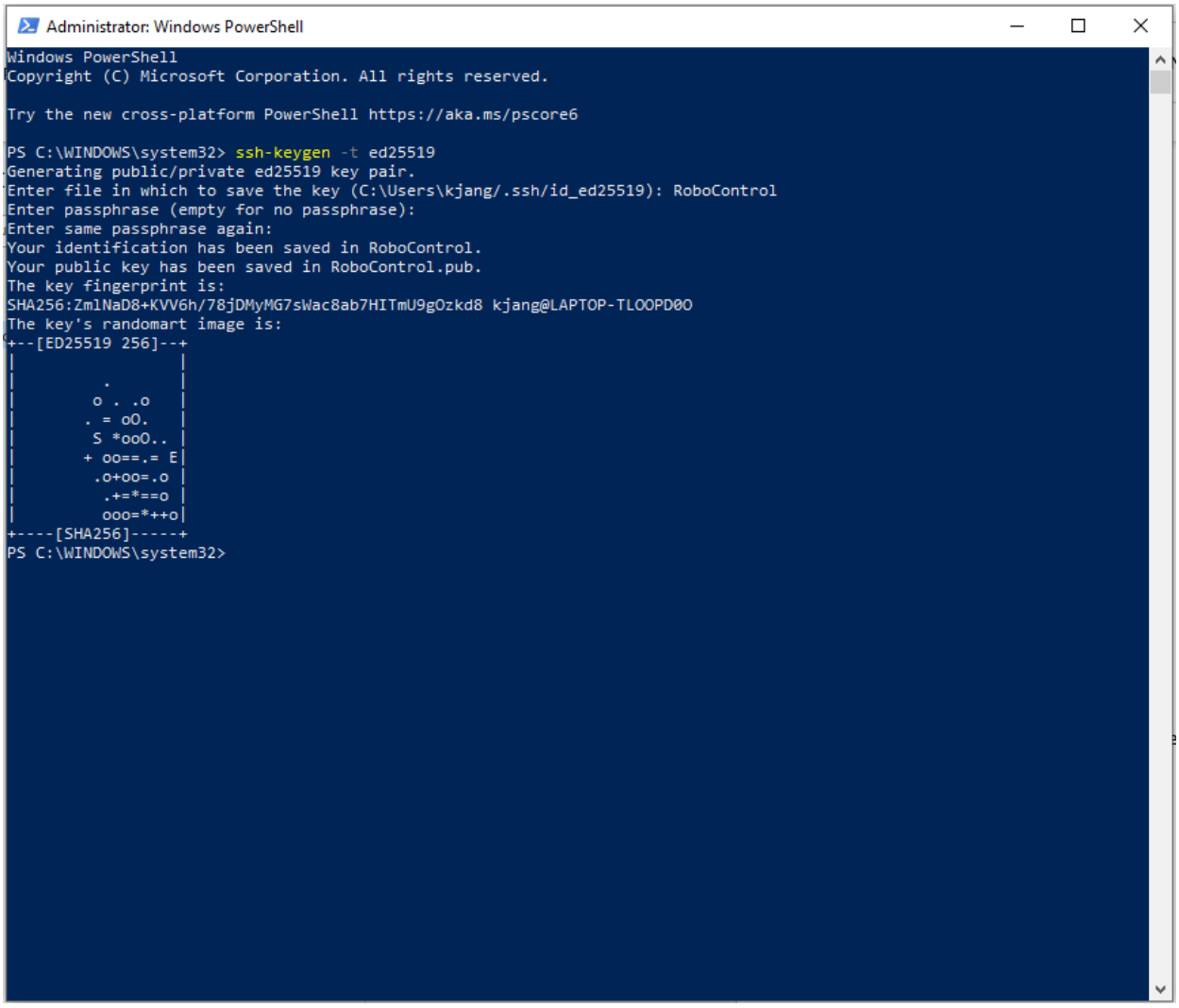
Contents

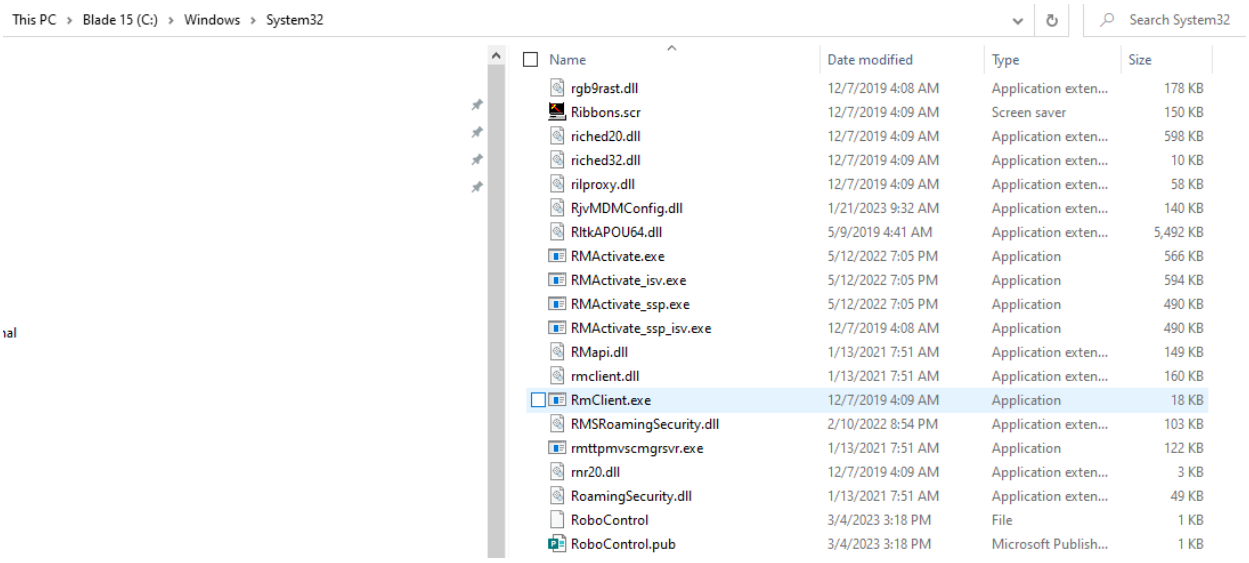
SSH Key Generation

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.



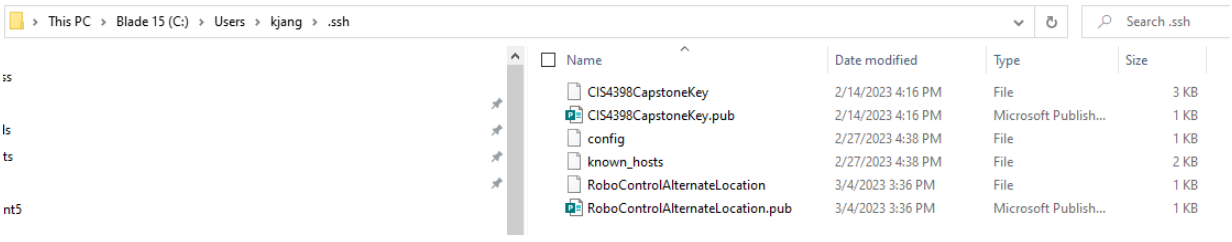
2. Submit your public key to the person with server access control.
    a. Note: Your public/private keys will be generated within the working directory you executed the command. It is recommended that you change your working directory.



