

Contents

SSH Key Generation	1
Changing and Deploying Code:	7
If testing website code:	8
If testing robot code:	12
If testing OpenCV code:	16

SSH Key Generation

To gain access to the server and robot itself, the developer should follow these steps.

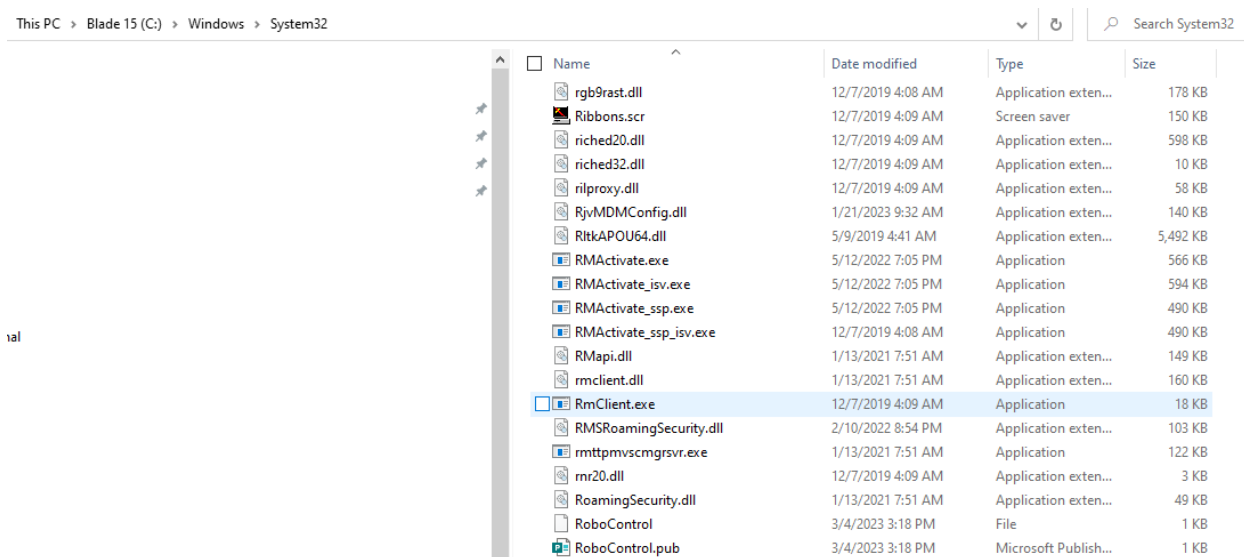
1. Generate ssh keys (preferably using the ed25519 protocol). This protocol is available within OpenSSH version 6.5. To do this on windows, the user should follow the below steps.
 - a. Open a powershell terminal.
 - b. Enter the command without quotations: "ssh-keygen -t ed25519"
 - c. Follow directions to create your public/private key pairs. You should specify a folder to place these keys so as not to be mixed up with other keys.
 - d. Specify a password. Specifying a password provides an extra layer of security. Otherwise, maintain access control over your public/private keys.

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\WINDOWS\system32> ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (C:\Users\kjang/.ssh/id_ed25519): RoboControl
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in RoboControl.
Your public key has been saved in RoboControl.pub.
The key fingerprint is:
SHA256:Zm1NaD8+KVV6h/78jDMyMG7swac8ab7HITmU9g0zkd8 kjang@LAPTOP-TLOOPD00
The key's randomart image is:
+--[ED25519 256]--+
|
|   .
|  o . .o
| . = oO.
| S *ooO..
| + oo==.= E
| .o+oo=.o
| .+*=o
| ooo=*++o
+-----[SHA256]-----+
PS C:\WINDOWS\system32>
```

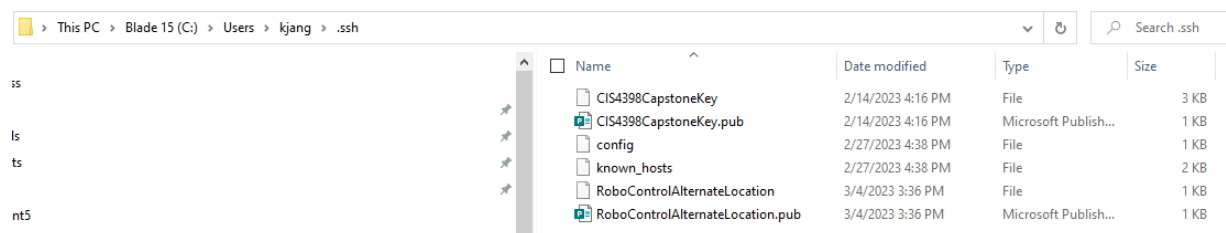
2. Submit your public key to the person with server access control.
 - a. Note: Your public/private keys will be generated within the working directory you executed the command. It is recommended that you change your working directory.



Notice the current working directory for RoboControl public and private keys.

