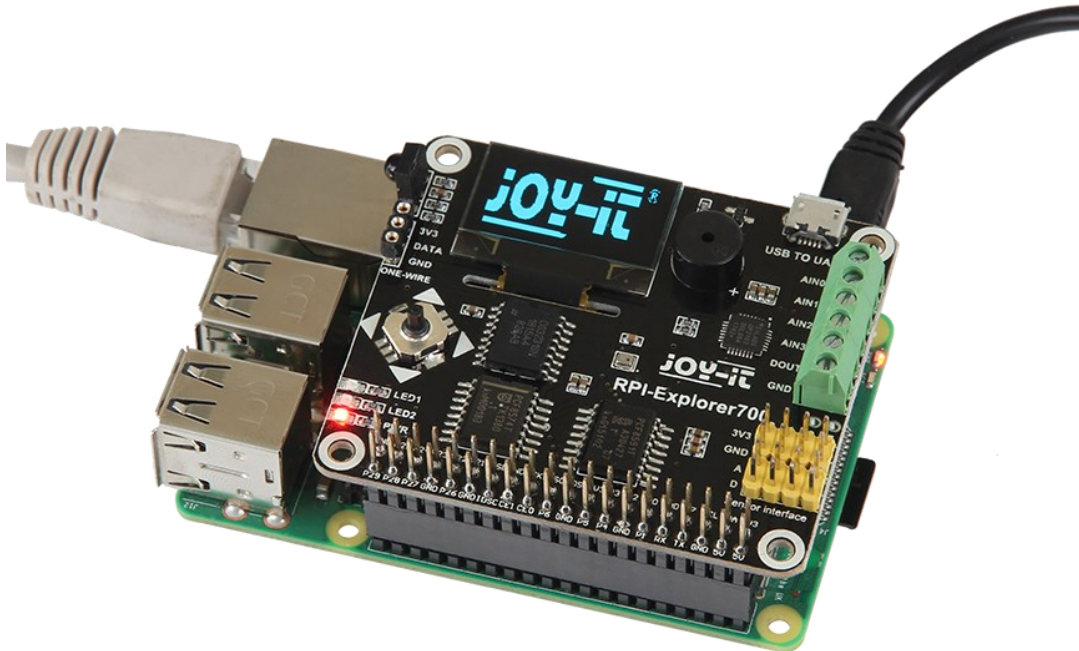


EXPLORER 700

Expansion board



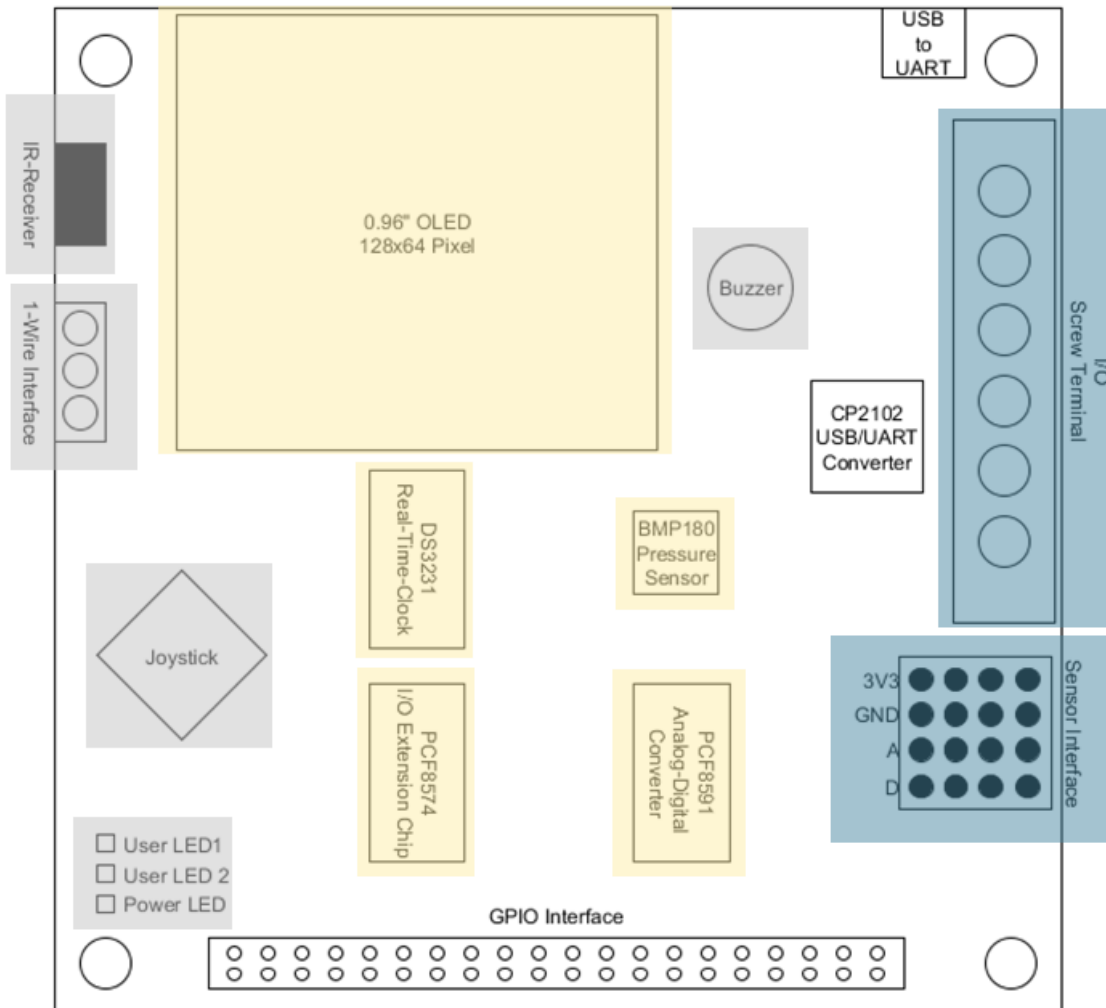
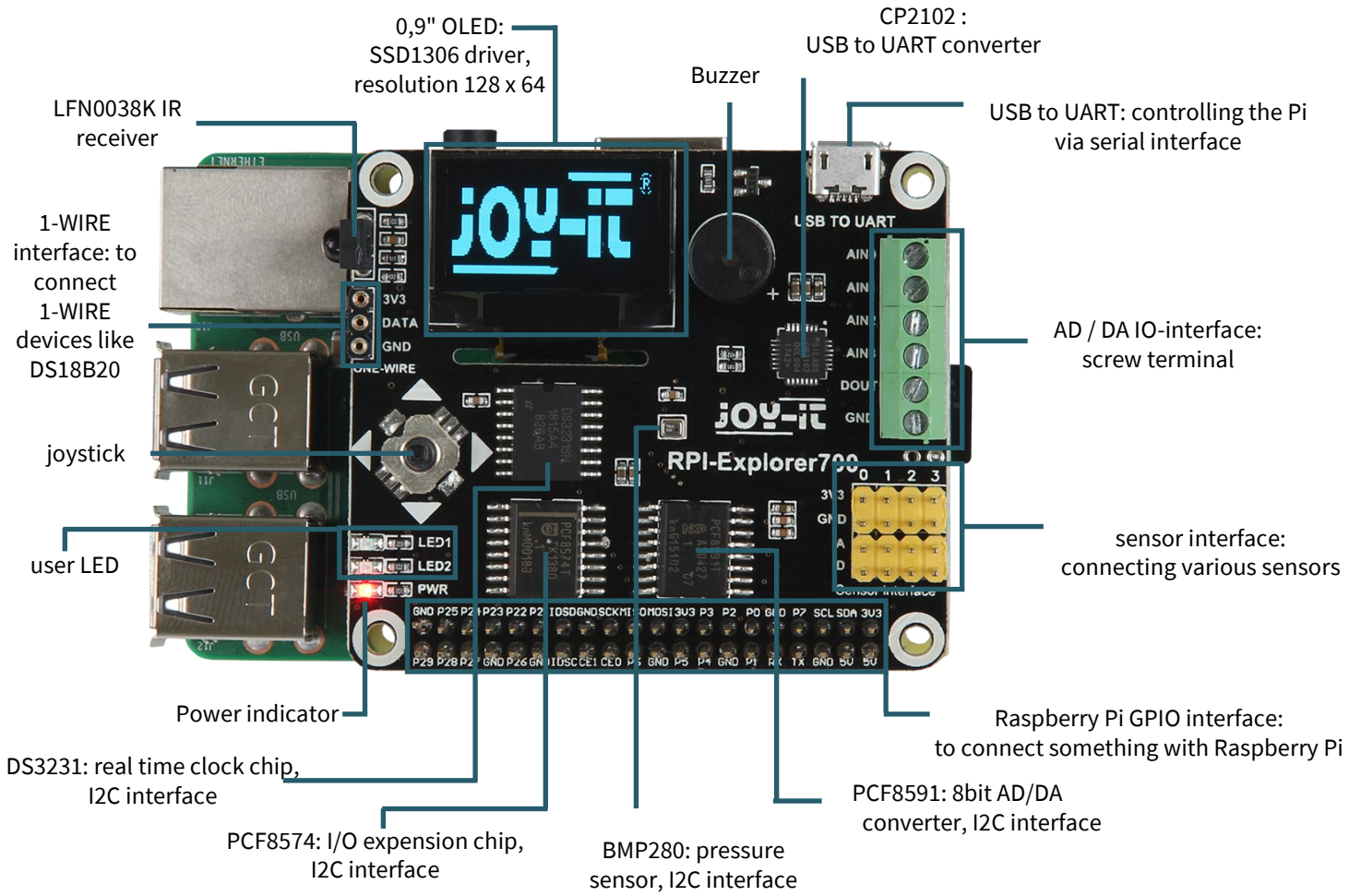
1. GENERAL INFORMATION

Dear customer,

Thank you for purchasing our product. In the following, we will show you which things should be noted during the use.

