Xaxxon Project – Oculus Robot Documentation – Part 2

Version 1.2

August 11, 2017

Research Co-op for Dr. Shahram Payandeh

Written by Lestley A. Gabo

Contents

Enter password to unlock your login keyring

Since this week, June 12, 2017, when turning on the Oculus robot the first thing I see is a prompt to input my login password. This password should be "[redacted]" however that is not the issue. Without inputting this login keyring first, the WiFi won't be able to connect. Because the Wifi does not connect the Google Chrome browser doesn't open and Chrome needs to open to turn on the Xaxxon web server (turned on automatically, see Documentation Part 1: section Server auto-start on boot). The WiFi being connected and the server turning on are essential to remotely control the Oculus robot. Without the two conditions above it is impossible to remote into the robot.

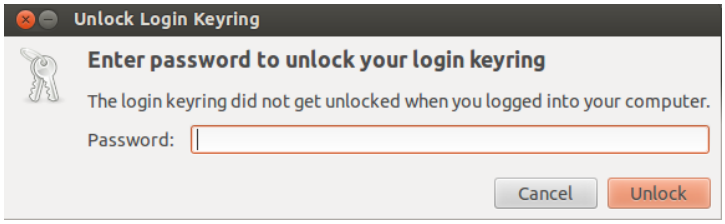

Figure 1: login keyring prompt

The login keyring prompt appears when Google Chrome is opened and Chrome is opened because we set up the server auto-start on boot.

Now to fix this issue.

Open terminal

Type in **sudo apt-get install seahorse**

Click the blue **Whisker menu**

Search for **Passwords and Keys**

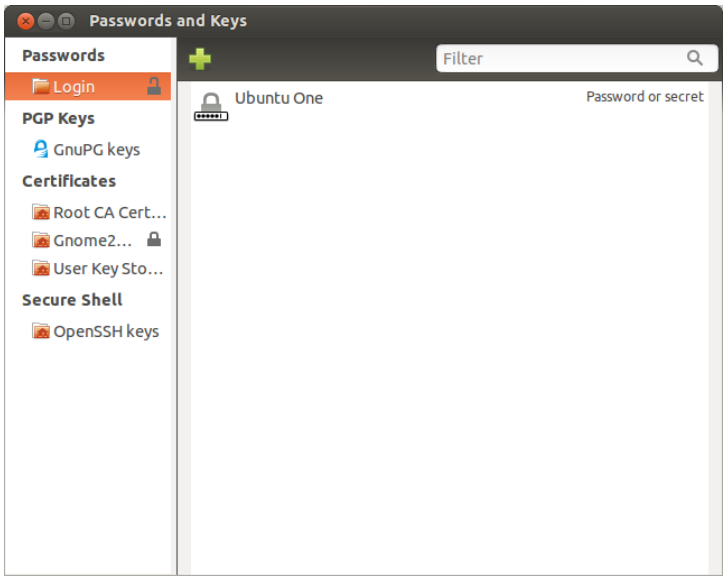

Figure 2: Password and Keys

Right-click Login in the Passwords tab and choose Change Password

Figure 3: leave the new password blank

Enter "[redacted]" without the quotation marks and **leave the new password blank** and click Continue

We leave the password blank so that when Google Chrome is opened again the prompt tries to show up again, but is automatically logged in.

Installing inside Ubuntu using command line

In this example, we are installing Eclipse into our robot.

If Eclipse is not installed in the Oculus robot, DO NOT INSTALL Eclipse because for now it is useless. These are steps that will be similar when installing other software in Ubuntu.

Unless you know how to use Eclipse and are familiar with it then go ahead and install Eclipse (if it wasn't installed already). See section **Issues with trying to edit Java files** for more information of why Eclipse is useless for now.

Now for the installation instructions.

Download the Eclipse package from here http://www.eclipse.org/downloads/ and then extract the Eclipse installer

Open the terminal and type in

tar -xvf /home/oculus/Downloads/eclipse-inst-linux64.tar.gz

Open the Eclipse installer folder and **double-click or execute eclipse-inst** and choose the right Eclipse. In our example, Eclipse for Java Developers was chosen.

To make the Eclipse Java easily accessible, we create a shortcut in the Whisker Menu.

Open the terminal and type in

cd /usr/share/applications

touch eclipse-java.desktop

Edit eclipse-java.desktop using nano. The Exec and Icon should be located to wherever directory you chose to install the Eclipse Java.

sudo nano eclipse-java.desktop

[Desktop Entry]

Name=Eclipse-Java

Comment=IDE

Exec=/home/oculus/java-mars2/eclipse/eclipse

Icon=/ home/oculus/java-mars2/eclipse /icon.xpm

Terminal=false

Type=Application

Categories=Development;

Issues with trying to edit Java files

I am not familiar with Eclipse however I installed it and hoped I could edit the Java files from the Xaxxon GitHub to play around with the Java functions.

