WPI

RBE 3001 – UNIFIED ROBOTICS III

LAB 0 – GET YOUR PC READY FOR LABS

1. INTRODUCTION:

Lab 0 marks the beginning, helping you set up your computers and pave the way for upcoming labs. To ensure a seamless experience in the upcoming labs, here's what you'll need:

1. **Operating System**: Linux --> Ubuntu 20.04
2. **Software**: MATLAB
3. **Version Control**: Git and GitHub
4. **Firmware**: Dynamixel SDK

2. OBJECTIVES:

By the end of this lab, you'll have Ubuntu 20.04 up and running on your computer. You'll also have MATLAB installed and activated through the WPI license. Plus, you'll gain the know-how to set up the Dynamixel SDK for OpenmanipulatorX arm control. And of course, you'll become skilled at installing and navigating Git and GitHub in the Linux environment.

3. IMPORTANT NOTE:

In RBE 3001 this year, we've employed new robotic arms located in the AK Labs, where you'll conduct your labs. Each team, consisting of three students, will have access to one robotic arm and a dedicated lab computer. Your lab activities fall into two categories:

1. **Analysis/Coding:** For this aspect, you're free to use your personal laptop inside or outside the lab. No physical presence in the lab is required.
2. **Implementing on the Robot:** To put your analyses and codes into action on the robot, you must be physically present in the lab because the robots are confined to the lab space and cannot be taken out. *It's crucial to connect the robot to the lab computer (not your personal laptop) for implementation.* The lab computers are already running Ubuntu 20.04 with MATLAB installed

and activated -- if you encounter any lab computer issues, please inform the LAs or SAs. However, since you'll also be working in the lab outside your regular lab hours when the lab computers might not be available to you, and since you will be working on the labs outside the lab environment for Analysis/Coding purposes, you need to ensure your personal computer is also prepared:

4. LINUX (UBUNTU 20.04):

Based on whether you're using a Windows or Mac PC, please follow the relevant instructions below: You are also encouraged to attend SA's office hours (see the google calendar in Canvas) to get live help with Linux installation on your PC.

4.1. Windows 10:

Here is an easy guide to follow for dual boot:
https://linuxconfig.org/how-to-install-ubuntu-20-04-alongside-windows-10-dual-boot

4.2. MacOS:

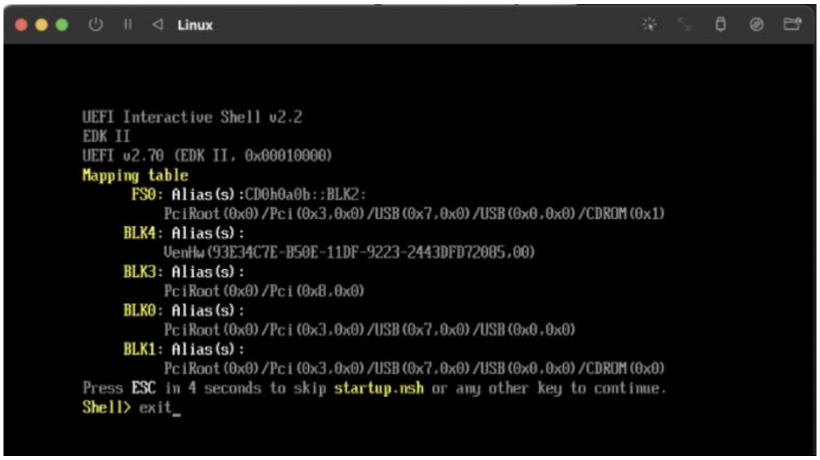
To Mac users: the lab activities heavily rely on Unix. If you're using macOS, you can stick with it and directly install MATLAB on macOS instead of Ubuntu. If this applies to you, you can skip this section of the assignment and proceed to Section 5. **However, it's important to keep two things in mind:**

1. **Future Linux Requirement:** While macOS may work for now, be aware that you'll eventually need a Linux system for RBE 3002. ***We strongly encourage you to use Linux*** to familiarize yourself with Linux during this course to make the transition smoother.
2. **Lab Computer Usage:** Regardless of your operating system, remember that **you'll still need to use the lab computers to connect with the robot** (refer to Section 3).

