

JOY-IT

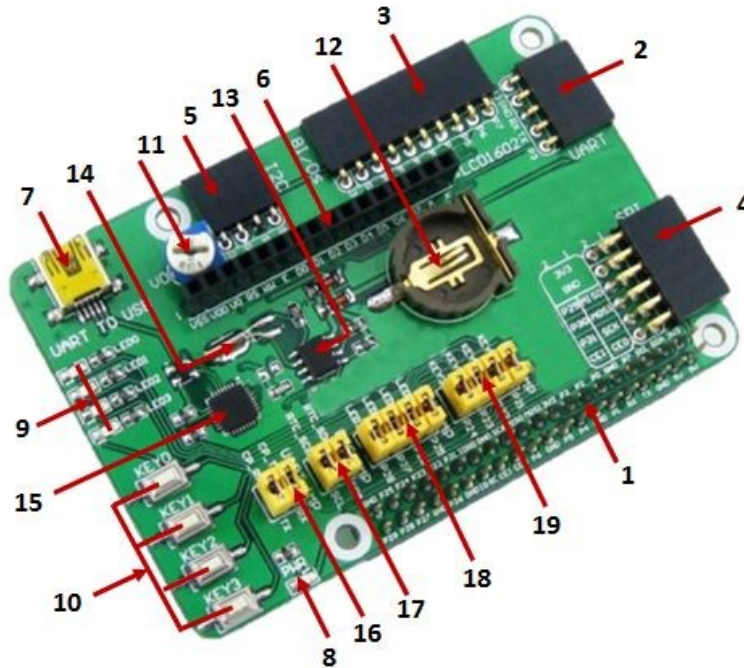
EXP500



Contents

1. Connections
2. Introduction
3. Basic operations
 - 3.1 System image file programming
 - 3.2 Serial debugging environment deploying
4. Installing the libraries required
 - 4.1 LED Demo
 - 4.2 Key Demo
 - 4.3 LCD1602 (Purchased Separately) Demo
 - 4.4 PCF8563 RTC Demo
 - 4.5 UART Demo
5. Libraries Installation for RPi
 - 5.1 Install WiringPi Library
 - 5.2 Install C Library bcm2835
 - 5.3 Install Python Library
 - 5.4 Configuring the interfaces
6. Support

1. Connection



1. Pin header to directly connect with the Raspberry B+, 2B or 3B
2. UART interface: connect UART modules easily like RS232, RS485, USB TO UART (TTL-cabel)
3. 8x I/O interface: connect controlled modules easily via I/O
4. SPI interface for SPI selected modules
5. I2C interface for I2C controlled modules
6. LCD interface for HD44780 industry standard lcd modules
7. USB to UART, converter with CP2102 chipset (mini-usb-connection)
8. Power LED
9. 4x programable LEDs
10. 4x programable buttons
11. Potentiometer: contrast control for LCD displays
12. RTC battery bracket
13. PCF8563: onboard RTC chip
14. 32.768L crystal: RTC crystal
15. CP2102: onboard USB TO UART chip, for debugging
16. CP2102: Jumper ON/OFF
17. RTC: jumper ON/OFF
18. User LEDs jumper for: ON/OFF
19. User keys jumper for: ON/OFF

2. Introduction

Raspberry Pi Expansion/Evaluation Board, features various interfaces

3. Basic operations

3.1 System image file programming

Please load our prebuild Raspbian Lite image file from our website: [Download](#)

Format your micro SD card with the SDFormatter.exe.

Notices: The capability of micro SD card in used here should be more than 4GB. In this operation, a micro SD card reader is also required, which has to be purchased separately.

Start the Win32DiskImager.exe, and select the system image file copied into your PC, then, click the button **Write** to program the system image file.

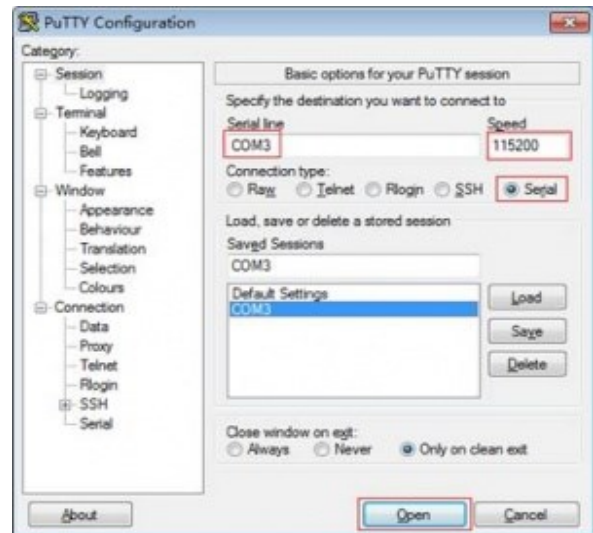
3.2 Serial debugging environment deploying

Connect your PC to the UART TO USB interface on EXP500 via a mini USB cable;

- **Notices: The USB part of the UART TO USB interface applied by EXP500 is a mini USB interface, while the one on Raspberry Pi board is a micro USB interface. Please take a note.**

Install the cp2102_driver;

Start the software PuTTY.exe, and configure the following parameters:



- Serial line: it is used to select corresponding serial port. The serial port in used can be check by Device Manager.
- Speed: it is used to set the Baud rate: 115200.
- Connection type: this option should be set to Serial.
And then, click the button **Open**.

