

Instalation Guide

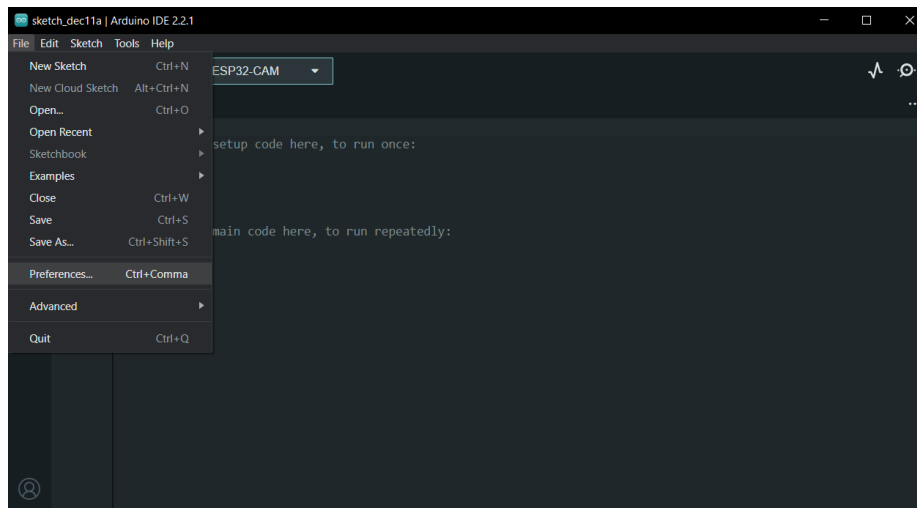
Model Development Notebook (webserver)

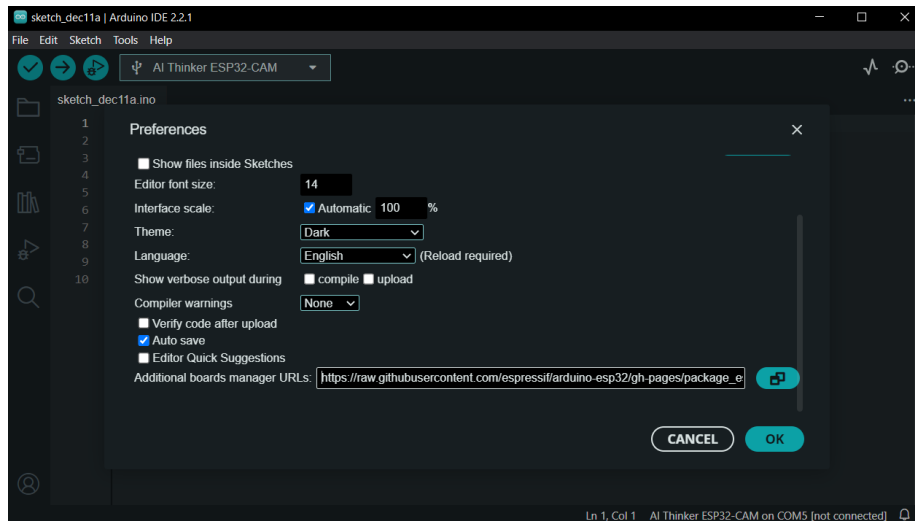
open the model_deployment directory and move the files there (the notebook and algea_dataset) to a google drive directory. With some minor changes to the notebook which is just updating the path of the the dataset depending on where you put it the code would be working if you run it on google colab. To run simply click on the file twice and google colab should automantically open.



