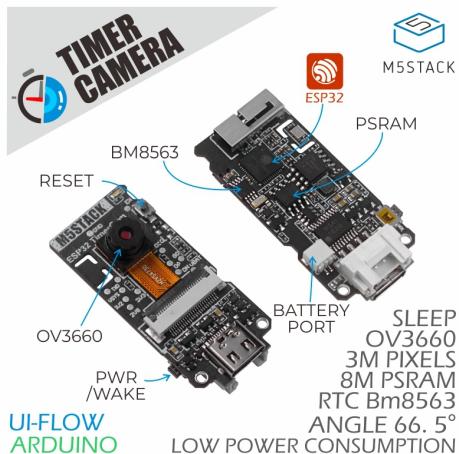


TimerCamera

SKU:U082



211-200921



Description

TimerCamera is a camera module based on the ESP32-D0WDQ6-V3, equipped with an onboard 8MB PSRAM and a 3MP camera (OV3660) with a DFOV of 66.5°, capable of capturing photos at resolutions up to 2048x1536. It features a status indicator and a RESET button, focusing on ultra-low power consumption design. Through the RTC (BM8563), it supports timed sleep and wake functions, with sleep current as low as 2µA. The board has a reserved battery connector for user-supplied battery power. The module supports Wi-Fi image transmission and USB port debugging, with a bottom HY2.0-4P port output for connecting other peripherals. Using M5Burner to flash the firmware, the **TimerCamera** can be configured directly via Camera-Tool, or the data can be processed in UiFlow.

Features

- Designed based on ESP32
- Wi-Fi image transmission
- Timed sleep and wake
- Status indicator
- Ultra-low power consumption design
- Development Platform: ESP-IDF / Arduino / UiFlow

