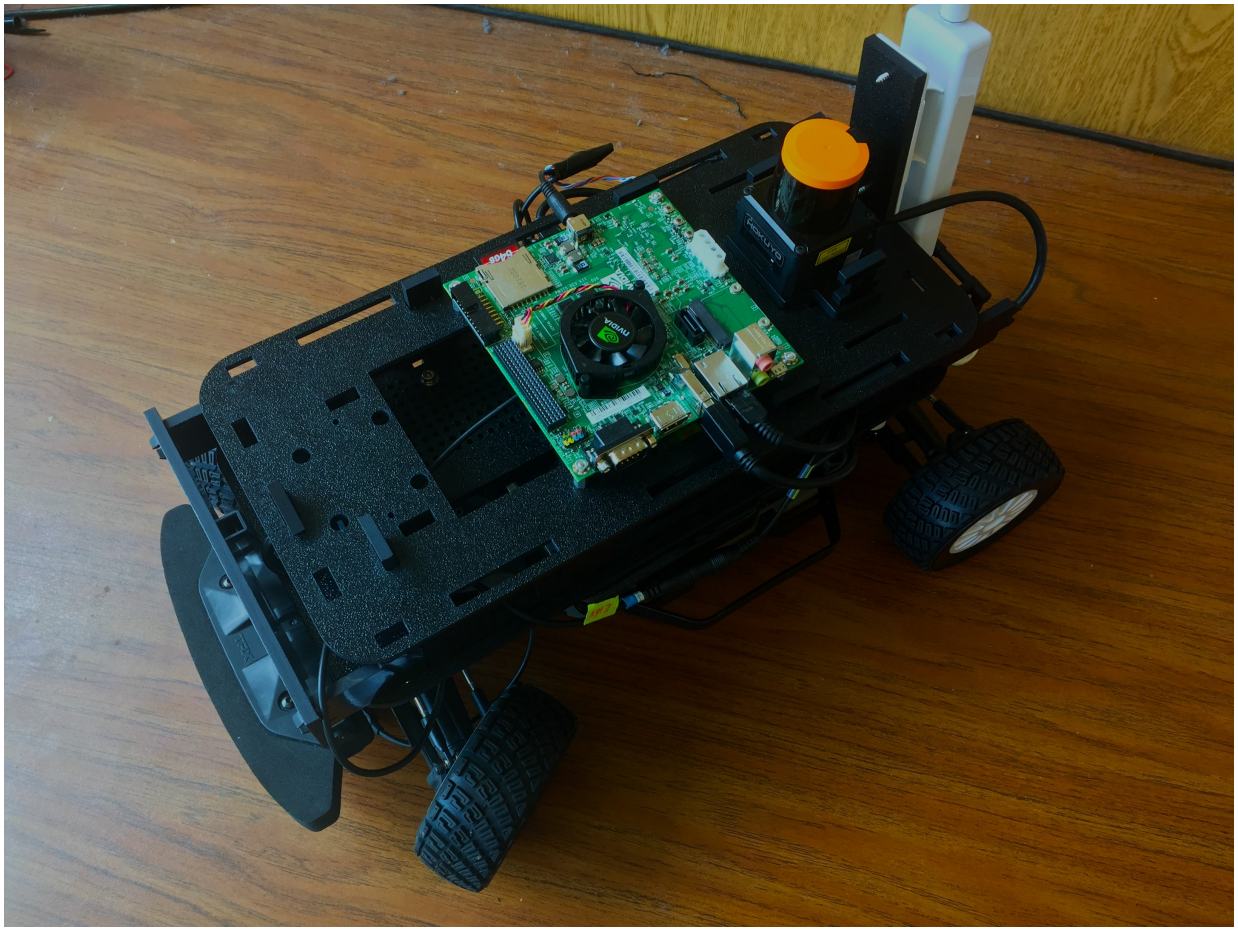
F1/10 Car

Instructions



Temple University

Summer 2017 Edition

Introduction

This document is a step by step instruction set to guide you through different stages of running your own F1/10 car. It starts with setting software up on your personal computer, and ends with the precise instruction on remotely controlling the F1/10 car. Different chapters will cover their focus. They may look little unrelated to each other, to some extent, but it is necessary that you know and understand the material before proceed to next chapter. There will be some pre-requirement for some chapters, so do not miss them. There also is a section of Q&A at the very end of this document, if you are running into some common questions, check it out and you may find some solution there. Keep on track and good luck!

Contributors

Dr. Li Bai
Kuang Jiang

Chapter One

Here is Ubuntu

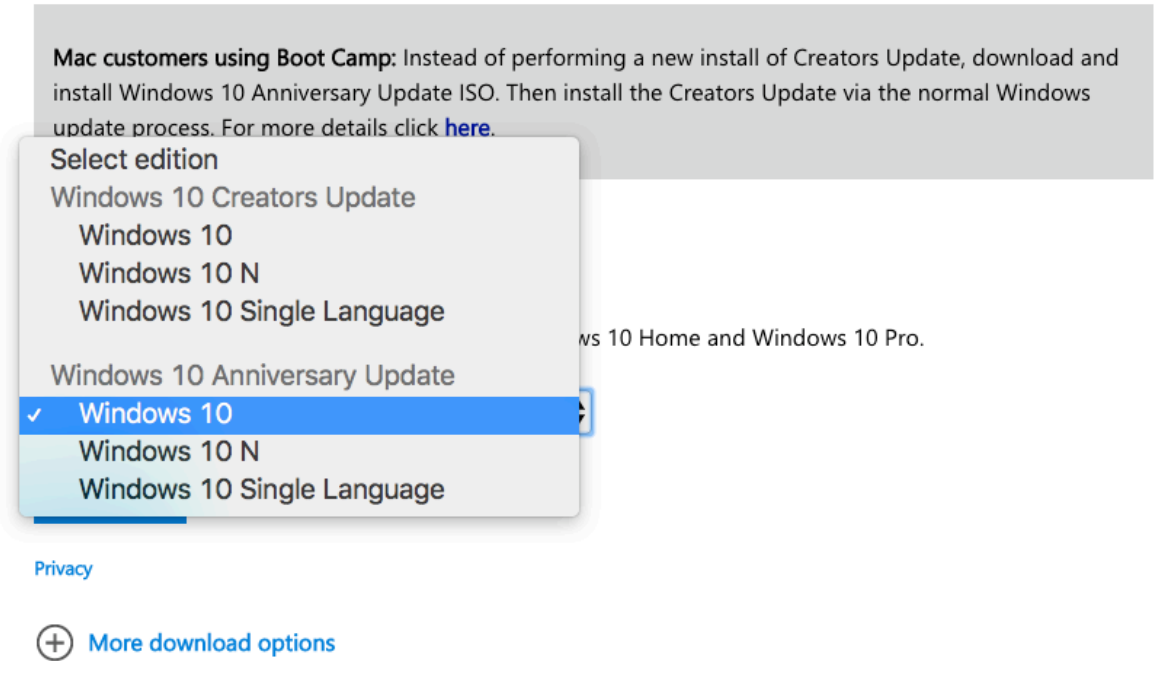
Ubuntu Installation

In this chapter, we will introduce the setup of software, including setting up Windows 10 on MacBook and setting up Ubuntu on Windows 10. In this instruction, we will use a MacBook Pro running macOS 10.12.5 as the default computer. If you are using a Windows PC, you can skip the first part of the instruction and jump into the Setting up Ubuntu on Windows 10 part. You can also try to googling different instructions on setting up ubuntu on your personal computer.