```
PS C:\Users\kjang\.ssh> ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (C:\Users\kjang\.ssh/id_ed25519): RoboControlAlternateLocation
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in RoboControlAlternateLocation.
Your public key has been saved in RoboControlAlternateLocation.pub.
The key fingerprint is:
SHA256:k3oIoepvACNRp6d7AdK3RnzJYubOrU9I1D1YluYzbz4 kjang@LAPTOP-TLOOPD00
The key's randomart image is:
+--[ED25519 256]--+
| .. . o. |
| .. + o *o |
| .. + @ *oo |
| +. % = +o |
| o.o B S+ |
| o * = o .o |
| . o = = .o |
| . o o . E |
| .o.... . |
+-----[SHA256]-----+
PS C:\Users\kjang\.ssh>
```

Notice the changed working directory for this instance of key generation.



And its appearance within the proper ssh working folder.

- Update your authorized keys with your private key. This can be accomplished by moving your public/private keys to within the ssh working directory.

» This PC » Blade 15 (C:) » Users » kjang » .ssh

	Name	Date modified	Type	Size
	CIS4398CapstoneKey	2/14/2023 4:16 PM	File	3 KB
	CIS4398CapstoneKey.pub	2/14/2023 4:16 PM	Microsoft Publish...	1 KB
	config	2/27/2023 4:38 PM	File	1 KB
	known_hosts	2/27/2023 4:38 PM	File	2 KB
	RoboControl	3/4/2023 3:18 PM	File	1 KB
	RoboControl.pub	3/4/2023 3:18 PM	Microsoft Publish...	1 KB
	RoboControlAlternateLocation	3/4/2023 3:36 PM	File	1 KB
	RoboControlAlternateLocation.pub	3/4/2023 3:36 PM	Microsoft Publish...	1 KB

Notice now that both sets of public/private keys generated are within the .ssh working directory.

4. Update your .ssh/config files with the following information (Replace RoboControl with the name of your private key. This is denoted by the lack of a .pub file extension):

Host Tread-Bot-Server

HostName 100.14.184.218

User robot

IdentityFile "~/.ssh/RoboControl"

Port 2626

Host Tread-Bot

HostName 192.168.2.3

User robot

IdentityFile "~/.ssh/RoboControl"

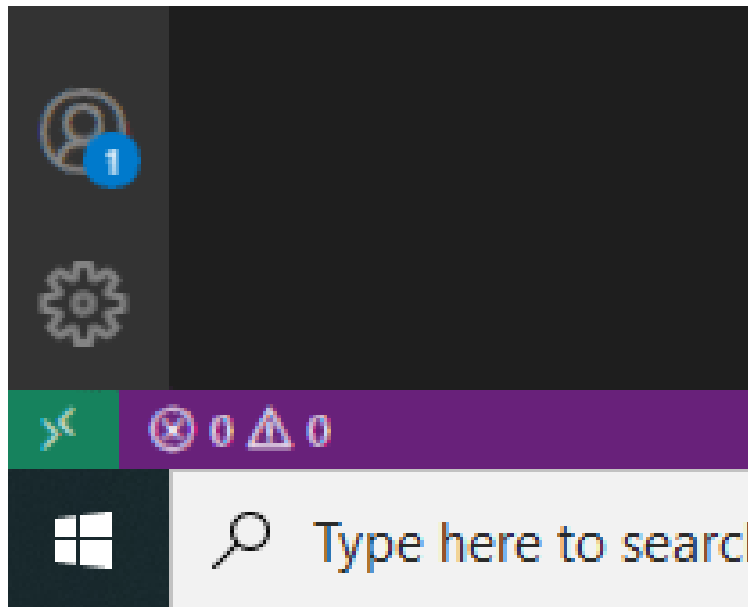
ProxyJump Tread-Bot-Server

Note here that the host names are static set by IP address. In addition, in order to access the tread-bot itself, we connect through a proxy jump first to the Tread Bot Server and then into the Robot itself.

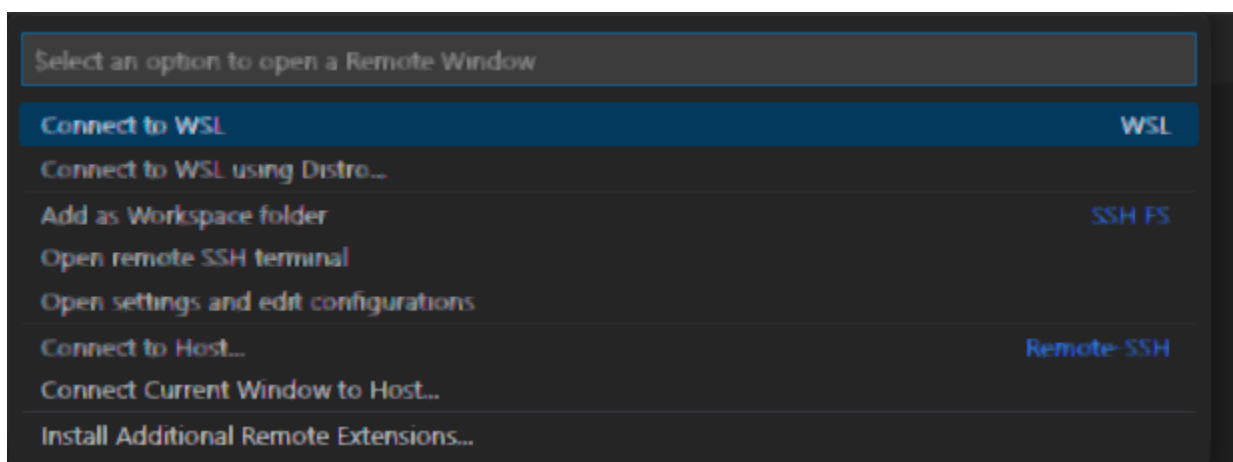
After the above steps are accomplished, now the developer will be able to gain access to both the server and the robot itself.

It is recommended that the developer use Visual Studio IDE in order to gain access through the IDE terminal itself. Access and performance cannot be guaranteed through powershell, windows terminal, or others beside the VSCode Terminal.

1. Open VS Code.
2. On the bottom left portion of your IDE. You should see the remote connection icon as currently disconnected.