Mac users with **Intel Chip** can use the following resources: https://www.makeuseof.com/tag/install-linux-macbook-pro/

Mac users with **M chip** can use UTM virtual machine to install and use Linux. There are quite some useful YouTube videos on how to do this. Here is an example: How to install Ubuntu on M1 Mac using UTM

You will find this link helpful as well: Ubuntu ISO - Use LTS. Good practice is to set aside a minimum of 50GB in your machine when you are using Virtual Machine. While following the steps in the YouTube video, if you are stuck in the window shown below, it means that you have used the wrong .iso image file. In that case, use this file instead, and you will be fine.



```
UEFI Interactive Shell v2.2
EDK II
UEFI v2.70 (EDK II, 0x00010000)
Mapping table
      FS0: Alias(s):CD0h0a0b::BLK2:
          PciRoot(0x0)/Pci(0x3,0x0)/USB(0x7,0x0)/USB(0x0,0x0)/CDROM(0x1)
     BLK4: Alias(s):
          VenHw(93E34C7E-B50E-11DF-9223-2443DFD72085,00)
     BLK3: Alias(s):
          PciRoot(0x0)/Pci(0x8,0x0)
     BLK0: Alias(s):
          PciRoot(0x0)/Pci(0x3,0x0)/USB(0x7,0x0)/USB(0x0,0x0)
     BLK1: Alias(s):
          PciRoot(0x0)/Pci(0x3,0x0)/USB(0x7,0x0)/USB(0x0,0x0)/CDROM(0x0)
Press ESC in 4 seconds to skip startup.nsh or any other key to continue.
Shell> exit_
```

*** Once you've reached the bash form of Linux, you'll need to install the Ubuntu GUI from the terminal screen. You can follow along with the video (the commands are shown below as well):*

```
$ sudo apt install tasksel
$ sudo tasksel install ubuntu-desktop
$ sudo reboot
```

Once you finish installing the Ubuntu GUI you'll need to navigate to the terminal (press command then search for terminal) and install Firefox to access the internet. Use the command below:

```
$ sudo apt install firefox
```

Note: In our lab context, it's advisable to avoid using Virtual Machines (VMs) whenever possible. VMs should be employed only if you encounter difficulties setting up a dual boot Linux on your Mac. **If you use VM, you will NOT be able to connect to the robot.** You might find YouTube tutorials outlining how to dual boot Linux with M chips on Mac, but you need to try and assess their compatibility on your computer. Here is an example:

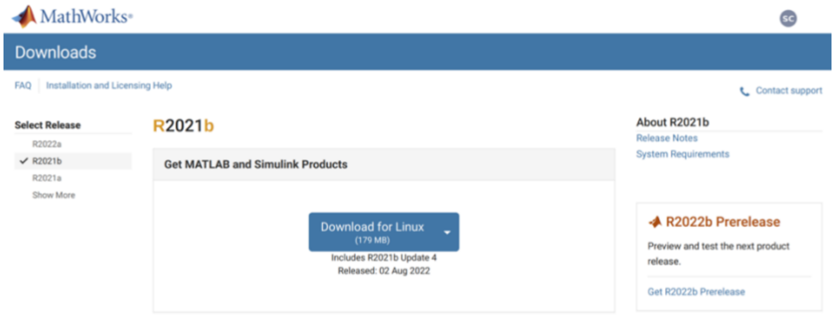
https://www.youtube.com/watch?v=aXh_d1OixRI

5. MATLAB:

Once you have Ubuntu 20.04 installed, the next step is to install MATLAB. You can download, install, and activate Matlab using WPI license. Please follow the steps below to get started.
Steps to Install Matlab under Linux on your personal system:

1. Navigate to the url - https://www.mathworks.com/mwaccount/register
2. Create a mathworks account using your WPI email in the above page
3. Follow the instructions in that page to complete registration
4. Navigate to the following url to get the WPI license - https://www.mathworks.com/academia/tah-portal/worcester-polytechnic-institute-40552010.html
5. Click on Get Started and login using your WPI MathWorks account credentials

