die serielle Schnittstelle auf den Pins 8 [TXD] und 10 [RXD] auf dem GPIO-Header des Raspberry Pi, lässt sich somit eine Verbindung zur Terminalkonsole aufbauen und Befehle für das Betriebsystem eingeben, ohne dass eine Verbindung per Netzwerk aufgebaut oder ein Bildschirm per HDMI angeschlossen werden muss.

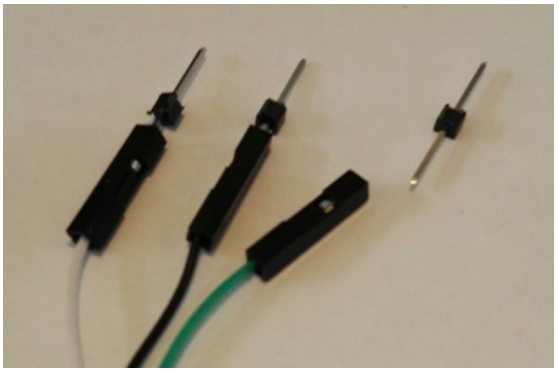
Hierzu gehen Sie wie folgt vor:

Schließen Sie das große Datenkabel auf die GPIO-Leiste auf, wie es auf dem Bild auf der rechten Seite gezeigt wird. Achten Sie hierbei darauf, dass die rot markierte Seite des Kabels nach außen zeigt.

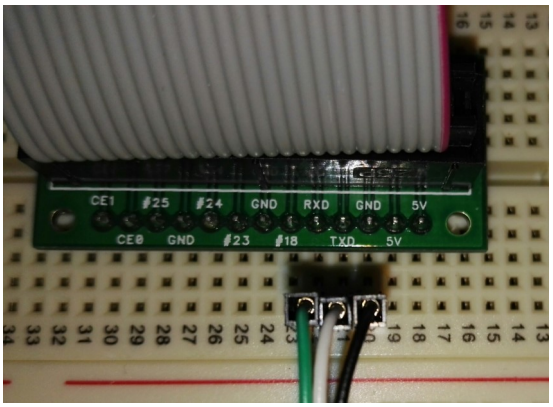
Hiernach setzen Sie die Cobbler Verbindungplatine auf das Steckbrett auf und verbinden das Datenkabel, welches vom Raspberry Pi kommt.



Setzen Sie nun in die Verbindungsstücke des USB-TTL Verbindungskabels die Pin-Stiftadapter ein; diese Sind im beiliegenden Kabel-Set enthalten.



Nun haben Sie die Möglichkeit die Verbindungsstücke des USB-TTL Kabels in das Steckbrett zu stecken.
Auf der Cobbler Verbindungsplatine, sehen Sie die entsprechenden Pinbezeichnungen des Raspberry Pi‘s aufgedruckt.



Die Verbindung muss nun wie folgt aufgebaut werden:

USB-TTL [TX] [Grün] - > Raspberry Pi [RX]
USB-TTL [RX] [Weiß] - > Raspberry Pi [TX]
USB-TTL [GND] [Schwarz] - > Raspberry Pi [GND]

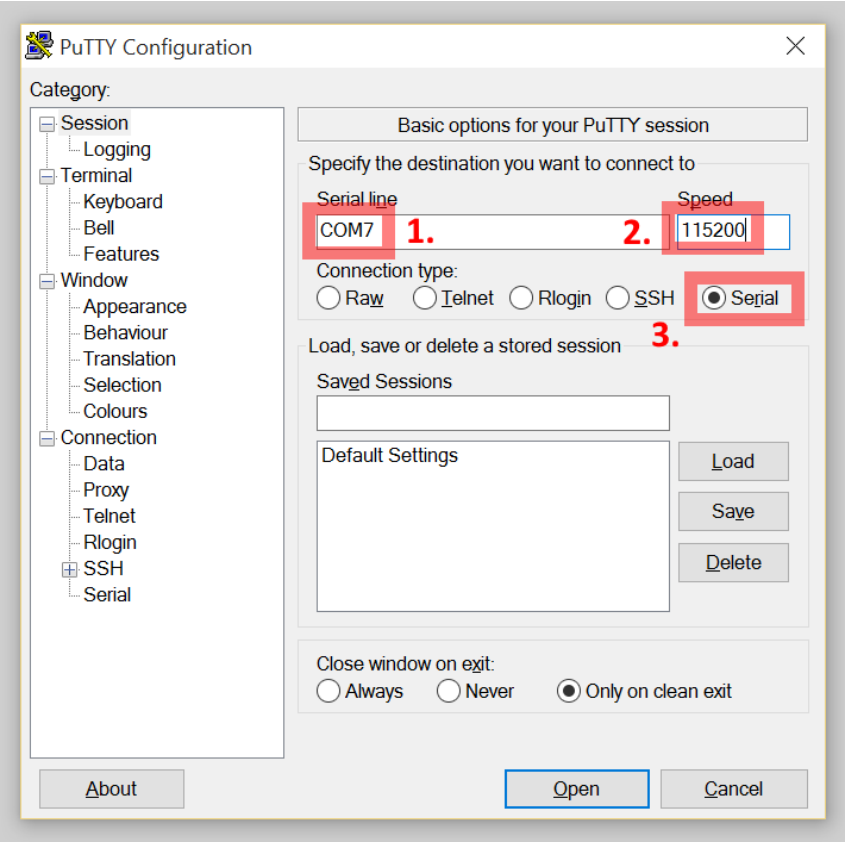
*Das rote Kabel muss nicht angeschlossen werden

Hiernach können Sie das andere Ende des USB-TTL-Kabels in einen freien USB-Port Ihres PC stecken. Sollten Sie bislang den Raspberry Pi nicht vorbereitet haben, stecken Sie nun die beiliegende microSD Karte in den dazugehörigen Slot auf der Unterseite und verbinden Sie das microUSB-Netzteil an den microUSB Eingang des Raspberry Pi. Stecken Sie das Netzteil noch nicht in die Wandsteckdose an.