Setting up Windows on Mac

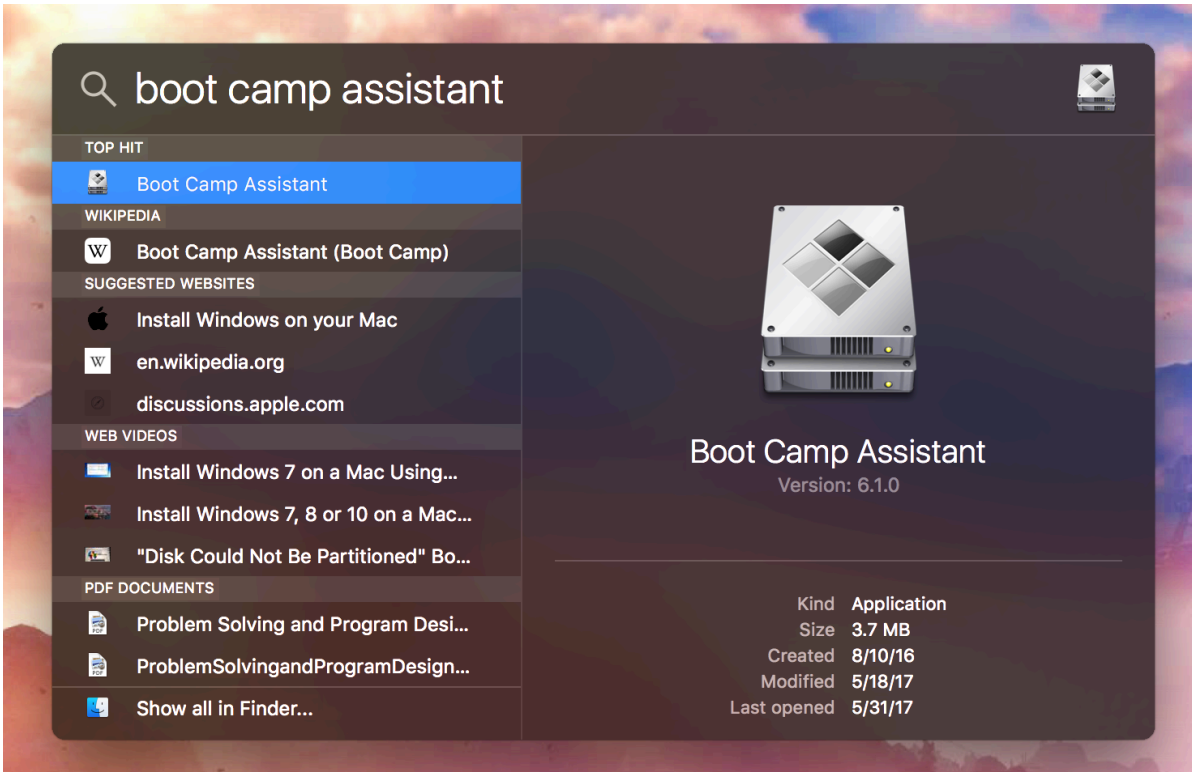
In this part, we will give step by step instruction on setting up Windows 10 on a MacBook. Before you try any step, make sure that the system is backed up.

1. Make sure that you have sufficient storage space for your Mac.
2. Back up MacBook via Time Machine or else.
3. Download Windows 10 iso file from Microsoft's official website. The iso file should be about 4 GB. In the website, make sure do NOT select Windows 10 under creators but select the other one.
   Here is the link: https://www.microsoft.com/en-us/software-download/windows10ISO



4. After download the iso file, make sure MacBook has sufficient storage to install Windows. Go to the apple logo on the top left corner of the screen, click and go to "About this Mac" > "Storage". You will need at least 70GB free space to make sure all three system will function properly without storage problems.
   Generally, Windows 10 takes about 30 GB for system file, so we recommend 50GB for Windows 10; Ubuntu takes about 5GB for system file, so we recommend 10 GB for ubuntu. The installation takes about 8GB of your system storage, which will not be accessible for either of the system. Please CHECK the further requirement here: https://support.apple.com/en-us/HT201468

5. Open Boot Camp Assistant.
   It can be found under Dashboard > Utilities or search it from Spotlight Search or ask Siri to find it for you.



6. Click Continue.

7. Check "Create a Windows 7 or later version install disk". Then Continue.

8. Select the iso file just downloaded from Microsoft, then Continue.

9. In “Create a Partition for Windows”, the partition of Windows means the total size of Windows 10 and Ubuntu. So as recommended, slide the bar and make sure it is 60GB for Windows. Then Install.
   During the process, Mac will automatically download the support file and drives. It will restart when it is done.

10. Once Mac reboot into the installation process, select your desired language, number formats, and keyboard. Then click on Next.

11. If you have a active key, active Windows. Otherwise click on Skip.

12. Once prompt “Where do you want to install Windows”, select the drive that has 60GB, usually it will show up as “Drive 0 partition X: BOOTCAMP” and format the drive. After formation, click on Next.

13. Windows now will start the installation.

14. Once Mac reboot into Windows, if not prompted Boot Camp Installer, go to My Computer > OSXRESERVED > BootCamp. Select and run Setup.exe to install BootCamp on Windows. Bootcamp ensure compatibility with Windows 10’s software and Mac’s hardware.

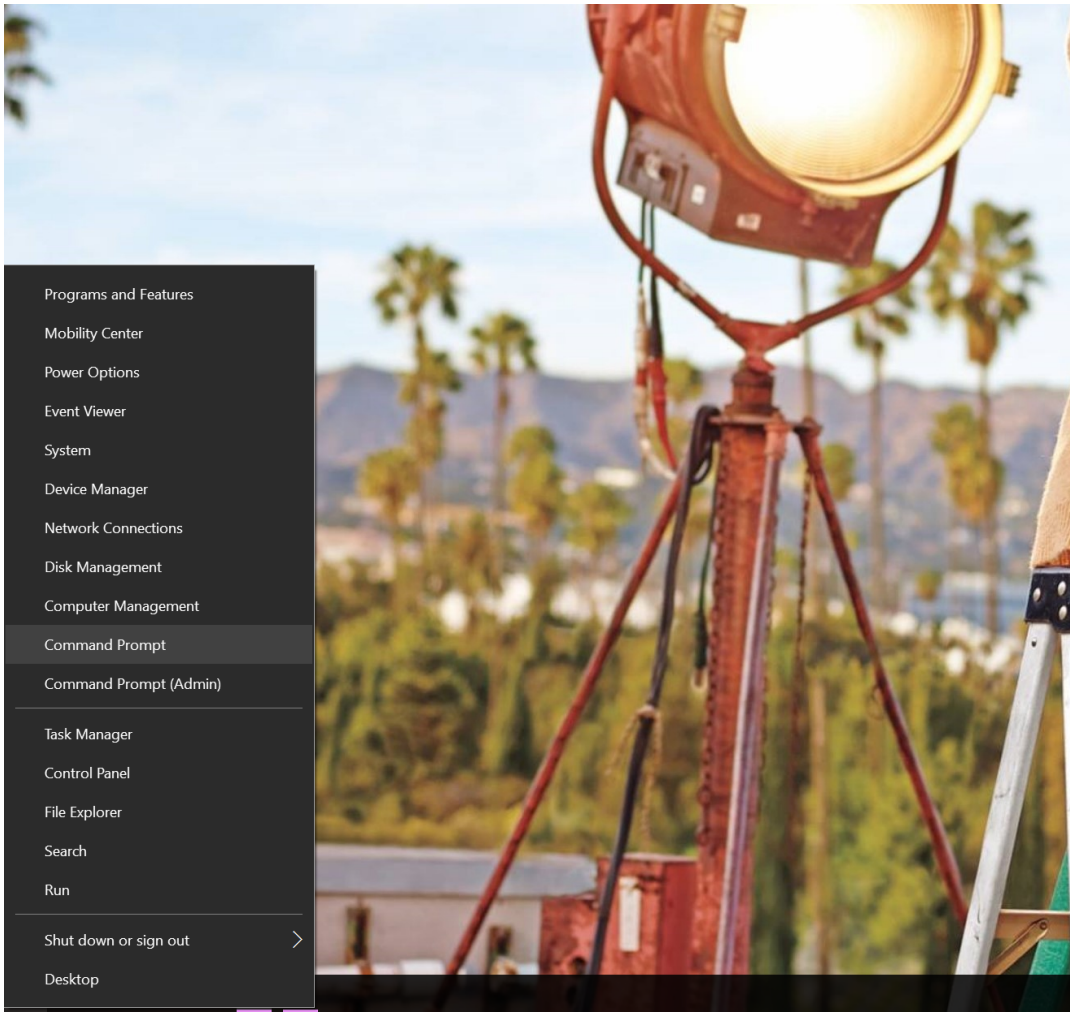
15. Now Windows should be running correctly on Mac. You can switch back and forth from macOS and Windows by holding on ALT button upon boot. You can also right click on the bootcamp icon in Windows’ task bar and select “Restart in OS X” to reboot into macOS.