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

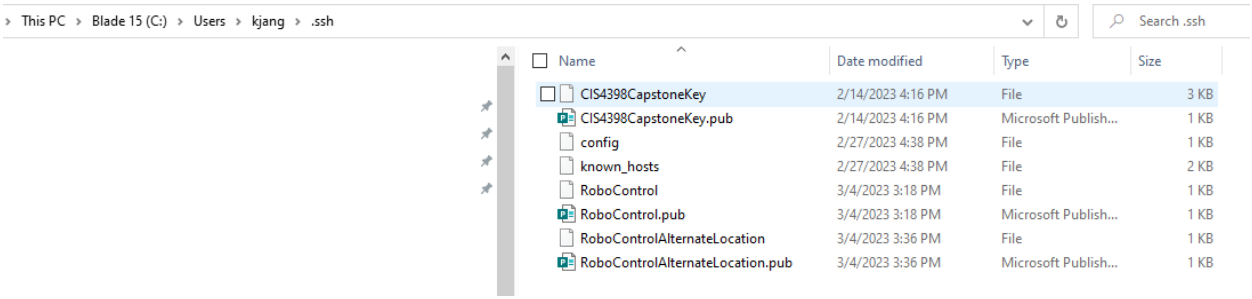
```
PS C:\Users\kjang\.ssh> ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (C:\Users\kjang/.ssh/id_ed25519): RoboControlAlternateLocation
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in RoboControlAlternateLocation.
Your public key has been saved in RoboControlAlternateLocation.pub.
The key fingerprint is:
SHA256:k3oIoepvACNRp6d7AdK3RnzJYubOrU9I1D1YluYzbz4 kjang@LAPTOP-TLOOPD0O
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.

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



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):

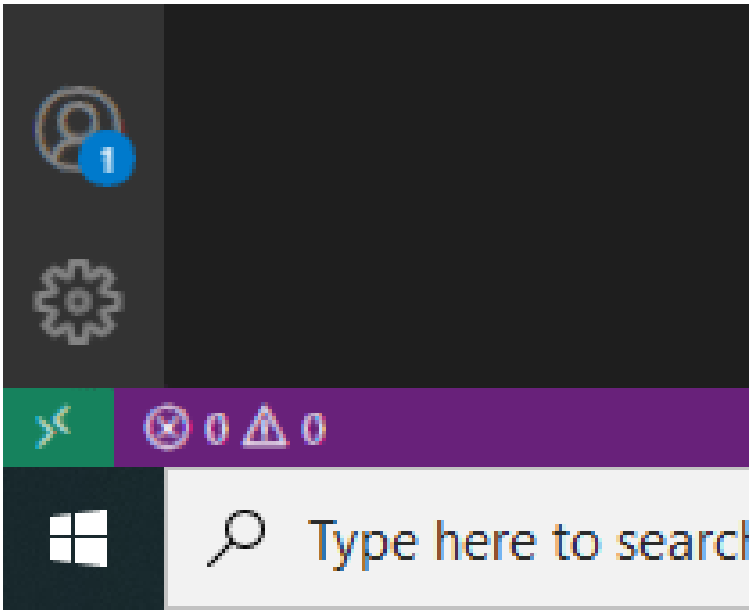
```
Host Tread-Bot-Server
  HostName 100.14.184.218
  User robot
  IdentityFile "~/.ssh/RoboControl"
  Port 2626

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.

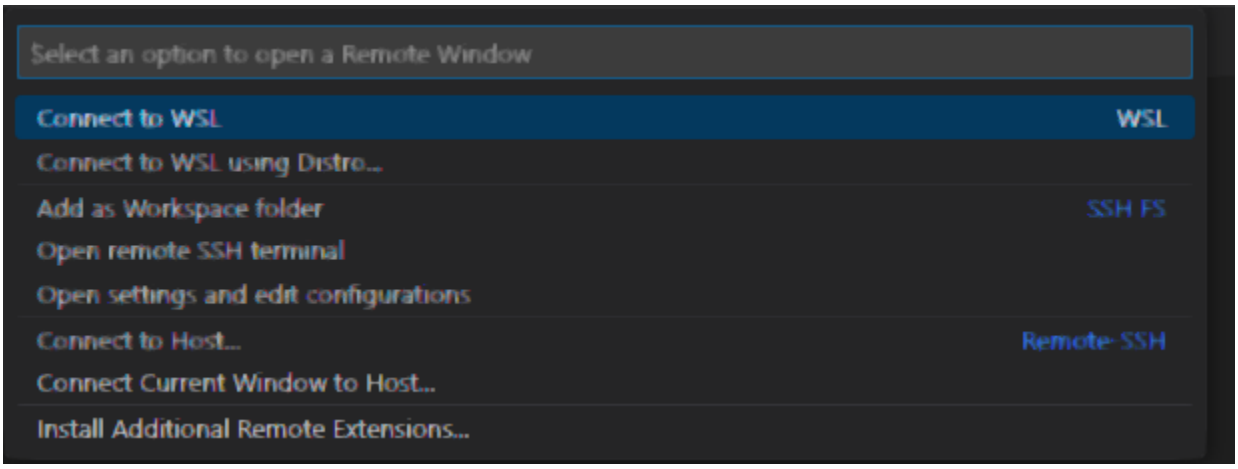
After the above steps are accomplished, now the developer will able to gain access to both the server and the robot itself.

It is recommended that the developer uses 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.

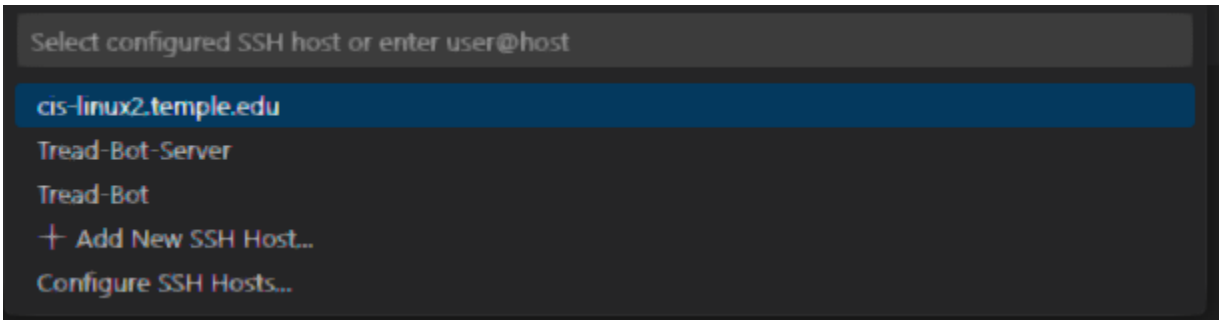


3. Click on the green terminal icon and a connection drop down menu should drop. Select Connect to Host under the Remote-SSH section.