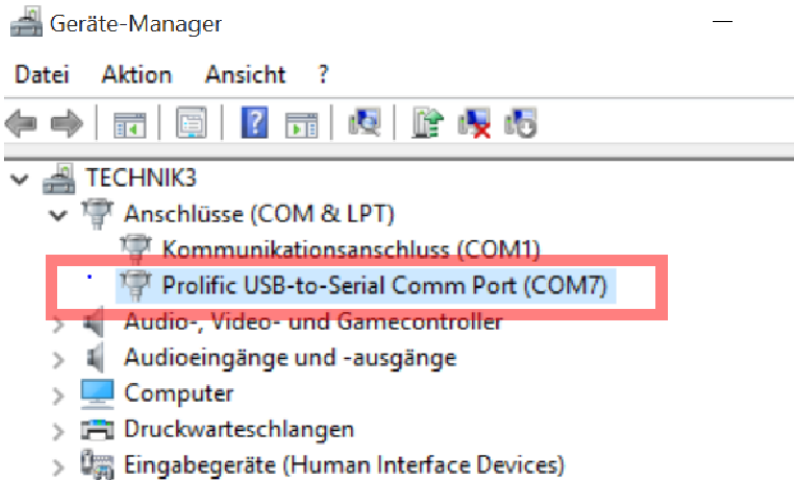
Die folgenden Schritte sind abhängig von Ihrem verwendeten Betriebsystem—im Folgenden zeigen wir auf, wie Sie die Terminal-Verbindung an einem Windows-PC durchführen können; die Vorgehensweise auf anderen Betriebsystemen sind jedoch in ihren Grundzügen ähnlich.

Für die Verbindung zur Terminalkonsole verwenden wir das Programm Putty. Dieses ist als Freeware unter http://www.putty.org/ zum Download verfügbar und wurde unter der MIT License veröffentlicht.

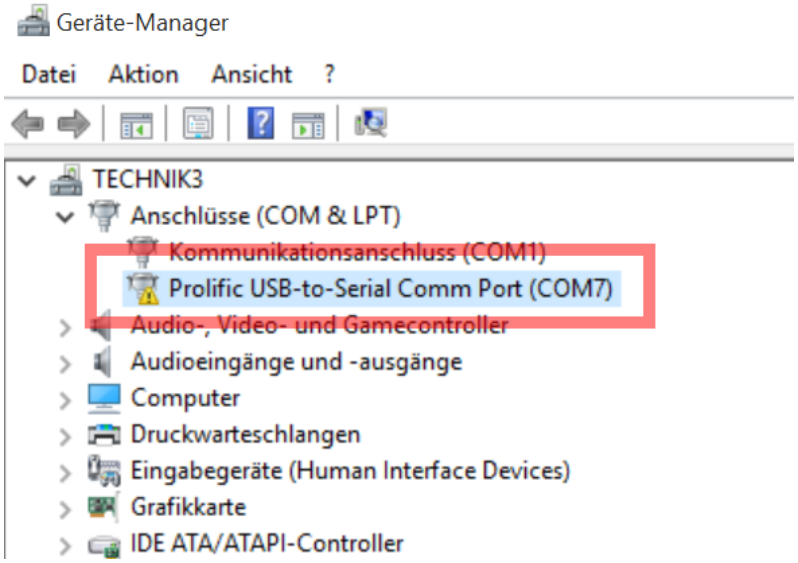
Laden Sie die „putty.exe“ auf der o.g. Website herunter und starten Sie das Programm. Sie Sehen nun die folgende Maske—Dort müssen die beschriebenen Werte eingefügt bzw. geändert werden:



*Die unter Punkt 1 aufgezeigte COM-Nummer ist abhängig von Ihrem verwendeten System und kann variieren. Um die richtige COM-Nummer zu erhalten, gehen Sie in den Windows Gerätemanager und schauen nach, welche COM-Nummer dem USB-TTL Kabel vom Betriebsystem vergeben wurde.
Hierzu gehen Sie im Gerätemanager unter „Anschlüsse“