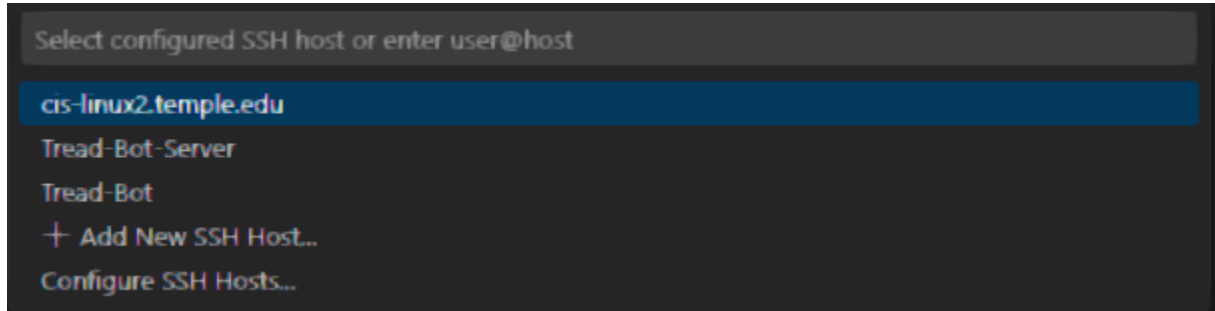


3. Click on the green terminal icon and a connection drop down menu should drop. Select Connect to Host under the Remote-SSH section.

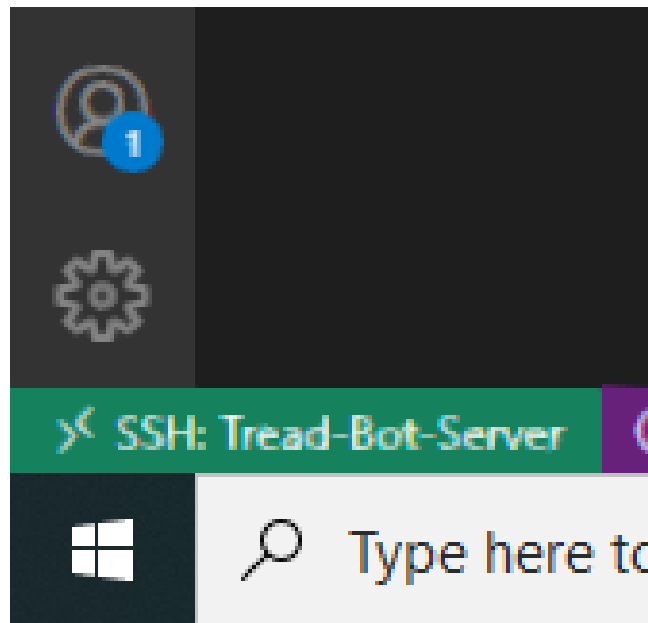


Notice: The Connect to Host option is under the Remote-SSH section. In this example, it is the third option from the bottom.

4. Now Select the tread bot connection you wish to connect to. This will either be the server or the bot itself.



5. Upon selection, a new window should be opened with a successful connection in the bottom left where the green terminal connection status will display which host you are connected to.

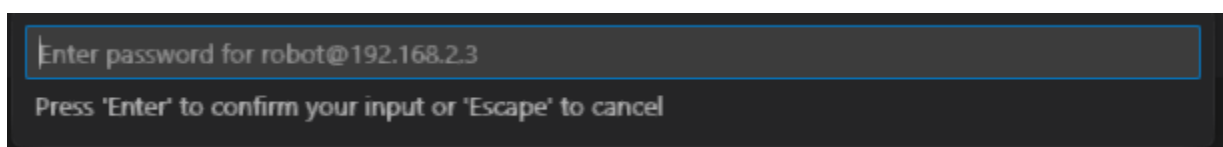


In the above example, we have a successful SSH connection to the Tread-Bot-Server.

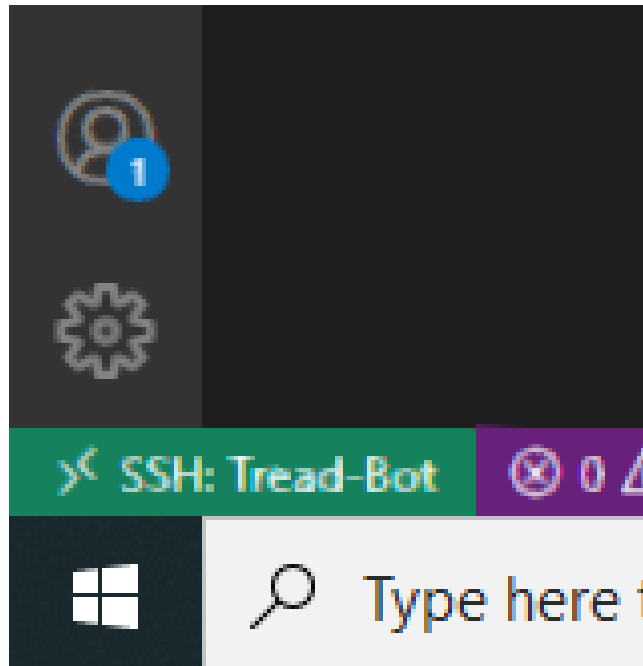
In order to connect to the Tread-Bot itself, we must bring the ssh connection to the Tread-Bot host.

Note: For this to work, the tread bot itself must be turned on.

1. You will be asked for a username and password. The host you will connect to is "robot". The password is also "robot".



2. After entering the password, you should be able to connect to the tread bot itself and a successful connection icon to the host: Tread-Bot should be displayed in the bottom left corner of your IDE.



In the above example, we have a successful SSH connection to the Tread-Bot itself. Note for this to work, the tread bot itself must be turned on. Otherwise, the connection attempt will fail.

Changing and Deploying Code:

Rules:

1. Do not push code while connected to either the Tread-Bot host or the Tread-Bot-Server host. If you push from the Tread-Bot, this may change interactions with other components of code. Only pull code to the Tread-Bot from the main branch. If you push code from the Tread-Bot-Server, all pushes and commits will be made under Ryan's account.
2. Make sure the IDE is disconnected from both the Tread-Bot-Server and Tread-Bot. From here you can push and commit code to your branch. Afterwards, PR's should be made to merge code from branches into main.
3. If you are working on portions that will involve server interaction, the developer should connect only the Tread-Bot-Server after relevant code changes have been made on their branch. The reason for this is so all commits and pushes can be assigned to the appropriate developer without it being mislabeled under Ryan. Then pull from the appropriate branch into the developer's directory. Code should be executed from this directory.

Before testing any code, make sure that your directory is cloned from the github branch.

You can do this from the terminal after connecting to Tread-Bot-Server

