

## Introduction

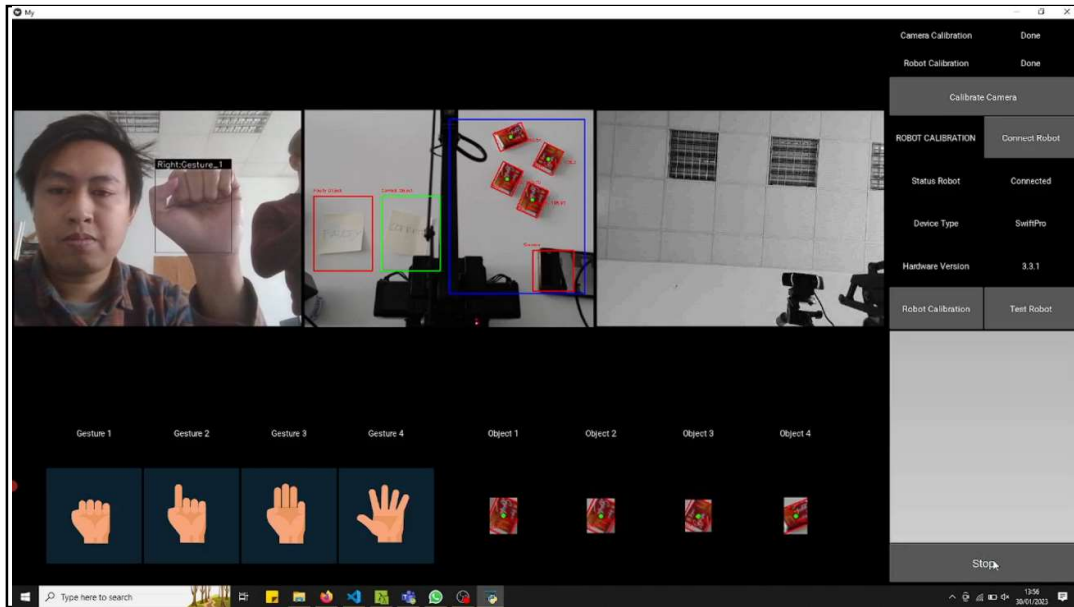


Figure 1. Application Interface

The application allows you to control a robot using hand gestures. As you can see in *Figure 1*, the application used two cameras as the input, the first camera pointed to the user and the second to the robot. The first camera will capture the user's hand gesture, and the second camera will capture the object. Both cameras will capture images in real-time and send the frame to the application. The application will process the frame, for the first camera, it will recognize the user's hand gesture which we will use as a command for the robot, and for the second camera, it will identify the object inside the blue area in the frame. The application will calculate the coordinate of every object in the blue area in the frame and convert it from an image coordinate to a real-world coordinate for the robot. The robot will grab an object based on a hand gesture from the user and move it to the specific coordinate. The application will recognize only four hand gestures you can see in *Figure 2*, and for every hand gesture will be assigned only one object, as you can see at the bottom of *Figure 1*. If there are more than four objects in the frame, the application will sort all the objects and show object number one until four. After one object moves, the rank will keep updating until there is no more object in the frame.

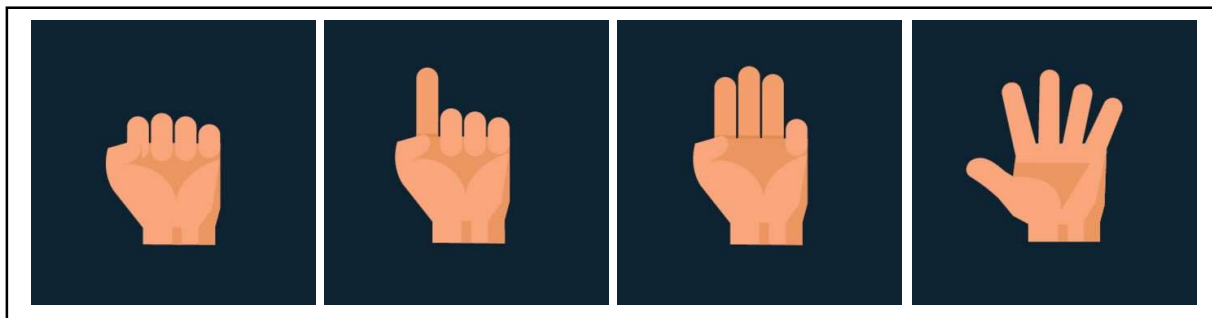


Figure 2. Hand Gesture

Before using the application, there are four steps you have to do.

1. Calibrate the camera that is pointing to the robot and object
2. Connect the robot
3. Calibrate the robot
4. Test the robot
5. Result

## 1. Camera Calibration

This step aims to determine the geometric parameters of the image formation process. This is crucial because we will get intrinsic parameters (focal length, principal point, skew of axis) and extrinsic parameters (rotation and translation). We will use all the parameters to calculate the robot's transformation matrix. There are several methods to calibrate the camera, but we have to use the ArUco module for this application. Using the ArUco module, calibration can be performed based on ArUco markers corners.

There are several steps to do the calibration using ArUco:

1. Before calibrating the camera, you must ensure you **disconnected** the robot from the laptop.
2. Download the ArUco board from this [link](#) and print it on the A4 paper
3. Start the application and click **Calibrate Camera** button in the top right corner

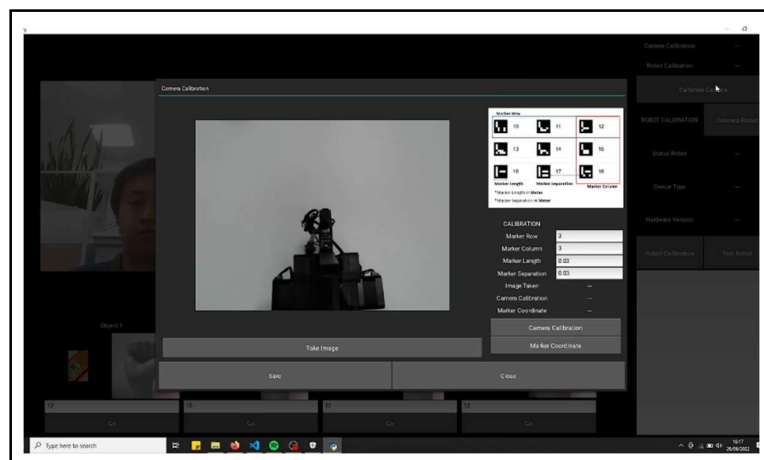
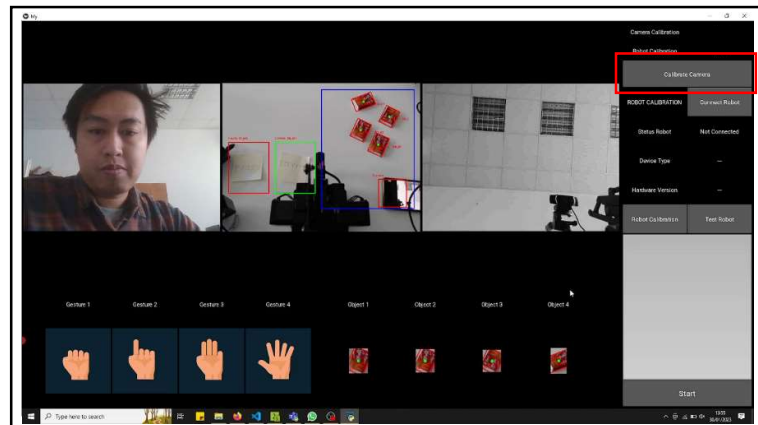