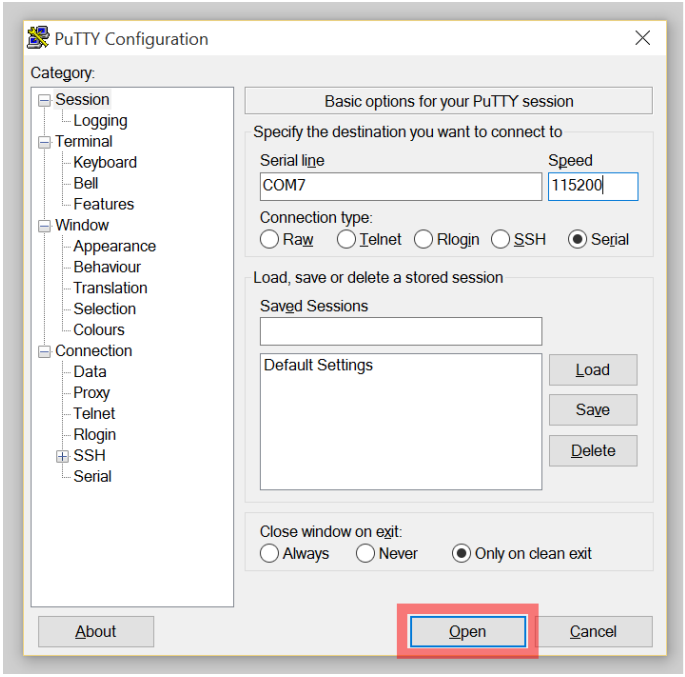


In unserem Fall ist dies COM7. Sollten Sie jedoch ein gelbes Ausrufezeichen neben dem „USB-to-Serial“ Adapter-Symbol haben, wie im folgenden Bild…

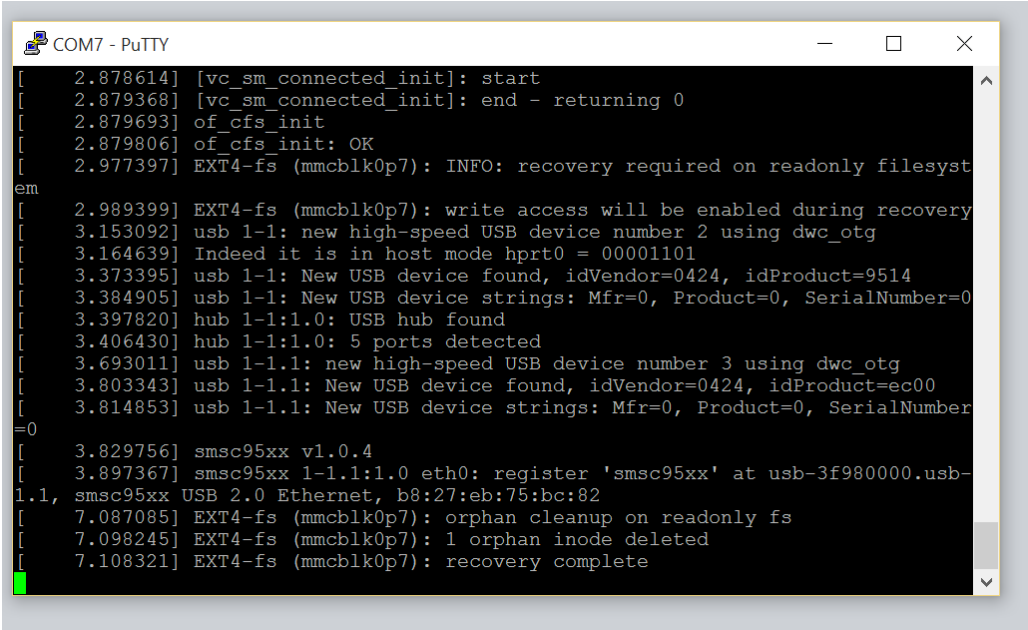


… so ist die Treiber-Installation von Windows für den Adapter noch nicht vollständig. Um diese abzuschließen, folgen Sie bitte der Treiber-Installation Anleitung im folgenden Link.

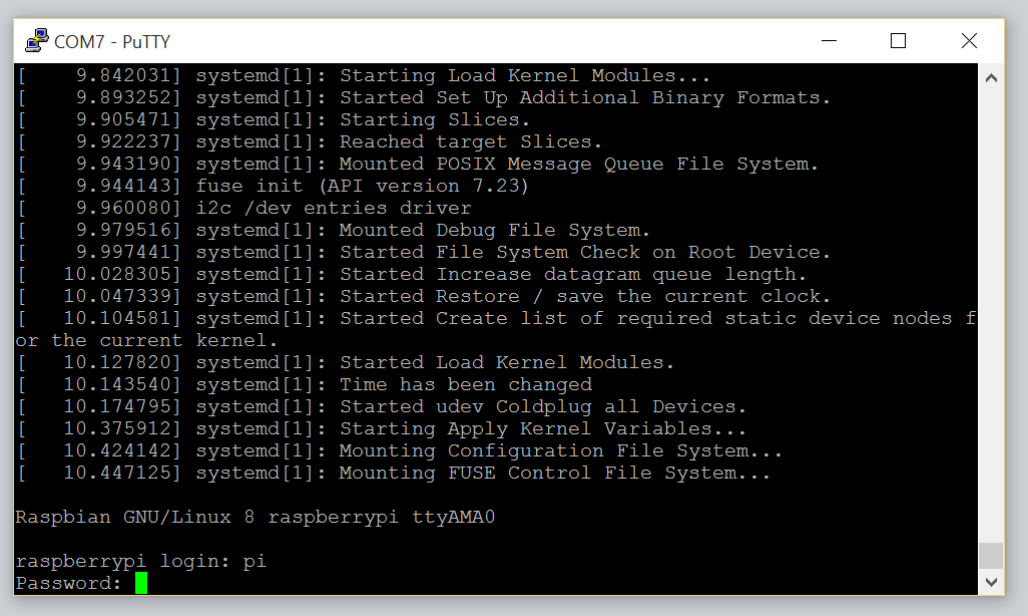
Nach dieser Vorkonfiguration kann mit einem Drücken auf „Open“ die serielle Verbindung aufgebaut werden.