```
robot@robot-MS-7C02:~$ ls
Antonio Anu Cindy Downloads Ed flask-video-streaming Kevin Olivia project-robocontrol Qian Ryan snap
robot@robot-MS-7C02:~$ cd Kevin
robot@robot-MS-7C02:~/Kevin$ ls
```

Then run git clone (github link)

```
robot@robot-MS-7C02:~/Kevin$ git clone https://github.com/Capstone-Projects-2023-Spring/project-robocontrol
Cloning into 'project-robocontrol'...
remote: Enumerating objects: 1204, done.
remote: Counting objects: 100% (654/654), done.
remote: Compressing objects: 100% (220/220), done.
remote: Total 1204 (delta 509), reused 512 (delta 429), pack reused 550
Receiving objects: 100% (1204/1204), 9.20 MiB | 20.12 MiB/s, done.
Resolving deltas: 100% (619/619), done.
robot@robot-MS-7C02:~/Kevin$ ls
project-robocontrol
robot@robot-MS-7C02:~/Kevin$ cd project-robocontrol
robot@robot-MS-7C02:~/Kevin/project-robocontrol$ ls
documentation README.md tread-bot tread-bot-opencv tread-bot-website
robot@robot-MS-7C02:~/Kevin/project-robocontrol$
```

Now you have a copy of the main branch in your directory. From here you can make changes and execute code.

If testing website code:

1. Change directory into website directory in YOUR directory

ex: cd Kevin/project-robocontrol/tread-bot-website/

```
robot@robot-MS-7C02:~$ ls
Antonio Anu Cindy Downloads Ed flask-video-streaming Kevin Olivia project-robocontrol Qian Ryan snap
robot@robot-MS-7C02:~$ cd Kevin/project-robocontrol/tread-bot-website/
robot@robot-MS-7C02:~/Kevin/project-robocontrol/tread-bot-website$ ls
nginx-config.txt package.json package-lock.json public README.md src tsconfig.json
robot@robot-MS-7C02:~/Kevin/project-robocontrol/tread-bot-website$
```

2. NOTE: Only do this step right after cloning repo. This step installs typescript libraries

ex: npm i


```

robot@robot-MS-702: ~/Kevin/project/robotcontrol/tread-bot-website$ ls
nginx.config.txt package.json package-lock.json public README.md src tsconfig.json
robot@robot-MS-702: ~/Kevin/project/robotcontrol/tread-bot-website$ npm i
npm WARN deprecated @types/socket.io-client@0.0: This is a stub types definition. socket.io-client provides its own type definitions, so you do not need this installed.
npm WARN deprecated stable@0.1.8: Modern JS already guarantees Array#sort() is a stable sort, so this library is deprecated. See the compatibility table on MDN: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/sort#browser_compatibility
npm WARN deprecated rollup-plugin-terser@7.0.2: This package has been deprecated and is no longer maintained. Please use @rollup/plugin-terser
npm WARN deprecated wc-h-timing@1.0.2: Use your platform's native performance.now() and performance.timeOrigin.
npm WARN deprecated source-map-code@1.4.3: Please use @jridgeway/source-map-code instead
npm WARN deprecated svgo@1.3.2: This SVGO version is no longer supported. Upgrade to v2.x.x.

added 1525 packages, and audited 1526 packages in 9s

236 packages are looking for funding
  run `npm fund` for details

6 high severity vulnerabilities

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
robot@robot-MS-702: ~/Kevin/project/robotcontrol/tread-bot-website$

```

3. Start the website with the following command:

ex: npm start

You should be met with this terminal screen after starting the NGINX.

```

Compiled successfully!

You can now view tread-bot-website in the browser.

Local:      http://localhost:3000
On Your Network:  http://192.168.1.84:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
No issues found.

```

4. Ctrl click or copy Local address(starting with http://localhost:) url into browser.

This starts the NPM server but it is NOT deployed.

To deploy updates to ryanhodge.net, execute: `npm run nginx`

```
robot@robot-MS-7C02:~/Kevin/project-robocontrol/tread-bot-website$ npm run nginx

> tread-bot-website@0.1.0 nginx
> react-scripts build && cp -r ./build/* /var/www/ryanhodge/html

Creating an optimized production build...
Compiled successfully.

