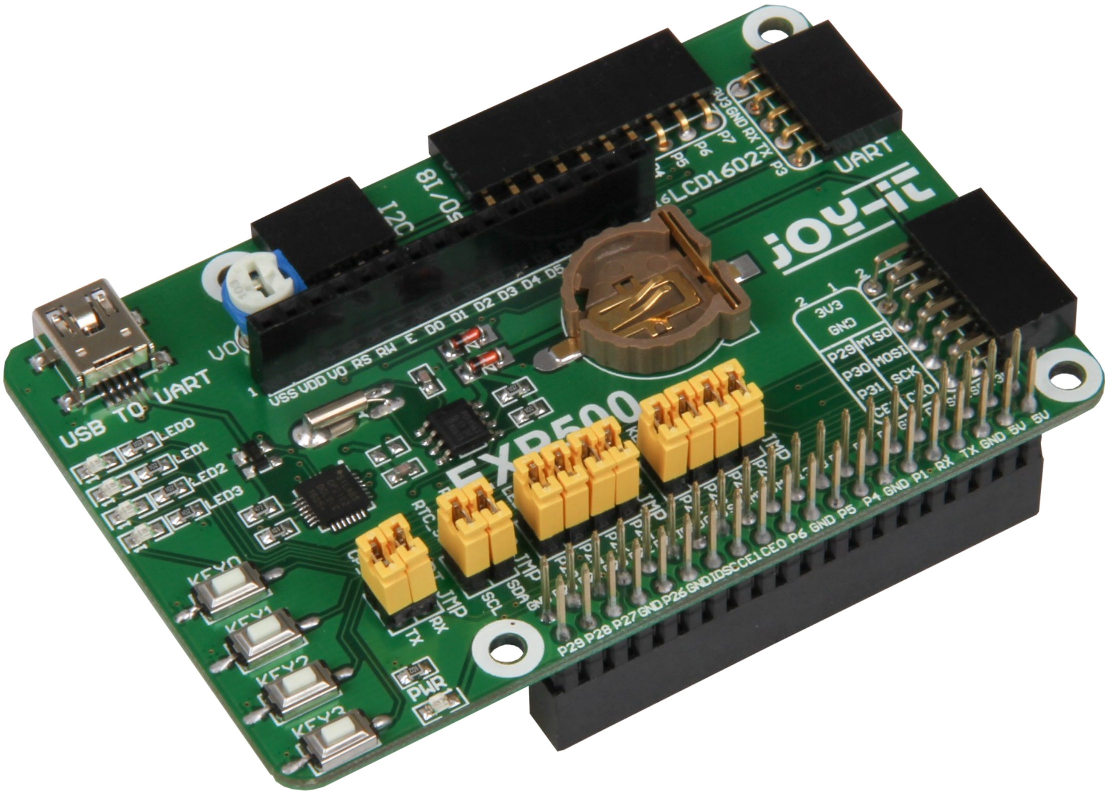
# EXPLORER 500

Expansionboard

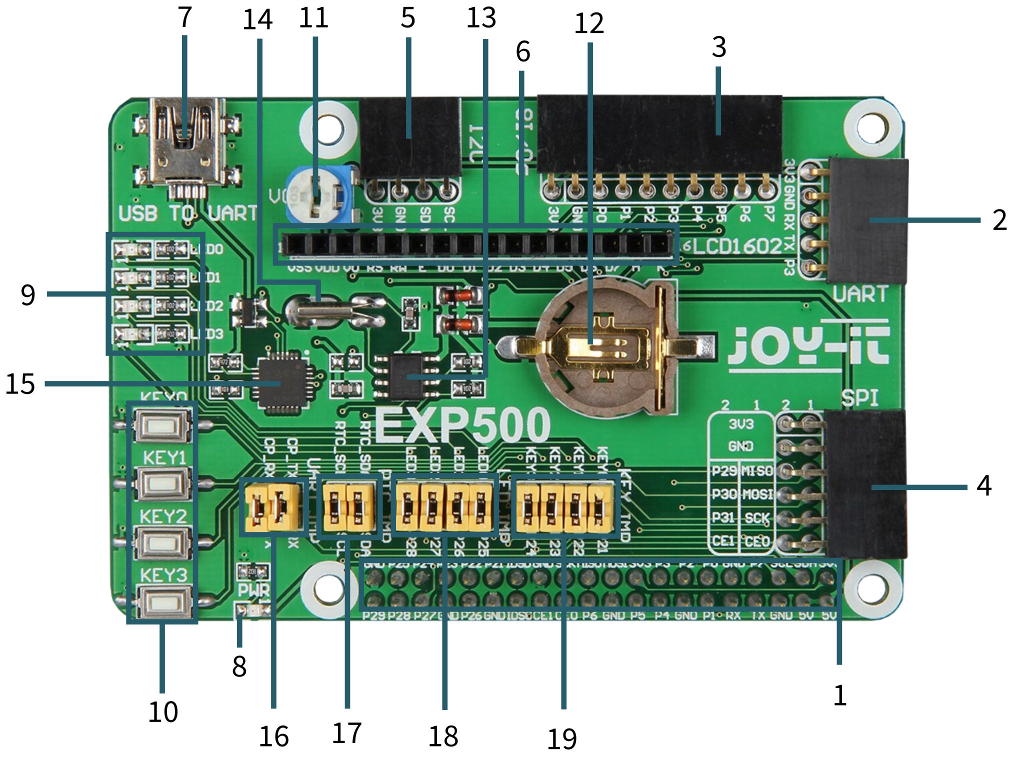


## 1. GENERAL INFORMATION

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
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)

## 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.**

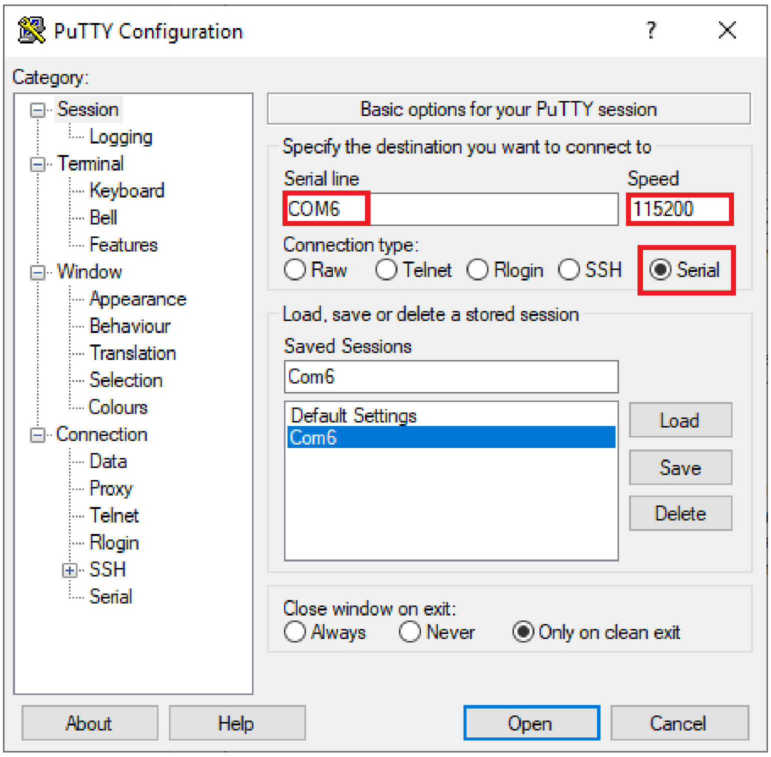
Start the file “Win32Disklmager.exe” and select the copied image onto your PC. Click now on the button “Write” to write the image onto the card.

## Install serial debug environment

Connect your PC with the UART TO USB interface from the Explorer 500 via miniUSB-cable.

**Note:** **Please keep in mind that the UART TO USB interface has a miniUSB-port though the raspberry Pi has a microUSB-interface.**

Install the driver “cp2102_driver”.
Start the program “PuTTY.exe” (Download) and configure the following parameter:



Specify the destination you want to connect to

Serial line: COM6

Speed: 115200

Connection type: ○ Raw ○ Telnet ○ Rlogin ○ SSH ● Serial

Load, save or delete a stored session

Saved Sessions: Com6

Default Settings
Com6

Close window on exit: ○ Always ○ Never ● Only on clean exit

Serial line: This is used to select the according serial port. You can verify the current used serial port via the device manager.

