

Run and download the program

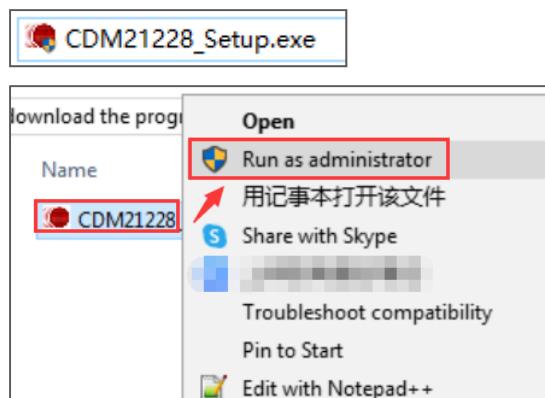
1. Install USB to serial driver (if installed, please ignore)

1.1 Download website: https://cn.dl.sipeed.com/MAIX/tools/ftdi_vcp_driver

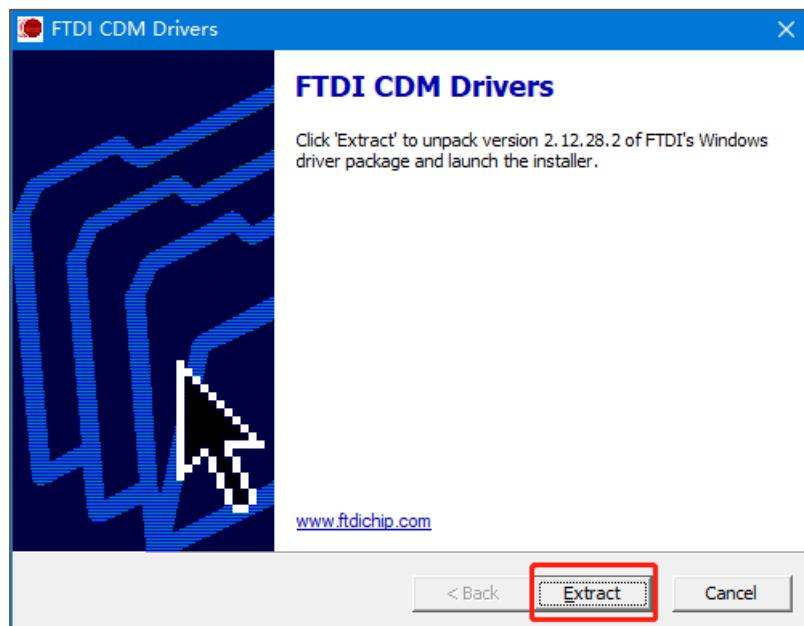
After download is complete, we can get a.zip file. Please extract this zip file and we will get a .exe file, As shown below.