Setting up Ubuntu on Windows

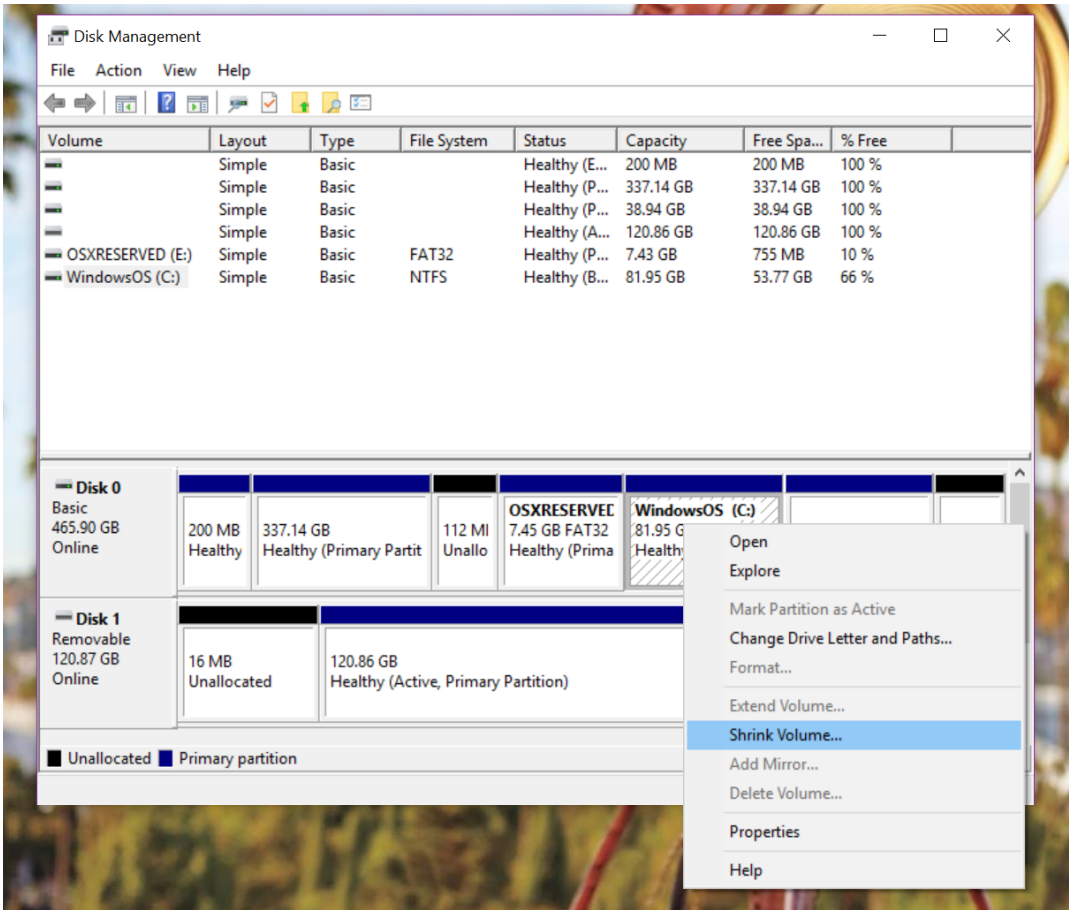
In this section, we will walk through the installation of Ubuntu on a Windows. If you are a Mac owner, make sure you have Windows installed and stay in Windows for this part of the instruction.

1. Back up Windows 10 into an external storage unit.

2. In Windows 10, right click on the Windows logo in the task bar and select “Disk Management”.

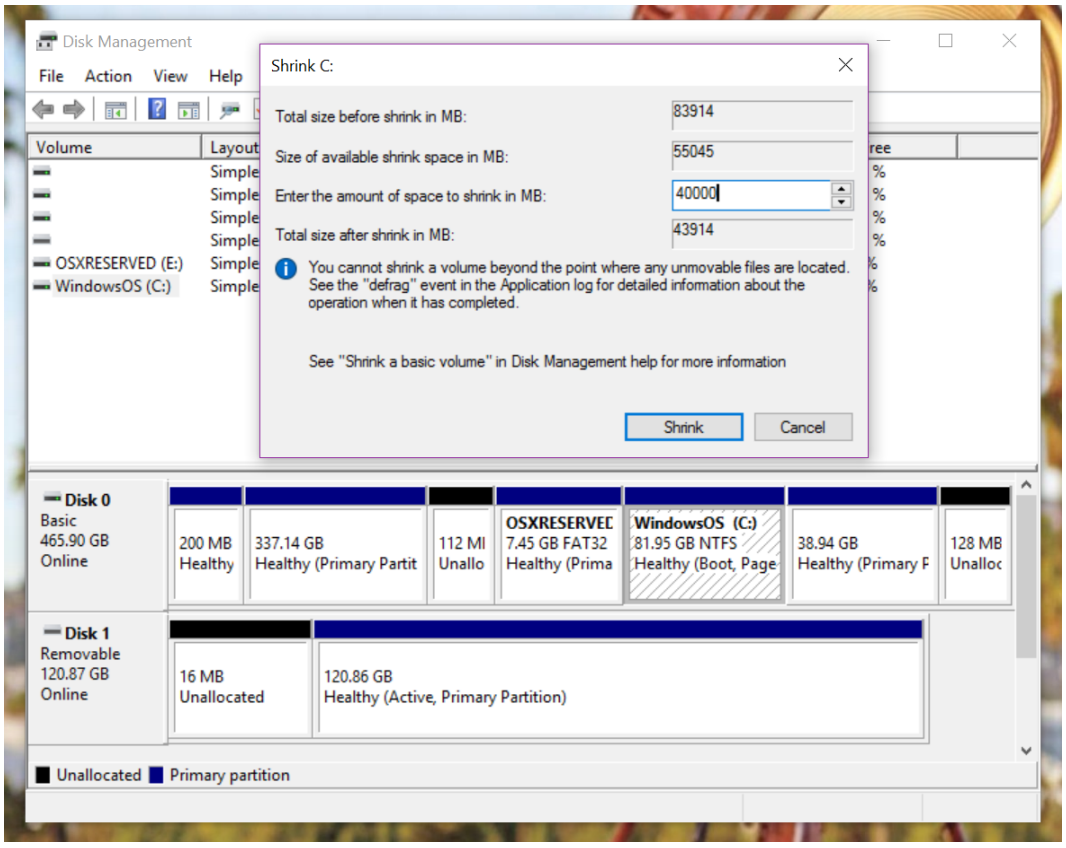


3. In Disk Management, select WindowsOS Volume, (C:), and right click then select Shrink.



4. In the Shrink Menu, enter the amount of space to shrink.
   NOTICE: The entered amount will be your Ubuntu volume and the rest will stay in

Windows. Use the equation 1GB = 1000MB to determine the desired size.



5. After finish shrinking the disk, format the new volume into NTFS format by right click on the corresponding volume in Disk Management and select format.
6. Go to Ubuntu's official site and download desired firmware for ubuntu. Usually, we use the latest desktop LTS version unless suggested otherwise. Here is the link: https://www.ubuntu.com/download/desktop
7. Plug in a USB drive, at least 2GB, and create a bootable USB from Windows 10. MAC: Upon reboot into the bootable USB, hold ALT button to choose the USB. Here is the instruction on creating a bootable USB on Windows 10: https://tutorials.ubuntu.com/tutorial/tutorial-create-a-usb-stick-on-windows#0
8. Once it starts up, select "Try Ubuntu" using arrow keys.
9. In the Try Ubuntu session, make sure everything works, including network and keyboard.
10. Double click on install Ubuntu on the desktop to start the installation. Do NOT worry if Wifi is not connected during the installation.

11. During the pre-installation, follow the instruction until prompted “Allocate drive space”.
12. In Allocate drive space, select Something else.
13. Select the desired disk and format it into EXT4 format. If you need a swap space, you can separate desired portion from the EXT4 disk and format into Swap.
14. Select /dev/sda1 as boot-loader installation. Then click on install Now.
15. Follow the instruction during the installation.
16. If you are a Mac User, in “keyboard layout”, select Macintosh keyboard layout.
17. In “Who are you?”, make sure to check the “Log in automatically” selection.
18. After the installation, if you are a Mac User, do NOT restart Mac now and go to the following section.

Fix Problems with Tri-booting a Mac

In this section, we will walk through proper tri-booting a Mac method.

- Fix non-bootable Windows 10, the blue screen Windows 10.
    1. In Ubuntu, make sure you are connected to the internet, open Terminal
    2. Use command `$ sudo apt-get install gdisk` to install gdisk.
    3. Use command `$ sudo gdisk /dev/sda`
    4. Press “P” to verify if you are working on the correct disk. Press “Q” to exit if not.
    5. In gdisk, press “X” to enter expert menu.
    6. Press “N” to creat a new and the press “W” to write and confirm the change.
- Change default boot to Mac in Ubuntu
    1. In Ubuntu, open Terminal.
    2. Use command `$ sudo efibootmgr - v` to load the boot manager
    3. Use command `$ sudo efibootmgr -o 0080,0000` to change the order of boot. Initially, Mac is 0080 and ubuntu is 0000.
    4. Once finish, make sure that Mac is 0000 and ubuntu is 0080.

- Enter tri-boot screen correctly and by default

    1. Reboot into recovery mode by holding “Command” and “R” at the same time.
    2. Click Utilities > Terminal.
    3. In the Terminal window, type in `$ csrutil disable` and press “return”.
    4. Reboot into macOS and log in.
    5. Go to System Preferences > Startup disk, and make sure it is locked onto “macOS”.
    6. Download the “binary zip file” for rEFInd file and put the file somewhere safe on the disk. Here is the link: http://www.rodsbooks.com/refind/getting.html
    7. After finish downloading, open Terminal.
    8. Drag the refind-install file from the just downloaded file into the Terminal and press “return”.
    9. Enter your passcode and press “return”.
    10. When it shows “Installation has completed successfully”, it means it is properly installed.

Ubuntu Setup

In this part of the installation, we will suggest some necessary application in Ubuntu as well as some recommended tweaks to improve your experience with Ubuntu Linux system.

Ubuntu is a open source software that is recommended for software development. If you are a Mac user, Ubuntu looks very similar to macOS that is running on your Mac devices. If you are a Windows user, Ubuntu may looks like a brand new operating system to you. The most important part of Ubuntu is that you have to be familiar with controlling the system via Terminal app. Almost all the commands that you want to make to your Ubuntu system are going through inputs from Terminal, including installing, removing, and updating software and apps.

