

Raspberry
ELECTRONIC EDUCATION SET

INHALTSVERZEICHNIS

1. Einführung
2. Das Breadboard
3. Gyroskop & Beschleunigungssensor
4. Magnetfeldsensor
5. Explorer 700
6. Informations- und Rücknahmepflicht
7. Support

1. EINFÜHRUNG

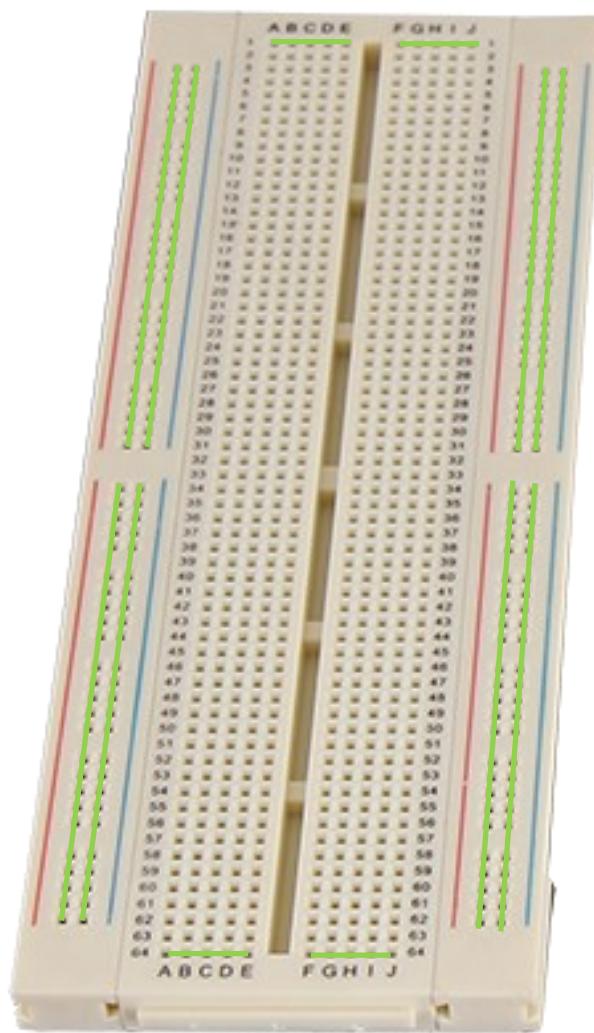
Sehr geehrter Kunde,
vielen Dank, dass Sie sich für unser Produkt entschieden haben.
Im Folgenden zeigen wir Ihnen, was bei der Inbetriebnahme und der Verwendung zu beachten ist. Sollten Sie während der Verwendung unerwartet auf Probleme stoßen, so können Sie uns selbstverständlich gerne kontaktieren.

Dieses Raspberry Pi Kit führt Sie in die Welt der Sensoren und der Programmierung. In diesem Set sind Sensoren und das Explorer-Board 700 enthalten um einen vereinfachten Einstieg zu ermöglichen, denn Sie können erlernen einen Magnetfeldsensor und einen Beschleunigungssensor vereint mit einem Gyroskop zu verwenden. Durch die 32 GB MSD-Karte mit dem vorbereitetem Image wird der Einstieg um ein Vielfaches erleichtert.

Im gesamten Set ist ein Dish-Board aus Acryl, ein Verbindungskabelset, eine 32 GB MSD-Karte mit dem vorbereitetem Image Noobs-Betriebssystem, ein Modul mit einem kombinierten Gyroskop- und Beschleunigungssensor, ein Magnetfeldsensor, das Explorer-Board 700 enthalten in einem Kunststoff Mehrzweckkoffer.

