



- 1x Model Housing Kit
- 1x Camera Back Clip (LEGO compatible)
- 1x HY2.0-4P Cable

## Applications

- Remote Monitoring
- Camera Timer

## Specifications

Specifications	Parameters
ESP32-WROOM-32E	240MHz Dual Core, 600DMIPS, 520KB SRAM, Wi-Fi, Dual Mode Bluetooth
Flash	4MB
Wide Angle Fisheye Lens	Focal length 1.0±5%mm
	Aperture 2.2±5%
	Lens type 1/4inch
	Field of view 160°
	Pixels 200w
	Reverberation -135%
Conventional view lens	Sensitive chip OV2640
	Focal length 4.8±5%mm
	Aperture 2.4±5%
	Lens type 1/4inch
	Field of view 65°
	Pixels 200w
Firmware default communication method	Reverberation <1%
	Sensor chip OV2640
Camera firmware default output image	UART: 115200bps 8N1
Sensor output format support	QVGA@28fps, VGA@13fps, support for adjustments (UXGA, SXGA, XGA, and other larger image sizes output)
Net Weight	YUV(422/420)/YCbCr422,8-bit compressed data,RGB565/555,8-/10-bit raw RGB data
Gross Weight	21.8g
Model board	49.1g
Fisheye camera assembly size	11 * 66 * 181mm
	40 * 24 * 17.2mm

## Specifications

## Parameters

Assembly size for normal camera

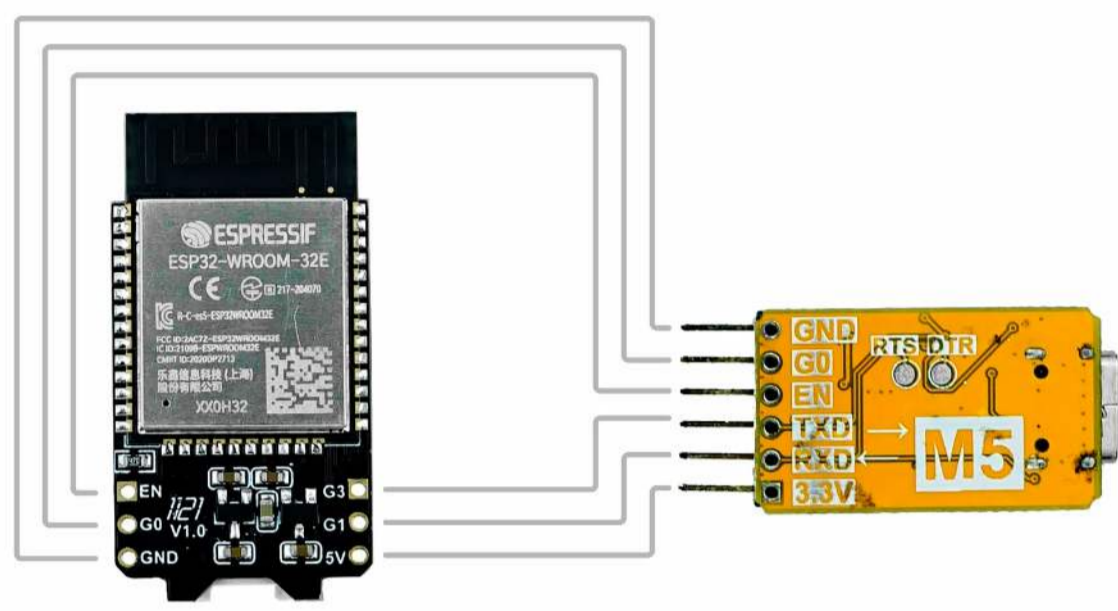
40 \* 24 \* 11m

Package Size

25 \* 73 \* 220mm

## Flash Burning Download

The Unit CAM does not include a program download circuit. If you need to download an updated program for the ESP32, you can do so by connecting an external USB-TTL downloader.



