

User Manual

Running Model Deployment

Once you have gone through the installation guide for the Model Deployment Program you can run the ESP-IDF 5.0 CMD as shown below and run the following commands to make an inference on a hardcoded test image.

Make sure you are in the model_deployment directory

```
cd model_deployment
```

Make sure that COMX is the current port that your ESP32 Chip is connected to i.e. COM5

```
idf.py set-target esp32
```

Check if the flash is configured to 4MB by typing

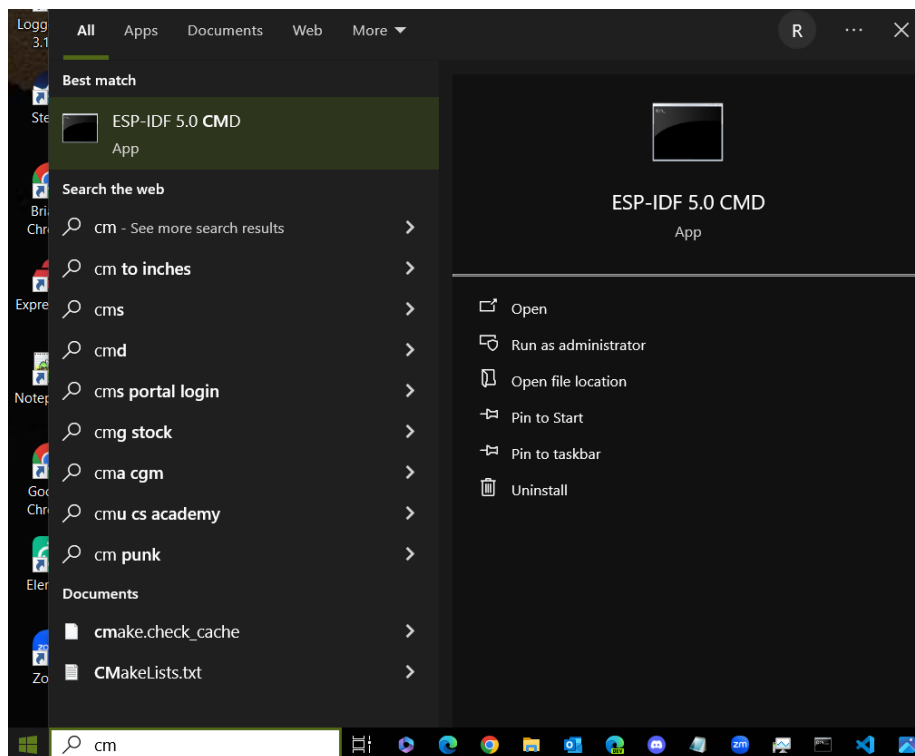
```
idf.py menuconfig
```

Then compile and run by doing the following.

```
idf.py build
```

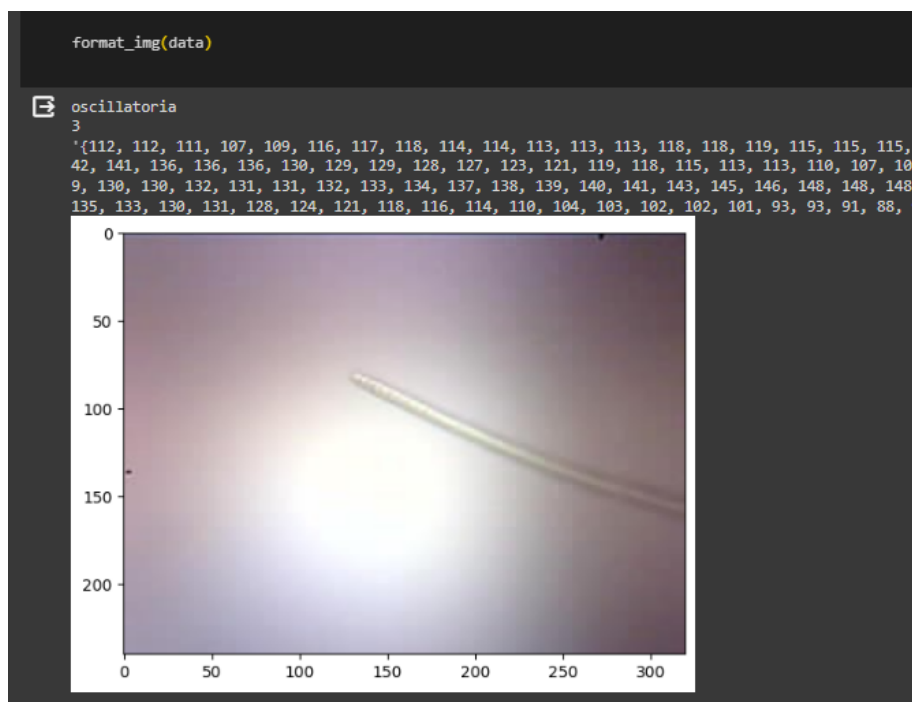
```
idf.py -p COMX flash
```

```
idf.py -p COMX monitor
```



How to Change The Sample Image

You can generate new example images that you can classify directly on the chip by doing the following. Run the notebook found in `model_deployment` and scroll to the bottom of the notebook under the heading “Testing”. There is some code there you can run to turn an image at the index you desire into a C++ array which you can copy and paste it as the example image in the `model_deployment` program.



