第十五讲
ESP32（II）
Lecture 15
ESP32 (II)

明玉瑞 Yurui Ming

yrming@gmail.com

声明
Disclaimer

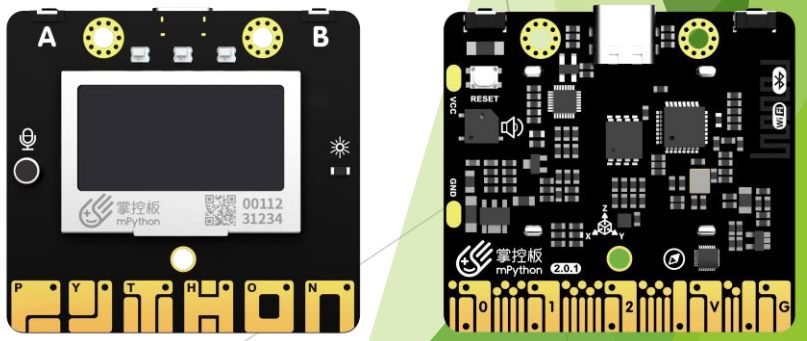
- 本讲义在准备过程中由于时间所限，所用材料来源并未规范标示引用来源。所引材料仅用于教学所用，作者无意侵犯原著者之知识版权，所引材料之知识版权均归原著者所有；若原著者介意之，请联系作者更正及删除。

  The time limit during the preparation of these slides incurs the situation that not all the sources of the used materials (texts or images) are properly referenced or clearly manifested. However, all materials in these slides are solely for teaching and the author is with no intention to infringe the copyright bestowed on the original authors or manufacturers. All credits go to corresponding IP holders. Please address the author for any concern for remedy including deletion.

掌控板
HandBit

- 掌控板是由盛思设计、制造与发行的基于ESP32的开源硬件开发板，主要面向物联网在教育中的应用。该开发板上集成了OLED显示屏、RGB灯、加速度计、麦克风、光线传感器、蜂鸣器、按键开关、触摸开关、金手指外部拓展接口，支持图形化及Python代码编程，利用掌控上丰富的传感器，可实现智能机器人、创客智造作品等智能控制类应用。同时，结合它尺寸小的特点还可以做很多智能穿戴、电子饰品等各种DIY作品应用。

- HandBit is an open-sourced board manufactured by the company Labplus based on ESP32. It's mainly targeting the student projects in the primary education and implementing STEM pedagogical model. This board also integrates OLED display, RGB LED, accelerometer, microphone, light sensor, buzzer, key switch, touch switch, external expansion interface, etc. It supports graphical and Python programming as well. All these features facilitates its usage in intelligent robotic and controls for makers. In addition, given its small dimension of size, it can also be used to make smart wear, electronic jewelry and other DIY work applications.

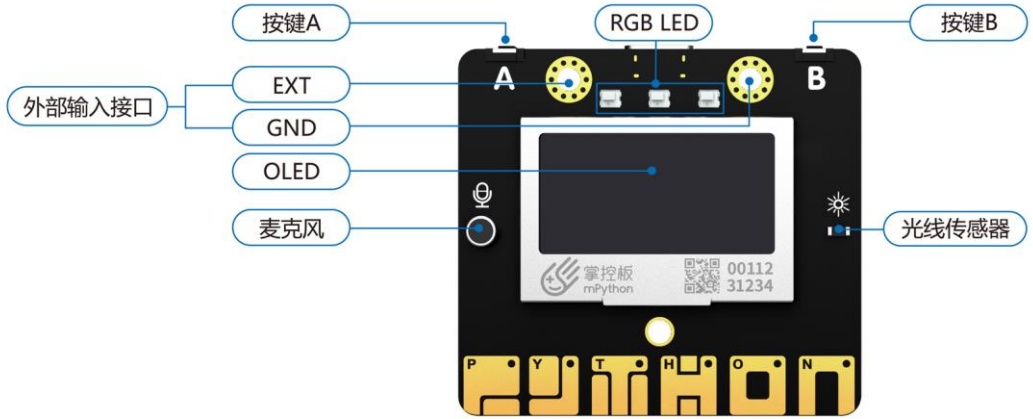


掌控板
HandBit

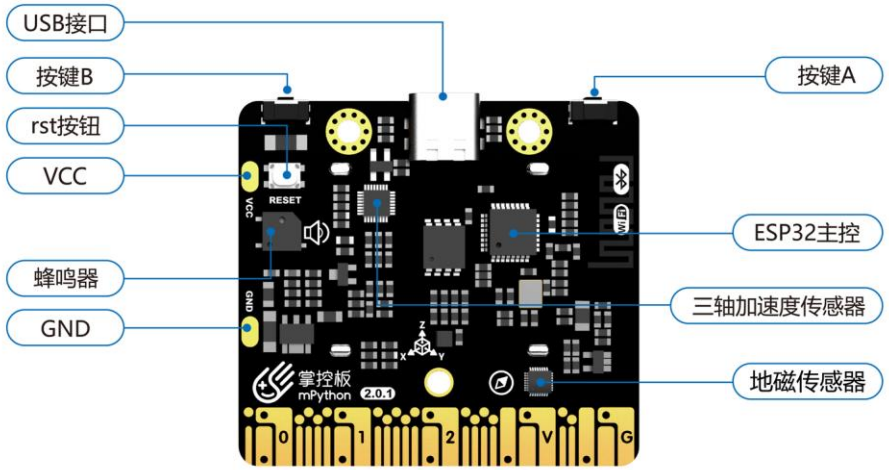
- 由于是开源硬件，因此掌控板的元件布局与引脚定义可在官网直接查询得到。另外，由于该板被定位于MicroPython微控制器板，因此可以很好的支持基于MicroPython的编程。下图是元件布局的图示：

  Since it is an open-sourced hardware, you can find the layout of the components and pin definitions on their official website (illustrated in the figures below). In addition, because the board is aligned with the requirement of MicroPython microcontroller board, it supports the MicroPython-based programming paradigm well.

元件布局正面



元件布局背面



掌控板
HandBit

- 尽管掌控板可以无缝地基于Arduino IDE进行开发，但用于教育时基于Python的开发可以降低使用难度。因此，需要掌控板运行MicroPython环境以便通过Python脚本操控掌控板。尽管可以用米思齐开发环境初始化固件，但推荐使用官方提供的固件。

  While it is feasible and actually seamless to program HandBit via the Arduino IDE, the education-oriented applications tend to leverage Python to relieve the difficulty in practice. Therefore, HandBit needs to run the MicroPython environment in order to be controlled via Python scripts. Although the firmware can be initialized with the Mixly IDE, it is recommended to use the official provided firmware.