6. You will be navigated to the following page - https://www.mathworks.com/downloads/



R2021b

Get MATLAB and Simulink Products

Includes R2021b Update 4
Released: 02 Aug 2022

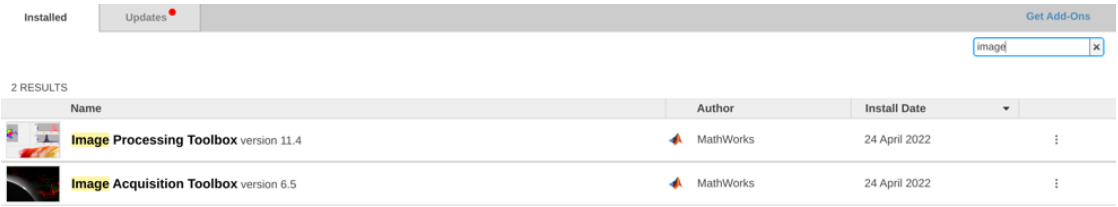
About R2021b
Release Notes
System Requirements

R2022b Prerelease
Preview and test the next product release.
Get R2022b Prerelease

7. Select R2021b and click on Download for Linux button (*R2021b is not the latest version, but it is still the best version to continue with our projects*)
8. Move the matlab zip file from Downloads to Home folder
9. Run the following commands in the terminal:
   a. **$ sudo unzip -X -K matlab_R2021b_glnxa64.zip -d matlab**
   b. **$ cd matlab**
   c. **$ xhost +SI:localuser:root**
   d. **$ sudo ./install**

10. In the Matlab installer, sign in and make sure the license is selected
11. Keep clicking on Next until select products page
12. The following products are used in our course – select them:
    a. Matlab
    b. Image Processing Toolbox
    c. Image Acquisition Toolbox
    d. Symbolic math Toolbox

13. Click next until the products are installed. This may take some time
14. Go to your Terminal and type matlab, and matlab will open for the first time
15. Under the home tab on the rightmost corner, you will see a button named Add-Ons
16. Click on the button, then click on Manage Add-Ons
17. Make sure the following add-ons are there:
    a. Image Processing Toolbox
    b. Image Acquisition Toolbox



2 RESULTS

Name	Author	Install Date
Image Processing Toolbox version 11.4	MathWorks	24 April 2022
Image Acquisition Toolbox version 6.5	MathWorks	24 April 2022

18. Learn more from matlab tutorials:
    With the Ubuntu 20.04 LTS and MATLAB installed, you can follow the tutorials on MATLAB documentations to get familiar with the MATLAB environment and commands: https://www.mathworks.com/help/matlab/getting-started-with-matlab.html.

19. **Assignment:** Complete the MATLAB Onramp Course and **submit the certificate through Canvas**. **This assignment is due before the start of your first lab.**

6. GIT AND GITHUB:

Git and GitHub are used for version control and collaboration on software projects. Git enables developers to track changes to code, making it easy to understand what was changed, when it was changed, and who changed it. This history can be invaluable for debugging and understanding the evolution of a project. It also helps different developers working on different parts of the project without impacting each other's work. This section will help you download and install git and navigate between git tool and github.com webpage.

6.1. Install Git:

Type the following command in your terminal to install git tool in your linux machine.

```
$ cd
```

The above command will take you to your home directory.

```
$ sudo apt install git-all
```

The above command will install the git tool in your local machine

6.2. Your Identity:

Type the following commands in your terminal

```
$ git config --global user.name "John Doe"
$ git config --global user.email jdoe@wpi.edu
```

6.3. Setting Up the SSH Key for Your Local Machine:

SSH (Secure Shell) keys in Git are used to provide a secure method of authentication when connecting to a remote repository that is repository saved in your **github.com** (more on github.com later) account.

Follow the steps below to generate and add a new SSH key

1. Type the following command in terminal

   ```
   $ ssh-keygen -t ed25519 -C "your_email@example.com"
   ```

   This will create a new SSH key using the provided email as a label
2. When you're prompted to "Enter a file in which to save the key," press Enter. This accepts the default file location. Just press enter
3. At the prompt, type a secure passphrase. The following prompt will show, press enter in both prompts for Empty pass key

   ```
   > Enter passphrase (empty for no passphrase): [Type a passphrase]
   ```

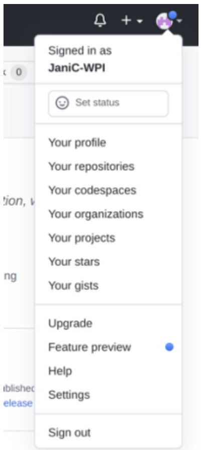
> Enter same passphrase again: [Type a passphrase]

