

# **Vision Sensor Mode Commands**

## **for MU OS**

MorpX Inc.

Version: V 1.1.3

2017-11-20

## Introduction

MU vision sensor module is a versatile sensor designed by Morpx based on advanced computer vision technology. A developer can communicate with MU using UART or serial USB, setting parameters and streaming detection results. This is intended for connecting with Arduino, Raspberry Pi, bluetooth module, cell phone or PC. Additionally, one can make use of the two sets of servo connectors to create a 2-degree-of-freedom vision sensor platform. These standard servo connectors output PWM signals and can be integrated into other robotic platforms. All these interfaces make it easier to build complex robots.

MU vision sensor contains 3 modes: running mode, setup mode and firmware update mode.

**Running mode:** Normal working mode of the sensor when power on. LED light will show red or blue. red means no object detected, and blue means detected.

**Setup mode:** Connect MU with PC through a micro USB cable, Short press mode button to enter setup mode. In this mode, LED shows purple, a COM device is opened and settings of the sensor can be changed. Use a serial port software to send commands. This mode is also available when send "CMD+SENSOR\_SETUP\n" directly through RX port on MU. Check Command Set for more information.

**Firmware update mode:** Press and hold mode button and press reset button in the mean time to enter firmware update mode. LED shows green. This mode is used for firmware update. You can query the firmware version by sending command CMD+SENSOR\_VERSION. Visit our firmware update page to get latest firmware. <http://mu.morpx.com/firmware/>

## Command Set

### General Command Format

A MU command is a text string used to set a parameter for a menu item, in this format:

**CMD+AA\_BB=CC**

**CMD:** Command Marker. Currently only "CMD" is supported.

**AA:** Menu item.

**BB:** Sub menu item. (optional)

**CC:** Parameter

Note 1: All character must be in UPPER CASE.

Note 2: Every command must be followed by "CR LF" at the end, e.g. 0x0D 0x0A

Note 3: a single question marker "?" in the place of the parameter CC can be used to query the current setting of this parameter.

Note 4: Receive "OK" when change successfully, or receive "ERROR" when failed.

## SENSOR Menu

**CMD+SENSOR\_SETUP**

**Parameter:** None

**Description:** Enter “Setup” mode  
Receive “Setup” when in setup mode.

### **CMD+SENSOR\_HELP**

**Parameter:** None

**Description:** Display help information.

### **CMD+SENSOR\_STATUS**

**Parameter:** None

**Description:** Query the current parameters of all the menu items.

### **CMD+SENSOR\_RESET**

**Parameter:** None

**Description:** Restore factory defaults.

### **CMD+SENSOR\_VERSION**

**Parameter:** None

**Description:** Query the version information.

### **CMD+SENSOR\_SAVE**

**Parameter:** None

**Description:** Save the parameters.

### **CMD+SENSOR\_EXIT**

**Parameter:** None

**Description:** Exit the Setup mode and change into running mode.

You should save the parameters before exit, otherwise the changed settings are invalid.

## **VISION Menu**

### **CMD+VISION\_TYPE=XXXX**

**Parameter:** BALL, LINE, BODY, FACE, MOVINGOBJECT, ?

**Description:** Select or query the vision detection algorithm. The available detection algorithms are: ball, line, upper body, face and moving object.

**Default:** BODY

### **CMD+VISION\_LEVEL=XXXX**

**Parameter:** 1, 2, 3, 4, 5, ?

**Description:** Select or query the performance index of algorithm. Higher level means higher detection accuracy and lower false alarm rate. But detection speed will be slower, and missing report rate increases. Conversely, the lower the number, the lower the detection accuracy, the faster the detection speed.

This setting only works in body and ball detection algorithm at present.

**Default:** 1

### **CMD+VISION\_ZOOM=XXXX**

**Parameter:** 1, 2, 3, 4, 5, ?

**Description:** Select or query the vision field of the camera. Higher level means detect farther objects, and narrow horizontal field of view. Conversely, the lower the level, the wider the horizontal field of view, the closer the detection range.

When setting to face detection, zoom will be automatically changed to level 3.

**Default:** 1

### **CMD+VISION\_STATUS=XXXX**

**Parameter:** ENABLE, DISABLE, ?

**Description:** Enable, disable, or query the status of the vision detection functions. By default, vision detection is "ENABLE"

**Default:** ENABLE

## **USB Menu**

### **CMD+USB\_STATUS=XXXX**

**Parameter:** ENABLE, DISABLE, 1, 0, ?

**Description:** Enable, disable or query the status of serial USB output of the vision detection results, (For details please see Appendix 1).

