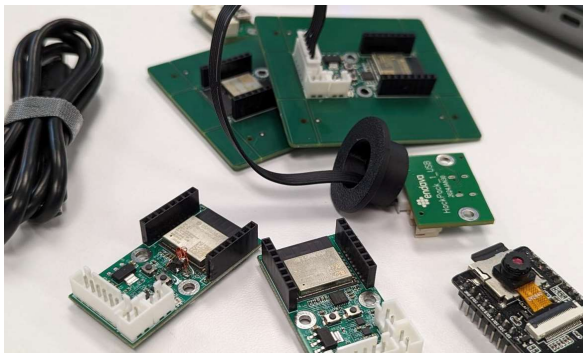


Introduction to the HackPack Launcher Firmware Setup and System Control Guide

Welcome to the setup guide for the HackPack Launcher, an advanced automated projectile launching system that combines machine vision and precise mechanics, controlled entirely through a web interface. This guide will walk you through installing the Arduino IDE, configuring necessary libraries, and setting up the launcher base and camera module co-processor.

The HackPack Launcher is a dynamic tool for both learning and demonstrating robotics and computer vision. It can autonomously detect and track faces, adjust its orientation, and execute commands through its web-based control interface, illustrating the possibilities within embedded systems despite resource limitations.

Additionally, this guide covers safe handling and operational procedures to ensure that you can confidently manage the launcher's automated features. Ideal for students and robotics enthusiasts, this guide aims to get your HackPack Launcher operational and inspire further exploration in advanced robotics.

Contents

General Operation and Safety Guidelines

Understanding the operation and safety precautions of the HackPack Launcher is essential for optimal performance and safety. Below are key guidelines that detail the launcher's operation and critical safety measures to prevent accidents and damage.

1) **Automated System Behavior**

   The HackPack Launcher operates automatically and may move unexpectedly in response to commands. Always handle with care, especially when powered on.

2) **Startup and Calibration**

   Power Connection: Connecting power starts a calibration sequence, moving components to calibrate sensors and stabilizers.

   Caution: Do not touch the launcher during startup as interference could affect performance.

   Subsequent Power-Ups: Calibration occurs with every startup after the initial firmware upload.

3) **System Stabilization and Orientation**

   The system position automatically stabilizes to a set orientation. Manual repositioning while powered is not recommended and will cause damage. User should use a command interface to reposition the unit by sending command to the unit.

4) **Mechanical Constraints and Movement**

   Hard Stops on Pan Axis: Prevents endless rotation and potential internal damage.

   USB Port Positioning: The USB port should face front when powering up, aligning the system for optimal operation.

5) **Handling and Maintenance**

   Move axes gently when powered off. Abrupt movements can harm the unit.

These guidelines ensure you understand how to operate and maintain the HackPack Launcher safely. Adhering to these instructions guarantees a safe and efficient experience with this advanced system.

Prerequisites for Setting Up the HackPack Launcher

Before beginning the software setup for the HackPack project, ensure the following:

1) **Operating System**

   Use a Windows 10 or later PC or laptop; Windows 11 is recommended for best performance.

   Mac systems are compatible but not documented here. Most steps are generally the same just Mac specific items are not noted.

2) **Launcher Base Assembly**

   Completely assemble the Launcher Base before starting the software installation. This ensures that you can immediately test and troubleshoot during the setup process. Follow the assembly instructions provided with your HackPack kit.

Installation of the Arduino IDE and Required Libraries

1) Software IDE Installation

a) Download and install the Arduino IDE from the official website:

https://www.arduino.cc/en/software

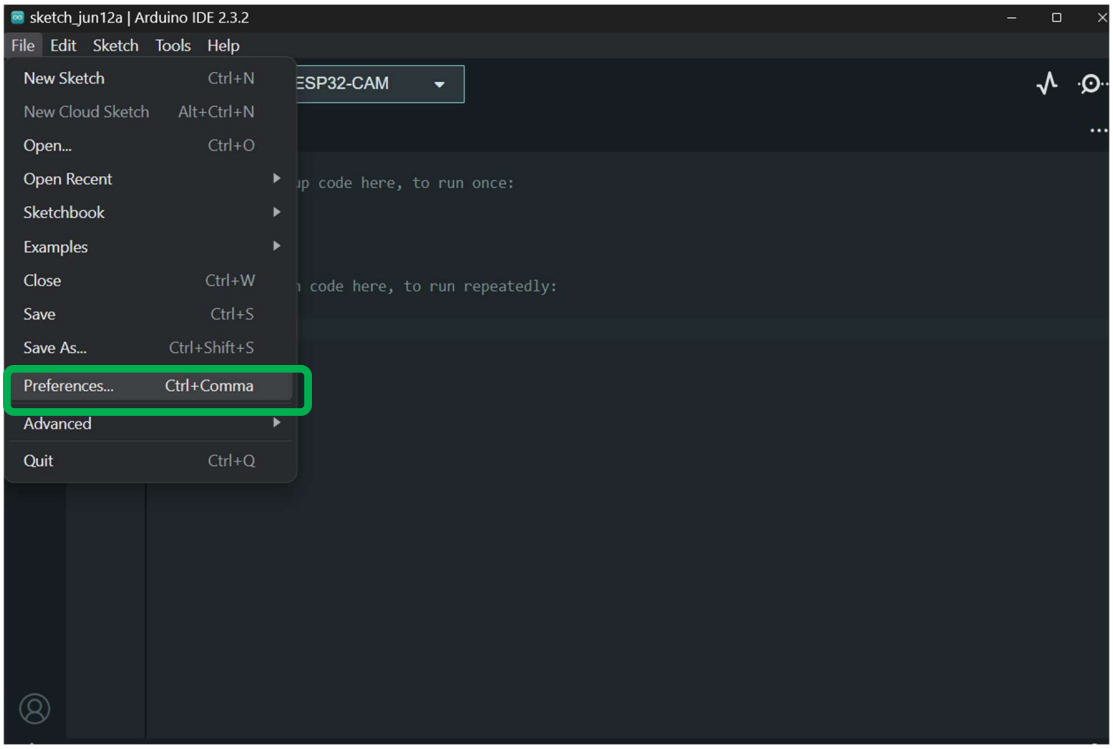
b) After installation, launch the Arduino IDE.

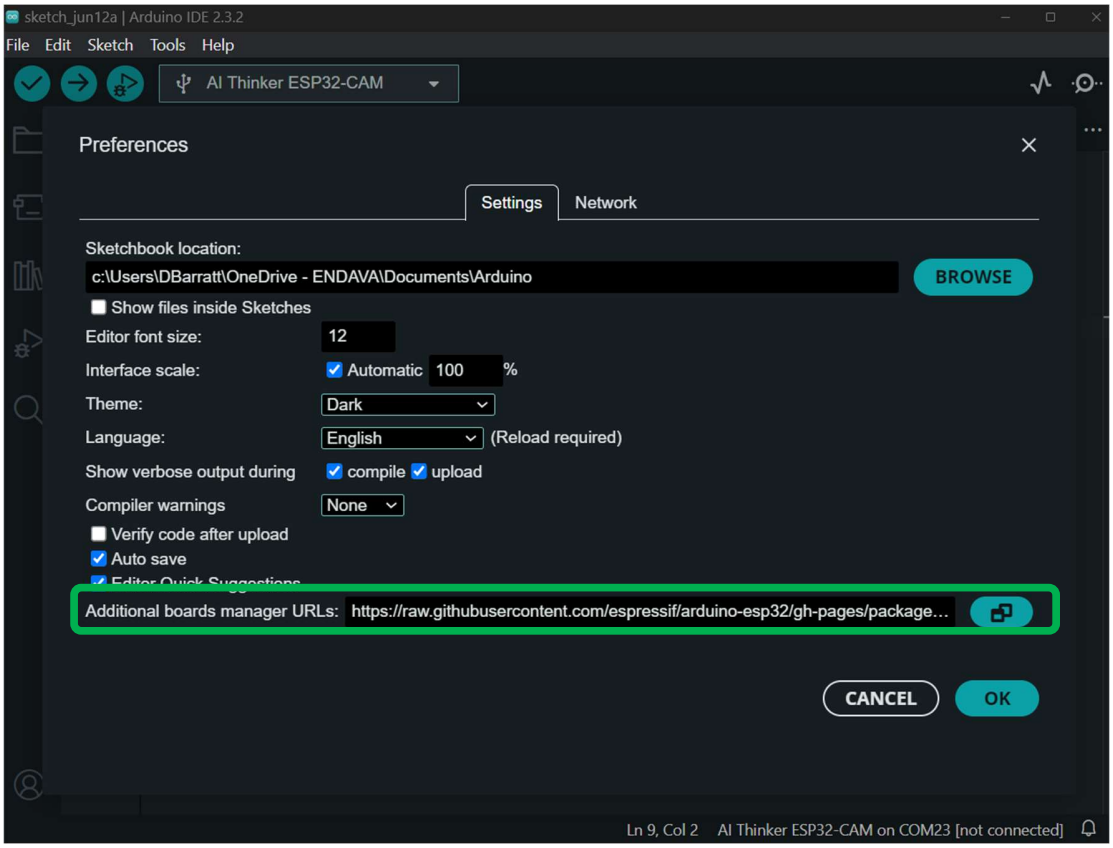
2) Adding ESP32 Board Support

a) In the Arduino IDE, navigate to **File > Preferences**.

