MicroPython Using Report

Made by Chen Derun, Liu Jinbo & Zhang Wengyu

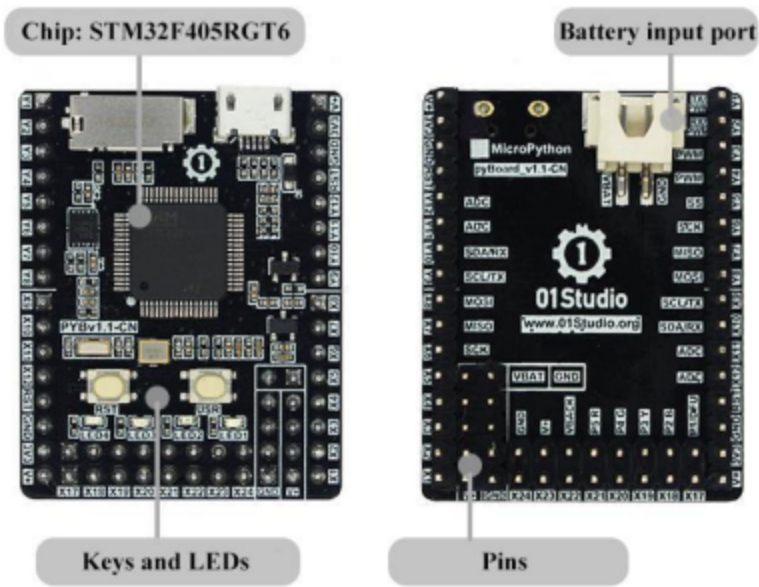
1.MicroPython introduction

MicroPython is a programming language based on Python 3, it can run in a limited environment e.g. Microcontroller. MicroPython is designed to be as compatible as possible with Python. Similar to Python, MicroPython also encapsulates a large number of libraries, so it is easy to control LED, button, motor, AD/DA and other sensors in MicroPython by importing functions.

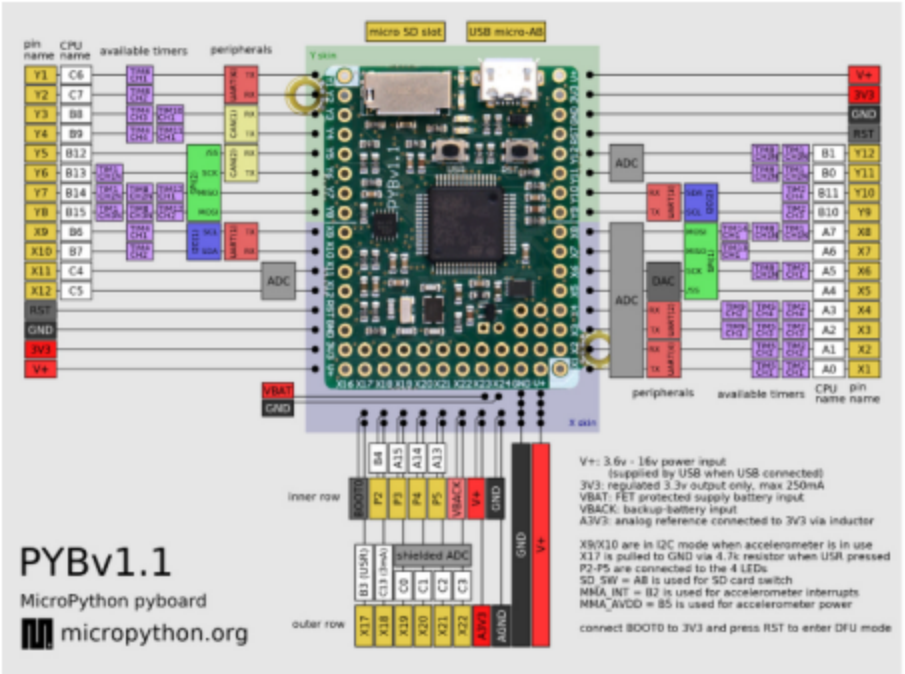
2. Microcontroller for MicroPython

MicroPython is supporting 5 platforms empoldered by 01Studio so far, which are STM32, ESP8266, ESP32, CC3200, K210. What is the most common and basic one is STM32(its specific model number is STM32F405RGT6) that mainly encodes PyBoard v1.1-CN version.

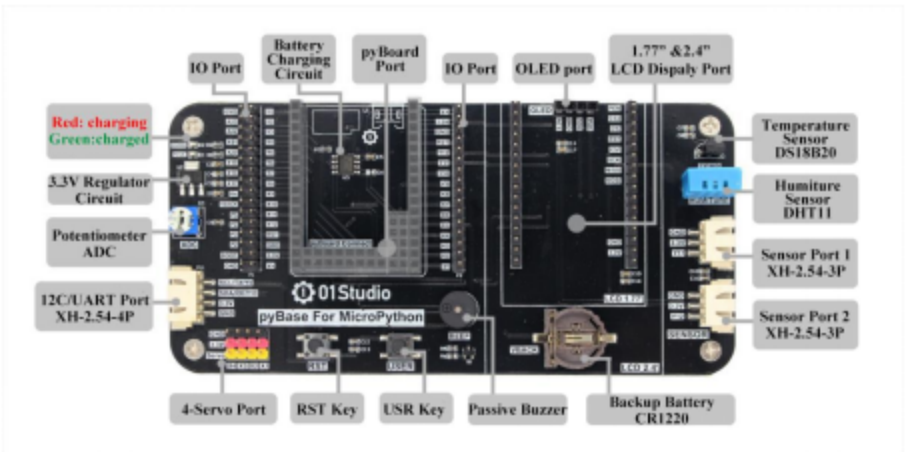
01Studio community: ***www.01Studio.org***



PyBoard v1.1-CN



PyBoard v1.1-CN position of pins



PyBase For MicroPython

Core board	PyBoard v1.1-CN
Bottom board	PyBase
Display screen	0.9 inch OLED screen