4. To add the SSH key to the ssh agent type the following command in terminal

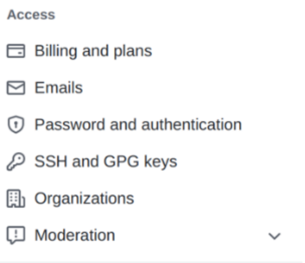
```
$ eval "$(ssh-agent -s)"
$ ssh-add ~/.ssh/id_ed25519
```

   Note: For further details to generate ssh key please see page https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent

5. Type **$ cat ~/.ssh/id_ed25519.pub** in your terminal. Copy the key that is generated.
6. Open Github and sign in. Go to settings from the upper right corner



7. In the "Access" section of the sidebar, click **SSH and GPG keys**.



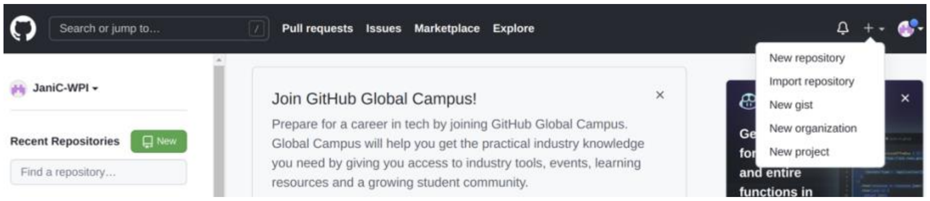
8. Click New SSH key or Add SSH key
9. Add a proper title in the Title section and paste the copied key in the Key section.
10. Click on Add SSH key.

Note – The following section can only be done if the above is completed.

6.4. Setting Up a Repository from Your Local Machine:

If you have a project that you need to put on Github.com, we need to follow these steps

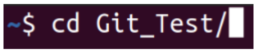
1. Sign in to github.com
2. Click on the new repository button from the top right corner as shown below



Join GitHub Global Campus!

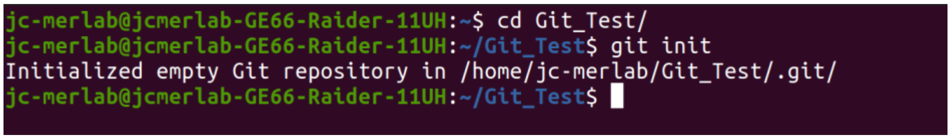
Prepare for a career in tech by joining GitHub Global Campus. Global Campus will help you get the practical industry knowledge you need by giving you access to industry tools, events, learning resources and a growing student community.

3. Name your repository, select your access preference (Public or Private)
4. Do not select Add Readme and click on Create Repository
5. Navigate to the project folder from your terminal. Here Git_Test in the following command is a test folder created in the instructor's local machine for the following instruction steps. Replace it with your own project folder name.



```
~$ cd Git_Test/
```

6. Type **git init** and you will get the following message



```
jc-merlab@jcmerlab-GE66-Raider-11UH:~$ cd Git_Test/
jc-merlab@jcmerlab-GE66-Raider-11UH:~/Git_Test$ git init
Initialized empty Git repository in /home/jc-merlab/Git_Test/.git/
jc-merlab@jcmerlab-GE66-Raider-11UH:~/Git_Test$
```

7. In your local project folder create a new file or change an existing file
8. In the terminal type the following command

   $ git add <filename>.<ext>

9. In case you are making changes to multiple file type

   $ git add --all

10. To commit your changes, you need to type the command -

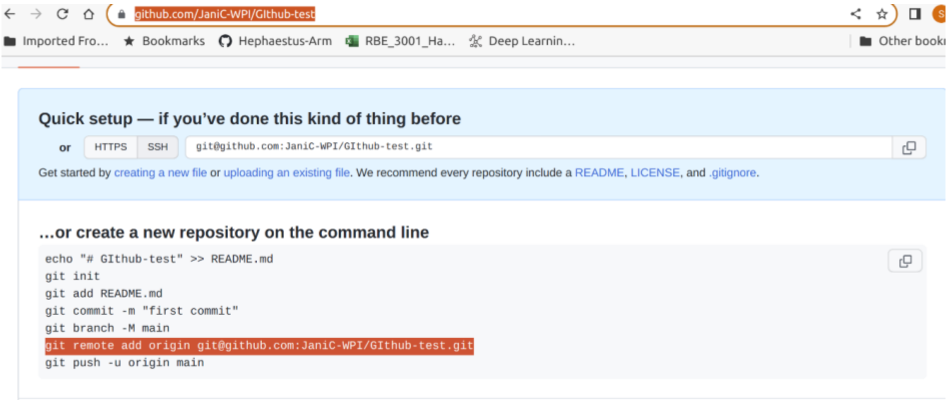
    $ git commit -m "first commit"

    In the above two commands, **git add** stages the changes to commit, **-m** is the commit message. You can change the statement *"first commit"* on the basis of the type of change you are making.