b) Enter the following URL in the "Additional Board Manager URLs" field:

https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json



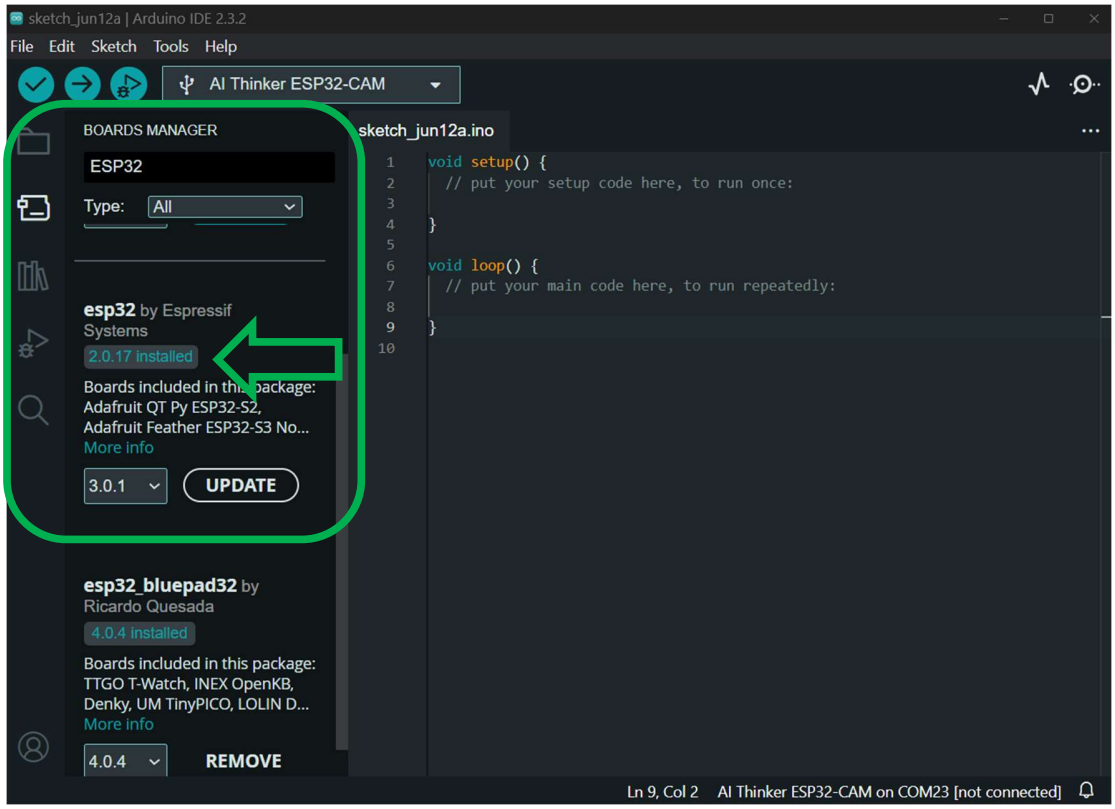


c) Click "OK" to save your changes.

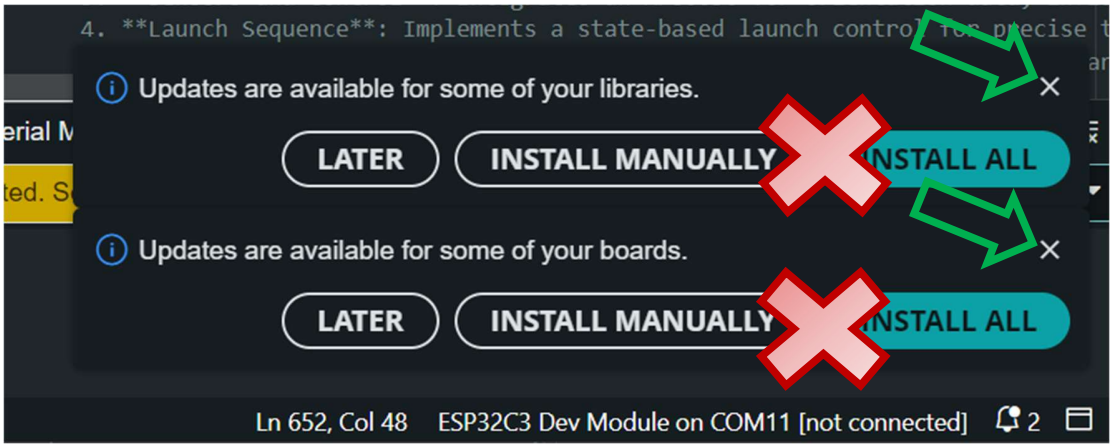
3) Installing the ESP32 Board Library

a) Open the Board Manager by going to **Tools > Board > Boards Manager**.

b) Search for **"ESP32"** and install **version 2.0.17** of the ESP32 board library. This version is mandatory to ensure functionality of the firmware.



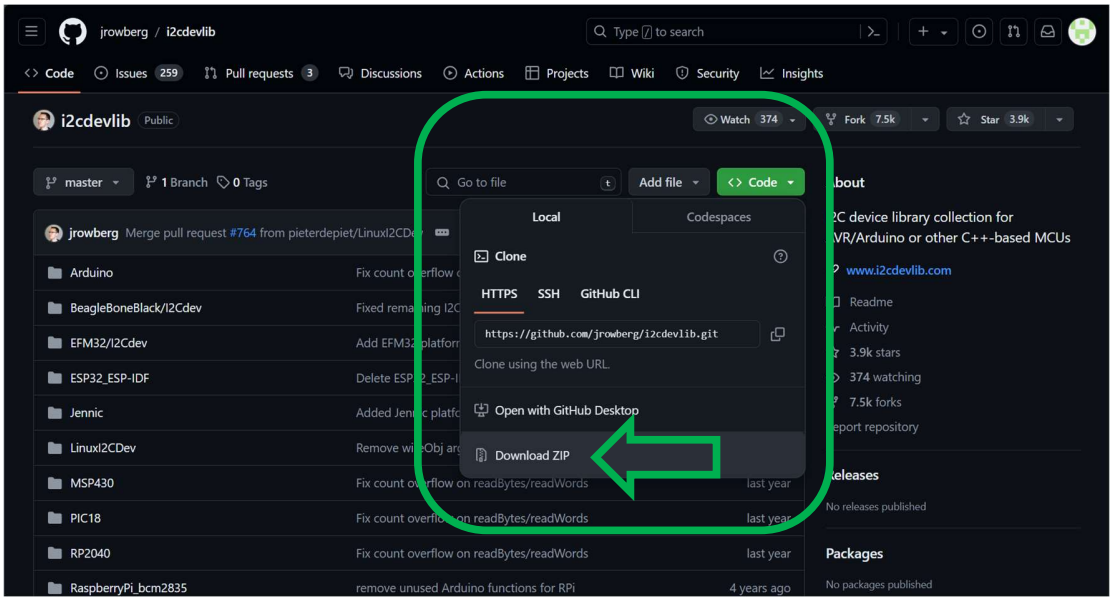
c) NOTE: When starting the Arduino IDE, it will ask you to update libraries. **DO NOT UPDATE**. Ignore these messages.



4) Installing Third-Party Libraries

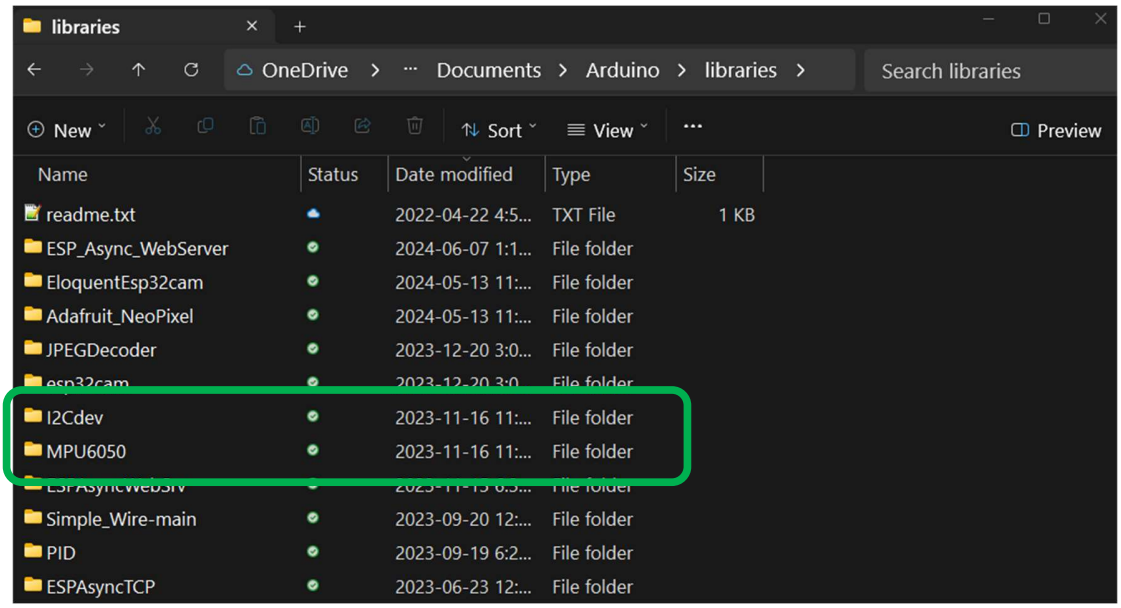
a) Download the **i2cdevlib library** from GitHub:

https://github.com/jrowberg/i2cdevlib/tree/master

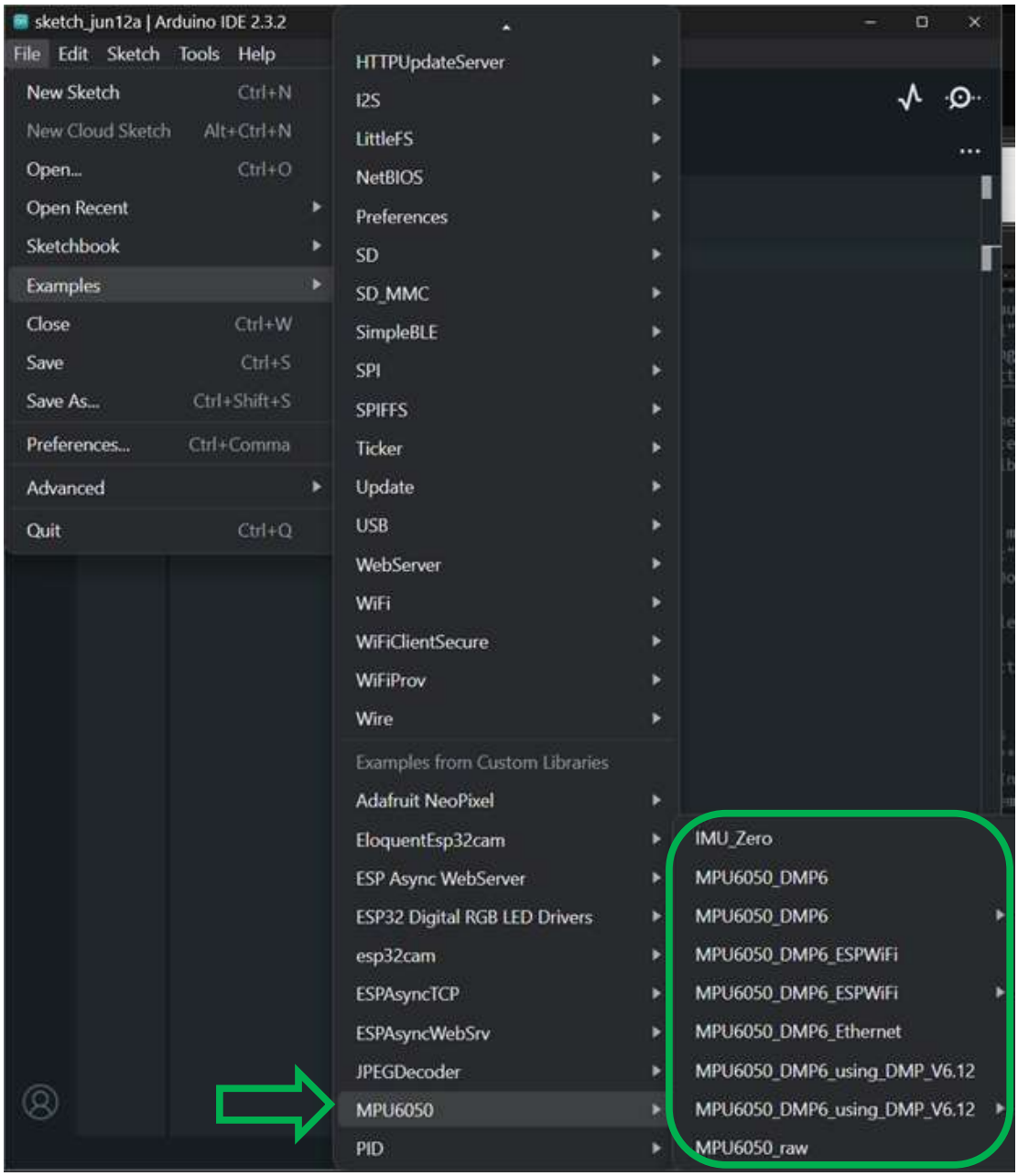


b) Extract the downloaded zip file and navigate to the **Arduino subfolder**. Copy the `***I2Cdev***` and `***MPU6050***` folders into **your Arduino libraries directory on your PC**.

Typically, this is located at ***C:\Users\<YOURNAME>\Documents\Arduino\libraries***



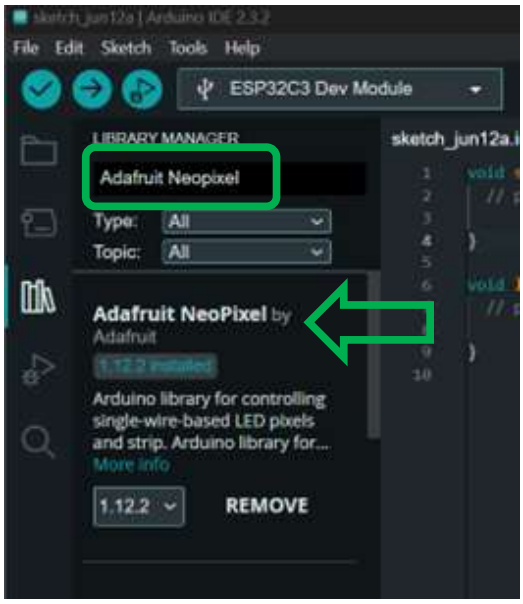
c) Check if libraries are present in Arduino. Restart Arduino. Navigate to **File > Examples** scroll to the bottom and confirm there is ***"MPU6050"*** listed.



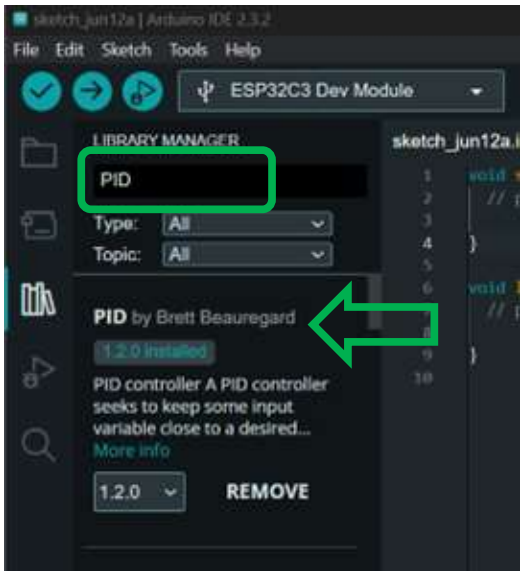
5) Installing Additional Libraries

a) Use the Library Manager in the Arduino IDE to install the following libraries:

"Adafruit_Neopixel" by Adafruit



"PID" by Brett Beauregard



Configuration and Setup for the Launcher Base Control Board

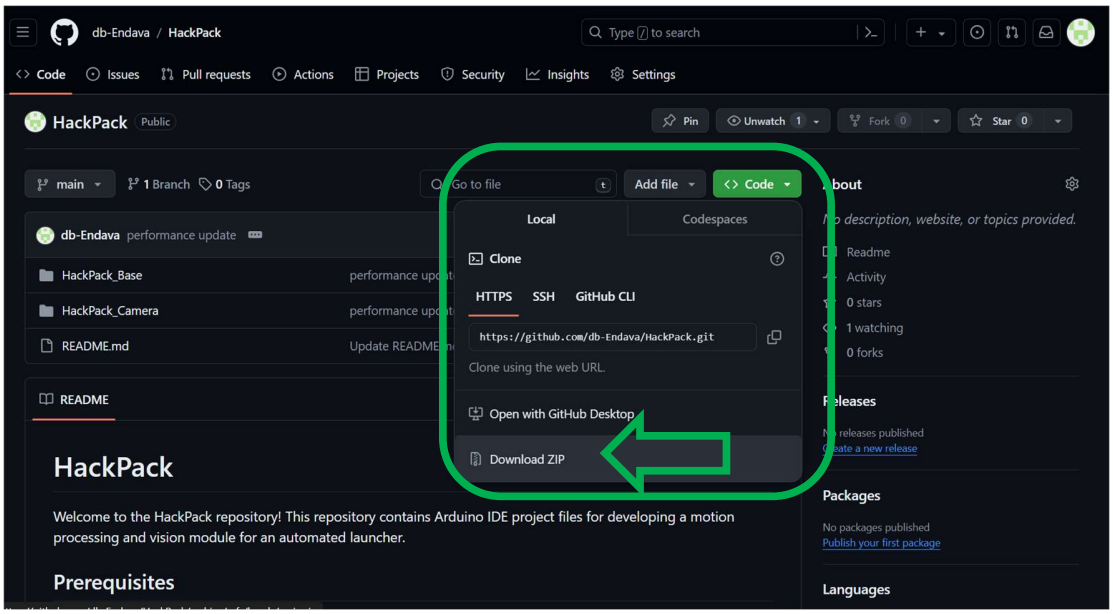
1) Project Preparation

1. Download the **HackPack project files** from HackPack GitHub Repository and extract them to a convenient location.

   https://github.com/db-Endava/HackPack

   Inside this folder you will see 2 project files containing the code for each unit.