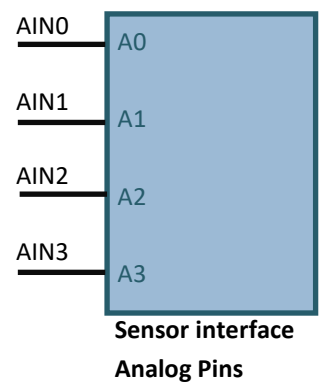
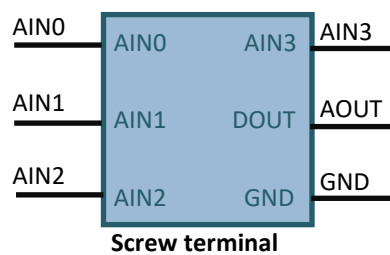
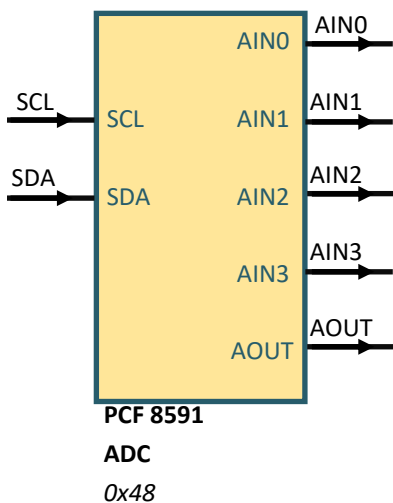
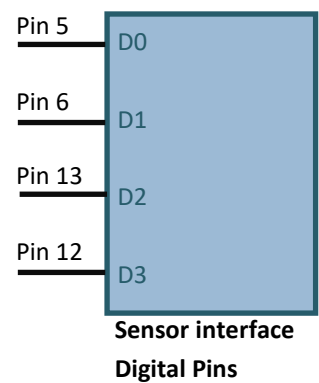
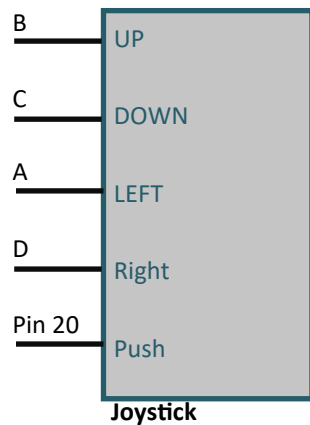
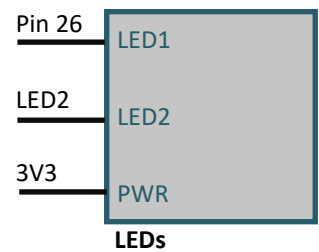
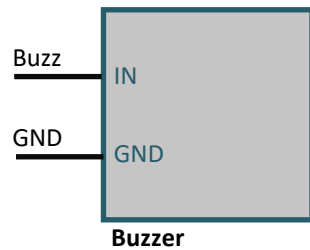
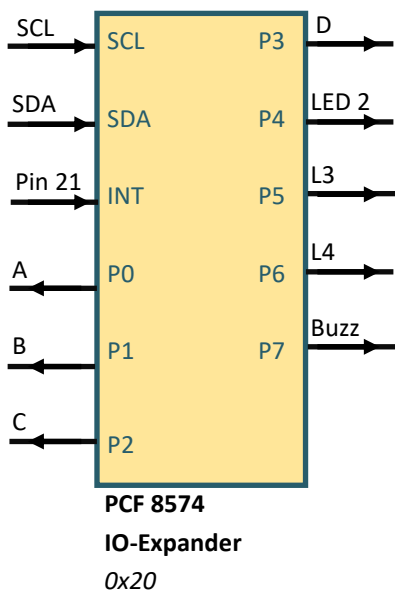
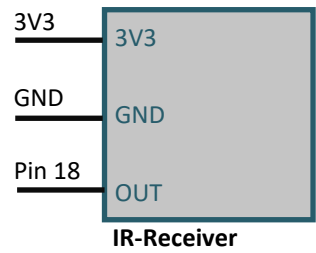
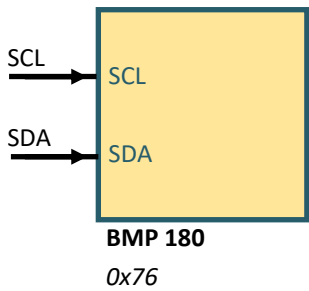
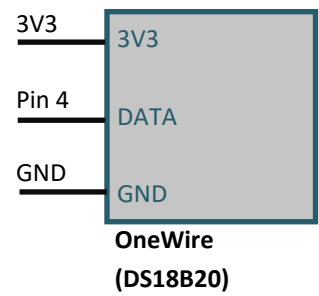
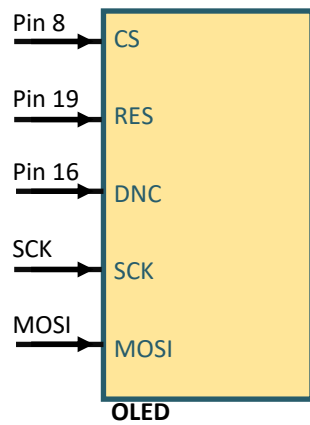
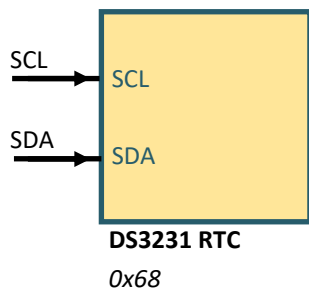
Should you encounter any unexpected problems, do not hesitate to contact us.

2. CONNECTIONS



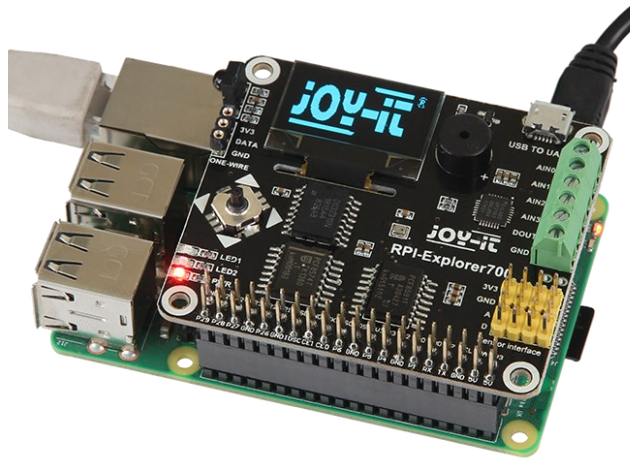
Schematic overview

All pins listed here refer to the GPIO/BCM pins of the Raspberry Pi.