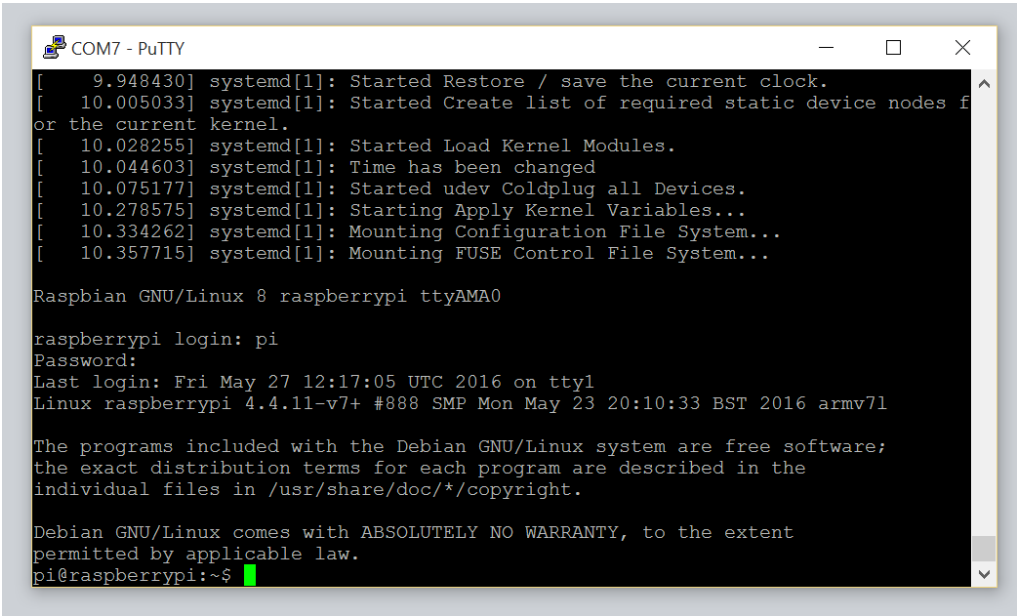
Zu allererst wird Ihnen nur ein schwarzes Fenster gezeigt—nun können Sie das Netzteil in die Wandsteckdose stecken und somit den Raspberry Pi 3 starten.
Nun sollte nach kurzer Zeit die Terminalausgabe beginnen



Nachdem das Betriebsystem komplett hochgefahren ist, wird als letzte Zeile „raspberrypi login:“ angezeigt. Nun können Sie sich anmelden indem Sie „pi“ als Benutzernamen und „raspberry“ als passwort eingeben und die jeweiligen Eingaben mit Enter bestätigen (das Passwort wird bei der Eingabe nicht angezeigt; dieses muss man nach der Eingabe „blind“ bestätigen)

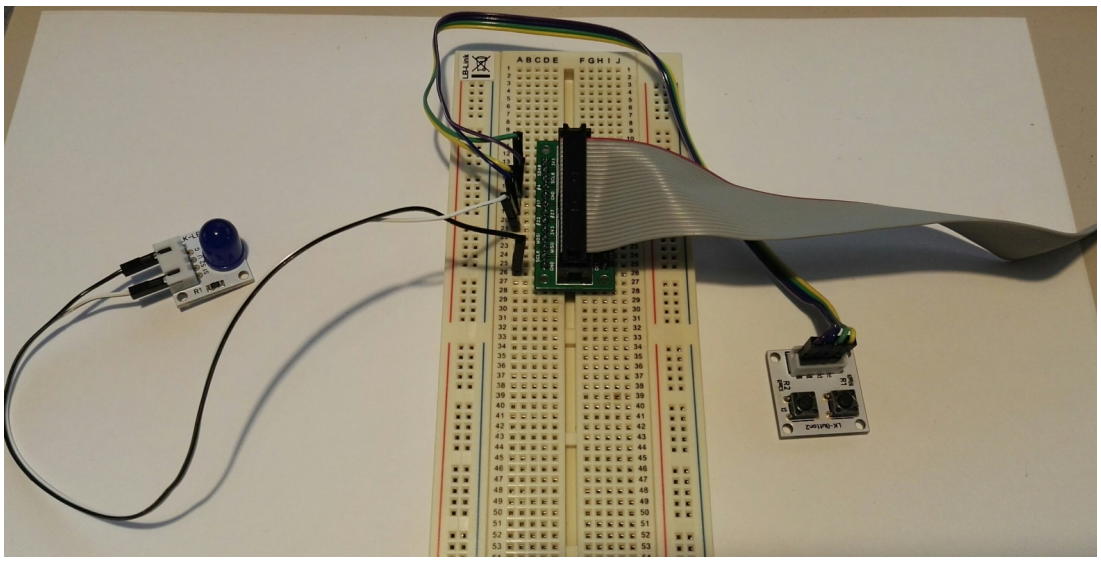


Danach sind Sie mit dem Raspbian Terminal per serieller Kommunikationsschnittstelle verbunden.