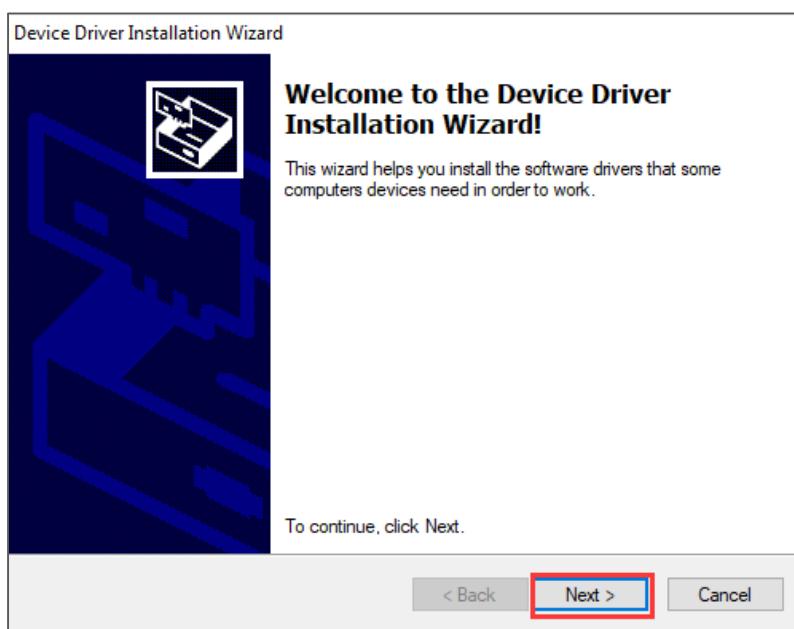
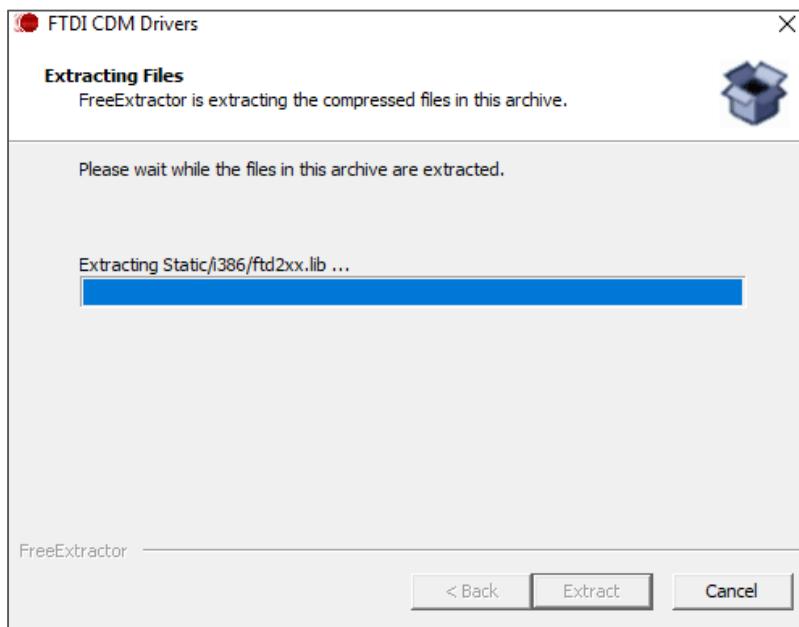
1.2 Right click to run the driver file as an administrator.



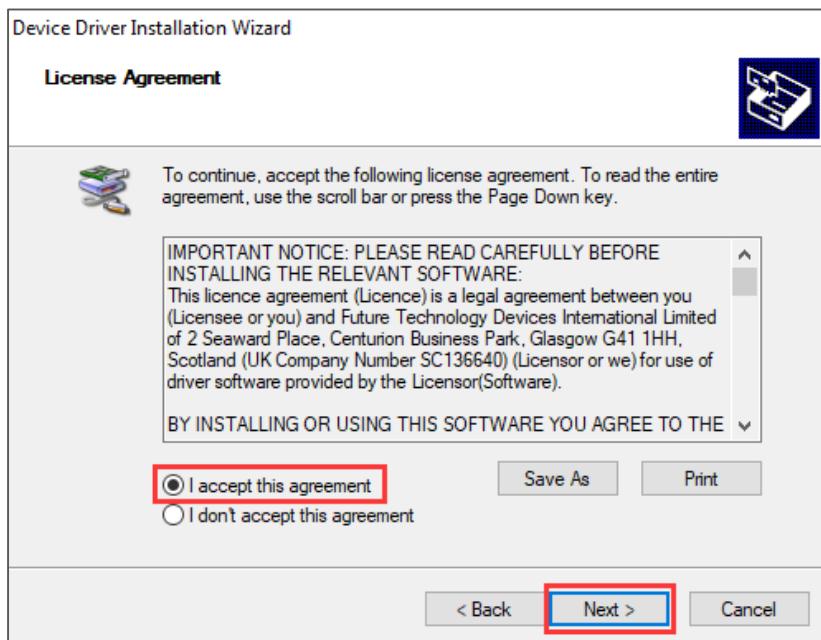
1.3 Click "Extract".