3. CONNECTING THE EXPANSIONBOARD

The Explorer 700 expansion board supports the Raspberry Pi A+, B+, 2, 3 Model B and 4. Before you can perform the code examples, you should ensure that the Explorer 700 is connected to your Raspberry Pi and it is provided with power.



4. AVAILABLE DOWNLOAD

We provide you with a pre installed image, which saves you the trouble of installing and setting up libraries. You can download the image [here](#).

To install the image on your microSD card, you can use [Win32DiskImager](#). Here you select the microSD card to write to and the downloaded image. Now click on **Write** to write on this SD card.

If you are not using our image, you can download our sample codes [here](#), which will help you to access these samples. But then please note the following two chapters.

These chapters can be skipped when using our image.

5. INSTALLATION OF THE LIBRARIES

To use API examples which we offer, certain libraries are required which should be installed manually, if you are not using our image. Python libraries for Raspbian (includes Raspberry Pi, GPIO and spidev installation packages. See: <https://pypi.python.org/pypi/RPi.GPIO> <https://pypi.python.org/pypi/spidev>) You get them with the command **wget**.

Please note that the Raspberry Pi must be connected with the internet if you want to use this command to install the library. Beforehand, you can use the following command to update your software to the latest.

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

Enter now the following command to install the Python-dev package.

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

Enter the following commands to install the smbus and smbus2 library (I2C interfaces functions):

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

Enter the following command to install the serial library which includes the UART interface functions.

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

For our code examples, the libraries [RPi.GPIO](#) and [spidev](#) are mandatory. You can install the RPi.GPIO library with the following command:

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

The spidev library with this command:

```
sudo pip3 install spidev
```

To use the OLED, you must install this library :

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

Furthermore, for the BMP2802 you need libraries from Adafruit [CircuitPython BMP280](#) and [CircuitPython Bundle](#), which are both published under the MIT license ([BMP280](#) and [Bundle](#)) . You install them as follows:

```
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
```

6. CONFIGURATION OF THE INTERFACES

Before running the API codes we provide, you should activate the appropriate drivers for the interfaces.

Activation of the I2C-function

Enter the following command to configure your Raspberry Pi board.

```
sudo raspi-config
```

Choose in *Interfacing Options* → *I2C* → *Yes* to start the core driver of I2C. Afterwards, you should adjust the configuration file. To open the configuration file, enter the following:

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

Add the following two lines at the end of the configuration file:

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

Press the key combination **CTRL + X** to leave the settings and press **Y** to save them.

Activation of the serial functions

The serial port of the Raspberry Pi is set on the debugging via console mode by default. If you want to use the serial port as a normal IO, you have to adjust the configuration of the Raspberry Pi. If the debugging via console function is deactivated, you can not access the Raspberry Pi board via the serial port. If you want to control the Raspberry, you must activate the debugging of the serial port via the console function again.

```
sudo raspi-config
```

Choose in **Interfacing Options** → **Serial**. Through selecting the option **No** the debugging via the console function will be deactivated. Afterwards, the serial port can be used for the serial communication. With selecting the option **Yes** the debugging via console function can be activated again.

You also need to activate the 1-Wire interface. To do this, select **Interfacing Options** → **1-Wire** and activate the function with **Yes**.

You should now restart the Raspberry Pi.

Note: The serial port of the Raspberry Pi 3B is not available because pin14 and 15 are connected with its own Bluetooth.

To still use the serial functions, you have to enable the SPI-function. For that, you open the terminal and enter the following command:

```
sudo raspi-config
```

Now you only have to choose in **Interfacing Options** → **SPI** → **Yes** to enable the SPI functions.

7. CHANGING THE LED STATUS

The explorer board has LEDs which can be controlled by our code examples. LED1 can be used directly. To use this LED open the terminal and enter the following commands:

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

LED1 should now start blinking. Press the keys **CTRL + C** to exit this program. There is another code for LED1 which lets it pulsate.

```
sudo python3 pwm.py
```

Press the keys **CTRL + C** to exit happens.

8. PRESS FUNCTION OF THE JOYSTICK

The joystick has next to the press function the common directions of movement. Our code example shows when the press function is used.

Open the Linux console and enter the following commands:

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

By pressing the Joystick, the console will show the following:

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

Press the keys **CTRL + C** to exit this program.

9. I/O-EXPANDER - PCF8574

The I/O Expander PCF8574 also controls the LED2, the buzzer and the joystick.

Open the console and enter the following commands:

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

With this program, LED2 is blinking.
Press the keys **CTRL + C** to exit the program.

```
sudo python3 joystick.py
```

This program combines the functions of the Explorerboard, which are controlled by the PCF8574. When the joystick is pressed in one direction, the buzzer sounds, LED2 lights up and the console outputs the direction in which the joystick is pressed.

```
up  
left  
down  
right
```

Press the keys **CTRL + C** to exit this program.

10. BAROMETER - BMP280

This sample code is based on two Adafruit libraries (**chapter 5**). This sensor can measure temperature and air pressure and can therefore also calculate the altitude.

Note that the temperature can be influenced by the heat of the Raspberry Pi.

In order to calculate a correct altitude, change the sea air pressure to the air pressure at the current location in the sample code.

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

Press the keys **CTRL + C** to exit this program.

11. REAL - TIME CLOCK - DS3231

You can terminate the connection to the RTC with the following command:

```
sudo i2cdetect -y 1
```

Now you should see an address with which you can control the RTC. With us it is 0x68. **If this address is different, you must change it in the code example.**

You can embed this RTC into your system as a real-time clock or use it only in a code. Please note that a real-time clock is only fully functional if a battery is installed in the corresponding battery holder.