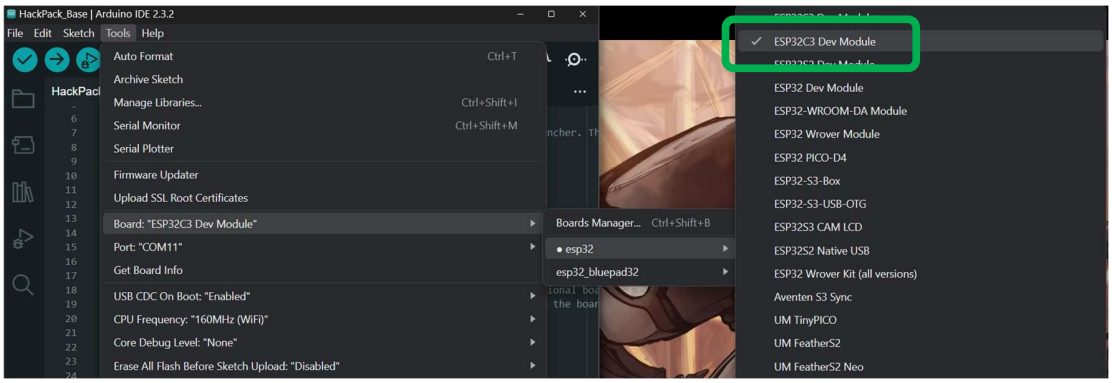
   - **HackPack_Base:** This is the main code of the project for controlling the functions of Launcher Base.
   - **HackPack_Camera:** This is the companion code to be loaded on the Camera Module co-porcessor.



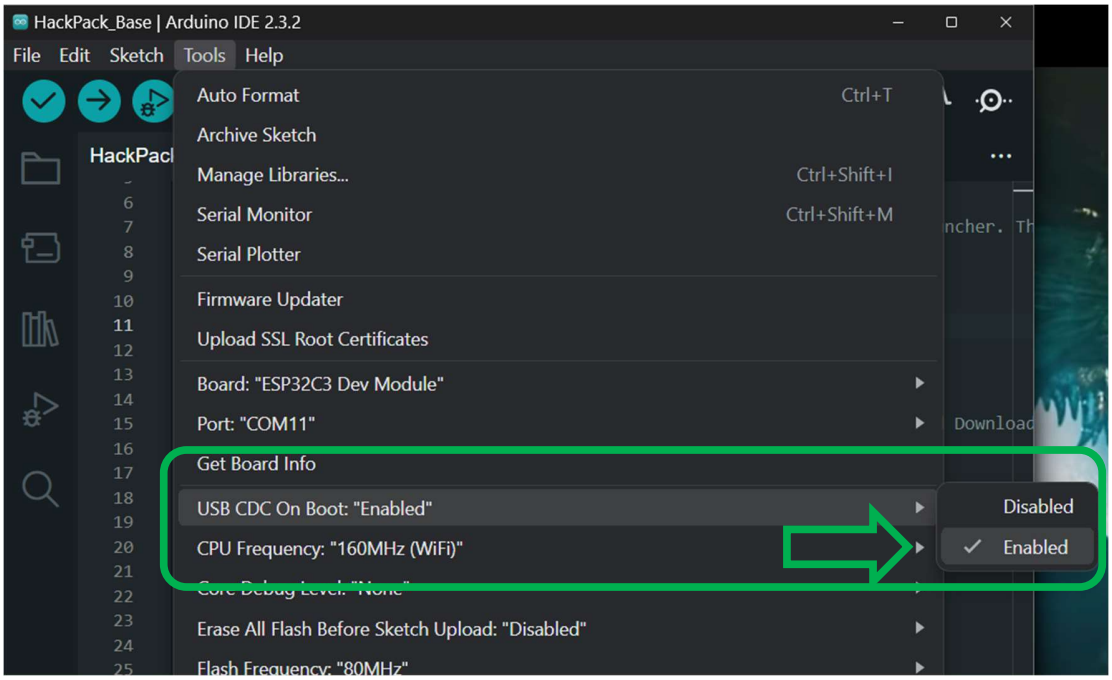
2) Board Configuration

a) Open the `***HackPack_Base.ino***` project file in Arduino IDE.

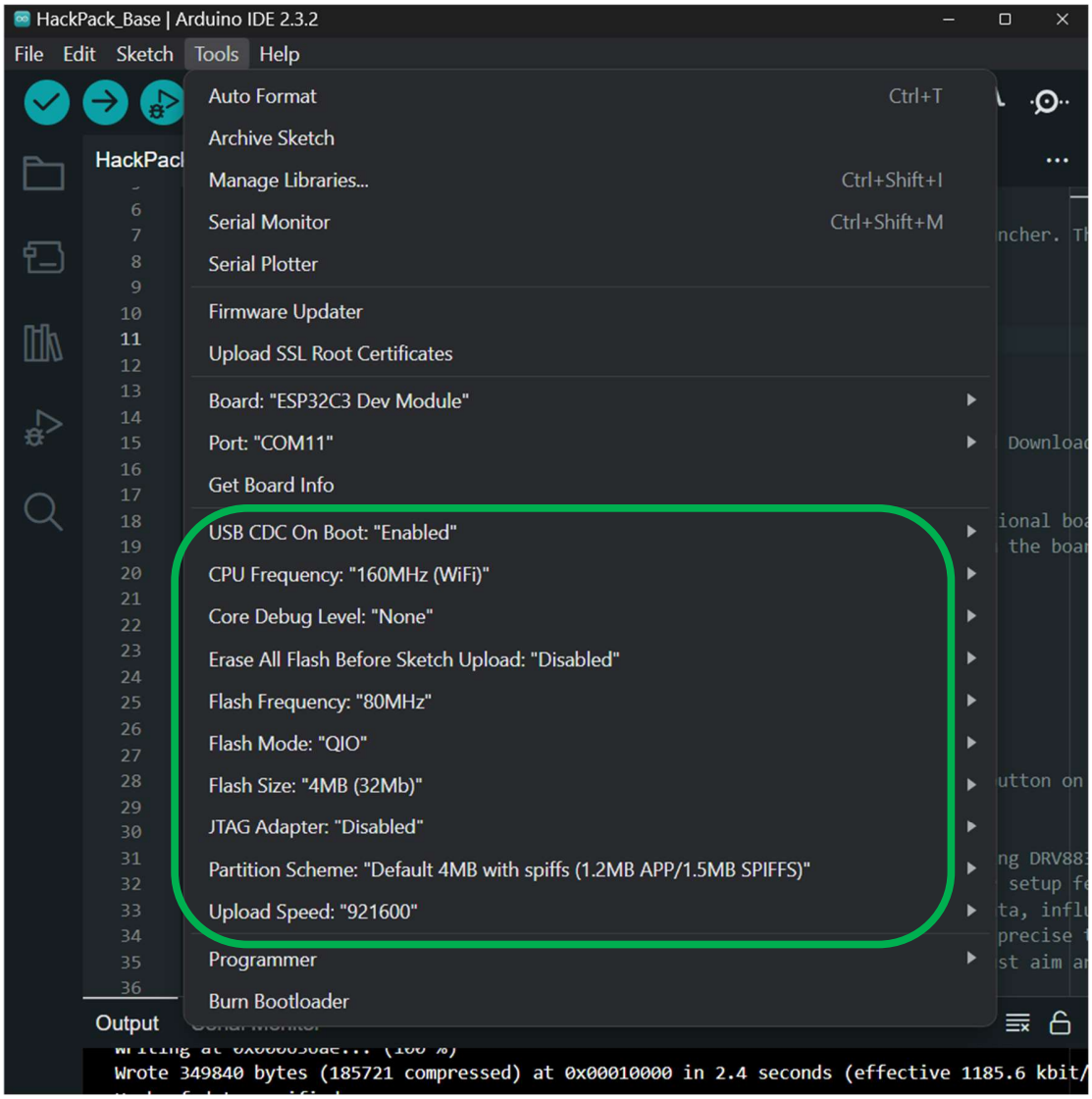
b) Set the board by navigating to **Tools > Board** and selecting ***"ESP32C3 Dev Module"***.



c) Under Tools, enable ***"CDC USB on Boot"***.



d) Check to ensure all other configuration options match the attach image. They should default to these options.