Speed: This is used to set the baud rate.

Connection type: This option should be set to serial. After that, confirm with “Open”.

After the system has started, you should enter the following information:

User name: pi
Password: raspberry

Afterwards, you can open the serial terminal to communicate with the Raspberry Pi.

**Note:In this document the program PuTTY is used to control the Raspberry Pi via the serial communication if nothing else is indicated. If a serial port for the Raspberry Pi terminal uses debug, this can not be used as a normal serial port anymore because the system is occupied by the debug function.**

## 4. CODE EXAMPLES

**Important:** **Should you use a new Raspbian and not one of our prefabricated images, you should start with point 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.

**Note:** **Every example should be in the according folder, for example if you want to use the LED-example in python, you enter the following command:**

```
cd Desktop/EXP500/LED/python/
sudo python led.py
```

This changes the folder:

```
cd Desktop/EXP500/LED/python
```

The sudo means that the command is run by the super user.

```
sudo python led.py
```

The sudo is not required if you are declared as the root user, The python led.py runs the file led.py with python.

The C-programms of bcm2835, wiringPi, sysfs and so on must be compiled in advance. In our examples it already is compiled. If you want to change something at the source code, you must remove the old file with...

```
make clean
```

To compile it again, run this command:

```
make
```

## LED - example

**bcm2835-program:**

1. Open the Linux console and run the following command:

```
cd Desktop/EXP500/LED/bcm2835
sudo ./led
```

2. All 4 LEDs will glow one after the other.
3. Press the keys *STRG + C* to close the example.

**wiringPi-program:**

1. Open the Linux console and run the following command:

```
cd Desktop/EXP500/LED/wiringPi
sudo ./led
```

2. All 4 LEDs will glow one after the other.
3. Press the keys *STRG + C* to close the example.

**Python-program:**

1. Open the Linux console and run the following command:

```
cd Desktop/EXP500/LED/python
sudo python blink.py
```

2. All 4 LEDs will glow one after the other.
3. Press the keys *STRG + C* to close the example.

## KEY - example

**bcm2835-program:**

1. Open the Linux console and run the following command:

```
cd Desktop/EXP500/KEY/bcm2835
sudo ./key
```

2. The terminal shows if a button is pressed.
3. Press the keys *STRG + C* to close the example.

**wiringPi-program:**

1. Open the Linux console and run the following command:

```
cd Desktop/EXP500/KEY/wiringPi
sudo ./key
```

2. The terminal shows if a button is pressed.
3. Press the keys *STRG + C* to close the example.

**Python-program:**

1. Open the Linux console and run the following command:

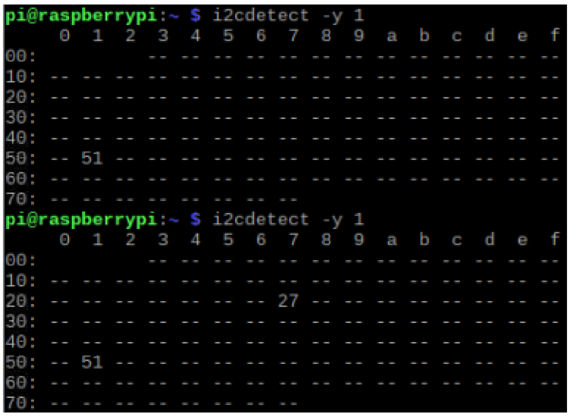
```
cd Desktop/EXP500/KEY/python
sudo python key.py
```

2. The terminal shows if a button is pressed.
3. Press the keys *STRG + C* to close the example.

## LCD 16x2 - example (separately acquirable)

To figure out which I2C-address the display has, enter the following command while the display is not connected and afterwards while the display is connected.

```
i2cdetect -y 1
```



```
pi@raspberrypi:~ $ i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- 51 -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
pi@raspberrypi:~ $ i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- 27 -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- 51 -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

In our example the display has address 27. Address 51 belongs to the PCF8563 (real time clock).

After you have connected the display and figured out the I2C-address, run the following commands:

```
cd Desktop/EXP500/LCD1602/python
sudo python lcd1602.py
```

With the following command, you can change what is shown on the display.