Embedding in the system

First, you edit the *modules* file. You do this with the following command:

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

Add this line to the end of the file:

```
rtc-ds3231
```

You can save the file with **CTRL + O** and exit the editor with **CTRL + X**. Now you have to edit the *config.txt*. Open it with the following command:

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

There you add this line to the end of the file:

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

You can save the file again with **CTRL + O** and exit the editor **with CTRL + X**. Now, you have to install the driver of the RTC. The driver of DS1307 (another RTC) is also used by DS3231.

```
sudo modprobe rtc-ds1307
```

Now edit the following file to set the system time after a start to the time of the RTC:

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

Above the command *exit 0* add the following three lines:

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

Close the file again with **CTRL + O** and **CTRL + X** and restart the Raspberry Pi. Now the time of the system after each restart is set by the RTC.

Note that a CR1220 battery must be inserted in the battery holder so that the real-time clock can maintain the correct time.

Afterwards, you can control the RTC as one of the system. Therefore, you can now use the following commands:

```
sudo hwclock -r
```

With this command the time of the RTC can be output.

```
sudo hwclock -w
```

This command adjusts the RTC to the time of the system.

Use in code

If you want to use your RTC in the code, you must uninstall the drivers of the RTC or do not embed the RTC in the system, because the driver of the RTC blocks it for further processes. You can uninstall the driver with:

```
sudo rmdir rtc-ds1307
```

You only need to execute this command if you have embedded the RTC in the system. You should then also remove the added lines in *rc.local* to avoid incorrect changes in the system.

In our code example we use the library [RTC SDL DS3231](#) published by [SwitchDoc Labs](#).

To execute the code, you must go first into subfolder . You do this with the following command:

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

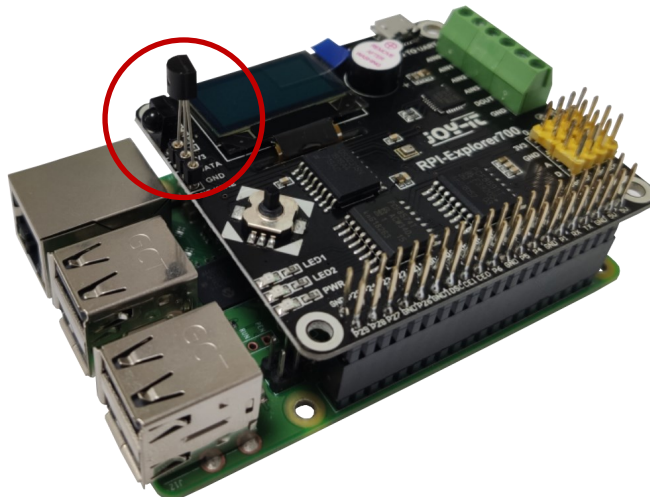
There you can execute our code, which maps the RTC to system time and sets the time and date every second. You can execute this code with the following command:

```
sudo python3 ds3231.py
```

Press the keys **CTRL + C** to exit this program.

12. TEMPERATURE SENSOR - DS18B20

The temperature sensor which is included in the scope of delivery, must be connected separately and is not on the board. Thereby, you should note that the straight side of the sensor is showing to the display and the rounded side is pointing towards the Raspberry Pi's USB and ethernet ports.



For this example, open the console and enter the following command:

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

The console will display the temperature in Celsius.

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

Press the keys **CTRL + C** to exit this program.

13. IRM - INFRARED REFLECTING SENSOR

On the board is an infrared receiver which can convert infrared into electric signals. This signals include the infrared values which are issued by the console via this code example.

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

Therefore, an infrared remote controll is requiered, which is not included in the scope of delievery.

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

Press the keys **CTRL + C** to exit this program.

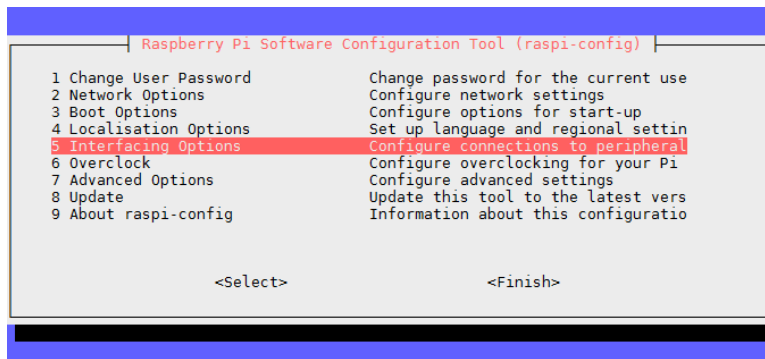
14. UART - TRANSMISSON OF SERIAL DATA

The serial interface of the Raspberry Pi is by default deactivated. For this program, you must enable this function so that the Raspberry Pi is not communicating via the serial port. Instead, other methods must be used to communicate with the Raspberry Pi. To use this code example, you first have to change the serial function in the configuration of the Raspberry Pi.

