

Starter Kit for Raspberry Pi

The Raspberry Pi is a low-cost mini single board computer that can run different versions of Unix / Linux (Pi 2 also support Windows 10), can act as a media player or even like a micro controller. Whether you are a beginner or an expert in electronics, you will always be in searching for a good Raspberry Pi kit for your project. This is a starter kit for Raspberry Pi. This kit come with the latest Raspberry Pi2-model B, Raspberry Pi B+ and Raspberry Pi3-model B. This is an inclusive kit with an 3.5 inch touch screen TFT display, you can direct operate the Raspberry Pi by the touch screen Also it equipped with a Crowtail- Stackable Board, so you can easy to use the GPIO port, UART port and I2C port on the Raspberry Pi. And it's work perfectly with our crowtail series products.

Chapter 1. How to Control Raspberry Pi GPIO Pins.

Raspberry kernel has been compiled with the gpio driver, we often complete the specific operation through a number of third-party library functions.

The more common operating library functions are:

1.1 Python GPIO

Development language: Python

Basic introduction: It is recommended in the Raspberry official data and easy to use. Python GPIO is a small python library that can help you perform raspberry pi-related IO operations, but the python GPIO library does not yet support SPI, I2C, or 1-wire bus interfaces.

Download Website : <https://sourceforge.net/projects/raspberry-gpio-python/>

Development language: C

Basic introduction: The wiringPi is suitable for those who have a C language foundation and have already been exposed to microcontroller or embedded development prior to contact with the raspberry pi. The API function of wiringPi is very similar to Arduino, which makes it very popular. The author gives a lot of instructions and sample code, these sample code also includes UART devices, I2C

devices and SPI devices.

Official Website : <http://wiringpi.com/>

1.3 BCM2835 C Library

Development language: C

Basic introduction: BCM2835 C Library can be understood as the use of C language related to the underlying driver, BCM2835 C Library driver library including GPIO, SPI and UART, etc. can learn BCM2835 C Library through the BCM2835 related register operation. If you have the opportunity to develop Raspberry Pi to send the Linux driver, or self-developed python or PHP extension driver, you can find from the BCM2835 C Library a lot of "inspiration."

Official Website: <http://www.airspayce.com/mikem/bcm2835/>

2.1 Raspberry Pi GPIO numbering method

BCM Encoding	wPi Encoding	Function	Mode	V	Physical Interface	V	Mode	Function	wPi Encoding	BCM Encoding
		3.3v			1 2			5v		
2	8	SDA.1	ALTO	1	3 4			5V		
3	9	SCL.1	ALTO	1	5 6			0v		
4	7	GPIO. 7	IN	1	7 8	0	ALTO	TxD	15	14
		0v			9 10	1	ALTO	RxD	16	15
17	0	GPIO. 0	IN	0	11 12	0	IN	GPIO. 1	1	18
27	2	GPIO. 2	IN	0	13 14			0v		
22	3	GPIO. 3	IN	0	15 16	0	IN	GPIO. 4	4	23
		3.3v			17 18	1	OUT	GPIO. 5	5	24
10	12	MOSI	ALTO	0	19 20			0v		
9	13	MISO	ALTO	1	21 22	1	OUT	GPIO. 6	6	25
11	14	SCLK	ALTO	1	23 24	1	ALTO	CE0	10	8
		0v			25 26	1	ALTO	CE1	11	7
0	30	SDA.0	ALTO	1	27 28	1	ALTO	SCL.0	31	1
5	21	GPIO.21	IN	1	29 30			0v		
6	22	GPIO.22	IN	1	31 32	0	IN	GPIO.26	26	12
13	23	GPIO.23	IN	0	33 34			0v		
19	24	GPIO.24	IN	0	35 36	0	IN	GPIO.27	27	16
26	25	GPIO.25	IN	0	37 38	0	IN	GPIO.28	28	20
		0v			39 40	0	IN	GPIO.29	29	21

3.1 python GPIO

3.1.1 Firstly, install python-dev, enter the following command.

Sudo apt-get install python-dev

3.1.2 Install RPi.GPIO, and then enter the following commands.

- Download: <https://pypi.python.org/pypi/RPi.GPIO/>
- Unzip: tar xvzf RPi.GPIO-0.6.3a.tar.gz
- Enter the directory after unpacking: cd RPi.GPIO-0.6.3a
- Start the installation: sudo python setup.py install