**Default:** DISABLE

## **UART Menu**

### **CMD+UART\_STATUS=XXXX**

**Parameter:** ENABLE, DISABLE, 1, 0, ?

**Description:** Enable, disable or query the status of UART output of the vision detection results, (For details please see Appendix 1).

**Default:** ENABLE

### **CMD+UART\_BAUD=XXXX**

**Parameter:** 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, ?

**Description:** Set or query the UART baud rate.

Other parameters of UART are fixed , data bits: 8 bits ,1 end bit, no check bit.

**Default:** 115200

## **SERVOX Menu**

### **CMD+SERVOX\_STATUS=XXXX**

**Parameter:** ENABLE, DISABLE, ?

**Description:** Enable, disable or query the status of Servo X (horizontal moving servo)

**Default:** DISABLE

### **CMD+SERVOX\_STEP=XXXX**

**Parameter:** integer in [-100, 100], ?

**Description:** Set or query the step size of SERVO X, [-100, 100] map to multiplier of [-10, 10]. When the parameter is negative, the servo will turn into the reverse direction. For example, 15 maps to a speed multiplier of 1.5. -20 maps to a speed of 2 in the reverse direction.

**Default:** 10 (which means multiplier 1)

**CMD+SERVOX\_INITANGLE=XXXX**

**Parameter:** integer in [0, 180], ?

**Description:** Set or query the initial angle of SERVO X

**Default:** 90

**CMD+SERVOX\_REVERSE=XXXX**

**Parameter:** ENABLE, DISABLE, ?

**Description:** Enable, disable or query if SERVO X is set to the reverse rotation.

**Default:** DISABLE

## SERVOY Menu

**CMD+SERVOY\_STATUS=XXXX**

**CMD+SERVOY\_STEP=XXXX**

**CMD+SERVOY\_INITANGLE=XXXX**

**CMD+SERVOY\_REVERSE=XXXX**

**Description:** These four menu items controls the Y direction (Vertical) servo. They have the same parameter meanings and default values as SERVOX menu.

## Appendix 1: Output Format

Once the Vision Sensor is in the running mode, it will keep emitting detection results once per video frame based on the current setting. The Vision Detection Result Output format is a binary format with a length of 8 bytes. Meaning of the output message differs among algorithms .

For **BODY, BALL, FACE**:

Byte	0	1	2	3	4	5	6	7
Content	0xFF	0xFE	Detected	X	Y	Width	Height	0xED

**0xFF**: Header byte 1

**0xFE**: Header byte 2

**Detected**: Whether the detection target is found in the current frame, 1: detected, 0: not detected

**X**: X (horizontal) coordinate of the center of the detection rectangle, in the range of [0, 100].

**Y**: Y (vertical) coordinate of the center of the detection rectangle, in the range of [0, 100].

**Width**: The width of the detection rectangle, in the range of [0, 100].

**Height**: The height of the detection rectangle, in the range of [0, 100].

**0xED**: End byte

For **LINE**:

Byte	0	1	2	3	4	5	6	7
Content	0xFF	0xFE	Detected	Up-X	Up-Y	Down-X	Down-Y	0xED

**0xFF**: Header byte 1

**0xFE**: Header byte 2

**Detected**: Whether the detection target is found in the current frame, 1: detected, 0: not detected

**Up-X**: X (horizontal) coordinate of the up end of the line, in the range of [0, 100].

**Up-Y**: Y (vertical) coordinate of the up end of the line, in the range of [0, 100].

**Down-X**: X (horizontal) coordinate of the below end of the line, in the range of [0, 100].

**Down-Y**: Y (vertical) coordinate of the below end of the line, in the range of [0, 100].

**0xED**: End byte

For **MOVINGOBJECT**:

Byte	0	1	2	3	4	5	6	7
Content	0xFF	0xFE	Detected	X	Y	NA	NA	0xED

**0xFF**: Header byte 1

**0xFE**: Header byte 2

**Detected**: Whether the detection target is found in the current frame, 1: detected, 0: not detected

**X**: X (horizontal) coordinate of the center of the moving object, in the range of [0, 100].

**Y**: Y (vertical) coordinate of the center of the moving object, in the range of [0, 100].

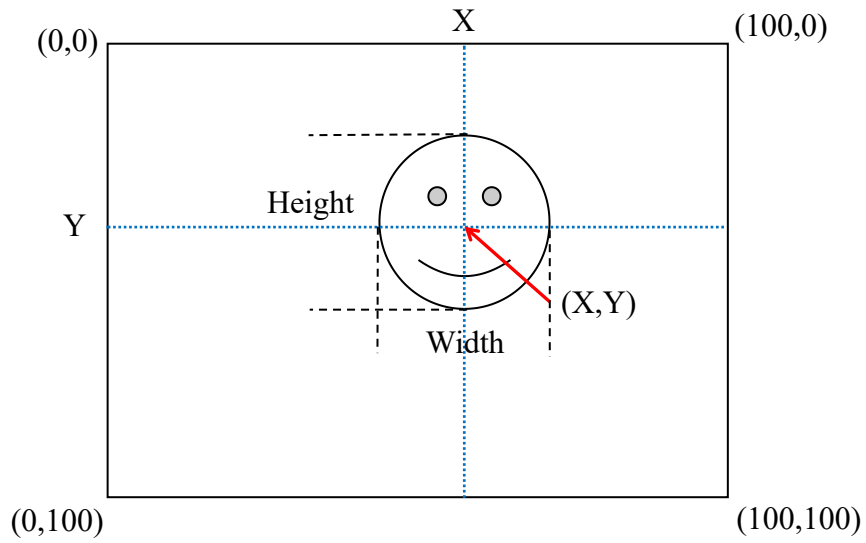
**NA**: none.

**NA**: none.

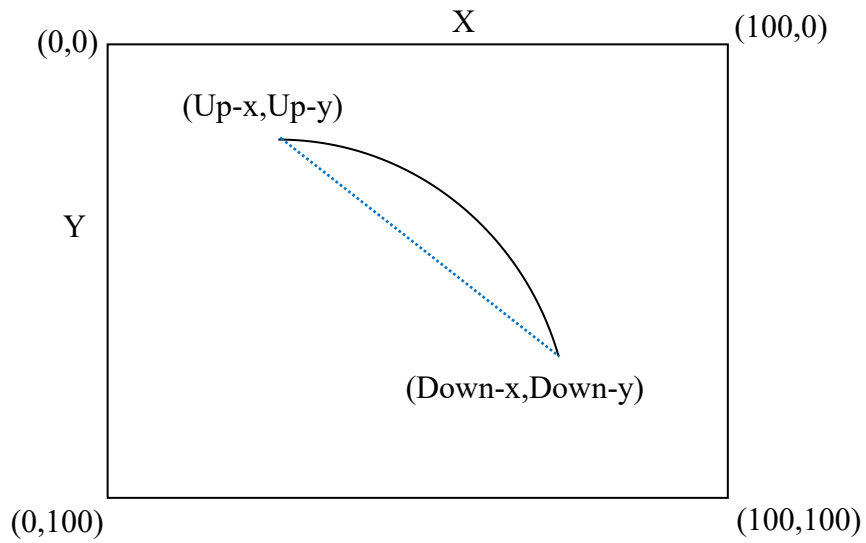
**0xED**: End byte

Note that X, Y, Width, Height use relative coordinates with respect to the width and height of the image frame. As show in the figure.

For **BODY, BALL, FACE, MOVINGOBJECT**:



For **LINE**:



## Appendix 2: A Simple UART Example

This is an example of using UART to communicate between MU and Arduino for an face detection application.

1. Firstly, you have to install a Serial Debugger or Serial Port Terminal (Ubuntu) software, which can be downloaded from the Internet.
2. Short press mode button to switch MU to “Setup Mode”, with the LED showing purple.
3. Connect MU to the computer via an micro USB cable and open the Serial Debugger.
4. Set the Application's parameters as: Baudrate = 115200, Data bits = 8bits, Stop bit = 1, Parity = None, and then, open the serial com port
5. Send the following commands:

CMD+SENSOR\_SETUP

CMD+VISION\_TYPE=FACE

CMD+UART\_STATUS=ENABLE (Default is ENABLE)

CMD+UART\_BAUD=115200 (Default is 115200)

CMD+SENSOR\_SAVE

CMD+SENSOR\_EXIT

After sending command CMD+SENSOR\_EXIT, MU will automatically change to “Runing Mode”. Now the vision program will start running and your Arduino program can start receive detection result from UART.

The setting in Vision Sensor will be saved even after power off.



## Appendix 3: A Simple SERVO Example

This is an example of using cradle head to control MU for an face detection application.

1. Repeat the above steps from 1 to 5
2. Send the following commands:

CMD+SENSOR\_SETUP

CMD+VISION\_TYPE=FACE

CMD+UART\_STATUS=DISABLE

CMD+SERVOX\_STATUS=ENABLE

CMD+SERVOX\_INITANGLE=90

CMD+SERVOX\_REVERSE=1 (Check the direction)

CMD+SERVOY\_STATUS=ENABLE

CMD+SERVOY\_INITANGLE=70 (Set SERVOY a little higher)

CMD+SENSOR\_SAVE

CMD+SENSOR\_EXIT

Connect a cradle head to the IO ports and it will control the MU to follow a ball.