# How to program the Embedded Pi with Raspberry Pi

Since the release of Embedded Pi, there are complaints about not being able to use the Raspberry Pi itself to program the STM32 on Embedded Pi, as CoIDE is currently Windows specific. We value feedback of every customer and had been searching for a solution for this. Good news is that now this issue is addressed by using an open source ISP tool on the Raspberry Pi. This article introduces how to install and use the ISP tool on the Raspberry Pi.

First, download the ISP tool STM32Loader from https://github.com/coocox/stm32loader. This tool, original version by Ivan A-R <ivan@tuxotronic.org>, is written in Python script which will talk to the STM32 bootloader to upload and download firmware. It's recommended of you to use Python 2.6 or 2.7, and the installation details won't be described here.

***Usage of STM32Loader:***

```
./stm32loader.py [-hqVewvr] [-l length] [-p port] [-b baud] [-a addr] [file.bin]
```

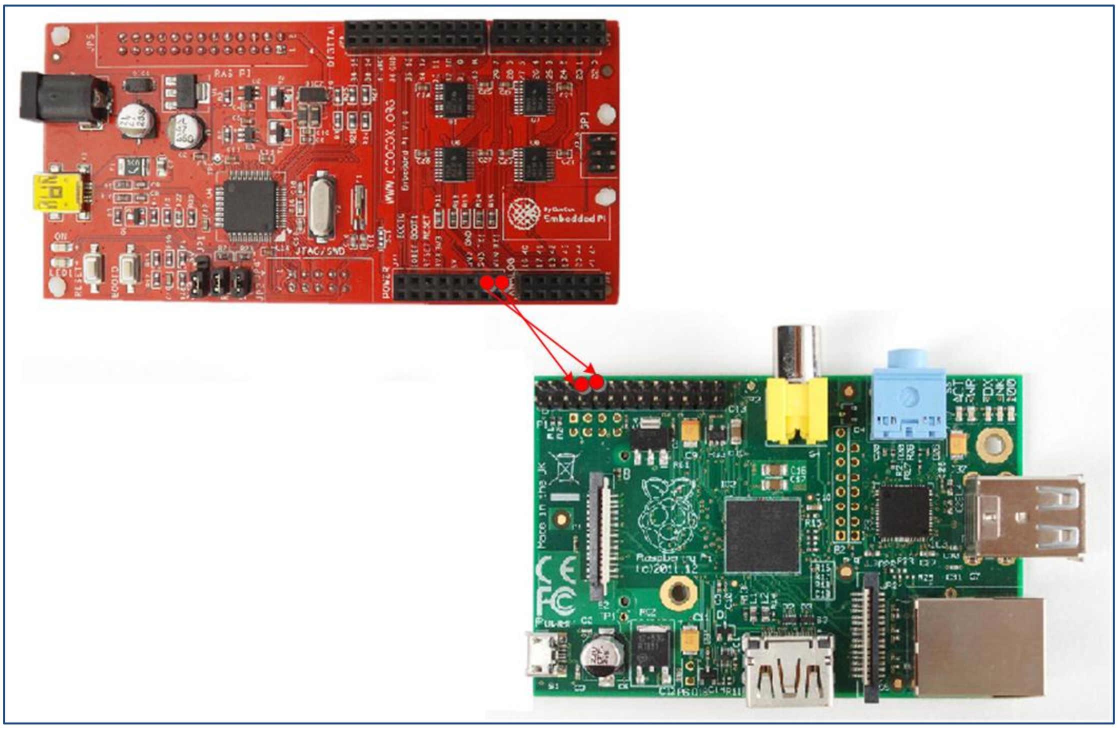
| | |
|---|---|
| -h | This help |
| -q | Quiet |
| -V | Verbose |
| -e | Erase |
| -w | Write |
| -v | Verify |
| -r | Read |
| -l length | Length of read |
| -p port | Serial port (default: /dev/tty.usbserial-ftCYPMYJ) |
| -b baud | Baud speed (default: 115200) |
| -a addr | Target address |

The Raspberry Pi can program the STM32 via the serial port (JP7-TX1 and JP7-RX1) when the Embedded Pi is in ISP mode, refer to section 3.6.1 in Embedded Pi User Manual.

To use this mode, you need to follow the steps below:

1. Set BOOT0 to 1 (high level), and BOOT1 to 0 (low level) – which has been done on hardware. In this case, you only need to press the BOOT0 button to enter this mode when Embedded Pi is powered on.

2. Connect the UART interface on Raspberry Pi to the UART1 interface on Embedded Pi with wires (connect Embedded Pi JP7-TX1 with Raspberry Pi P1-D10, JP7-RX1 with P1-D8), as shown in the figure below.



3. Open a Terminal session, 'cd' to wherever you downloaded the STM32Loader and then issue the following command to download bin file to the Embedded Pi.

   ```
   sudo python stm32loader.py -e -w -v Test.bin
   ```

   An error prompt will appear if your Raspberry Pi has no PySerial library installed. Next part introduces the installation of the PySerial library.

***Install PySerial***

Whatever your operating system is, download the .tar.gz installation package for PySerial 2.6, i.e. pyserial-2.6.tar.gz, from https://pypi.python.org/pypi/pyserial.