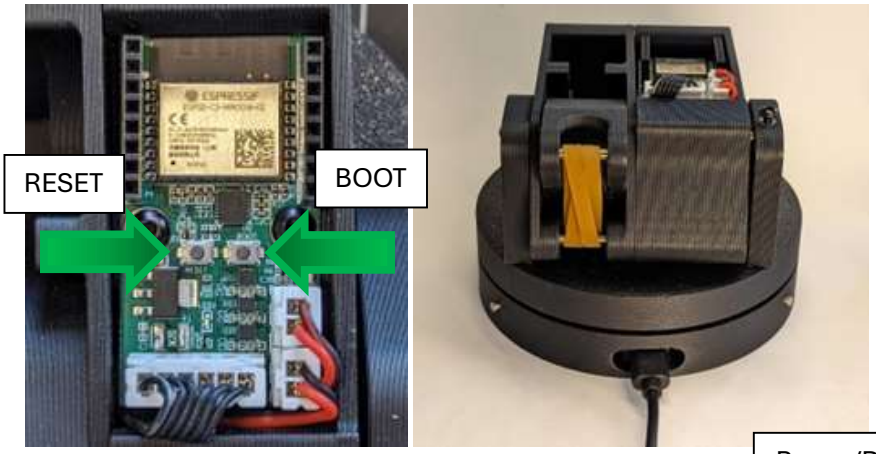


3) Board Connection and Initial Upload

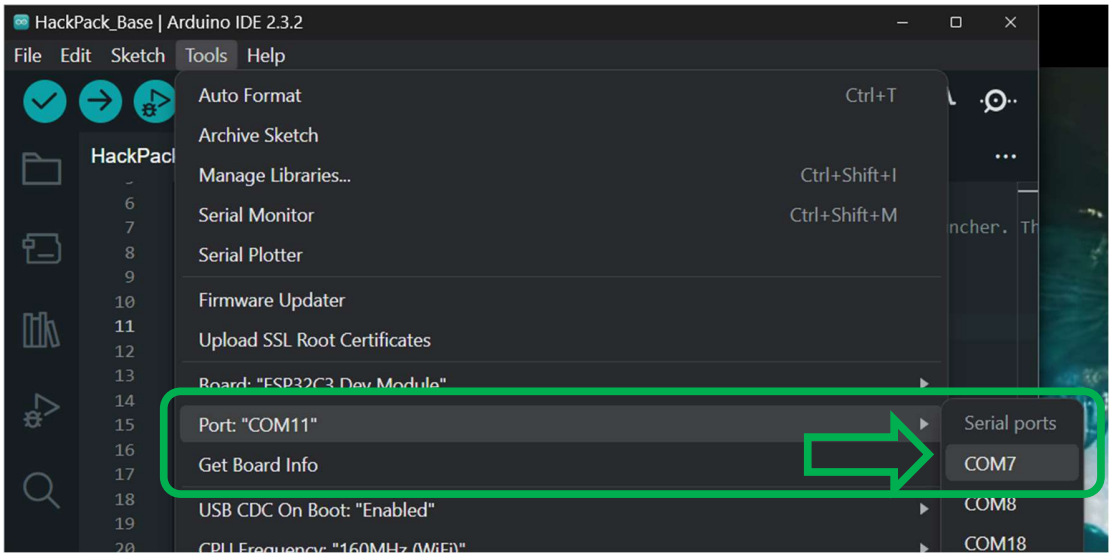
a) When connecting a power supply to the Launcher Base. The system will initiate a startup sequence. During this sequence, the launcher will move as it calibrates its sensors and stabilizers. **Caution:** Do not touch the launcher during the startup sequence. Interference could disrupt the calibration process and affect the system's performance. This process will may not happen on the first firmware upload but will happen on every subsequent power up.

b) First connect the **USB-C cable** just to your computer.

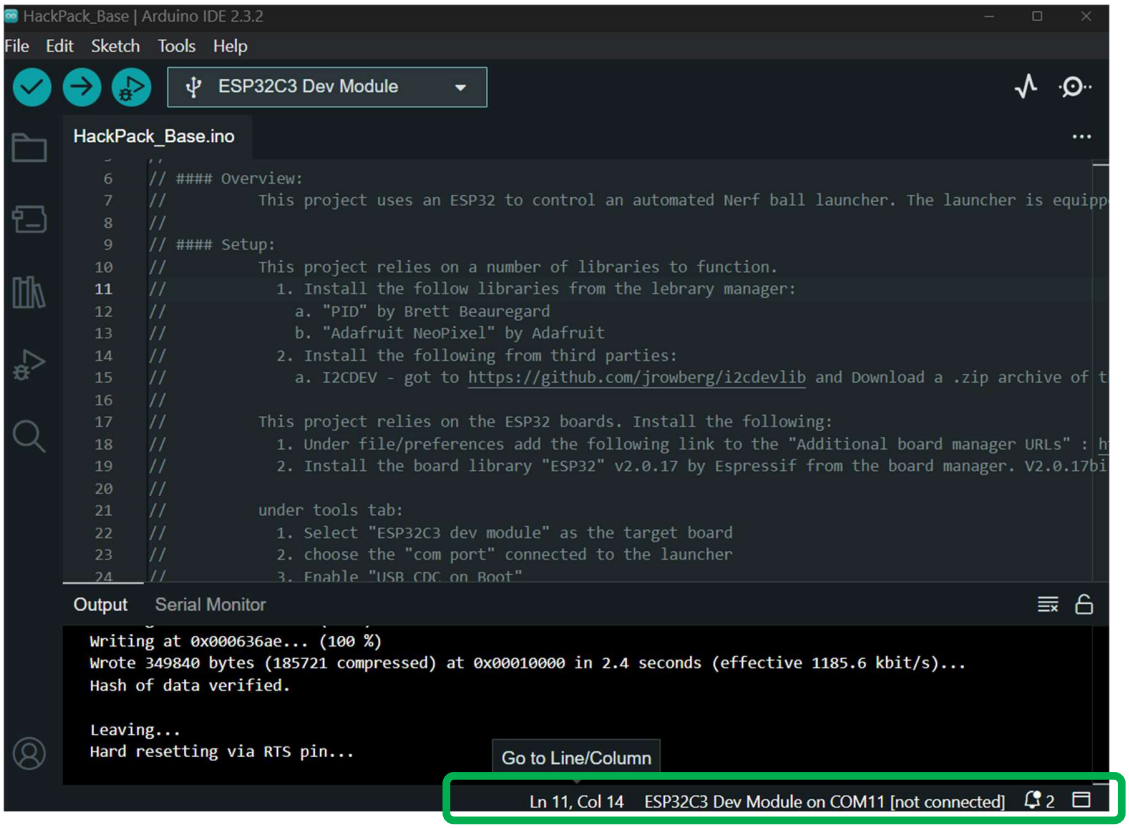
c) Before connecting the USB cable to the Launcher Base, press and hold the **"BOOT"** button on the launcher base, connect the USB cable to the launcher base, then release the button after 1-2 seconds. This will force the control board to enter the **BOOT mode**.



d) Select the correct COM port under **Tools > Port.** Your Port will most likely be different than the one shown.



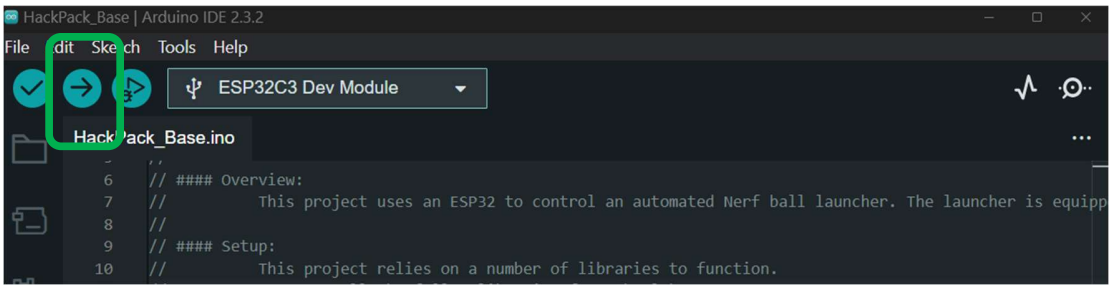
e) Ensure the Arduino IDE is **connected** to the control board. At the bottom right of the window is a simple indication to show if the system is connected or not.



