



MU Vision Sensor III Programming Guide

For MicroPython



V0.8.1

2019/06/29

Table of Contents

| | |
|--|----|
| Disclaimer & Copyright..... | 2 |
| 1 Preparations before Coding..... | 3 |
| 2 Connect MU Vision Sensor to Micro:bit..... | 4 |
| 3 Usage of APIs..... | 5 |
| 3.1 Initialize MU Vision Sensor..... | 5 |
| 3.2 Enable Algorithms..... | 6 |
| 3.3 Set Performance Level..... | 7 |
| 3.4 Enable High FPS Mode..... | 8 |
| 3.5 Set White Balance Mode..... | 8 |
| 3.6 Set Digital Zoom Ratio..... | 9 |
| 3.7 Restore Default Settings..... | 10 |
| 3.8 Restart..... | 10 |
| 3.9 LED Settings..... | 10 |
| 3.10 Get Results of Detections..... | 11 |
| Technical Support..... | 13 |
| Revision..... | 14 |

Disclaimer & Copyright

- The information in this manual applies to the MU Vision Sensor III produced by Morpx Inc. Please check the Morpx Inc's website <http://www.morpx.com> for the latest version of the firmware and library functions. Updates are subject to change without notice.
- Please read this manual carefully before using MU Vision Sensor and make sure you understand it, incorrect operation may cause the device to stop working , worse detection results, or even damaging the device.
- Morpx Inc will not warrant the damage caused by unauthorized repair or modification of electronic components on the product.
- The technical solution, vision algorithms, communication protocol mentioned in this manual is developed by Morpx and protected by intellectual property rights. No organization or individual may copy or plagiarize the technical achievements of Morpx Inc. In case of any infringement, Morpx will take legal actions to protect its rights.
- Morpx is the registered trademark of Morpx.Inc., and MU is the registered trademark of MU Vision Sensor. All trademarks (names and patterns) presented here in the text or figures belong to the holders of the marks.

1. Preparations before Coding

If you want to use Micro:bit to control the MU Vision Sensor, you need to import the MicroPython firmware that contains the 'MuVisionSensor' module. Please follow the steps below:

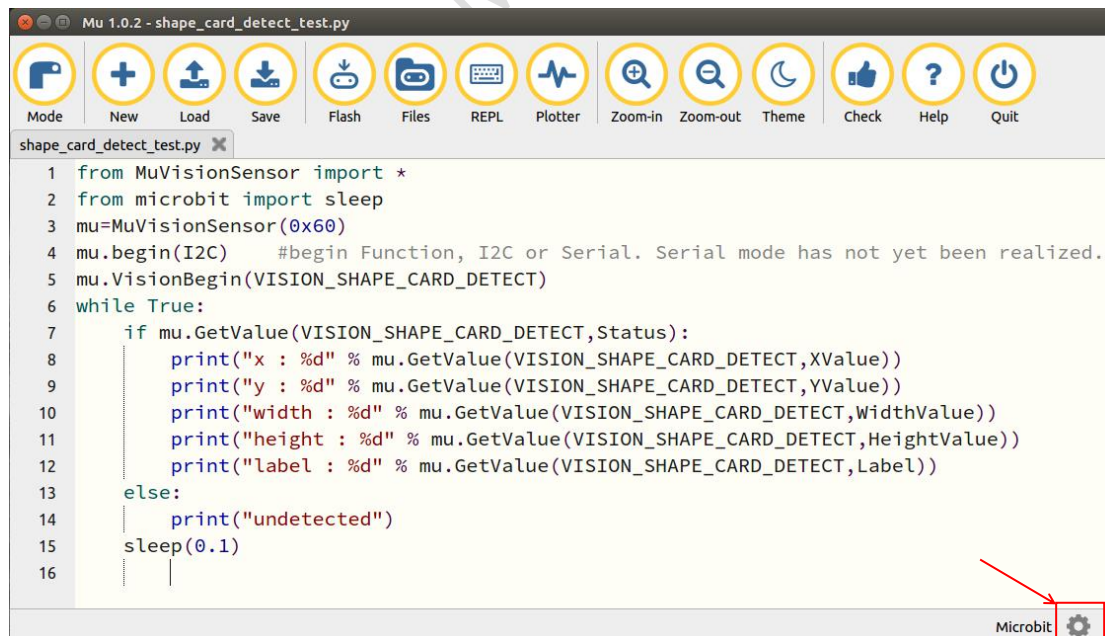
(1) Download the firmware:

GitHub: <https://github.com/mu-opensource/MuVisionSensorIII-MicroPython>

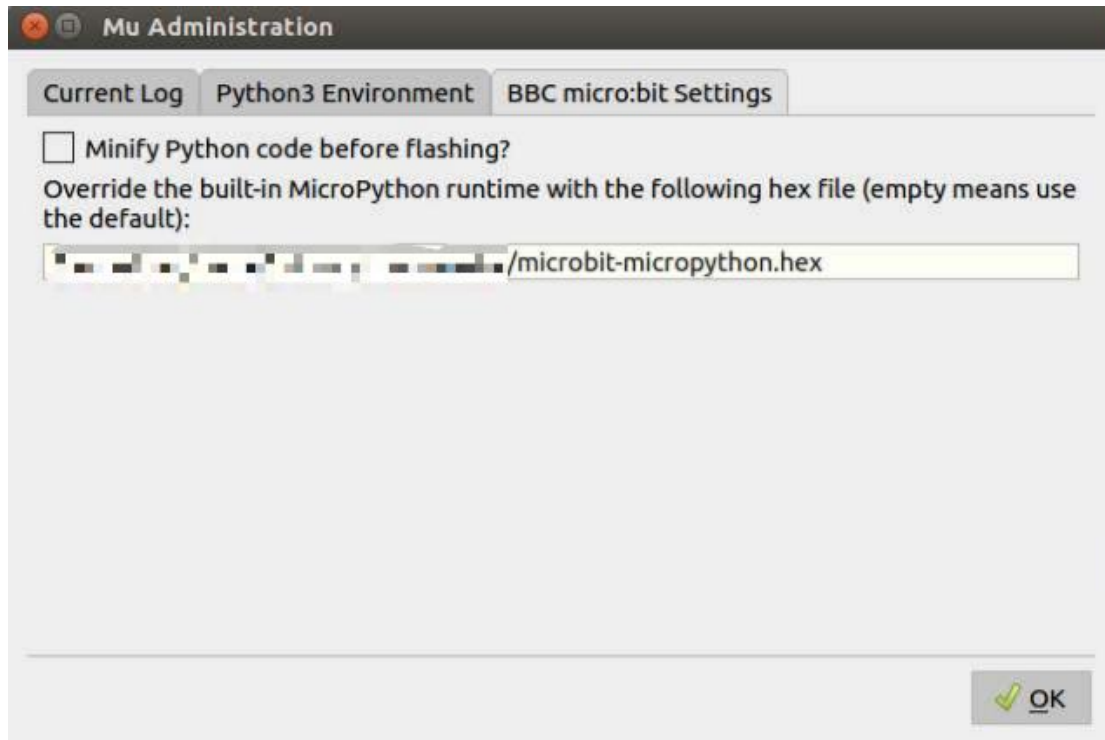
Official Website: <http://mai.morpx.com/page.php?a=sensor-support>

(2) Mu Editor is a simple Python code editor for beginner programmers with friendly GUI ([Learn more about Mu](#)).

Download, install and run Mu Editor, then open the setting menu in the lower right corner



(3) Switch to the BBC Micro:bit setting Tab, input the store path of 'microbit-micropython.hex', then save your setting.



(4) Import module

Type the code below and press enter to run

```
>>>from MuVisionSensor import *
```

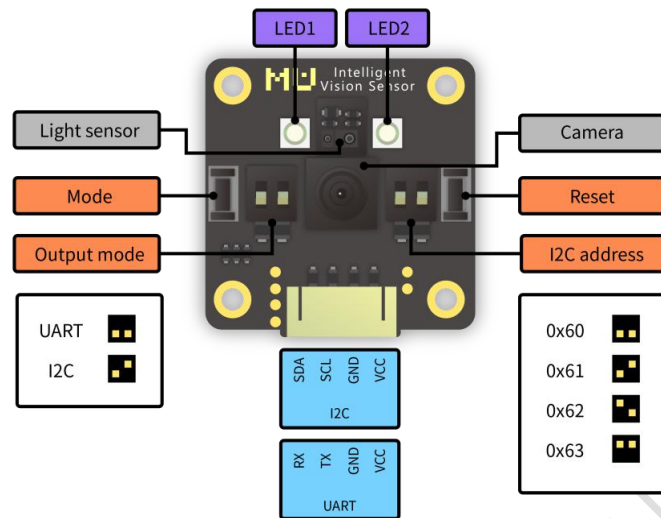
Now you can call all the public APIs in 'MuVisionSensor'