3. Developing environment setup

Mu is a development softwares for MicroPython

3.1 Mu install

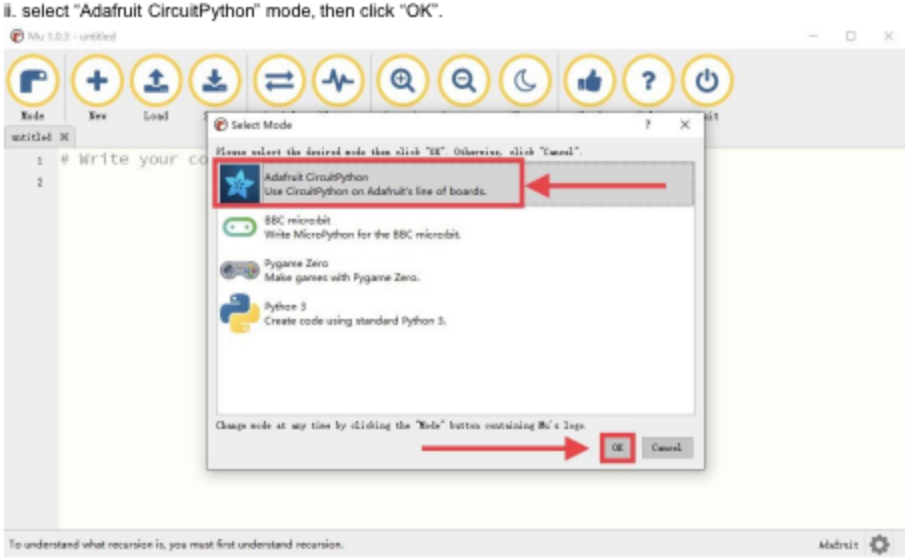
Mu is an open source development software for MicroPython
Download website:
https://codewith.mu/en/download

3.1.1 Mu for Windows users

get Mu install package" from Google Drive link below:
https://drive.google.com/drive/folders/1nQgsahe_f2YkG_ku6AwDro-efzjCye67?usp=sharing
(Windows>Mu>32/64>Mu-32/Mu-64.exe)

i.Click "Windows 64-bit"blue link to download Mu.
(Ps:Before that you should check your computer digit from "My computer" or "This PC")

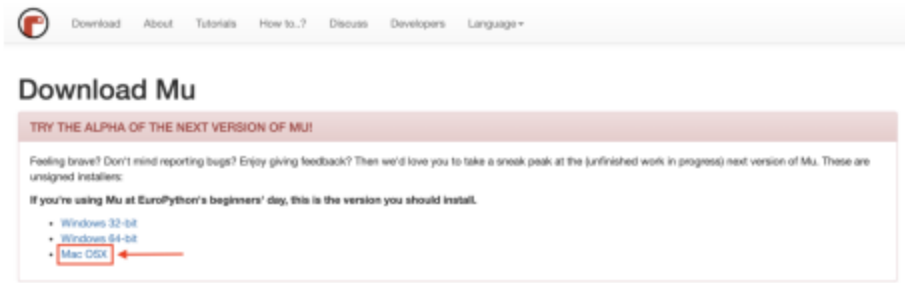
ii. select "Adafruit CircuitPython" mode, then click "OK".



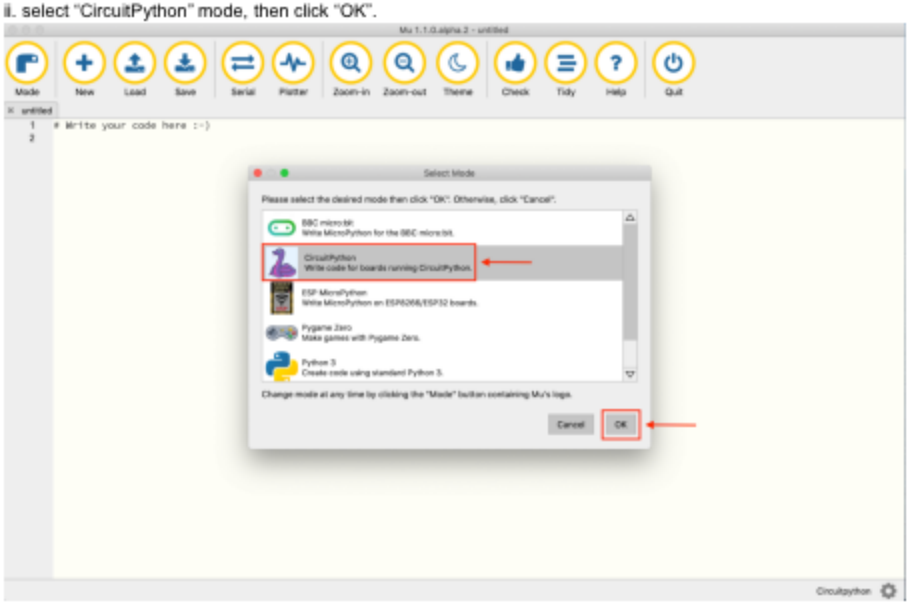
Tips:
(Must install Python 3 at the beginning) When the users download the Mu, after Windows users download Mu, the icon may be different when entering the software. When the mode is selected, different systems will display different modes. The first choice is Circuit Python and If your desktop did not display this option, and that Adafruit Circuit Python can get the same work.

3.1.2 Mu for Mac users

i.Click "Mac OSX" blue link to download Mu



ii. select "CircuitPython" mode, then click "OK".



4. REPL serial port interaction setting

REPL: Read, Eval, Print, Loop.