4.1 Introduction

4.1.1 GPIO.setmode(GPIO.BOARD), Use socket pin numbering.

4.1.2 As a result of the socket pin number, where the 11 feet equivalent to the number of BCM2835 register pin 11.

4.1 How to install wiringPi

4.1.1 Scenario A - Use the GIT tool

Get the wiringPi source code through GIT

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

```
cd wiringPi
```

```
./build
```

The build script will help you compile and install the wiringPi

4.1.2 Scenario B - Direct download

We can refer to this site <https://git.drogon.net/?p=wiringPi;a=summary> to directly download the latest version of the compiler to use

```
tar xzf wiringPi-xx.tar.gz
```

```
cd wiringPi-xx
```

```
./build
```

4.1.3 Test

The wiringPi includes a set of gpio commands. You can use the gpio command to control various interfaces on the raspberry. You can use the following command to

test whether the wiringPi is installed successfully.

```
gpio -v
```

```
gpio readall
```

4.1.3 Example:



```
[cpp]      
  
int main(void  
  
    {  
    for (int i = 0; i < 2; i++)  
        printf("wiringPi version: %d.%d\n",  
              WPI_VERSION_MAJOR, WPI_VERSION_MINOR);  
    }  
}
```

4.1.5 Compile and run:

In the raspberries:

```
gcc -Wall -o test test.c -lwiringPi
```

```
sudo. / test
```

In a virtual machine:

```
am-linux-gcc -Wall -o test test.c -lwiringPi
```

```
sudo. / Test
```

4.1.6 Note

- IO number slightly different ways, using wiring coding.
- -lwiringPi said that the dynamic loading wiringPi shared library.

5.1 BCM2835 C Library

- Download: <http://www.airspayce.com/mikem/bcm2835/bcm2835-1.35.tar.gz>
- Unzip: `tar xvzf bcm2835-1.35.tar.gz`
- Enter the directory after unpacking: `cd bcm2835-1.35`

- The configuration: `/ configure`
- From the source code to generate the installation package: `make`
- The implementation of inspection: `sudo make check`
- Install bcm2835 library: `sudo make install`

5.2Note:

- IO number slightly different ways, using wiring coding.
- -lwiringPi said that the dynamic loading wiringPi shared library

Chapter 2. How to choose Raspberry Pi UART boot overlay

The raspberry official system Raspbian Jessie releases default serial port is for Bluetooth use. If you want the serial port as the control terminal debug port that serial console, you need to modify the system configuration, turn off Bluetooth and map pl011 UART to pin14 and pin15.

1.1 So that the console serial steps are as follows:

Step1: Install and configure system

The Raspbian Jessie releases system is installed on the SD card.configure system:

```
Sudo raspi-config
```

Run:

```
Expand filesystem and enable serial on advanced page exit and reboot
```

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

```
sudo reboot
```

Step2:Modify configuration

Edit/lib/systemd/system/hciuart.server, then repalce the file ttyAMA0 to ttyS0

```
sudo vi /lib/systemd/system/hciuart.server
```

Repalce the ttyAMA0 to ttyS0

Step3 Setup the device tree

```
sudo vi /boot/config.txt
```

At the end of the file add:

```
dtoverlay=pi3-miniuart-bt
```

Save and ESC.

Step4 Enable serial terminal console serial

Edit /boot/cmdline.txt file

```
Sudo vi/boot/cmdline.txt
```

Change the content of:

```
dwc_otg.lpm_enable=0 console=serial1,115200 console=tty1
```

```
root=/dev/mmcblk0p2 kgdboc=serial1,115200 rootfstype=ext4
```

```
elevator=deadline fsck.repair=yes rootwait
```

Save the file and restart the system

```
sudo reboot
```

At this time, you can use the serial input and output control and log PI;

PC serial port configuration: 115200 8n1 (8 bits, no parity, 1 stop bit)

2.1 Set serial port for universal serial communication with external

devices

If you want to set the serial port as universal serial port, it is to close the console serial. Specific steps are as follows:

In the completion of the above 3 step premise, edit the /boot/cmdline.txt file

```
sudo/boot/cmdline.txt
```

Modify the contents of the file as:

```
dwc_otg.lpm_enable=0 console=tty1 root=/dev/mmcblk0p2
```

```
rootfstype=ext4 elevator=deadline fsck.repair=yes rootwait
```

Save the file and ESC.

At this time we can use the serial port, serial port device number is /dev/ttyAMA0

Serial port test:

Will be connected to the serial port through the PC to the PC, the usb-ttl side to open the serial port debugging assistant, serial port is set to 115200 8n1

Install minicom first: `sudo apt-get install minicom`

Then open minicom : `sudo minicom-d/dev/ttyAMA0 -b115200`

```
pi@raspberrypi: ~  
Welcome to minicom 2.7  
  
OPTIONS: I18n  
Compiled on Jan 12 2014, 05:42:53.  
Port /dev/ttyAMA0, 14:00:01  
  
Press CTRL-A Z for help on special keys  
█
```

Input ctrl-a-> e to switch between display and non display input.