11. Type command

    $ git branch -M main

12. Now in the GitHub webpage once you have clicked the create repository button you will have the following page opened:



Quick setup — if you've done this kind of thing before

or git@github.com:JaniC-WPI/GIthub-test.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

```
echo "# GIthub-test" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin git@github.com:JaniC-WPI/GIthub-test.git
git push -u origin main
```

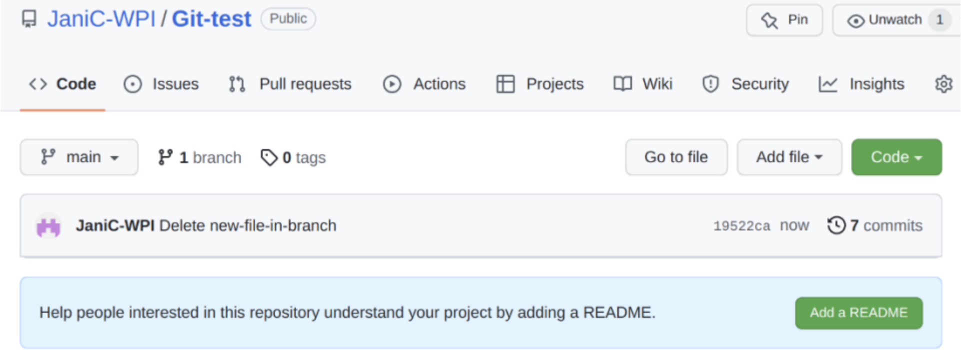
13. Copy the Quick set up URL for the repository starting with **git@github.**
14. Back in your terminal type the following command

```
$ git remote add origin <your repo url>
```

15.

```
$ git push -u origin main
```

16. A repository is created in github.com from your local machine.



JaniC-WPI Delete new-file-in-branch 19522ca now 7 commits

Help people interested in this repository understand your project by adding a README.

17. If you create a new file in the github.com page, or any changes in the GitHub made by some other user's local machine, you pull them in your local machine by typing the following command.

```
$ git pull origin main
```

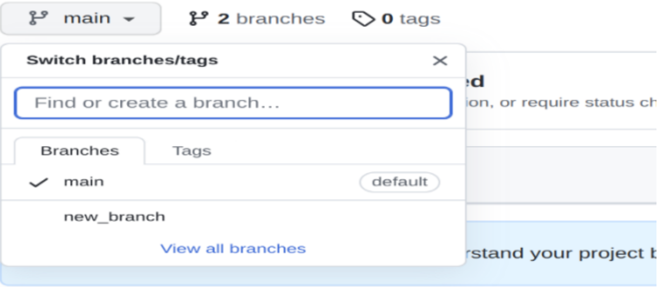
18. If you want to create a branch other than 'main' type the following

```
$ git branch <branch name>
```

The above command will only create a branch. Any changes you make in GitHub will still be made in the main branch. To switch to make a commit in the new branch, you need to type the following commands

```
$ git checkout <branch name>
$ git commit --all
$ git push origin <branch name>
```

Now the branch will be switched to the new branch and will be reflected in the GitHub page (see the below screenshot where new branch under Branches tab is created). Now you can commit and push changes in this new branch:



19. Once you have made necessary changes in your branch if you want to merge the changes with the main branch, type the following command

```
$ git checkout main
$ git merge <branch name>
```

20. To delete the branch, switch to main branch using the checkout command and type the following.

    ```
    $ git branch -delete <branch name>
    ```

21. All the above steps are valid when you are cloning an existing repository in GitHub.
22. To clone you use

    ```
    $ git clone <repo url>
    ```

23. Check the list of commits in all the branches in a repository by:

    ```
    $ git log –all
    ```

Most of you are probably new to Git versioning control and GitHub, so it will be of great advantage if you follow this Hello World exercise to get started with GitHub: https://docs.github.com/en/get-started/quickstart/hello-world and use the interactive tool below to learn the use of Git via terminal: https://learngitbranching.js.org