After the system is started up, you should input following information: User name: pi and Password: raspberry. Then, you can enter the serial terminal to communicate with the Raspberry Pi.

Notices: In this document, the software PuTTY is always used to control the Raspberry Pi via serial port communication, unless otherwise specified. When a serial port is used for Raspberry Pi terminal debugging, it cannot service as a common serial port any more, since it is occupied by system debugging function.

4. Installing the libraries required

Before using the EXP500 programs, you shall install the the bcm2835, wiringPi and python libraries to the RPi to add the additional APIs, and configure the settings to start up the core drivers of I2C, SPI and UART after the libraries installed.

In case that you don't want to reinstall the libraries, you can program this ready-to-use system image file to your Raspberry Pi board.

You can find the API source code in the directory /home/pi/EXP500 only if using the ready-to-use system image.

Note: Each demo should be run under the corresponding directory, e.g. to use the bcm2835 demo of the LED, execute the command:

```
pi@raspberrypi ~/EXP500/LED/bcm2835 $ sudo ./led
```

This means that, you should first enter the path ~/EXP500/LED/bcm2835 than execute the command

```
sudo ./led
```

 (./led is a compiled executable file).

The sudo means that the command is executed by a super user.
The sudo is not required if you log in as a root user, e.g.

```
root@raspberrypi:/home/pi/EXP500/LED/bcm2835# ./led
```

Unless otherwise specified, the following demos are pre-compiled to executable files which can be run directly. If you intend to compile the C program of bcm2835, wiringPi, sysfs, etc. then please enter the demo directory and execute the command make to compile the program while the command make clean to remove the executable files. For example:

```
pi@raspberrypi ~/EXP500/LED/bcm2835 $ make clean  
rm led  
pi@raspberrypi ~/EXP500/LED/bcm2835 $ make  
gcc -Wall led.c -o led -lbcm2835
```

Python program can be run directly without compilations, e.g.

```
pi@raspberrypi ~/EXP500/LED/python $ sudo python led.py
```

4.1 LED Demo

bcm2835 program:

1. Enter the Linux terminal, and run the following commands:

```
pi@raspberrypi ~/EXP500/LED/bcm2835 $ sudo ./led
```

2. The 4 LEDs will light up one by one.
3. Press the keys **Ctrl+C** to end the demo.

wiringPi program:

1. Enter the Linux terminal, and run the following commands:

```
pi@raspberrypi ~/EXP500/LED/wiringPi $ sudo ./led
```

2. The 4 LEDs will light up one by one.
3. Press the keys **Ctrl+C** to end the program.

Python program:

1. Enter the Linux terminal, and run the following commands:

```
pi@raspberrypi ~/EXP500/LED/python $ sudo python led.py
```

2. The 4 LEDs will light up one by one.
3. Press the keys **Ctrl+C** to end the program.

shell program:

1. Turn on LED0 by this command:

```
pi@raspberrypi ~/EXP500/LED/shell $ sudo ./LED.sh 26 1
```

2. Turn off LED0 by this command:

```
pi@raspberrypi ~/EXP500/LED/shell $ sudo ./LED.sh 26 0
```


4.2 Key Demo

bcm2835 program:

1. Enter the Linux terminal, and run the following commands:

```
pi@raspberrypi ~/EXP500/KEY/bcm2835 $ sudo ./key
```

2. The terminal will show whether there is key-press.
3. Press the keys Ctrl+C to end the program.

wiringPi program:

1. Enter the Linux terminal, and run the following commands:

```
pi@raspberrypi ~/EXP500/KEY/bcm2835 $ sudo ./key
```

2. The terminal will show whether there is key-press.
3. Press the keys Ctrl+C to end the program.

Python program:

1. Enter the Linux terminal, and run the following commands:

```
pi@raspberrypi ~/EXP500/KEY/bcm2835 $ sudo ./key
```

2. The terminal will show whether there is key-press.
3. Press the keys Ctrl+C to end the program.

4.3 LCD1602 (Purchased Separately) Demo

Insert the LCD1602 into the LCD1602 interface on EXP500.
Enter the Linux terminal, and run the following commands:

```
pi@raspberrypi ~/EXP500/LCD1602 $ sudo ./lcd1602
```

LCD1602 will display relative information. If there is nothing shown, please adjust the potentiometer on the LCD1602. The red box in Figure 11 shows the position of the potentiometer.



Inserting the LCD1602 into the LCD1602 interface on EXP500

4.4 PCF8563 RTC Demo

Set the jumpers on EXP500:

- Connect RTC_SDA to SDA
- Connect RTC_SCL to SCL

Install i2c-tools. If you have not installed i2c-tools yet, please enter the terminal, and input:

```
root@ raspberrypi:/# apt-get install i2c-tools
```

Enter the Linux terminal, and run the following commands:

```
root@ raspberrypi:/# i2cdetect -y 1
```

Then, you will see the device address of PCF8563 connected to Raspberry Pi module. Here, the device address of PCF8563 is 51, indicating that the PCF8563 is detected by Raspberry Pi.