## Pin Mapping

### OV2640 Interface

<i>interface</i>	<i>camera pins</i>	<i>unit CAM</i>
SCCB Clock	SIOC	IO23
SCCB Data	SIOD	IO25
System Clock	XCLK	IO27
Vertical Sync	VSYNC	IO22
Horizontal Reference	HREF	IO26
Pixel Clock	PCLK	IO21
Pixel Data Bit 0	D0	IO32
Pixel Data Bit 1	D1	IO35
Pixel Data Bit 2	D2	IO34
Pixel Data Bit 3	D3	IO5
Pixel Data Bit 4	D4	IO39
Pixel Data Bit 5	D5	IO18
Pixel Data Bit 6	D6	IO36
Pixel Data Bit 7	D7	IO19

Camera Reset	RESET / IO15	IO15
Camera Power Down	PWDN	-1
Power Supply 3.3V	3V3	3V3
Ground	GND	GND

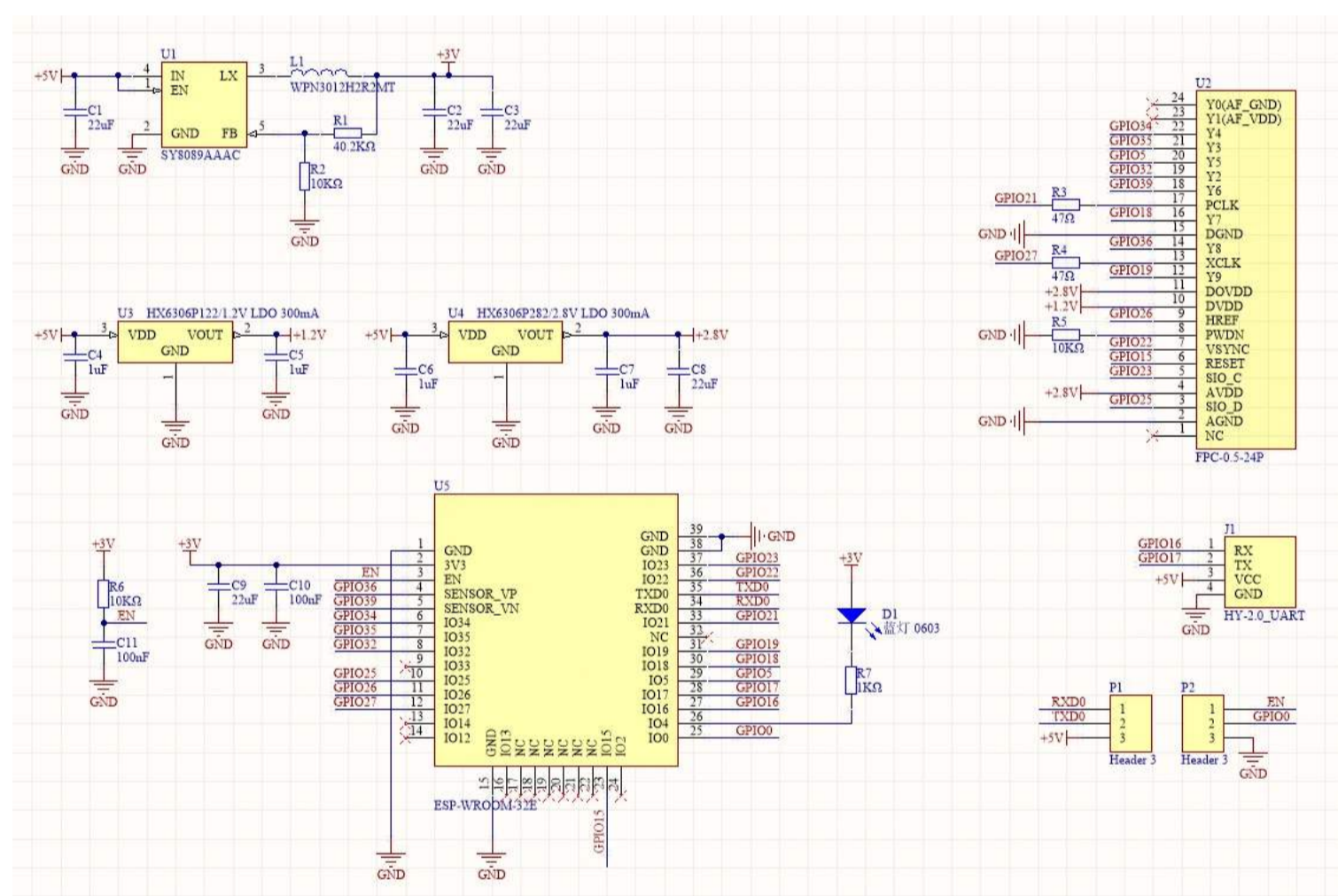
### HY2.0-4P Interface.