** The cheat sheet in the link below features the most important and commonly used Git commands for easy reference: https://education.github.com/git-cheat-sheet-education.pdf*

7. INSTALL DYNAMIXEL SDK:

Important: This section is designed for establishing communication with the robotic arm. As outlined in Section 3, it's crucial to utilize the lab computer for these tasks. *Curious about using your personal laptop when working with the robot in the lab? You can give it a shot but be ready for any troubleshooting. While even if it may be possible to connect to the robot using your laptop, we can't promise a seamless experience. Proceed at your discretion, as any potential issues/ troubleshooting will be your responsibility. However, if you are using a 'native' Ubuntu 20.04 Operating system on your personal computer, all the below instructions should be working properly. Remember, VM will not work!*
The following instructions are for installing dynamixel in Ubuntu 20.04 Operating System, whether it is your personal computer or the lab's computer.

First, create a folder called 'rbe3001' in your home directory to store your files for this course. This can be done by writing the following commands in the terminal in Ubuntu:

```
$ mkdir  ~/rbe3001 [This makes a directory (folder) named rbe3001.]
```

Note: "by using "~/" you are asking the computer to create this folder in your "home directory" or "root directory". You do not have to use "~/" as you may want to create the rbe3001 folder in another directory/folder different from the home directory. In that case, use "**$ cd** [foldername]" to navigate to the folder of your choice and then type "**$ mkdir** rbe3001" to create the folder in that directory.

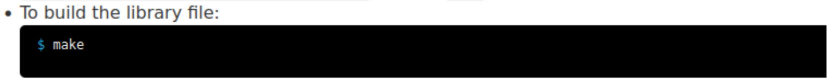
Then, follow the Dynamixel SDK installation instructions to install the Dynamixel SDK on your Ubuntu. Not all steps are required, so please see the bullet-point notes below.

[https://emanual.robotis.com/docs/en/software/dynamixel/dynamixel_sdk/download/ - repository](https://emanual.robotis.com/docs/en/software/dynamixel/dynamixel_sdk/download/#repository)

Important Installation Notes:

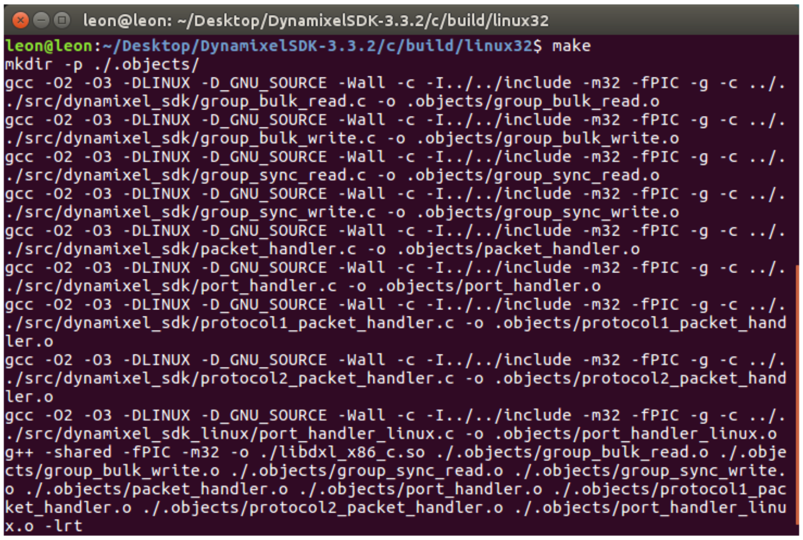
- Start at step "2. Download". Use option 1 under "2.1. Repository". Clone the repository into the 'rbe3001' folder made above:
    - **$ cd** ~/rbe3001 *[this navigates to the folder you just made above]*
    - **$ git clone** https://github.com/ROBOTIS-GIT/DynamixelSDK.git *[This downloads the repository/folder you need from the source on the internet/GitHub.]*
    - As a result, the path to the Dynamixel SDK referenced in future steps ([DynamixelSDK folder]) is ~/rbe3001/DynamixelSDK
- Skip step 3 and go to step 4, "4. Library Setup".
- In step 4, navigate first to 4.2.1.4 and build the library (**do not continue to 4.2.2**),