This is how the output would look like from the google notebook. The C++ array generated are the numbers you see above the image.

To save this as an example image that the chip can classify you would take this C++ array and paste it here so that `example_element[]` is now equal to the new array you generated.

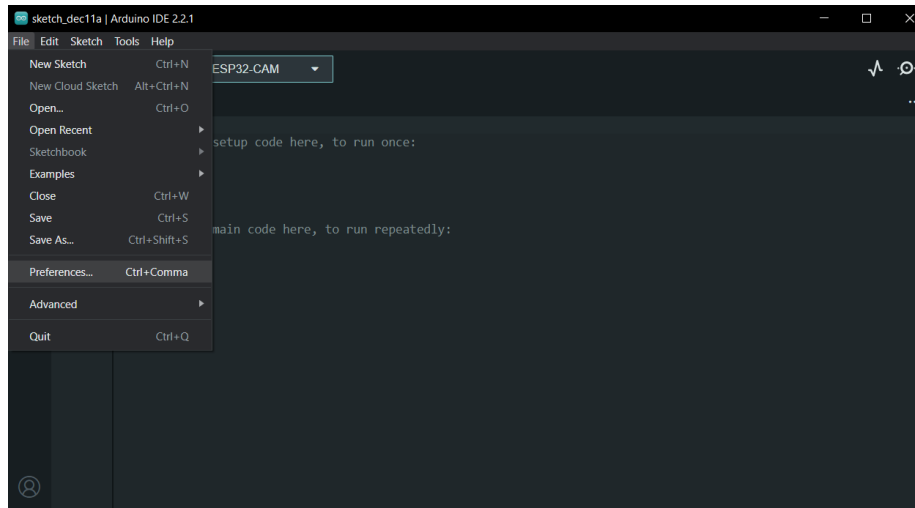
```

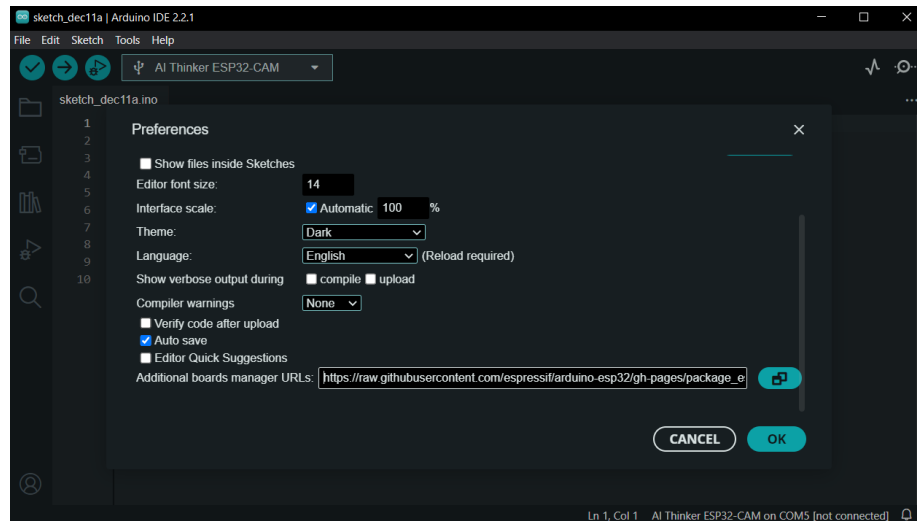
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include "esp_system.h"
4  #include "freertos/FreeRTOS.h"
5  #include "freertos/task.h"
6
7  #include "dl_tool.hpp"
8  #include "model_define.hpp"
9
10
11
12
13  __attribute__((aligned(16))) int16_t example_element[] = {61, 67, 83, 82, 97, 1
137, 135, 134, 134, 140, 139, 140, 140, 140, 144, 146, 146, 143, 143, 143,