<i>HY2.0-4P</i>	<i>Unit CAM</i>
RX	IO16
TX	IO17
5V	5V
GND	GND

### LED (Blue)

<i>LED</i>	<i>Unit CAM</i>
D1	IO4

## schematic



## Related Links

- [Datasheet](#)
  - [esp32-wroom-32e](#)
  - [OV2640](#)

## Case program

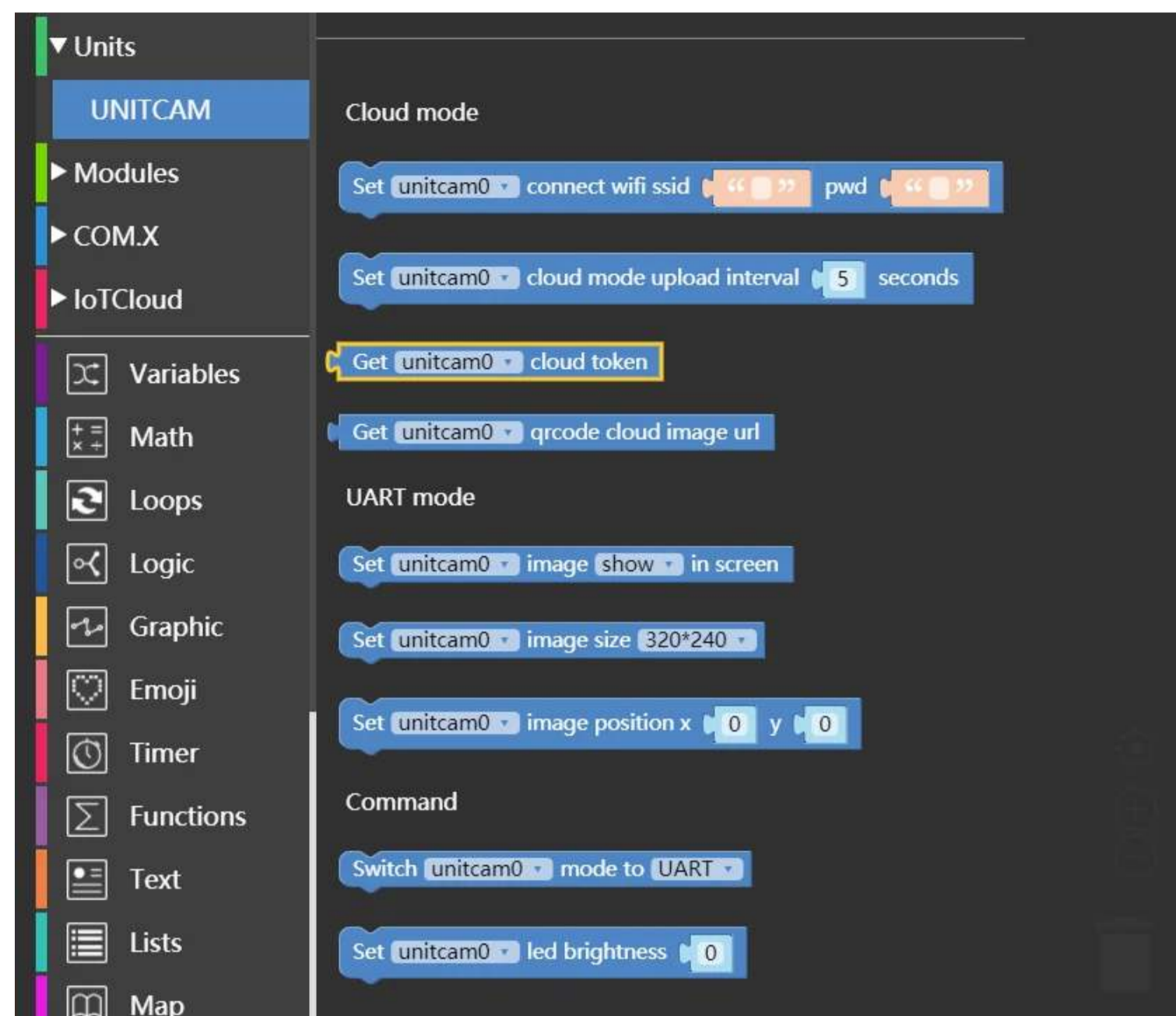
### Arduino

- [Camera UART to Core](#)

### UIFlow

The default firmware of UnitCAM supports both UART and WiFi modes. WiFi mode supports taking pictures at regular intervals and uploading them to M5 cloud automatically, and returning the picture calling interface (URL). UART mode supports users to use other master devices to get picture frames and modify picture parameters through UART communication.