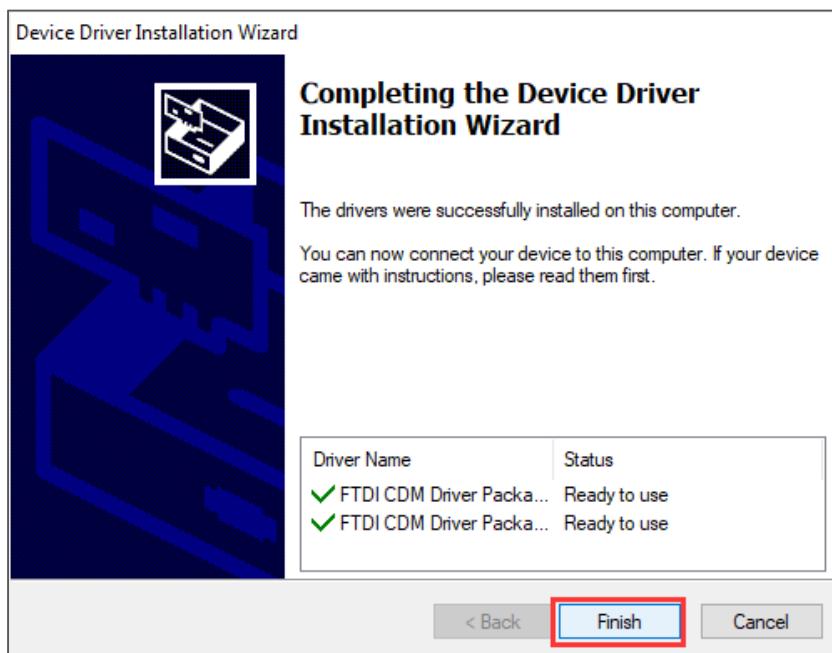
1.4 Wait extract is complete.



1.5 Click "i accept the license" and click "Next".



1.6 Click "Finish".

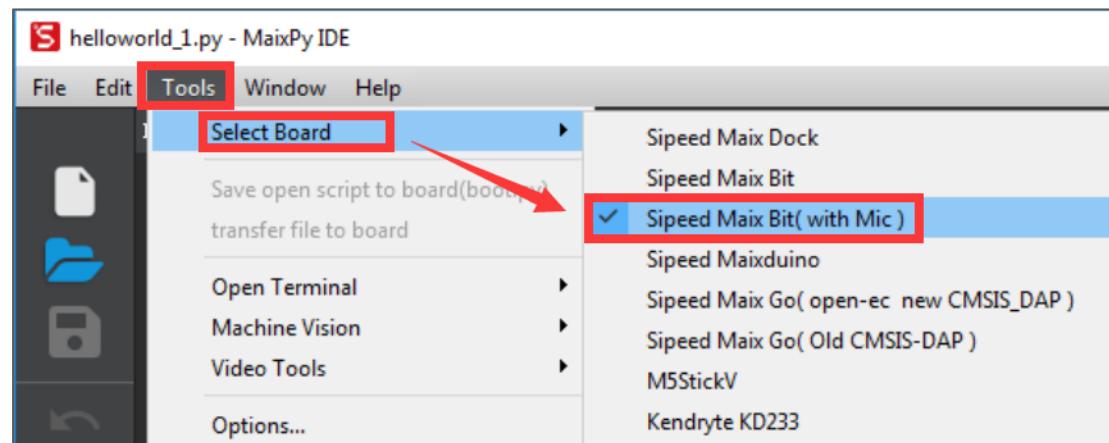


When we connect Maix Go development board to our computer, open the device manager to view the serial number, Maix Go development board has two COM ports.