Abfrage eines Buttons / Leuchten einer LED mittels Python

Schließen die beiliegenden Komponenten, wie auf dem folgenden Bild an



Pin-Verbindungsübersicht:

Button-Module:
- S1 ——> GPIO4 / #4 [Raspberry Pi]
- S2 ——> GPIO17 / #17 [Raspberry Pi]
- V ——> 3V3
- G ——> GND [Raspberry Pi]

LED-Module:
- S1 ——> GPIO22 / #22 [Raspberry Pi]
- S2 ——> nicht verbunden
- V ——> nicht verbunden
- G ——> GND [Raspberry Pi]

Nachdem alles angeschlossen ist, muss nun die Datei mit dem ausführenden Programm generiert werden. Dazu verbinden Sie sich mit dem Terminal des Raspbian Betriebssystems und geben den folgenden Befehl ein…

```
nano LED_Button_Test.py
```

… um die Datei „LED_Button_Test.py“ zu erstellen. Nun befinden Sie sich im Editor. Im folgenden sehen Sie ein Code-Beispiel geschrieben in der Code-Sprache Python. Dieses muss nun in den Editor kopiert werden.

```
# Benoetigte Module werden importiert und eingerichtet
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)

# Hier wird der Eingangs-Pin deklariert, an dem der Button angeschlossen ist.
Button_S1_PIN = 4
# Hier wird der Ausgang-Pin deklariert, an dem die LED angeschlossen ist.
LED_PIN = 22
GPIO.setup(Button_S1_PIN, GPIO.IN)
GPIO.setup(LED_PIN, GPIO.OUT)

print "Button-LED-Test [druecken Sie STRG+C, um den Test zu beenden]"

# Hauptprogrammschleife
try:
    while True:
        if (GPIO.input(Button_S1_PIN) == 1):
            GPIO.output(LED_PIN,1)
        else:
            GPIO.output(LED_PIN,0)

# Aufraeumarbeiten nachdem das Programm beendet wurde
except KeyboardInterrupt:
    GPIO.cleanup()
```

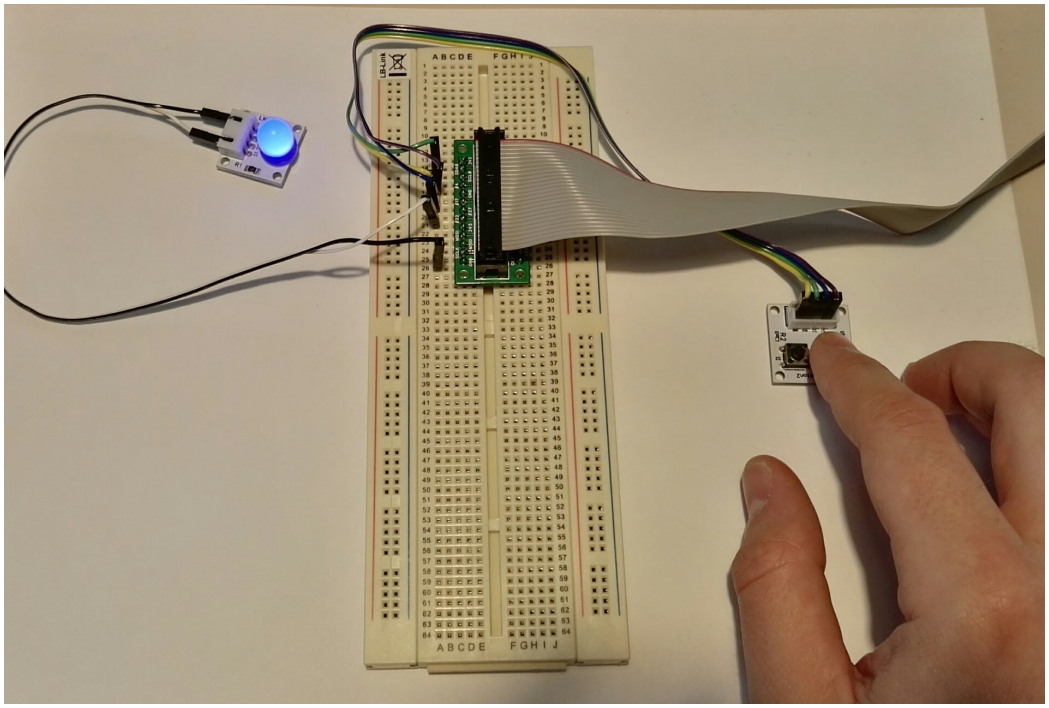
* Vorsicht beim kopieren/abschreiben: Bei der Programmiersprache Python sind die Einrückungen von Bedeutung; dadurch wird eine Programmstruktur aufgebaut*

Nachdem man den Code kopiert hat, kann man den Editor mit der Tastenkombination „Strg+X“ beenden. Man wird dann gefragt, ob man die Datei speichern möchte, was man mit „Y“ bestätigen muss.