Figure 3. Camera Calibration Interface

4. After you click it, it will appear Camera Calibration interface like *Figure 3*
5. Before calibrating the camera, you must take images of the ArUco board in different positions and angles. See *Figure 4* for the exact location of the board. **You must ensure that all the marker is in the frame and not cut off or blocked.**

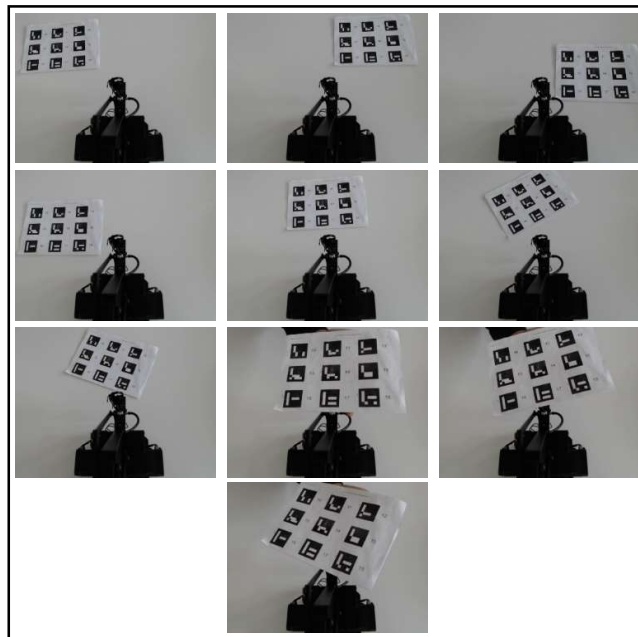


Figure 4. Marker Position

6. For every position, you have to take the image by clicking the **Take Image** button at the bottom of the image

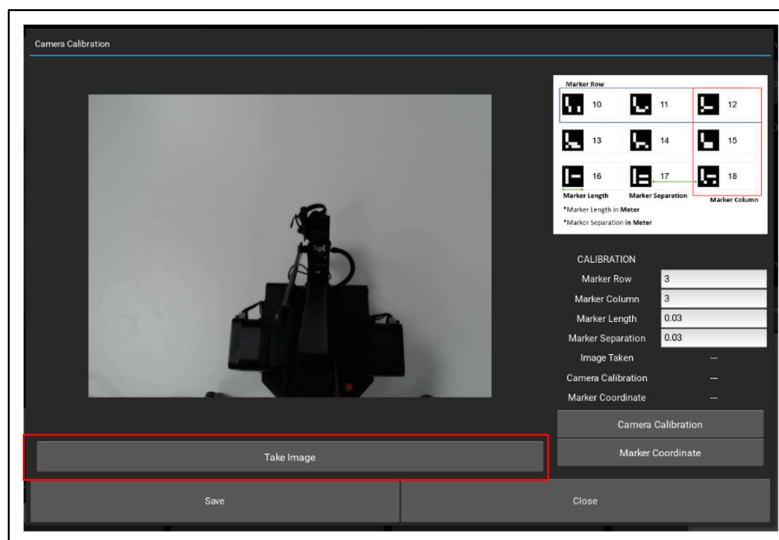


Figure 5. Camera Calibration Interface 2

7. After you take all the images, there are some parameters you have to set, like **Marker Row**, **Marker Column**, **Marker Length**, and **Marker Separation**. Explanation of every parameter:

- a. **Marker Row**: How many markers in a row
- b. **Marker Column**: How many markers are in the column
- c. **Marker Length**: The length of the marker in Meter
- d. **Marker Separation**: The lengths of separation between markers in Meter

See Figure 6 for the detail.

**Note:** If you are using the ArUco board that is given in this tutorial, you don't have to change the parameters

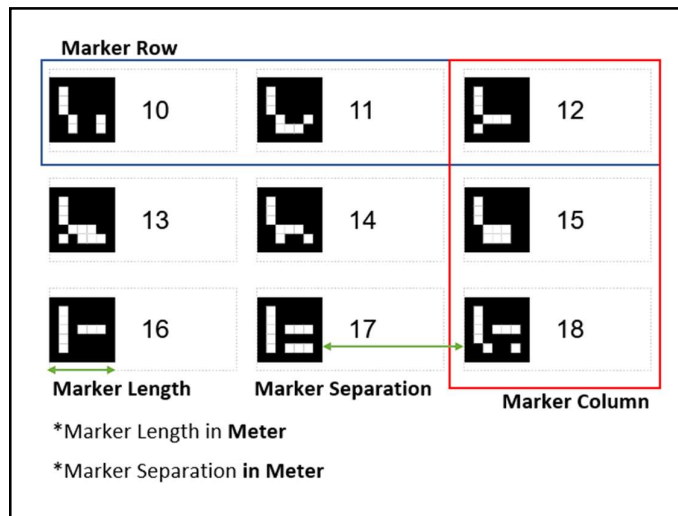


Figure 6. Parameter Explanation

- After the parameter set, you can click the **Camera Calibration** button in the bottom right corner. Make sure that the Camera Calibration status change from “-” to “Ok.”