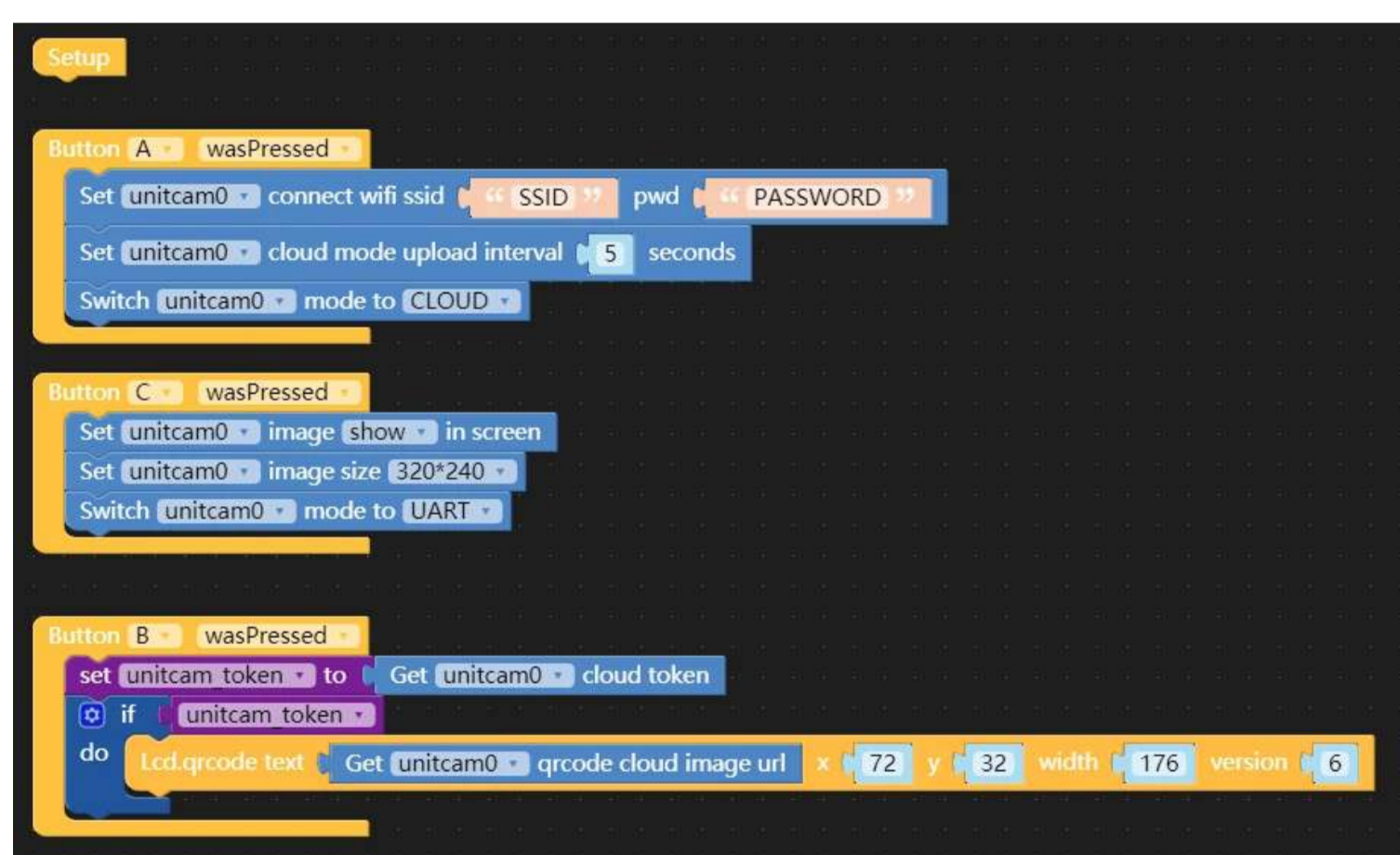
## Unit CAM UIFlow Block Instructions



## Unit CAM UIFlow Use Case

Press A to enable WiFi connection mode, connect to the designated WiFi, the camera takes pictures at 5s intervals and upload them to the M5 cloud, and press B to obtain the camera Token, and use the cloud image URL generated by the Token to create a QR code.

Press C to enable UART connection mode, configure the picture size to be 320\*240, and start the picture display thread to continuously refresh the picture to the screen.



## Unit CAM UIFlow Block function description:

### - Cloud Mode Functions:

*#Configure the camera to connect to the specified WiFi*

**Set-unitcam-connect-wifi-ssid**

*#Configure the camera image upload cloud interval*

**Set-unitcam-cloud-mode-upload-interval**

*#Get camera Token*

**Get-unitcam-cloud-token**

```
#Get the URL of the camera cloud image
```

```
Get-unitcam-cloud-image-url
```

```
- UART Mode Functions:
```

```
#In UART mode, start the screen display image thread, the default display coordinates are 0,0
```

```
Set-unitcam-image-show-in-screen
```

```
#Set the camera to return the image size
```

```
Set-unitcam-image-size
```

```
#Set the image display position
```

```
Set-unitcam-image-position-x-y
```

```
- Command:
```

```
#Switch UART/CLOUD mode
```

```
Switch-unitcam-mode-to-UART/CLOUD
