MU Vision Sensor III

Datasheet

MORPX

Version V0.8
2019.06.06

Table of Contents

Disclaimer & Copyright

•This manual is only applicable to the MU Vision Sensor (Model # MUVS-AB2, firmware V311) manufactured by Morpx.Inc. Upcoming firmware updates can improve the performance or introduce new functions, please go to www.morpx.com to get the latest firmware versions. We might not publish notifications for every firmware update.

•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, getting worse detection results, or even damaging the device.

•You are not allowed to repair or modify the MU Vision Sensor without the authorization from Morpx. No warranty will be provided for any damage caused thereby.

•The technical solution, vision algorithms, communication protocols 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.

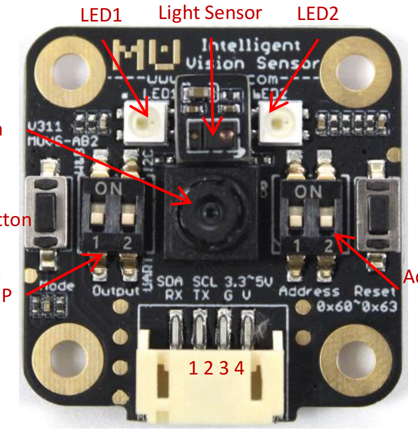
1 Overview

1.1 Product Description

MU Vision Sensor is a device for vision based detection and recognition, designed for the educational robotics and hobbyist kits market. Unlike traditional sensors which usually have one signal for any kind of objects, MU vision sensor's built-in deep learning algorithms can recognize and locate a variety of objects, such as colors, balls, humans and cards. The sensor supports UART or I2C communication modes. The sensor is small, low power, and process information locally without the need of network. It can be widely used in smart toys, AI educational robots, maker kits, digital signages and other application areas.

2 Hardware

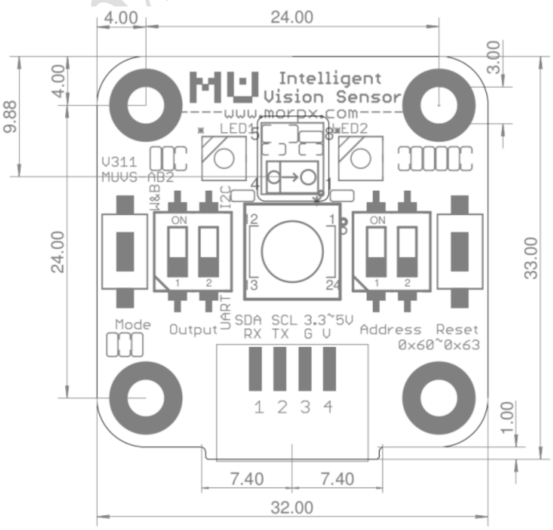
2.1 Appearance



Top View

Bot View

2.2 Dimensions



Top View

	Value
Long	33mm
Width	32mm
Height	11.5mm
Weight	6.8g

2.3 Hardware

(1) Processor

The new generation of MU Vision Sensor adopts the ESP32 module solution, which has a 240MHz dual-core processor and larger memory. Compared with the previous generation, it has faster processing speed and higher recognition rate, and has more vision algorithms. In addition, this module integrates bluetooth and WiFi capabilities. We will provide wireless data transmission functionality in a future firmware release.

(2) Camera

MU Vision Sensor uses a 300k (640*480) pixel RGB camera with a wide-angle lens.

Resolution	640x480
FPS(max)	60fps
Sensor Size	1/4”
Field of View - FOV	84°
Focal Length - f	2.5mm
Aperture - F	2.2mm
Len diameter	5.5mm

User configurable camera settings include CAM_ZOOM, CAM_ROTATE, CAM_FPS, CAM_AWB, details see the register 0x10:CAMERA_CONF1.

Note 1: The sensor lens could get dirty overtime and you can use a soft cloth to wipe it clean in clock-wise motion. Do not use water or other solutions.

Note 2: In the case of the camera module drops off the PCB board, you can remount the sensor to the board by matching the 1, 12, 13, 24 pin numbers to the corresponding numbers on the board. The camera won’t work if it is mounted in reverse.

(3) Light Intensity & Motion Sensor

MU Sensor module uses a highly integrated light intensity & motion sensor to assist the camera parameter adjustment. The sensor can detect light intensity, infra-red proximity, and hand glide gestures.

Note 1: the sensor’s functionality will be open to users in a later firmware update.

(4) Level Conversion IC

This IC is used for level conversion between the sensor and the controller. It is compatible with 3.3v or 5v system.

(5) Output Port

A PH2.0 - 4pin interface is used for transmitting signal to the controller. Pin definitions:

Pin	UART	I2C
1	RX	SDA
2	TX	SCL
3	GND	GND
4	VDD	VDD

(6) Output Mode DIP Switch

A 2-bits DIP switch is used to select the output mode:

Bit1	WiFi	Bit2	Output Mode
0	WiFi On*	0	UART
1	WiFi Off*	1	I2C

Note1: *not available in the current version;
Note2: Switch up to set 1, or switch down to for 0.

(7) Address DIP Switch

A 2-bits DIP switch is used to set the device’s I2C address, this allows you to connect up to 4 MU Vision sensors on a single I2C bus.

Bit1	Bit2	Address
0	0	0x60
0	1	0x61
1	0	0x62
1	1	0x63

Note1: MU will also respond to the 0x00 broadcast message sent from the host;
Note2: Switch up to set 1, or switch down to set 0.

(8) Reset Button

Restart the sensor.

(9) Mode Button

Running Mode: Press and release to toggle between different working modes. Reserved.

Flash Mode: Press and hold the Mode Button and then press and release the Reset Button to enter the flash mode for firmware upgrade.

(10) Programmable RGB LED

This two programmable RGB-LEDs can be used to indicate the working status of MU or be used for filling lights in low light conditions.

(11) Power LED

The White-LED will light up if the power supply is normal.

(12) Status LED

The Blue-LED is used to indicate the wireless status in a future firmware update.

2.4 Electrical Characteristics

Item	Unit	Min	Typical	Max
Input Voltage	V	3.3	5	5.5
Supply Current (1)	mA	500	-	-
UART Standard Mode Current (2)	mA	78	80	83
I2C Standard Mode Current (2)	mA	91	94	95
UART Baud Rate	bps	9600	115200	921600
I2C Speed	Kbps	-	100	400
Operating Temperature	℃	-20	25	70
WiFi frequency	MHz	2412	-	2484

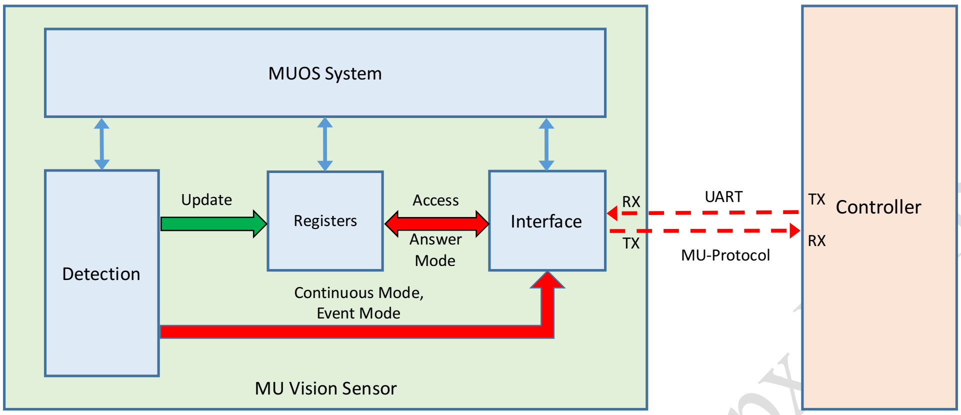
Note 1: When WiFi mode is on, the initialization of the antenna needs at least 500mA current. The current will reduce to normal level after the initialization.
Note 2: These working current are obtained under WiFi off conditions, it also depends on the factors such as the LED brightness, camera sampling fps, and how many vision algorithms are on.

