###README

Capstone Group: PN13

Name of the group members: Aryan Chahardovalee, Anna Wang, Kris Zhou, Rain Zhang, CK Chan

Name of the Project: Gesture Controlled Vlogging Drone

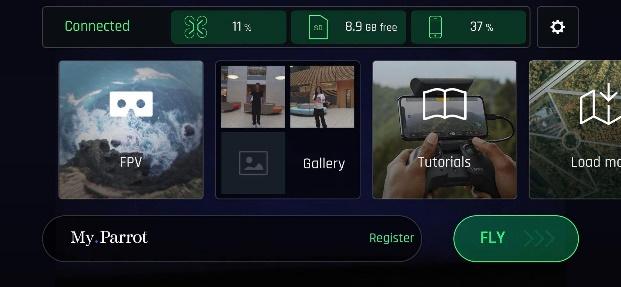
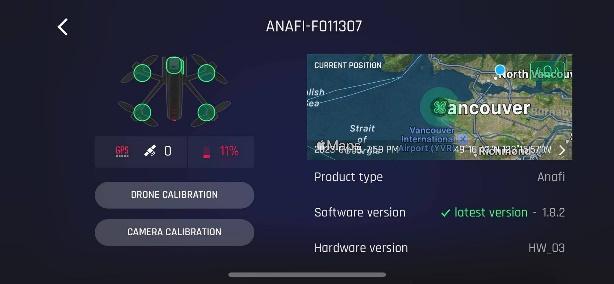
Hardware in Use:

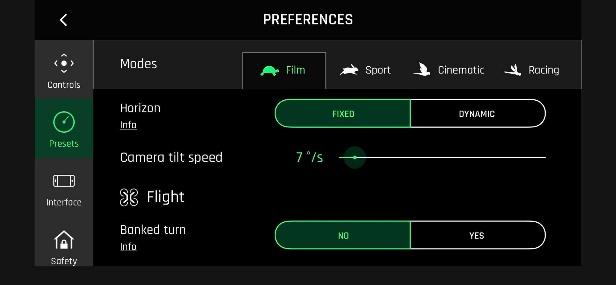
* Sensor Module: Arduino Nano 33 IoT + 2 Polulu time-of-flight (TOF) sensors + LiPO battery + one front sensor + sensor case + sensor clamp for mounting the sensor on the drone
* Drone: Parrot Anafi Base Drone + extra battery
  + 1080p video
  + 25 minutes flight time
  + For full documentation, refer to useful links
* Huawei Atlas board 200 DK + bootable SD Card (Linux 18.04)
* TP-Link AC750 Wi-Fi Travel Router
* PC (not included; the PC needs to run Windows with MobaXterm Installed and all the instructions in Section 5 – Quickstart DEMO must be followed)

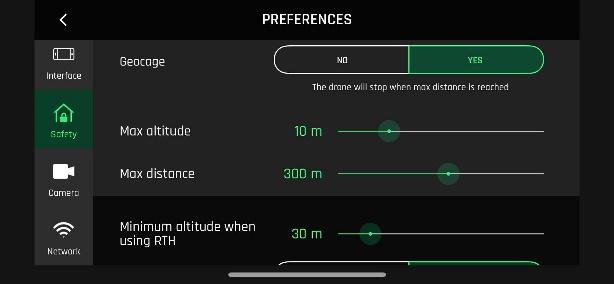
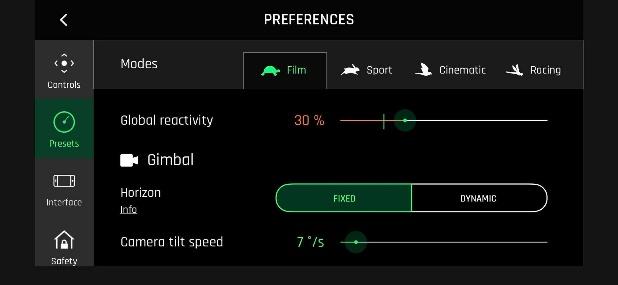
Upon receiving the package from Capstone Group 13, the client can jump to Section 5 to perform a demo. It should be mentioned that as a result of a recent crash the front left-side leg of the drone broke and after capstone team 13 changed the broken leg the drone flies abnormally. The fix for this is to change the current rotor back to the old rotor. Instructions on how to do this are below in Section 1 - Drone. The 3D printed arduino casing is cracked and both Pololu boards (normally containing 6 sensors each) are missing one sensor chip each. We recommend reprinting the 3D case (files provided) and replacing the sensors. The rest of the sections are included for better understanding the system and to be able to reproduce the system if need be. Pay attention that upon each unsuccessful flight (CRASH), the drone needs to be re-calibrated for best performance (Section 1 - Drone). Additionally, the sensor reset button needs to be pushed once to reset the sensor if the connection to the drone is lost (Section 3 - Arduino).