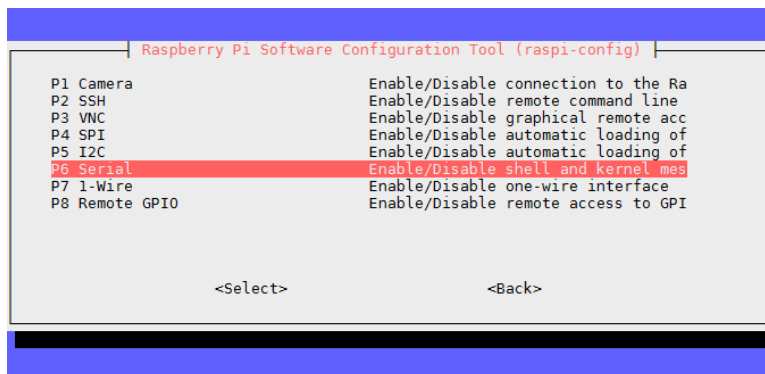
For that, enter the following command into the console:

```
sudo raspi-config
```

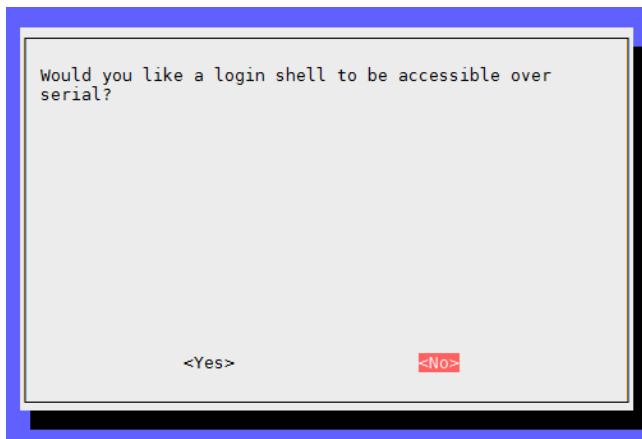
Now navigate with the arrowkey to the option *5 Interfacing Options*.



After that choose **Punkt P6 Serial**.



Answer the question *Would you like a login shell to be accessible over serial?* with **No**.



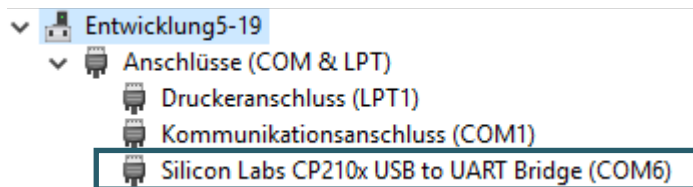
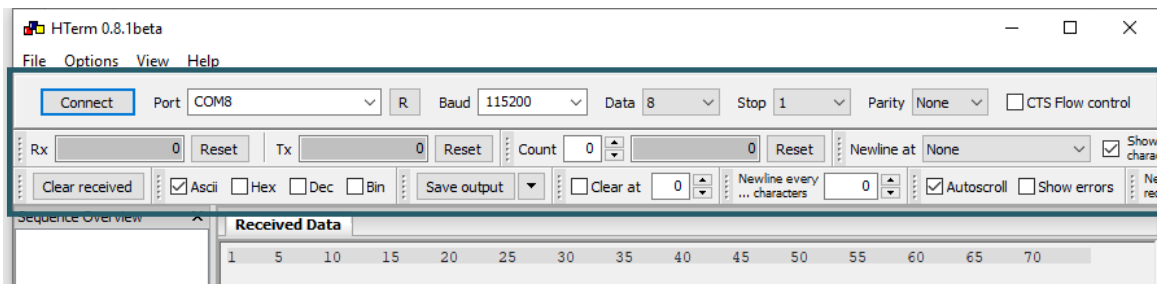
Answer the question *Would you like the serial port hardware to be enabled?* with **Yes**.



The USB to UART interface of your Raspberry Pi allows you to connect the Pi with your computer and to let them communicate without a connection with the internet.

In this example, we use HTerm. Download this program [here](#) and install it.

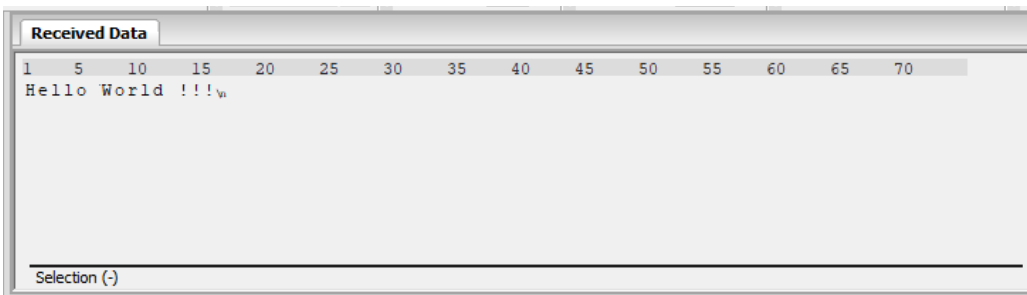
First of all, open HTerm and assume the settings like shown in the picture. Therefore, it is to note that you set the right baudrate (115200) and the right port. You should also note that your port could be different which you should ensure in the device manager.



Now you can click on Connect to establish a connection. As the next step, you have to start the code example on your Raspberry Pi. For that enter the following:

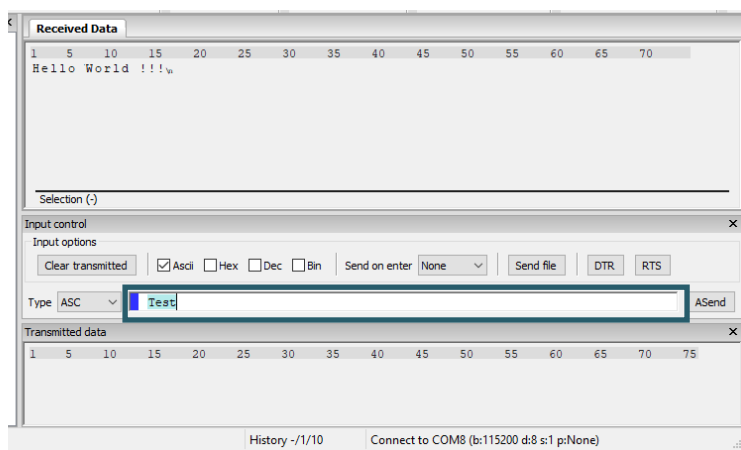
```
cd ~/Desktop/Explorer-700
sudo python3 uart.py
```