In Ubuntu, you will find the power of Terminal and the sofiticaticon of the system. People usually have an opinion that it is hard to use Ubuntu, which is decently true, comparing with using macOS or Windows. But do not worry, in the following instruction, we will provide sufficient suggestions that will help you with your Ubuntu.

Installing nmap

Nmap is a useful application in Ubuntu that allows you to visit all the connection for your Wifi host. This application will also be used in the coming instruction.

1. Use command `$ sudo apt-get update` to update your application source list.
2. Use command `$ sudo apt-get install nmap` to install nmap application.

After the installation, nmap application should be installed in your Ubuntu System.

Installing ROS on Personal Computers

ROS system, also known as robot operating system, is going to be the main system that we are going to use to control the Jetson car. The introduction of ROS will be given in the coming chapter.

NOTICE I : This installation process works on AMD64, which is also know as x86_84 by Apple and Intel 64 by Windows. Please check your system and make sure it is using one of those three structures. If not, use the first seven steps of the following instruction and for further installation, go to the link: http://www.ros.org/install/ to select desired software to install.

NOTICE II: The Latest built of ROS that supports the latest Ubuntu LTS version is highly recommended.

NOTICE III: This instruction will using ROS-Kinetic as an example installing on Ubuntu 16.04 LTS.

1. Log into Ubuntu and make sure the network is connected.
2. Open Terminal.
3. Use command `$ sudo apt-get update` to update your application list. Enter your passcode for sudo command.
4. Use command `$ sudo apt-get upgrade` to upgrade your application if there is any update available.
5. Go to “Software & Updates” > “Ubuntu Software”.
6. Make sure all four repositories are checked — main, universe, restricted, multiverse.
7. Click “close” to execute the change.
8. Open Terminal.

9. Use command `$ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'` to set up the ROS source list.
10. Use command `$ sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80 --recv-key 421C365BD9FF1F717815A3895523BAEEB01FA116` to setup your ROS keys.
11. Use command `$ sudo apt-get update` to update your application list.
12. Use command `$ sudo apt-get install ros-kinetic-desktop-full` to install the full desktop version of ROS Kinetic.
13. Use command `$ sudo rosdep init` and `$ rosdep update` to initialize rosdep.
14. Use command `$ echo "source /opt/ros/kinetic/setup.bash" >> ~/.bashrc` to add future packages to bash session automatically.
15. Use command `$ sudo apt-get install python-rosinstall` to install rosinstall.

Now the ROS system should be installed on your Ubuntu.

NOTICE IV: You can install different packages for ROS Kinetic by simply using command `$ sudo apt-get install ros-kinetic-(package name)` to install different packages.

NOTICE V: You can also search for different packages for ROS Kinetic by simply using command `$ apt-cache search ros-kinetic`.

Chapter Two

From Bottom Up

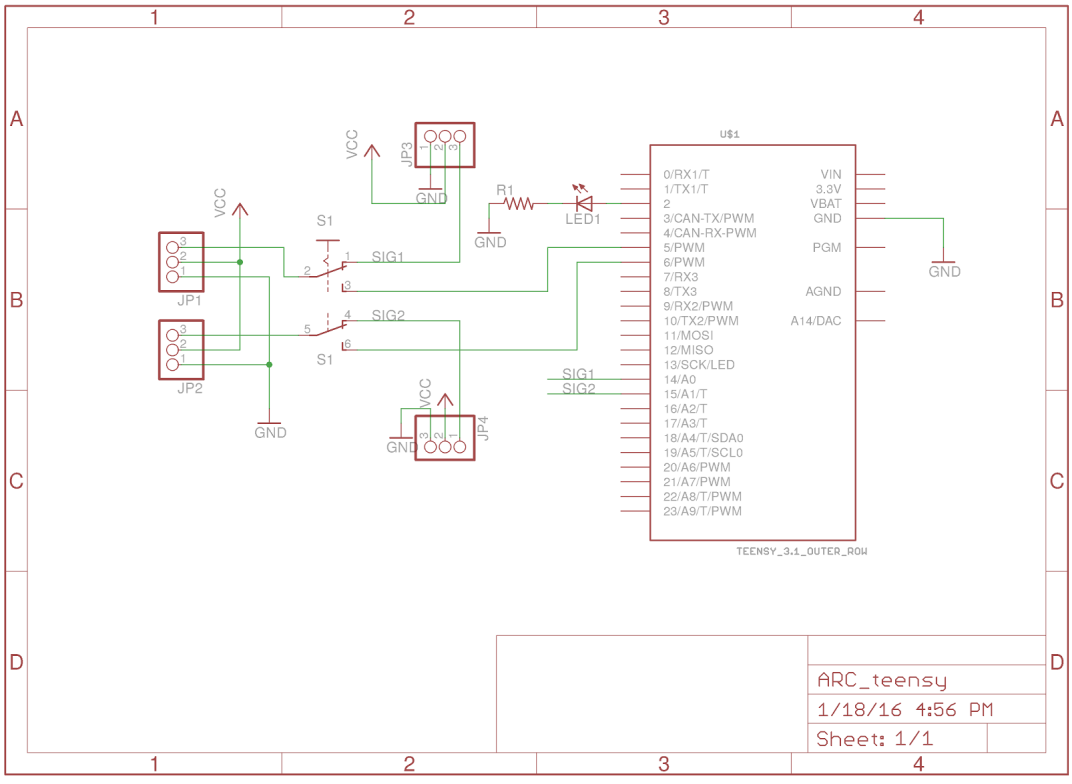
The Building Instruction

In this chapter we will walk through the important part of the building instruction provided by the official site. In our build of the car, we are making some adjustments compared with the official one.

Wiring the Teensy Board

The Teensy Board is give as an Arduino based board. Due to its limited pin connection, we will have to wire a base board for it and connect all the necessary wire on the base board before plug it in to the Teensy.

1. Build the connection provided in the picture on a white board. NOTICE: SIG 1 and SIG 2 are supposed to connected to JP3.3 and JP4.3.



2. For connection, Channel 1, the one close to the edge, in the radio box should be connected to the ESC, VXL-3s; Channel 3 in the radio box should be connected to the servo that controls turning.
3. Make sure that those two wire are easily distinguishable after closing the radio box.
4. Connect the VXL wire, the servo wire, and two radio box wire to the white board and test the connect with the controller.
5. Once the connection works as desired, solder the connection to a green board carefully. Make sure to use different color wires for different meaning.
6. Test the connection and double check the circuit before doing the next step.
7. Put Teensy into the board and test the connection again.

NOTICE I: The power of the controlling system, VCC, is coming from the ESC, which gets its power from the battery. Do NOT connect the VCC to the Teensy's 3V outlet.

NOTICE II: The connection wires from the ESC, the servo, and the radio box should follow the rule which indicates the white pin is connected to ground, the red pin is connected to VCC, and the black pin is connected to signal input.

NOTICE III: Make sure that you will remember which switch controls which part of the signal receiving process, such as Switch 1 in position A connects ESC to Teensy and in position B connects ESC to the radio box.

Construct the Car

The instruction of the Car is generally given by the instruction from the official F1/10 organization. As mentioned before, we are only using part of the whole instruction and please follow the instruction below. In this section, the walk through will be matched by the title of the given official instruction. Only mentioned part is necessary.

1. Connecting the Teensy
2. Mounting the Teensy Board
3. Mounting the Chassis
4. Assemble the Chassis
5. Assemble the Chassis Top
6. Connecting LIDAR to Mount

7. Mounting Wifi to Chassis
8. Mounting Structure Cam to Chassis (Optional)
9. Mounting Front Plate to Chassis
10. Mounting Rear Plate to Chassis (Optional, Not Recommended)
11. Label Wires
12. Wiring Teensy
13. Wiring Nvidia, Jetson

You may use less components if you decided certain components are unnecessary or blocking your wiring. You SHOULD also add some parts that prevent crash damage of the car during future field test and protect on board devices, such as the LINDAR.

NOTICE: In our build of the car, Summer 2017 edition, there is NO IMU unit included in the car.

Chapter Three

Ready Up, System!

On Board Software Configuration

In this chapter, we will walk through the on board system configuration. It includes the setup of the Wifi, the Jetson board, the LIDAR sensor, and the Teensy board.

Switch Default Boot Location of the Jetson

By default, the Jetson will be booted using its on board software. Due to its limited storage space, we will change the default boot location from on board software to external SD card that we put inside the SD card slot.