- 盛思提供了基于MicroPython的mPython编程环境，但由于其更新速度落后于乐鑫的更新速度，其提供的刷固件的方法存在一定的问题，因此我们可以下载乐鑫的IDF工具集，配合盛思提供mPython固件进行初始化。

  Labplus provides the mPython package which is based on MicroPython, however, its update tends to lag behind Espressif's update, the original way to flash the bin file of Labplus might render some problem. Here we download Espressif's IDF toolset and together with Espressif's mPython firmware for initialization.

掌控板
HandBit

- IDF可以直接从乐鑫的官网下载安装。之前，ESP-IDF常作为命令行工具使用。同时，乐鑫也提供了与图形化IDE如Arduino IDE、Visual Studio Code（通过PlatformIO）集成的方式。在新的版本中，乐鑫也提供了基于图形化的集成开发环境。如果我们不准备基于IDF进行开发，则安装命令行工具就好。其下载网址为：

  IDF can be downloaded and installed directly from Espressif's official website. Conventionally, ESP-IDF was often used in a command-line way, although Espressif also provides the integration with graphical IDEs such as Arduino IDE and Visual Studio Code (via PlatformIO). In the new version, Espressif also provides a graphical integrated development environment. If we are not going to intensively develop based on IDF, just install the command line tool is enough. The webpage is as follows:

  https://dl.espressif.cn/dl/esp-idf/

- 盛思提供的mPython的固件可以从如下网址下载，推荐下载最新版本：

  The bin file which contains firmware provided by Labplus can be downloaded as below, the newer version the better:

  https://mpython.readthedocs.io/zh/master/release.html#release

- 注意上面这些信息可能会在未经提示的情况下更新。

  The above information might get updated without notification.

掌控板
HandBit

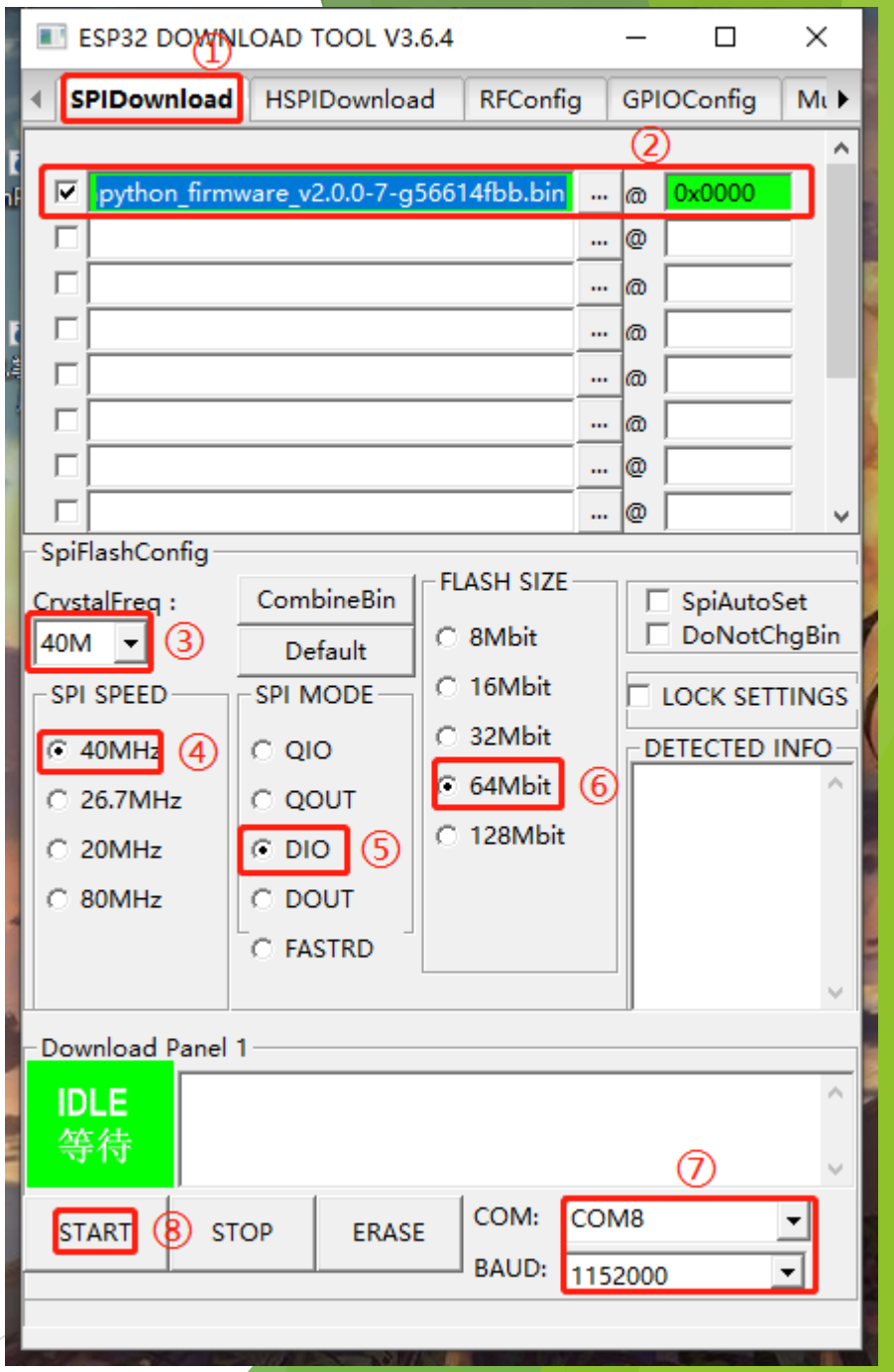
- 虽然盛思提供的初始化固件的方式有些问题，但初始化时的一些参数设置还是需要记下来，以便在用IDF工具进行初始化时使用。其相关参数如右图所示，特别注意要设置地址是0x00：

  Although the way provided by Labplus to initialize the firmware might not be workable, however, it is still necessary to reference to the parameter setting when flashing the firmware with the IDF tool. The relevant parameters are shown in the figure on the right, pay special attention to setting the address 0x00:

- 然后，用户从安装程序中，打开IDF命令行，即可执行刷固件操作，注意选择正确的COM口：

  Then, the user launches the IDF command line from the installed program to flash the firmware by choosing the appropriate COM:

```
esptool.py --chip esp32 --port COM? --baud 460800
write_flash -z 0x0000 mpython_v2.0.1.bin
```