File sizes after gzip:

 62.76 kB  build/static/js/main.62a2d2fa.js
 1.78 kB   build/static/js/787.31c89dde.chunk.js
 264 B     build/static/css/main.e6c13ad2.css

The project was built assuming it is hosted at /.
You can control this with the homepage field in your package.json.

The build folder is ready to be deployed.
You may serve it with a static server:

  npm install -g serve
  serve -s build

Find out more about deployment here:

  https://cra.link/deployment

robot@robot-MS-7C02:~/Kevin/project-robocontrol/tread-bot-website$
```

Then go to ryanhodge.net to see website updates. May take a few minutes to update. Note: in order to interact with the robot, the robot should be turned on.

Robot Off:

- Forward
- Backward
- Right
- Left
- Stop

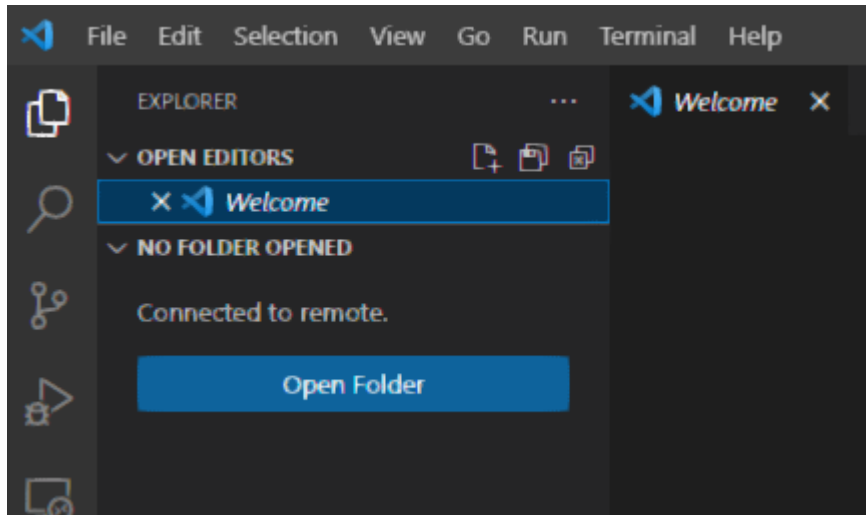
Robot On:

- Forward
- Backward
- Right
- Left
- Stop