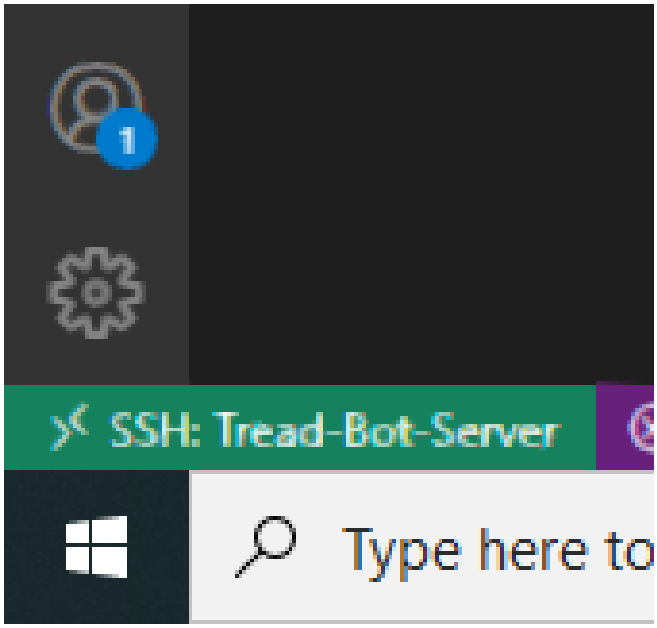


Notice: The Connect to Host option is under the Remote-SSH section. In this example, it is the third option from the bottom.

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



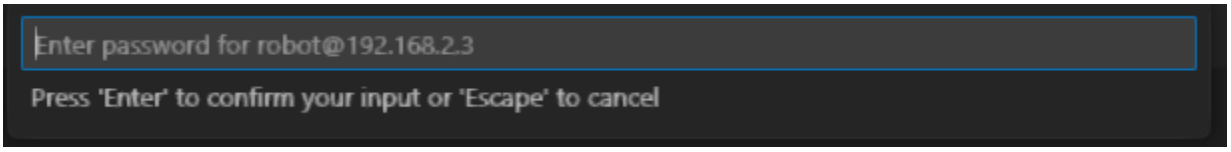
5. Upon selection, a new window should be opened with a successful connection in the bottom left where the green terminal connection status will display which host you are connected to.



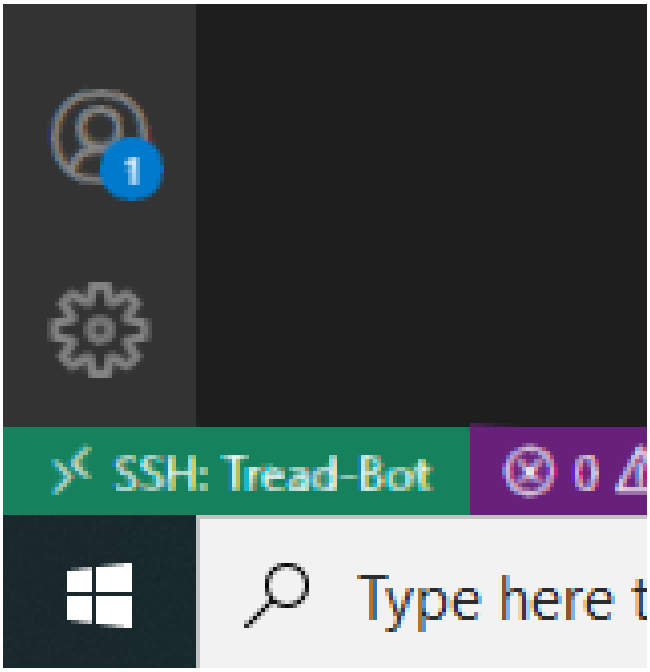
In the above example, we have a successful SSH connection to the Tread-Bot-Server.

In order to connect to the Tread-Bot itself, we must being 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.

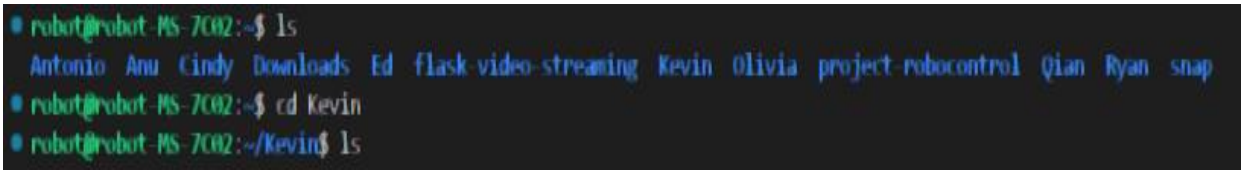
Changing and Deploying Code:

Rules:

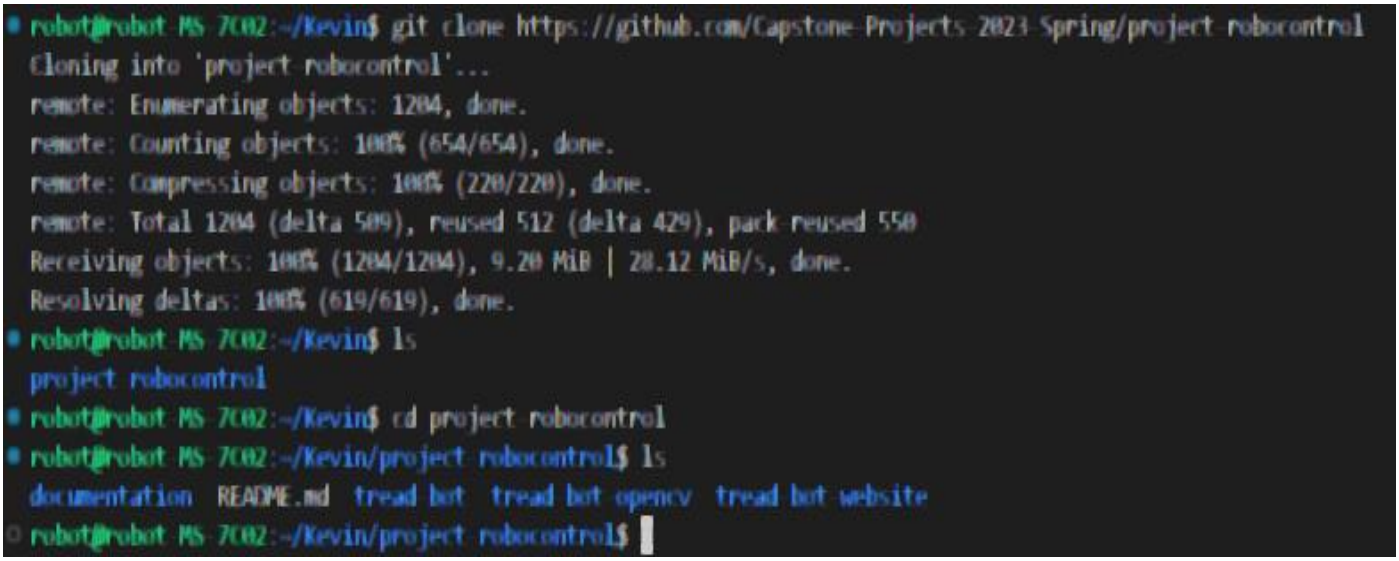
1. Do not push code while connected to either the Tread-Bot host or the Tread-Bot-Server host. If you push from the Tread-Bot, this may change interactions with other components of code. Only pull code to the Tread-Bot from the main branch. If you push code from the Tread-Bot-Server, all pushes and commits 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