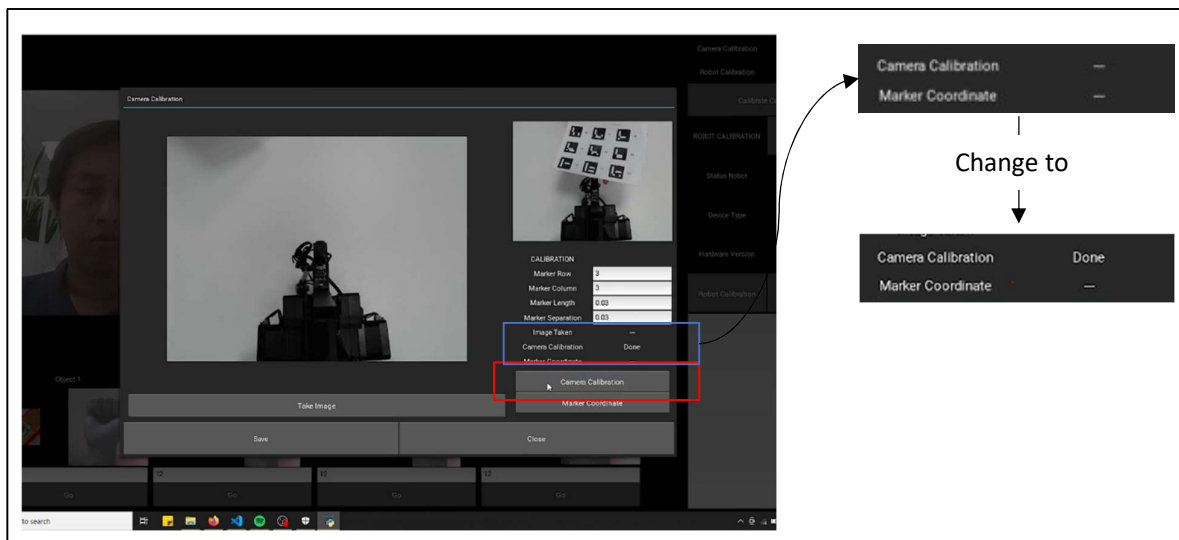


Figure 7. Camera Calibration Interface 3

- After calibrating the camera, we have to do one more step: detect every marker in the ArUco board. This step is also important to find the Marker ID, coordinate and translation vector for every marker. Before we take marker coordinates, we have to place the ArUco board in front of the robot, like in *Figure 8*. **You must ensure that all the marker is in the frame and not cut off or blocked.**

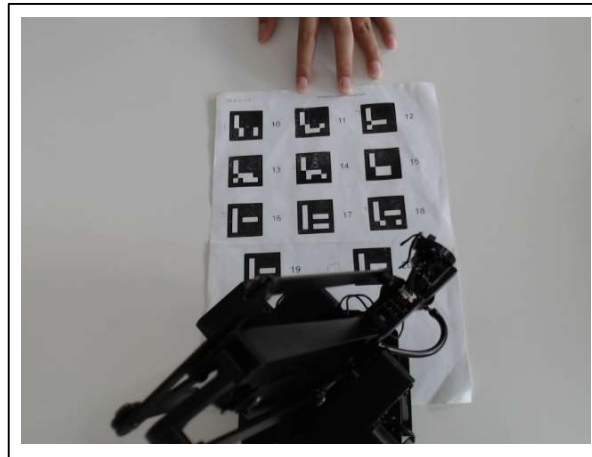


Figure 8. ArUco Board Position

10. After you place the Aruco board in the right position, you have to click the Marker Coordinate button in the bottom right corner. Make sure that the Marker Coordinate status change from “-” to “Ok.”

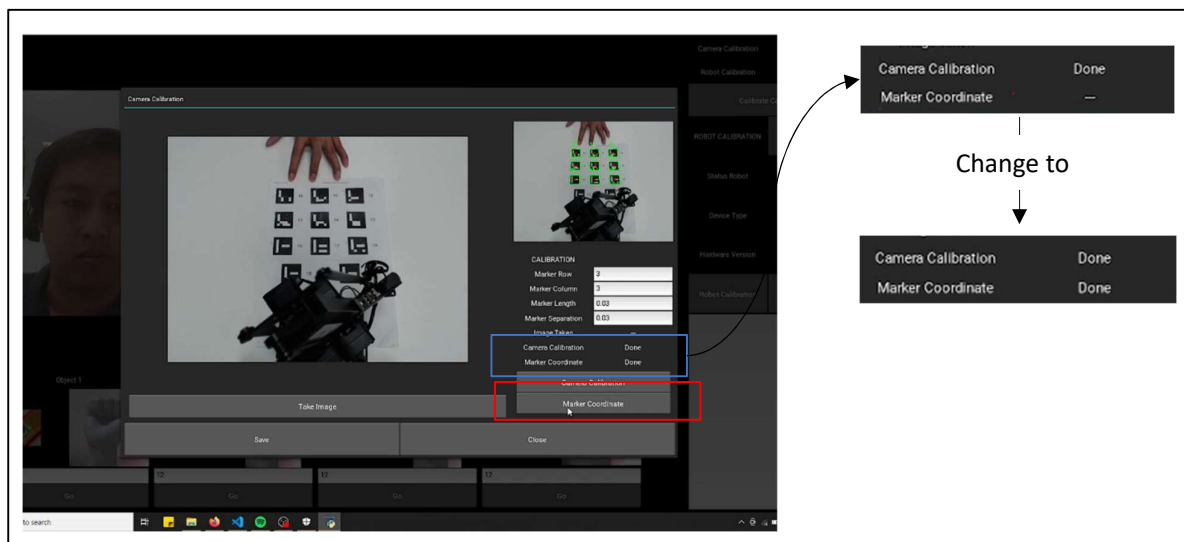


Figure 9. Camera Calibration Interface 3

11. You can see in *Figure 10* that the application can detect all the markers in the Aruco board, which means that we have already finished the camera calibration step.

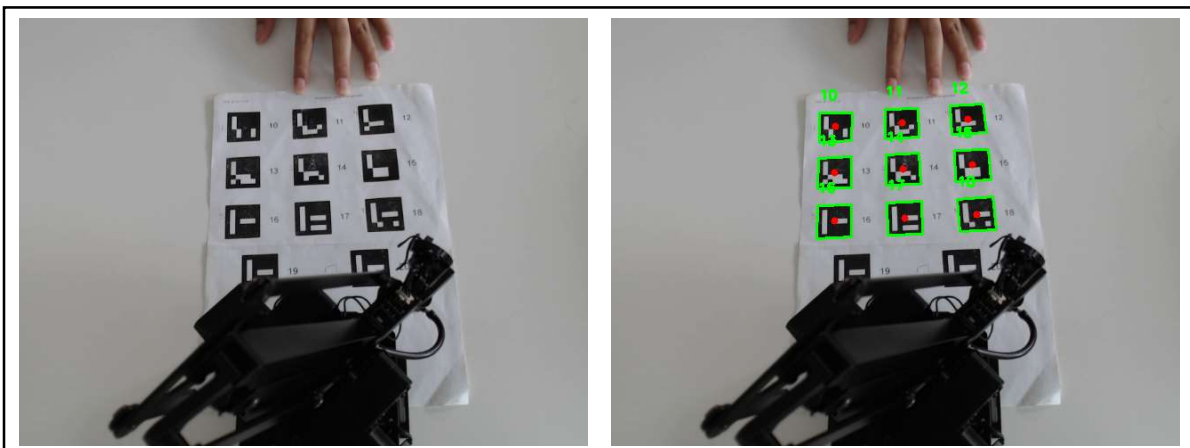


Figure 10. Marker Recognitions

12. Don't forget to click the **Save** button in the bottom left corner before you close the interface; this step is important to save all the parameters into a txt file, so we don't have to calibrate the camera again the next time as long as the camera position and robot position doesn't change.