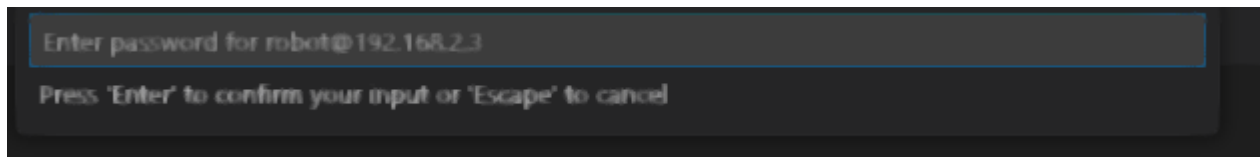


If testing robot code:

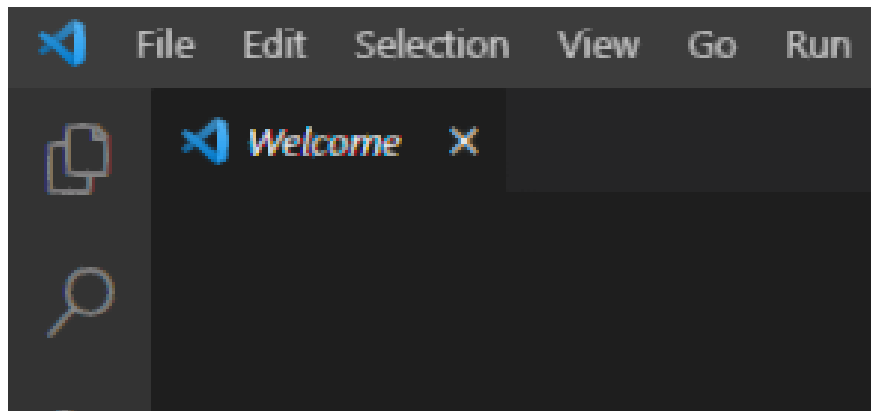
1. Make sure robot is turned on. Otherwise, SSH connection will hang.
2. SSH connect to Tread-Bot.



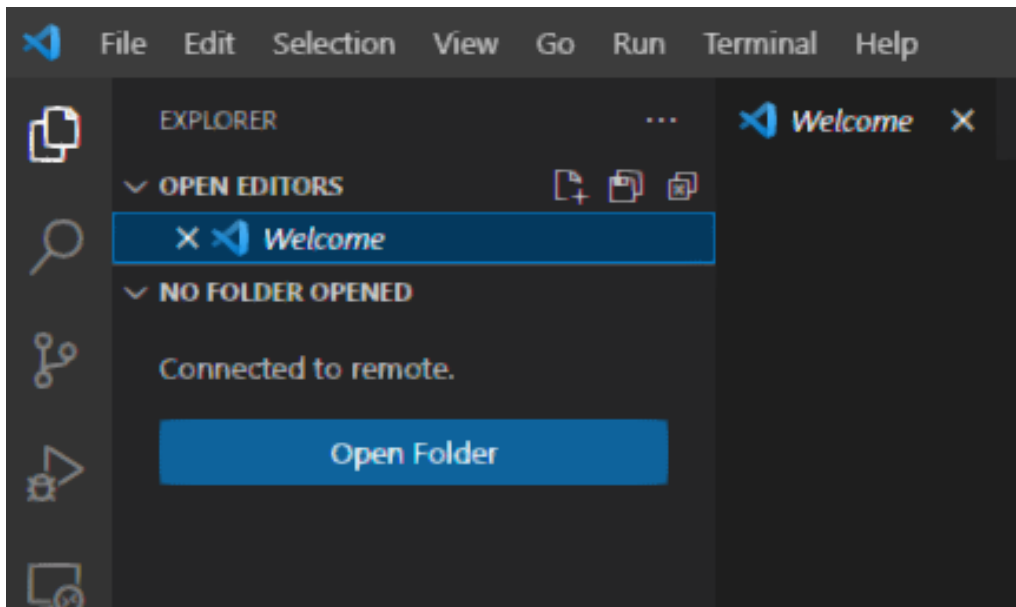
3. Enter password robot.



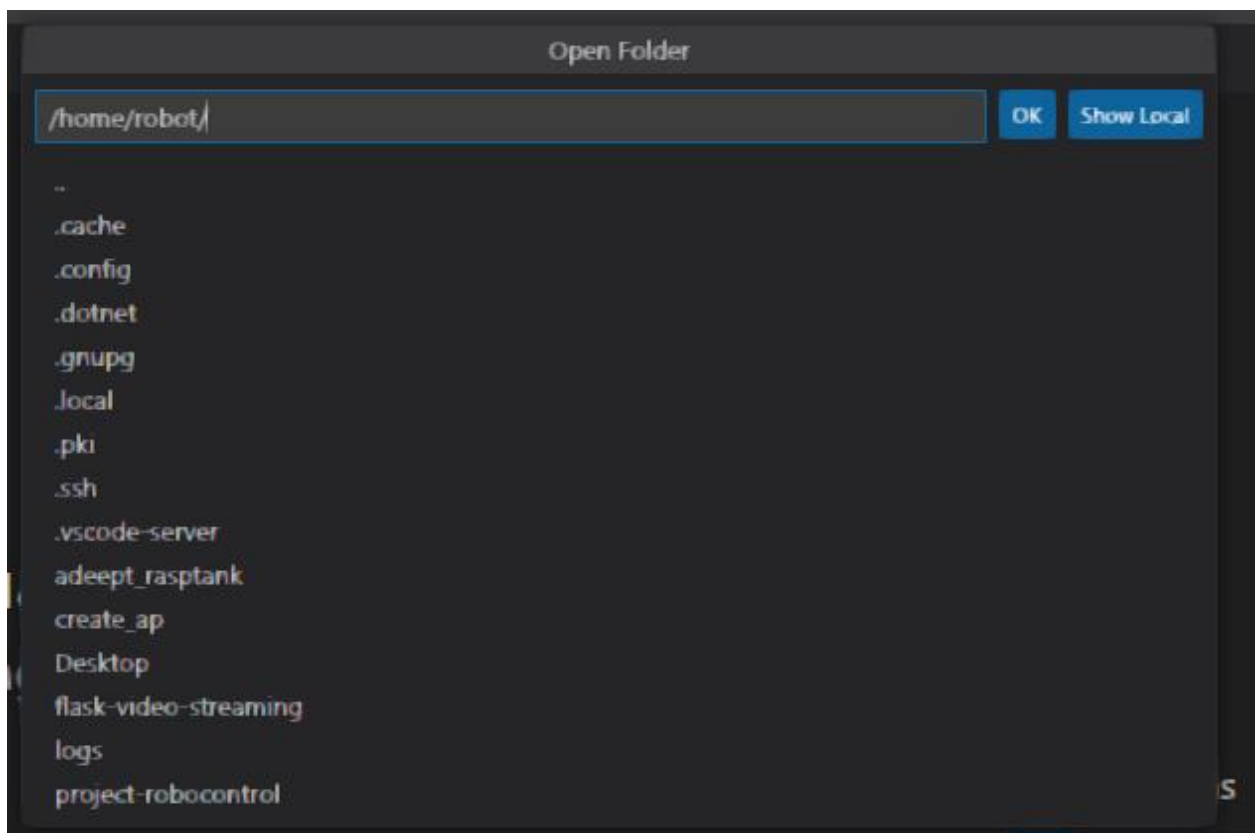
4. Open files on robot by clicking on explorer icon in top left portion of VSCode IDE:



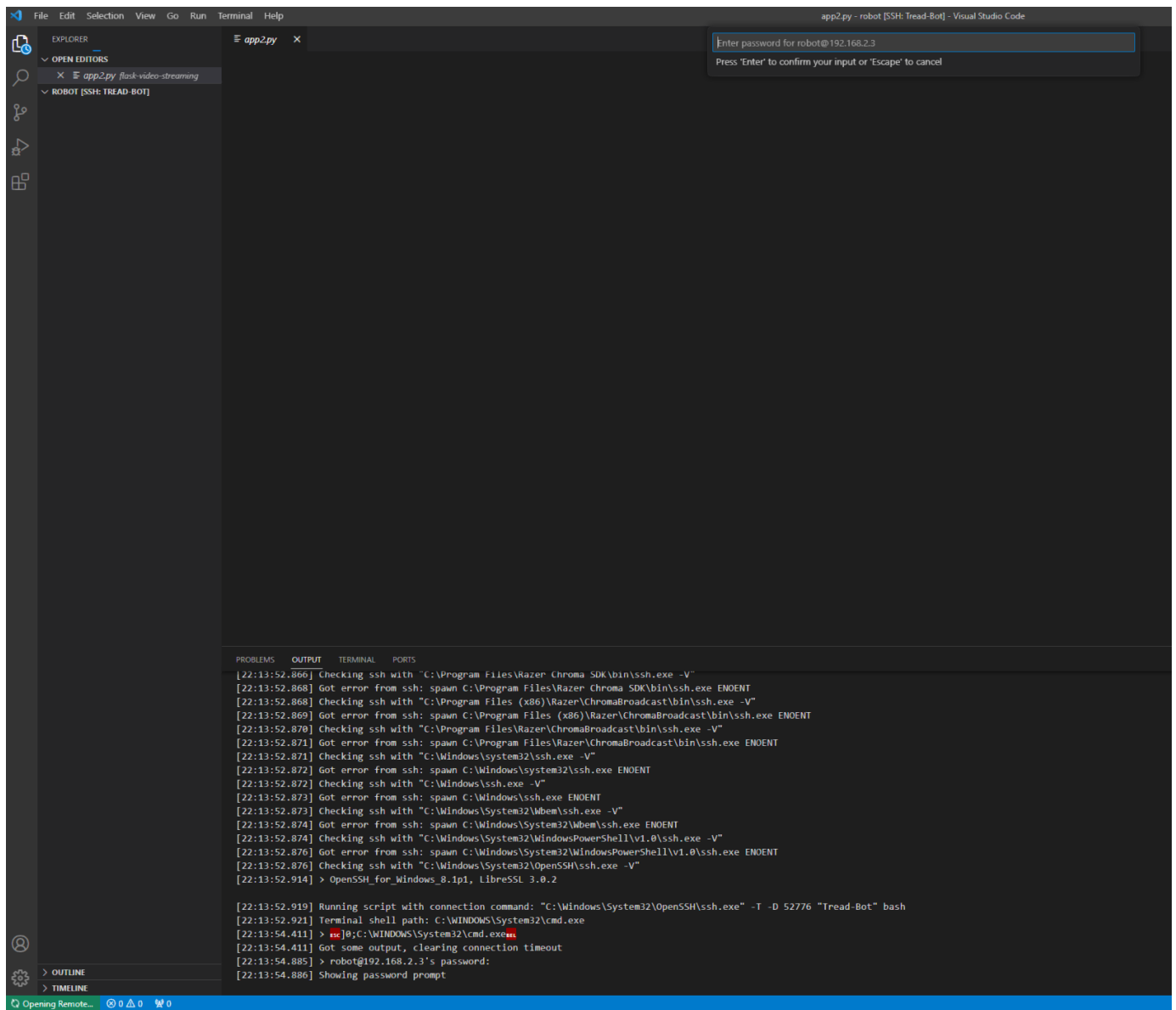
5. Click on open folder option inside explorer.



6. Hit OK to open all files. You do not need to specify which subdirectory you wish to access.



7. Your IDE should not try to reconnect and request access to the bot for file access. Re-enter the same password and you should be given access to the files on the Tread-Bot itself.



8. After entering the password, you should now be given file access and read/write privileges.