In HTerm you will see the following:

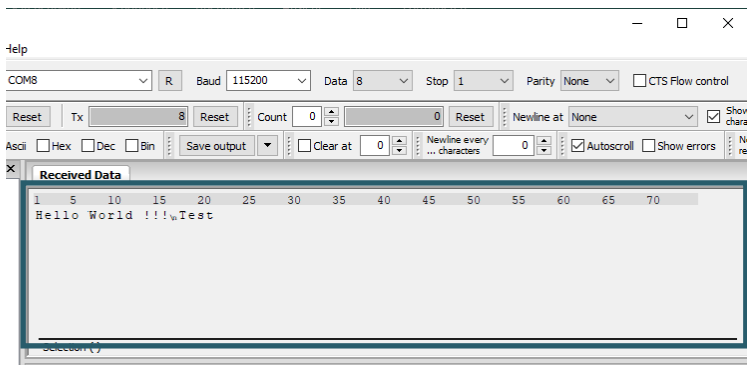


Now you can send texts to your Raspberry Pi. The code examples allows that the Raspberry Pi will send all sent texts from your computer back to your computer.

You can enter these texts in the text field and send them with **ASend**.



After that your computer will receive the following:

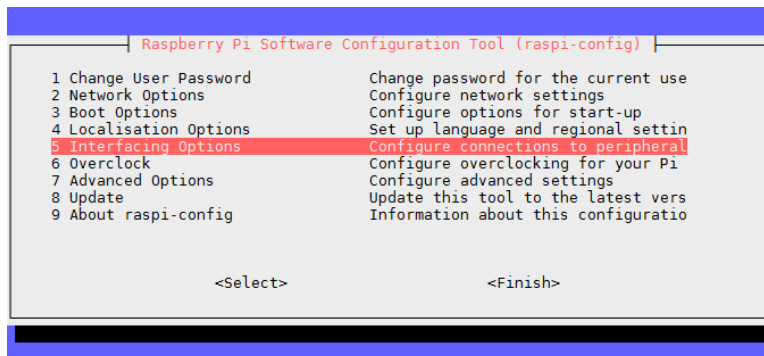


You can open the terminal console via the UART connection instead of an external display or via the internet. Therefore, you can use PuTTY, MobaXterm or others for the connection. For that, you have to change your settings again.

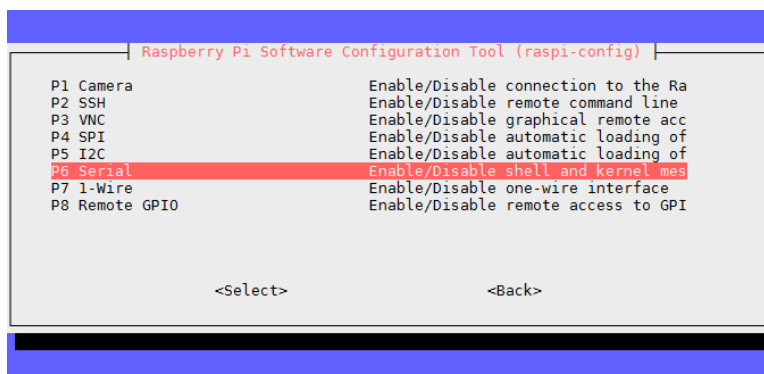
First, enter the following command into the console:

```
sudo raspi-config
```

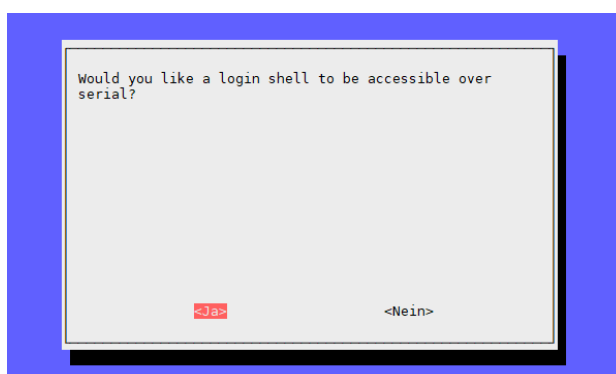
Navigate now with the arrow key to the option **5 Interfacing Options**.



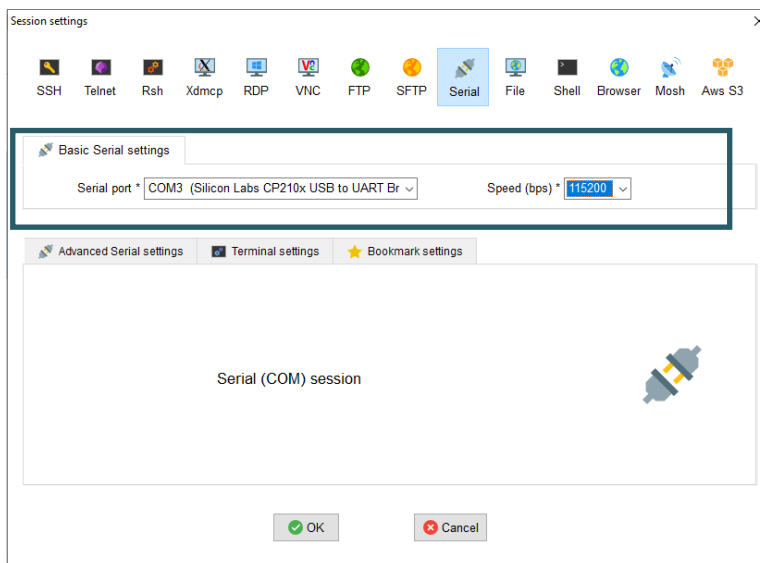
Then choose **point P6 Serial**.