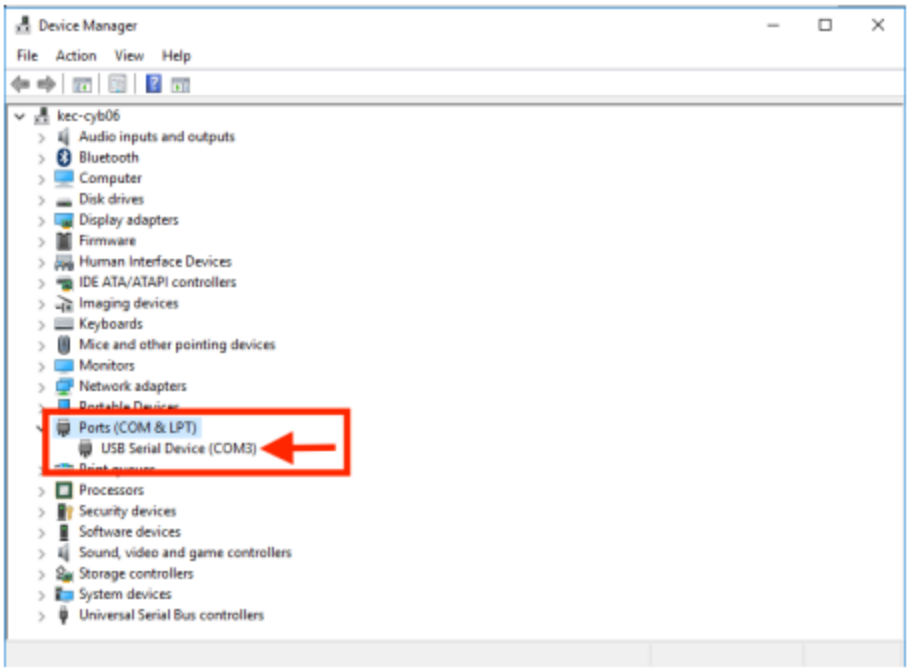
4.1 for Windows users

We can use a serial port terminal software "PuTTY" or "hypertrm" to test pyBoard seiral port.

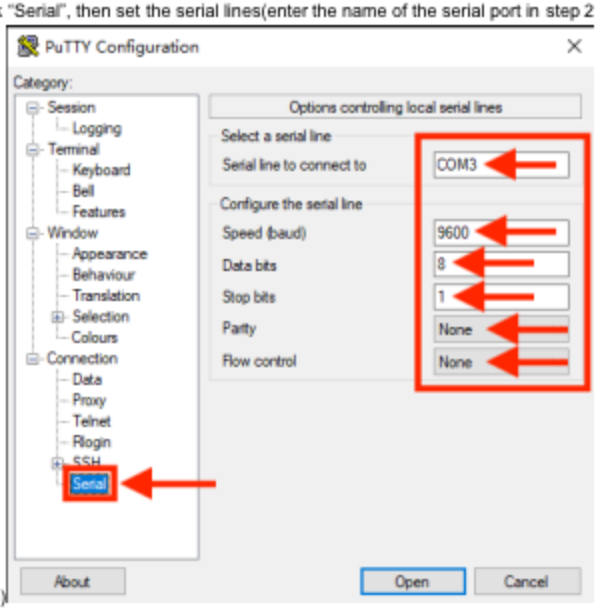
4.1.1.PuTTY:

1)Connect pyboard with PC

2)Find the name of the serial port(from the "Device Manager", here is "COM3")



3)Open "putty.exe"
get the "putty.exe" from the Google Drive link below:
https://drive.google.com/drive/folders/1nQgsahe_f2YkG_ku6AwDro-efzjCye67?usp=sharing
(Windows>REPL>PuTTY>putty.exe)
4)Click "Serial", then set the serial lines(enter the name of the serial port in step 2 e.g. COM3)

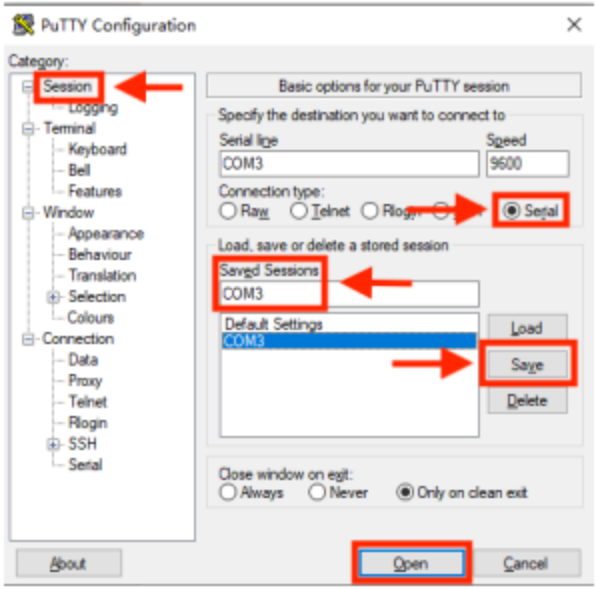


5)**DO NOT CLICK** "Open" directly, click "Session"

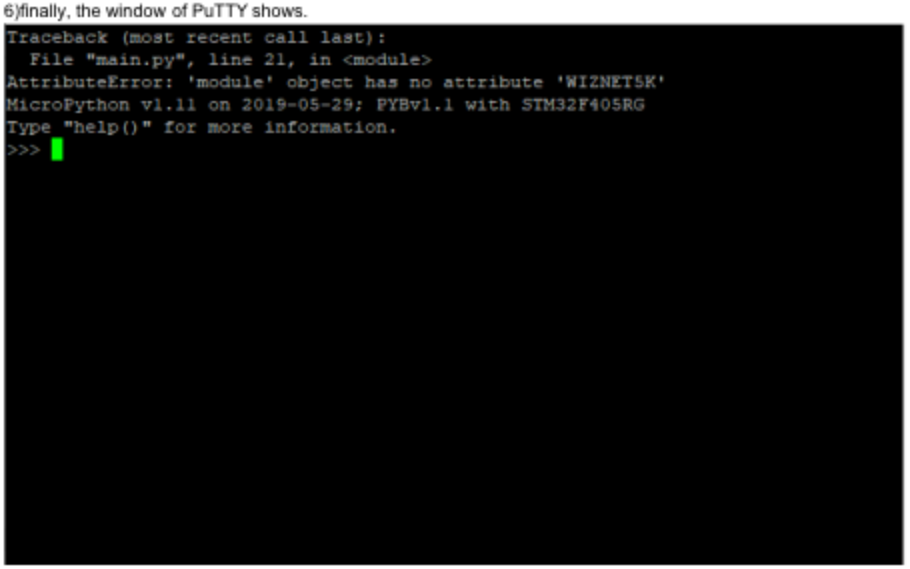
i)choose "Serial" in Connection type, then the Serial line and the Speed will be shown automatically;

ii)enter the name of the serial line in the "Saved Session"(e.g. COM3), then click "Save";

iii)click "Open".