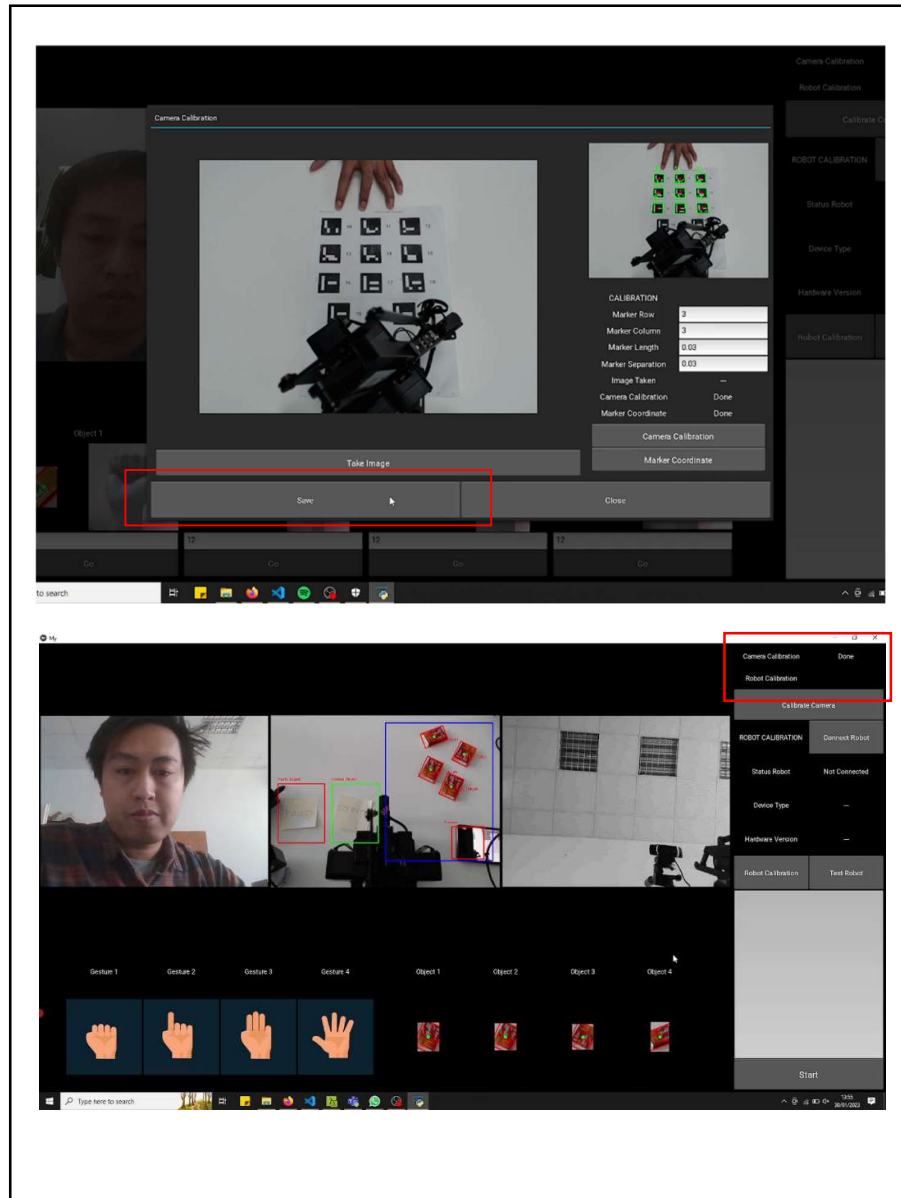


Figure 11. Camera Calibration Status

13. After you click the **Close** button, you can see in the upper right corner that the status of Camera Calibration has been changed to Done. It means we successfully calibrate the camera.

## 2. Connect the robot

After you calibrate the camera, one small step you have to do before starting to calibrate the robot, which is to connect the robot to the application. Follow this step to connect the robot:

1. Connect the robot to the computer or laptop and turn on the robot
2. Click **Connect Robot** button in the middle right of the application

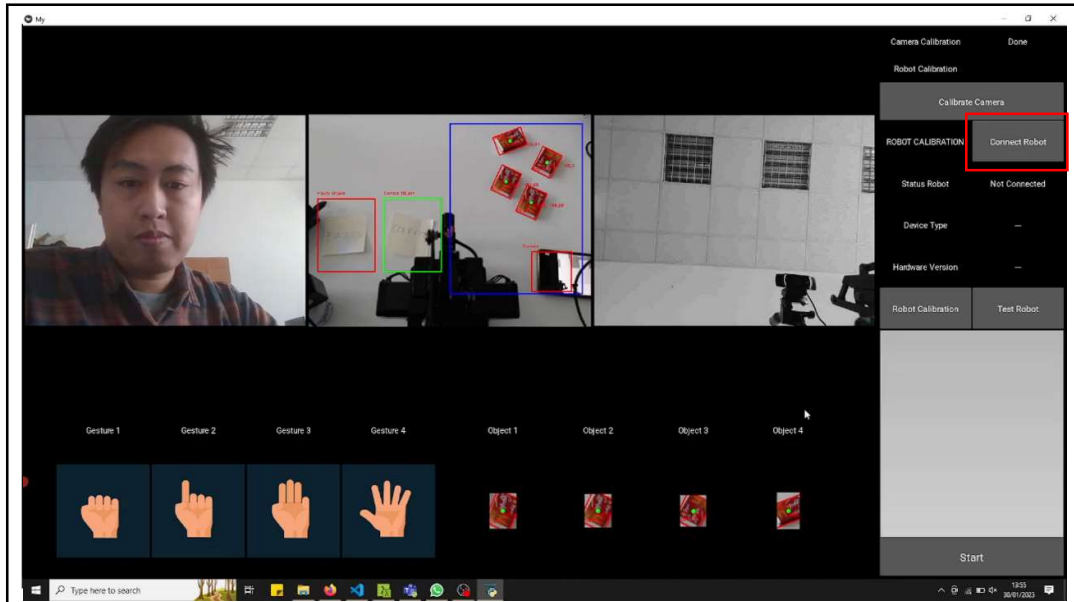


Figure 12. Connect Robot Button

3. If the robot successfully connected with the application, then the information of the robot will be shown in the interface.
4. If the information doesn't show after you click the button, the application can't recognize the robot. Try to turn off the robot and repeat the step.