```

1 //usr/bin/env python
2 from importlib import import module
3 import os
4 from flask import Flask, render_template, Response
5
6 # import camera driver
7
8 # if os.environ.get('CAMERA'):
9 #     camera = import module('camera' + os.environ['CAMERA']).Camera
10 # else:
11 #     from camera import Camera
12
13 # Raspberry Pi camera module (requires picamera package)
14 from camera_pi import Camera
15
16 app = Flask(__name__)
17
18 #pycache
19
20 @app.route('/')
21 def index():
22     """Video streaming home page."""
23     return render_template('index.html')
24
25 def gen(camera):
26     """Video streaming generator function."""
27     yield b'<frame>'
28     while True:
29         frame = camera.get_frame()
30         yield b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n--frame\r\n'
31
32 @app.route('/video_feed')
33 def video_feed():
34     """Video streaming route. Put this in the src attribute of an img tag."""
35     return Response(gen(camera)),
36         mimetype='multipart/x-mixed-replace; boundary=frame')
37
38 if __name__ == '__main__':
39     app.run(host='0.0.0.0', threaded=True)
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

```

[22:15:14.964] Remote server is listening on 44675
[22:15:14.964] Parsed server configuration: {"ServerConfiguration": {"remoteListening": {"port": 44675, "releaseId": "nauphan", "arch": "armv7l", "webAccessToken": "", "sshAuthSock": "", "display": "", "tmpDir": "/run/user/1000"}, "connectionId": "adaf1111111111111111111111111111", "installImageCode": ""}}
[22:15:14.965] Starting forwarding server. localPort 52788 -> sockPort 52776 -> remotePort 44675
[22:15:14.970] Forwarding server listening on 52788
[22:15:14.970] Waiting for ssh tunnel to be ready
[22:15:14.972] [forwarding server 52788] got connection 0
[22:15:14.974] tunneled 44675 to local port 52788
[22:15:14.974] Reversed "ssh remote+7b22080f73744e31d85223a22547205b36420820770227d" to "127.0.0.1:52788"
[22:15:14.980] -----
[22:15:15.000] [forwarding server 52788] got connection 1
[22:15:15.047] [forwarding server 52788] got connection 2