2. DAS BREADBOARD

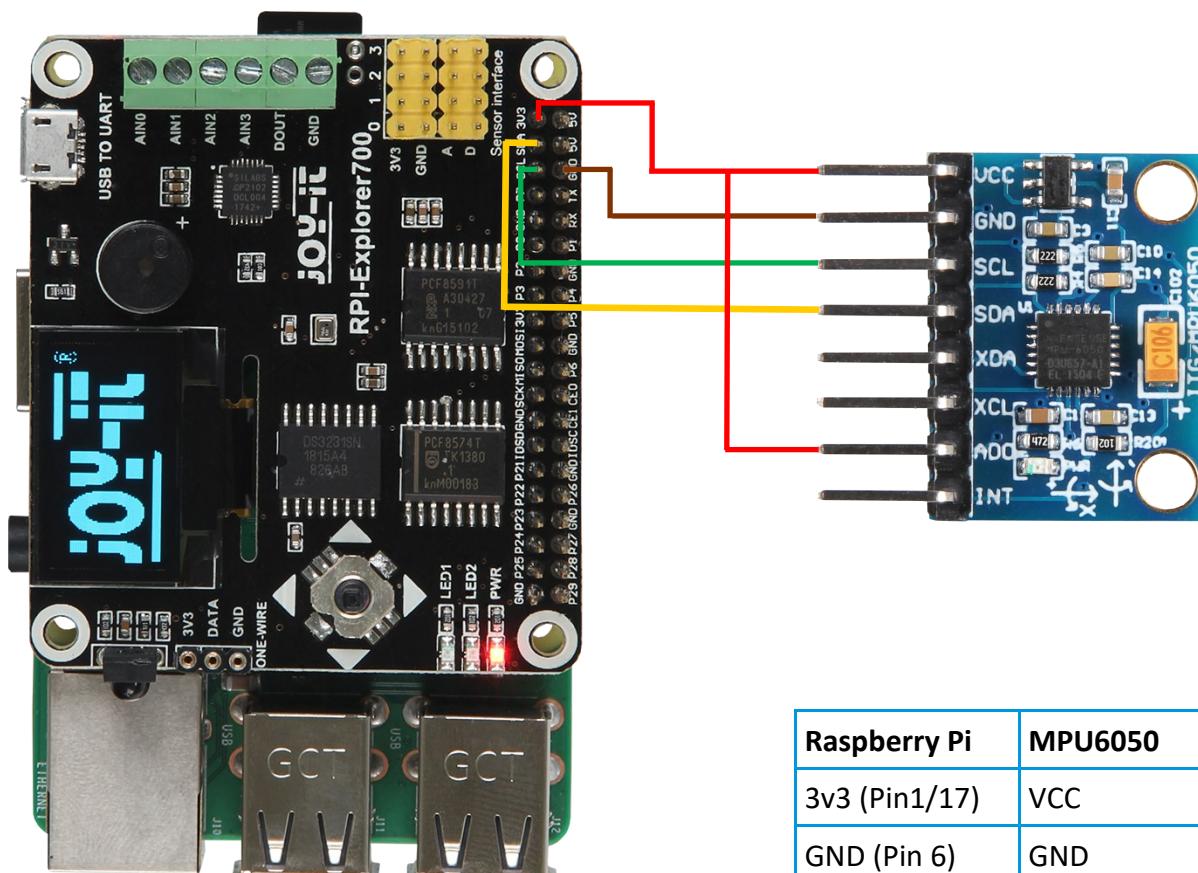
Mit Hilfe des Breadboards lassen sich benutzerdefinierte Schaltungen einfach und übersichtlich gestalten.



Die ersten und letzten 30 Spalten + und - sind jeweils senkrecht durchverbunden.
Die Zeilen sind jeweils von A bis E und von F bis J waagerecht durchverbunden.

Diese durchverbundenen Spalten und Zeilen sind hier **grün** markiert.

3. GYROSkop & BESCHLEUNIGUNGSSENSOR



Raspberry Pi	MPU6050
3v3 (Pin1/17)	VCC
GND (Pin 6)	GND
SCL (Pin 5)	SCL
SDA (Pin 3)	SDA
3v3 (Pin 1/17)	ADO

ADO müssen Sie nur anschließen, wenn Sie zeitgleich das Explorer 700 verwenden. ADO wird verwendet um die I2C-Adresse des Sensors zu ändern, da die Adresse bereits vom Explorer 700 genutzt wird.