1. Power on the Jetson and connect it to an external monitor and a set of mouse and keyboard.
2. Insert a SD card to the SD card slot.
3. Use the following commands for preparations:
```
$ sudo midst.ext4 /dev/mmcblk1p1
$ wget http://developer.download.nvidia.com/devzone/devcenter/mobile/
jetpack_tk1/007/linux-x64/Tegra124_Linux_R21.4.0_armhf.tbz2
$ wget http://developer.download.nvidia.com/devzone/devcenter/mobile/
jetpack_tk1/007/linux-x64/Tegra_Linux_Sample-Root-Filesystem_R21.4.0_armhf.tbz2
```
4. Use the following commands to Mount SD card and extract them to SD card:
```
$ sudo tar xpf Tegra124_Linux_R21.4.0_armhf.tbz2
$ sudo mount /dev/mmcblk1p1 Linux_for_Tegra/rootfs
$ cd Linux_for_Tegra/rootfs
$ sudo tar xpf ../../Tegra_Linux_Sample-Root-Filesystem_R21.4.0_armhf.tbz2
```
5. Use to following commands to reRun the apply_binaries.sh script to copy the NVIDIA libraries into rootfs directory
```
$ cd ..
$ sudo ./apply_binaries.sh
```
6. Use the following commands to copy extlinux.conf filePath to a kernel file and kernel parameters are written in /boot/extlinux/extlinux.conf. U-Boot read this file and load kernel. If U-Boot failed to read this file, it tries to boot from other storage.

```
$ cd rootfs/boot/extlinux/
$ sudo cp jetson-tk1_extlinux.conf.sdcard extlinux.conf
```

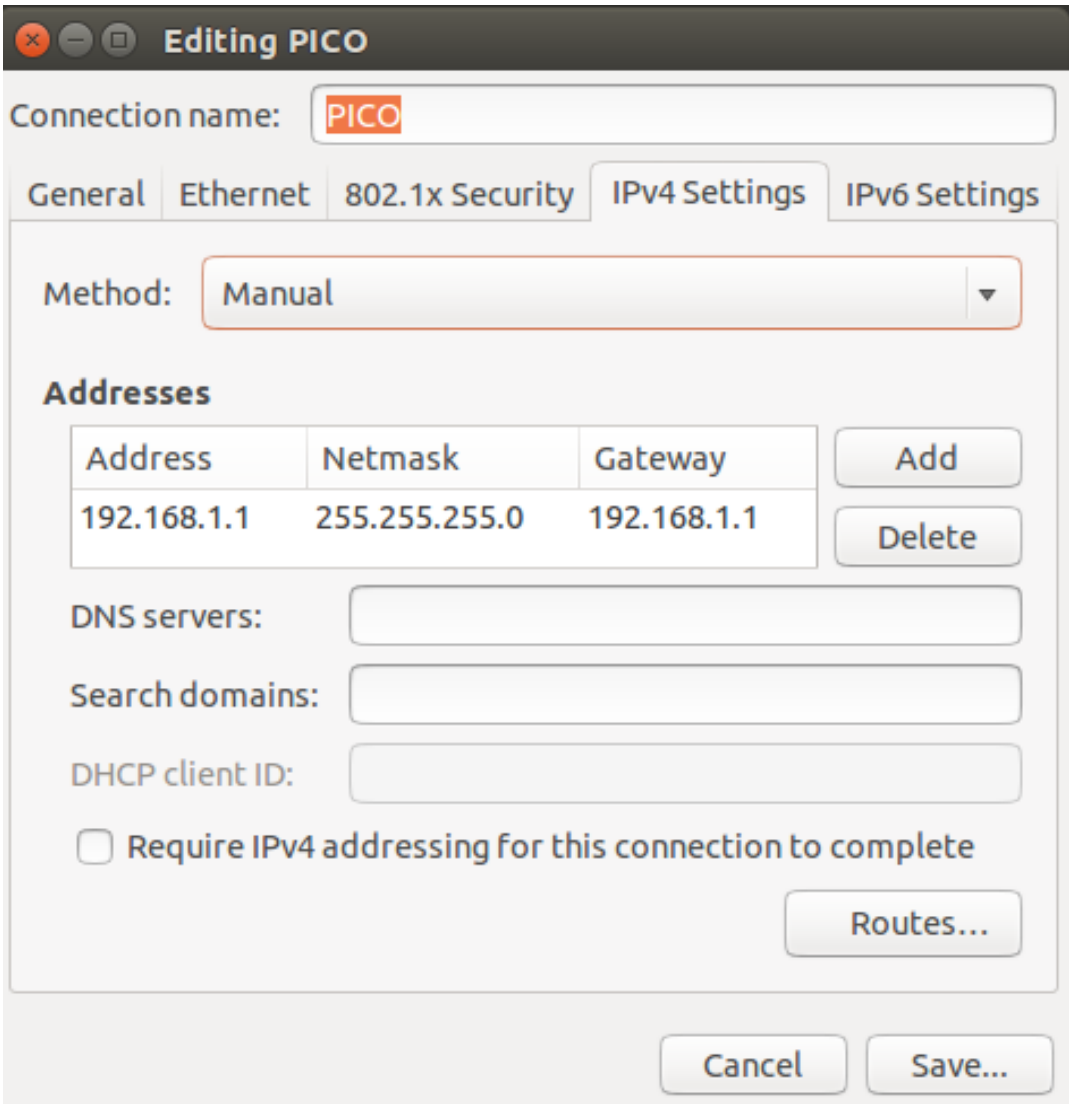
7. Use to following commands to unmount SD card
```
$ cd ../../../
$ sync
$ sudo umount rootfs
```
8. Log into the Jetson and use command `$ df -h` to check root file size.
9. The root size should be the size of the SD card.
10. You HAVE TO make sure that the Jetson is running on SD card before proceed to the next section.

Configuring the PICO Station

The Pico Station will be the access point of the Jetson for remote control and internet access for software installation and updates.

1. Power on the Jetson and connect it to an external monitor and a set of mouse and keyboard. Make sure that Pico Station's cable is plugged in.
2. Open Terminal and use command `$ ls /sys/class/net` to verify the internet connection of the Jetson. Make sure you know which on is Pico Station.
3. Use command `$ ifconfig` if you want to know more about the connection.
4. Go to "System Setting" > "Network" > "Wired" and select corresponding wired connection.
5. Click on "Options" and go to "IPv4 Setting" Tab.
6. Select Method: Mannual.
7. Add the following address — Address: 192.168.1.1 Netmask: 255.255.255.0 Gateway: 192.168.1.1

8. Save and close



9. Open a web browser and access Pico Station's IP address, then log in. The default IP address is "192.168.1.20" and the default user name and passcode are "ubnt".

10. For internet access uses: upgrade system file and install applications go to Step 11 - 14. For access uses: connect to the Jetson broad on other laptops without internet connection, please SKIP Step 11 - 14. However, we do not recommend to put the Jetson in access use only. You can always ssh into the Jetson if it is in the same Wifi network as your laptop.

11. Select and go to "Wireless" tab.

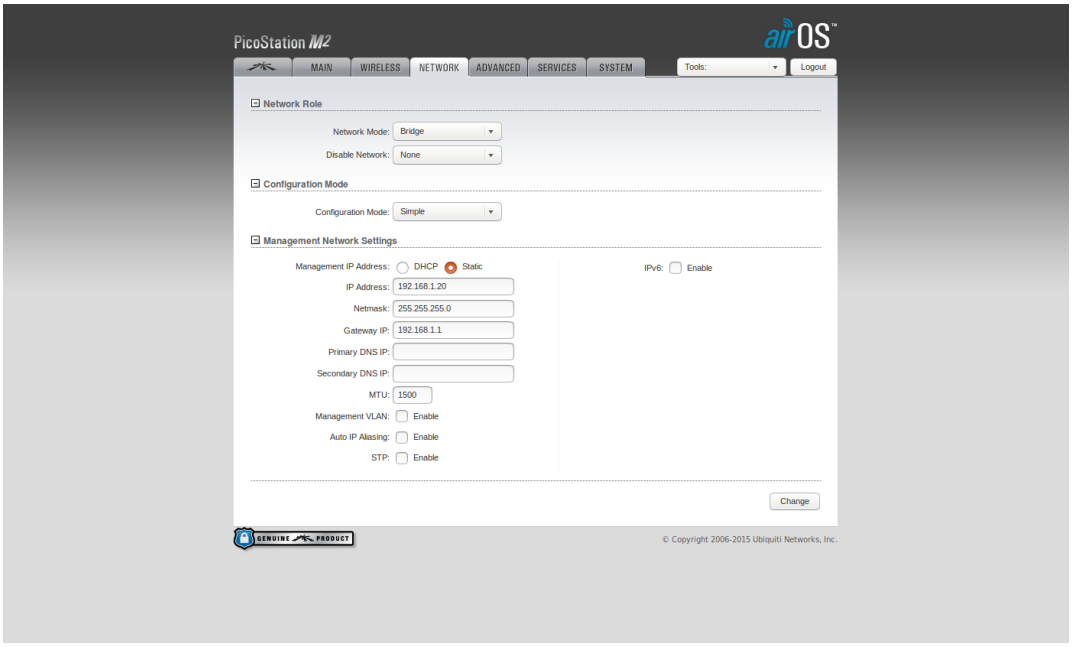
12. Change the following sections:
    Wireless Mode: Station
    SSID: (Corresponding SSID of the Wifi)