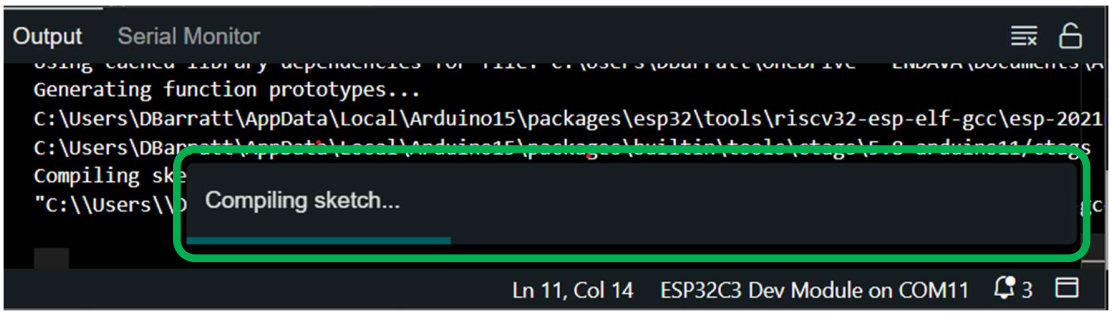
f) Click the "***Upload***" button to program the launcher base.

- If **upload issues** arise, use the "***RESET***" button to reset the board or **repeat the above steps** to force the unit into **BOOT mode.**

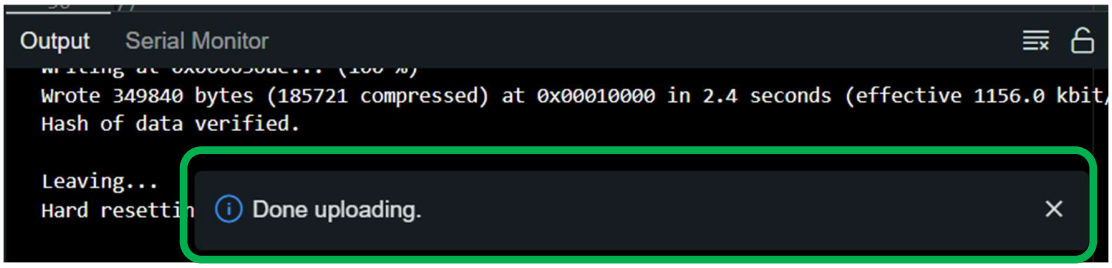
- If **compilation** errors occur, check if libraries are installed correctly and the correct target board is selected and configured correctly.



Compiling Firmware:



Upload Complete:



Startup Sequence of the Launcher Base (Without Camera Module)

To ensure safety and proper function, adhere to these guidelines during the startup sequence of the Launcher Base:

1) **Caution:** Do not touch the Launcher Base once power is applied. The system is fully automated and may move unexpectedly during calibration and stabilization.

2) **Power On:** When power is applied, the system begins by initializing its internal components and performing a diagnostic check to ensure all systems are functional.

3) **Calibration:** The base automatically calibrates its sensors and stabilizers. It performs slight adjustments to optimize performance. Ensure the unit is on a stable, flat surface to avoid calibration errors.

4) **Positioning and Stabilization**: After calibration, the Launcher Base automatically moves to a preset position and stabilizes. This is the default operational stance for executing commands.

5) **Status LED Display**: The status LED displays a rainbow color show, indicating that the system is in normal operation mode and functioning correctly.

Following this sequence ensures that the Launcher Base is properly calibrated, stabilized, and ready for use, emphasizing safety with the system's automated nature.

Interacting with the Launcher Base Using Serial Commands

After successfully loading the firmware onto the Launcher Base, you can control the unit via simple text commands through the Serial Monitor in the Arduino IDE. Here's how to set up and use the Serial Monitor for command input:

1) Setting Up the Serial Monitor

a) Click on the **Serial Monitor** button in the upper right corner of the IDE or navigate to **Tools > Serial Monitor**.

b) Set the baud rate to **"115200"** and **"Both NL & CR"** are selected in the right corner of the Serial Monitor window tab.

2) Command Instructions for testing.

a) Once the Serial Monitor is set up, you can enter the following text commands to control the Launcher Base:

- **"a"**: Move left.
- **"d"**: Move right.
- **"w"**: Tilt up.
- **"s"**: Tilt down.
- **"home"**: Center all axes. This command brings the launcher to its default position, aligning all moving parts to their central configuration.
- **"l"**: Launch. This command activates the launcher mechanism to execute a launch.

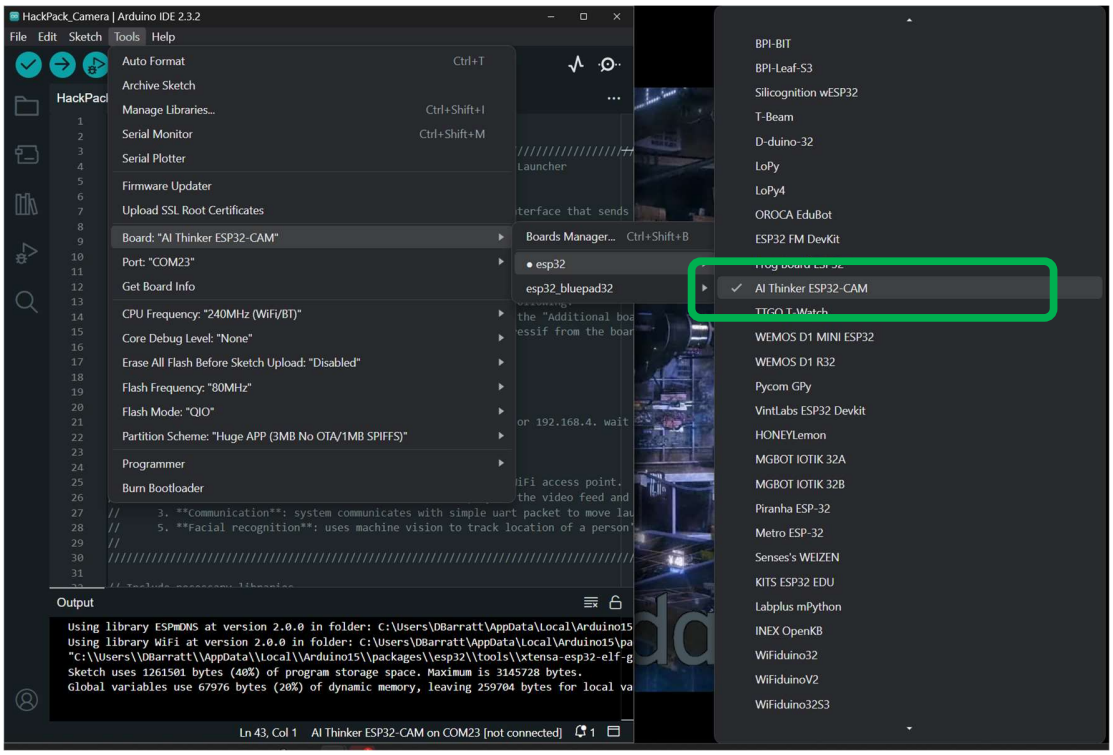
Type these commands one by one into the Serial Monitor and press **Enter** to send each command to the Launcher Base. The launcher will respond by moving according to the specified command, allowing you to control its orientation and launching actions directly from your computer.

Configuration and Setup for the Camera Module Co-processor

1) Board Configuration

a) Open the `***HackPack_Camera.ino***` project file in Arduino IDE.

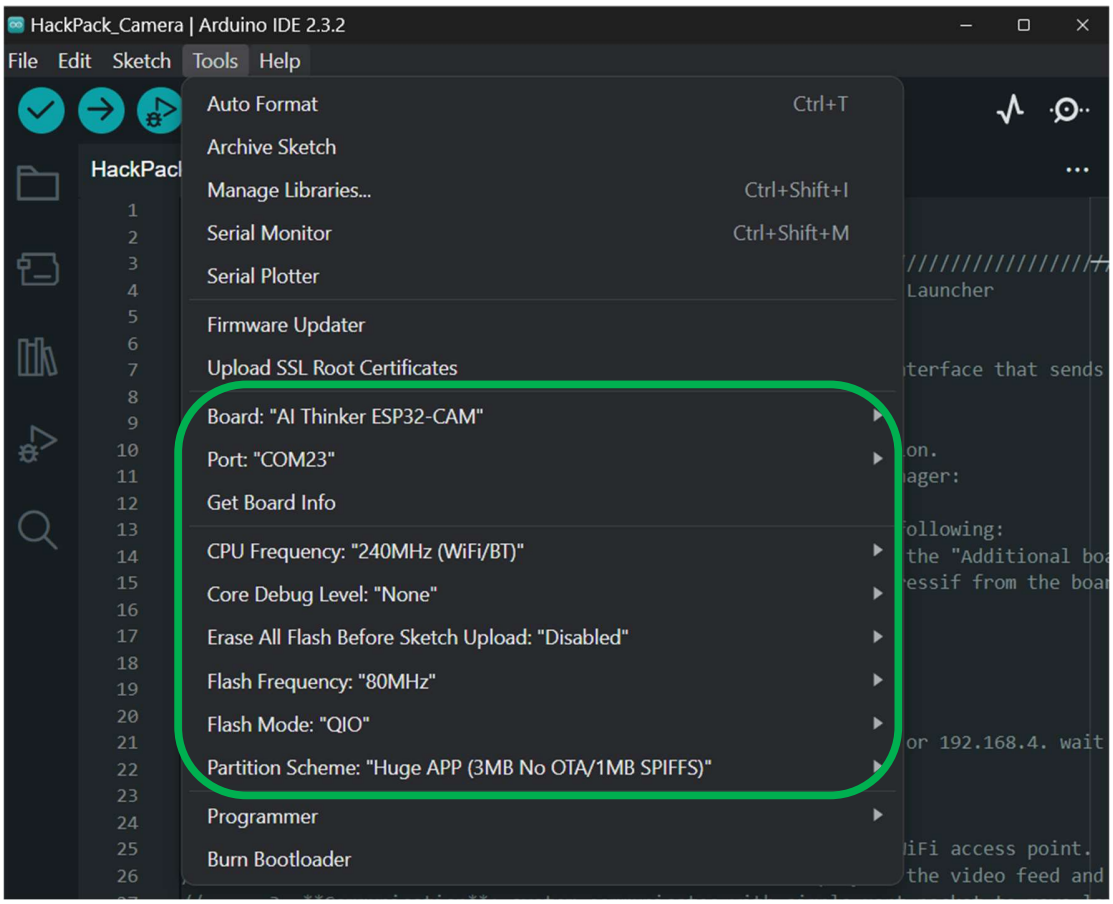
b) Set the board by navigating to ***Tools > Board > esp32*** and selecting ***"AI Thinker ESP32-CAM"***.