3 Detailed Functionalities

3.1 Communication

MU Vision Sensor exchanges data with the controller through its registers and supports UART and I2C communication modes.

3.1.1 UART Mode



In this mode, you access registers using message exchange based on the MU-Protocol (Please see Sec 4 Mu Protocol for details). Detection results can be output in three ways: response mode, continuous mode and event mode.

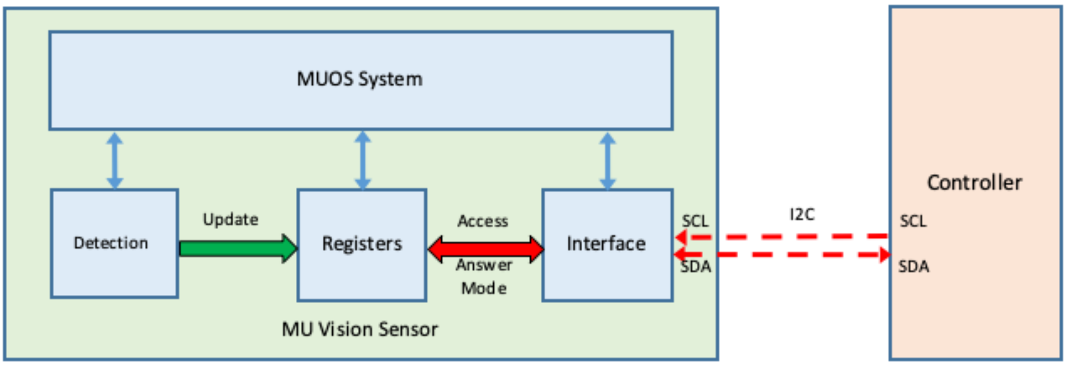
(1) Response Mode: MU returns the latest results message to the controller while it receives Request Message 0x12;

(2) Continuous Mode: MU returns results message to the controller after each frame has been processed;

(3) Event Mode: MU returns result message to the controller when it detects the object, otherwise, no message output.

3.1.2 I2C Mode

This mode uses the standard I2C protocol to read or set registers.



3.2 Operation Instructions

The MU Vision sensor comes with many example codes so you do not have to worry about registers and protocols. To quickly get your project started, please visit https://github.com/mu-opensource/MuVisionSensorIII for the latest example source code. The following sections are for those who want to understand the implementation details of the MU vision sensor.

3.2.1 General Operation Steps

(1) Set device address;
(2) Set output mode;
(3) Check version;
(4) Configure hardware settings;
(5) Configure vision settings;
(6) Read detection results.

3.2.2 Set Device Address

Set the address of MU through the DIP switch, see details at 2.3(7).
Note: Each time you change the address, you will need to restart the sensor by pressing the reset button on the sensor.

3.2.3 Set Output Mode

Set the communication mode through the DIP switch, see details at 2.3(6).
Note: Each time you change the output mode, you need to restart the sensor by pressing the reset button on the sensor.

3.2.4 Check Version

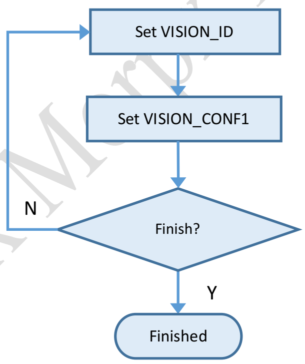
Read the PROTOCOL_VER and FIRMWARE_VER registers to check the firmware version.

3.2.5 Configure Hardware Settings

Set hardware-related registers per the actual application and requirements.

3.2.6 Configure Vision Settings

Algorithm for each VISION_ID has its own configuration registers. Set VISION_ID before setting the algorithm parameters.

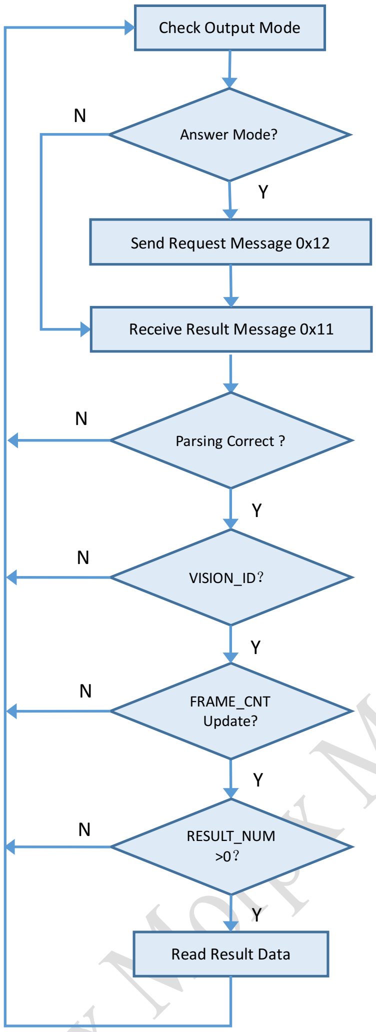


3.2.7 Get Results Data by UART Message

Answer Mode: Send the Request Message – (CMD=0x12) first and wait for the Result Message - 0x11.

Continuous Mode: Waiting for Result Message – (CMD=0x11).

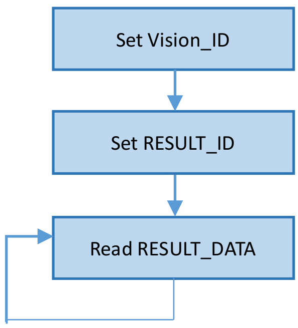
Event Mode: Waiting for Result Message – (CMD=0x11).



3.2.8 Get Results Data via Registers

This is suitable for direct operation of registers in UART or I2C mode. Reading process has multiple different optimized schemes depends on the actual situation.

A simple example:



Here are some registers can be used to improve read efficiency and reliability:

(1) LOCK:

Register read-write security lock

Register Read-Write Security Lock, this is can be used to protect other registers from being overwritten while reading result data;

(2) FRAME_CNT:

Frame Count, this is can be used to check if the result data is from the next frame;

(3) VISION_STATUS:

Vision Algorithm Enabled Status, this is can be used to quickly find out whether an algorithm has been enabled;

(4) VISION_DET:

Vision Detection Status, this is can be used to fast query whether an object has been detected;

(5) RESULT_NUM:

Result Number, this is can be used to query how many detection results can be read;

(6) READ_STATUS:

Results Read Status, this is can be used to query the registers RESULT_DATA1~5 reading status, the bit will be set 1 while new date is updated and be cleared 0 if it has been read, this can avoid duplicate reads or determine whether the results have been updated.