Includes

- 1 x TimerCamera

Applications

- Timed photography
- Remote monitoring

Specifications

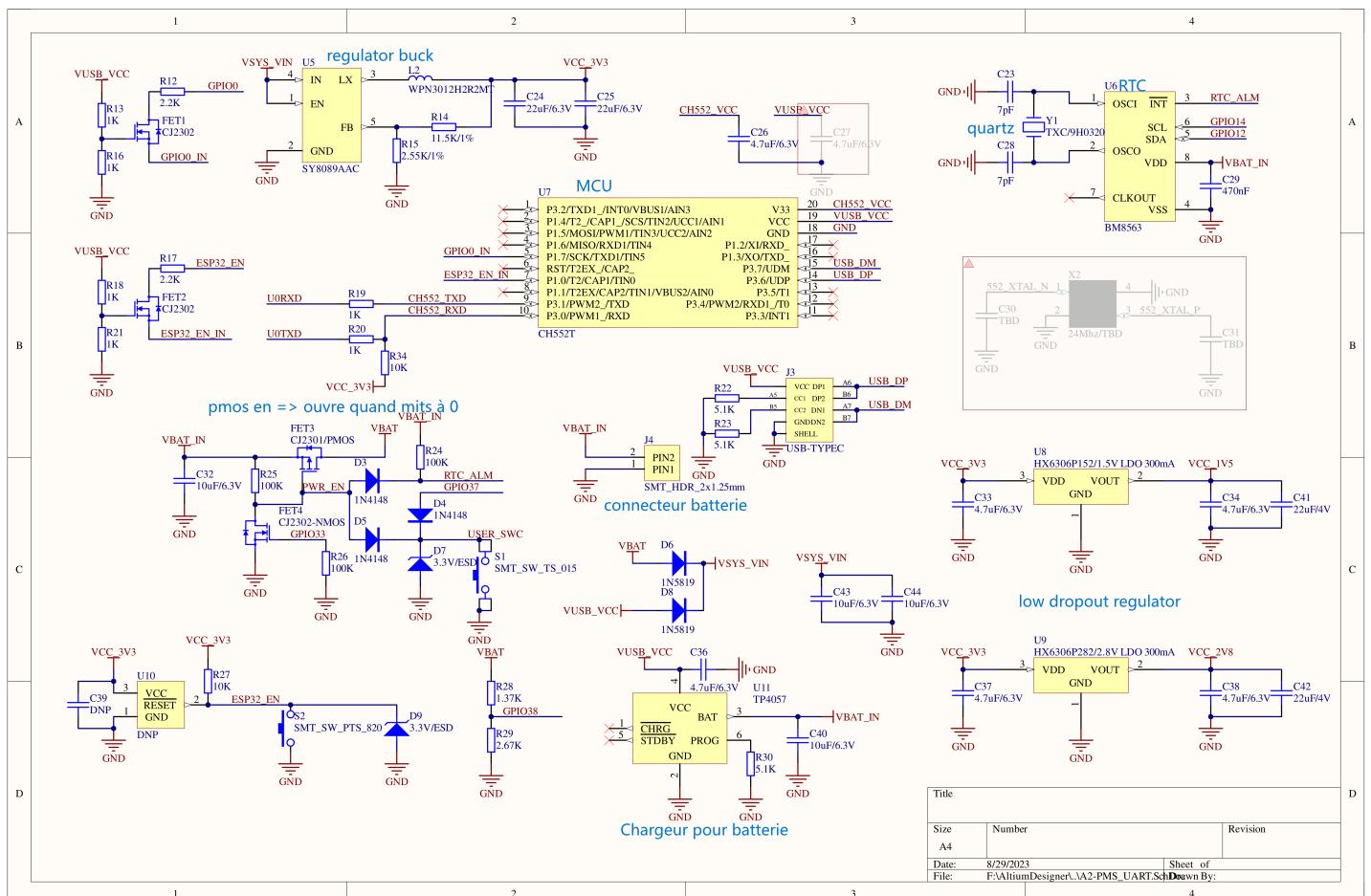
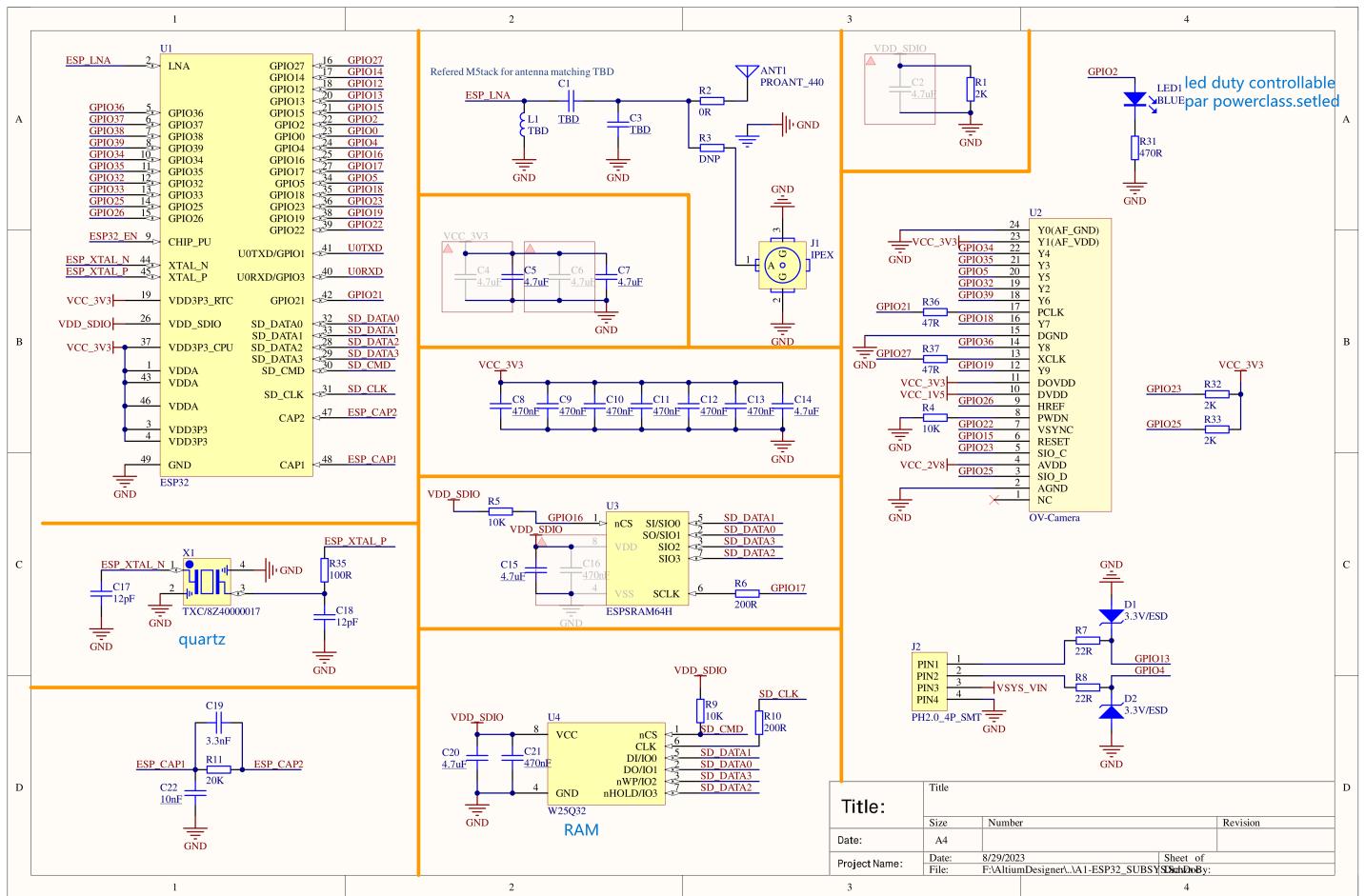
Specification	Parameter
PSRAM	8MB
Flash	4M
Image Sensor	OV3660
Max Resolution	3MP
Output Format	8-/10-Bit RAW, RGB and YCbCr output, compression
DFOV	66.5°
Battery Port	SH1.0-2P
Product Size	45 x 20 x 12mm
Product Weight	6g
Package Size	60 x 60 x 15mm
Gross Weight	17g