Lock to AP: (Corresponding MAC address of the Wifi)
Country Code: United States
IEEE 802.11 Mode: B/G/N mixed
Channel Width: 20MHz
Frequency Scan List, MHz: DISABLE
Calculate EIRP Limit: Enable
Antenna Gain: 0 dBi
Cable Loss: 0 dB
Output Power: 27 dBm
Data Rate Module: Default
Max TX Rate, MBPS: Auto
Security: (Corresponding Wifi Security)
WPA Authenitication: PSK
WPA Preshared Key: (Corresponding Wifi Passwords)

13. Select and Go to “Network” tab.

14. Change the following sections:
    Network Mode: Bridege
    Disable Network: None
    Configuration Mode: Simple
    Management IP Address: Static
    IPv6: DISABLE
    IP Address: 192.168.1.20
    Netmask: 255.255.255.0

Gateway IP: 192.168.1.1
Primary DNS IP: BLANK
Secondary DNS IP: BLANK
MTU: 1500
Management VLAN: DISABLE
AUTO IP Aliasing: DISABLE
STP: DISABLE



15. For internet access uses, please SKIP Step 16 - 20.

16. Go to “Wireless” Tab

17. Change the following sections:
    Wireless Mode: Access Point
    WDS: DISABLE
    SSID: F1TenthBot
    Country Code: United States
    IEEE 802.11 Mode: B/G/N mixed
    Channel Width: 20 MHz
    Frequency, MHz: 2412
    Extension Channel: None
    Frequency List, MHz: DISABLE
    EIRP Limit: Enable
    Antenna Gain: 0 dBi
    Cable Loss: 0 dB
    Output Power: 27 dBm

Data Rate Module: Defualt
MAX TX Rate, Mbps: Auto
Security: WPA2-AES
WPA Authentication: PSK
WPA Preshared Key: fast300a
MAC ACL:DISABLE

18. Select and go to the very first tab, the one before “Main”
19. Uncheck “airMax”, in other words DISABLE airMAX.
20. The “Network” tab should be kept as same as the internet access setup.
21. Open a Terminal.
22. Use command `$ sudo nano /etc/network/interfaces` to access the interfaces file.
23. Inside the interfaces file, make sure that there is NOTHING executed BESIDES the line starts with: source-directory
24. Quit and make sure the configuration of the LAN.
25. In the internet access mode, now you can ssh into the Jetson as long as your laptop is in the same network as the Jetson. In the access mode, now you can ssh into the Jetson as long as your laptop is in the network of the Pico Station.

NOTICE: It is recommended to set the Pico Station in Station mode and connected to certain portable Wifi network.

Installing ROS on the Jetson

ROS will be the system on the Jetson that is controlling the car for different operations. Our command will be loaded as ROS commands and then executed in the Jetson for the car to make any movement.

NOTICE I: In this instruction, the Jetson TK1 is running Ubuntu 14.04 LTS with an ARM32 core.

NOTICE II: In this instruction, we will be installing ROS Indigo on board.

NOTICE III: Although it is recommended to use a full desktop version of ROS, but we are going for the Bare Bone version, which means we will be adding different packages once we need them in the future.

1. In Terminal, use command `$ lsb_release -a` or go to “System Setting” > “Details” to find out the Ubuntu that is running on the Jetson board.
2. Go to “System Setting” > “Software Updates”.
3. In the “Software & Updates” tab, make sure all four repositories are checked — main, universe, restricted, multiverse.
4. Click “close” to execute changes if there is any, otherwise it will just close.
5. Open a Terminal.
6. Use command `$ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'` to setup the source list
7. Use command `$ sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80 --recv-key 421C365BD9FF1F717815A3895523BAEEB01FA116` to setup the ROS key for Jetson.
8. Use command `$ sudo apt-get update` to update the package list.
9. Use Command `$ sudo apt-get install ros-indigo-ros-base` to install ROS Indigo Bare Bones.
10. Use command `$ sudo rosdep init` and `$ rosdep update` to initialize rosdep.
11. Use command `$ echo "source /opt/ros/kinetic/setup.bash" >> ~/.bashrc` to add future packages to bash session automatically.
12. Use command `$ sudo apt-get install python-rosinstall` to install rosinstall.

Now ROS Indigo should be installed on the Jetson.

NOTICE IV: You can install different packages for ROS Indigo by simply using command `$ sudo apt-get install ros-indigo-(package name)` to install different packages.

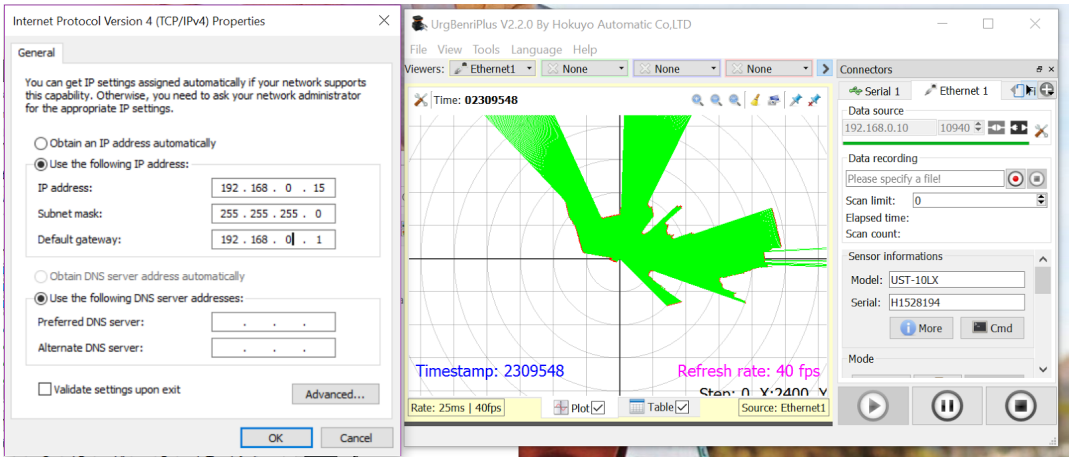
NOTICE V: You can also search for different packages for ROS Indigo by simply using command `$ apt-cache search ros-indigo`.

Configuring the LINDAR Sensor

In this section, we will configure the LINAR sensor and make it suitable to run alongside Jetson with proper data reading and communicating. Also LINDAR will be providing realtime geo-information based on its scan result for the Jetson to understand its location.

1. Log into Windows 10 and download the configuration software UrgBenri. Here is the link: https://sourceforge.net/projects/urgbenri/
2. Once finish downloading, run and install the software.
3. Connect the LIDAR to your laptop via an ethernet cable.
4. Go to “Control Pannel” > “View Network Status and Task” > “Change Adapter Settings”.
5. Select and right click on the corresponding ethernet connection and select properties.
6. Double click and open the Internet Protocol Version 4 (IPv4) tab.
7. Select “Use the following IP address”, and change the value into following values:
   IP Address: 192.168.0.15
   Subnet Mask: 255.255.255.0

Default Gateway: 192.168.0.1



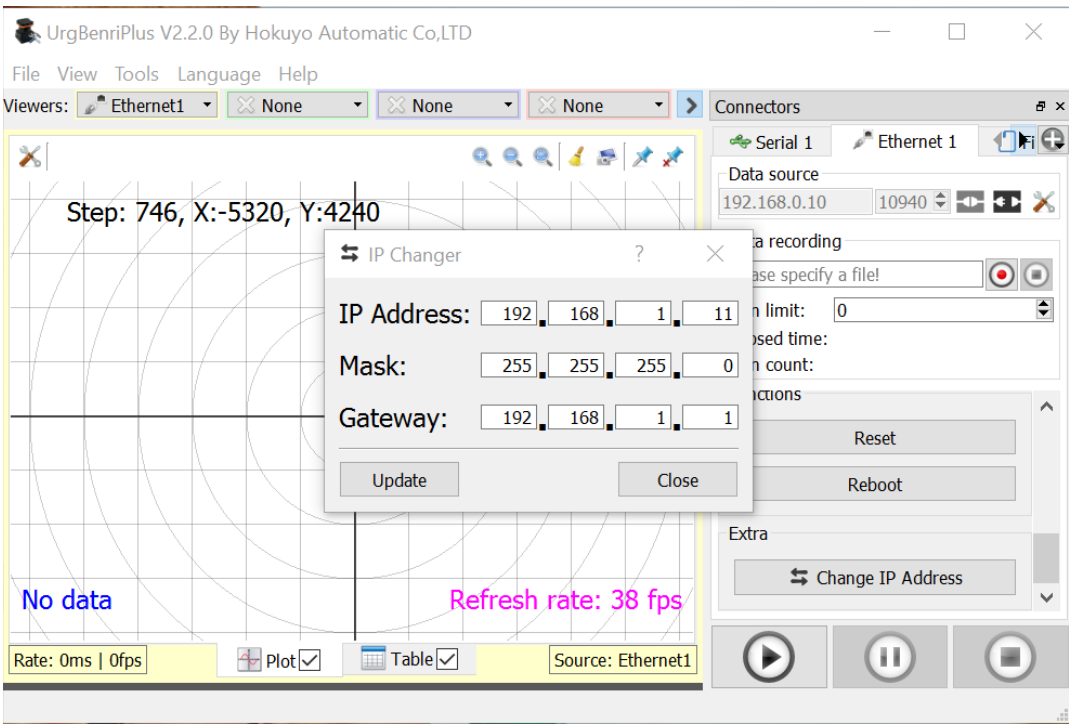
8. Click OK to apply and the UrgBenri will show a map.