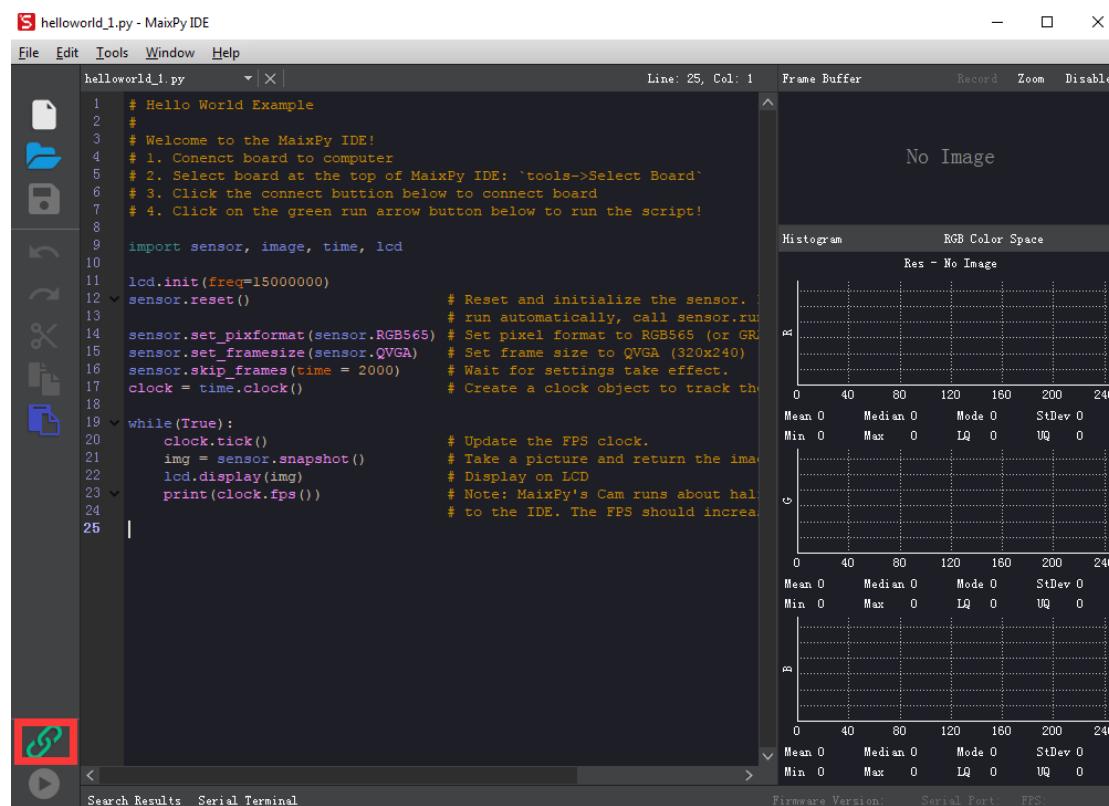
2. Choose development board

Open the MaixPy IDE software, click "Tools" --> "Select Board" --> "Sipeed Maix Bit(with MIC)".



3. Connect Maix Bit

3.1 Open the MaixPy IDE, and click the connect button in the lower left corner.



3.2 2 Select the serial number of Maix Bit and click “OK”. If other serial port software opens the serial port, please close it and try again. The Maix Bit development board has two serial port numbers, choose one first. If it is not possible to connect, we can select the another port.

3.3 After the connection is successful, the connect button turns red, the run button turns green, and the version information is displayed on the right.