Nachdem man nach dem Schließen des Editors wieder im Terminal gelangt, kann man nun das Programm mit folgenden Befehl starten:

```
sudo python LED_Button_Test.py
```

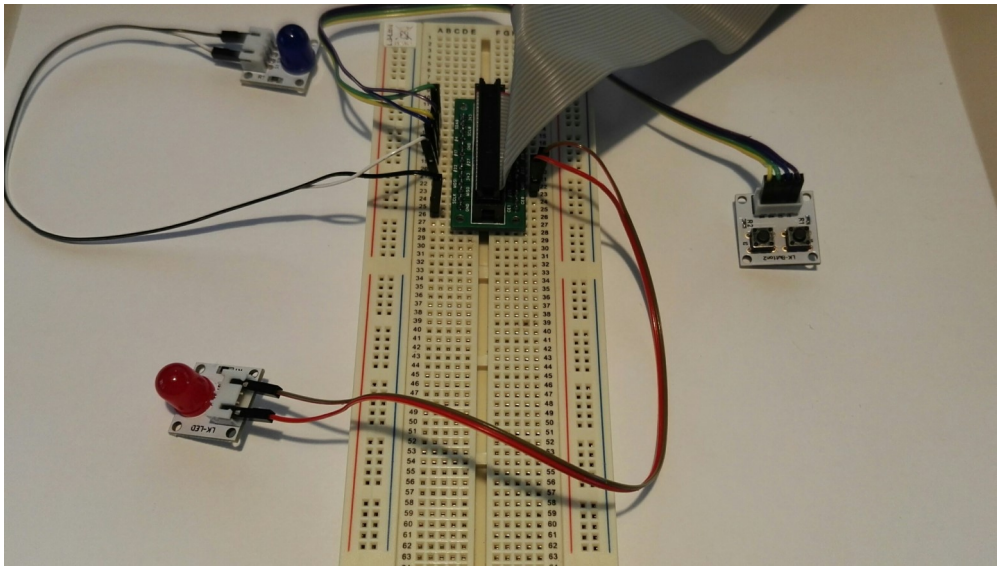
Wird nun der rechte Button1 gedrückt, so leuchtet die LED auf.



Das Programm kann man danach mit Drücken der Tastenkombination „Strg+C“ beenden.

Erweiterung des Beispiels um eine zweite LED

Schließen die beiliegenden Komponenten, wie auf dem folgenden Bild an



Pin-Verbindungsübersicht:

Button-Module:
- S1 ——> GPIO4 / #4 [Raspberry Pi]
- S2 ——> GPIO17 / #17 [Raspberry Pi]
- V ——> 3V3
- G ——> GND [Raspberry Pi]

LED-Module-Blau:
- S1 ——> GPIO22 / #22 [Raspberry Pi]
- S2 ——> nicht verbunden
- V ——> nicht verbunden
- G ——> GND [Raspberry Pi]

LED-Module-Rot:
- S1 ——> GPIO24 / #24 [Raspberry Pi]
- S2 ——> nicht verbunden
- V ——> nicht verbunden
- G ——> GND [Raspberry Pi]

Das vorherige Beispiel kann, da auf dem Button-Modul zwei Knöpfe vorhanden sind, um eine zweite LED erweitert werden. Hierzu geben Sie folgenden Befehl...

```
nano 2LED_2Button_Test.py
```

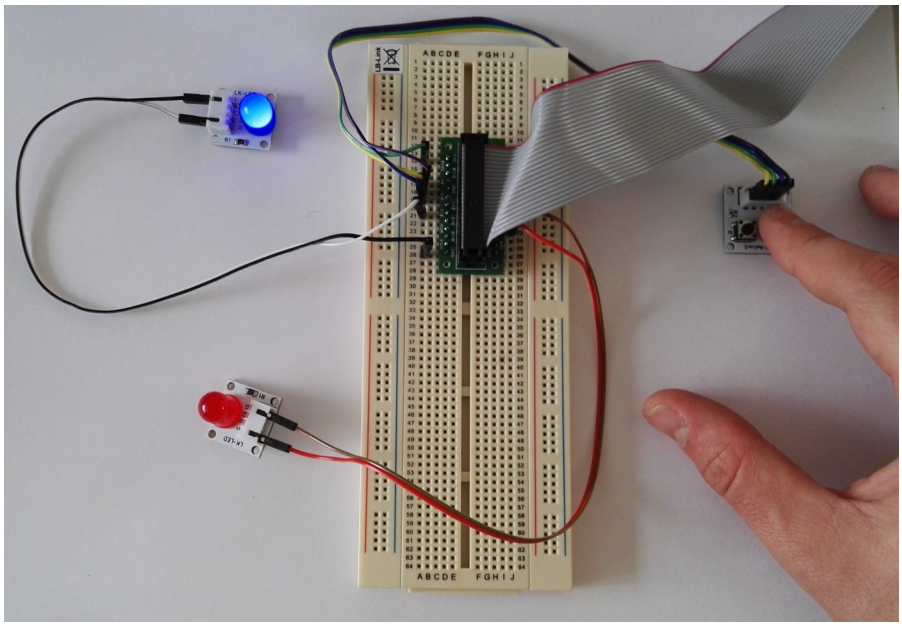
... in die Konsole ein, um die Datei „2LED_2Button_Test.py“ zu erstellen. Das Code-Beispiel von vorhin, ist im Folgenden in einer abgeänderten Fassung, in der der zweite Knopf und die zweite LED integriert sind.

```
# Benoetigte Module werden importiert und eingerichtet
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)

# Hier wird der Eingangs-Pin deklariert, an dem der Button angeschlossen ist.
Button_S1_PIN = 4
Button_S2_PIN = 17
# Hier wird der Ausgang-Pin deklariert, an dem die LED angeschlossen ist.
LED_BLAU_PIN = 22
LED_ROT_PIN = 24

GPIO.setup(Button_S1_PIN, GPIO.IN)
GPIO.setup(Button_S2_PIN, GPIO.IN)
GPIO.setup(LED_BLAU_PIN, GPIO.OUT)
GPIO.setup(LED_ROT_PIN, GPIO.OUT)

print "Button-LED-Test [druecken Sie STRG+C, um den Test zu beenden]"

# Hauptprogrammschleife
try:
    while True:
        if (GPIO.input(Button_S1_PIN) == 1):
            GPIO.output(LED_BLAU_PIN,1)
        else:
            GPIO.output(LED_BLAU_PIN,0)
        if (GPIO.input(Button_S2_PIN) == 1):
            GPIO.output(LED_ROT_PIN,1)
        else:
            GPIO.output(LED_ROT_PIN,0)

# Aufraeumarbeiten nachdem das Programm beendet wurde
except KeyboardInterrupt:
    GPIO.cleanup()
```