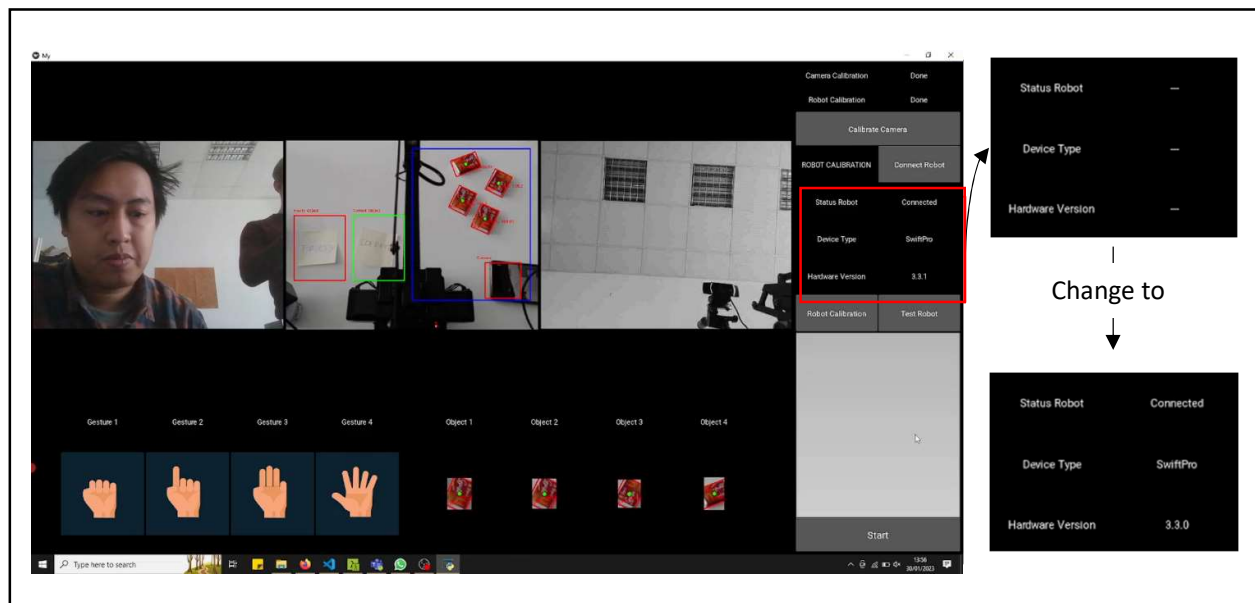


Figure 13. Robot Information

### 3. Robot Calibration

This step aims to get the robot coordinate of the marker we choose. We will select four of the markers in every corner of the ArUco board. After we get the robot coordinate, we will calculate the transformation matrix from the robot coordinate and marker coordinate in the frame. Follow this step to get the robot coordinate:

1. Click the **Robot Calibration** button in the middle right of the application.

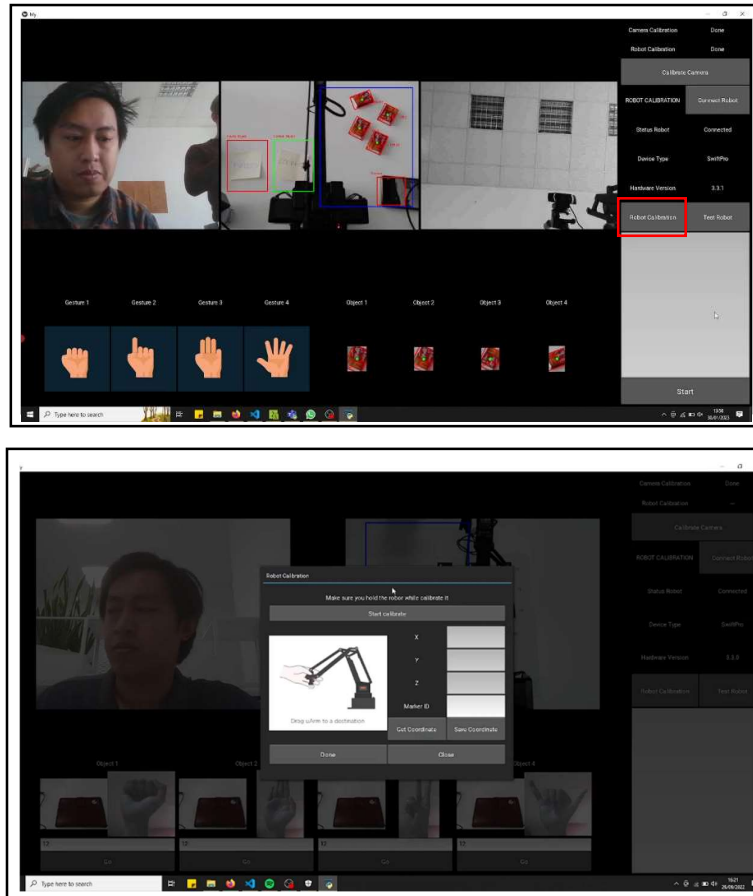


Figure 14. Robot Calibration Button

2. After you click it, it will appear Robot Calibration interface like *Figure 14*
3. **Make sure you read the note in the interface before calibrating the robot.** It is important because the robot will lose its hand power while calibrating proses.