**Section 1 - Drone:**

After each crash, the drone needs to be recalibrated. Download the FreeFlight6 App from the Apple AppStore and connect to the drone with your phone to see many of the settings that can be controlled through the App. The App can be used to calibrate the drone and the camera as necessary. The specific pre-sets of the drone, such as maximum height and tilt, can also be adjusted as specified in the app. Videos can be viewed from the drone and can also be downloaded to the phone.

Make sure to disconnect the drone from your phone before attempting to fly the drone from the ATLAS board with code. Make sure that the drone's propellers are not jittery after a crash. If the propellers are jittery, turn the drone on and off. The drone takes some time to be pingable after you turn it on and off; after each power-on, wait for all the four propellers to move slightly and then wait for the gimbal to calibrate. Then the drone will be pingable again. If this doesn't solve the ping issue, go to the Section 2 - Wi-Fi Modem.

Finally pay a lot of attention when charging the drone, we have burned one once, and it cost a lot! Also, this drone is not commercially available anymore, so you have to resort to third party vendors!

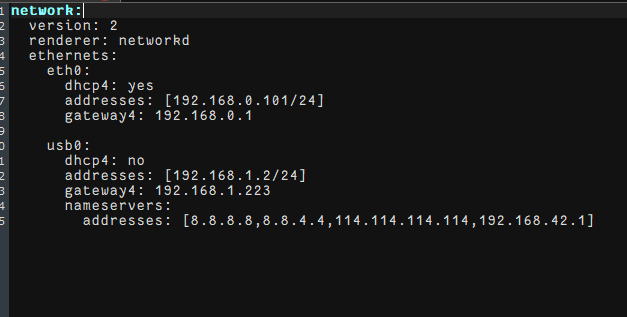
One leg of the drone broke on the very last day of the project. Capstone Team PN-13 replaced the broken leg of the drone with the old drone’s corresponding leg. As a result the drone behaves abnormally even during a simple takeoff. The solution could be to change the motor that was mounted on the broken leg of the drone to what it was previously. Take the old rotor out of the broken leg and replace it with the rotor that is already mounted on the drone. The specific leg rotor is marked with two tape strips on the leg! A video is attached in useful links that explains how the Anafi can be disassembled and reassembled. Hopefully this will fix the drone, if not consult the parrot Anafi forum and ask for help online.

**Section 2 – Wi-Fi Modem:**

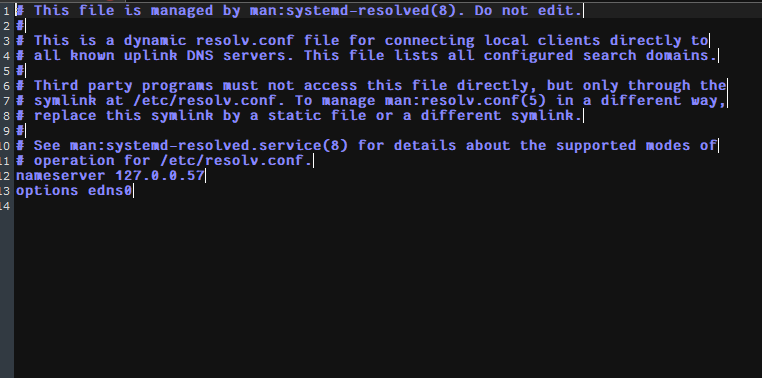
Two files are responsible for the wifi connection. At the moment, they are specified so that the connection to the drone and sensor is properly established. If you need the internet to download something, some modifications must be done to one of these files and then a certain sequence of events have to be performed to establish an internet connection. To get the drone and sensor connection back there is a similar sequence that must be followed and is explained further in this section.

To connect to the drone and the sensor the following files should be as displayed:

cd /etc/netplan then sudo vi 01-netcfg.yaml: insert and change as specified (you need to know how to use Vi or you can just edit and save using the text editor)