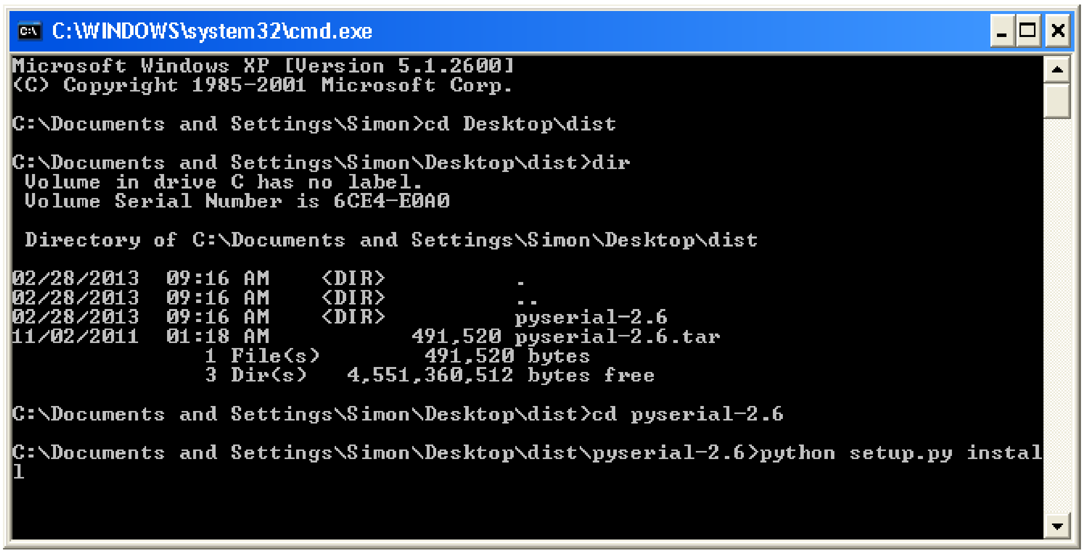
If you are using Windows system, you need to decompress the file into a folder. It is not a normal zip file, so you may need to use a tool such as 7-zip (http://www.7-zip.org/).

If you are using a Mac or Linux computer, then open a Terminal session, 'cd' to wherever you downloaded the pyserial-2.6.tar.gz and then issue the following command to unpack the installation folder.

```
$ tar -xzf pyserial-2.6.tar.gz
```

The rest of the procedure is the same whatever your operating system is. Use you Command Prompt / Terminal session and “cd” into the pyserial-2.6 folder, and then run the command:

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



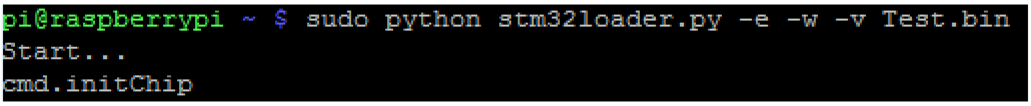
```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Simon>cd Desktop\dist

C:\Documents and Settings\Simon\Desktop\dist>dir
 Volume in drive C has no label.
 Volume Serial Number is 6CE4-E0A0

 Directory of C:\Documents and Settings\Simon\Desktop\dist

02/28/2013  09:16 AM    <DIR>          .
02/28/2013  09:16 AM    <DIR>          ..
02/28/2013  09:16 AM    <DIR>          pyserial-2.6
11/02/2011  01:18 AM           491,520 pyserial-2.6.tar
               1 File(s)        491,520 bytes
               3 Dir(s)   4,551,360,512 bytes free

C:\Documents and Settings\Simon\Desktop\dist>cd pyserial-2.6

C:\Documents and Settings\Simon\Desktop\dist\pyserial-2.6>python setup.py instal
l
```

When the installation is finished, issue the following command to download bin file to the Embedded Pi.

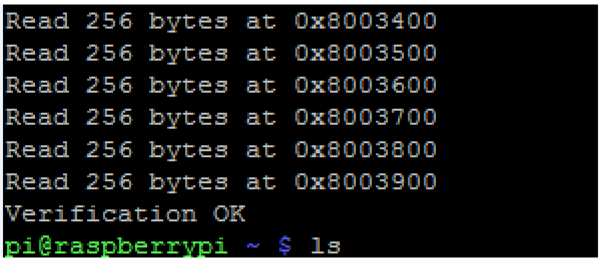
```
sudo python stm32loader.py -e -w -v Test.bin
```

This will pre-erase flash, program Test.bin to the flash on the device, and then perform verification after the programming is finished.

A prompt of “Verification OK” will appear after successful programming, as shown in the figure below:



```
pi@raspberrypi ~ $ sudo python stm32loader.py -e -w -v Test.bin
Start...
cmd.initChip
```



```
Read 256 bytes at 0x8003400
Read 256 bytes at 0x8003500
Read 256 bytes at 0x8003600
Read 256 bytes at 0x8003700
Read 256 bytes at 0x8003800
Read 256 bytes at 0x8003900
Verification OK
pi@raspberrypi ~ $ ls
```

Reset the STM32 and the LED on the Embedded Pi will blink.

If the program is not downloaded successfully, react according to the prompt appears in the Command Prompt / Terminal session:

1. If a prompt of “None” always appears, redo step 1 (press BOOT0 to reset the STM32).

2. If an error prompt like “NACK” appears, retry the download command.

Hope you all enjoy playing with the Embedded Pi. Please feel free to post a question or share your story with the Embedded Pi on our Embedded Pi Forum. Come join us to make the Pi better!