Answer the question **Would you like a login shell to be accessible over serial?** with Yes.



After a reboot, you can connect your Raspberry Pi with your computer. Now MobaXterm, which you can download [here](#) and establish a connection with the following setting:



Therefore, you should note that you have to choose the connection type **Serial** and the right port (viewable in device manager) as well as to set the **baudrate to 115200**.

Now, the console will be opened and you must log in on your Raspberry Pi. If you use our image, the user name is *pi* and the password is *raspberry*.

Attention, the keystrokes will not be shown if you enter your password.

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

After you logged in successfully, the console will open and you can use the terminal of the Raspberry Pi the same way as via SSH only without an internet connection.

```
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:~$ █
```

15. OLED - DISPLAY

To execute these codes, make sure the **I2C** and **SPI** are activated. In **chapter 6** you can read how to activate both. The PIL library must also be installed. How to install it can be read in **chapter 5**.

Our code examples can generate different displays at the OLED. These differentiate in significant aspects from each other.

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

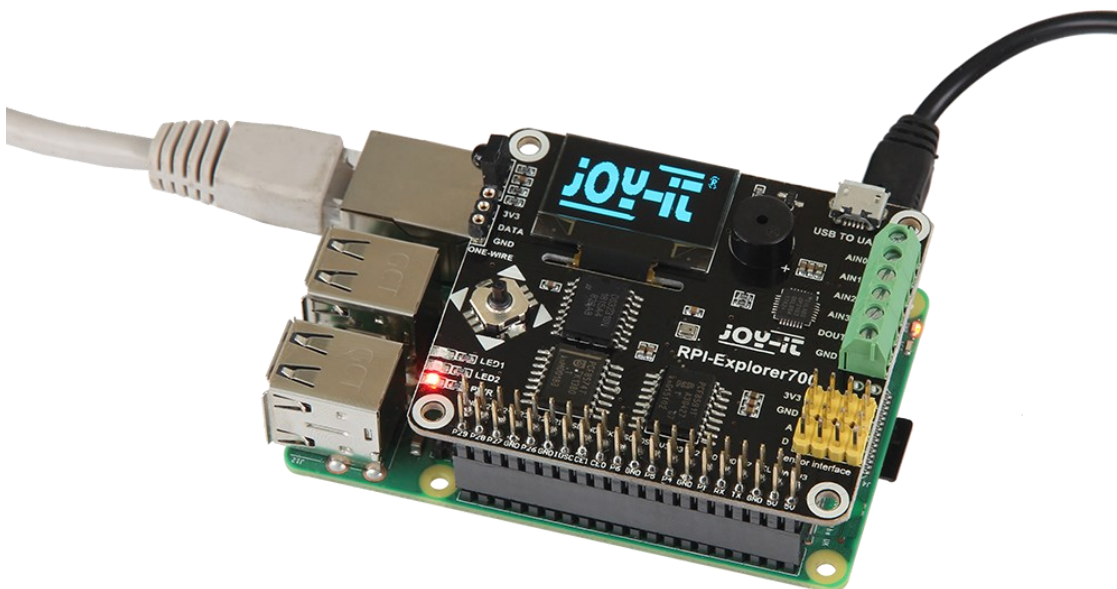
This script shows a text on the display which can be changed in the source code.

```
sudo python3 image.py
```

This script displays a picture on the display. This image can also be changed in the code.

```
sudo python3 animate.py
```

The last script generates an animation on your OLED.



16. OTHER INFORMATION

Our Information and Take-back Obligations according to the Electrical and Electronic Equipment Act (ElektroG)

Symbol on Electrical and Electronic Products:

This crossed-out bin means that electrical and electronic products do not belong into the household waste. You must hand over your old appliance to a registration place. Before you can hand over the old appliance, you must remove used batteries and replacement batteries which are not enclosed by the device.



Return Options:

As the end user, you can hand over your old appliance (which has essentially the same functions as the new one bought with us) free of charge for disposal with the purchase of a new device.

Small devices, which do not have outer dimensions bigger than 25 cm can be handed in for disposal independently of the purchase of a new product in normal household quantities.

1. Possibility of return at our company location during our opening hours

Simac Electronics Handel GmbH, Pascalstr. 8, D-47506 Neukirchen-Vluyn

2. Possibility of return nearby

We will send you a parcel stamp with which you can send us your old appliance free of charge. For this possibility, please contact us via e-mail at pickup@joy-it.net or via telephone, then please dial 02845 93 60 -23

Information about Package:

Please package your old appliance safe for transport. Should you not have suitable packaging material or you do not want to use your own material, you can contact us and we will send you an appropriate package.

17. SUPPORT

If any questions remained open or problems may arise after your purchase, we are available by e-mail, telephone and ticket support system to answer these.

E-Mail: service@joy-it.net

Ticket-system: <http://support.joy-it.net>

Telephone: +49 (0)2845 98469 – 66 (10 - 17 o'clock)

For further information visit our website:

www.joy-it.net