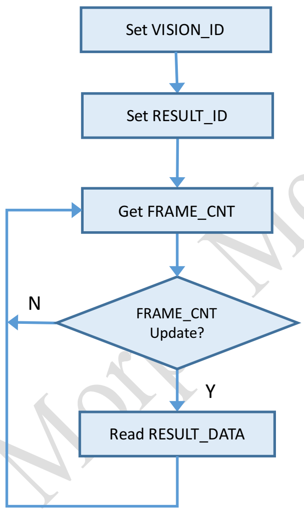
Note1: See the chapter 5 Registers for details of the registers mentioned above;

Note2: The default value of register RESULT_ID is 1. After setting VISION_ID, you can read RESULT_DATA registers directly if the algorithm does not support multi-object detection;

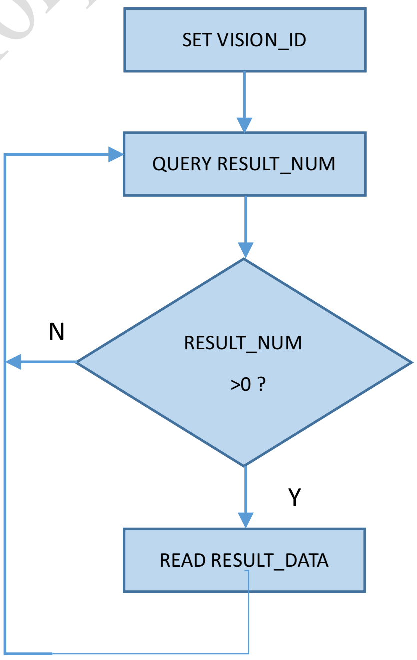
Note3: For the real-time applications, it is more efficient to read directly the RESULT_DATA registers.

3.2.9 Examples

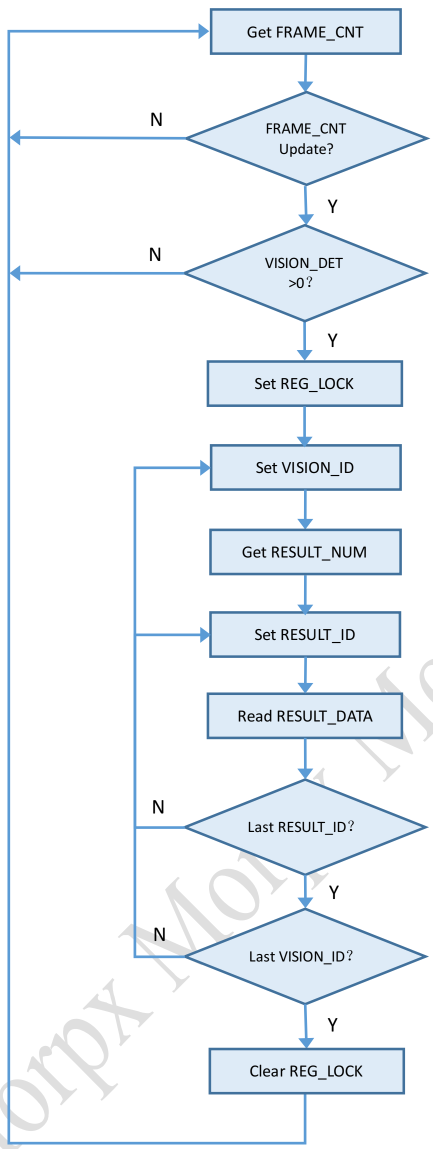
(1) Fast read



(2) Read result only when an object is detected (RESULT_ID defaults to 1)



(3) Read all the results of each algorithm



3.3 Vision Algorithms

MU Vision Sensor III integrates a variety of computer vision algorithms, with built-in embedded deep learning technology that supports different object detection and recognition.

3.3.1 Notes on Vision Recognition

Computer vision is known to be sensitive to the light source, color, background or the moving speed of the objects, etc. The detection results will be affected by different environment settings. To obtain better detection results, one should

(1) Try to avoid using vision sensor in a dark room, under a single direct light or facing a strong backlight (e.g. light from windows);

(2) Do not let the bright light, or strong sun light, directly lit on the objects that you want to detect to avoid the specular highlight;

(3) Avoid using colored lights or fast changing lightings, stable, uniform distributed white light source is the best;

(4) Do not point the camera towards a bright light source;

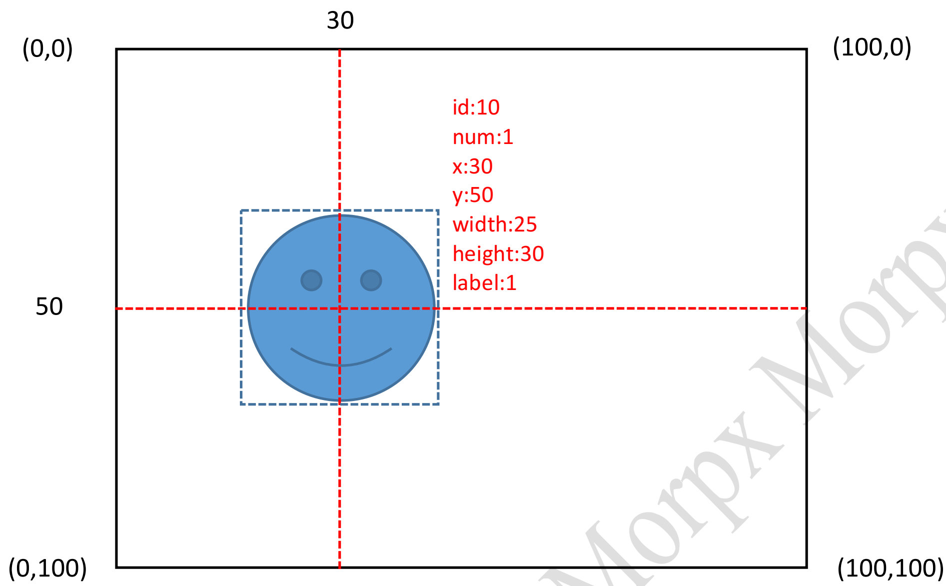
(5) Color sensitive algorithms should not be used against a background with a similar color. For

example, green tennis ball should not be put on the green carpet;

(6) Avoid environments with similar objects. For example, an orange beside the ping-pong ball.