Inside the Oculus robot, there are no .java files. There are only compiled .class files. These .class files are then run once when the Oculus robot turns on. So, I downloaded the source code .java files from Xaxxon GitHub (https://github.com/xaxxontech/oculusPrime). My idea was to download the .java files, edit them, and copy the edited version into the working directory of the Oculus robot.

The problem is I don't know how to compile the Java files. I have no idea where the Main function is in all the .java files. Therefore, I cannot compile the .java files into .class files.

Installing inside Ubuntu using Ubuntu Software Center

Open the Whisker Menu click on Ubuntu Software Center. It should be in Favorites. It has an icon with a purple square and inside it a downwards white arrow.

In this example, we are **installing Sublime Text 2**.

Inside Ubuntu Software Center on the upper right is the search bar. **Search for sublime**.

Double click on Sublime Text 2 and on the right side **click Install**.

You will be prompted to enter the **login password**, in our case it is **“[redacted]”**.

See section **Setting up panels with executables using launcher to create** to create a shortcut for sublime.

Setting up panels with executables using launcher to create shortcuts

In this example, we are adding a Sublime Text 2 shortcut assuming you followed section **Installing inside Ubuntu using Ubuntu Software Center** to install Sublime Text 2.

On the upper right of the screen, in the panel, should be the date and clock, **right-click the clock**

Hover on **Panel**

Click **Add New Items…**

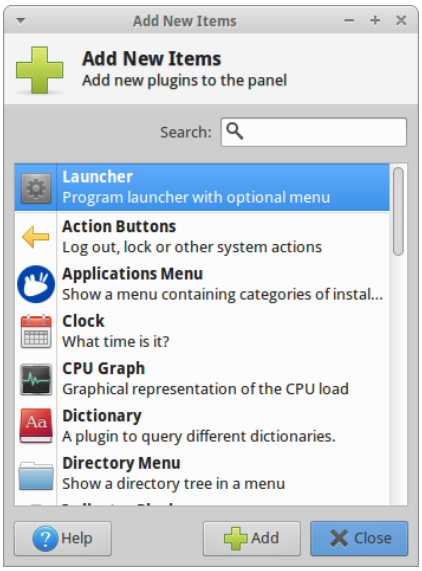

Figure 4: Add Launcher

search for Launcher, click Launcher, and click Add in the bottom then click Close

Now you have a blank launcher to the right of your clock on the panel

Right-click the blank launcher and click Properties

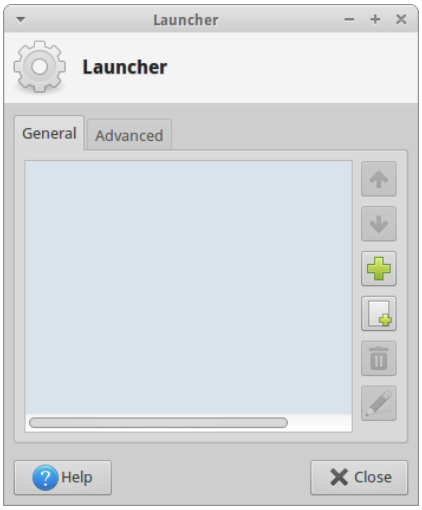

Figure 5: blank launcher

Click the **green plus sign**

Search for Sublime Text 2 and double-click it then click Close

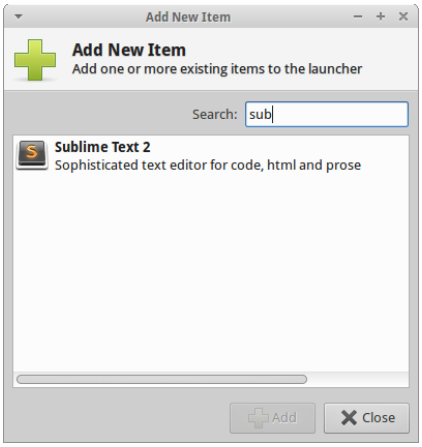

Figure 6: Adding Sublime Text 2 into the Launcher

If you cannot find your desired program, then keep following the steps below.

If you installed a program using the command line then do the following steps instead because the green plus sign above won't have your command line installed program. In this example, I downloaded Sublime Text 3, which is not found in Ubuntu Software Center.