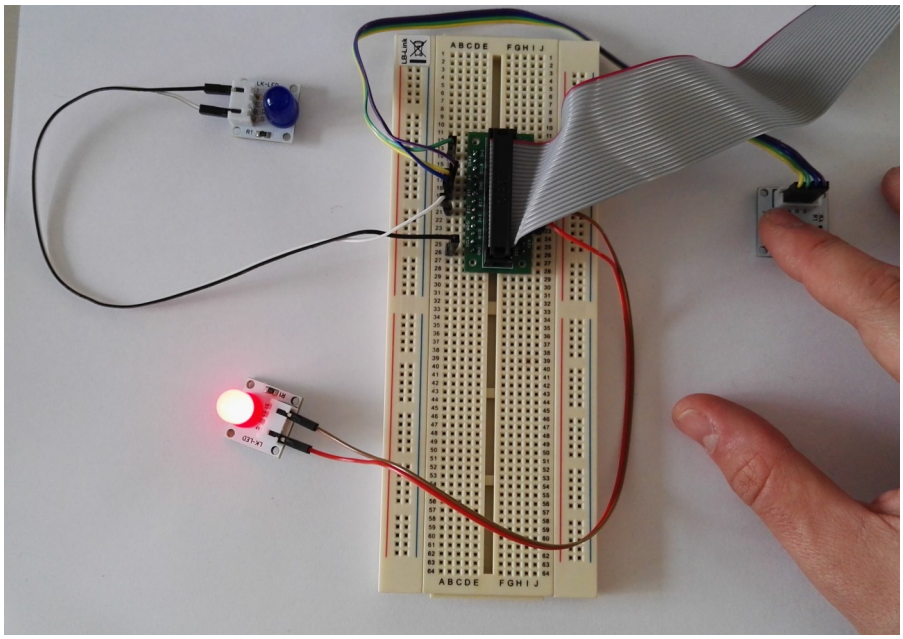
Nachdem man nach dem Schließen des Editors wieder im Terminal gelangt, kann man nun das Programm mit folgenden Befehl starten:

```
sudo python 2LED_2Button_Test.py
```

Wird nun der rechte Button1 gedrückt, so leuchtet die blaue LED auf.

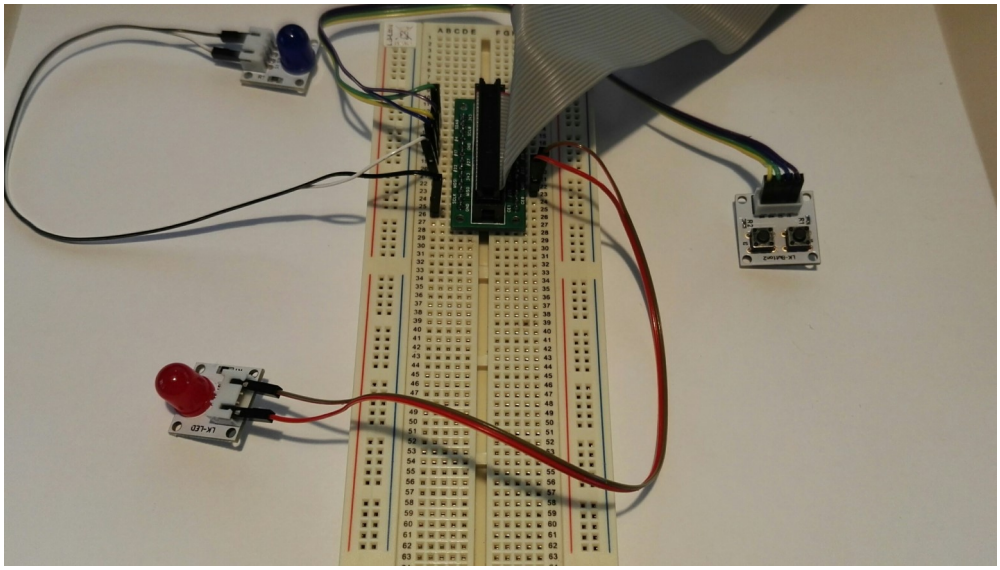


Wird der linke Button2 gedrückt, so leuchtet nun die rote LED auf.