```

9. Now change your working directory to:

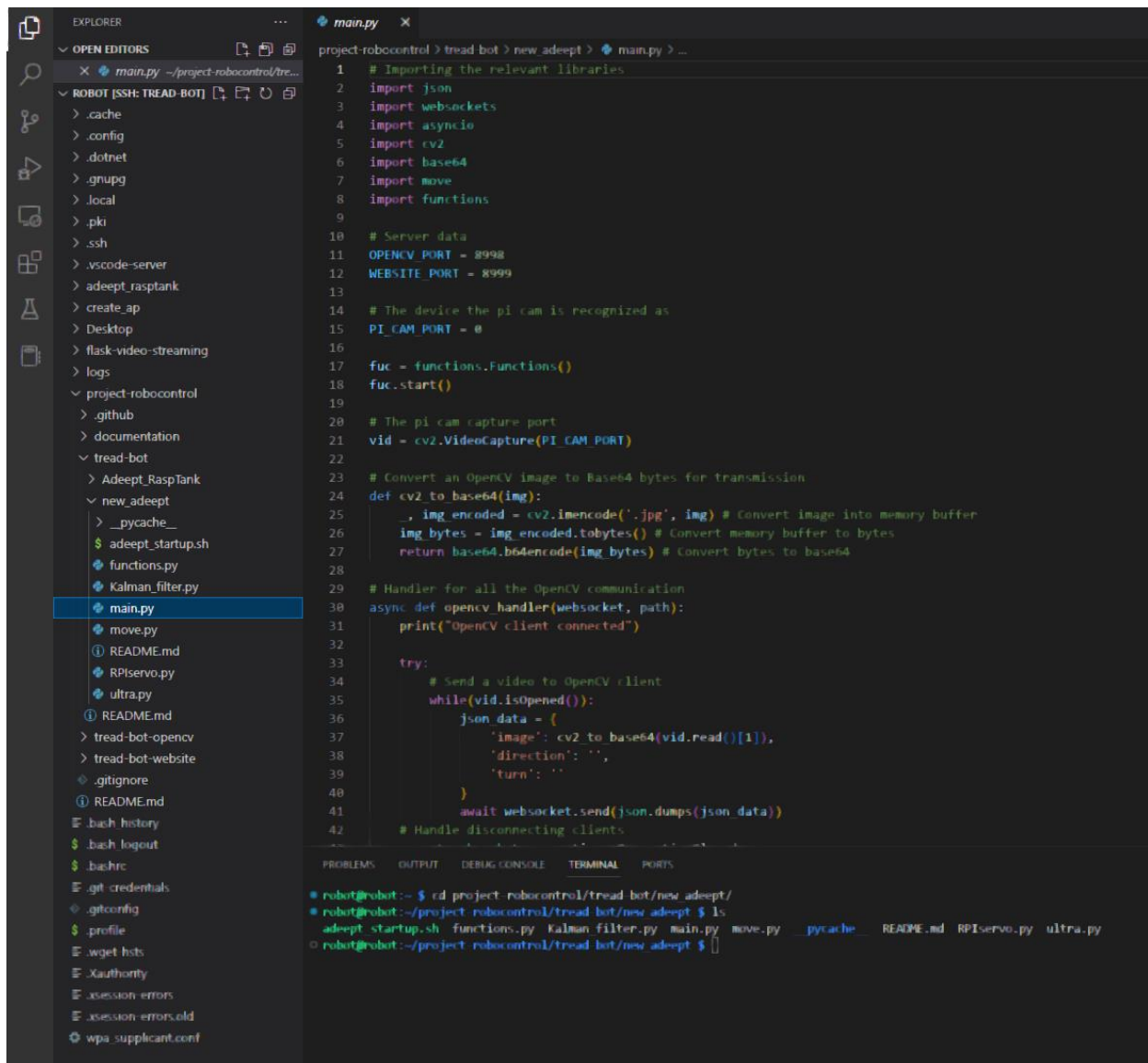
```
cd project-robocontrol/tread-bot/new_adeept/
```

```

robot@robot:~$ cd project-robocontrol/tread-bot/new_adeept/
robot@robot:~/project-robocontrol/tread-bot/new_adeept$ ls
adeept startup.sh functions.py Kalman filter.py main.py move.py __pycache__ README.md RPi servo.py ultra.py
robot@robot:~/project-robocontrol/tread-bot/new_adeept$

```

10. This is the location where changes are pulled from the github within the subdirectory new_adeept. At this moment, in order to execute any changes, run the main.py file. This file is the controller class for the code on the robot.



11. The main code for the robot will be left within the Adept_RaspTank subdirectory. Avoid pushing changes arbitrarily onto this subdirectory as it will affect normal operations within the robot.

If testing OpenCV code:

1. SSH into the tread-bot server with command:
2. Change directory into openCV directory in YOUR directory
ex: cd Kevin/project-robocontrol/tread-bot-opencv/


```
● robot@robot-MS-7C02:~$ cd Kevin/project-robocontrol/tread-bot-opencv/  
● robot@robot-MS-7C02:~/Kevin/project-robocontrol/tread-bot-opencv$ ls  
  README.md  robot_connection.py  
○ robot@robot-MS-7C02:~/Kevin/project-robocontrol/tread-bot-opencv$
```

From here you can checkout and pull from the OpenCV portions of your github branch. Afterwards, you can execute the following command to run your code.

3. Type this command into the terminal:

```
python3 robot_connection.py
```

This will then wait for the tread-bot to become available and execute the code.

```
robot@robot-MS-7C02:~/Kevin/project-robocontrol/tread-bot-opencv$ python3 robot_connection.py  
Listening for ws://192.168.2.3:8998
```