The screenshot shows the MaixPy IDE interface. On the left is a code editor with the file `helloworld_1.py` containing a simple script to initialize a sensor and display images. On the right is a camera preview window titled "直方图" (Histogram) and "RGB色彩空间" (RGB Color Space). The histogram shows three empty plots for R, G, and B channels. Below the preview is a status bar with the text "固件版本: 0.5.0" (Firmware Version: 0.5.0), "串行端口: COM68" (Serial Port: COM68), and "FPS: 0". A red box highlights the "串行端口" field.

```

1 # Hello World Example
2 #
3 # Welcome to the MaixPy IDE!
4 # 1. Connect board to computer
5 # 2. Select board at the top of MaixPy IDE: 'tools->Select Board'
6 # 3. Click the connect button below to connect board
7 # 4. Click on the green run arrow button below to run the script!
8
9 import sensor, image, time, lcd
10
11 lcd.init(freq=15000000)
12 sensor.reset()           # Reset and initialize the sensor.
13 sensor.set_pixformat(sensor.RGB565) # Set pixel format to RGB565 (or G
14 sensor.set_framesize(sensor.QVGA)   # Set frame size to QVGA (320x240)
15 sensor.skip_frames(time = 2000)    # Wait for settings take effect.
16 clock = time.clock()           # Create a clock object to track t
17
18 while(True):
19     clock.tick()               # Update the FPS clock.
20     img = sensor.snapshot()    # Take a picture and return the im
21     lcd.display(img)          # Display on LCD
22     print(clock.fps())        # Note: MaixPy's Cam runs about ha
23
24
25

```

4. Running program

4.1 Click the green run button in the lower left corner.

The screenshot shows the MaixPy IDE interface after running the script. The green play button in the bottom-left corner has turned red, indicating the connection is lost. The camera preview window still shows the histogram and color space analysis. The status bar now displays "串行端口: COM63" (Serial Port: COM63) instead of COM68. A red box highlights the "串行端口" field.

```

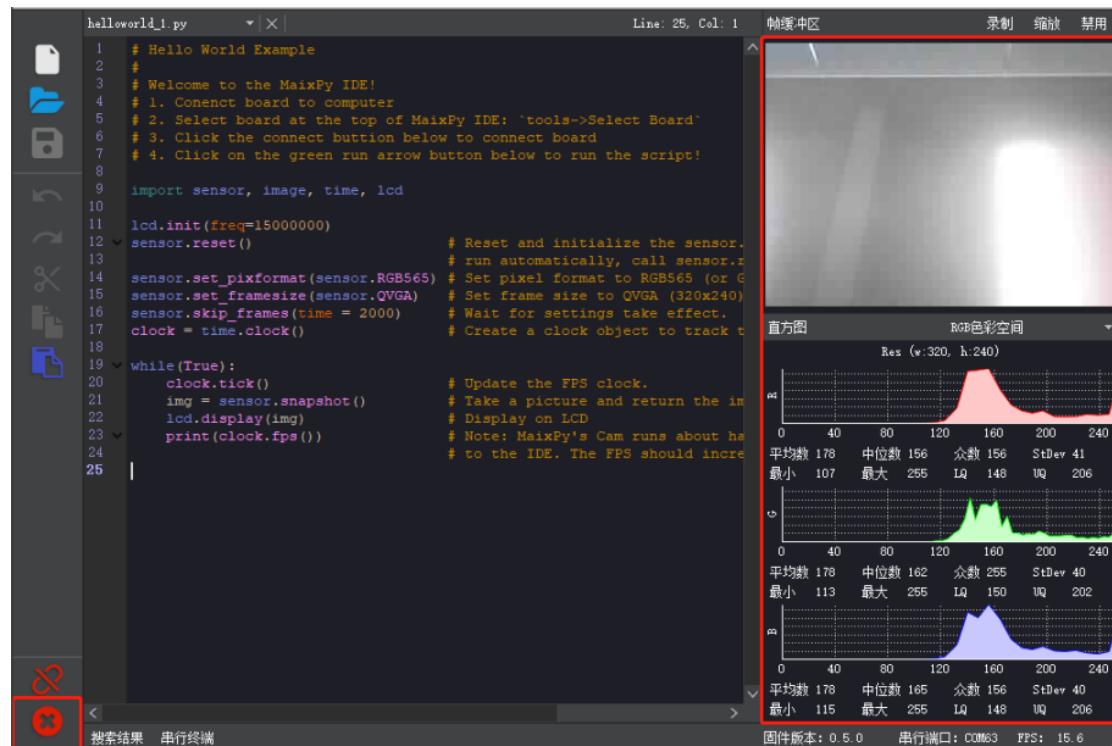
1 # Hello World Example
2 #
3 # Welcome to the MaixPy IDE!
4 # 1. Connect board to computer
5 # 2. Select board at the top of MaixPy IDE: 'tools->Select Board'
6 # 3. Click the connect button below to connect board
7 # 4. Click on the green run arrow button below to run the script!
8
9 import sensor, image, time, lcd
10
11 lcd.init(freq=15000000)
12 sensor.reset()           # Reset and initialize the sensor.
13 sensor.set_pixformat(sensor.RGB565) # Set pixel format to RGB565 (or G
14 sensor.set_framesize(sensor.QVGA)   # Set frame size to QVGA (320x240)
15 sensor.skip_frames(time = 2000)    # Wait for settings take effect.
16 clock = time.clock()           # Create a clock object to track t
17
18 while(True):
19     clock.tick()               # Update the FPS clock.
20     img = sensor.snapshot()    # Take a picture and return the im
21     lcd.display(img)          # Display on LCD
22     print(clock.fps())        # Note: MaixPy's Cam runs about ha
23
24
25