```
sudo nano /home/pi/Desktop/EXP500/LCD1602/python/lcd1602.py
```

## PCF8563 RTC - example

**Put the jumper of the Explorer 500:**

connect RTC_SDA with SDA
connect RTC_SCL with SCL

If the I2C list (see LCD 16x2 - example) shows two devices when a LCD is connected and one when it is not, the Raspberry Pi has identified the PCF8563. Thereupon, the following commands can be entered to test the real time clock.

**bcm2835-program:**

```
cd Desktop/EXP500/PCF8563/bcm2835
sudo ./pcf8563
```

**wiringPi-program:**

```
cd Desktop/EXP500/PCF8563/wiringPi
sudo ./pcf8563
```

**Python-program**

```
cd Desktop/EXP500/PCF8563/python
sudo python pcf8563.py
```

The time of the PCF8563 will now be shown in the console.

## UART - example

**Note:The serial port of the Raspberry Pi is normally used for the debug ging via the console. In order that you can use UART, you have to disable this function by entering the following command:**

```
sudo raspi-config
```

Go to **Interfacing-Options -> Serial** and switch off debugging via the console by choosing **No** and enable the serial hardware port by choosing **Yes.**

Connect the installed UART TO USB port via miniUSB-cable to a PC.
Open the console and run the following commands:

**wiringPi-program:**

```
cd Desktop/EXP500/UART/RPI3/wiringPi
sudo ./UART
```

**Python-program:**

```
cd Desktop/EXP500/UART/RPI3/python
sudo python uart.py
```

Start the serial monitor program, e.g. HTerm, which you can download here.

Search for the according COM-port and set the baud rate to 115200. The text which will be sent from the program, will be shown via a serial connection.

**Note: Should you use a prefabricated image, you can skip point 5.**

## 5. INSTALLATION OF LIBRARIES FOR RASPBERRY PI

**To use our API-examples, you must have libraries which should be installed manually.**

**bcm2836 - library**
**wiringPi - library**

### Install the wiringPi - library

If GIT is not already installed, you can install it with the following command.

```
sudo apt-get install git-core
```

Should problems occur, update your Raspberry Pi.

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

To get wiringPi with GIT, enter the following command.

```
git clone git://git.drogon.net/wiringPi
```

Afterwards run the following commands.

```
cd wiringPi
git pull origin
```

For the installation, the following commands must be run.

```
cd wiringPi
./build
```

## Install the C-library bcm2835

First download the **bcm2836 - library** .
To install the bcm2835, enter the following commands (apply the version number of your downloaded file in this command):

```
tar zxvf bcm2835-1.xx.tar.gz
cd bcm2835-1.xx
./configure
make
sudo make check
sudo make install
```

## Install the Python - library

Download the **RPi.GPIO** and the **spidev** library.
Unzip both files and navigate via the cd-command in the unzipped path.
After that use the following command in both folders.

```
sudo python setup.py install
```

Enter the following command to install the python-dev package.

```
sudo apt-get install python-dev
```

Enter the following command to install the smbus-library (I2C-interface functions).

```
sudo apt-get install python-smbus
```

Enter the following command to install the serial library which contains the URAT-interface functions.

```
sudo apt-get install python-serial
```

## 6. CONFIGURATION OF INTERFACES

**Activation of the I2C-function**
Enter the following command to configurate your Raspberry Pi board.

```
sudo raspi-config
```

Choose in “Interfaceing 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.

## 7. FURTHER INFORMATION

### Our information and redemption obligation according to the electro-law (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 office. Before you can hand over the old appliance, you must remove used batteries and accumulators which are not enclosed by the device.

**Return options :**
As the end user, you can hand over with the purchase of a new device your old appliance (which has essentially the same functions as the new one) free of charge for disposal. Small devices which do not have outer dimensions greater than 25 cm can be submitted independently of the purchase of a new product in normal household quantities.

**Possibility of restitution at our company location during our opening hours :**
Simac GmbH, Pascalstr. 8, D-47506 Neukirchen-Vluyn

**possibility of restitution nearby :**
We send you a parcel stamp with which you can send us your old appliance free of charge. For this possibility, you must contact us via e-mail at service@joy-it.net or via telephone.

**information about packaging:**
Please package your old appliance safe during transport. Should you not have a 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 remain open or problems arise after your purchase, we are available by e-mail, telephone and with a 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**

Published: 27.11.2019