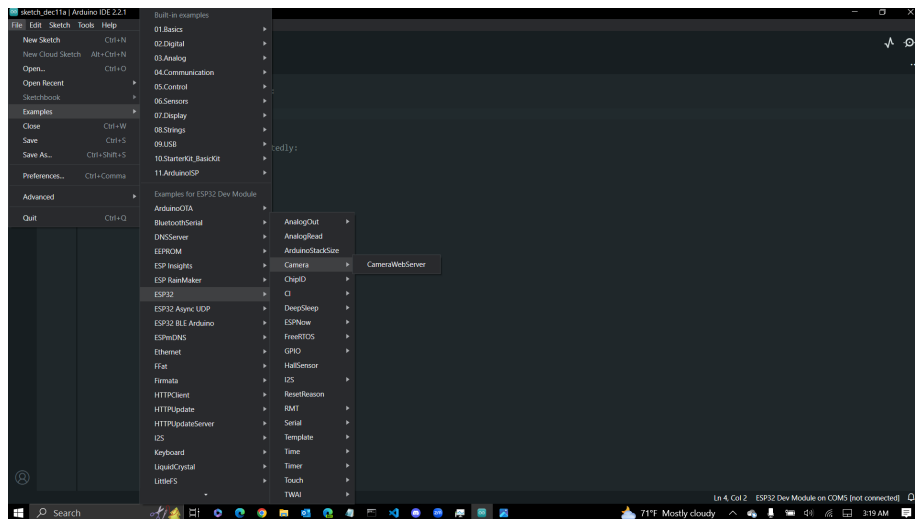
Camera Web App

- 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

```

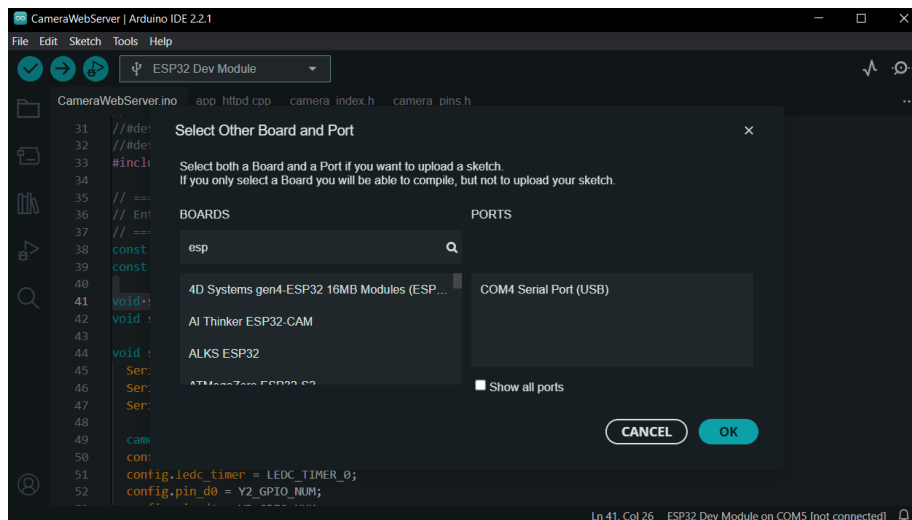
You will then need to fill out the required credentials which will be the internet connection name under ssid and password under the password

