

# Robohacks City Sweep

## Code Instructions

For this challenge, you are given an ESP32 WROVER CAM to use for building your robot. This document outlines the steps you'll need to take to be able to program your ESP32.

### Requirements:

- Laptop
- USB-C Cable (provided)
- Arduino or Arduino IDE (instructions below)
- ESP32 Boards Add-on in Arduino or Arduino IDE (instructions below)
- Robohacks Libraries Zip File (optional)

If you have met all the requirements above, you can skip to page 2 for the instructions on using the Robohacks Libraries (if you plan on using them).

### Arduino Installation

Arduino or the Arduino IDE can both be installed at the following link:

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

(If you do not have much space on your laptop, the Legacy Arduino IDE 1.8.19 is a more lightweight version, available if you scroll down on the page)

Follow the steps to install Arduino.

### ESP32 Installation

The ESP32 boards must be added to the Arduino IDE to be able to code with them. To do this, you can follow the tutorial at the following link:

<https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/>

You only need to follow the steps in the “**Installing ESP32 Add-on in Arduino IDE**” section. (The website also contains troubleshooting help if you have any issues)

# Robohacks Library Installation and Usage

## **Reminder that this library is optional, and you can write your own communication files**

The library zip can be downloaded from the discord channel “Software”. Once unzipped, you will see two folders and a zip file. The RHComms.zip file contains the two header files you will need to import and include in your Arduino sketch.

The “comms\_example” folder contains an Arduino script that showcases how to use the Robohacks Communication library. As a simple script, all it does is set up the WiFi network from the ESP32, and then turns on and off an LED if a button is pressed on the Python GUI. Use this as a template to create your own script for this challenge.

When creating your own script (using the “File>New” in Arduino IDE), make sure to include the “RHComms.zip” package by clicking “Sketch>Include Library>Add .ZIP Library” and selecting the package. Then, add the library to your sketch by selecting “Sketch>Include Library>RHComms”.

After creating your own script, make sure to also select “ESP32 Wrover Module” in “Tools>Boards>ESP32 Arduino”. If you do not see the option, make sure you followed the steps above to install the ESP32 boards in Arduino, and try restarting the Arduino IDE for changes to take effect. Once your code compiles properly, you can plug in the ESP32 with the USB-C cable, and select the right port in “Tools>Port”. If you are unsure which one it is, try unplugging and plugging back in the ESP32 to see which option appears and disappears.

## Python GUI

For the Python End, you will need the following:

- Python (3.7+)
- Robohacks Package
- Python packages - *pygame* and *pygame\_gui*

If you do not have Python installed, you can install it here: <https://www.python.org/downloads/>  
Once Python is installed, you can install the required packages by open a Command Line Window (WIN+R + “cmd”) by running the following command:

***python -m pip install pygame pygame\_gui***

To run the Python GUI, run “main.py” in the “Python Files” folder in the IDE of your choice. You can also run it using Command Line, by navigating to the “Python Files” folder, and then running: “python main.py”. If you want to customize the GUI, follow the instructions in the comments of the “main.py” file.

**If you have any other questions, don't hesitate to ask a Robohacks Organizer!**

# API Reference

It is not recommended to edit the Python files, other than explicitly said in the “main.py” file. However, if you would like, “communication.py” and “gui.py” have all the code to run the GUI and connect to the ESP32.

## Arduino Reference

**int run()** - This function should be called in the main loop, as it parses all incoming data from the Python GUI into the variables accessible by Arduino. It returns -1 if there is an error, otherwise it returns the number of messages that it read (should be 1 normally).

**int hasClient()** - This function checks if the Python GUI is currently connected to the ESP32. If it is, the function returns 1, and a 0 if not.

**float getValue(char\* category, char\* parameter)** - This function accesses the received data from the GUI. It is accessed by writing a category and parameter, which returns the float value received. Example: “float analogStick1X = controller.getValue(“mouseStick1”, “x”);” which would return a number from -1 to 1 (value from the stick in the GUI)

**void setServer(WiFiServer\* server)** - This function passes the address of the WiFi server to the Robohacks library, so that it can parse incoming data. Only need to be called once in the setup() function.

**int connectClient()** - This function tries to connect to the Python GUI. It returns a 0 if it is successful and the GUI connects, and a 1 if it did not connect to the GUI.