3.3.2 Image Coordinate System and Detection Result Data

In the image coordinate system, the upper left corner is the origin of coordinate (0,0), the ratio of image length to width is 4:3, and the detection results are normalized to the range of 0~100:



Detection Result Data：

（1）id: Vision ID

（2）num: the number of detection results [0 or 1]

（2）x: normalized horizontal center coordinate [0 … 100]

（3）y: normalized vertical center coordinate [0 … 100]

（4）width: the normalized width of the object [0 … 100]

（5）height: the normalized height of the object [0 … 100]

（6）label: classification label number

3.3.3 List of included Vision Algorithms

Vision ID	Algorithm	Object	Result Data					Notes
			data1	data2	data3	data4	data5	
1	Color Detection	Specified color	x	y	width	height	label	Good white light source
2	Color Recognition	Specified area	R	G	B	/	label	Good white light source
3	Ball	Ping-pong ball(orange) Tennis ball(green)	x	y	width	height	label	Simple background, avoid similar objects
4	reserved	/	/	/	/	/	/	/
5	Body	Human upper body	x	y	width	height	/	Head feature need to be included, people facing the camera
6	Shape Card	Check, Cross, Circle, Square, Triangle Cards	x	y	width	height	label	The card should be placed perpendicular to the camera, rotate<15°, inclination<30°
7	Traffic Card	Forward, Left, Right, U Turn, Park Cards	x	y	width	height	label	
8	Number Card	0~9 Number Cards	x	y	width	height	label	

3.3.4 Color Detection, Vision-ID: 1

Description:

This algorithm is used to detect a specified color which is set by user and returns the color region's coordinate and its size.

Object:

Label	Object	Picture	Label	Object	Picture
1	Black (Dark Gray)		2	White (Light Gray)	
3	Red		4	Yellow(Orange)	
5	Green		6	Cyan	
7	Blue		8	Purple	
0	Unknown	N/A			

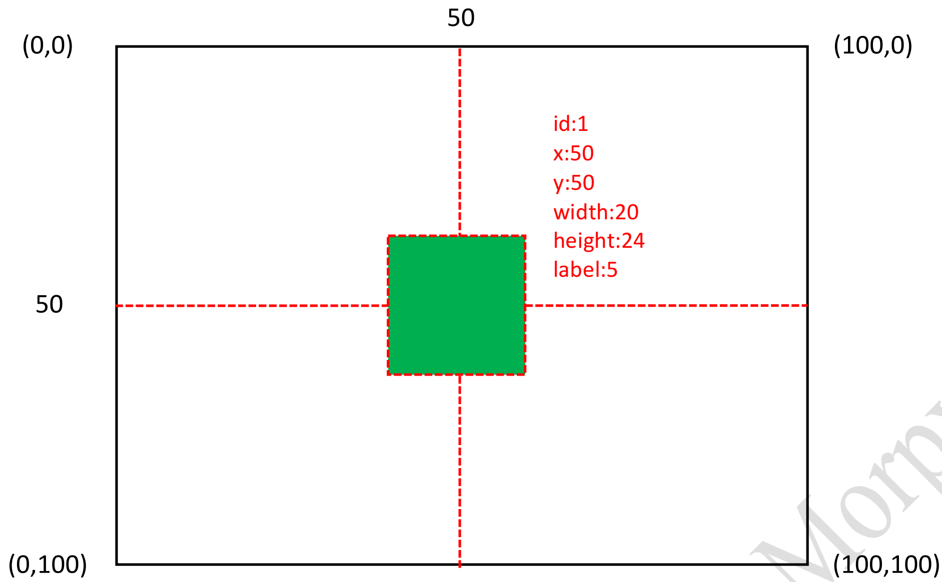
Note to the algorithm:

(1) Color detection is sensitive to light source. A white color stable light source is the best.
(2) Before using the color recognition, we recommend to lock the white balance. Point the camera to a white paper and set the CAM_AWB to auto-lock, the camera will adjust and store the white balance value.
(3) The algorithm performs best when recognizing single color object, objects with mixed color might not work well.
(4) The shadow or uneven lighting caused by the light source might affect the detection result.
(5) There should be no object of similar color in the background.
(6) You should set the minimum detection area large enough to remove the effects of distracting colors or objects.
(7) The larger the object is, the further away it can still be recognized.
(8) Due to the reason that LED color might interfere the color detection, the algorithm will automatically turn off the LED display (Auto mode) such that the color of the LED will not change when object is detected. You can turn the LED to white color to add additional light source from the sensor.
(9) The default sampling speed is high (50fps) in this algorithm.

Parameters:

Register	Parameters	Range	Default
0x25 Param1	invalid	/	/
0x26 Param2	invalid	/	/
0x27 Param3	minimum width of a valid region	0～100%	10
0x28 Param4	minimum height of a valid region	0～100%	10
0x29 Param5	color label to be detected	0～8	3 (red)

Results: The coordinate and size of a valid color region.



3.3.5 Color Recognition, Vision-ID: 2

Description:

This algorithm is used to recognize the color of a specified region which is set by user , returns the R、G、B channels average value and the color label.

Object:

Label	Object	Picture	Label	Object	Picture
1	Black(Dark Gray)		2	White(Light Gray)	
3	Red		4	Yellow(Orange)	
5	Green		6	Cyan	
7	Blue		8	Purple	
0	Unknown	N/A			

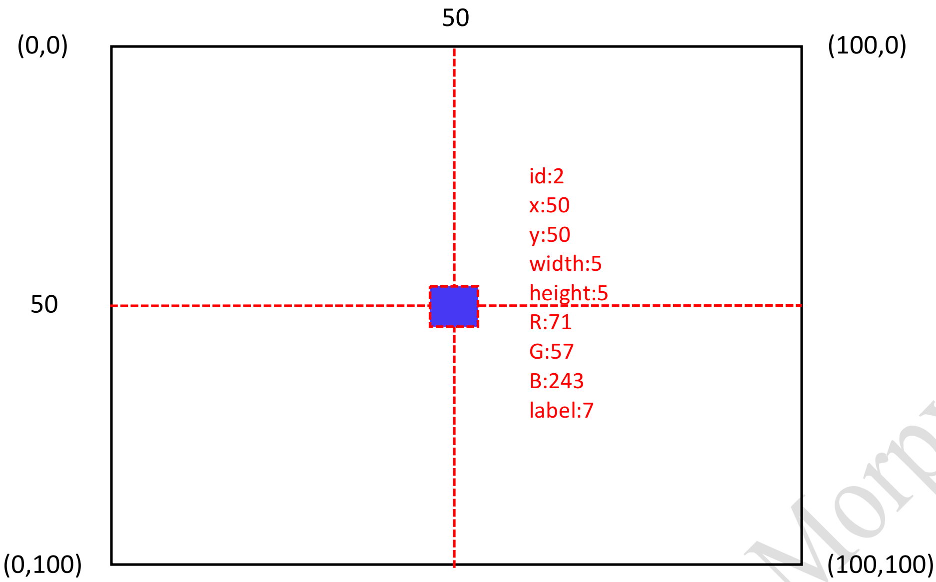
Parameters:

Register	Parameters	Range	Default
0x25 Param1	the x-center coordinate of the region which to be recognized	0～100%	50
0x26 Param2	the y-center coordinate of the region	0～100%	50
0x27 Param3	region width	0～100%	5
0x28 Param4	region height	0～100%	5
0x29 Param5	invalid	/	/

Note to the Algorithm:

(1) The algorithm is used to recognize the color of the specified region.
(2) Color detection is sensitive to light source. A white color stable light source is the best.
(3) Before using the color recognition, we recommend to lock the white balance. Point the camera to a white paper and set the CAM_AWB to auto-lock, the camera will adjust and store the white balance value.
(4) The algorithm performs best when recognizing single color object, objects with mixed color might not work well.
(5) The computation time is proportional to the size of the area selected.
(6) The larger the object is, the further away it can still be recognized.
(7) If there is not enough light, you can set the on-board LED to manual mode and use white color to add additional light to the scene.