Wenn Sie ADO nicht anschließen müssen Sie jedoch die I2C-Adresse im Beispielcode anpassen und auf 0x68 ändern.

```
address = 0x68      # I2C-Adresse
```

3. GYROSkop & BESCHLEUNIGUNGSSENSOR

Codebeispiel

Nachdem der Sensor angeschlossen ist, können Sie das Programm, welches sich bereits auf dem vorbereitetem Image befindet, starten.

Dazu müssen Sie zunächst mit folgendem Befehl in das Dokumente-Verzeichnis wechseln:

```
cd Documents/
```

Nun können Sie das Programm mit folgendem Befehl starten:

```
sudo python MPU6050.py
```

Zur Übersicht können Sie sich den Quelltext auf dieser und der folgenden Seite noch einmal anschauen.

```
#!/usr/bin/python
import smbus
import math
import time

power_mgmt_1 = 0x6b
power_mgmt_2 = 0x6c

def read_byte(reg):
    return bus.read_byte_data(address, reg)

def read_word(reg):
    h = bus.read_byte_data(address, reg)
    l = bus.read_byte_data(address, reg+1)
    value = (h << 8) + l
    return value

def read_word_2c(reg):
    val = read_word(reg)
    if (val >= 0x8000):
        return -((65535 - val) + 1)
    else:
        return val

def dist(a,b):
    return math.sqrt((a*a)+(b*b))

def get_y_rotation(x,y,z):
    radians = math.atan2(x, dist(y,z))
    return -math.degrees(radians)

def get_x_rotation(x,y,z):
    radians = math.atan2(y, dist(x,z))
    return math.degrees(radians)
```