ESP32-CAM

- ESP32-CAM是基于ESP32的常用模块，集成的摄像头为OV2640，同时还集成microSD卡接口。

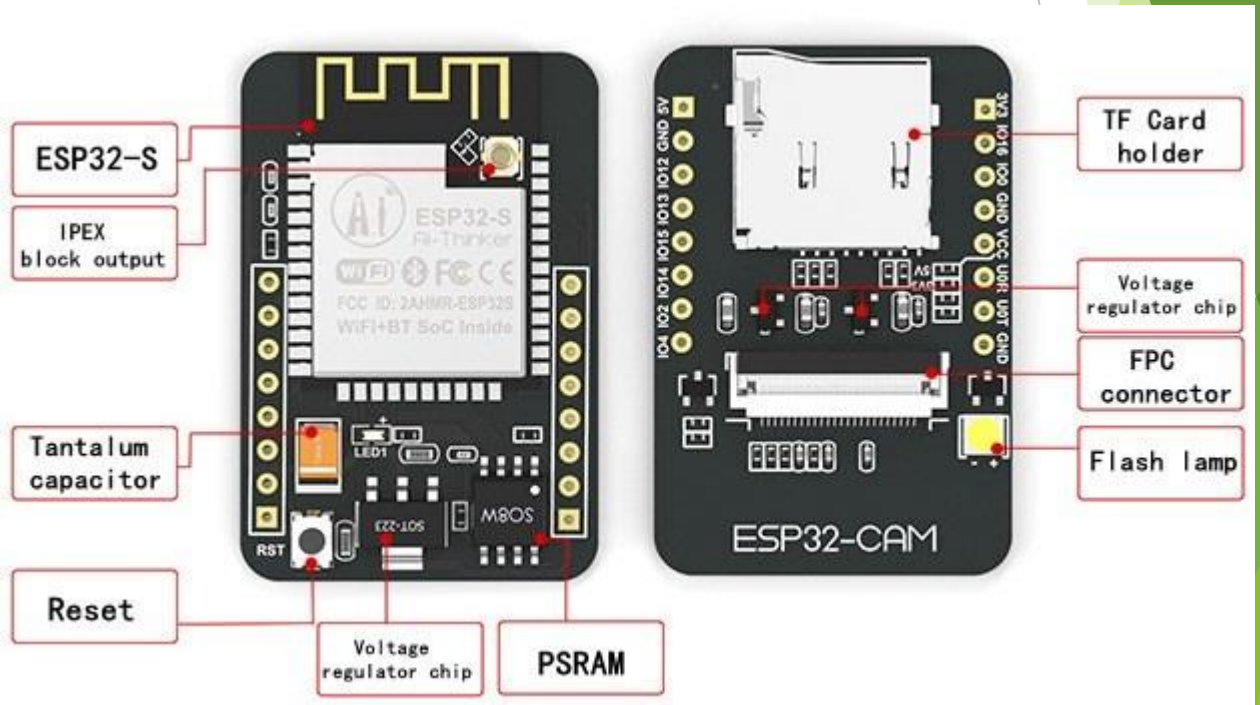
  ESP32-CAM is a popular module based on ESP32. The integrated camera is OV2640, and the board also integrates a microSD card interface.

- 其输入输出管脚为ESP32的子集，具有板载天线，同时提供了外接天线接口。

  Its input and output pins are a subset of ESP32, which has an onboard antenna and provides an external antenna interface.

- ESP32-CAM可以工作在3.3V或5V，但由于没有USB接口，烧录程序时需要通过USB转串口的转接板进行烧录。

  ESP32-CAM can work at 3.3V or 5V, but since there is no USB port, it needs to be programmed with an adapter with USB to Serial module.



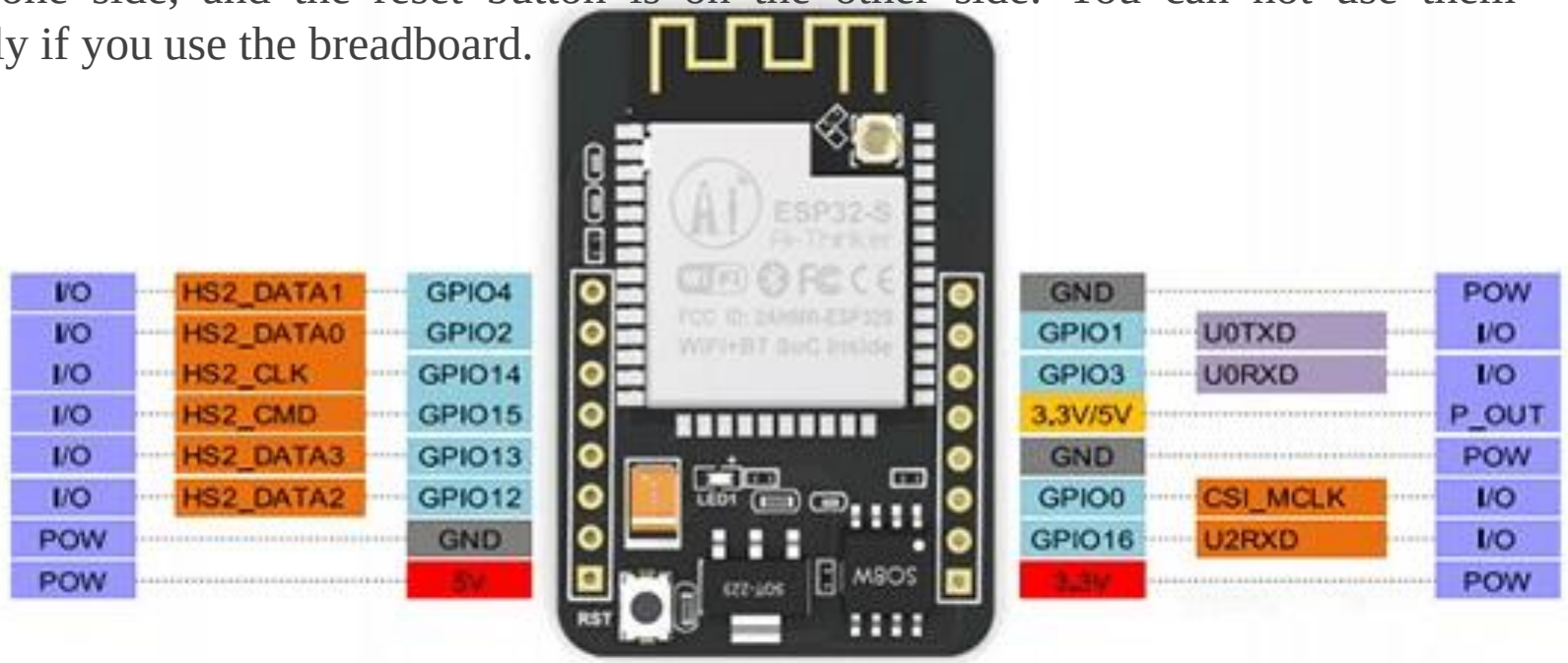
ESP32-CAM

- ESP32-CAM的管脚如下图所示：

  The pinouts of ESP32-CAM are shown as follows:

- ESP32-CAM上的元器件在板子的两面，因此不容易使用面包板，因为一面有摄像头，另一面有重置按钮，不能兼顾。

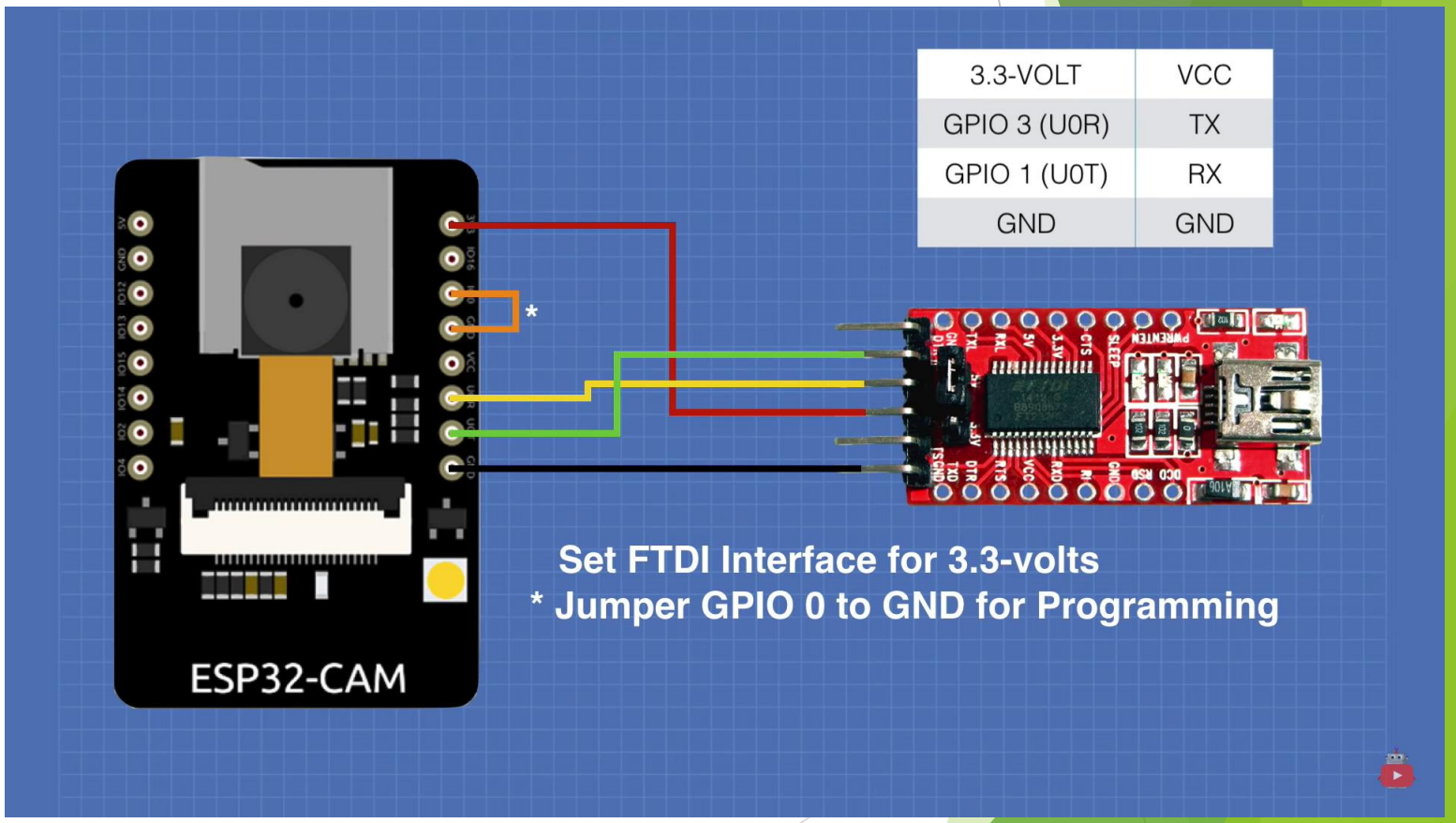
  The components of ESP32-CAM are on both side of the board, for example, the camera is mounted on one side, and the reset button is on the other side. You can not use them simultaneously if you use the breadboard.



ESP32-CAM

- 如前所述，由于ESP32-CAM本身没有USB接口，因此需要用具有USB转串口的转接板来进行程序的烧录。其管脚的连接方式如右图所示，注意需要将IO0管脚与地短接，以便开发板从下载模式启动进行烧录。

  As mentioned earlier, since ESP32-CAM itself does not have a USB interface, it is necessary to use an adapter board with a USB-to-serial port to burn the program. The connection method of its pins is shown in the figure on the right. Note that the IO0 pin needs to be short-circuited with the ground so that the development board can be started from the download mode for programming.



ESP32-CAM

- ESP32是只有少量RAM的小型CPU，我们通常说的“4MB”或“8MB”指的是flash的大小，而不是RAM的大小。ESP32的RAM固定为512KB，其中大约200KB用于IRAM缓存/代码部分，剩下大约320KB用于程序内存，其中一半可用于动态分配。

  ESP32 is a small CPU that has a small amount of RAM. The usual "4MB" or “8MB” in ESP32 refers to the size of flash, not the size of RAM. The ESP32 is fixed at 512KB, roughly 200KB of which is used by IRAM cache/code sections, leaving around 320KB for program memory, half of which is available for dynamic allocation.

- 我们知道堆是用于程序分配存储空间的内存池。虽然通常我们不能扩展ESP32上的堆，但我们可以以外部 PSRAM 的形式添加额外的存储——通过 SPI 访问的额外 RAM。

  We know the heap is the pool of memory available to your program to allocate storage. Generally we cannot extend the heap on the ESP32. However, we can add additional storage in the form of external PSRAM - additional RAM that is accessed via SPI.