```

4.2 We can see that the green play button on the left become a red stop button, and the image and RGB color space collected by the camera can be displayed on the right,

which indicates that the program be run successfully.

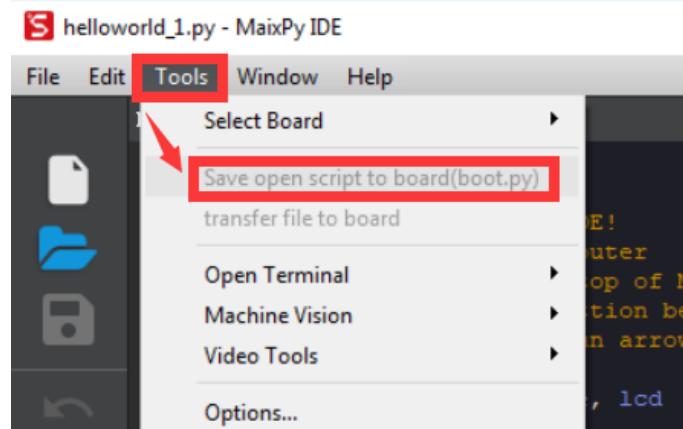
4.3 When we want to stop the current program, click the red stop button in the lower left corner.



5. Download program

The running program described above is only in the SRAM of the chip. After the chip is restarted, the program will not be saved. When we want the program to start automatically, we need to flash the program into flash.