3. GYROSkop & BESCHLEUNIGUNGSSENSOR

```

bus = smbus.SMBus(1)
address = 0x69          # I2C-Adresse

bus.write_byte_data(address, power_mgmt_1, 0)

while True:
    gyroskop_xout = read_word_2c(0x43)
    gyroskop_yout = read_word_2c(0x45)
    gyroskop_zout = read_word_2c(0x47)

    print
    print ("MPU6050")
    print ("-----")

    print ("gyroskop_xout: "), ("%5d" % gyroskop_xout), (" skaliert: "), (gyroskop_xout / 131)
    print ("gyroskop_yout: "), ("%5d" % gyroskop_yout), (" skaliert: "), (gyroskop_yout / 131)
    print ("gyroskop_zout: "), ("%5d" % gyroskop_zout), (" skaliert: "), (gyroskop_zout / 131)
    print ("")
    beschleunigung_xout = read_word_2c(0x3b)
    beschleunigung_yout = read_word_2c(0x3d)
    beschleunigung_zout = read_word_2c(0x3f)
    print ("")
    beschleunigung_xout_skaliert = beschleunigung_xout / 16384.0
    beschleunigung_yout_skaliert = beschleunigung_yout / 16384.0
    beschleunigung_zout_skaliert = beschleunigung_zout / 16384.0

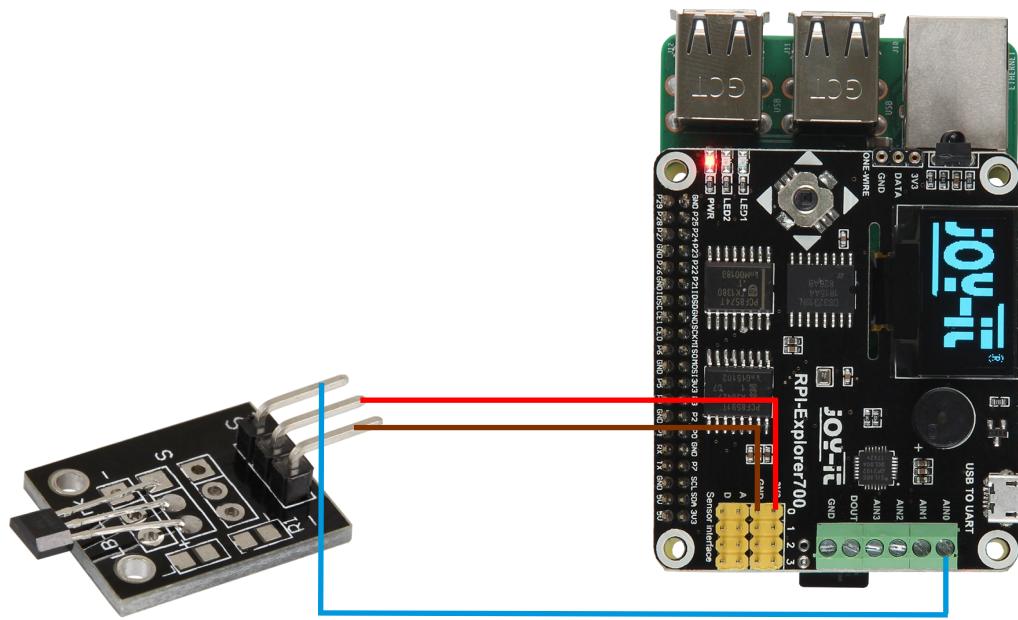
    print ("beschleunigung_xout: "), ("%6d" % beschleunigung_xout), (" skaliert: "), beschleu-
    nigung_xout_skaliert
    print ("beschleunigung_yout: "), ("%6d" % beschleunigung_yout), (" skaliert: "), beschleu-
    nigung_yout_skaliert
    print ("beschleunigung_zout: "), ("%6d" % beschleunigung_zout), (" skaliert: "), beschleu-
    nigung_zout_skaliert
    print ("")
    print ("X Rotation: ") , get_x_rotation(beschleunigung_xout_skaliert, beschleuni-
    gung_yout_skaliert, beschleunigung_zout_skaliert)
    print ("Y Rotation: ") , get_y_rotation(beschleunigung_xout_skaliert, beschleuni-
    gung_yout_skaliert, beschleunigung_zout_skaliert)
    print ("-----")
    time.sleep(1) # Zeit zwischen den Messungen in Sekunden

```

4. MAGNETFELDSENSOR

Da es sich bei dem Magnetfeldsensor um einen analogen Sensor handelt und der Raspberry Pi keine analogen Eingänge besitzt, müssen Sie, um diesen Sensor verwenden zu können, einen ADC (Analog-Digital Konverter) benutzen.

In diesem Beispiel verwenden wir den im Explorer 700 integrierten ADC, welcher im Lieferumfang enthalten ist.



Explorer700	KY035BM
3v3	VCC
GND	GND
AIN0	Signal

4. MAGNETFELDSENSOR

Codebeispiel

Nachdem der Sensor angeschlossen ist, können Sie das Programm, welches sich bereits auf dem vorbereitetem Image befindet, starten.

Dazu müssen Sie zunächst mit folgendem Befehl in das Dokumente-Verzeichnis wechseln:

```
cd /Documents
```

Nun können Sie das Programm mit folgendem Befehl starten:

```
sudo python Magnet.py
```

Zur Übersicht können Sie sich den Quelltext hier noch einmal anschauen.

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
import smbus
import time

address = 0x48
A0 = 0x40
A1 = 0x41
A2 = 0x42
A3 = 0x43
bus = smbus.SMBus(1)

while True:
    bus.write_byte(address,A0)
    value = bus.read_byte(address)

    if value == 128:
        print("Kein Magnetfeld erkannt! "), (" Wert: "), (value)