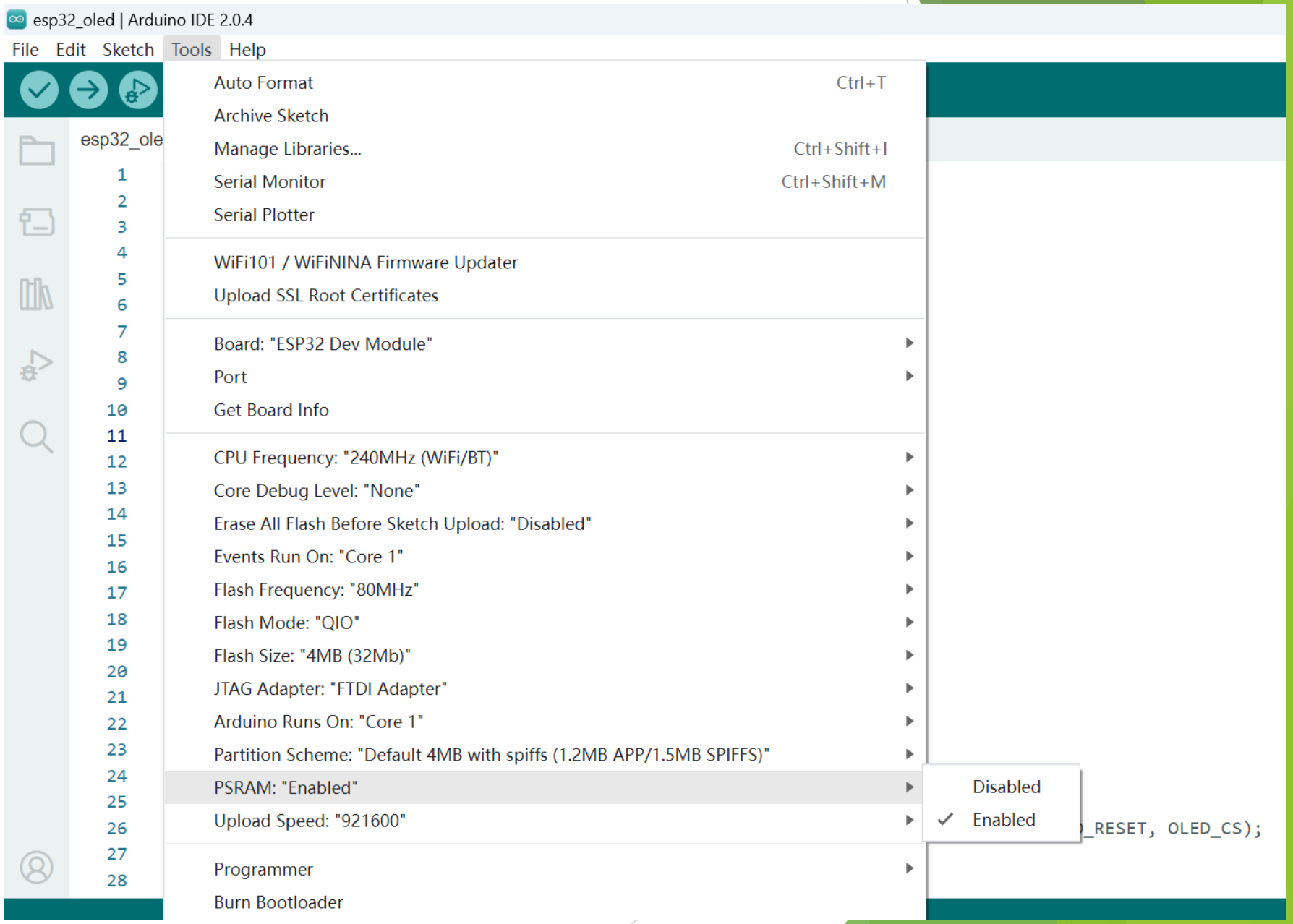
- PSRAM比内部RAM慢，并且对其使用有一些限制，例如，调用堆栈不能存在于PSRAM中，某些DMA缓冲区不能位于PSRAM中。

  PSRAM will be slower than internal RAM and will have some restrictions on its use, for instance, call stacks cannot live in PSRAM and certain DMA buffers cannot be located there.

ESP32-CAM

- 我们常用的ESP32-CAM模型是CAMERA_MODEL_AI_THINKER，并且有PSRAM模块，但在Arduino IDE中使用中，推荐我们显式将其打开：

  The commonly-used ESP32-CAM is of the model CAMERA_MODEL_AI_THINKER, which has the PSRAM on board. It is recommended to explicitly enable the PSRAM especially programming via Arduino IDE:



ESP32-CAM

- ESP32-CAM搭载的是OV2640图像传感器。它于2005年由OmniVision发布，尽管于2009年停产，但在今天的产品中仍然大量使用。作为一款SOC传感器，OV2640具有片上ISP，可以在小的传感器封装中进行自动曝光、自动白平衡等。OV2640最重要的特性是硬件 JPEG编码器，它减轻了微控制器的处理压力并减少了内存占用。同时，若使用较低的图像分辨率或较高的压缩率，输出的JPEG图像尺寸将更小，可以方便地在微控制器内部 RAM存储器中存储和处理。这些特性使该图像传感器在物联网相机应用中广受欢迎。

  OV2640 was released by OmniVision in 2005. Although it discontinued in 2009, yet it is still used a lot in today's products. As a SOC sensor, the OV2640 has an on-chip ISP which can do auto-exposure, auto-white balance in a packed sensor to produce nice images. The eminent feature of OV2640 is the hardware JPEG encoder, it off-loads the processing power and reduces memory footprint usage from the microcontrollers. With lower image resolution or higher compression ratio, the output JPEG image size will be even smaller, that can be easily stored and processed in microcontrollers internal RAM memory. In conclusion, the OV2640 image sensor unique and popular for IoT camera applications at all times.

ESP32-CAM

- OV2640图像格式通过以下语句配置：

  config.pixel_format = PIXFORMAT_XXX;

  具体可以是以下格式选项之一：

  PIXFORMAT_YUV422

  PIXFORMAT_GRAYSCALE

  PIXFORMAT_RGB565

  PIXFORMAT_JPEG

  然后，可以进一步设置帧大小、JPEG质量和帧缓冲区计数等。同时，可以根据带PSRAM还是不带PSRAM的相机来选择不同的设置。

- OV2640 image format could be configured via the following statement:

  config.pixel_format = PIXFORMAT_XXX;

  The image format can be one of the following options:

  PIXFORMAT_YUV422

  PIXFORMAT_GRAYSCALE

  PIXFORMAT_RGB565

  PIXFORMAT_JPEG

  Then, we can further set the frame size, jpeg quality and framebuffer count. Meantime, different settings can be applied based on that a camera with PSRAM or without PSRAM.