Learn

The low-power power management solution used by the Timer Camera series is different from that of CORE and StickC devices. The PWR button is used as the power-on button (long press for 2 seconds). To power off the device, you need to use a software API or press the reset button on the PCB. When using external power, the device will remain powered on.

Schematics

- [TimerCamera Schematics PDF](#)



PinMap

OV3660

Interface	Camera Pin	Timer Camera
SCCB Clock	SIOC	G23
SCCB Data	SIOD	G25
System Clock	XCLK	G27
Vertical Sync	VSYNC	G22
Horizontal Reference	HREF	G26
Pixel Clock	PCLK	G21
Pixel Data Bit 0	D0	G32
Pixel Data Bit 1	D1	G35
Pixel Data Bit 2	D2	G34
Pixel Data Bit 3	D3	G5
Pixel Data Bit 4	D4	G39
Pixel Data Bit 5	D5	G18
Pixel Data Bit 6	D6	G36
Pixel Data Bit 7	D7	G19
Camera Reset	RESET	G15
Camera Power Down	PWDN	-1
Power Supply 3.3V	3V3	3V3
Ground	GND	GND

HY2.0-4P

HY2.0-4P	TimerCamera		
SCL	G13	Jaune	
SDA	G4	Blanc	
5V	5V	Rouge	
GND	GND	Noir	

LED

LED	TimerCamera	
LED_Pin		G2

BM8563

BM8563	TimerCamera	
SCL		G14
SDA		G12

BAT

BAT	TimerCamera	
BAT_ADC_Pin		G38
BAT_HOLD_Pin		G33

Datasheets

- [ESP32-D0WDQ6-V3](#)
- [OV3660](#)

Softwares

Quick Start

- [Camera-Tool Guide](#)
- [UiFlow-Media Trans](#)
- [UiFlow-UART/TIMER](#)

- o Arduino Quick Start
- o Timer Folder Pusher
- o Timer Amazon S3 Folder Pusher

Arduino

- o TimerCamera-Arduino

ESP-IDF

- o FactoryTest
- o AWS-S3-PUT
- o Ali-OSS-PUT
- o SMB-OSS-PUT
- o Timer Wakeup
- o HTTP Stream
- o Wireless-send

Under-voltage / Power-off Protection

When the device is undervoltage, the camera may trigger undervoltage protection, causing a reset.

ESP-IDF

In ESP-IDF, you can configure and disable this in menuconfig when compiling the project.