  IMPORTANT: In this step, where you are asked to run the following command



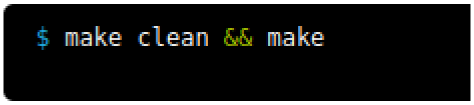
- To build the library file:

```
$ make
```



```
leon@leon: ~/Desktop/DynamixelSDK-3.3.2/c/build/linux32
leon@leon:~/Desktop/DynamixelSDK-3.3.2/c/build/linux32$ make
mkdir -p ./.objects/
gcc -O2 -O3 -DLINUX -D_GNU_SOURCE -Wall -c -I../../include -m32 -fPIC -g -c ../.
./src/dynamixel_sdk/group_bulk_read.c -o .objects/group_bulk_read.o
gcc -O2 -O3 -DLINUX -D_GNU_SOURCE -Wall -c -I../../include -m32 -fPIC -g -c ../.
./src/dynamixel_sdk/group_bulk_write.c -o .objects/group_bulk_write.o
gcc -O2 -O3 -DLINUX -D_GNU_SOURCE -Wall -c -I../../include -m32 -fPIC -g -c ../.
./src/dynamixel_sdk/group_sync_read.c -o .objects/group_sync_read.o
gcc -O2 -O3 -DLINUX -D_GNU_SOURCE -Wall -c -I../../include -m32 -fPIC -g -c ../.
./src/dynamixel_sdk/group_sync_write.c -o .objects/group_sync_write.o
gcc -O2 -O3 -DLINUX -D_GNU_SOURCE -Wall -c -I../../include -m32 -fPIC -g -c ../.
./src/dynamixel_sdk/packet_handler.c -o .objects/packet_handler.o
gcc -O2 -O3 -DLINUX -D_GNU_SOURCE -Wall -c -I../../include -m32 -fPIC -g -c ../.
./src/dynamixel_sdk/port_handler.c -o .objects/port_handler.o
gcc -O2 -O3 -DLINUX -D_GNU_SOURCE -Wall -c -I../../include -m32 -fPIC -g -c ../.
./src/dynamixel_sdk/protocol1_packet_handler.c -o .objects/protocol1_packet_hand
ler.o
gcc -O2 -O3 -DLINUX -D_GNU_SOURCE -Wall -c -I../../include -m32 -fPIC -g -c ../.
./src/dynamixel_sdk/protocol2_packet_handler.c -o .objects/protocol2_packet_hand
ler.o
gcc -O2 -O3 -DLINUX -D_GNU_SOURCE -Wall -c -I../../include -m32 -fPIC -g -c ../.
./src/dynamixel_sdk_linux/port_handler_linux.c -o .objects/port_handler_linux.o
g++ -shared -fPIC -m32 -o ./libdxl_x86_c.so ./.objects/group_bulk_read.o ./.obje
cts/group_bulk_write.o ./.objects/group_sync_read.o ./.objects/group_sync_write.
o ./.objects/packet_handler.o ./.objects/port_handler.o ./.objects/protocol1_pac
ket_handler.o ./.objects/protocol2_packet_handler.o ./.objects/port_handler_linu
x.o -lrt
```

Instead of running that run the following even if there is no error.



```
$ make clean && make
```

- Then proceed to the 4.15. MATLAB Linux section and follow the instructions, keeping in mind the notes for this step:
    - **IMPORTANT**: There are other sections on the same page to download the SDK for MATLAB use on other Operating Systems. Use of the Dynamixel SDK and other tools for this course have not been tested for Windows or macOS. Since you are using the lab computer which natively runs Linux-Ubuntu 20.04, you should be all set, so simply ignore those sections related to Windows and macOS.
    - Skip the matlab installation steps if you already have matlab installed (you should at this point).
    - Follow the steps in "4.15.3 Building and Running the Sample Code" **until you reach "4.15.3.3. Run example". Make the changes to the MATLAB path but do not worry**

    about running the example. We will test if everything works in Lab 1, using the lab1_base.m script provided with the starter code.

    - **Before you start testing anything please make sure the USB cable is connected to your computer and the power cable is connected to the power outlet and the power switch on the electronics board is 'ON'**

- After completing the parts listed above in the relevant section for MATLAB on your OS in step 4, run the following command (filling in the path to your Dynamixel SDK folder) in a new terminal window/tab to build the C Dynamixel SDK that the MATLAB SDK depends on:

    -
      ```
      cd ~/rbe3001/DynamixelSDK/c/build/linux64 && make clean && make
      ```

*If you receive any errors or warnings, ask an LA or SA for assistance.