9. Scroll down the right column and select “Change IP Address”.

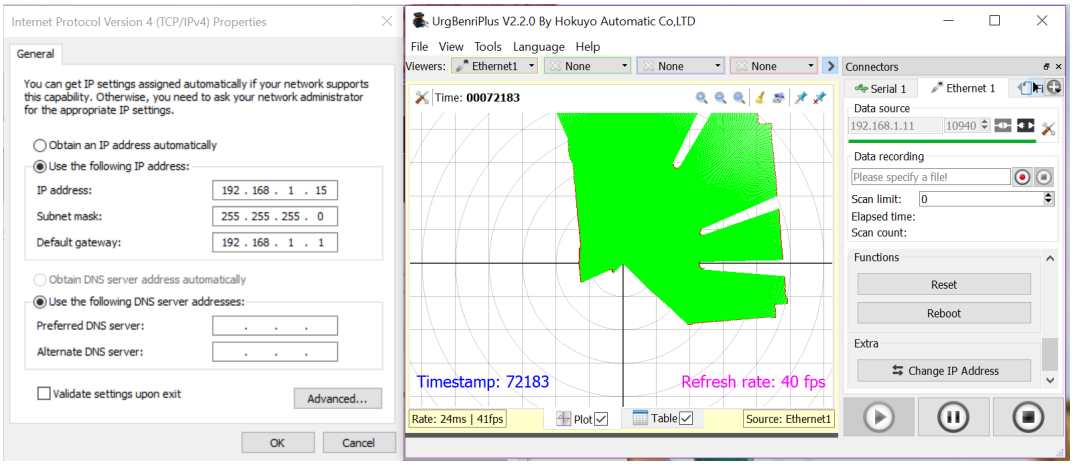
10. Change the LIDAR’s IP address into following values:

    IP Address: 192.168.1.11
    Mask: 255.255.255.0
    Gateway: 12.168.1.1

NOTICE I: This IP Address will be shown up in Ubuntu later on. Make sure you remember this IP Address.



11. Click “Update” to apply changes.
12. The LINDAR will restart and reconnect, the map will go away for now.
13. After LINDAR’s reconnection, change the IPv4 setting into these values:
    IP Address: 192.168.1.15
    Subnet Mask: 255.255.255.0
    Default Gateway: 192.168.1.1



14. Click “Okay” to apply changes.
15. If the map is regenerated in the UrgBenri, then it means the IP Address of the LIDAR has been changed successfully.

Adding a bridge on the Jetson Board

The bridge between the ethernet connection 1, the Pico Station, and the ethernet connection 2, the LINDAR, will provide the access needed for the Jetson to obtain different data from different devices.

1. Put Pico Station in Station Mode to access the internet.
2. Open a Terminal and use command `$ sudo apt-get update` and `$ sudo apt-get upgrade` to update and upgrade your packages.
3. Use command `$ sudo apt-get install autotools-dev` to install autotools.
4. Use command `$ sudo apt-get install autoconf` to install autoconf.
5. Use the following commands to install brctl command function:
   ```
   $ git clone git://git.kernel.org/pub/scm/linux/kernel/git/shemminger/bridge-utils.git
   $ cd bridge-utils
   $ autoconf
   ```

`$ ./configure`
After executing these commands, brctl should be in your command tools.

6. Use command `$ sudo apt-get update` and `$ sudo apt-get dist-upgrade` to force update and upgrade your packages again.

7. Use the following commands to install Grinch Kernel to the Jetson:
`$ git clone https://github.com/jetsonhacks/installGrinch.git` to clone the file under directory: `./installGrinch`
`$ cd ./installGrinch`
`$./installGrinch.sh`

8. After install Grinch Kernel, reboot the Jetson.

9. After rebooting the Jetson, open a Terminal.

10. Use command `$.installGrinch/installGrinchNoDownload.sh` to verify the install.

11. Use command `$ sudo apt-get install bridge-utils` to install bridge utility.

12. Use command `$ sudo apt-get install nmap` to install nmap.

13. Use command `$ sudo apt-get install ros-indigo-urg-node` to install urg-node package.

14. On Desktop, right click and create a new file.

15. Inside the file, put in the following lines:
    sudo brctl addbr br0
    sudo brctl addif br0 eth0
    sudo brctl addif br0 eth1
    sudo ifconfig br0 192.168.1.99 netmask 255.255.255.0 up
    ifconfig br0

16. Save and exit.

17. Right click on the file and go to properties.

18. In “Properties” > “Basic”, change the name into “Bridge.sh”.

19. In “Properties” > “Permission”, allow executing file as program.

20. Save and exit “Properties”.

21. Drag the file into a Terminal and run the file. It will show up the configuration as “br0" once established the connection.

22. Use command `$ nmap -sP 192.168.1.0/24` to find out all the connection to the network that the Jetson is on. The LIDAR should also show up in the form of its corresponding IP address.
23. Use command `$ roscore` to initialize ROS on Jetson.
24. Use command `$ rosrun urg_node urg_node _ip_address:=192.168.1.11` to bring LINDAR online and streaming data.

Create a Brake Bridge Executable

In this section, we will create an executable file to break the bridge we created for the LINDAR in the pervious section.

1. Create a new file on the Desktop.
2. In the file, enter the following lines:
```
sudo ifconfig br0 192.168.1.99 netmask 255.255.255.0 down
sudo brutal delif br0 eth0
sudo brutal delif br0 eth1
sudo brutal delbr br0
ifconfig br0
```
3. Save and exit the file.
4. Right click on the file and go to “Properties” > “Basic”
5. Change the file name into “breakbridge.sh”
6. Go to “Properties” > “Permissions”
7. Allow executing files as program.
8. Click “Close” to exit.
9. Now if you drag this file into a Terminal, it will break the bridge created for the LINDAR. If it shows up as “error fetching interface information: Device not found”, it means the bridge has been deleted.

Setting up Teensy

Teensy will be given as a blank Arduino board. In this section, we will load up the Teensy with correct software for it to control the car.

NOTICE I: The Jetson TK1 is running Ubuntu 14.04 LTS with an ARM32 core.

1. Download Arduino IDE from the official site. Here is the link: https://www.arduino.cc/en/Main/Software
2. Extract the file to directory ":/home".
3. In a Terminal, use sudo command to run the install.sh file of Arduino.
4. Download Teensyduino from the official website. Here is the link: https://www.pjrc.com/teensy/td_download.html
5. Use command `$ sudo nano 49-teensy.rules` to create a new rule file.
6. Click on "Linux udev rules" on the top-right corner of the pervious website. Copy and paste everything inside into the Terminal window of 49-teensy.rules.
7. Save and quit the file editing window.
8. Use command `$ sudo cp 49-teensy.rules /etc/udev/rules.d/` to add the rule into the root system. This will give you access to the Teensy device.
9. Locate the downloaded Teensduino file and use command `$ chmod 755 (File Path)` to run the file in a Terminal. This will make the file executable.
10. Double click on the file to install Teensyduino.
11. In a Terminal, use command `$ sudo apt-get install ros-indigo-rosserial-arduino` and `$ sudo apt-get install ros-indigo-rosserial` to install rosserial packages.
12. Use command `$ mkdir -p catkin_ws_rosserial/src` to create a rosserial package path.
13. Use command `$ git clone https://github.com/ros-drivers/rosserial.git` to clone the rosserial packages from github.
14. In Files, Drag all file inside /rosserial folder into /catkin_ws_rosserial/src
15. Delete the clone folder once finish moving all those files.
16. In a NEW Terminal, use command `$ cd ~/catkin_ws_rosserial` to get inside the catkin_ws_rosserial folder in Terminal.
17. Use command `$ catkin_make` to make the catkin workspace.
18. Use command `$ cd ~` to go back to default page in Terminal or open a new Terminal.
19. Use command `$ sudo nano .bashrc` to access the bashrc file.
20. Add command: `source ~/catkin_ws_rosserial/devel/setup.bash` to the END of the file.