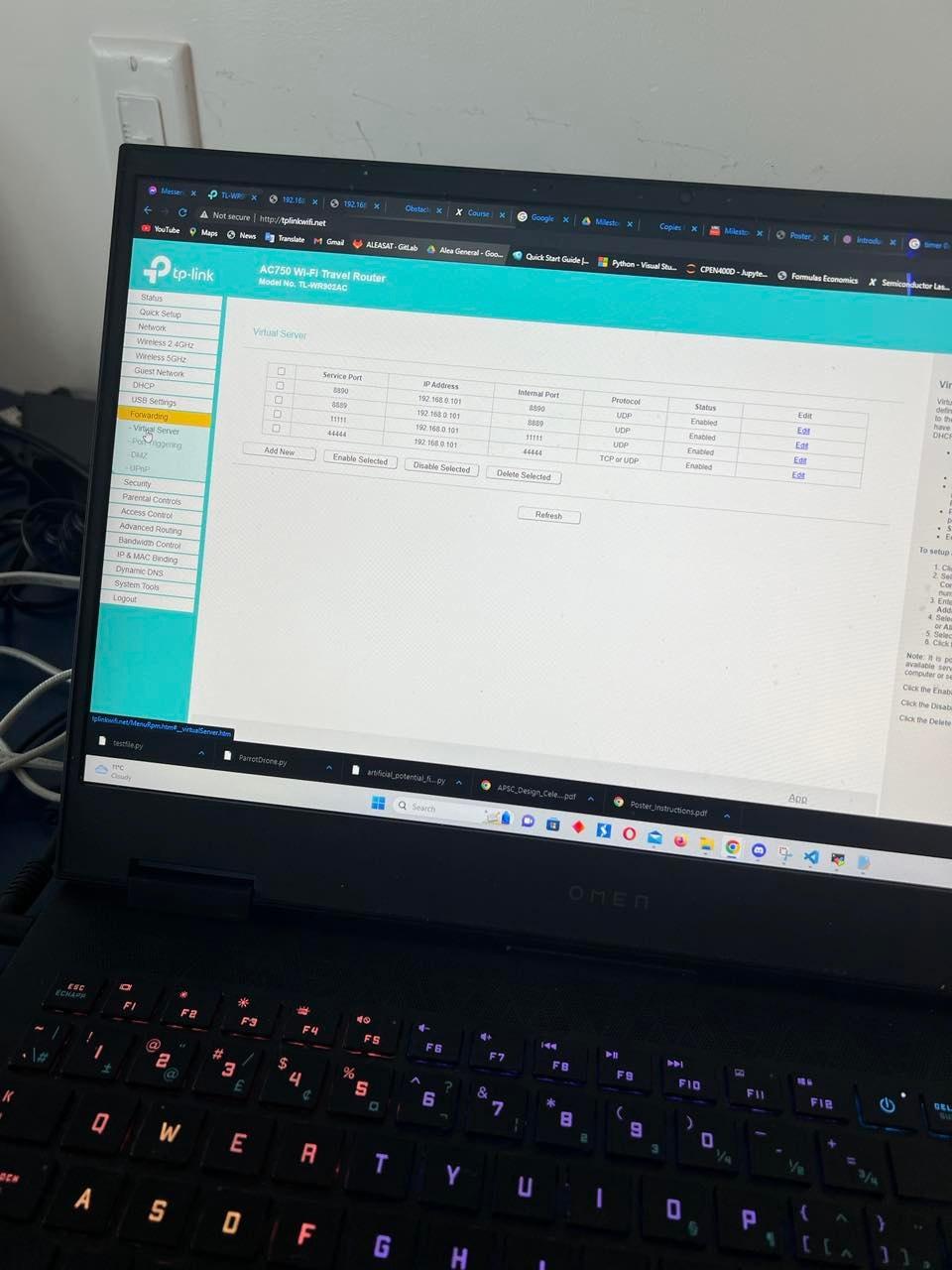


cd /etc then sudo resolv.conf:



**TPlink website: To connect a new drone to the TPLink Wifi or verify the drone and sensor information in the TPLink Modem:**

Connect the TP Link while it is powered on to your computer's Ethernet port. Then navigate to www.tplinkwifi.net and put the username: admin and password: admin. if unsuccessful to see the username and password prompt do either one of Solution One or Solution Two in the TP-link website (useful links). To connect a new drone, do "Quick Setup'' and follow the pre-set configurations. If you change the name of the TP-Link Wi-Fi from “TP-Link\_A5A3\_Parrot'' to something else or if you change the TP-Link modem’s password from “90321702” to something else you have to change the code running on the Arduino Nano to be able to communicate with the sensor module. If you overlook this, you will never get any data from the sensor. Additionally, some ports are specified in the TP-Link GUI as below:



The TP Link Quick Setup for 2.4 GHz network can recognize the Parrot Wi-Fi and you only need to type in the Drone's Wi-Fi password to setup the drone. This has already been done! If you switch the side button of the TP-Link Wi-Fi modem all of these settings would reset so make sure to avoid touching it (it has already been taped to Hotspot Mode!)

