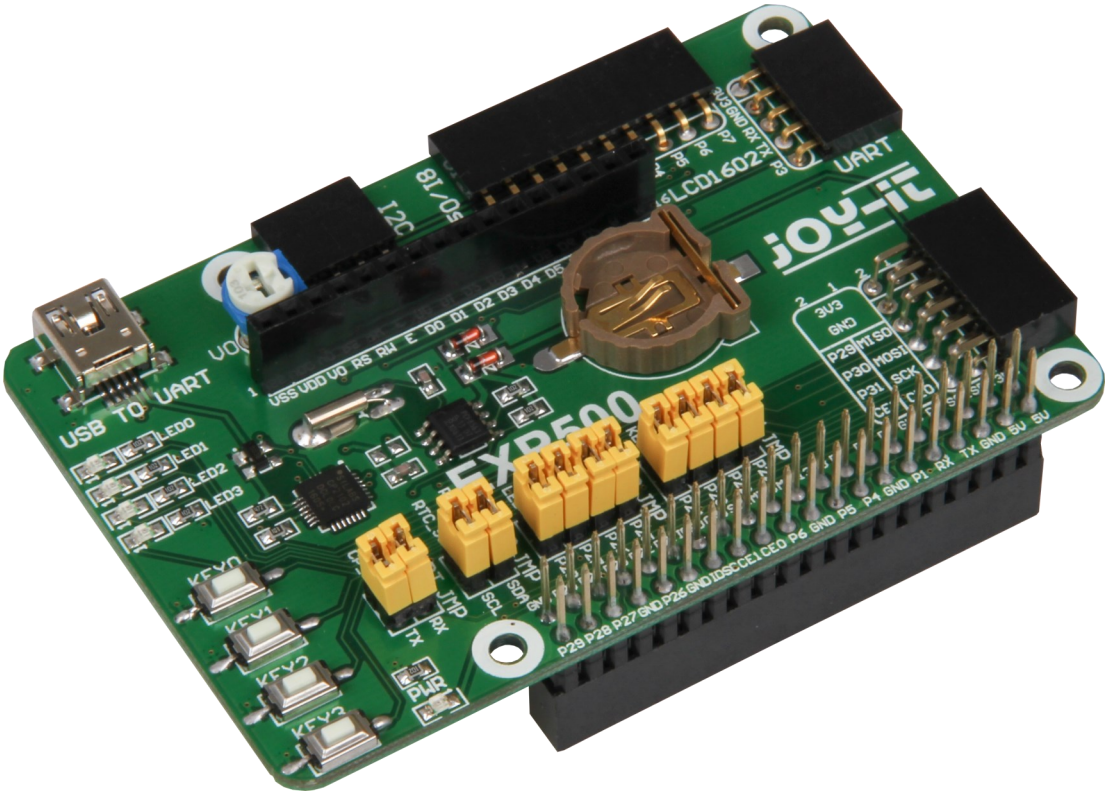
t

EXPLORER 500

Expansionboard

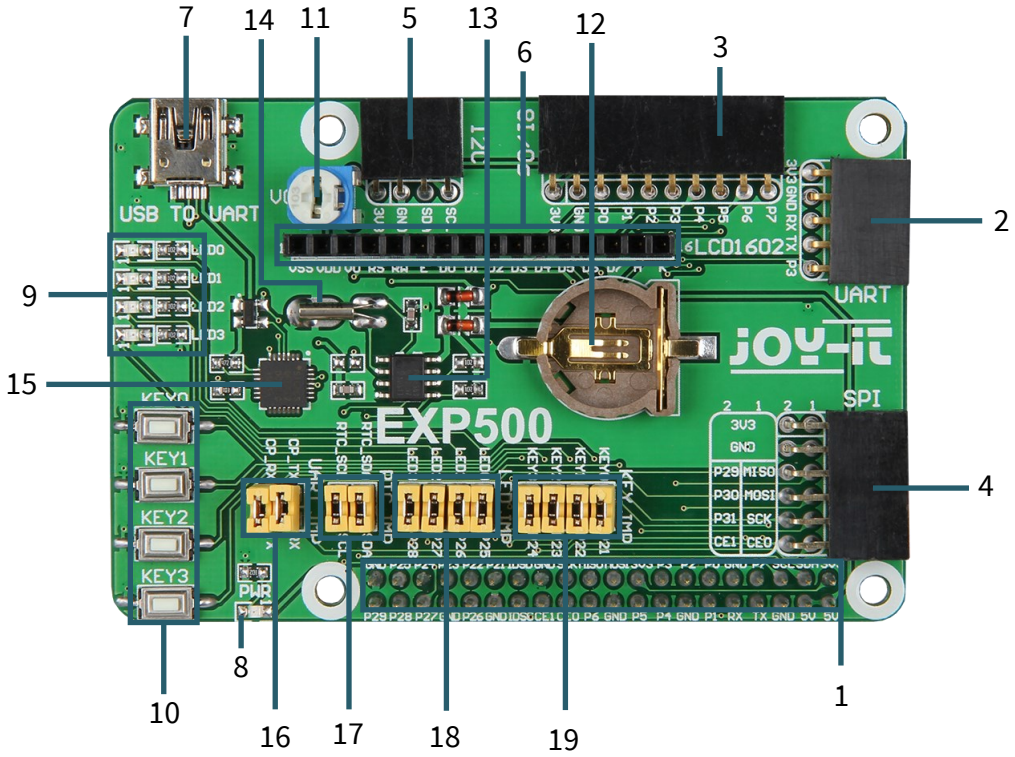


1. GENERAL INFORMATION

Dear customer,
thank you for choosing our product. In the following, we will show what is to observe during the commissioning and the usage.

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

2. CONNECTIONS



1. Pin-Header for the directly pinning on the Raspberry Pi B+, 2B, 3B or 3B+
2. UART-interface
3. 8x I/O interface
4. SPI-interface
5. I2C-interface
6. LCD-interface for HD44780 industrial-standard LCD modules
7. USB TO UART interface
8. Power LED
9. 4x programmable LED
10. 4x programmable buttons
11. Potentiometer → control of contrast for LCD displays
12. RTC battery holder for CR1220 batteries
13. PCF8563: onboard RTC chip
14. 32.768 crystal: RTC crystal
15. CP2102: onboard USB TO UART chip for debugging
16. CP2102: jumper for on / off
17. RTC: Jumper for on / off
18. User LEDs jumper for on / off (see point 9)
19. User keys jumper for on / off (see point 10)

3. IMAGE INSTALLATION

Download the newest prepared image from our website: Download

Format a microSD-card with the “SDFormatter.exe”.

Note: **The memory size of the microSD-card should be more than 4 GB. In this process a microSD-card reader is required which must be purchased separately.**

Start the file “Win32Disklmager.exe” and select the copied image onto your PC. Click now on the button “Write” to write the image onto the card.

4. INSTALLATION OF LIBRARIES FOR THE RASPBERRY PI

If you use the prepared image for the Explorer 500, you can skip this and the next chapter.

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
```

Enter the following command to install the python3-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 contains the UART-interface functions.

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

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 *STRG + C* 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 the console mode by default. If you want to use the serial port as a normal IO, you must adjust the configuration of the Raspberry Pi. If the debugging via the 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 should now restart the Raspberry Pi so that the settings come into force.

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

To still use the serial functions, you must activate the SPI-function. For that, you start the SPI-function and you enter meanwhile the following command:

```
sudo raspi-config
```

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

6. CODE EXAMPLES

Important: **Should you use a new Raspbian and not one of our prefabricated images, you should note chapter 4 and 5.**

Prior to the application of the Explorer 500 program, you should install the libraries of bcm2835, wiringPi and python on the Raspberry Pi that you can add further APIs. Additionally, you should adjust the settings to start the core driver of I2C, SPI and UART automatically after the installation of the libraries.
If you are not using our image, you can download the codes examples here.

LED

We provide for the different modules of the Explorerboard 500 example codes. Also for the four LEDs, which are located on the board. First, use the following command to go to the subfolder of the LEDs:

```
cd ~/Desktop/EXP500/LED
```

There you can find three codes for the LEDs. One is *led.py*, which lets the LED0 flash. You can run this program with the the following command:

```
sudo python3 led.py
```

The example code *pwm.py* makes LED1 pulsate and you can execute it with the following command:

```
sudo python3 pwm.py
```

The example code *blink.py* addresses all LEDs and lets them flash one after the other. You can trigger this with the following command:

```
sudo python3 blink.py
```

KEY

The Explorerboard 500 has four buttons which can be controlled. To do this, first use the following command to leave the subfolder LED:

```
cd ~/Desktop/EXP500/
```

Now, you can use the following command to control the buttons that are marked with KEY. The console displays which button is pressed.

```
sudo python3 key.py
```

LCD 16x2 (separately available)

On the Explorerboard there is an LCD interface with which a LCD can be easily connected. Our *COM-LCD16x2* can easily be plugged onto the board and is immediately ready for use with our sample code. You can execute it with the following commands:

```
cd ~/Desktop/EXP500/
```

```
sudo python3 lcd16x2.py
```

PCF8563- Real Time Clock

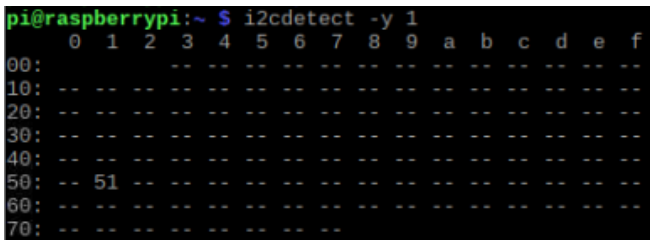
Check if these jumpers of the Explorer 500 are plugged in:

RTC_SDA with SDA
RTC_SCL with SCL

This is necessary to control the RTC.
You can check your connection with the following command:

```
sudo i2cdetect -y 1
```

Now you should see an address with which you can control the RTC in case everything is wired correctly.



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 the following command:

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

Add this line to the end of the file:

```
rtc-pcf8563
```

You can save the file with **CTRL + O** and exit the editor with **CTRL + X**. Now you must 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,pcf8563
```

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. Afterwards, you have to restart Raspberry Pi to make your changes take effect.

```
sudo modprobe rtc-pcf8563
```

Edit now 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 pcf8563 0x51 > /sys/class/i2c-adapter/i2c-1/new_device

hwclock --hctosys

date
```

Close the file again with **CTRL + O** and **CTRL + X** and restart the Raspberry Pi. Now the time of the system is after each restart 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 rmmod rtc-pcf8563
```

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_PCF8563 published by SwitchDoc Labs.
To execute the code, you must go first into subfolder . You do this with the following command:

```
cd ~/Desktop/EXP500/RTC_SDL_PCF8563
```

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 pcf8563.py
```

You can also run the sample code from SwitchDoc Labs,
which displays the time of the system and the RTC in a 10 second cycle. You start it with :

```
sudo python3 testSDL_PCF8563.py
```

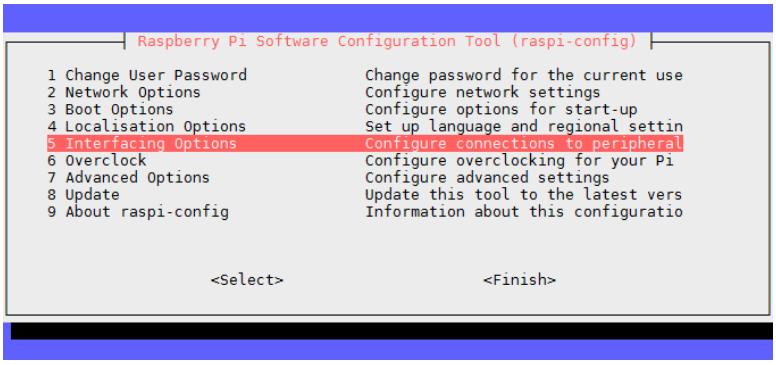
UART - Transmission of serial data

The serial interface of the Raspberry Pi is by default deactiveted. 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 have to change first the serial function in the configuration of the Raspberry Pi.

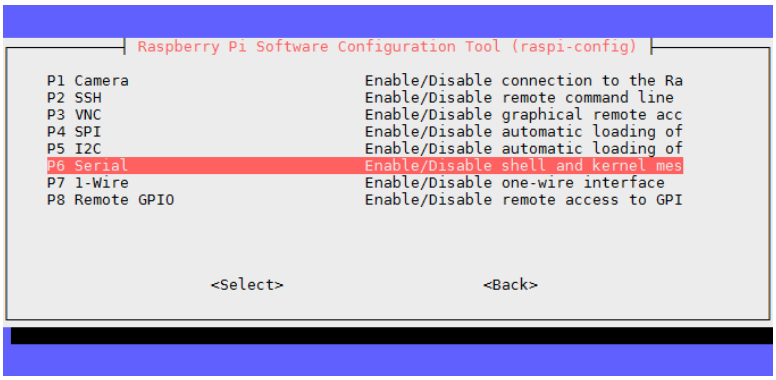
For that, enter the following command into the console:

```
sudo raspi-config
```

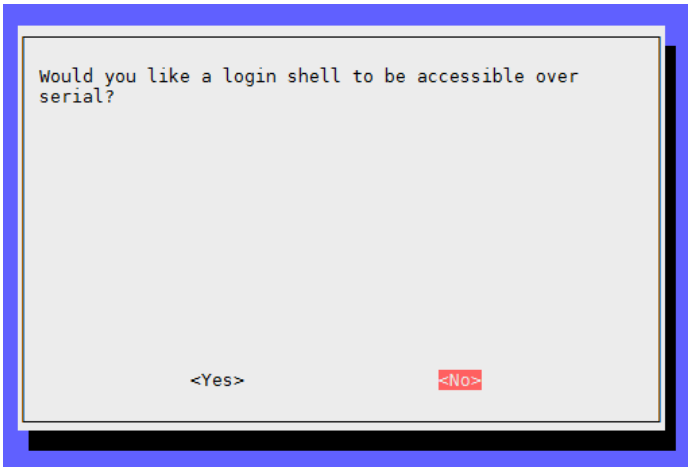
Now, navigate via the arrow keys into the option *5 Interfacing Options.*



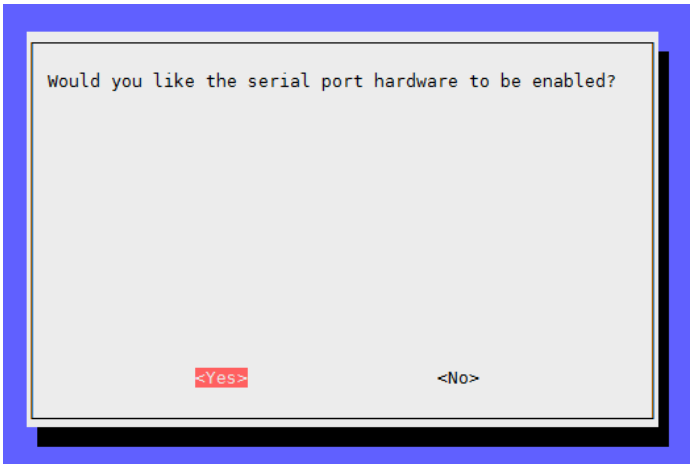
After that choose **Punkt P6 Serial**.



The question *Would you like a login shell to be accessible over serial?* should you answer with **No**.

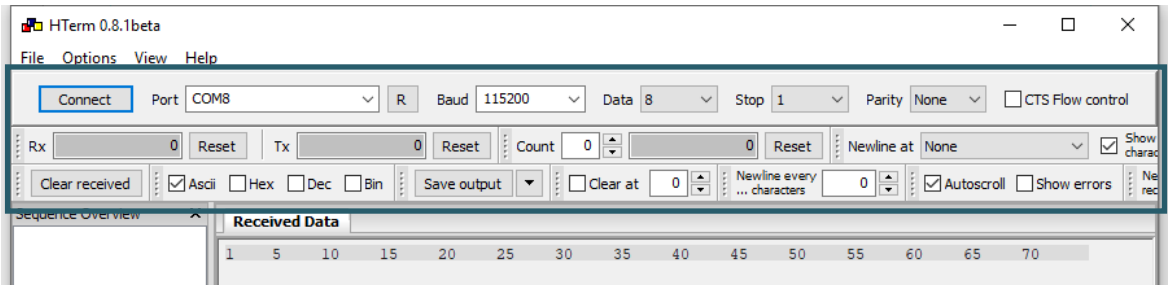


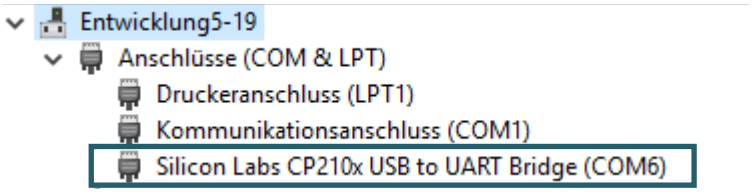
The question *Would you like the serial port hardware to be enabled?* answer with **Yes**.



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.

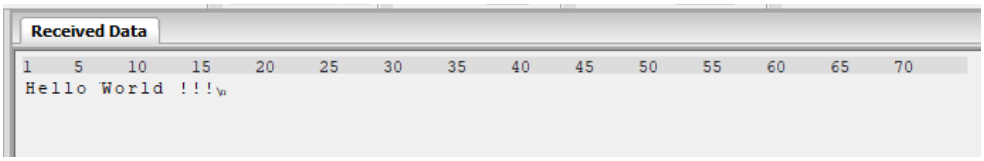




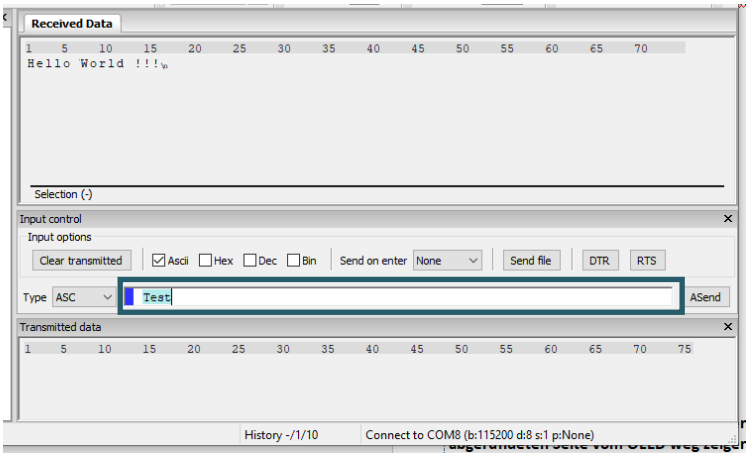
Now you can click on Connect to establish a connection. As the next step, you must start the code exampleon your Raspberry Pi. For that enter the following:

```
cd ~/Desktop/EXP500
sudo python3 uart.py
```

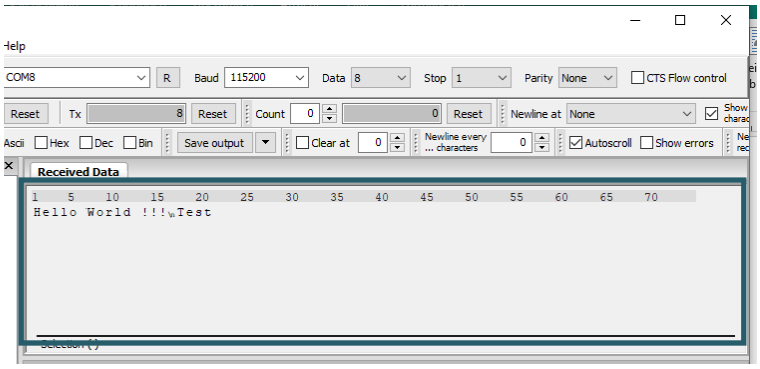
In HTerm you will see the following:



Now you can sen texts to your Raspberry Pi. The code examples allows that the Raspberry Pui will send all sended 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:

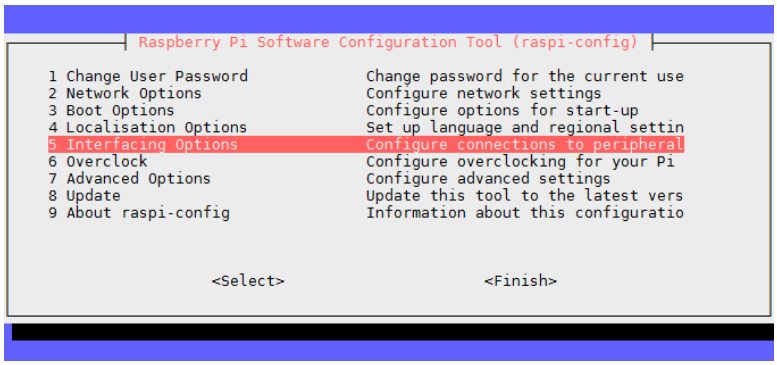


You can open the terminal console via the UART connection instead of an external display or via the internet. Therefore, you can use PuTTY, MobaX-term or others for the connection. For that, you have to change your settings again.

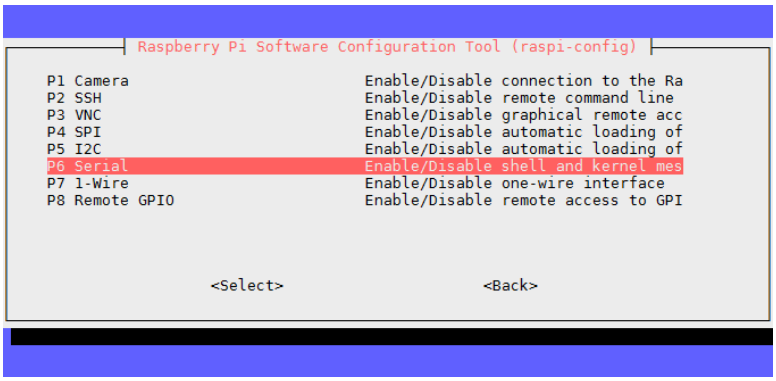
First, enter the following command into the console:

```
sudo raspi-config
```

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

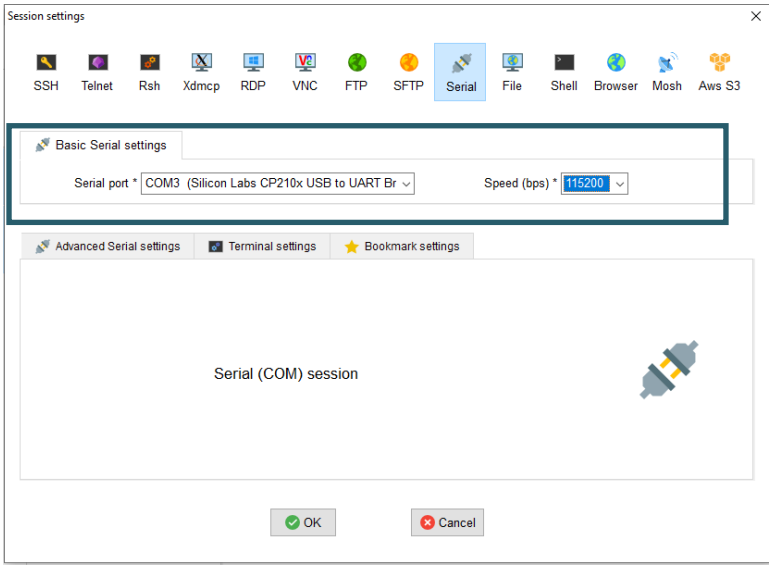


Then choose **point P6 Serial .**



The question ***Would you like a login shell to be accessible over serial?*** should you answer 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 a 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:~$
```

7. OTHER INFORMATION

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

Symbol on Electrial 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 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.

8. SUPPORT

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

Email: 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

Published: 06.05.2020