21. Save and exit.
22. Use command `$ git clone https://github.com/mlab-upenn/f1_10_code_public/` to obtain essential files.
23. Copy and paste teensy_drive.ino from /f1_10_code_public/Teensy Firmware to the /home page.
24. Double click on the teensy_drive.ino under /home and click okay once prompt.
25. Go inside the teensy_drive folder and create a folder named “libraries”.
26. Use command `$ cd /teensy_drive/libraries` to get inside the folder in Terminal.
27. Use following commands to create a ROS library for Teensy.

```
$ rm -rf ros_lib
$ rosrun rosserial_arduino make_libraries.py
```

28. After adding the library, restart the Arduino IDE. If you find ros_lib under “File” > “Example”, it means you add the libraries correctly.
29. Connect the Teensy to the Jetson via a USB.
30. Open the Teensy Drive file in Arduino IDE and then go to the Tool menu.
31. Inside the menu, make sure that the broad is “Teensy 3.2/3.1”; Use Type is “Serial”; CPU Speed is “96 MHz”.
32. Upload the drive to the Teensy.
33. Click the button on Teensy once you finish uploading. There should also be a window pop out in the desktop named Teensy.

Chapter Four

Running Partially

Basic Mapping and Controlling

In this chapter, we will introduce the mapping structure and the keyboard controlling structure of the Jetson. The mapping and controlling will be set up individually and run before we close the circle and make it automatic.

Mapping in Realtime

In this section, we will make your laptop receiving the data information generated from the LIDAR and broadcasted by the Jetson.

1. Click on the Wifi on the top-right and view "Connection Information"
2. Get the IP address of the Jetson.
3. Open a Terminal in the Jetson.
4. Use command `$ sudo nano .bashrc` to edit the bashrc file.
5. At the very END of the file, add the following lines:
   ```
   export ROS_MASTER_URI=http://(Jetson's IP Address):11311
   export ROS_IP=(Jetson's IP Address)
   ```
6. Save, exit and close the Terminal.
7. Open a new Terminal. Drag and run the "Bridge.sh" file on the desktop.
8. Use command `$ roscore` to bring ROS online.
9. Open another Terminal and use command `$ rosrun urg_node urg_node _ip_address:=192.168.1.11` to start LIDAR streaming.
10. On your laptop in Ubuntu, ssh into Jetson.
11. Use the following command to connect to Jetson's ROS Master:
    ```
    $ export ROS_MASTER_URI=http://(Jetson's IP Address):11311
    $ export ROS_IP=(Your Laptop's IP Address)
    ```
12. Use command `$ rosrun rviz rviz -l` to view realtime map.

Mapping the Structure

In this section, we will set up the Jetson, the LINDAR, and your laptop to generate a continuous mapping structure. You will be able to walk around the room and obtain the map of the room once it is done.

1. On Jetson, drag and run the "Bridge.sh" file from Desktop in a Terminal.
2. Check Jetson's IP Address. If there is a change, edit the .bashrc file and reopen Terminal.
3. Use command `$ urg_node urg_node _ip_address:=192.168.1.11` to bring LINDAR online.
4. On your laptop in Ubuntu, Use following command to connect to Jetson's ROS Master:

   ```
   $ export ROS_MASTER_URI=http://(Jetson's IP Address):11311
   $ export ROS_IP=(Your Laptop's IP Address)
   ```

5. Download the hector_slam_example file once given. Put the file inside ~/f1tenth folder. If you do not have the folder, create one.
6. Use command `$ cd f1tenth/hector_slam_example/launch` and `$ roslaunch hector_hokuyo.launch` to launch a pre-built rviz.
7. Take a walk with the Jetson.
8. The structure map will be generated in rviz.

Drive with Keyboard

In this section, we will get be able to control the car via keyboard. You will be able to control the servo and the ESC with arrow keys.

1. On Jetson, use command `$ mkdir -p catkin_ws_race/src` in a new Terminal to create file directory for race topic.
2. Use command `$ cd catkin_ws_race` to go inside the folder.
3. Copy and paste the race folder from f1_10_code_public/week_2/ to catkin_ws_race/src

4. After finished the copying process, use command `$ catkin_make` to create the race topic.
5. Copy and paste keyboard.py and talker.py under f1_10_code_public/week_2/race/src into catkin_ws_race/install/share/race
6. Right click on those two files and select “Properties” > “Permissions”. Make these two files “allow executing file as program”.
7. Open a new Terminal and use command `$ sudo nano .bashrc` to edit the bashrc file.
8. At the END of the file, add the following line:
   ```
   source ~/catkin_ws_race/install/setup.bash
   ```
9. Check if the Jetson’s IP Address has changed. If it changes, change the corresponding line in the bashrc file.
10. Save and exit the bashrc file. Close ALL Terminal.
11. Open a Terminal, use command `$ roscore` to bring ROS online.
12. On your laptop, open four Terminals and ssh into Jetson.
13. In Terminal 1, use command `$ rosrun race talker.py` to initialize talker.
14. In Terminal 2, use command `$ rosrun rosserial_python serial_node.py /dev/ttyACM0` to initialize the connection between the Teensy and the Jetson.
15. In Terminal 3, use command `$ rosrun race keyboard.py` to initialize keyboard control for the car.
16. Now the car should be respond to the keyboard commands. It should start moving at up/down 5ish and left/right depending on the visibility of the input angle.
17. Once finish, press “q” in Terminal 3 and “Control” + “C” in Terminal 1 & 2 to quit.

Chapter Five

Where is It Going

Localization and PID Control

In this chapter, we will talk about the Localization and PID control of the car. After this chapter, you will have a better idea on how the car can locate itself in certain environment and how the car can make the decision of which way it is going. Also for the PID control, you will need to create your own python code and in the future, up load to the F1/10 car.

Localization

Localization of the car helps it to know where it is and the surrounding. With the help of the LINDAR, it will be able to make a general guess on where it is and provide essential details for the making of the PID control decision.

PID Control

Also known as proportional–integral–derivative controller, it is a loop feedback mechanism used to make decisions of the turning for the car. In this section, we will talk about what exactly is the PID control and how are we going to utilize it for the car.

Chapter Six

Go by Itself

The Automatic Mode

In this chapter, we will introduce the remote control of the closed system. The car should be running by itself once upload these command to the system.

NOTICE I: Make sure to try to drive with keyboard first before continue.

NOTICE II: You have to complete the code in python file control.py and dist_finder.py.

NOTICE III: After moving those three files mentioned below, there should be five python files in the directory, two of them should be moved in the pervious “Drive with Keyboard” section.

1. On the Jetson, open two python files in directory, ~/catkin_ws_race/arc/race/src, named control.py and dist_finder.py.
2. Create and input your code into those two files.
3. After you finish those two files together with kill.py in the same directory, copy them to directory ~/catkin_ws_race/install/share/race.
4. In your laptop, open seven terminals and SSH all of them into the Jetson.
5. In Terminal 1, use command `$ roscore` to initialize ROS system.
6. In Terminal 2, use command `$ ‘/home/ubuntu/Desktop/Bridge.sh’` to build the bridge of the LIDAR and the Jetson.
7. In Terminal 2, use command `$ rosrun urg_node urg_node _ip_address:=192.168.1.11` to start LIDAR broadcasting.
8. In Terminal 3, use command `$ rosrun rosserial_python serial_node.py /dev/ttyACM0` to start the communication between the Teensy and the Jetson.
9. In Terminal 4, use command `$ rosrun race talker.py` to start the talker.
10. In Terminal 5, use command `$ rosrun race kill.py` to make the kill operation standby.

11. In Terminal 6, use command `$ rosrun race control.py` to start the auto-control of the car.
12. In Terminal 7, use command `$ rosrun race dist_finder.py` to start the distance finding procedure. And once you enter this command, the car will start moving.
13. Use "\" for normal stop and "(SPACE)" for emergency stop of the car in Terminal 5.

Chapter Seven

Oh, Obstacles

Q&A for common errors

In this chapter, we will address some common mistakes and errors that may come across the procedure of building and running the car.

Ubuntu Cannot Connected to Enterprise Wifi Network?

Ubuntu is able to connect to enterprise wifi network, such as school network. However, you will need to change some setting in the Wifi connection Tab.

The UrgBenri Does Not Show the Map after Clicking “Okay”?

Double check your input into the IPv4 setting and try it again. If you have already changed the default LINDAR IP address, change the input IP address into corresponding subnet. If it still does not show up, it probably means that the LINDAR is broken.

No Internet Connection after Established Bridge on Jetson?

It is common that once the bridge on Jetson is set up, the internet will NOT work. If you want to access the internet, break and delete the bridge. You can also restart the Jetson to remove the bridge.

Pico Station cannot access to the Internet?

Probably, there is something wrong with the setup of the Pico Station. Make sure that the setup looks exactly like the instruction. If there still is a problem, make sure that it is actually locked on to certain IP Address’ MAC address.

The ESC, VXL-3s, is Blinking Green Once Connected to Teensy?

It is blinking green means some kind of calibration is needed. Check your connection, and make sure they are all connected correctly. And also double check the connecting pins,

there is a chance that the pin is broken. In order to switch for different pins, check the teensy diagram and select the pin that has PWM as output signal.

The Car is Not Responding to Inputs?

Make sure that the power to the VXL is connected and the VXL is switched on. Check the wiring of the Teensy board, the connection of the pluggable wire, and the switch position. For the switch position, both switch should be in the corresponding position. You will not be able to control the car if one of the switch is switched to the controller while the other is to the Teensy.