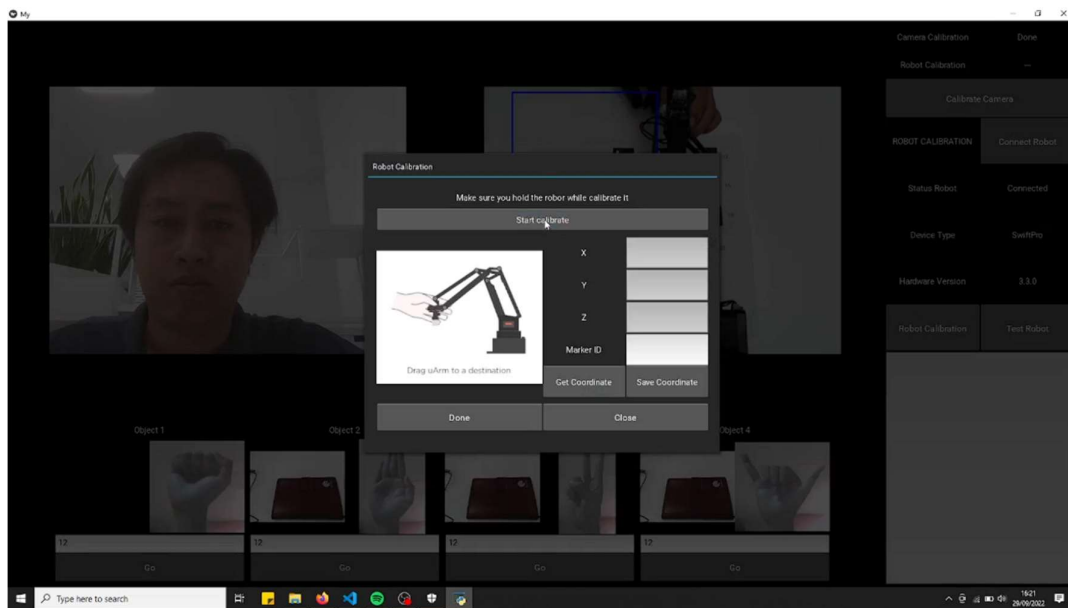


Figure 15. Calibrate the Robot

4. While you hold the robot's hand, click the **Start Calibrate** button to start calibrating the robot. After you click it, the robot will lose its power hand.
5. Try moving the robot slowly to ensure it loses power.
6. For this step, we will use the ArUco board again and places the ArUco board in front of the robot. We will select four of the markers for this calibration. The marker we select is the marker where the place is in the corner of the board. See Figure 16 for the detail.

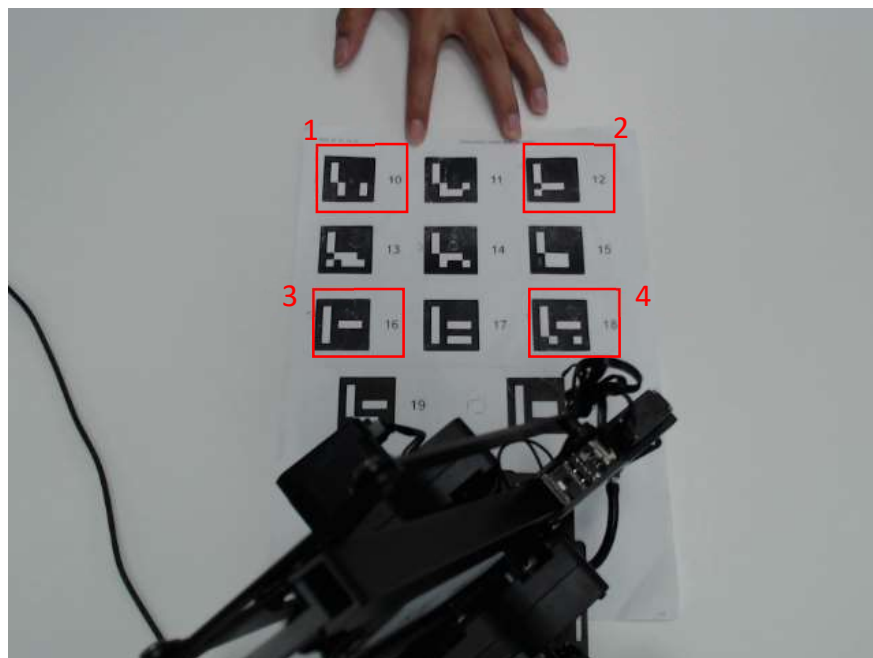


Figure 16. Position of the Marker

7. Pointed the hand of the robot to the center of the marker and clicked the **Get Coordinate** button to get the robot to coordinate. After you click it, the coordinate will show in the interface.

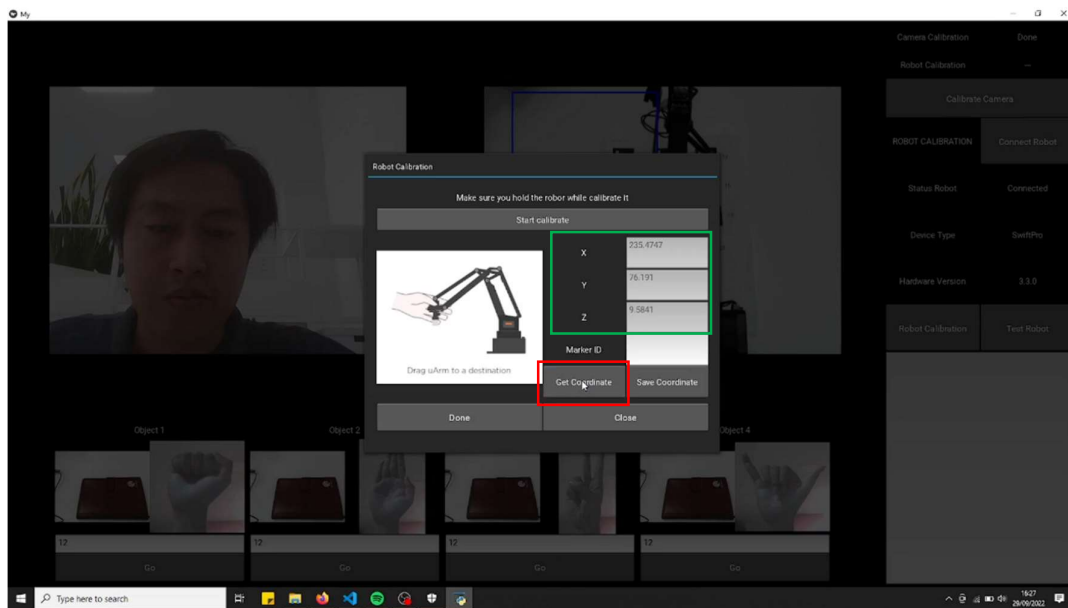


Figure 17. Calibrate the Robot 2

8. Fill out the Marker ID form with the marker ID you choose for calibrating the camera (See Figure 16). Make sure you fill it right because if you fill it with the wrong Marker ID, the robot will fail to recognize the marker and coordinate.