5.1 Click “Tools”--> “Save open script to board(boot.py)”.

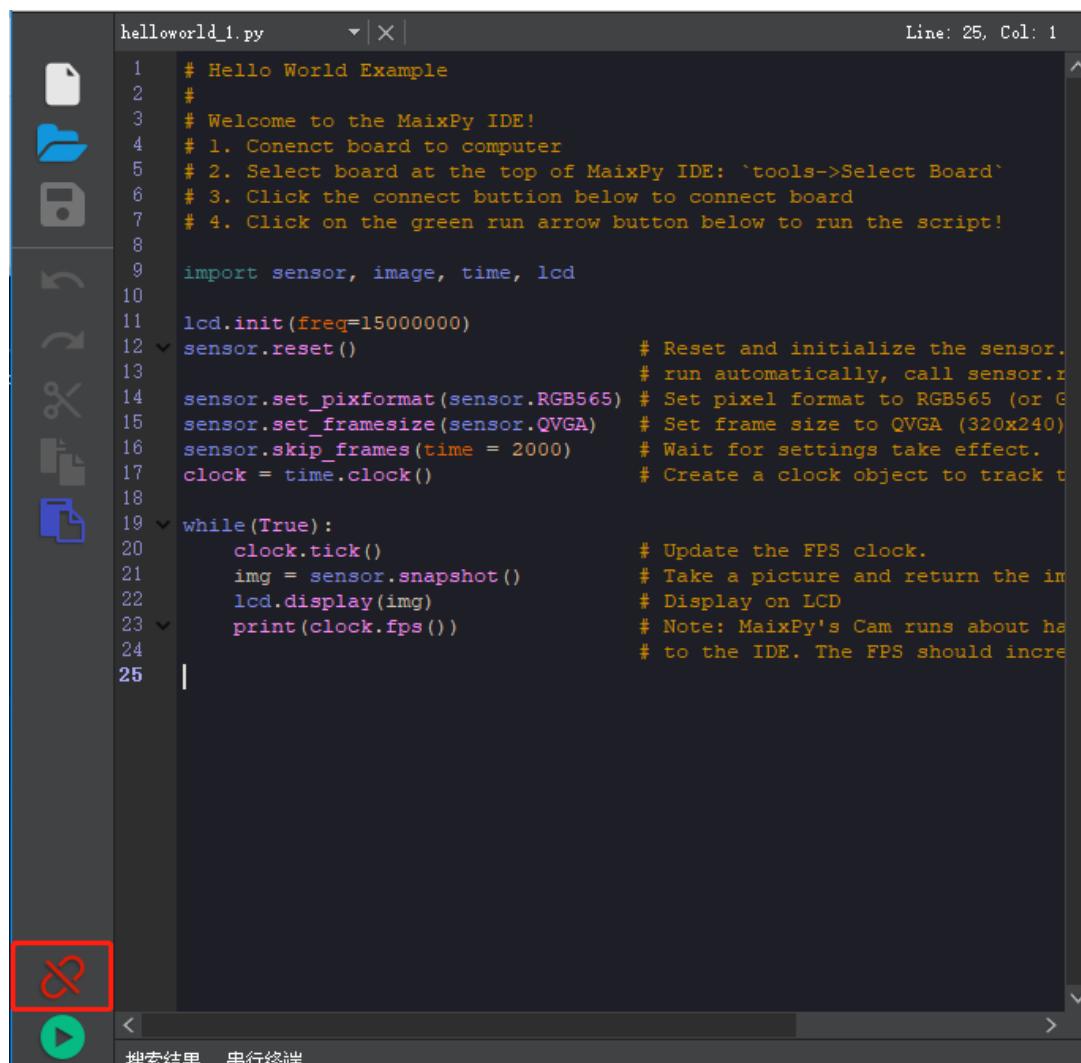


5.2 If the “Remove comments and convert spaces to tabs” prompt appears,,we need to click “Yes”. And save script.

5.3 The program will run automatically after restarting.

6. Disconnect

Click the red connect button in the lower left corner.



```
helloworld_1.py      ▾ | X | Line: 25, Col: 1
1 # Hello World Example
2 #
3 # Welcome to the MaixPy IDE!
4 # 1. Conenct board to computer
5 # 2. Select board at the top of MaixPy IDE: `tools->Select Board`
6 # 3. Click the connect button below to connect board
7 # 4. Click on the green run arrow button below to run the script!
8
9 import sensor, image, time, lcd
10
11 lcd.init(freq=15000000)
12 sensor.reset()                      # Reset and initialize the sensor.
13                                     # run automatically, call sensor.r
14 sensor.set_pixformat(sensor.RGB565)  # Set pixel format to RGB565 (or G
15 sensor.set_framesize(sensor.QVGA)     # Set frame size to QVGA (320x240)
16 sensor.skip_frames(time = 2000)       # Wait for settings take effect.
17 clock = time.clock()                # Create a clock object to track t
18
19 while(True):
20     clock.tick()                   # Update the FPS clock.
21     img = sensor.snapshot()        # Take a picture and return the im
22     lcd.display(img)              # Display on LCD
23     print(clock.fps())            # Note: MaixPy's Cam runs about ha
24                                     # to the IDE. The FPS should incre
25
```