```

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

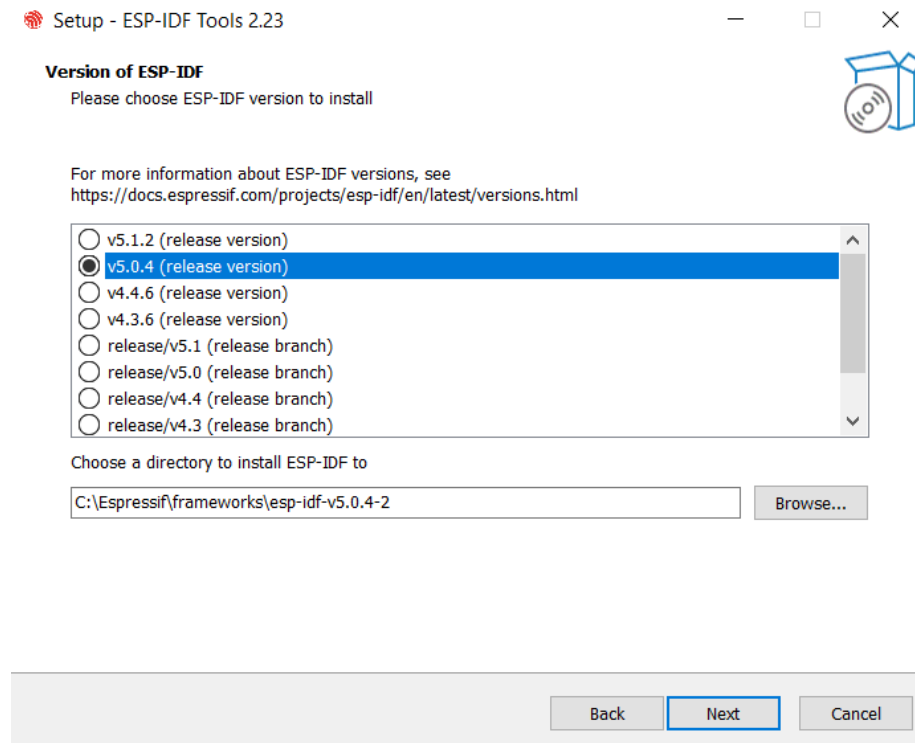
in the IDE to see the input from the chip. This input will provide a link to access the live camera input. Copy and paste this link onto your browser to interact and view the live video.

Model Deployment Program (model_deployment)

The model deployment code runs using ESP-IDF instead of the Arduino IDE. To install ESP-IDF download the installer in here: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/#manual-installation>. The model deployment program **ONLY** works on ESP-IDF version 5.0.x. This program is known to work on version 5.0.4. If you install the newer versions like 5.1.2 you will encounter issues. In this instalation guide I will.

ESP-IDF Windows instalation

Grab the installer here <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/windows-setup.html>. Go though all the steps and leave everything default **EXCEPT** remeber to install the version 5.0.4



ESP-IDF Linux instalation

- Ubuntu and Debian

```
sudo apt-get install git wget flex bison gperf python3 python3-pip python3-venv
cmake ninja-build ccache libffi-dev libssl-dev dfu-util libusb-1.0-0
```

- CentOS 7 & 8

```
sudo yum -y update && sudo yum install git wget flex bison gperf
python3 cmake ninja-build ccache dfu-util libusb
```

- Arch

```
sudo pacman -S --needed gcc git make flex bison gperf
python cmake ninja ccache dfu-util libusb
```

Downloading ESP-DL

Finally we need to download ESP-DL in `model_deployment\components`. clone the ESP-DL from the Github repository.

```
cd model_deployment\components
git clone --recursive https://github.com/espressif/esp-dl.git
```

Model Quantize Program (model_quantize)

The code found in this directory is used to quantize the onnx file that was produced in the Model Development Notebook. To run this you will first need to install anaconda which is a distribution of python that will help you manage the packages that you want to install. Then install the appropriate packages and python version listed.

Create a new environment in anaconda and install the following packages and python version.

Module	How to install
Python == 3.7	
Numba == 0.53.1	<code>pip install Numba==0.53.1</code>
ONNX == 1.9.0	<code>pip install ONNX==1.9.0</code>
ONNX Runtime == 1.7.0	<code>pip install ONNXRuntime==1.7.0</code>
ONNX Optimizer == 0.2.6	<code>pip install ONNXOptimizer==0.2.6</code>

To quickly configure your conda environment and install the required packages run the following commands: First make sure you are in the `model_quantize` directory.

```
cd model_quantize
```

Then run these commands:

```
conda create -n <env-name> python=3.7
pip install -r requirements.txt
conda activate <env-name>
```

Anaconda Guide

After you have installed anaconda using the instructions bellow then you can create and manage your environments using the following comands. For more information on how to manage and set up your conda enviornment

To create an environement:

```
conda create -n <env-name> python=3.9
```

To use your environment you would type the following command:

```
conda activate <env-name>
```

In our project we have a file called `requierments.txt` which has the informations on what packages needs to be installed to use this file you would need to run the following comand and all the packages listed in the file would be automatically installed into your conda environment for you.

```
pip install -r requirements.txt
```

To use your enviroment you would need to run:

```
conda install -c conda-forge <package-name>
```

or if you prefer you can also install with pip

```
pip install <package-name>
```

When in doubt you can always search “pip install <package-name>” or “anaconda install <package-name>” in your browser and the appropriate way to install your package should pop up.