```

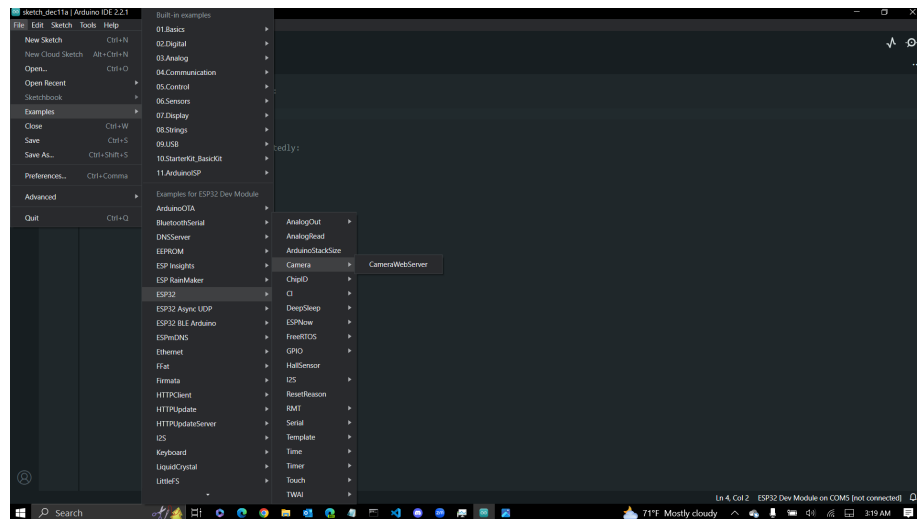
How to Run The Webserver

- First make sure that your ESP-32 chip is mounted onto the microscope and that it is connected to your local computer through a USB cable.
- To run the camera-web server onto the chip you would need to download the Arduino IDE to compile and flash the web server code onto the chip. You can install Arduino IDE from <https://www.arduino.cc/en/software>
- Go to file, preferences, and add the additional board manager url https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json





- To download the camera web server code you can clone our repository to access it <https://github.com/rdgbrian/cap-2-project-algae-detection>, and open the webserver code in your IDE. Or alternatively you can access it through the IDE. You can access it through the IDE by setting the current board to be the ESP32 Dev Module. Then selecting file->examples->ESP32->Camera->CameraWebServer



Once selected be sure to use the AI Thinker model and not the ESP 32 Eye cam model:

```

13 // =====
14 // Select camera model
15 // =====
16 // #define CAMERA_MODEL_WROVER_KIT // Has PSRAM
17 // #define CAMERA_MODEL_ESP_EYE // Has PSRAM
18 // #define CAMERA_MODEL_ESP32S3_EYE // Has PSRAM
19 // #define CAMERA_MODEL_M5STACK_PSRAM // Has PSRAM
20 // #define CAMERA_MODEL_M5STACK_V2_PSRAM // M5Camera version B Has PSRAM
21 // #define CAMERA_MODEL_M5STACK_WIDE // Has PSRAM
22 // #define CAMERA_MODEL_M5STACK_ESP32CAM // No PSRAM
23 // #define CAMERA_MODEL_M5STACK_UNITCAM // No PSRAM
24 #define CAMERA_MODEL_AI_THINKER // Has PSRAM
25 // #define CAMERA_MODEL_TTGO_T_JOURNAL // No PSRAM
26 // #define CAMERA_MODEL_XIAO_ESP32S3 // Has PSRAM
27 // ** Espressif Internal Boards **
28 // #define CAMERA_MODEL_ESP32_CAM_BOARD
29 // #define CAMERA_MODEL_ESP32S2_CAM_BOARD
30 // #define CAMERA_MODEL_ESP32S3_CAM_LCD
31 // #define CAMERA_MODEL_DFRobot_FireBeetle2_ESP32S3 // Has PSRAM
32 // #define CAMERA_MODEL_DFRobot_Romeo_ESP32S3 // Has PSRAM

```

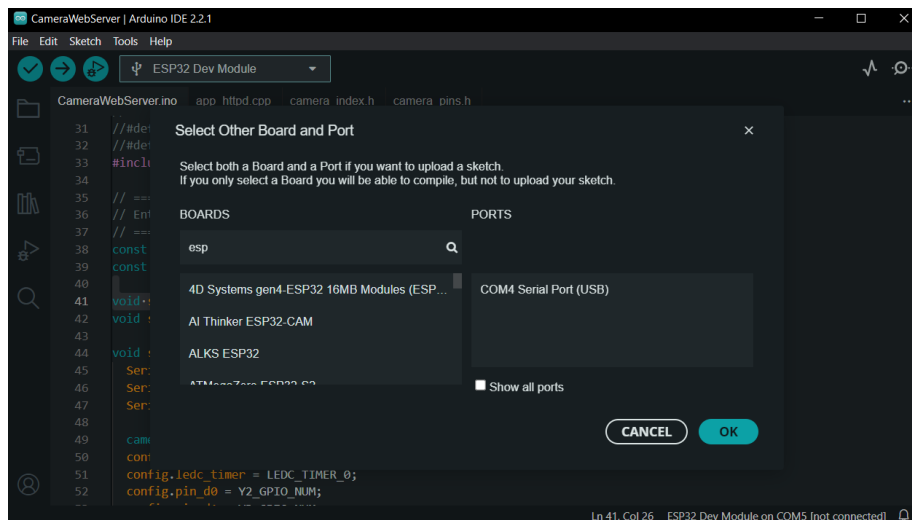
You will then need to fill out the required credentials which will be the internet connection name under ssid and password under the password

```

38 const char* ssid = "*****";
39 const char* password = "*****";

```

- Once you are ready to run your program. Check if the correct board and port (USB connection) is selected by pressing the button that has the usb symbol in the top middle of your screen.



- Then compile and flash your code onto the chip. Open the serial monitor

in the IDE to see the input from the chip. This input will provide a link to access the live camera input. Copy and paste this link onto your browser to interact and view the live video.