6)finally, the window of PuTTY shows.

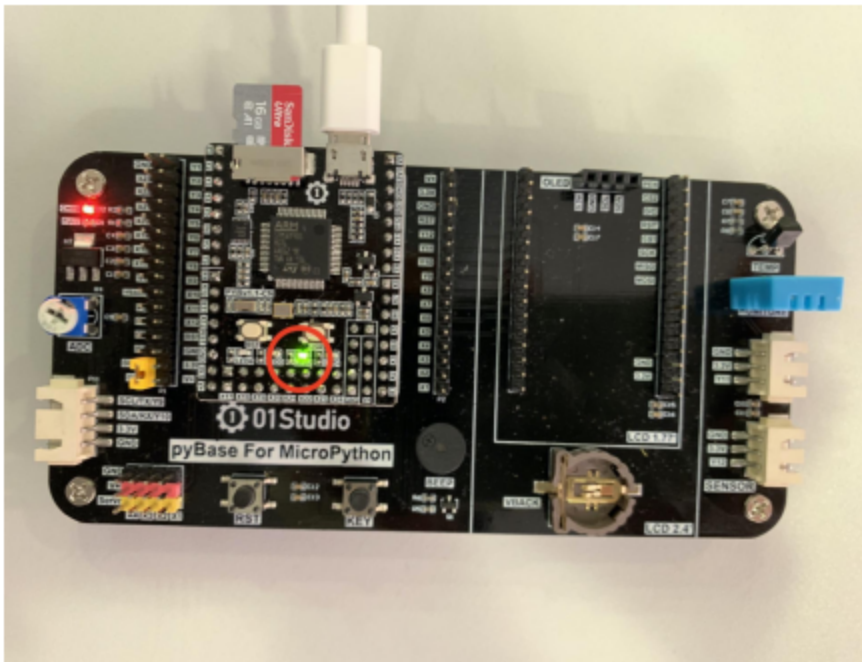


7)Test MicroPython pyboard in REPL with PuTTY.

Enter:
>>>import pyb
>>>pyb.LED(2).on()

```
Traceback (most recent call last):
  File "main.py", line 21, in <module>
AttributeError: 'module' object has no attribute 'WIZNET5K'
MicroPython v1.11 on 2019-05-29; PYBv1.1 with STM32F405RG
Type "help()" for more information.
>>> import pyb
>>> pyb.LED(2).on()
>>>
```

LED2 light up



REMINDER: when using PuTTY in Windows, PuTTY will be shut down after pressing the "PST" reset key. Press [Ctrl+D] to reopen pyboard

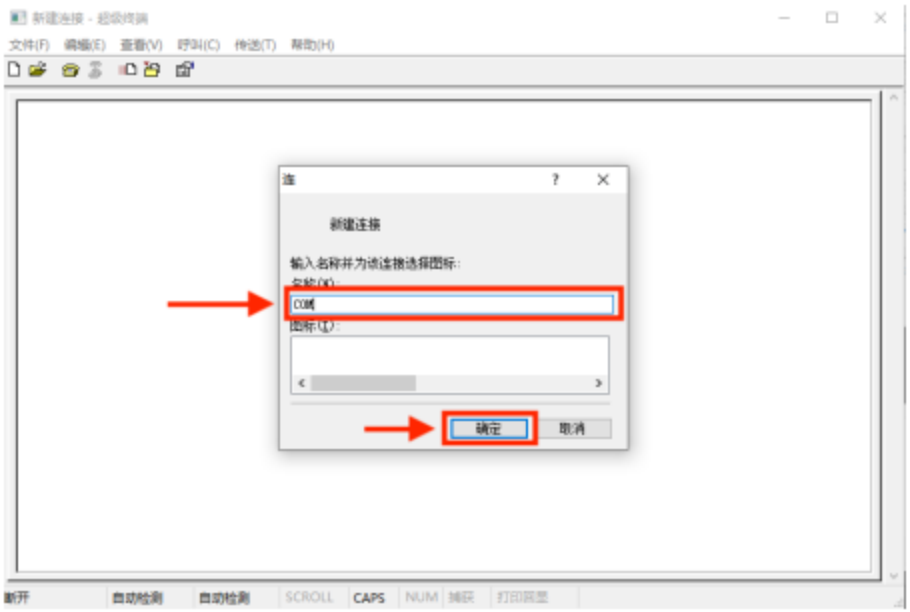
4.2.2 hypertrm:
using hypertrm can solve the shutdown problem in PuTTY.