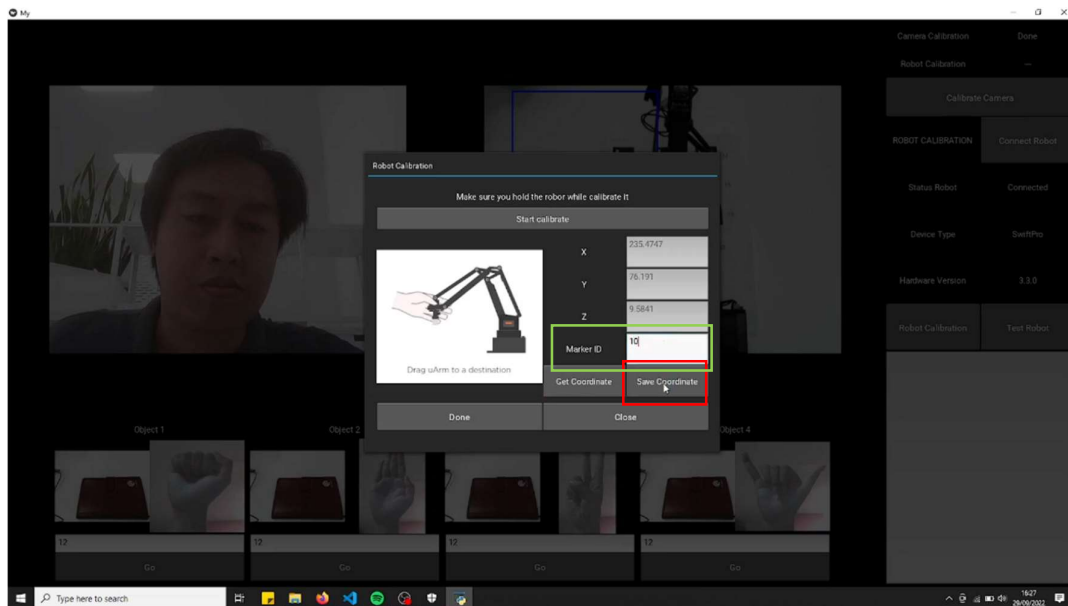


Figure 18. Calibrate the Robot 3

9. Click **Save Coordinate** button to save the coordinate of the robot and the Marker ID
10. Repeat steps four to nine for three markers left.
11. After you get the market ID and the coordinate, don't forget to click the **Save** button in the bottom left corner before you close the interface; this step is important to save all the parameters into a txt file, so we don't have to calibrate the robot again the next time as long as the marker position and robot position doesn't change

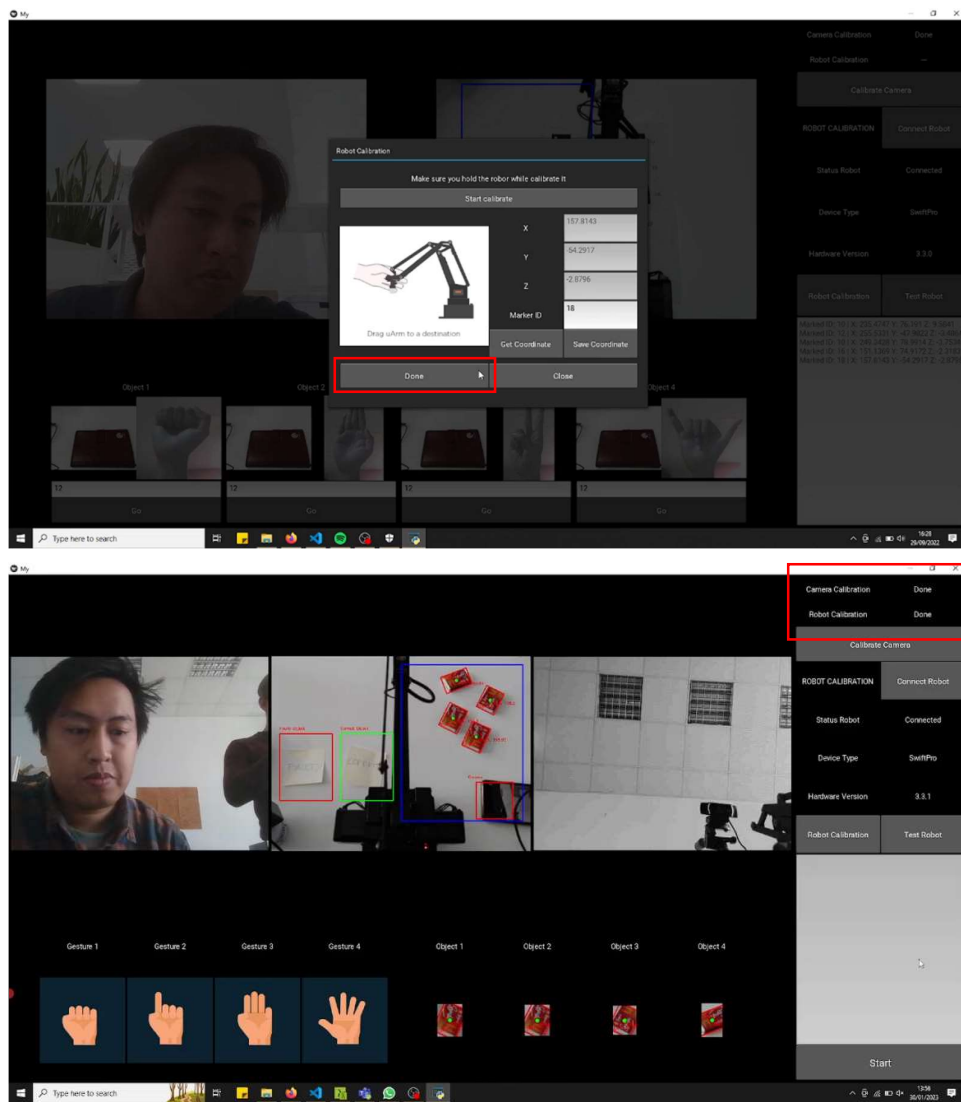


Figure 19. Robot Calibration Status

12. After you click the **Close** button, you can see in the upper right corner that the status of Robot Calibration has been changed to Done. It means we successfully calibrate the robot.

#### 4. Test Robot

This step is mandatory for you to do because, in this step, we will test our calibration. In this step, we will give the command to the robot to move to the specific ArUco marker. After the command is sent, the robot will move to the marker and touch it, so you must place the ArUco board in front of the robot again (Figure 16). **Ensure the robot's position and the ArUco board are in the same position as we calibrate the robot.**

1. Make sure you already calibrate the Camera and the Robot, and don't forget to connect the robot to the computer or laptop