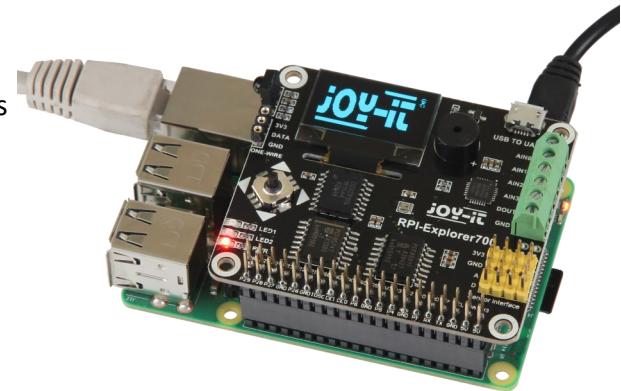
    if value < 128:
        print("Negatives Magnetfeld erkannt! "), (" Wert: "), (value)

    if value > 128:
        print("Positives Magnetfeld erkannt! "), (" Wert: "), (value)

    time.sleep(1)
    print("-----")
```

5. EXPLORER 700

Durch das vorbereitete Image können Sie die Funktionen des Explorer 700 in vollem Umfang nutzen.



LEDs

Auf dem Explorer-Board sind zwei Status-LEDs verbaut, die wir in diesem Beispiel benutzen werden. Wechseln Sie mit folgendem Befehl in das Verzeichnis der Beispieldatei:

```
cd Desktop/RB-Explorer700/LED/python
```

Mit dem folgenden Befehl können Sie das Programm starten, welches LED1 blinken lässt.

```
sudo python led.py
```

Sie können die LED auch langsam immer heller werden lassen die geschieht mit folgendem Befehl:

```
sudo python pwm.py
```

6. EXPLORER 700

Joystick

In diesem Beispiel gibt der Buzzer einen Ton aus wenn der Joystick bewegt wird, außerdem wird die Richtung in der Konsole ausgegeben.

Dazu wechseln Sie mit folgendem Befehl in das Verzeichnis der Beispieldatei:

```
cd Desktop/RB-Explorer700/PCF8574/python
```

Mit folgendem Befehl können Sie nun das Programm starten:

```
sudo python pcf8574.py
```

Um das Herunterdrücken des Joystick anzeigen zulassen, müssen Sie ein anderes Programm starten. Es zeigt in der Konsole an wenn der Joystick herunter gedrückt wurde.

Dazu wechseln Sie mit folgendem Befehl in das Verzeichnis der Beispieldatei:

```
cd Desktop/RB-Explorer700/KEY/python
```

Mit folgendem Befehl können Sie nun das Programm starten:

```
sudo python key.py
```

BMP280

Der BMP280 ist ein Luftdruck- und Temperatursensor, in diesem Beispiel benutzen wir Ihn um uns seine Werte anzeigen zulassen.

Wechseln Sie dafür mit folgendem Befehl in das Verzeichnis der Beispieldatei:

```
cd Desktop/RB-Explorer700/BMP280/python
```

Mit dem folgenden Befehl können Sie das Programm starten, welches Temperatur und Luftdruck anzeigt:

```
sudo python temperature-and-pressure.py
```

Mit dem folgendem Befehl können Sie sich die relative Höhe anzeigen lassen.

```
sudo python relative-altitude.py
```

6. EXPLORER 700

Echtzeituhr (RTC)

Für die Echtzeituhr müssen Sie eine Batterie (CR-1225) in die Batteriehalterung auf der Rückseite der Platine einlegen.

Mit Hilfe der Echtzeituhr kann Ihr Raspberry Pi selbst bei einem Stromausfall weiterhin die aktuelle Uhrzeit speichern.

In diesem Beispiel wird die in der RTC gespeicherte Uhrzeit ausgelesen.

Wechseln Sie dafür mit folgendem Befehl in das Verzeichnis der Beispieldatei:

```
cd Desktop/RB-Explorer700/DS3231/python
```

Mit folgendem Befehl können Sie das Programm starten:

```
sudo python ds3231.py
```

OLED

In diesem Beispiel wird ein Bild auf dem integrierten OLED-Display des Explorer-Boards angezeigt.

Wechseln Sie dafür mit folgendem Befehl in das Verzeichnis der Beispieldatei:

```
cd Desktop/RB-Explorer700/OLED/python
```

Mit folgendem Befehl können Sie das Programm starten:

```
sudo python image.py
```

IR-Receiver

In diesem Beispiel wird das vom IR-Receiver empfangene Signal in der Konsole ausgegeben.

Dazu wechseln Sie mit folgendem Befehl in das Verzeichnis der Beispieldatei:

```
cd Desktop/RB-Explorer700/IRM/python
```

Mit folgendem Befehl können Sie das Programm starten:

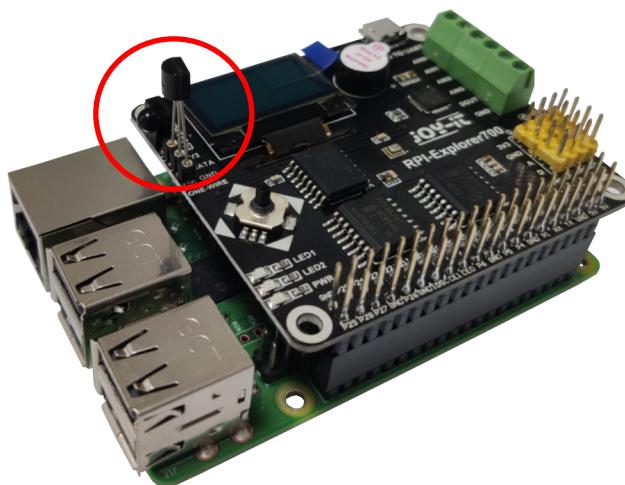
```
sudo python irm.py
```

6. EXPLORER 700

DS18B20

Der DS18B20 ist ein One-Wire Temperatursensor.

Bevor Sie den Sensor benutzen können müssen Sie ihn zunächst in das Board stecken.



Achten Sie unbedingt darauf, dass der Sensor wie im Bild zusehen, mit der abgerundeten Seite vom OLED weg zeigen muss.

Danach wechseln Sie mit folgendem Befehl in das Verzeichnis der Beispieldatei:

```
cd Desktop/RB-Explorer700/DS18B20/python
```

Mit folgendem Befehl können Sie das Programm starten:

```
sudo python ds18b20.py
```

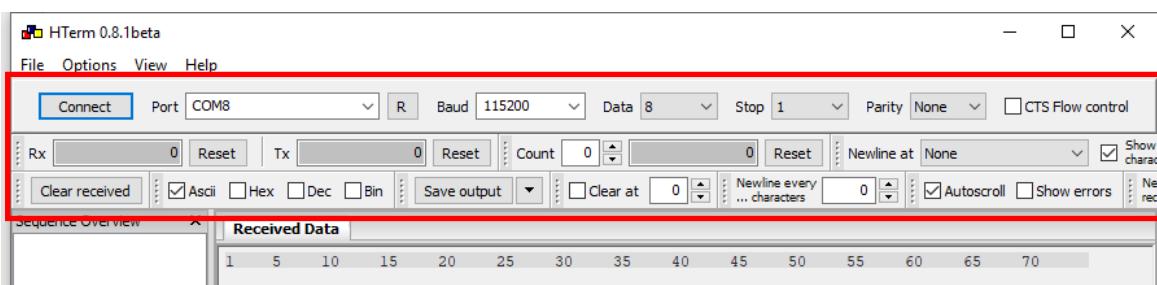
6. EXPLORER 700

USB to UART

Sie können über die USB to UART Schnittstelle Ihren Raspberry Pi mit einem Computer verbinden und kommunizieren lassen.

Dazu benutzen wir in diesem Beispiel HTerm. Laden Sie sich das Programm [hier](#) herunter und installieren Sie es.

Öffnen Sie HTerm und übernehmen Sie die Einstellungen, wie Sie in der untere Abbildung zu sehen ist. Jedoch kann der Port bei Ihnen abweichen, was Sie im Geräte-Manager überprüfen können.



Nun können Sie mit einem Klick auf Connect die Verbindung herstellen.

Als nächstes müssen Sie das Beispielprogramm auf Ihrem Raspberry Pi starten.

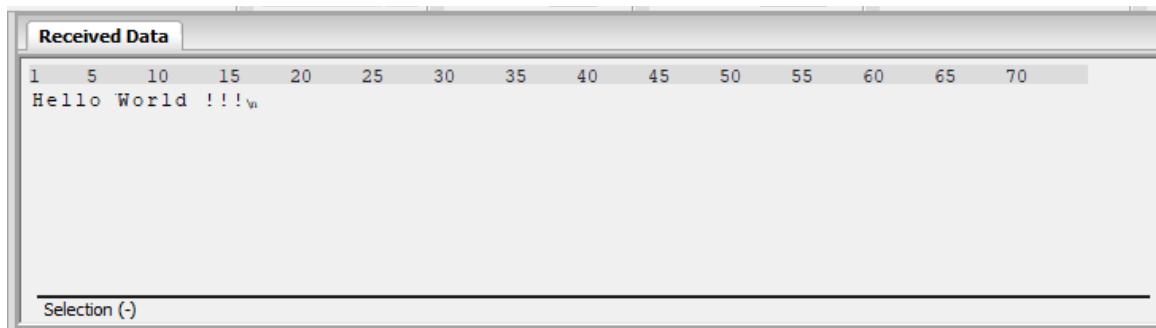
Wechseln Sie dafür mit folgendem Befehl in das Zielverzeichnis:

```
cd Desktop/RB-Explorer700/UART/RPI3/python
```

Nun können Sie mit dem folgendem Befehl das Programm ausführen:

```
sudo python uart.py
```

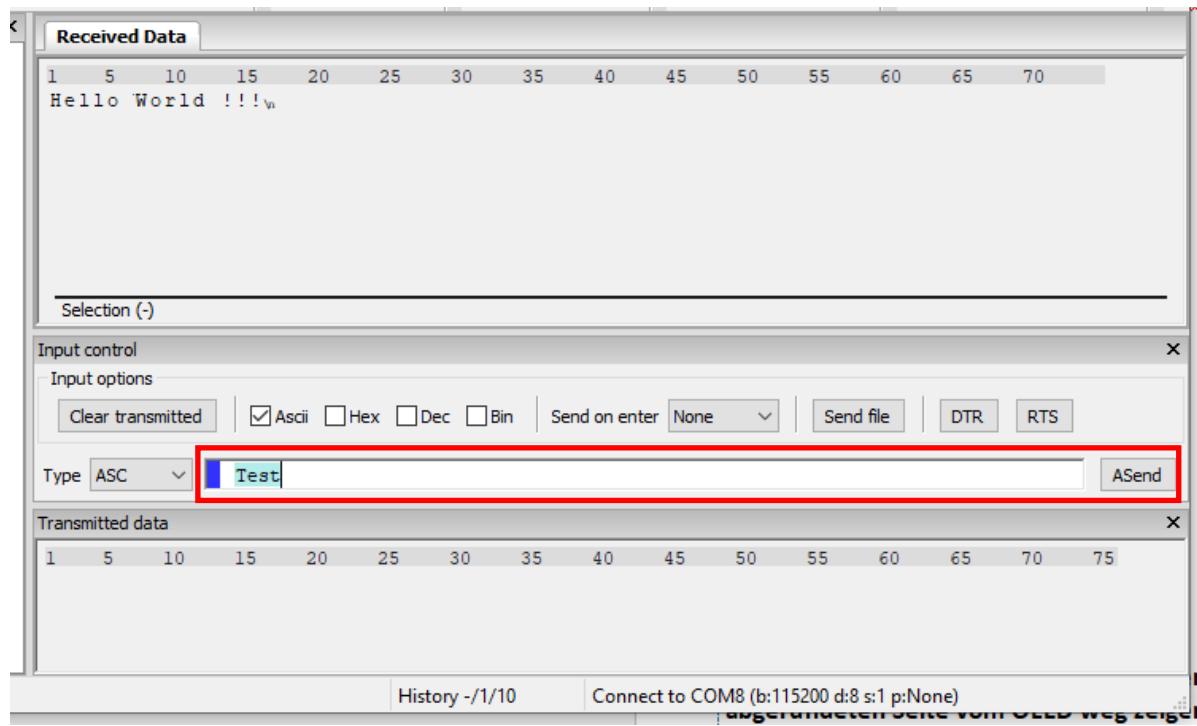
Der Raspberry Pi sendet nun „Hello World!!!“ an den PC und schickt alles was er gesendet bekommt wieder zurück.



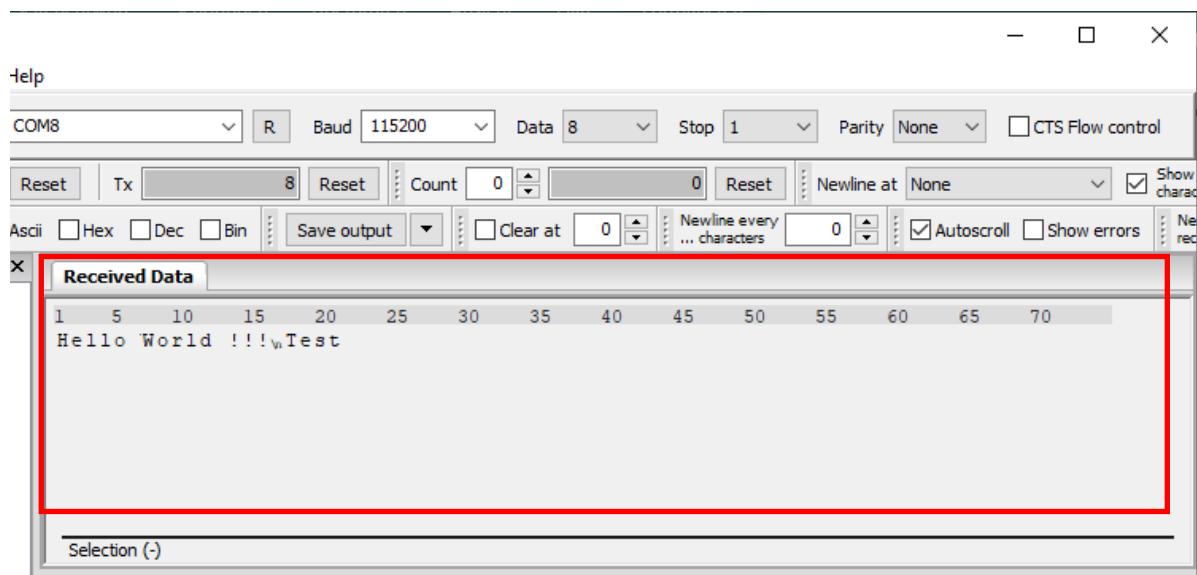
Diese Ausgabe sollten Sie erhalten, nachdem Sie das Programm gestartet haben.

6. EXPLORER 700

Nun können Sie etwas an den Raspberry Pi schicken, indem Sie es unten in das Feld eingeben und auf „ASend“ klicken.



Danach sollten Sie die folgende Ausgabe erhalten:



6. INFORMATIONS- UND RÜCKNAHMEPFLICHTEN

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 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 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.

7. SUPPORT

Makerfactory ist powered by Simac GmbH.

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

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

Dies ist eine Publikation der Simac Electronics Handel GmbH | Pascalstr. 8 | 47506 NV

Alle Rechte einschließlich Übersetzung vorbehalten. Reproduktion jeder Art, z. B. Fotokopie, Mikroverfilmung, oder die Erfassung in elektronischen Datenverarbeitungsanlagen, bedürfen der schriftlichen Genehmigung des Herausgebers. Nachdruck, auch auszugsweise, verboten. Die Publikation entspricht dem technischen Stand bei Drucklegung.

Copyright 2019 by Simac GmbH