1)Connect pyboard with PC

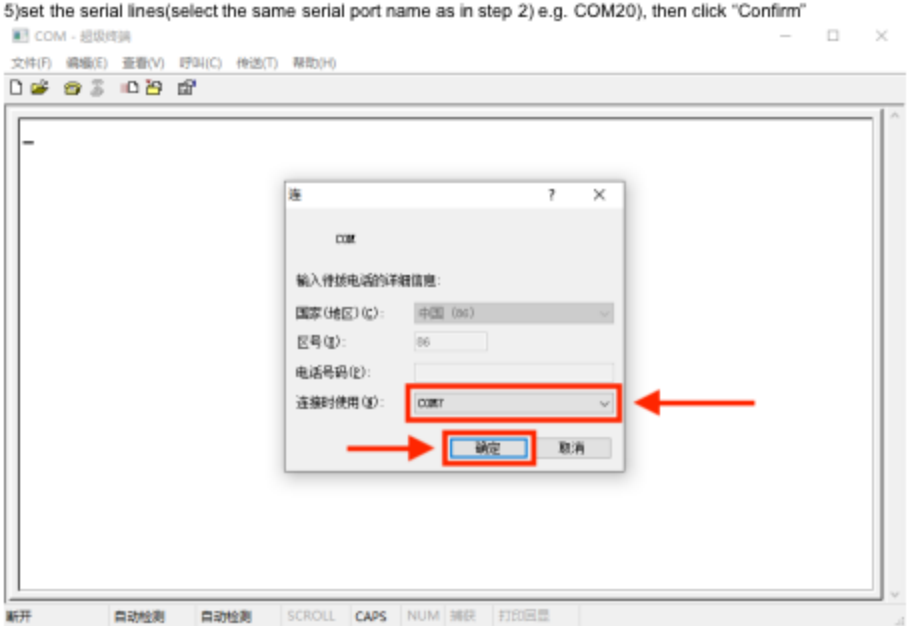
2)Find the name of the serial port(from the "Device Manager", e.g. "COM20")

3)open hypertrm.exe
get "hypertrm.exe" from Google Drive link below:
https://drive.google.com/drive/folders/1nQgsahe_f2YkG_ku6AwDro-efzjCye67?usp=sharing
(Windows>REPL>Windows hypertrm>hypertrm.exe)

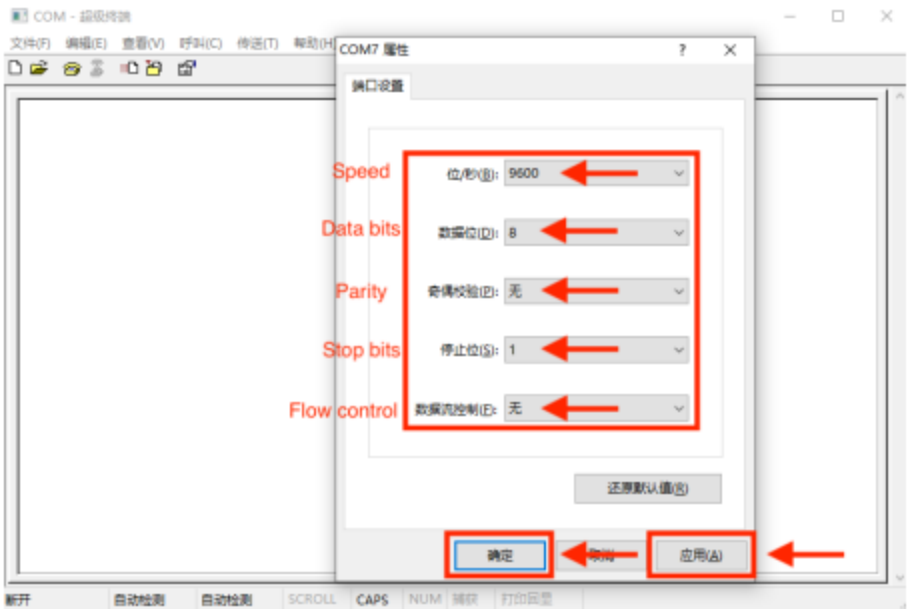
4)enter an arbitrary name of collection (e.g. "COM"), then click "Confirm".



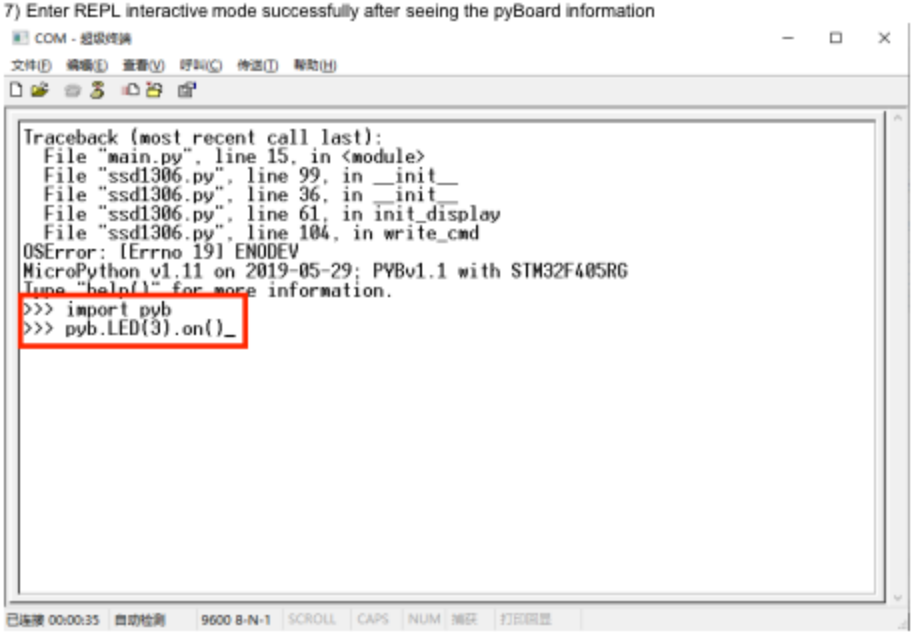
5)set the serial lines(select the same serial port name as in step 2) e.g. COM20), then click "Confirm"



6) serial port setting
follow the settings below, press "Apply" first, then press "Confirm"