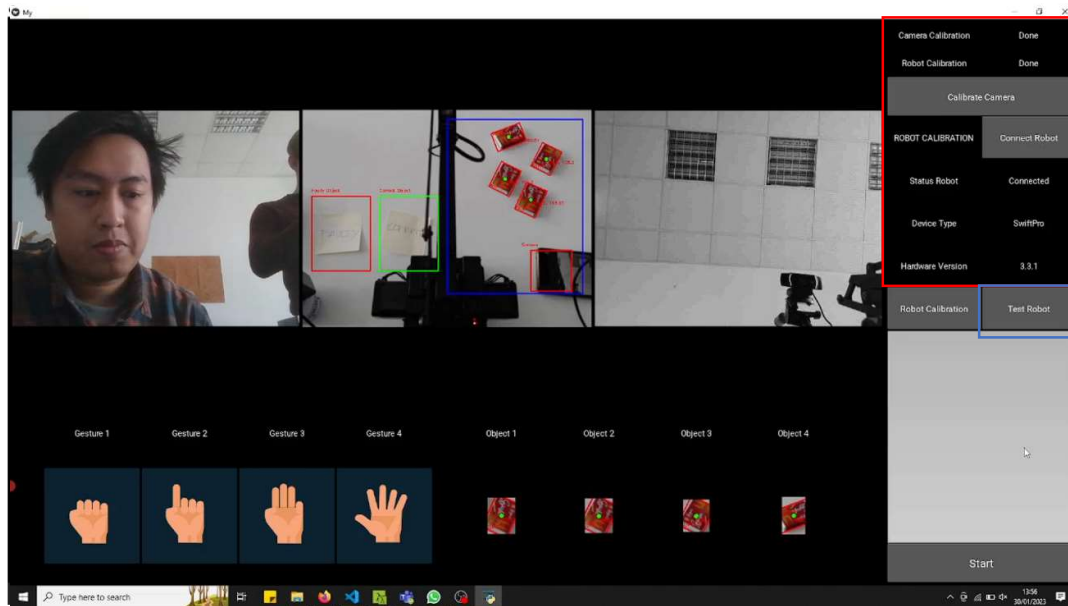


Figure 20. Status of the Calibration

2. After that, click the **Test Robot** button to start testing the robot
3. After you click the button, it will appear Robot Calibration interface like Figure 21

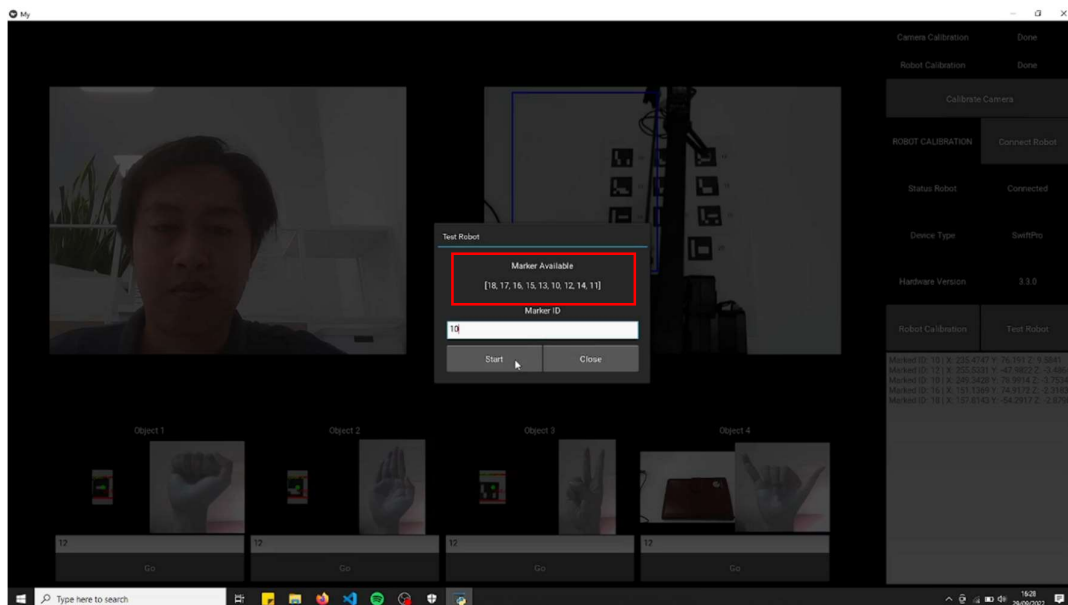


Figure 21. Test the Robot

4. The interface will show you the availability of the marker, and you have to choose one of the available markers. You must fill out the Marker ID form with the number of Marker

ID and click the Start button. After you click it, the robot will touch the marker you choose.

5. **If the robot doesn't touch the marker, you have to calibrate the robot again.**

## 5. Result

After you Calibrate the Camera, Robot and Test it, you can finally try to move the object using hand gestures. As you can see in Figure 22, the first camera can recognize the user's hand gesture which we will use as a command for the robot, and the second camera will identify the object inside the blue area in the frame. At the bottom of the application is information about which object the application can identify and which hand gesture we must show to move it. There is a form and button for every hand gesture and object. **You must fill the form with the object's height (in millimeters).** There is a button as an alternative if the robot does move while you give a hand gesture.

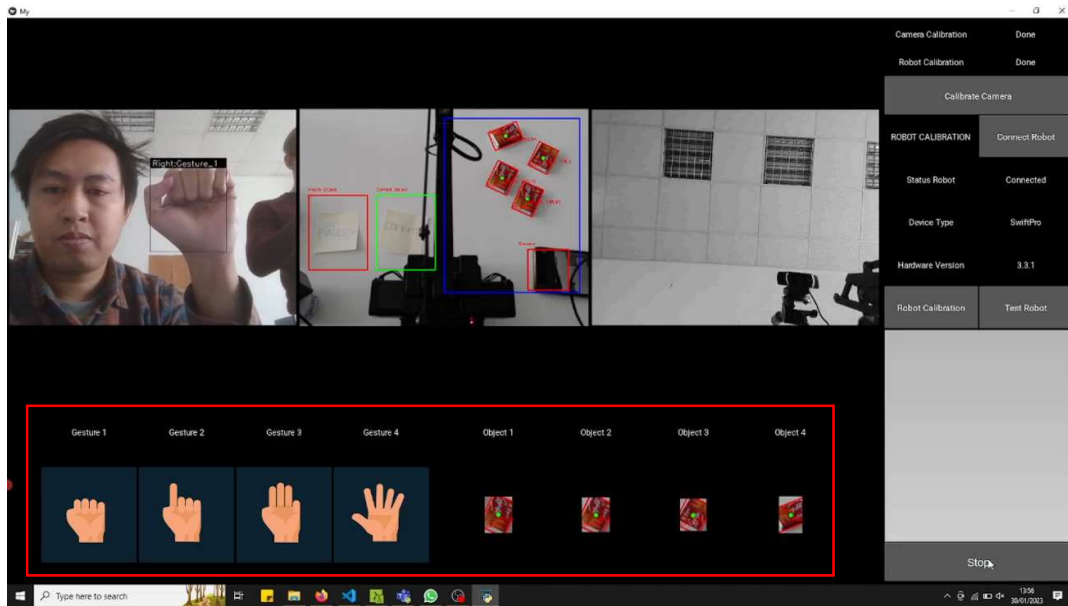


Figure 22. Application Interface

What you have to do after you place the object inside the area is show a hand gesture in front of the camera and hold your position for three seconds. After three seconds, the robot will pick the object based on the hand gesture you show and move it in a specific position.