ESP32-CAM

- 帧大小可以设置为以下选项之一：

  FRAMESIZE_UXGA (1600 x 1200)

  FRAMESIZE_QVGA (320 x 240)

  FRAMESIZE_CIF (352 x 288)

  FRAMESIZE_VGA (640 x 480)

  FRAMESIZE_SVGA (800 x 600)

  FRAMESIZE_XGA (1024 x 768)

  FRAMESIZE_SXGA (1280 x 1024)

- 图像质量可以是0到63之间的数字，数字越小意味着质量越高。然而，数值非常小的图像质量，特别是在较高分辨率下可能会使ESP32-CAM崩溃或可能无法正确拍摄照片。

- The frame size can be set to one of these options:

  FRAMESIZE_UXGA (1600 x 1200)

  FRAMESIZE_QVGA (320 x 240)

  FRAMESIZE_CIF (352 x 288)

  FRAMESIZE_VGA (640 x 480)

  FRAMESIZE_SVGA (800 x 600)

  FRAMESIZE_XGA (1024 x 768)

  FRAMESIZE_SXGA (1280 x 1024)

- The image quality or JPEG quality can be a number between 0 and 63. A lower number means a higher quality. However, very low numbers for image quality, specially at higher resolution can make the ESP32-CAM to crash or it may not be able to take the photos properly.

ESP32-CAM

- OV2640的其它设置包括亮度、对比度、饱和度、白平衡、曝光等，读者可以自行检索手册并设置。

  Other settings of OV2640 include brightness, contrast, saturation, white balance, exposure, and more, the users can consult the datasheet and configure them.

- ESP32-CAM有Flash LED，是通过管脚GPIO4进行驱动。具体是由GPIO4输出接到S8050基极，驱动串接在S8050集电极的LED发光。在某些情况下不需要该LED时，可将其关闭或变暗，这通过让GPIO4输出PWM并调节占空比可以实现。

  ESP32-CAM has Flash LED, which is driven by pin GPIO4. Specifically, the GPIO4 output is connected to the base of the S8050 to drive the LEDs connected in series to the collector of the S8050 to emit light. In some cases, when the LED is not needed, it can be turned off or dimmed, which can be achieved by letting GPIO4 output PWM and adjust the duty cycle.

- 如果ESP32板和FTDI编程器无法建立建立串行通讯，则需要仔细检查它们之间的连接。有些编程器需要交叉连接ESP32板与FTDI编程器的TX与RX，而有些需要直连。如果一种有问题的话，可以试试另一种。

  If the ESP32-CAM is not able to establish a serial communication with your computer, Double-check the connections between your ESP32 board and the FTDI programmer. Some programmer requires cross-link, aka., RX goes to TX and TX goes to RX. However, some requires direct link.

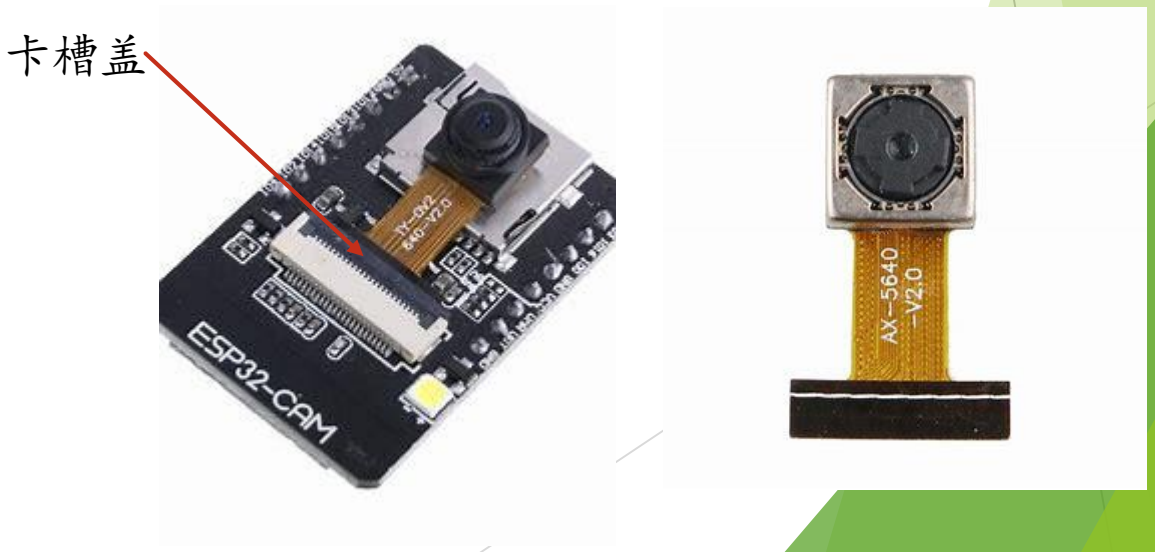
ESP32-CAM

- 由于OV2640为定焦摄像头，在远景监控等方面有较好的应用，当需要采集较近距离的图像，如进行文字识别时，不是最优，因此可将OV2460替换为具有较好性能的OV5640。该摄像头具有自动对焦功能，输出图像格式与帧率也较为丰富。

  OV2640 camera is only suitable for applications such as long-range monitoring due to the fixed focus. However, it is not the optimal choice for applications such as text recognition since it is required to capture images at close range. In this scenario, it can be replaced with the high-performance OV5640 camera. This camera has autofocus function and supports a variety of output image formats at different frame rates.

- 更换摄像头时，需要将紧固摄像头的卡槽打开，然后将摄像头排线的金手指插入并压紧。

  When replacing the camera, first open the slot that locks the camera, and then insert the gold finger end of the camera cable and press it firmly.



ESP-Prog

- 在程序设计时，特别是物联网项目涉及的嵌入式编程，有些时候可能通过串口打印便能解决存在的问题，但有些时候需要进行深入地调试工作才能定位问题。而ESP-Prog便是进行程序调试的利器。

  In programming, especially embedded programming involved in IoT projects, sometimes defects can be sorted out via information from serial printing, but sometimes in-depth debugging is required to locate the bug. ESP-Prog is a powerful tool to assist debugging.

- ESP-Prog是乐鑫开发的调试工具，具有固件自动下载、串口通信、JTAG 在线调试等功能。 ESP-Prog 只需使用一根 USB 数据线即可轻松连接到 PC，然后，PC 可以通过端口号识别开发板的下载和JTAG接口（功能）。考虑到不同用户板的供电电压可能不同，ESP-Prog接口可以通过排针提供5V或3.3V供电，以保证电源兼容性。

  ESP-Prog is a debugging tool developed by Espressif, with functions including automatic firmware downloading, serial communication, and JTAG online debugging. ESP-Prog can easily connect to a PC with the use of only one USB cable. Then, the PC can identify the board’s downloading and JTAG interfaces (functions) by their port numbers. Given that the power supply voltage may vary on different user boards, either of the ESP-Prog interfaces can provide the 5V or the 3.3V power supply through pin headers, in order to ensure power compatibility.