Enter the terminal, and execute corresponding commands:

bcm2835 program:

```
pi@raspberrypi ~/EXP500/PCF8563/bcm2835 $ sudo ./pcf8563
```

wiringPi program:

```
pi@raspberrypi ~/EXP500/PCF8563/wiringPi $ sudo ./pcf8563
```

python program:

```
pi@raspberrypi ~/EXP500/PCF8563/python $ sudo ./pcf8563.py
```

Time of PCF8563 will be shown on the terminal.

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

PCF8563 is detected by Raspberry Pi

4.5 UART Demo

Note: The Serial Port of Raspberry pi is used for terminal debugging by default, so for using UART function, the Serial terminal debugging function needs to be closed.

Connect the onboard UART TO USB port to a PC via a mini USB cable.

Enter the terminal, and execute corresponding commands:

wiringPi program:

```
pi@raspberrypi ~/EXP500/UART/wiringPi $ sudo ./UART
```

python program:

```
pi@raspberrypi ~/EXP500/UART/python $ sudo ./uart.py
```

Run Serial monitor software, choose a corresponding COM port and set the baud rate as 115200. The field sent by the software will be sent back and shown on the software.

5. Libraries Installation for RPi

In order to use the API examples we provide, related libraries are required, which should be installed manually.

[bcm2835 libraries](#)

[wiringPi libraries](#)

5.1 Install WiringPi Library

[Click to download the wiringPi libraries](#), or you can also obtain the latest version from the WiringPi website:

<https://projects.drogon.net/raspberry-pi/wiringpi/download-and-install/>

Copy the installation package into your own system via a USB flash drive, enter the WiringPi folder, then do this to install:

```
chmod 777 build  
./build
```

Run the following command to check the installation:

```
gpio -v
```

5.2 Install C Library bcm2835

Click to download the [bcm2835-1.50 libraries](#).

You can also obtain the latest version from the bcm2835 website: <http://www.airspayce.com/mikem/bcm2835/>

Copy the installation package into your own system, enter the bcm2835 libraries folder, then do this to install:

```
./configure  
make  
sudo make check  
sudo make install
```

5.3 Install Python Library

Python Libraries for Raspbian (contain RPi.GPIO and spidev installation packages.

See: <https://pypi.python.org/pypi/RPi.GPIO> <https://pypi.python.org/pypi/spidev>) get it by apt-get commands.

Please take a note, your Raspberry Pi should be connected to the network when using the command apt-get to install the library. Before the installation, you can run the following command to update your software list.

```
sudo apt-get update
```

1. Run the following command to install the package python-dev

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

2. Installing the RPi.GPIO package (GPIO interface functions). Copy the installation package RPi.GPIO to your RPi board, and unzip it. Enter the unzipped file under the terminal, and run the following command to install the library:

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

3. Run the following command to install the library smbus (I2C interface functions)

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

4. Run the following command to install the library serial, which contains UART interface functions

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

5. Installing the library spidev (SPI functions). Copy the installation package spidev to your RPi board, and unzip it. Enter the unzip file under the terminal, and run the following command to install the library:

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

5.4 Configuring the interfaces

(Before running the API codes we provided, you should start up the corresponding core drivers of the interfaces. In the ready-to-use system image file, both I2C and SPI are set to Enable by default, but the serial port is still in the terminal debugging function mode.)

1. **Enable the I2C function.** Run the following command to configure your Raspberry Pi board:

```
sudo raspi-config
```

Select Advanced Options -> I2C -> yes, to start up the I2C core driver.

Then you also need to modify the configuration file. Run the following command to open the configuration file:

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

Add the following two lines to the configuration file

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

Press the keys Ctrl+X to exit, and input Y to save the settings. Then, reboot the module to make the settings take effect.

- 2. Enable the serial function.** The serial port of RPi is set to serial terminal debugging function mode by default.

If you want the serial port services as a common IO, you should modify the settings on the RPi. When the terminal debugging function is disabled, you cannot access RPi board via the serial port any more. If you want to control the RPi, you may need to enable the debugging function of the serial port again.

```
sudo raspi-config
```

Select Advanced Options -> Serial. Select the option no can disable the serial debugging function. And then, the serial port can be used for serial communication. And select the option yes can enable the serial debugging function.

You should reboot the module to make the settings take effect.

Note: the serial port on Raspberry Pi 3 Model B is unusable, because Pin 14 and Pin 15 is connected to the on-board Bluetooth model.

Start up the spi function, run the following command:

```
sudo raspi-config
```

Select Advanced Options -> I2C -> yes, to start up I2C core driver.

6. Support

If you have any questions , or problems please contact us via e-mail, phone or support system.

E-Mail: service@joy-it.net

Support-System: <http://support.joy-it.net>

Phone: +49 (0)2845 98469 – 66 (11- 18 Uhr)

Please visit our website for more information:

www.joy-it.net