Then run git clone (github link)



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/

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

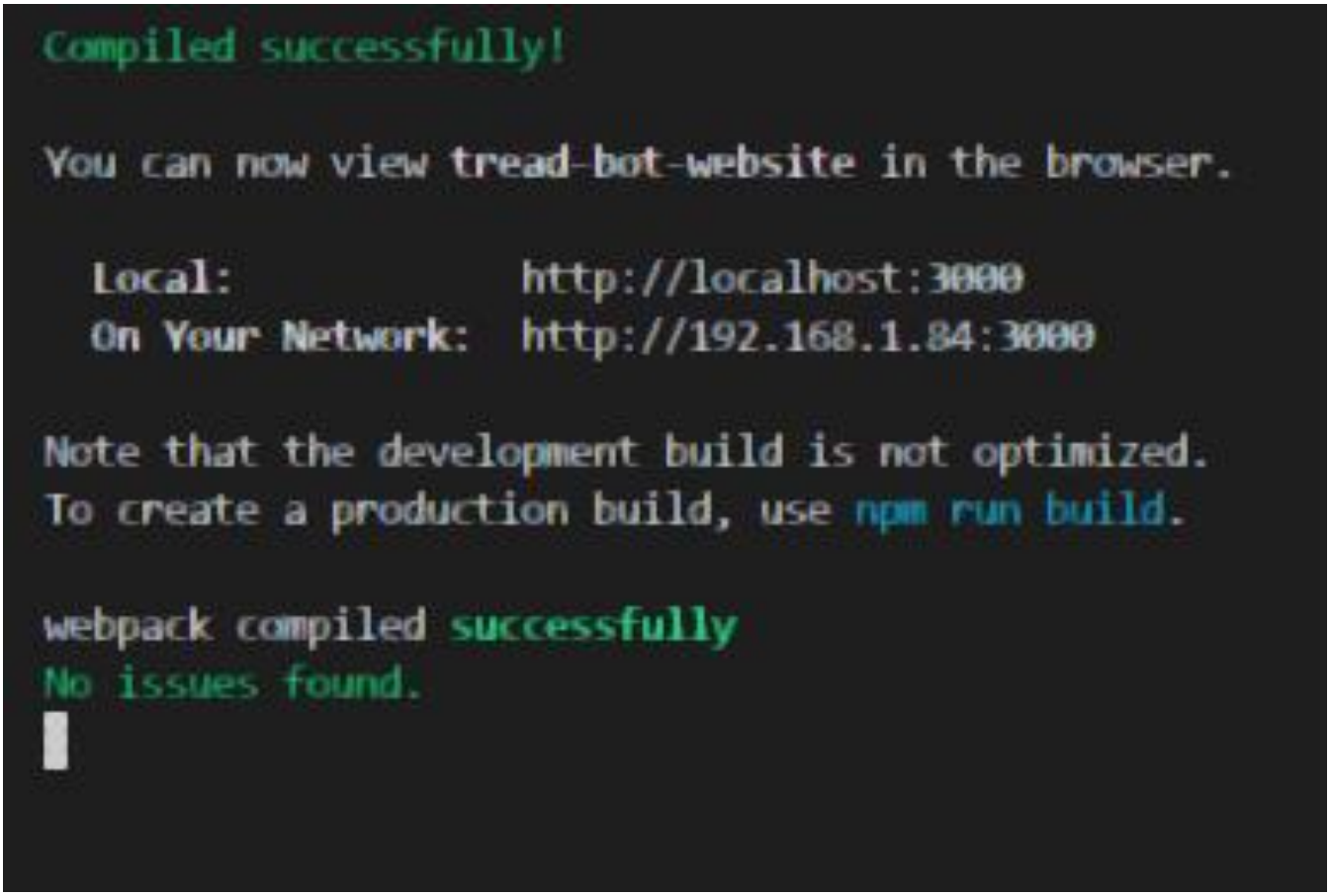
ex: npm i

```
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$ npm i
npm WARN deprecated @types/socket.io-client@3.0.0: 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 w3c-hr-time@1.0.2: Use your platform's native performance.now() and performance.timeOrigin.
npm WARN deprecated sourcemap-codec@1.4.8: Please use @jridgewell/sourcemap-codec 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-7C02:~/Kevin/project-robocontrol/tread-bot-website$
```