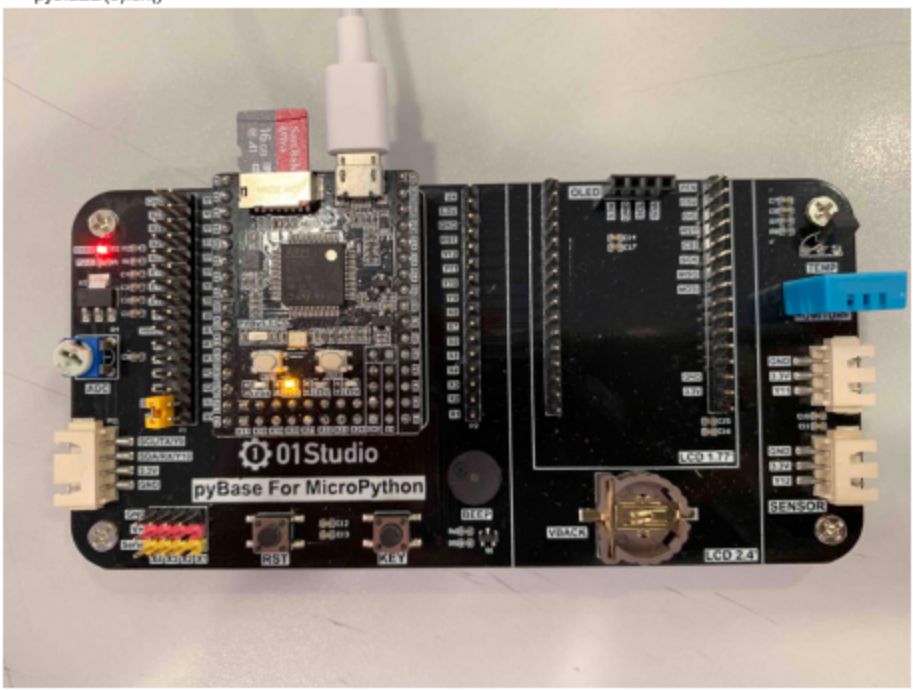


7) Enter REPL interactive mode successfully after seeing the pyBoard information



8)Test MicroPython pyboard in REPL with hypertrm.

Enter:
>>>import pyb
>>>pyb.LED(3).on()



LED3 light up

4.2 for Mac users

We can use **i) MAC's built-in serial terminal via the "screen" command** or **ii) serial port terminal "minicom"** to test pyBoard seiral port and interact with the pyBoard.

4.2.1 screen command

(Red represents CODES, Blue represents TIPS)

1) Enter screen /dev/tty.usbmodem and press 'tab' for completion of system device serial number

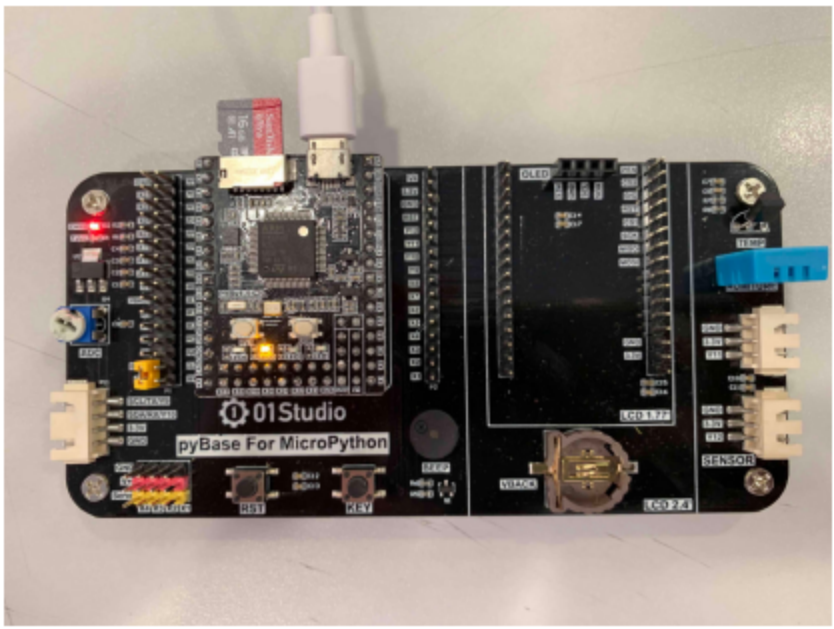
```
[xiaoshanpodeMacBook:~ Jerry$ screen /dev/tty.usbmodem206E3089594D2
```

2) Replenish Bps The Bps here is 115200 after the serial number with '*' above and press 'Return' to get the coding interface Don't forget the space behind the serial number

```
[xiaoshanpodeMacBook:~ Jerry$ screen /dev/tty.usbmodem206E3089594D2 *115200
MicroPython v1.11 on 2019-05-29; PYBv1.1 with STM32F405RG
Type "help()" for more information.
>>>
```

3) Test MicroPython pyboard in REPL with Screen.
Enter:
>>>import pyb
>>>pyb.LED(3).on()

```
MicroPython v1.11 on 2019-05-29; PYBv1.1 with STM32F405RG
Type "help()" for more information.
>>> import pyb
>>> pyb.LED(3).on()
>>>
```



if you want to quit 'Screen' built-in terminal, Enter control + A and then enter control + \, it will ask you whether close the windows, press y to back to the normal terminal and n to stay in 'Screen'. Except this, you can also shut down the terminal progress to close 'Screen'

```
MicroPython v1.11 on 2019-05-29; PYBv1.1 with STM32F405RG
Type "help()" for more information.
>>>
```

```
Really quit and kill all your windows [y/n]
```

4.2.2 serial port terminal "minicom"

(Compare to 'Screen' built-in serial terminal, Recommend Minicom serial terminal with simple use)