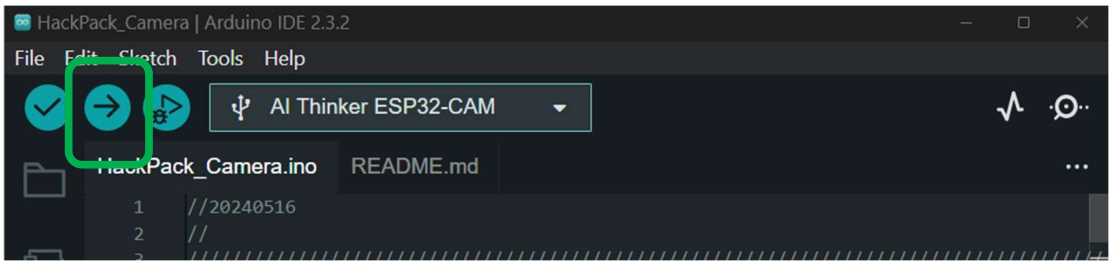
2) Connecting and Uploading Firmware

a) Connect the camera module to the programming interface board using a micro-USB cable

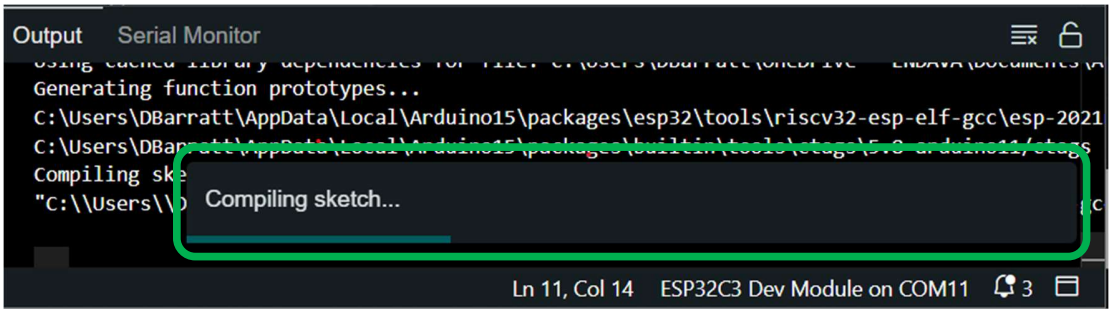
b) Ensure the correct settings are selected under ***Tools***, then choose the appropriate COM port.



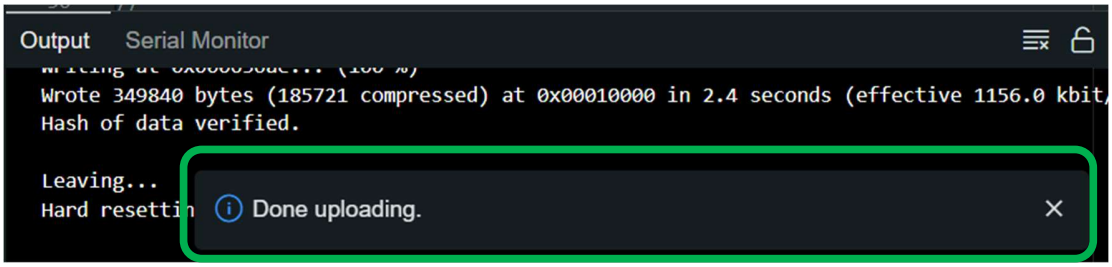
c) Click the ***"Upload"*** button to program the camera module.



Compiling Firmware:



Upload Complete:



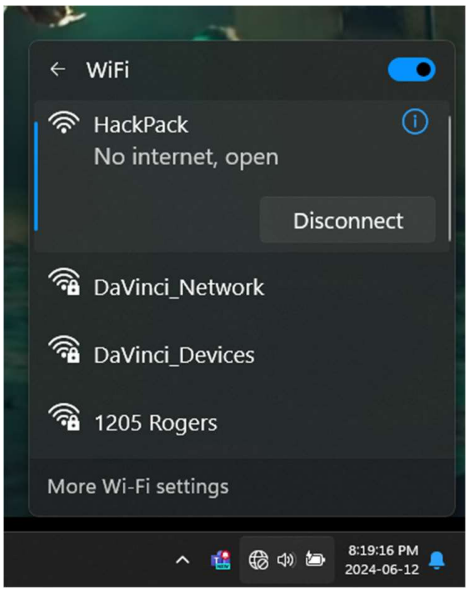
Testing the Camera Module Co-processor

Before integrating the Camera Module Co-processor into the Launcher Base, it's essential to thoroughly test its functionality while it is still connected to the programming module. This preliminary testing allows you to confirm the operation of the camera and the web interface without the complexity of the fully assembled system. Follow these steps to connect to the camera module and test its functionality:

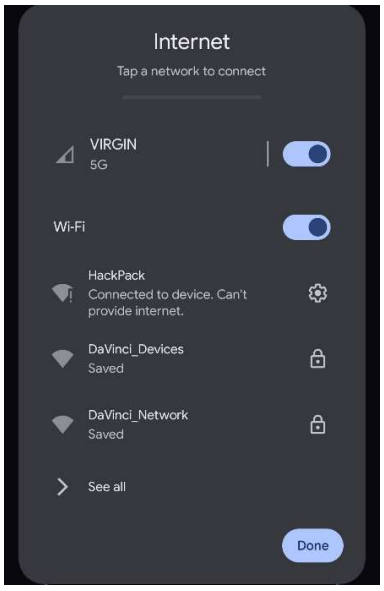
1) Connecting to the HackPack WiFi Network

a) On your device (mobile phone, laptop, or PC), navigate to your WiFi settings and connect to the network named **"HackPack"**.

PC / LAPTOP



Mobile



b) Since this WiFi network does not provide internet access, some devices may prompt you to confirm whether you want to stay connected to a network without internet. Select **"Yes"** to maintain the connection.

c) Note: On mobile devices, you may need to disable mobile data to prevent your device from switching networks searching for internet access.

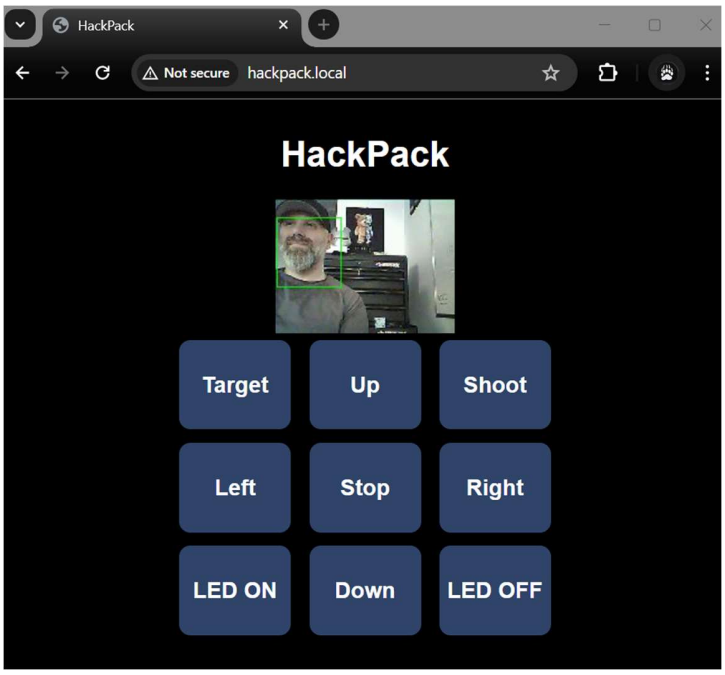
2) Accessing the Web Interface

a) Open a web browser on your connected device. Google Chrome is recommended for its compatibility and performance.

b) Enter the URL `***http://hackpack.local***` or the IP address `***192.168.4.1***` into the browser's address bar.

c) Press **Enter** to navigate to the web interface hosted by the Camera Module.