Results: The R、G、B channels average value(range 0 ~ 255) and the color label.

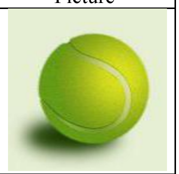


3.3.6 Ball, Vision-ID: 3

Description:

This algorithm is used to detect a ping-pong ball or a tennis ball, returns its coordinates, size and label.

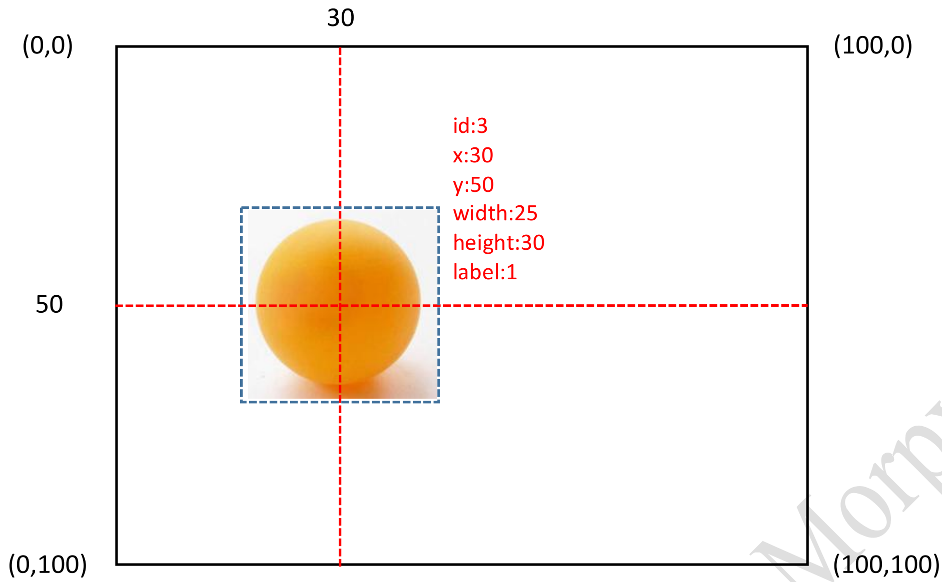
Object:

Label	Object	Picture	Label	Object	Picture
1	Ping-pong ball(orange)		2	Tennis ball (green)	

Note to the algorithm:

(1) Please make sure the ball is not occluded and all the edge features visible to the camera. Partial occlution will reduce the detection performance.
(2) Please do not use this algorithm in background of similar colors.
(3) Please make sure there are no object of similar color or shape in the scene, such as oranges or circular patterns.
(4) Due to the small size of the Ping-pong ball, the detection distance will be shorter than the tennis ball. You can use the CAM_ZOOM parameter to detect ping-pong balls that are far away.
(5) If the ball is moving, it might create motion blur that affects the detection. You can try to increase the CAM_FPS parameter to reduce the motion blur.
(6) To achieve the best performance, we recommend a white light environment. For yellow lighting environment, you could change the white balance parameters to yellow light.
(7) Do not put the ball under direct side light causing extreme bright & dark transition on the ball surface.
(8) When there are multiple balls in the scene, the one with highest score is returned.

Results: The center coordinate of the ball, the size of its external border and the label.

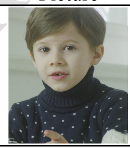


3.3.7 Body, Vision-ID: 5

Description:

This algorithm is used to detect the presence of human body in the field of vision and returns its coordinate and size. Human features refer to the area enclosed by the head and the upper body.

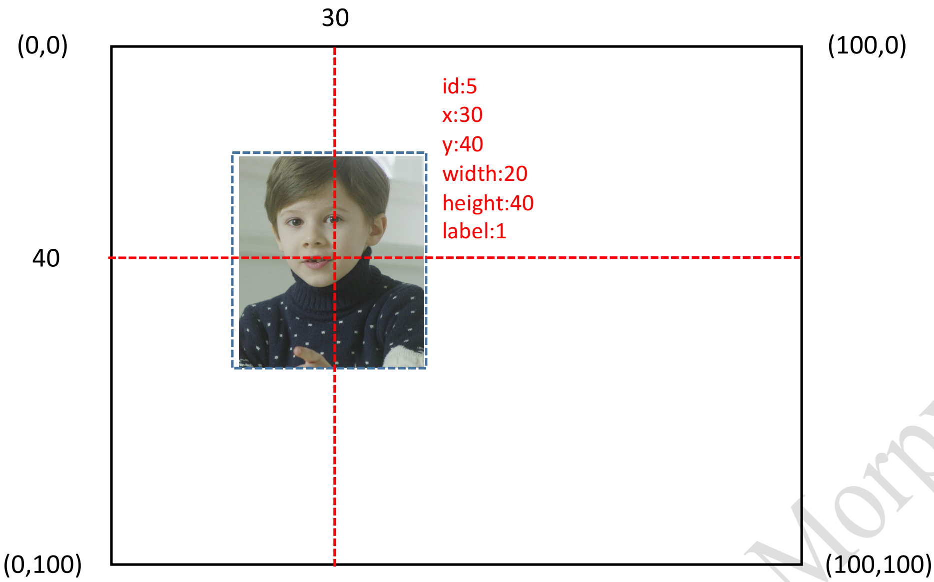
Object:

Label	Object	Picture
Invalid	Upper-body	

Note to the algorithm:

(1) The detection algorithm includes the area from head to the upper torso.
(2) Human detection does not depend on gender or age, but the size of the body will affect the detection result. Smaller body sizes will have a closer max detection distance.
(3) You can achieve best detection when the human faces frontal to the camera. The detection performance might degrade when the camera or the human is tilting at an angle.
(4) It is recommended that people's clothing are not similar to the background color.
(5) When there are multiple people, the one with highest confidence (usually the largest one) will be returned.

Results: The center coordinate of the human, the size of the upper body bounding box.

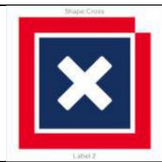
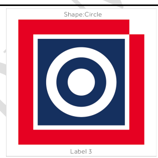


3.3.8 Shape Cards, Vision-ID: 6

Description:

This algorithm is used to detect a specified shape card which is shown in the below,recognize and return its label.