ESP-Prog

- JTAG（联合测试行动小组的英文名称的首字母）是一个行业标准，用于在制造后验证设计和测试印刷电路板。作为数字仿真的补充工具，JTAG在电子设计自动化(EDA)中规范了在片上建立测试机制的标准。它指定使用专用调试端口实现串行通信接口以实现低开销访问，而无需直接从外部访问系统地址和数据总线。该接口连接到片上测试访问端口(TAP)，该端口执行状态协议以访问一组测试寄存器，这些寄存器显示芯片逻辑电平和各个部分的设备功能。

  JTAG (Joint Test Action Group) is an industry standard for verifying designs and testing printed circuit boards after manufacture. JTAG implements standards for on-chip instrumentation in electronic design automation (EDA) as a complementary tool to digital simulation. It specifies the use of a dedicated debug port implementing a serial communications interface for low-overhead access without requiring direct external access to the system address and data buses. The interface connects to an on-chip Test Access Port (TAP) that implements a stateful protocol to access a set of test registers that present chip logic levels and device capabilities of various parts.

- 今天，JTAG被用作访问集成电路子模块的主要方式，成为调试可能没有任何其他具有调试功能的通信通道的嵌入式系统的基本机制。

  Today JTAG is used as the primary means of accessing sub-blocks of integrated circuits, making it an essential mechanism for debugging embedded systems which might not have any other debug-capable communications channel.

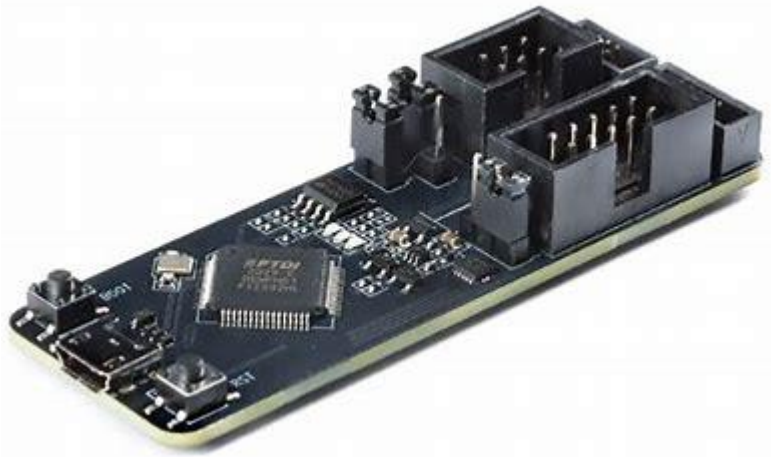
ESP-Prog

- 在大多数系统上，基于JTAG的调试可以中断CPU复位后的第一条指令，这让其能够调试在任何设置生效之前的包括启动阶段的程序。通过JTAG在线仿真器直接访问目标CPU的片上调试模块，软件开发人员在需要时直接在机器指令级别，或在高级语言源代码层面调试嵌入式系统的软件。

  On most systems, JTAG-based debugging is available from the very first instruction after CPU reset, letting it assist with development of early boot software which runs before anything is set up. Via JTAG in-circuit emulator to directly access on-chip debug modules inside the target CPU, software developers to debug the software of an embedded system directly at the machine instruction level when needed, or in terms of high level language source code.

- 乐鑫提供了ESP-Prog作为ESP32平台的开发和调试工具。除了JTAG在线调试功能，ESP-Prog还可用于自动固件下载、串行通信。

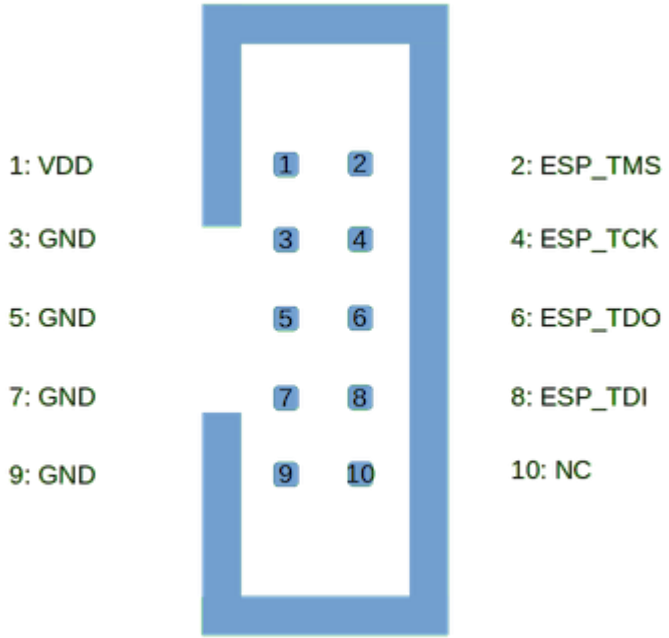
  Espressif provides ESP-Prog as the tool for development and debugging. Besides JTAG online debugging, its functions include automatic firmware downloading, serial communication, etc.



ESP-Prog

- ESP-Prog上关于JTAG功能接口十个引脚，与JTAG标准相关的有四个引脚，其图示与相应功能说明如下：

  A JTAG interface in ESP-Prog presents 10 pins. The description and diagram of four pins related to JTAG standard are as follows:

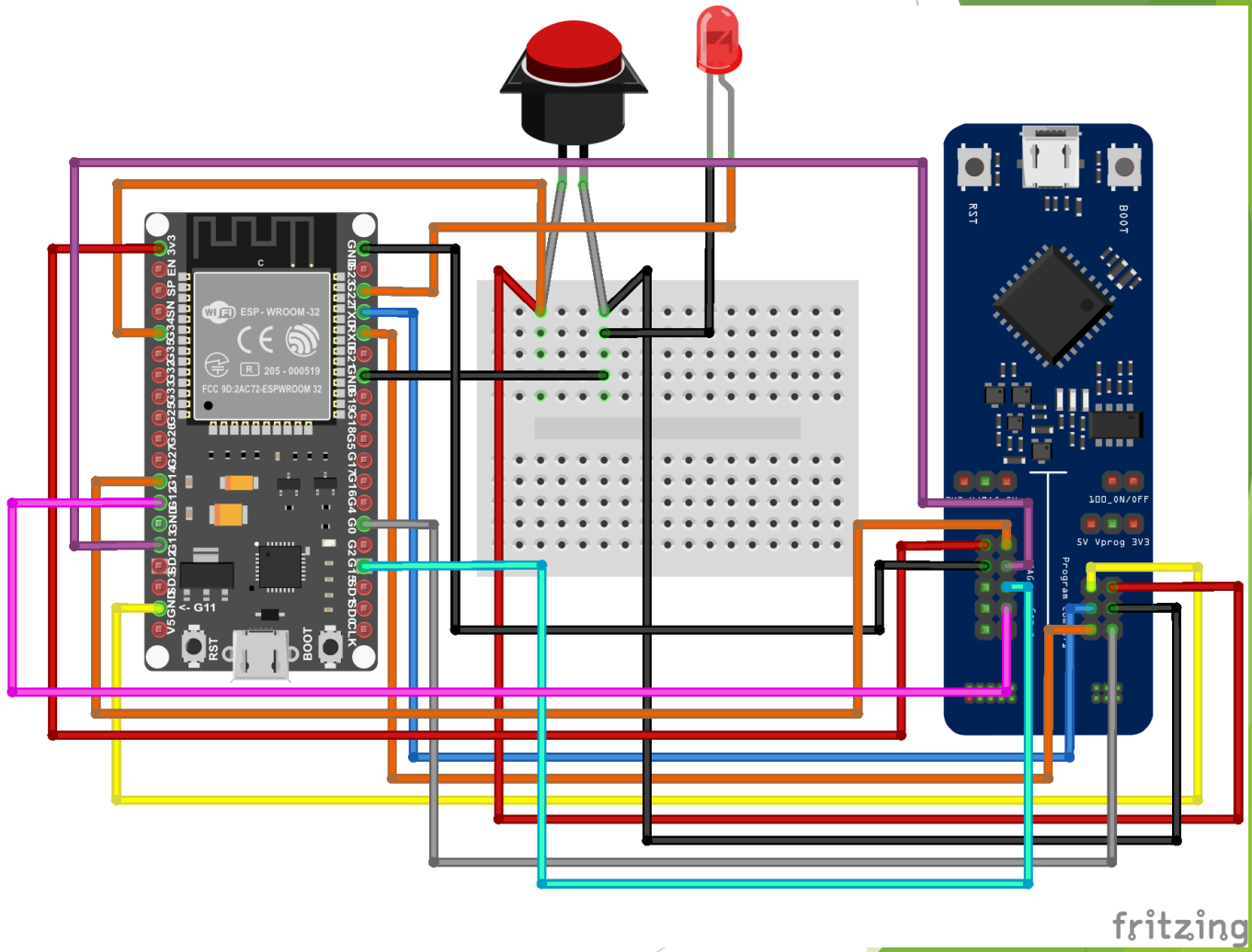


JTAG Pin	Description	ESP32 Pins
VDD		3.3V
GND		GND
TMS	Test Mode State	GPIO14
TCK	Test Clock	GPIO13
TDO	Test Data Out	GPIO15
TDI	Test Data In	GPIO12

ESP-Prog

- 由于ESP-Prog既可以进行调试，又可以进行烧录。一种典型的连接方式如右图所示。我们以当按键按下时，切换LED灯的明暗为例，来说明ESP-Prog的使用。

Since ESP-Prog can be used not only to debug the program, but also flash the board. A stereotypical wiring between ESP-Prog and ESP32 is demonstrated as in the right figure. We exemplify how to use ESP-Prog by considering an example, namely, to switch on/off the LED by pressing the button.



ESP-Prog

- 首先，建立存放ESP项目的文件夹，然后，将安装目录下examples\get-started\hello_world文件夹拷贝到新建的文件夹：

  First, create a folder that holds the ESP project. Then copy the illustrating example which locates under examples\get-started\hello_world of the installed directory into the newly created directory:

  mkdir E:\esp32-projects

  xcopy /e /i %IDF_PATH%\examples\get-started\hello_world hello_world

- 接下来，进入项目文件夹，进行初步配置与构建工程：

  Now enter the project directory, to configure and build the project initially:

  cd E:\esp32-projects\hello_world

  idf.py set-target esp32

  idf.py menuconfig

  idf.py build

- 然后，根据实际情况修改代码再次构建。

  Then rebuild the project after applying some modification to the source code.

ESP-Prog

- 接下来，按之前所述连接好所有连线之后，将ESP-Prog连到电脑上。此时，如果在设备管理器中仅出现COM设备，可能需要利用Zadig（https://zadig.akeo.ie/）工具更新驱动，使FDTI呈现为通用串行总线设备。不然在使用OpenOCD时，可能会出现无法打开设备的情况。

  Now, prepare the setup as illustrated previously and plug the ESP-Prog into the computer. If only COM devices appear in the device manager, you might have to update the device driver via Zadig (https://zadig.akeo.ie/), which enforce FDTI device as the generic USB device. Otherwise, the OpenOCD might fail to open the device upon launching.

- 在ESP-IDF环境中，启动OpenOCD：

  Launch OpenOCD in the ESP-IDF environment:

  openocd -f interface/ftdi/esp32_devkitj_v1.cfg -f board/esp-wroom-32.cfg

- 创建包含右侧内容的gdbinit文件，然后进行调试：

  Create the file named gdbinit with content as right, then launch the gdb:

  cd E:\esp32-projects\hello_world

  xtensa-esp32-elf-gdb -x gdbinit build\hello_world.elf

```
target remote :3333
set remote hardware-watchpoint-limit 2
mon reset halt
flushregs
thb app_main
c
```

ESP-Now

- ESP-NOW 是一种由乐鑫公司定义的无连接 Wi-Fi 通信协议。在 ESP-NOW 中，应用程序数据被封装在各个供应商的动作帧中，然后在无连接的情况下，从一个 Wi-Fi 设备传输到另一个 Wi-Fi 设备。 CTR 与 CBC-MAC 协议 (CCMP) 可用来保护动作帧的安全。ESP-NOW 广泛应用于智能照明、远程控制、传感器等领域。

  ESP-NOW is a kind of connectionless Wi-Fi communication protocol that is defined by Espressif. In ESP-NOW, application data is encapsulated in a vendor-specific action frame and then transmitted from one Wi-Fi device to another without connection. CTR with CBC-MAC Protocol(CCMP) is used to protect the action frame for security. ESP-NOW is widely used in smart light, remote controlling, sensor, etc.

- 可以通过乐鑫提供的库函数，使用ESP-Now。

  Users can use the ESP-Now protocol by harnessing library provided by Espressif.