Right-click the blank launcher and click Properties

Click on **Add a new empty item**, the white rectangle with a green plus sign

Fill in the labels

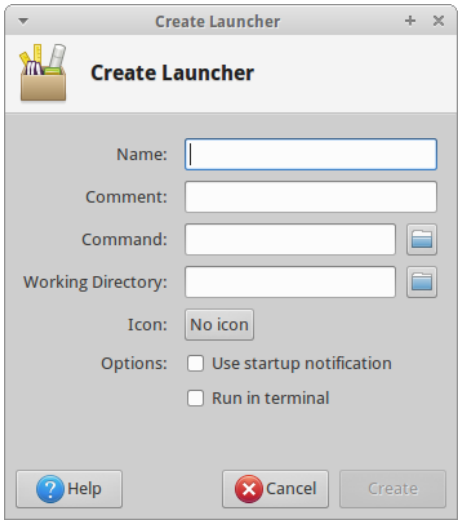

Figure 7: Manually filling in Launcher

Name: Sublime

Command: /opt/sublime_text_3/sublime_text

Icon: /opt/sublime_text_icon_3/Icon

Click Create

Now we have a Sublime Text 3 launcher in the panel. Usually, command and icon are found where the.tar.gz was extracted. In this case I moved it from the downloads folder into /opt/

Open the terminal and type in

sudo mv /home/oculus/Downloads/sublime_text_3 /opt/sublime_text_3

If asked for a password the password is **“[redacted]”**

I would recommend you open HTML, CSS, and JavaScript files in Sublime and set it as your default text editor.

Installing and setting up git

Open the Terminal Emulator and type in

sudo apt-get install git

I created an email account and a GitHub account. The usernames and passwords for the accounts are below.

Gmail account:

Username: [redacted]

Password: [redacted]

GitHub account:

Username: elderlyrobotics@gmail.com

Password: [redacted]

We then set up the SSH key.

Open the terminal and type in

ls -al ~/.ssh

This shows the existing SSH keys present and you should see something similar to

-rw-r--r-- 1 oculus oculus 751 Jun 26 13:22 id_rsa.pub

Because I already generated an SSH key and connected it to the elderlyrobotics GitHub account.

The SSH key generation instructions are here https://help.github.com/articles/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent/, so go there if the steps here for generating SSH key don't make sense.

ssh-keygen -t rsa -b 4096 –C elderlyrobotics@gmail.com

Just save it in the default file location, in our case it was in /home/oculus/ it will ask to create a passphrase. I used "[redacted]" as my passphrase.

Then you add the SSH key to the ssh-agent.

Open the terminal and type in

eval "$(ssh-agent -s)"

"Add your SSH private key to the ssh-agent. If you created your key with a different name, or if you are adding an existing key that has a different name, replace id_rsa in the command with the name of your private key file."

ssh-add ~/.ssh/id_rsa

Now we add the SSH key to our GitHub account, the following link has the step by step instructions to do this: https://help.github.com/articles/adding-a-new-ssh-key-to-your-github-account/. However, the first step of copying the SSH key can be tedious so here is my way of copying the SSH key to the clipboard.

Since we saved the generated SSH key to the default file location, it should be at /home/oculus/.ssh

Open the terminal and type in

cd .ssh

vi id_rsa.pub

Select everything and copy. To exit vi, type in ":q" or just close the terminal.

After setting up the SSH key, we configure git to connect with our GitHub account.

Open the terminal and type in

git config --global user.name elderlyrobotics

git config --global user.email elderlyrobotics@gmail.com

For convenience sake, we made a folder called push_to_git located at /home/oculus/push_to_git/xaxxonproject and we already created a GitHub repository called xaxxonproject (https://github.com/elderlyrobotics/xaxxonproject).

If you want to create a new repository, create a new repository in GitHub with a relevant name then as instructed on the page right after you create the new repository.

Open the terminal and type in

mkdir /home/oculus/push_to_git/"repository name"

cd /home/oculus/push_to_git/"repository name"

echo "# "repository name"" >> README.md

git initgit add README.md

git commit -m "first commit"

git remote add origin git@github.com:elderlyrobotics/"repository name".git

git push -u origin master

Because we named our repository in GitHub "xaxxonproject", for simplicity we also named our directory "xaxxonproject" and it is located at /home/oculus/push_to_git/xaxxonproject/

To use git or to push anything you must follow these steps:

Open the terminal and type in

cd /home/oculus/push_to_git/xaxxonproject/

git status

git add .

git commit -m "comment of change"

git push origin master

Note: cd into the directory where you did your initial commit. The example above is because the directory of my initial commit is inside /home/oculus/push_to_git/xaxxonproject. The initial commit has the "-u" on the git push but for future pushes you can omit the "-u". Also, when in a branch you push by using " git push origin 'branch name' ".