3. Start the website with the following command:

ex: npm start

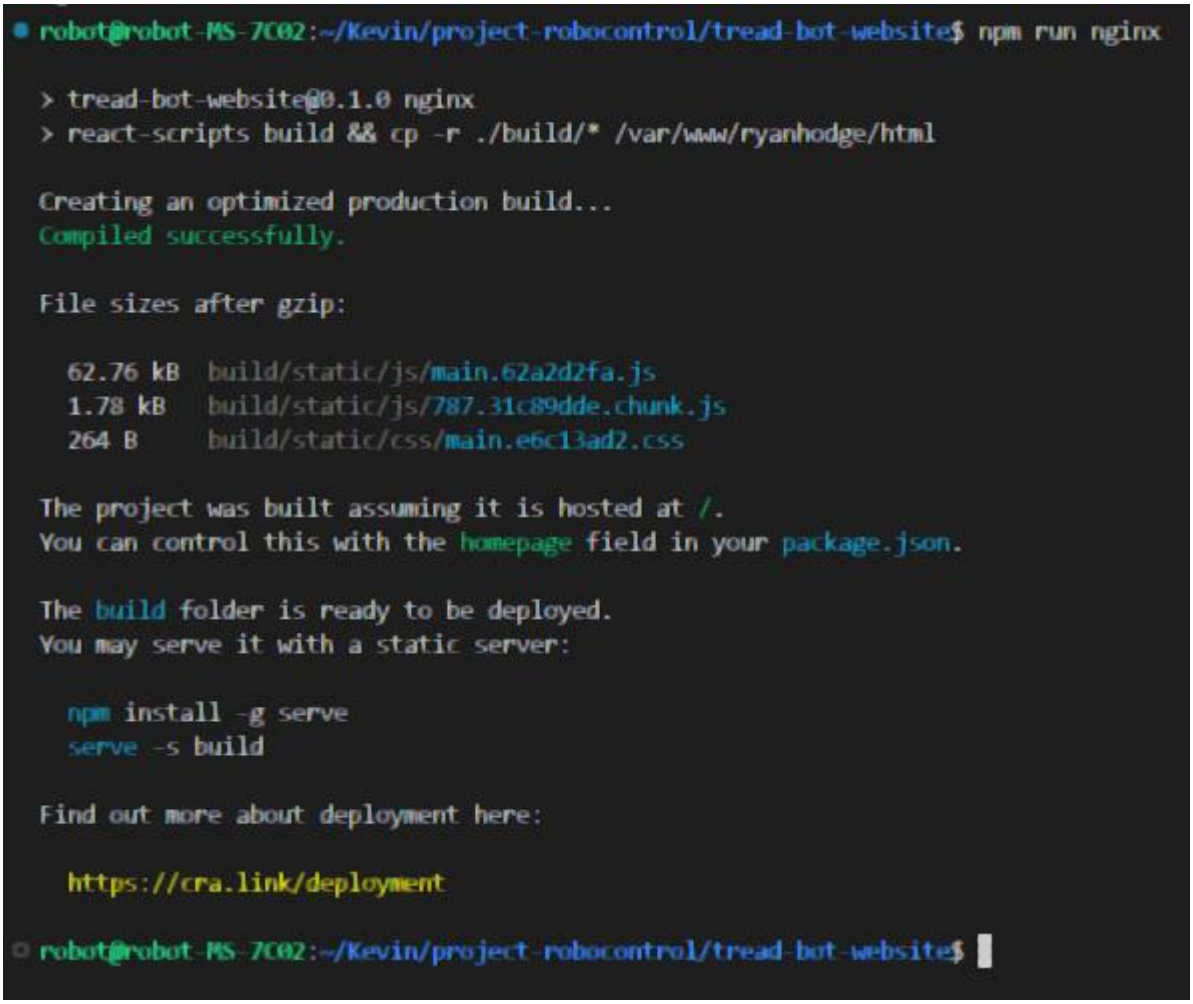
You should be met with this terminal screen after starting the NGINX.



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

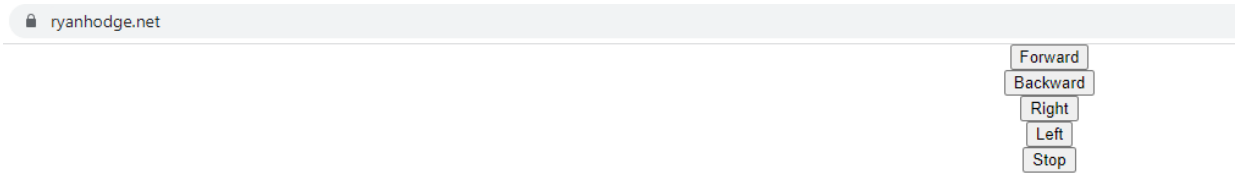
This starts the NPM server but it is NOT deployed.

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

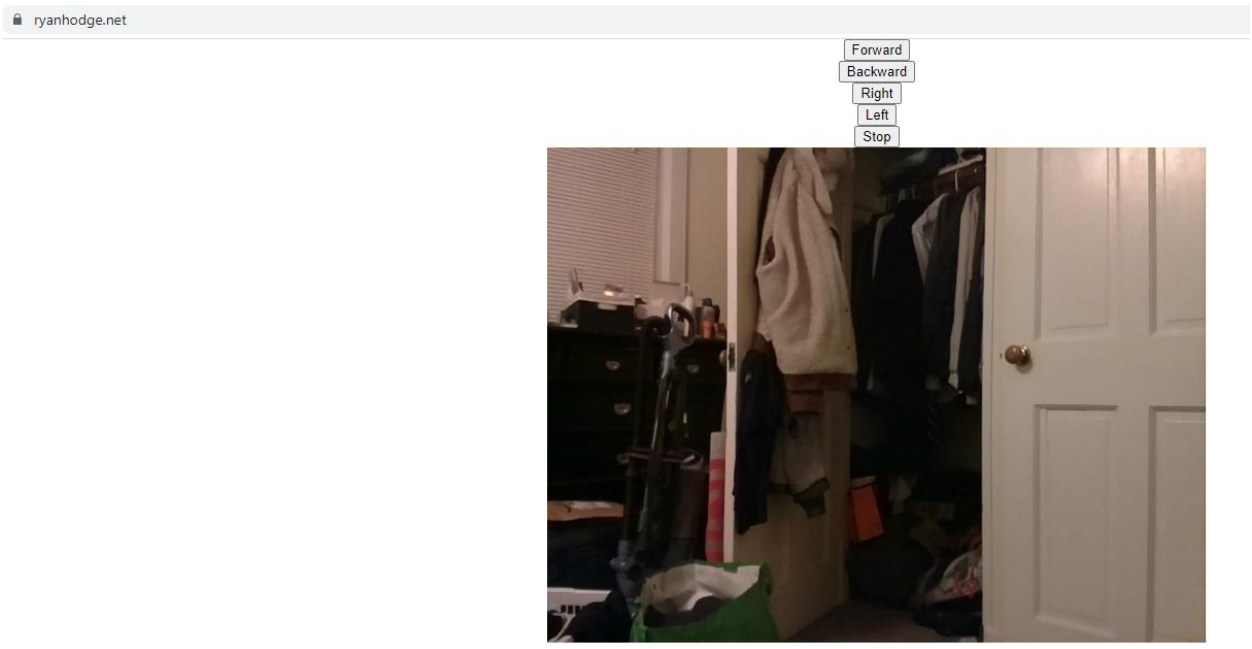


Then go to ryanhodge.net to see website updates. May take a few minutes to update. Note: in order to interact with the robot, the robot should be turned on.

Robot Off:

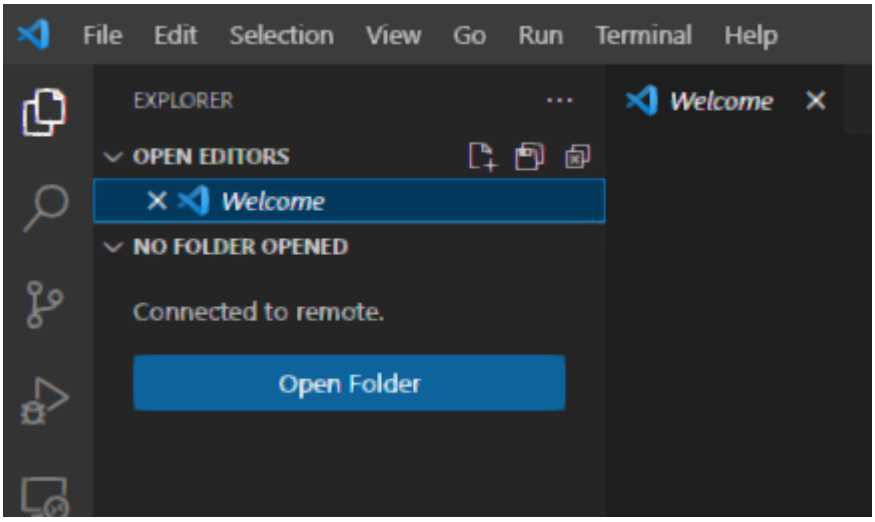


Robot On:

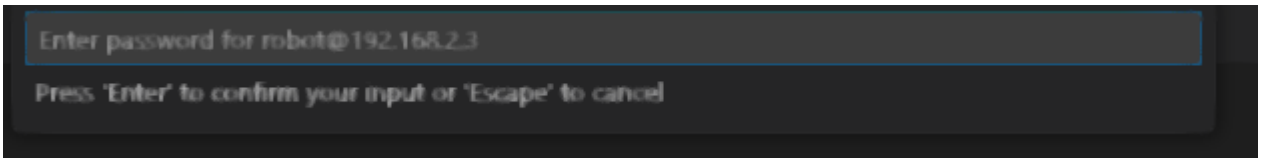


If testing robot code:

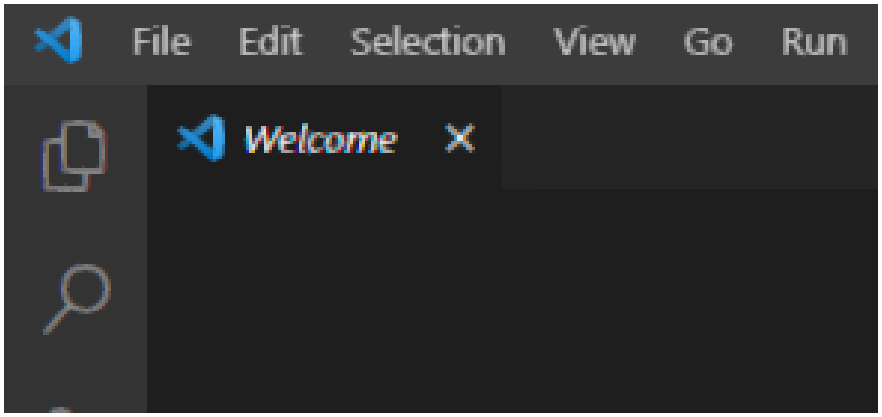
1. Make sure robot is turned on. Otherwise, SSH connection will hang.
2. SSH connect to Tread-Bot.



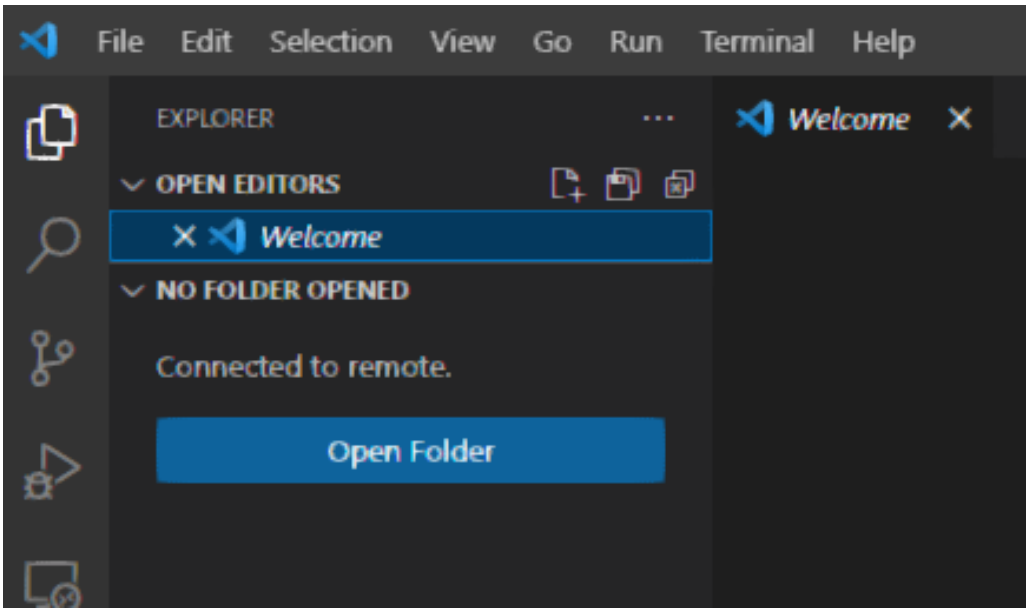
3. Enter password robot.



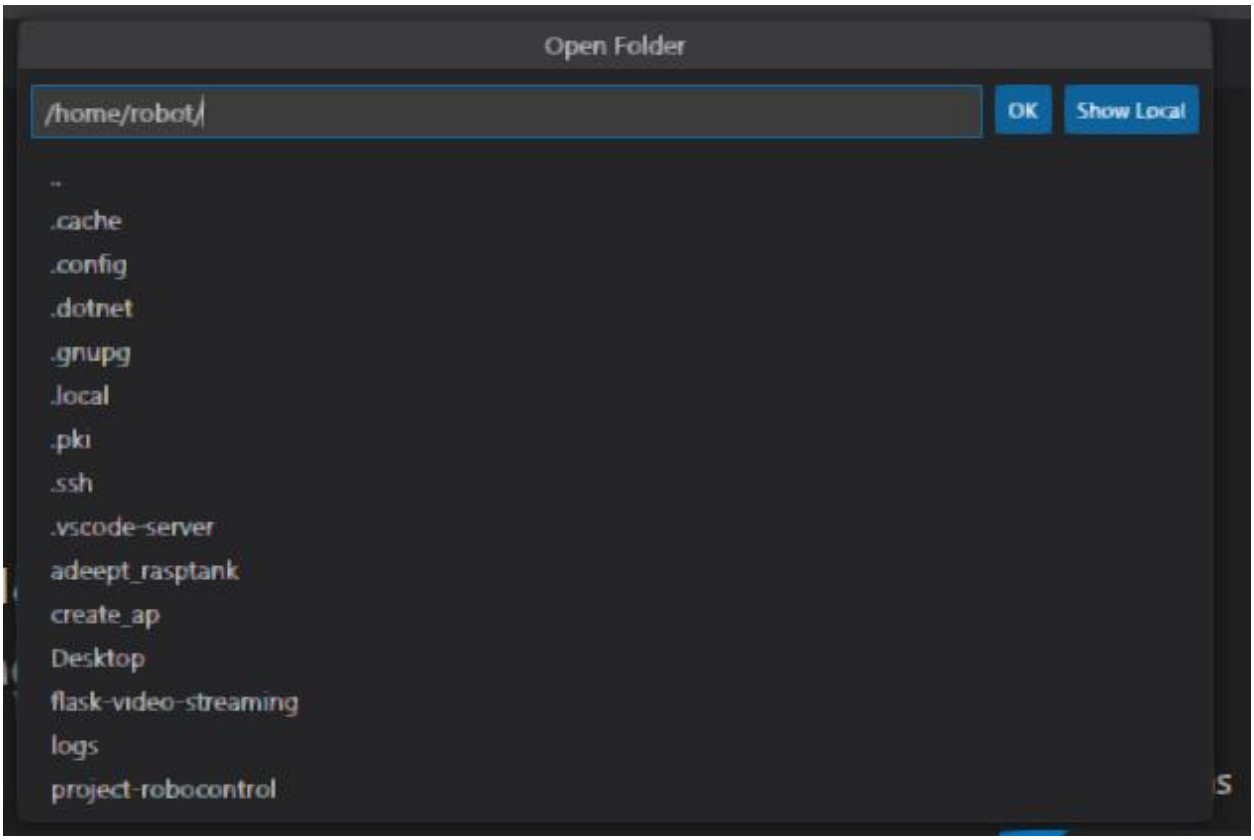
4. Open files on robot by clicking on explorer icon in top left portion of VSCode IDE:



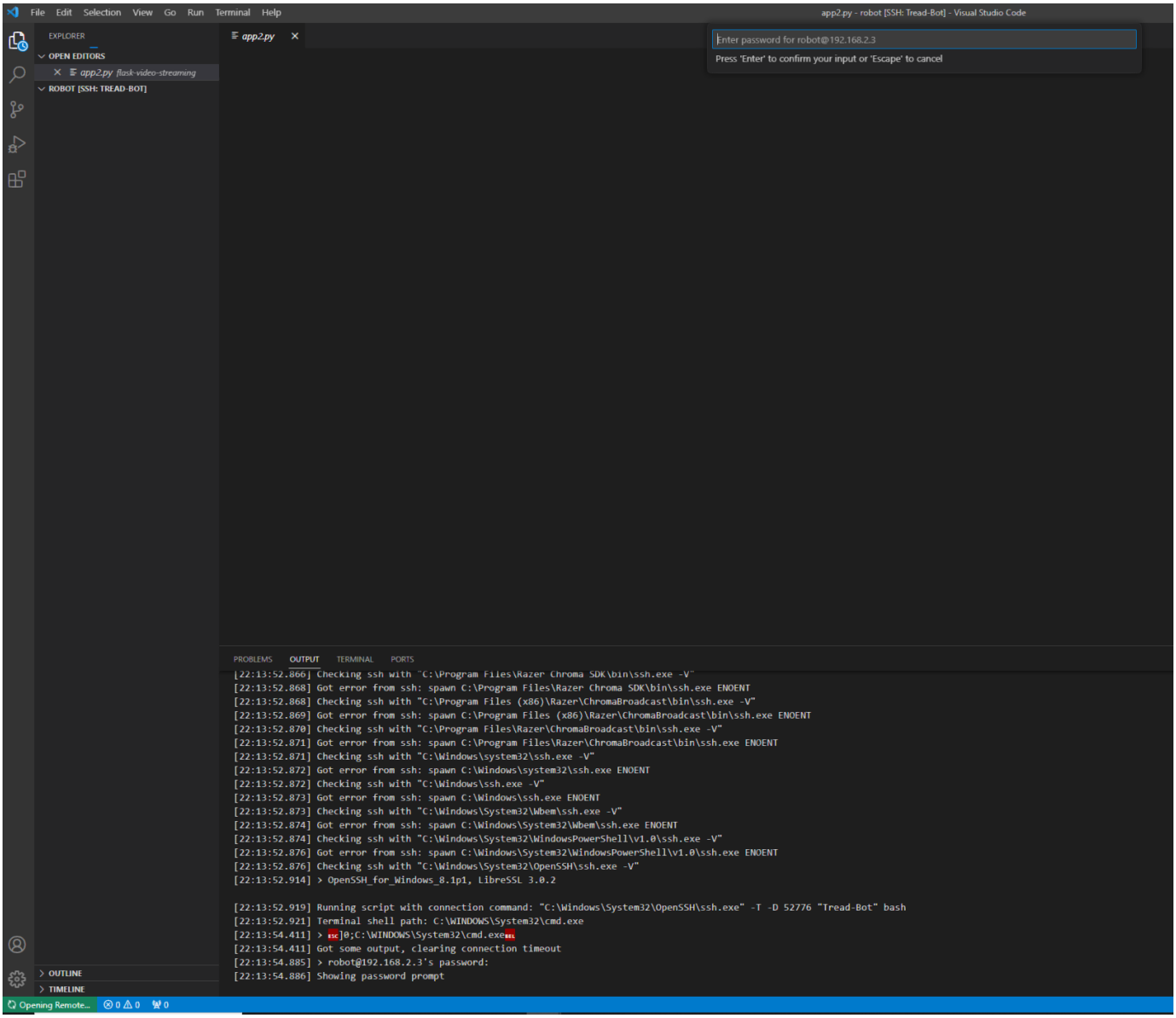
5. Click on open folder option inside explorer.



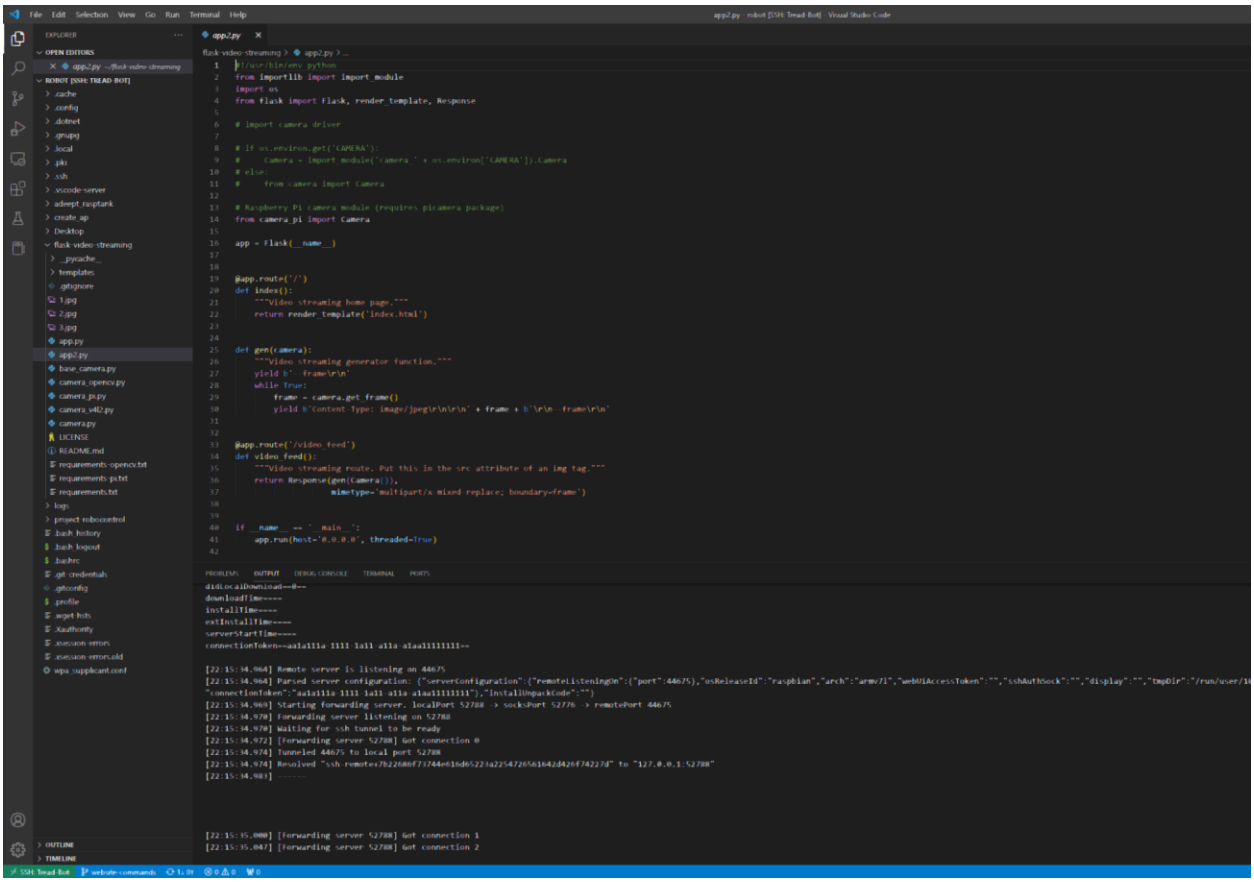
6. Hit OK to open all files. You do not need to specify which subdirectory you wish to access.