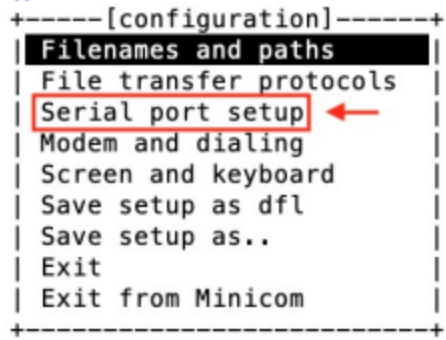
Open the Terminal.app in mac

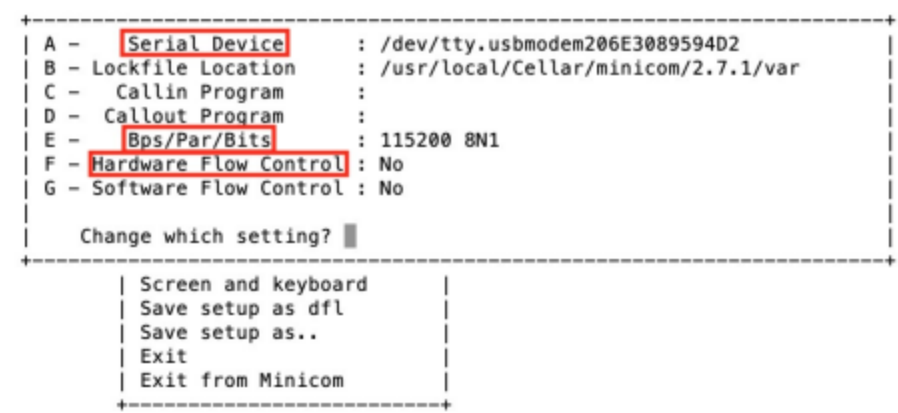
1) Enter curl -LsSf http://github.com/mxcl/homebrew/tarball/master | sudo tar xvz -C/usr/local --strip 1

2) Enter brew install minicom for updating Homebrew if it shows 'Command not found', please check appendix 1

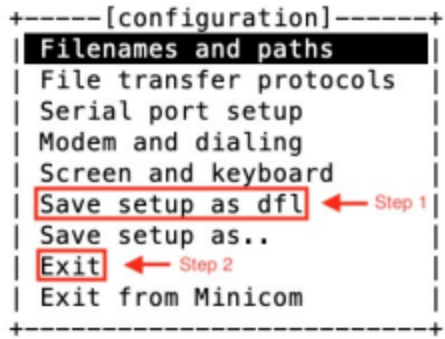
3) Enter minicom -s to open minicom terminal

4) Choose 'serial port setup', change 'Serial Device', 'Bps/Par/Bits' and 'Hardware Flow Control' and choose '115200 8N1' in 'Bps/Par/Bits' and choose 'No' in 'Hardware Flow Control' Check detailed steps please check Appendix 2

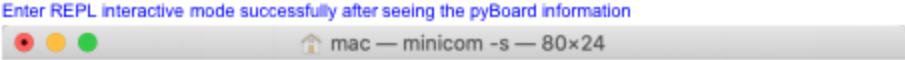




5) Press 'Return key' to the Upper-level catalogue and choose 'Save setup as dfl', then choose 'Exit' to enter minicom Next time just import 'minicom' can enter minicom directly



Enter REPL interactive mode successfully after seeing the pyBoard information

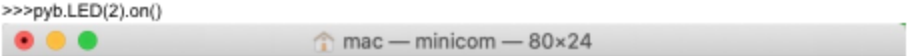



```
Welcome to minicom 2.7.1

OPTIONS:
Compiled on Oct  6 2019, 23:16:03.
Port /dev/tty.usbmodem206E3089594D2, 14:46:00

Press Meta-Z for help on special keys
```

6) Test MicroPython pyboard in REPL with Minicom.
Enter:
>>>import pyb
>>>pyb.LED(2).on()



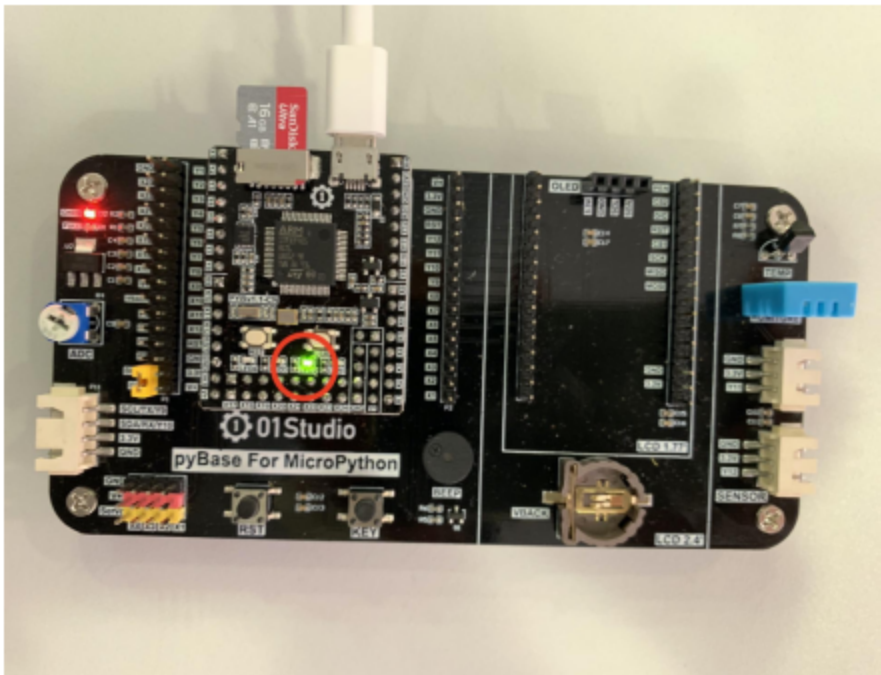

```
Welcome to minicom 2.7.1

OPTIONS:
Compiled on Oct  6 2019, 23:16:03.
Port /dev/tty.usbmodem206E3089594D2, 16:50:46

Press Meta-Z for help on special keys

import pyb
>>> pyb.LED(2).on()
>>>
```

LED2 lights up



Appendix 1:

If it shows 'Command not found' as follows:

```
bogon:Contents bishesu$ telnet
-bash: telnet: command not found
bogon:Contents bishesu$ brew install telnet
Error: Unknown command: install
bogon:Contents bishesu$ brew help
Please run brew update!
bogon:Contents bishesu$ brew --help
Please run brew update!
bogon:Contents bishesu$ brew update
Error: /usr/local must be writable!
bogon:Contents bishesu$
```


