Accessing the Robot Remotely

The purpose of this tutorial is to provide the user with the basic tools to:

1. Access the file system of the robot from a remote PC
2. Retrieve files from the robot
3. Deploy and test code remotely on robot

In order to access the robot remotely, the robot must be powered on, logged in, and connected to the university network. If the robot is turned off or loses power at any point, it must be restarted, and a user account must be logged in.

**1.0: Gathering Connection Information**

After logging into the robot, open a terminal and execute ip a or ifconfig to get the IP address of the robot. You will also need to make note of the user name that you have logged into the robot with. From here on you will not need to access the robot directly, but **the robot must remained powered on, connected to the university network, and logged in**.

Text

Description automatically generated

Your terminal window and IP information may appear different than this. The user name is on the cursor line immediately preceding the @ symbol. In this case “joseph”. The IP address can be generally determined by checking the device name in the IP list. In this case, the network device is named wlan0 (wireless local area network 0). Note that the IP address is not fixed and may change when the robot is restarted.

**2.0: Windows Instructions**  
  
The next portion will require installing several pieces of software on your remote PC. Install GIT using all default options in the installation wizard. If you know what you’re doing, you can use other options. This will allow us to use a Git Bash terminal to connect to the robot via SSH and manipulate the file system with the same interface as if you were working directly on the robot, as well as running bash scripts natively in windows. If you don’t plan on using bash scripts on windows, you can choose to only install Putty (next step).

<https://git-scm.com/download/win>

Type Git Bash into your search bar and you should see a shortcut if this was successful.

A screenshot of a computer

Description automatically generated with medium confidence

Next, install Putty. Putty will allow us to use pscp (Putty secure copy) on windows to send and receive files from the robot. It will also give you the ability to perform SSH directly through the windows command line if you prefer it over the Git Bash command line. You can use all default options with this program as well.

<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

Open command prompt (type cmd in the search bar) and execute the command putty. If installation was successful, you should see a small window pop up like the picture below.

A screenshot of a computer

Description automatically generated

Finally, install the university VPN client. This step is necessary. You may notice that you’re able to SSH into the robot while you’re working on it on campus. That is because you’re connected to the university network. If you try to connect to the robot using the same steps from your home or a coffee shop, you’ll be unable to without the VPN client.

<https://itconnect.uw.edu/connect/uw-networks/about-husky-onnet/use-husky-onnet/terms-conditions/>

**3.0: Connecting to the Robot**

Launch BIG-IP Edge Client from your start menu and log in with your UW credentials. The VPN client will remain active in your system tray. When you finish on the university network, I recommend disconnecting via the client, otherwise all of your internet traffic will be routed through and be visible to the university. This can be done by right clicking the icon in your system tray and selecting disconnect.

**Graphical user interface, application

Description automatically generated**

Next, we will copy a file from the robot. We will use Git Bash to connect to the robot and command prompt (with Putty) to perform the copy. First, open Git Bash. Next, using the credentials from step 1, we will connect to the robot via SSH. The syntax for the command is USER@IP. You will be prompted for a password just like in a linux terminal, use the password that you used to log into the user on the robot. If your client did not connect, check your VPN.

Text

Description automatically generated

Next, you can surf the file system to find a file you would like to copy. You will need the full location from the root directory.

A screenshot of a computer

Description automatically generated with medium confidence

We will use pscp to move our file from the robot to our local pc. The syntax for pscp is **pscp From\_Directory To\_Directory.**Since we are using two different file systems and a remote connection, it’s important to recognize that the robot target syntax will be different than the windows target syntax.

Linux syntax: user@ip:filePath  
Windows syntax: filePath (using right slashes “/”)

For this example we’ll make an empty file called test.c and copy it to the desktop of our local PC. Use **touch test.c** in your Git Bash terminal.

Text

Description automatically generated

Next, open command prompt. The following example is for my system, your path **will be different**.

Text

Description automatically generated

Small files will prompt for the password and copy the file to the target location only. Larger files will show a status bar.

A screenshot of a computer screen

Description automatically generated with medium confidence

To move files from the local PC to the remote robot, simply swap the from location and the to location in the pscp command.

Using SSH in either Git Bash or command prompt will allow you to execute python code or ros commands on the remote robot. At the time of writing I am unaware of how to open multiple SSH’d terminals on windows other than opening them one at a time and using SSH. For example, you can open one terminal, SSH to the robot, and run roscore. Then open another terminal, SSH to the robot, then run ros commands (rosnode, rosrun, etc.) This can quickly become cumbersome if you need multiple windows. Furthermore, this method does not transfer graphical application information and precludes the use of programs like gedit or rvis.

The major reason to use this method is to utilize local resources (such as heavy-weight IDEs) that can’t be supported on the robot and to deploy or retrieve code remotely.