**Key words auto-completion fuction is only available in REPL mode*

```
BBC micro:bit REPL

MicroPython v1.9.2-34-gd64154c73 on 2017-09-01; micro:bit v1.0.1 with nRF51822
Type "help()" for more information.
>>>
>>> from MuVisionSensor import *
```

2. Connect MU Vision Sensor to Micro:bit



I2C Mode(*only this mode works)

(1) Output Mode DIP Switch: set switch 1 downwards and switch 2 upwards

(2) Connect the output Pin1(SDA) to the Pin20 of Micro:bit, and Pin2(SCL) to Pin19 of Micro:bit. Also connecting the ground pin and 3.3v power pin to micro:bit.

(3) Change the I2C address of MU Vision Sensor by resetting Address DIP Switch.

In default both switches are downward and the address is 0x60.(Changing this setting is not recommended)

3. Usage of APIs

3.1 Initialize MU Vision Sensor

Two steps to initialize this sensor:

Step1. Call the 'MuVisionSensor('address')' to create an object , the value of 'address' should be consistent with the setting of Address DIP Switch (default is 0x60);

Step2. Call the 'begin()' function to start this sensor;

1. `#example:`
2. `from MuVisionSensor import *`
3. `mu = MuVisonSensor(0x60) #create an object,`
`with address between 0x60~0x63`
4. `mu.begin() #finish the initialize`

3.2 Enable Algorithms

API: `MuVisionSensor.VisionBegin('vision_type')`

All available 'vision_type's as follows:

- `VISION_COLOR_DETECT`
- `VISION_COLOR_RECOGNITION`
- `VISION_BALL_DETECT`
- `VISION_BODY_DETECT`
- `VISION_SHAPE_CARD_DETECT`
- `VISION_TRAFFIC_CARD_DETECT`
- `VISION_NUM_CARD_DETECT`
- `VISION_ALL`

1. `#example:`
2. `from MuVisionSensor import *`
3. `... #initialize`
4. `# Enable the Color Detect Algorithm`
5. `mu.VisionBegin(VISION_COLOR_DETECT)`

6. `# Enable the Shape Card Detect&Ball Detect Algorithm at the same time`
7. `mu.VisionBegin(VISION_SHAPE_CARD_DETECT | VISION_BALL_DETECT)`

3.3 Set Performance Level

API: `MuVisionSensor.VisionSetLevel('vision_type', 'level')`

'level' can be set to:

- 'LevelDefault ' Default
- 'LevelSpeed ' 'Speed' prior
- 'LevelBalance' Balanced
- 'LevelAccuracy' 'Accuracy' prior

1. `#example:`
2. `from MuVisionSensor import *`
3. `... #initialize`
4. `# Set the performance level of the ball detection algorithm to speed priority`
5. `mu.VisionSetLevel(VISION_BALL_DETECT, LevelSpeed)`

Get Performance Level

API: `MuVisionSensor.VisionSetLevel('vision_type')`

The return value is between 0~3, which represents the 4 levels

1. `#example:`
2. `from MuVisionSensor import *`
3. `... #initialize`


```
4. if mu.VisionSetLevel(VISION_BALL_DETECT) is LevelSpeed:
5.     print("Vision Level of Ball Recognition Algorithm is Speed priority")
```

3.4 Enable High FPS Mode

API:MuVisionSensor.CameraSetFPS('mode')

'mode' can be set to:

- FPSNormal
- FPSHigh

```
1. #example:
2. from MuVisionSensor import *
3. ... #initialize
4. # Enable High FPS mode
5. mu.CameraSetFPS(FPSHigh)
```

Get FPS Mode

API:MuVisionSensor.CameraGetFPS()

return '0'(FPSNormal) or '1'(FPSHigh)

3.5 Set White Balance Mode

*(*Adjust the color cast caused by the changes of external light sources)*

API:MuVisionSensor.CameraSetAwb('mode')

'mode' can be set to:

- AutoWhiteBalance
- LockWhiteBalance

- WhiteLight
- YellowLight

Get White Balance mode

API:MuVisionSensor.CameraGetAwb()

The return value is between 0~3,which represents the 4 WB modes

```
1. #example:
2. from MuVisionSensor import *
3. ... #initialize
4. if mu.CameraGetAwb() is AutoWhiteBalance:
5.     print("Current WB mode is set to 'AutoWhiteBalance'")
```

3.6 Set Digital Zoom Ratio

API:MuVisionSensor.CameraSetZoom('mode')

'mode' can be set to:

- ZoomDefault
- Zoom1
- Zoom2
- Zoom3
- Zoom4
- Zoom5

Get Digital Zoom Ratio Setting

API:MuVisionSensor.CameraGetZoom()

The return value is between 0~5, which represents the 6 zoom levels

3.7 Restore Default Settings

API:MuVisionSensor.SensorSetDefault()

3.8 Restart

API:MuVisionSensor.SensorSetRestart()

3.9 LED Settings

API:MuVisionSensor.LedSetColor('led', 'detected_color', 'undetected_color',
'level')

Explanations of these parameters:

'led': the LED you want to configure, the available values

- LED1
- LED2

'detected_color': colors which are detected, the available values as follows

- LedClose
- LedRed
- LedGreen
- LedYellow
- LedBlue
- LedPurple
- LedCyan
- LedWhite

'undetected_color': colors which are not detected, same available values as
'detected_color'

'level': set the brightness level; an integer between 0 and 15; the larger the
brighter;

3.10 Get Results of Detection

API: `MuVisionSensor.GetValue('vision_type', 'object_inf')`

The available values of 'vision_type' are as mentioned above.

'object_inf' can be set to:

- Status Detection status, '0' means non-detected, '1' means detected
- XValue x position of the object
- YValue y position of the object
- WidthValue width of the object
- HeightValue height of the object
- Label label of the object
- RValue value of Red channel(color recognition mode)
- GValue value of Green channel(color recognition mode)
- BValue value of Blue channel (color recognition mode)

```
1. #example:
2. from MuVisionSensor import *
3. ... #initialize
4. # Shape card detect
5. while True:
```

```
6.     if mu.GetValue(VISION_SHAPE_CARD_DETECT,Status):
7.         print("x: %d" % mu.GetValue(VISION_SHAPE_CARD_DETECT,XValue))
8.         print("y: %d" % mu.GetValue(VISION_SHAPE_CARD_DETECT,YValue))
9.         print("width: %d" % mu.GetValue(VISION_SHAPE_CARD_DETECT,WidthValue))
10.        print("height: %d" % mu.GetValue(VISION_SHAPE_CARD_DETECT,HeightValue))

11.        print("Label: %d" % mu.GetValue(VISION_SHAPE_CARD_DETECT,Label))
12.    else:
13.        print("undetected")

14. # color recognize

15. while True:
16.     if mu.GetValue(VISION_COLOR_RECOGNITION,Status):
17.         print("R: %d" % mu.GetValue(VISION_COLOR_RECOGNITION,RValue))
18.         print("G: %d" % mu.GetValue(VISION_COLOR_RECOGNITION,GValue))
19.         print("B: %d" % mu.GetValue(VISION_COLOR_RECOGNITION,BValue))
```

Morpx

Technical Support

Thanks for purchasing MU Vision Sensor ,and we would like to provide you with regular update services, please check our website: www.morpx.com. Updates are subject to change without notice.You can get the latest technical information from the following URLs:

GitHub: <https://github.com/mu-opensource/>

Official Website: <http://mai.morpx.com/page.php?a=sensor-support>

If there is any problem, you can contact us through telephone +86-571-81958588 or by email: support@morpx.com.

Official Technical Support

Wechat ID



Official Technical Support

QQ ID



Revision

| Date | Version | Release Notes |
|------------|-------------|---|
| 2019-3-28 | V0.0.1-beta | Draft |
| 2019-06-25 | V0.8.0 | 1. Standardized this manual; 2. Changed the firmware downloading address; 3. Replaced the photo of MU Vision Sensor; 4. Added the official technical support information; 5. Added Revision; 6. Corrected some other mistakes; |
| 2019-06-29 | V0.8.1 | Adjusted the bookmarks |
| | | |
| | | |