Enter:

sudo chown -R $(whoami):admin /usr/local

and if the conclusion is 'chown: /usr/local: Operation not permitted', it means that you need to re-download 'Homebrew' as follows:

```
xiaoshanpodeMacBook:~ Jerry$ sudo chown -R $(whoami):admin /usr/local
Password:
chown: /usr/local: Operation not permitted
```

1) Enter:

/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/uninstall)"

to delete the current Homebrew

Remember to press 'y' for delete confirmation

2) Enter:

/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"

to download Homebrew

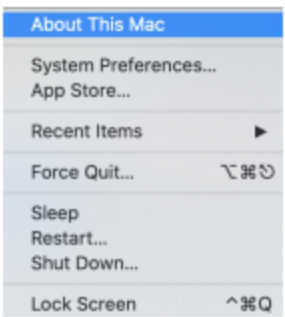
3) Then enter:

brew update' to update brew for install minicom → Import 'brew install minicom

to download minicom terminal

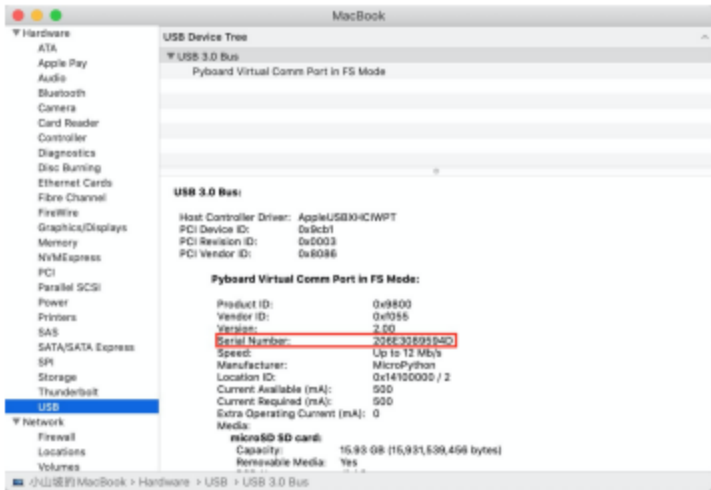
Appendix 2:

1) Back to desktop and press the Apple LOGO which is on the top left corner of the screen and choose 'About this Mac'



2) Enter into "System report" and find the device 'Pyboard Virtual Comm Port in FS mode' in 'USB'

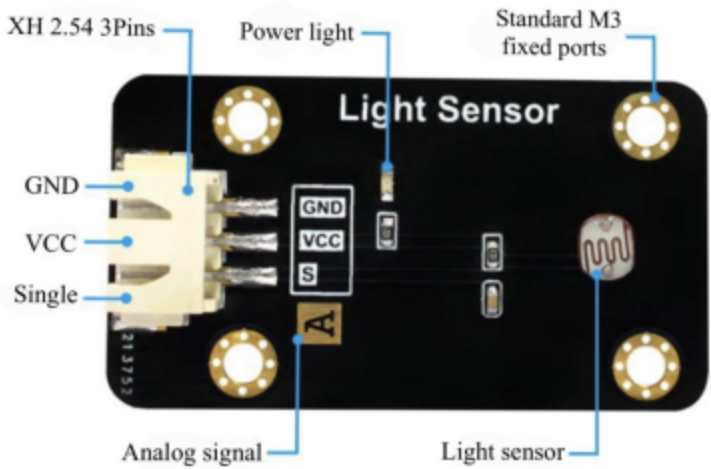
3) Find the 'Serial number' and copy it to the 'serial device' in minicom (Replace the serial number behind 'usbmodem' and press 'return key')



if can't paste it in to the minicom terminal, enter ls /dev/*tty* in the terminal and find the serial number, then copy the whole line including the prefix into 'serial device' in minicom and replace all

5.Example operation

Let's use the Light Sensor in this example.



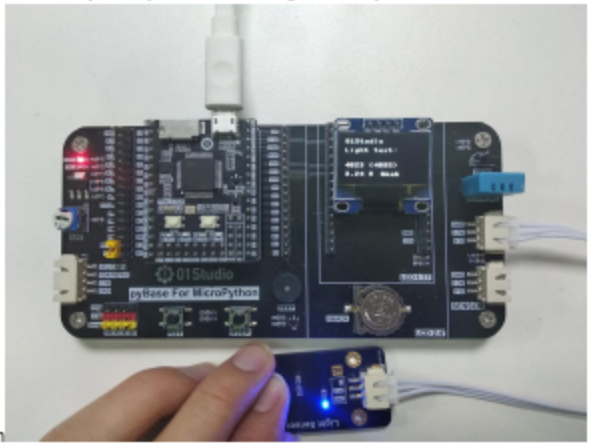
Light sensor module

Function parameter

Service Voltage	3.3-5V
Working Current	<20mA
Working temperature	-20˚C to 85˚C
Interface definition	XH 2.54 interface(3Pin) [GND, VCC, Single]
Output signal	Analog signal: 0-3.3V
Module size	4.5*2.5 cm

1)Connect pyboard with PC
2)find the "main.py" file in pyboard, and replace it with a new "main.py" corresponding to the light sensor, which can be found from Google Drive link below:
https://drive.google.com/file/d/1hOthhL8d0vQ-z2yuVka8tOqUZfC15q3Z/view?usp=sharing
(example>Light Sensor>main.py)
3)the file "ssd1306.py" in the folder also should be moved into the pyboard (example>Light Sensor>ssd1306.py)
4)connect OLED screen with pyboard
5)connect Light Sensor with port "Y11"

6)press "RST" key, then you can see the light intensity information on the screen



Made by Chen Derun, Liu Jinbo & Zhang Wengyu

MicroPython Using Report