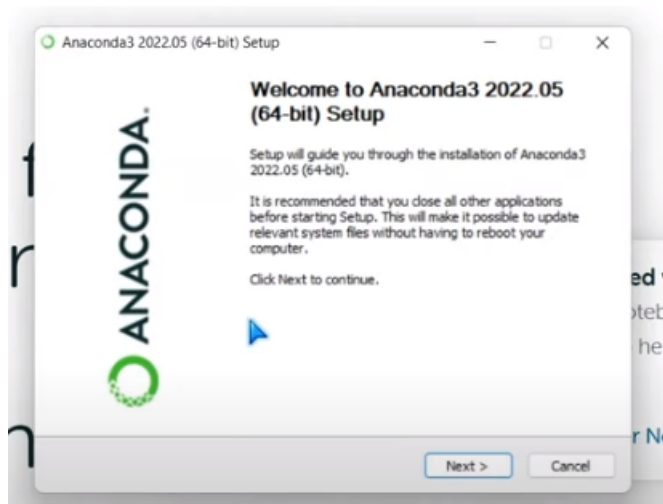
Once you’ve activated your enviornemnt you can add packages using either the following comands

Incase this is your first time using anaconda you can follow the instalation guide bellow.

Installing on windows

Download Anaconda

- Go to <https://www.anaconda.com/download> and download anaconda



The installer should look like this when you open it up. Go through it and you can leave all the settings on default. ##### Adding Anaconda To Path

where conda
where python

Notice where python.exe and conda.exe is installed. For example in my command line the following output was shown.

```
Command Prompt
Microsoft Windows [Version 10.0.19045.3693]
(c) Microsoft Corporation. All rights reserved.

C:\Users\rdgbr>where conda
C:\Users\rdgbr\anaconda3\Library\bin\conda.bat
C:\Users\rdgbr\anaconda3\Scripts\conda.exe
C:\Users\rdgbr\anaconda3\envs\pytorch\Library\bin\conda.bat

C:\Users\rdgbr>where python
C:\Python312\python.exe
C:\Users\rdgbr\anaconda3\python.exe
```

conda.exe would be in C:\Users\<user>\anaconda3 python.exe would be in C:\Users\<user>\anaconda3\Scripts

Finally edit your systems environment variables by adding the following directories we found above to path.

First you open System Properties. You can do this by searching

System Properties



Computer Name Hardware **Advanced** System Protection Remote

You must be logged on as an Administrator to make most of these changes.

Performance
Visual effects, processor scheduling, memory usage, and virtual memory

[Settings...](#)

User Profiles
Desktop settings related to your sign-in

[Settings...](#)

Startup and Recovery
System startup, system failure, and debugging information

[Settings...](#)

[Environment Variables...](#)

OK Cancel Apply

User variables for rdgbr

Variable	Value
IntelliJ IDEA Community E...	C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 20...
MYSQL_HOME	C:\ProgramData\MySQL\MySQL Server 8.0\bin
OneDrive	C:\Users\rdgbr\OneDrive
OneDriveConsumer	C:\Users\rdgbr\OneDrive
Path	C:\Program Files\MySQL\MySQL Shell 8.0\bin\;C:\Users\rdgbr\...
TEMP	C:\Users\rdgbr\AppData\Local\Temp
TMP	C:\Users\rdgbr\AppData\Local\Temp