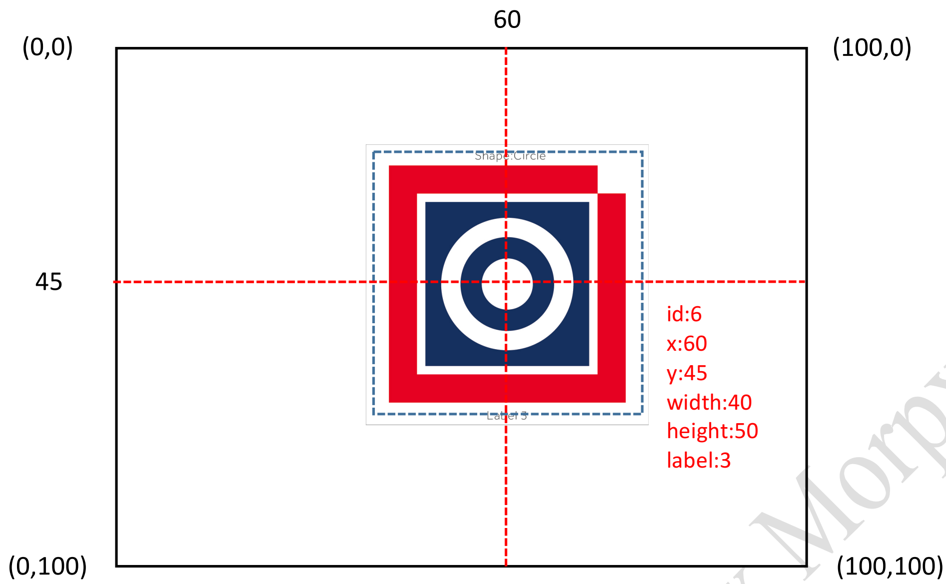
Object:

Label	Object	Picture	Label	Object	Picture
1	Check		2	Cross	
3	Circle		4	Square	
5	Triangle		0	Unknown	

Note to the algorithm:

(1) Currently, we only support recognizing the patterns in the table.
(2) The missing corner of the shape is used to indicate the upright position of the pattern.
(3) The red rectangle of the card is used to locate the pattern in the image, you should avoid such pattern in the scene.
(4) To achieve the best result, the card should be positioned UPRIGHT, with a rotation or tilting angle < 30 degree.
(5) A direct light shining to the card might cause specular reflection on the card surface and should be avoided.
(6) When there are multiple cards in the view, the card with highest detection score will be returned.
(7) In the case of requiring high detection accuracy, or there are multiple group of cards (shape / traffic, etc) in the scene, please select the performance priority mode. This mode will reduce the false recognition between different group of cards, but the detection speed will also reduce. To get better speed, please choose speed priority mode or balanced mode.

Results: The center coordinate of the card, the size of the outer edge and the label.

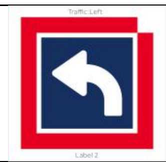


3.3.9 Traffic Cards, Vision-ID: 7

Description:

This algorithm is used to detect a specified traffic card which is shown in the below and recognize its label.

Object:

Label	Object	Picture	Label	Object	Picture
1	Forward		2	Left	
3	Right		4	Turn Around	
5	Park		0	Unknown	

Note to the algorithm:

Simiar to the shape cards.

Results:

Similar to the shape cards.

3.3.10 Number Card, Vision-ID: 8

Description:

This algorithm is used to detect a specified number card which is shown in the below and recognize its label.

Object:

Label	Object	Picture	Label	Object	Picture

1	One		2	Two	
3	Three		4	Four	
5	Five		6	Six	
7	Seven		8	Eight	
9	Nine		0	Zero	

Notes to the algorithm:
Similar to the shape cards.
Results:
Similar to the shape cards.

4 MU Protocol (for UART communication only)

The communication protocol in UART mode。

4.1 Protocol Format

START	LEN	ADDR	CMD	DATA	CHK	END

START ： Start code. Always 0xFF

LEN ： Length. The total bytes number from START to END

ADDR ： Device address. Range: 0x60~0x63. In addition, 0x00 is the broadcast address and all device will receive it.

CMD ： Command code / Response code

DATA ： Data

CHK ： Checksum. Cumulate sum all the bytes from START to DATA

example：FF 08 60 01 20 03 8B ED, the byte 8B is the checksum

END ： End code. Always 0xED

4.2 Response Code

Response	START	LEN	ADDR	RSP	DATA		CHK	END
Correct	0xFF	LEN	ADDR	0xE0	Command Code	Register Address	CHK	0xED
Error				0xE1				
Unknown Error				0xE2				
Timeout Error				0xE3				
Checksum Error				0xE4				
Length Error				0xE5				
Command Error				0xE6				
Register Address Error				0xE7				
Register Data Error				0xE8				
Read-only Error				0xE9				

4.3 Command Code

Command	START	LEN	ADDR	CMD	DATA				CHK	END
Write Register	0xFF	LEN	ADDR	0x01	REG		VAL		CHK	0xED
Read Register				0x02	REG					
Result Message				0x11	FRAME	VISION	NUM	RESULTS		
Request Message				0x12	VISION_ID					

4.3.1 Write Register - 0x01

Description: Write DATA(1Byte) to REG

START	LEN	ADDR	CMD	DATA		CHK	END
0xFF	0x08	Address	0x01	REG	DATA	Checksum	0xED

REG：Register address

DATA：Register data

Response:

Success:

START	LEN	ADDR	RSP	DATA		CHK	END
0xFF	0x08	Address	0xE0	0x01	REG	Checksum	0xED

Fail:

START	LEN	ADDR	RSP	CHK	END
0xFF	0x06	Address	Error	Checksum	0xED

4.3.2 Read Register - 0x02

Description: Read DATA(1Byte) from REG

START	LEN	ADDR	CMD	DATA	CHK	END
0xFF	0x07	Address	0x02	REG	Checksum	0xED

REG：Register address

Response when **successful:**

START	LEN	ADDR	RSP	DATA		CHK	END
0xFF	0x08	Address	0xE0	0x02	DATA	Checksum	0xED

DATA：The data of REG register

Response when failed:

START	LEN	ADDR	RSP	CHK	END
0xFF	0x06	Address	Error	Checksum	0xED

4.3.3 Result Message - 0x11

Description: The vision detection results message, no response.

START	LEN	ADDR	CMD	DATA				CHK	END
0xFF	Length	address	0x11	FRAME	VISION_ID	NUM	RESULTS	Checksum	0xED

FRAME：Frame count, cycles through 1 to 100, automatically increased by each frame processed and reset to 1 after reaching 100

VISION_ID：Vision algorithm ID

NUM：The number of detected results

RESULT：Result data. Each result data has 5 bytes, see the table below for details:

Vision	Byte1	Byte2	Byte3	Byte4	Byte5
Color Detection	Center_X	Center_Y	Width	Height	Label
Color Recognition	R	G	B	/	Label
Ball	Center_X	Center_Y	Width	Height	Label
Body	Center_X	Center_Y	Width	Height	/
Card	Center_X	Center_Y	Width	Height	Label

4.3.4 Request Message - 0x12

Description: MU will return detection results immediately while it receives this message。

START	LEN	ADDR	CMD	DATA	CHK	END
0xFF	0x07	Address	0x12	VISION_ID	Checksum	0xED

VISION_ID：The vision algorithm that you required

Returns

Success: return Result Message that mentioned in 4.3.3

Fail:

START	LEN	ADDR	CMD	CHK	END
0xFF	0x06	Address	Error	Checksum	0xED

5 Registers

Address	Register	Default	R/W	Bit	Symbol	Description
0x01	PROTOCOL_VER	0x03	R	[7:0]	PROTOCOL_VER	Protocol version
0x02	FIRMWARE_VER	N/A	R	[7:0]	FIRMWARE_VER	Firmware version
0x03	RESTART	0x00	W	[0]	RESTART	Restart sensor 0：processing 1：restart
0x04	SENSOR_CONF1	0x00	R/W	[0:1]	RESERVED	/
				[2]	DEFAULT	Restore default settings 0：current settings 1：Restore default settings
				[7:3]	RESERVED	/
0x05	LOCK	0x00	R/W	[0]	REG_LOCK	Register read-write security lock. After setting ‘1’, the sensor will not update the detection results until the LOCK is released. 0：unlock 1：lock
0x06	LED1	0x28	R/W	[0]	LED1_MODE	LED1 Mode 0：Auto mode(default) 1：Manual mode
			R/W	[3:1]	LED1_DETECT_COLOR	Auto mode：the color of LED1 when object is detected Manual mode：Manually set the color of LED1 000：off 001：red 010：green 011：yellow 100：blue(default) 101：purple 110：cyan 111：white
			R/W	[4]	LED1_HOLD	Whether to keep the color of the LED1 at each frame 0：turn off(default) 1：holding
			R/W	[7:5]	LED1_UNDETECT_COLOR	Auto mode：LED1 color when object is not detected Manual mode：invalid 000：off 001：red(default) 010：green 011：yellow 100：blue 101：purple 110：cyan 111：white
0x07	LED2	0x28	R/W	[0]	LED2_MODE	LED2Mode 0：Auto mode(default) 1：Manual mode
			R/W	[3:1]	LED2_DETECT_COLOR2	Auto mode：the color of LED2 when object is detected Manual mode：Manually set the color of LED2 000：off 001：red 010：green 011：yellow 100：blue(default) 101：purple 110：cyan 111：white
			R/W	[4]	LED2_HOLD	Whether to keep the color of the LED2 at each frame 0：turn off(default) 1：holding
			R/W	[7:5]	LED2_UNDETECT_COLOR	Auto mode：LED2 color when object is not detected Manual mode：invalid 000：off 001：red(default) 010：green 011：yellow 100：blue 101：purple 110：cyan

						111：white
0x08	LED_LEVEL	0x11	R/W	[3:0]	LED1_LEVEL	The luminous intensity level of LED1 range：0～15
				[7:4]	LED2_LEVEL	The luminous intensity level of LED2 range：0～15
0x09	UART	0x00	R/W	[2:0]	BAUDRATE	UART baud-rate. automatically saved if changed. Data-bits: 8, Stop-bits: 1, Parity: none 000：9600(default) 001：19200 010：38400 011：57600 100：115200 101：230400 110：460800 111：921600
				[7:3]	RESERVED	/
0x0A~ 0x0F	RESERVED	0x00	/			
0x10	CAMERA_CONF1	0x10	R/W	[2:0]	CAM_ZOOM	Digital Zoom 000：Default/Auto 001：ZOOM1 010：ZOOM2 011：ZOOM3 100：ZOOM4 101：ZOOM5
				[3]	CAM_ROTATE	Rotate the camera 0：default 1：rotate 180 degree
				[4]	CAM_FPS	Camera sample rate 0：Standard 1：High speed
				[6:5]	CAM_AWB	White Balance 00：Auto white balance(default) 01：Auto-Lock. The white balance will auto locked after sampling several frames if the bit is set '1'. This mode should be used for color sensitive algorithms or a large areas of color background environment. Keep the camera facing a white paper for calibrating the white balance. White lighting environment is the prefered. 10：White Light / Cloudy 11：Yellow Light / Sunny
				[7]	RESERVED	/
0x11~ 0x1E	RESERVED	0x00	/			
0x1F	FRAME_CNT	0x01	R	[0:7]	FRAME_CNT	Frame count. range：1～100
0x20	VISION_ID	0x00	R/W	[7:0]	VISION_ID	Vision ID range：1～16
0x21	VISION_CONF1	0x20	R/W	[0]	STATUS	Vision status 0：disable 1：enable
				[1]	DEFAULT_VISION	Vision default setting 0：current settings 1：Restore default settings
				[3:2]	OUTPUT_MODE	Message output mode 00：Answer mode，the detection results will be returned if the sensor is received the request message or bit[7] is set to '1'. 01：Continuous mode，output the result continuously. 10：Event mode，output the result when the object is detected.
				[5:4]	LEVEL	Vision performance level 00：Default 01：Speed 10：Balance(default) 11：Accuracy
				[6]	RESERVED	
				[7]	OUTPUT_EN	Message output enable

						0：disable 1：enable
0x22~0x24	RESERVED	0x00	/			
0x25	PARAM_VALUE1	0x00	R/W	[7:0]	PARAM_VALUE1	Vision configuration parameters
0x26	PARAM_VALUE2	0x00	R/W	[7:0]	PARAM_VALUE2	
0x27	PARAM_VALUE3	0x00	R/W	[7:0]	PARAM_VALUE3	
0x28	PARAM_VALUE4	0x00	R/W	[7:0]	PARAM_VALUE4	
0x29	PARAM_VALUE5	0x00	R/W	[7:0]	PARAM_VALUE5	
0x2A	VISION_STATUS1	0x00	R	[0]	VISION1	Vision 1 enabled status 0：disabled 1：enabled
				[1]	VISION2	Vision 2 enabled status
				[2]	VISION3	Vision 3 enabled status
				[3]	VISION4	Vision 4 enabled status
				[4]	VISION5	Vision 5 enabled status
				[5]	VISION6	Vision 6 enabled status
				[6]	VISION7	Vision 7 enabled status
				[7]	VISION8	Vision 8 enabled status
0x2B	VISION_STATUS2	0x00	R	[0]	VISION9	Vision 9 enabled status
				[1]	VISION10	Vision 10 enabled status
				[2]	VISION11	Vision 11 enabled status
				[3]	VISION12	Vision 12 enabled status
				[4]	VISION13	Vision 13 enabled status
				[5]	VISION14	Vision 14 enabled status
				[6]	VISION15	Vision 15 enabled status
				[7]	VISION16	Vision 16 enabled status
0x2C~0x2F	RESERVED 预留	0x00	/			
0x30	VISION_DET1	0x00	R	[0]	VISION1	Vision 1 detection result 0：undetected 1：detected
				[1]	VISION2	Vision 2 detection result
				[2]	VISION3	Vision 3 detection result
				[3]	VISION4	Vision 4 detection result
				[4]	VISION5	Vision 5 detection result
				[5]	VISION6	Vision 6 detection result
				[6]	VISION7	Vision 7 detection result
				[7]	VISION8	Vision 8 detection result
0x31	VISION_DET2	0x00	R	[0]	VISION9	Vision 9 detection result
				[1]	VISION10	Vision 10 detection result
				[2]	VISION11	Vision 11 detection result
				[3]	VISION12	Vision 12 detection result
				[4]	VISION13	Vision 13 detection result
				[5]	VISION14	Vision 14 detection result
				[6]	VISION15	Vision 15 detection result
				[7]	VISION16	Vision 16 detection result
0x32~0x33	RESERVED	0x00	/			
0x34	RESULT_NUM	0x00	R	[7:0]	RESULT_NUM	The number of detection results range : 0～10
0x35	RESULT_ID	0x01	R/W	[7:0]	RESULT_ID	Result ID range : 1～10
0x36	READ_STATUS1	0x00	R/W	[0]	DATA1	Result data1 read status 0：old data 1：new data
				[1]	DATA2	Result data2 read status
				[2]	DATA3	Result data3 read status
				[3]	DATA4	Result data4 read status
				[4]	DATA5	Result data5 read status
				[7:5]	RESERVED	/
0x37~0x3F	RESERVED	0x00	/			
0x40	RESULT_DATA1	0x00	R	[7:0]	RESULT_DATA1	Result data, has different meaning with different algorithms. Please check the algorithm section for details.
0x41	RESULT_DATA2	0x00	R	[7:0]	RESULT_DATA2	
0x42	RESULT_DATA3	0x00	R	[7:0]	RESULT_DATA3	
0x43	RESULT_DATA4	0x00	R	[7:0]	RESULT_DATA4	
0x44	RESULT_DATA5	0x00	R	[7:0]	RESULT_DATA5	
...	RESERVED	0x00	/			