Knopf durch Bewegungsmelder ersetzen

Schließen die beiliegenden Komponenten, wie auf dem folgenden Bild an



Pin-Verbindungsübersicht:

PIR-Modul:
- S1 ——> GPIO4 / #4 [Raspberry Pi]
- S2 ——> nicht verbunden
- V ——> 3V3
- G ——> GND [Raspberry Pi]

LED-Module-Rot:
- S1 ——> GPIO24 / #24 [Raspberry Pi]
- S2 ——> nicht verbunden
- V ——> nicht verbunden
- G ——> GND [Raspberry Pi]

Anstatt , dass ein Knopf gedrückt werden muss um die LED leuchten zu lassen, kann auch das PIR-Modul verwendet, was die Funktion eines Bewegungsmelders beinhaltet. Geben Sie folgenden Befehl...

```
nano PIR_LED_Test.py
```

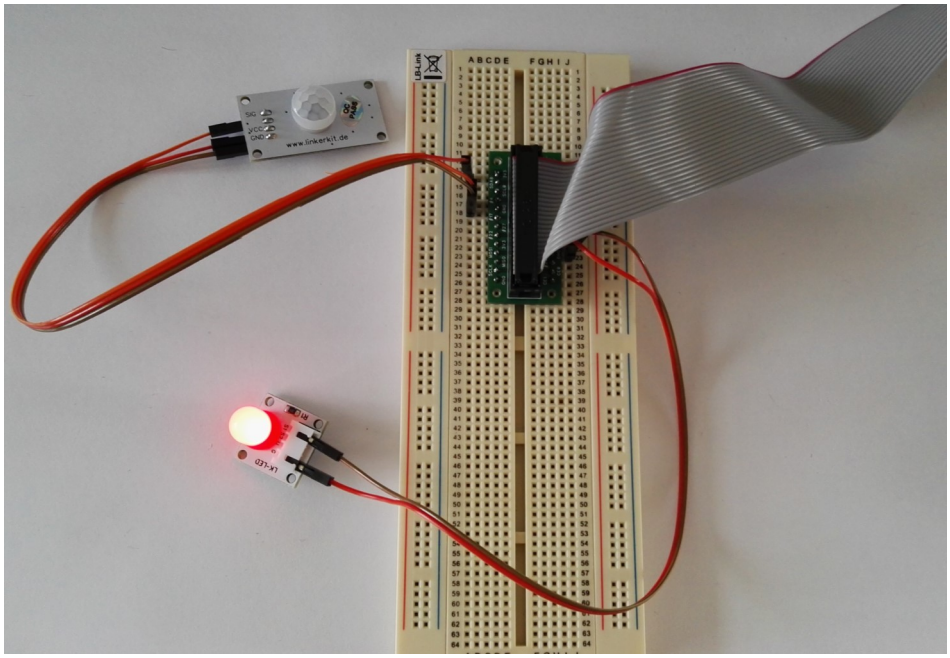
... in die Konsole ein, um die Datei „PIR_LED_Test.py“ zu erstellen und kopieren Sie das unten stehende Codebeispiel in den Editor.

```
# Benoetigte Module werden importiert und eingerichtet
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)

# Hier wird der Eingangs-Pin deklariert, an dem der PIR-Sensor
# angeschlossen ist. Vorkonfiguriert ist hierbei der LinkerKit Port:
# [12|13|V|G]
GPIO_PIN = 4
GPIO.setup(GPIO_PIN, GPIO.IN)

# Hier wird der Ausgangs-Pin deklariert, an dem z.B. eine Ausgabe LED
# oder ein LinkerKit Buzzer angeschlossen ist. Vorkonfiguriert ist
# hierbei der LinkerKit Port: [15|16|V|G]
OUTPUT_PIN = 24
GPIO.setup(OUTPUT_PIN, GPIO.OUT)

print "PIR-Sensor-Test [druecken Sie STRG+C, um den Test zu beenden]"

# Diese AusgabeFunktion wird bei Signaldetektion ausgefuehrt
def ausgabeFunktion(null):
        print("Bewegung erkannt")
    # Das Hauptprogramm wird solange unterbrochen, solang der
    # Sensor an seinem Ausgang die Detektion der Bewegung ausgibt
    # (3s). Dabei wird am Ausgangs-Pin ebenfalls ein Signal
    # ausgegeben...
    GPIO.output(OUTPUT_PIN,True)
    while (GPIO.input(GPIO_PIN)):
      time.sleep(1)
    # ...und danach wieder ausgeschaltet
    GPIO.output(OUTPUT_PIN,False)

# Beim Detektieren eines Signals (fallende Signalflanke) wird die
# Ausgabefunktion ausgeloest
GPIO.add_event_detect(GPIO_PIN, GPIO.RISING, callback=ausgabeFunktion, bouncetime=100)

# Hauptprogrammschleife
try:
        while True:
                time.sleep(1)

# Aufraeumarbeiten nachdem das Programm beendet wurde
except KeyboardInterrupt:
        GPIO.cleanup()
```

Nachdem man nach dem Schließen des Editors wieder im Terminal gelangt, kann man nun das Programm mit folgenden Befehl starten:

```
sudo python PIR_LED_Test.py
```

Wird nun vor dem PIR-Modul eine Bewegung erkannt, so leuchtet die LED auf.



Die maximale Reichweite der Bewegungserkennung kann über den Empfindlichkeitsregler an der Unterseite des PIR-Moduls mittels eines Kreuzschraubenziehers zwischen 3m und 6m eingestellt
werden. (Das Einstell-Potentiometer hat hierbei kein Anfangs- und Endpunkt)