```

```
#Set the brightness of the onboard LED of the camera
```

```
Set-unitcam, led-brightness
```

## Micropython API

```
import unit
```

```
#Create unitcam instance
```

```
unitcam0 = unit.get(unit.UNITCAM, unit.PORTB)
```

```
#UART Mode: unitcam0.UART_MODE
```

```
#WiFi mode: unitcam0.CLOUD_MODE
```

```
unitcam0.set_mode(unitcam0.UART_MODE)
```

```
#WiFi mode, configure the camera to connect to the specified WiFi
```

```
unitcam0.set_wifi('', '')
```

```
#WiFi mode, configure the camera image upload cloud interval
```

```
unitcam0.set_upload_time(5)
```

```
#Get camera token
```

```
unitcam_token = unitcam0.get_token()
```

```
#Stitching a fixed domain name field to get the URL of the camera cloud image
```

```
unicam_image_url = 'camera.m5stack.com/timer-cam/image?tok=' + unitcam0.get_token()
```

```
#UART mode, start the screen display image thread, the default display coordinates are 0,0
```

```
unitcam0.set_show(True)
```

```
#Set the camera to return the image size
```

```
#unitcam0.SIZE_800_600
```

```
#unitcam0.SIZE_640_480
```

```
#unitcam0.SIZE_320_240
```

```
#unitcam0.SIZE_240_240
```

```
#unitcam0.SIZE_160_120
```

```
unitcam0.set_image_size(unitcam0.SIZE_320_240)
```

*#Set Unit CAM onboard LED brightness value: 0~100*

`unitcam0.set_led_brightness(0)`

## | Video

---

- Unpacking and manual assembly