6 Firmware Upgrade

6.1 Install Flash Download Tools

The firmware is upgradable with the Flash Download Tools provided by Espressif.Inc.:

Download Link:
https://www.espressif.com/sites/default/files/tools/flash_download_tools_v3.6.5.zip

6.2 Down Latest Firmware of MU

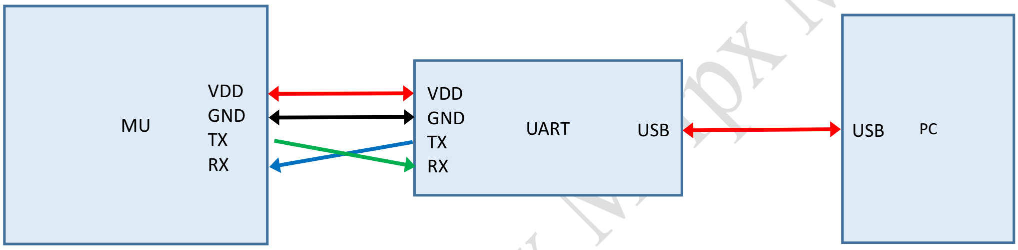
Please visit Morpx website or contact the support to get the latest version of firmware:

http://www.morpx.com/

6.3 Upgrade

(1) Connection

A usb-uart communication module is needed to connect MU Vision Sensor to the computer, and connect as shown below:

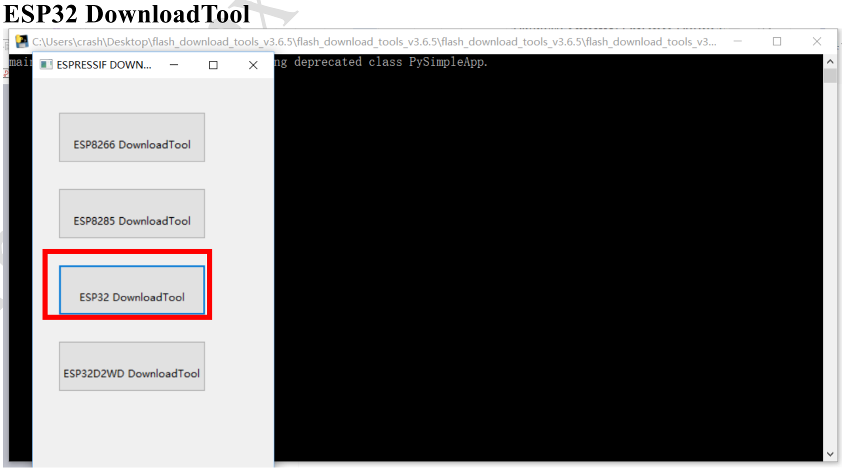


(2) Flash Mode

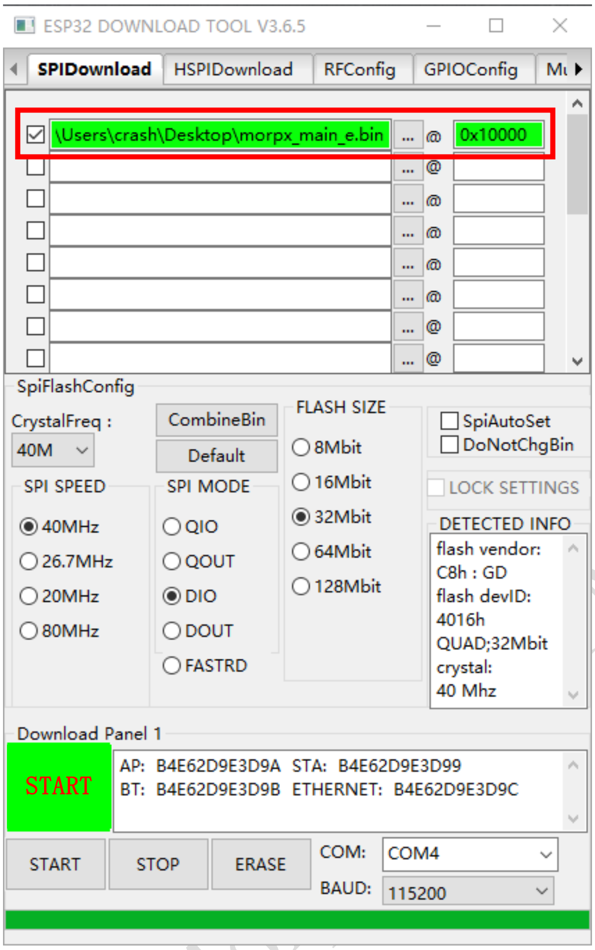
First press and hold the mode button, then press and release the reset button, and then release the mode button to enter the flash mode.

(3) Open the flash_download_tools_vx.x.x.exe

(4) Click ESP32 DownloadTool



(5) Set Parameter And Add The Firmware Path.



SPI SPEED: 40MHz
SPI MODE: DIO
FLASH SIZE: 32Mbit
BAUD: 115200

Note: A higher speed baud rate can be selected such as 921600 to burn faster if you have a high-performance serial port module.

COM： The corresponding COM port that connects to the sensor. You can use the Windows Device Manager to view all the COM port for the UART to USB module。

(6) Add the File Path

Click the "..." button in the red box to choose the firmware file path and check the box in the front.

(7) Enter Address

Enter **0x10000** after the @ symbol.

Important: Do not enter the wrong address or try another address, otherwise the build-in firmware may be damaged. If there is any problem, please contact the customer service of Morpx for help:

Tel: +86 0571 8195 8588
E-Mail: support@morpx.com

(8) Upgrade

Click the START button in the lower left corner to start the download and wait for the "FINISH" text.

(9) Restart MU

7. Product Upgrade Plan

We thank you for using the MU vision sensor. As a completely new category of products, MU vision sensor's launch are well received from our customers. The MU vision sensor allows users to better understand the visual recognition and experiment with its versatile applications in robotics. To continuously improve our product, we will push out firmware update regularly to unlock new features, such as new object recognition capabilities or new connectivity options such as wifi streaming. Please check our website (www.morpx.com) or follow us on Facebook (https://www.facebook.com/MU4Toys/)

Revision

Date	Version	Release Notes
2018.02.28	V0.4	draft
2019.04.22	V0.5	Firmware update v0422
2019.05.30	V0.8	Update algorithm notes, add light and motion sensor docs.