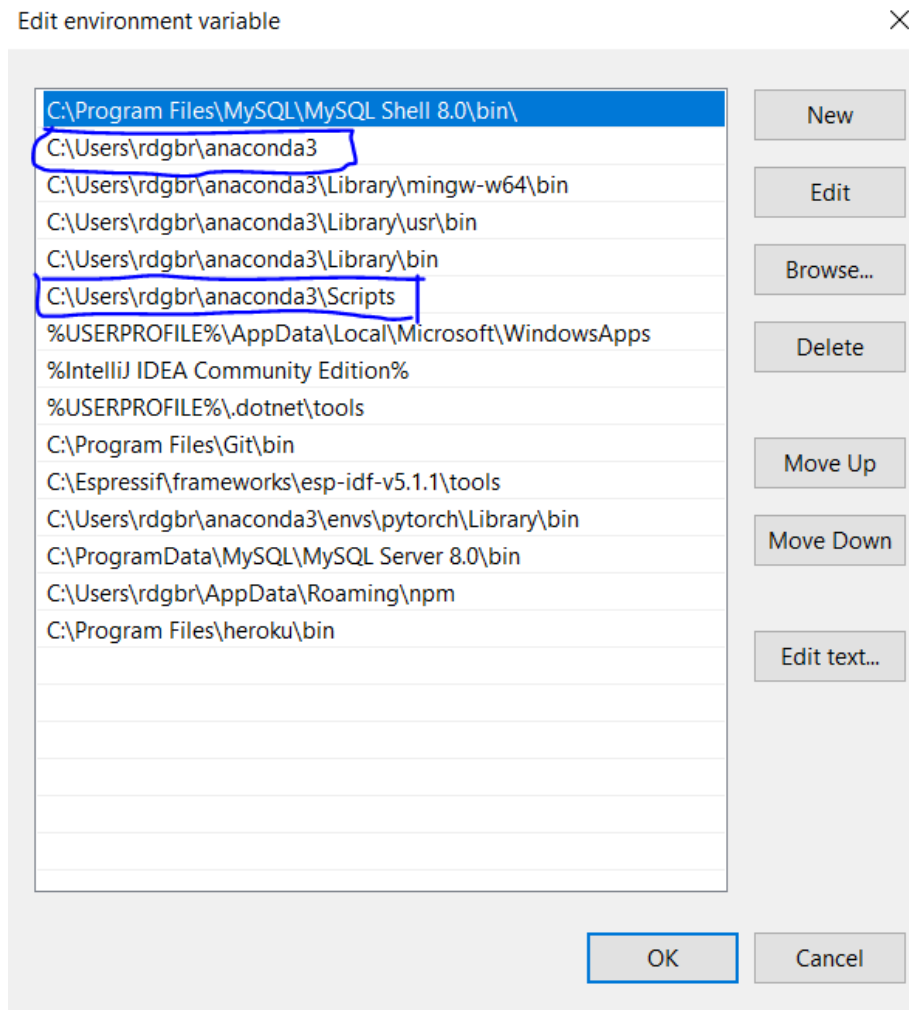
New... Edit... Delete

System variables

Variable	Value
Chocolatey\Install	C:\ProgramData\chocolatey
ComSpec	C:\WINDOWS\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	8
OS	Windows_NT
Path	C:\Program Files\Microsoft SDKs\Azure\CLI2\wbin\;C:\Python3...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC;.PY;.PYW
PROCESSOR_ARCHITECTURE	AMD64

New... Edit... Delete

OK Cancel



Installing on linux

Download Miniconda Installer

```
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh  
-O /opt/miniconda-installer.sh
```

Install Miniconda

```
bash /opt/miniconda-installer.sh
```

Miniconda3 will now be installed into this location:
/root/miniconda3

- Press ENTER to confirm the location
- Press CTRL-C to abort the installation
- Or specify a different location below

Just press ENTER and continue.

Do you wish the installer to initialize Miniconda3
by running conda init? [yes|no]
[no] >>>

Type 'yes', then hit ENTER. You should see this as an output.

you should add the path to your .bashrc file by running the command:

```
echo 'export PATH="/path/to/miniconda3/bin:$PATH"' >> ~/.bashrc
```

Its important to note that miniconda doesnt come with pip so you would have
do download it using the following comand.

```
sudo apt install python3-pip
```