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.

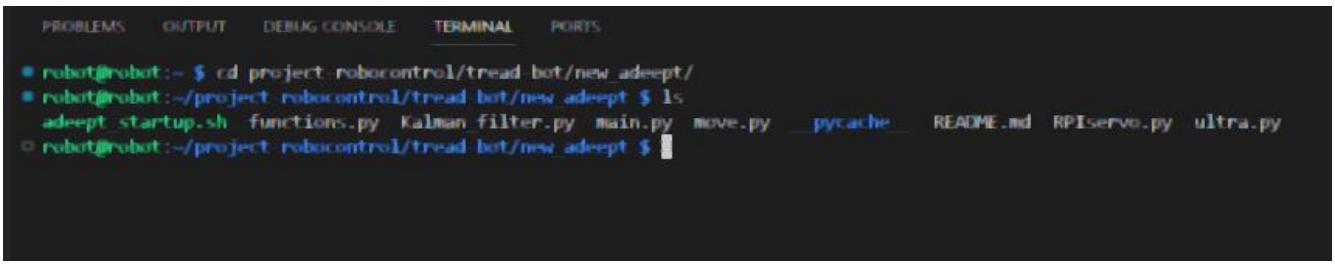


8. After entering the password, you should now be given file access and read/write privileges.

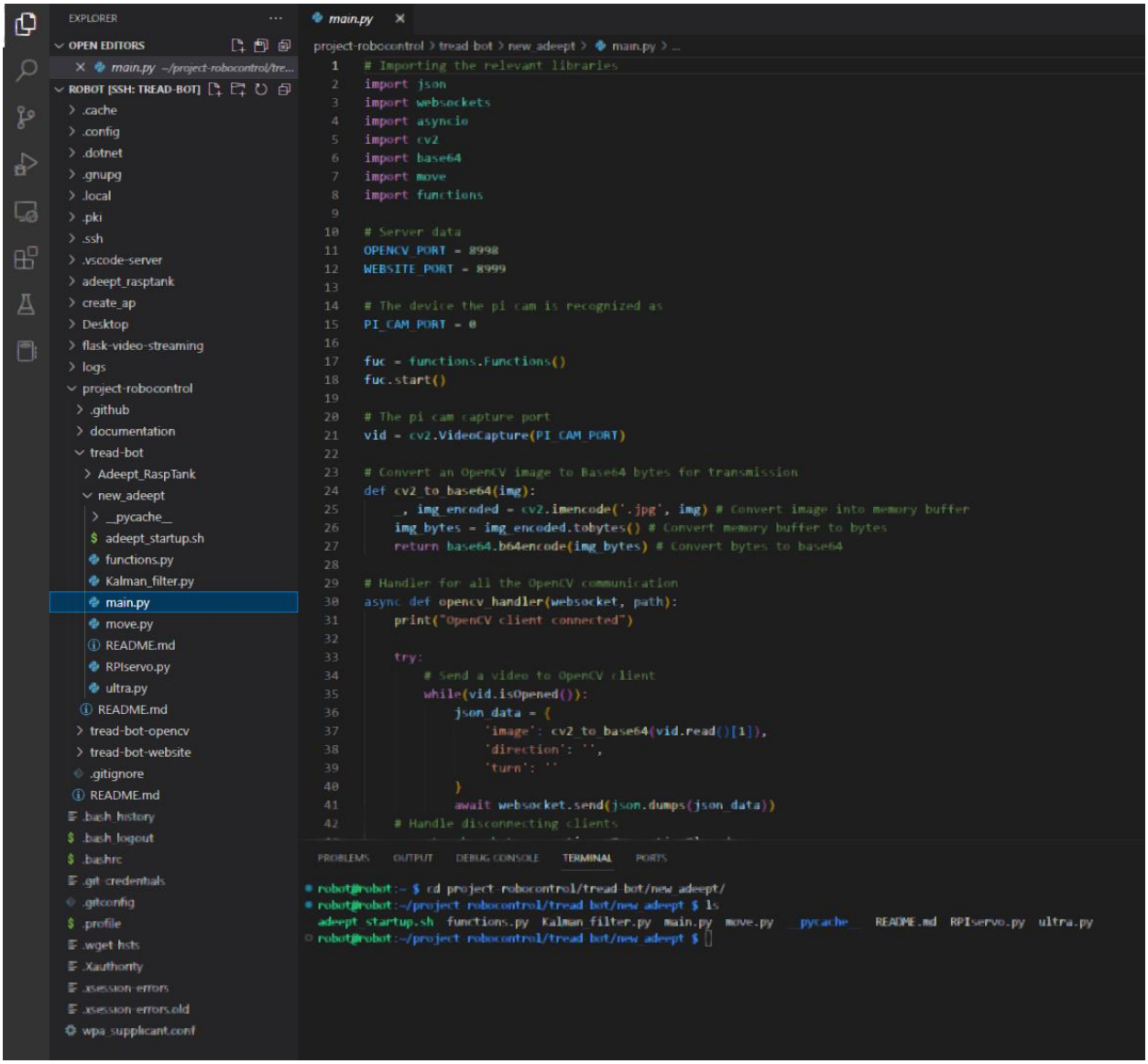


9. Now change your working directory to:

   cd project-robocontrol/tread-bot/new_adeept/



10. This is the location where changes are pulled from the github within the subdirectoy 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 Adeept_RaspTank subdirectory. Avoid pushing changes arbitrarily onto this subdirectory as it will affect normal operations within the robot.

If testing OpenCV code:

1. 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.