`idf.py menuconfig`

Component config -> ESP32-specific -> Hardware brownout detect & reset (disable)

```
(Top) → Component config → ESP32-specific
    Espressif IoT
    - - <ESP32_ECO3_CACHE_LOCK_FIX>
        Minimum Supported ESP32 Revision (Rev 0)  --->
(0) <ESP32_REV_MIN>
    - *- <ESP32_DPORT_WORKAROUND>
        CPU frequency (160 MHz)  --->
(160) <ESP32_DEFAULT_CPU_FREQ_MHZ>
    [ ] Support for external, SPI-connected RAM
        SPI RAM config  --->
    - - <ESP32_MEMMAP_TRACEMEM>
    - - <ESP32_MEMMAP_TRACEMEM_TWOBANKS>
    [ ] Use TRAX tracing feature
        - - Reserve memory for tracing both pro as well
(0x0) <ESP32_TRACEMEM_RESERVE_DRAM>
    Number of universally administered (by IEEE) MAC
(4) <ESP32_UNIVERSAL_MAC_ADDRESSES>
    [ ] Enable Ultra Low Power (ULP) Coprocessor
(0)     RTC slow memory reserved for coprocessor
[*] Make exception and panic handlers TAG/OCD aware
[!] Hardware brownout detect & reset
    Brownout voltage level  --->
() <ESP32_BROWNOUT_DET_LVL>
    - - Reduce PHY TX power when brownout reset
        Timers used for gettimeofday function (RTC and h
        RTC clock source (Internal 150kHz RC oscillator)
        Additional current for external 32kHz crystal -
(1024) Number of cycles for RTC_SLOW_CLK calibration
() Number of attempts to repeat 32k XTAL calibratio
```

Arduino

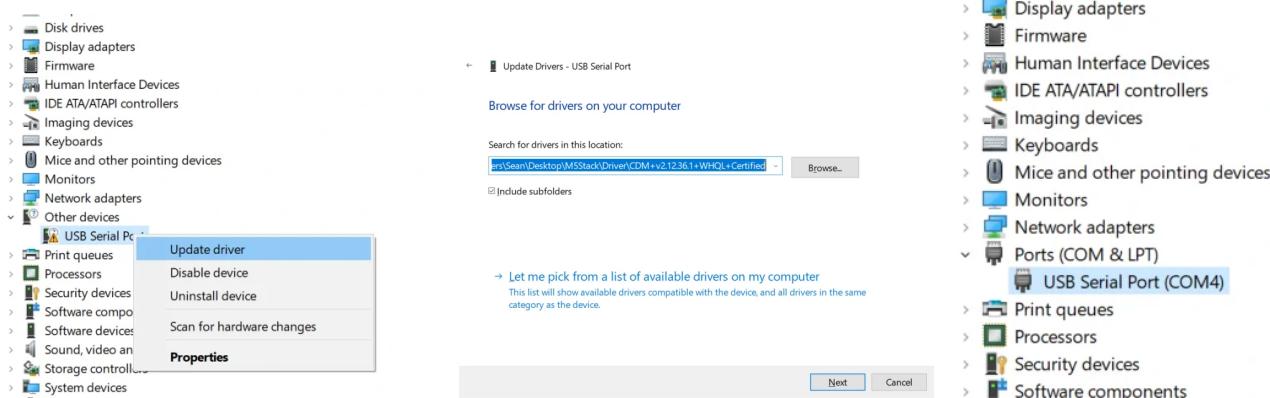
On the Arduino platform, you can disable it during initialization with the following sample code:

```
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"

void setup() {
    WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0); // disable detector
}
```

USB Driver

In some systems, TimerCAM may not work without a driver. You can fix this by manually installing the [FTDI driver](#). For example, in a Windows 10 environment, download the driver file that matches your OS, extract it, and install it via Device Manager. (Note: In some system environments, you may need to install the driver twice for it to take effect. The unrecognized device name is usually **M5Stack** or **USB Serial**. For Windows, it is recommended to install the driver file directly in Device Manager (custom update). The executable installer method may not work properly).



EasyLoader

Easyloader	Download	Note
TimerCamera Firmware Easyloader	download	/

Video

- Connect to the Unit TimerCAM hotspot, password 12345678. Open the browser at 192.168.4.1 to view the image. To use the timed photo function, please refer to the Quick Start guide.

[timer_cameraX.mp4](#)

[TimerCAM.mp4](#)

- Focal length adjustment

[timercam_lens_adj.mp4](#)

Product Comparison

To compare information on the TimerCamera series products, you can visit the [Product Selection Table](#), check the target products, and get the comparison results. The selection table covers key information such as core parameters and functional features, and supports comparison of multiple products simultaneously.