You can achieve a raspberry pie and PC via serial communication, baud rate can also be modified to 9600 and 38400.

Chapter 3. How to use IIC port.

Raspberry pi has integrated I2C drive, only need to modify the configuration file start on the line.

```
sudo nano /etc/modprobe.d/raspi-blacklist.conf
```

```
GNU nano 2.2.6 File: /etc/modprobe.d/raspi-blacklist.conf  
# blacklist spi and i2c by default (many users don't need them)  
#blacklist spi-bcm2708  
#blacklist i2c-bcm2708  
  
[ Read 4 lines ]  
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos  
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Note third lines out, CTRL + X save

Next start I2C module


```

GNU nano/2.2.6@be.d/raspi-b File: /etc/modules
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.
# Parameters can be specified after the module name.

snd-bcm2835
i2c-dev

```

Add a line of `i2c-dev`

Install tool

```
sudo apt-get install i2c-tools python-smbus
```

Add the PI user to the I2C group

```
sudo adduser pi i2c
```

Restart

```
sudo shutdown -r now
```

Test I2C

Connect the I2C device, and run the following command for raspberry pi A:

```
sudo i2cdetect -y 0
```

Run the following command for raspberry pi B:

```
sudo i2cdetect -y 1
```

Will see I2C address

```

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

```

Examples:

1. Button control LED

Use the Crowtail 3 pin cable connect the Crowtail- LED to the Crowtail- Breakout board D25, and connect the Crowtail- Button to the Crowtail- Breakout board D22.

Then download the configured system for 3.5 inch display:

<https://www.dropbox.com/s/bp57zo2ojs1odyt/LCD35-150602.img?dl=0> (B+ 2B)

<https://www.dropbox.com/s/js8ypj4rhs3zw19/onfigured%20system%20for%20PI3B.zip?dl=0>(3B)

Or if you have use yourselves system, then only need to down the driver and configured it to your system. How to configure the driver to the system that you can refer to this wiki site:

http://www.elecrow.com/wiki/index.php?title=3.5_Inch_480x320_TFT_Display_with_Touch_Screen_for_Raspberry_Pi

Connection like as bellow:



Open the terminal raspberry pi input `sudo Python` as administrator to open the python program. Enter the following instructions in turn:

```
>>>GPIO.setmode(GPIO.BCM)
>>>GPIO.setup(25,GPIO.OUT) #Set pin 25OUTPUT
>>>GPIO.output(25,GPIO.HIGH) #25 pin HIGH
>>>GPIO.output(25,GPIO.LOW) #25 pin LOW
```

The above instructions can control the output of the pin, the python program can also read the input pin, which will be reflected in the next example.

Although the instructions can control the I/O port, of course we want most is to generate the executable file, raspberry pi do this naturally is no problem. First, we need to create an empty text, in the text input the following program (the program content as shown in the notes).

```
Import RPi.GPIO as GPIO # in GPIO Library
Import time # transferred time library
GPIO.setmode (GPIO.BCM)
GPIO.setup (24, GPIO.IN) # 24 feet set as input
```

```
#GPIO.setup (24, GPIO.IN, pull_up_down=GPIO.PUD_UP)
# 24 feet set on the pull mode, if no pull-up resistor button can use this sentence.
GPIO.setup (23, GPIO.OUT) # 23 pin set to output
Count=0
While True: # loop
InputValue = GPIO.input (24) # read key
If (inputValue==0): # value judgment
Count=count+1
If (count%2==0):
Print ("THE LED IS ON NOW") # output terminal
Else:
Print ("THE LED IS OFF NOW") # output terminal
Time.sleep (0.3) # delay
If (count%2==0): # according to the count value LED lamp is opened and closed
GPIO.output (23, GPIO.HIGH)
Else:
GPIO.output (23, GPIO.LOW)
Time.sleep (0.01) # delay to the system idle time
```

Note: the number of # 72, 1 segments, with a space character level control program, Python program, C language instead of "{}". In 2, Python in the program comment character is "#" instead of "/".

Save text as KEYLED.py file. Then enter the raspberry pi input terminal, sudo Python keyed.py, run our written procedures. We can see that when the button is pressed, LED lights change, the terminal display LED IS ON NOW , when the button is pressed again, LED lights out, the terminal display LED IS OFF NOW .