PC /LAPTOP



MOBILE

3) Using the Web Interface for Initial Testing

a) The web interface provides a first-person view from the camera and features an array of control buttons. These controls allow you to:

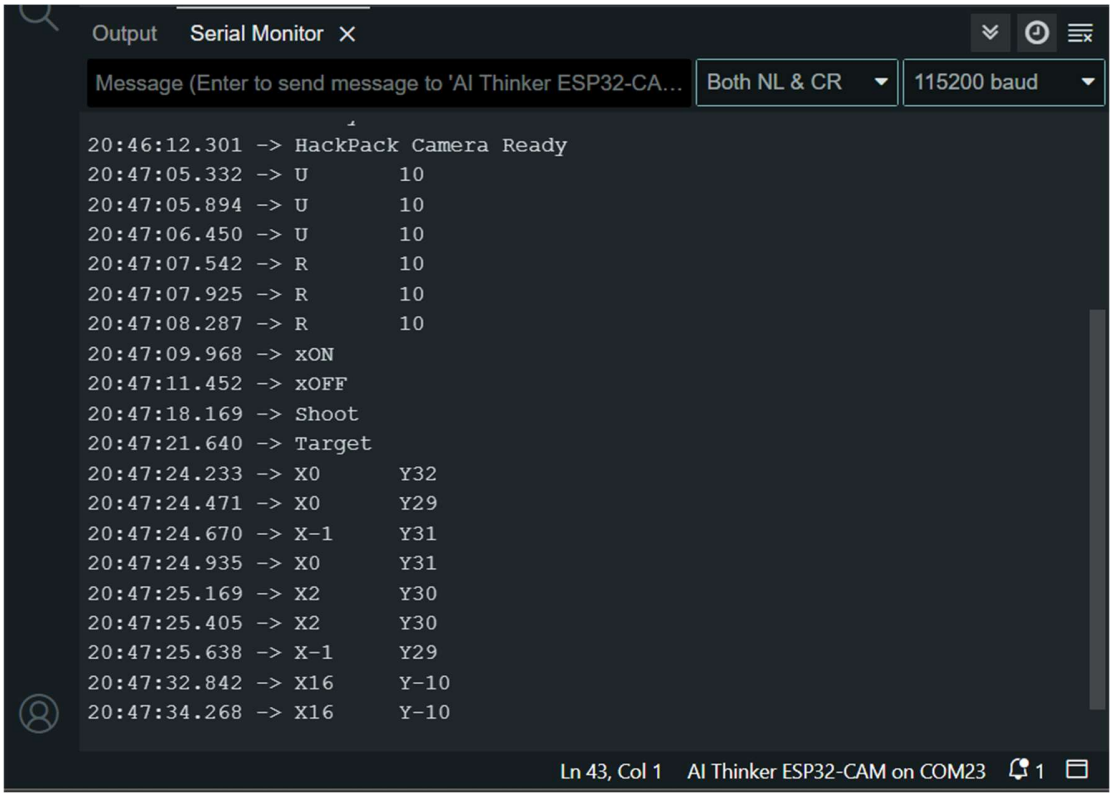
- **Move** the launcher up, down, left, or right.
- **Stop** all movement.
- **Enable Targeting Mode** for precise control.
- **Shoot** to activate the launch mechanism.
- Toggle the **Onboard Light** on or off for enhanced visibility.

4) Viewing Serial Commands in the Serial Monitor

a) While testing the Camera Module with the programming module still attached, open the Arduino IDE and navigate to the **Serial Monitor**.

b) Ensure that the baud rate is set correctly to **115200**.

c) As you interact with the web interface, observe the Serial Monitor in the Arduino IDE. This will display the serial commands being sent from the web interface to the camera module. It's a valuable step for troubleshooting and understanding the command flow from the web interface to the hardware.



By following these steps and confirming the functionality through the Serial Monitor, you can ensure that the Camera Module Co-processor is operating correctly before it is integrated into the Launcher Base. This process helps identify and resolve any issues early, simplifying the subsequent installation and integration steps.

Final Assembly and Configuration of the Launcher Base and Camera Module Co-processor

After testing the components individually, the final assembly involves integrating the Camera Module Co-processor with the Launcher Base. This configuration enables full control over the launcher's positioning, movement, launching capabilities, and automated targeting using machine vision. Follow these steps to assemble and prepare the system for use:

1) Preparation for Assembly

a) Ensure all devices are unplugged from power to safely handle the components during assembly.
b) Remove the Camera Module from the programming module. Care should be taken to handle the module by the edges to avoid damage to sensitive components.

2) Installing the Camera Module

a) Install the Camera Module into the headers on the Launcher Base control board. Reference the provided photos for correct placement and orientation. This ensures proper communication between the camera and the base's control systems.

3) Setting Up the Launcher Assembly

a) Place the assembled Launcher Base on a flat, stable surface to ensure accurate operation. This is crucial as the system uses gyro stabilization to maintain orientation.

4) Powering Up the System

a) Connect the power supply to the Launcher Base. The system will initiate a startup sequence. During this sequence, the launcher may move as it calibrates its sensors and stabilizers.

b) **Caution:** Do not touch the launcher during the startup sequence. Interference could disrupt the calibration process and affect the system's performance.

5) Connecting to the Web Interface

a) Once the startup sequence is complete, you can wirelessly connect any device with a web browser to the "HackPack" WiFi network hosted by the Camera Module.

b) Navigate to the web interface using `**http://hackpack.local**` or `**192.168.4.1**` to control the unit.

6) Using Targeting Mode

a) With the web interface, direct the launcher to position a person within the field of view of the camera.

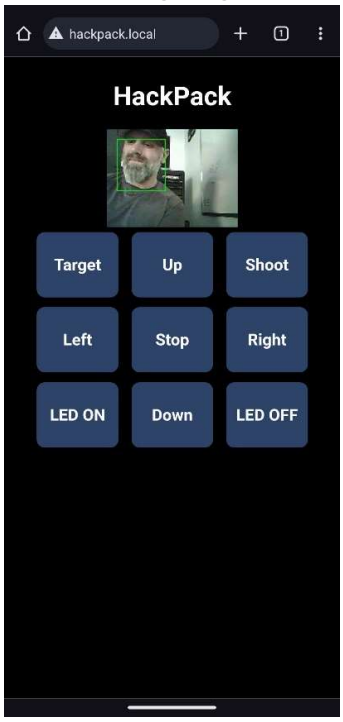
b) Enable **Targeting Mode** using the designated button on the web interface. In this mode, the launcher uses machine vision to detect faces and automatically adjust its position to center the detected face in the viewfinder.

c) The colored LEDs on the front of the control board will change from a rainbow color pattern (indicating normal mode) to a blue and orange flashing cycle, signaling that the system is in auto-targeting mode.

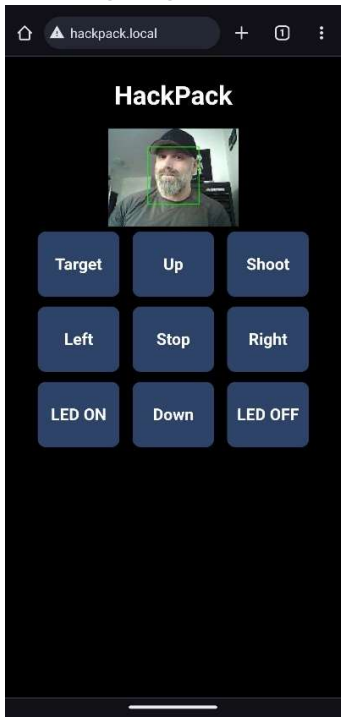
This final configuration demonstrates a practical application of machine vision in an embedded system, albeit with the limitations of processing a few frames per second due to resource constraints. The system is now fully assembled and ready for operation, offering an interactive experience with technology that bridges mechanical design and computer vision.

MOBILE Interface

Targeting OFF

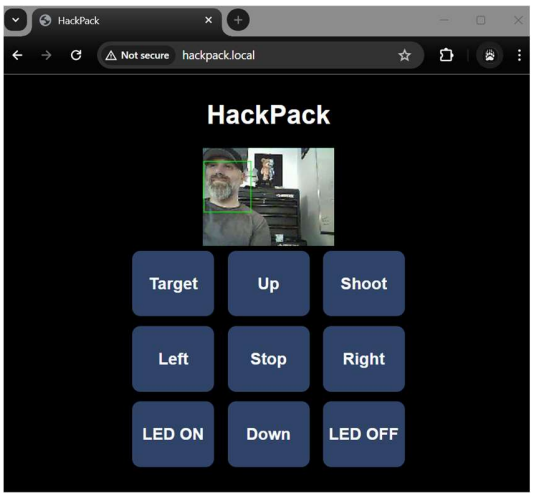


Targeting ON



PC / Laptop Interface

Targeting OFF



Targeting ON