**Connect to internet:**

1. To connect to the internet, you navigate to the 01-netcfg.yaml file and change it to the following:

Network:

version: 2

renderer: networkd

ethernets:

eth0:

dhcp4: yes

usb0:

dhcp4: no

addresses: [192.168.1.2/24]

gateway4: 192.168.1.223

nameservers:

addresses: [8.8.8.8, 8.8.4.4]

1. Then after you saved the new 01-netcfg.yaml file you must do “sudo netplan config”
2. Afterwards turn EVERYTHING off and turn everything on again. Wait for the RNDIS port to show up on “View Network Connections”
3. Right-click on the current Wi-Fi connection and select "Properties''. Then in the "Sharing" tab, **disable** the option "Allow other network users to connect through this computer's...". Then press ok, Exit and do the same thing but **enable** the option “Allow other network users to connect through this computer's...". Then, from the dropdown menu, choose the Ethernet connection that you assigned to the USB-C port, coming from the Atlas 200DK board.
4. Now, navigate back to "View Network Connections", right-click on the Ethernet connection that is RNDIS (coming from the Atlas 200 DK Board). Then, go to "Properties" and click on TCP/IPv4 connection in the dropdown menu. Choose "Use the following IP address:" and specify the IP address as 192.168.1.223 and the subnet mask to 255.255.255.0, then press OK and then make sure everything in view network connection is as specified above. Double-check for sure!!!
5. In MobaXterm, create an SSH terminal by clicking on Session - SSH, then Remote host is 192.168.1.2 and specify the username as "HwHiAiUser". Password is Mind@123
6. You should have internet now, if not try connecting the USB-C to other ports of your computer as this has proven to be a crucial factor in obtaining internet access. Try the above multiple times with different ports until you can “ping google”. If pinging error doesn’t give you any errors, you are connected to the internet! Most packages are already installed so there isn’t much need for internet connectivity for the Atlas Board, but you can try with the above and hopefully it will work. Finally delete the last two lines of the resolv.conf file and add 8.8.8.8 and 8.8.4.4 instead AS A LAST RESORT to see if you can get internet connectivity.

**Connect the Drone + Sensor after internet connectivity is not needed anymore!**

1. To connect to the drone again you navigate to the 01-netcfg.yaml file and resolv.conf file and change it to what it was initially at the beginning of the Section 2 – Wi-Fi Modem.
2. Then after you saved the newly-edited files you have to do ‘sudo netplan config’
3. Afterwards turn EVERYTHING off and turn everything on again. Wait for the RNDIS port to show up on “View Network Connections”
4. Right-click on the current Wi-Fi connection and select "Properties". Then in the "Sharing" tab, **disable** the option "Allow other network users to connect through this computer's...". Then press ok, Exit and do the same thing but **enable** the option “Allow other network users to connect through this computer's...". Then, from the dropdown menu, choose the Ethernet connection that you assigned to the USB-C port, coming from the Atlas 200DK board.
5. Now, navigate back to "View Network Connections", right-click on the Ethernet connection that is RNDIS (coming from the Atlas 200 DK Board). Then, go to "Properties" and click on TCP/IPv4 connection in the dropdown menu. Choose "Use the following IP address:" and specify the IP address as 192.168.1.223 and the subnet mask to 255.255.255.0, then press OK and then make sure everything in view network connection is as specified above. Double-check for sure!!!
6. In MobaXterm, create an SSH terminal by clicking on Session - SSH, then Remote host is 192.168.1.2 and specify the username as "HwHiAiUser". Password is Mind@123
7. Make sure the drone is on and ping the Drone: ping 192.168.42.1
8. Make sure the sensor is on and ping the Sensor: ping 192.168.0.152

**Section 3 - Sensor:**

When the sensor is powered on it first shows a green light, that means all the connections are good. Two of the sensors on top of the Polulu is broken and we recommend getting 2 new Polulu boards. At the moment the two broken sensors are being ignored in the python code. If after power on the LEDs are solid red, try resetting the arduino board by pressing the button on the arduino nano. If it stays red, that means that some connections are loose! To fix the connections use a tweezer and push the wires into the holders to make sure everything is connected properly. If the light turns green after power on it means all hardware connections are ok!

After the all-green LEDs, the LEDs turn dark blue, which means the sensor is trying to connect to the TP-Link Modem. If the name of the Wi-Fi (“TP-Link\_A5A3\_Parrot”) and the password (“90321702”) are not changed then the sensor will automatically connect to the TP-Link Modem. If for any reason these parameters change, we need to upload a new Arduino elec491proj.ino file into the Arduino Nano by connecting it to a computer and using the Arduino IDE application. After a brief moment when the LEDs are dark blue, they start flickering and some of them turn red and some will stay blue. This stage indicates that the sensors are active, and data is being sent to the TP-Link Modem.

Make sure in the python code that receives sensor data the UDP port is specified the same as in elec491proj.ino. This is essential for getting the data. It is currently set to 12345.

Sometimes although the sensors LEDs are flashing and changing, the sensor cannot be pinged. Simply press the reset button on the Arduino Nano and it will be pingable within seconds, if the above is done correctly.

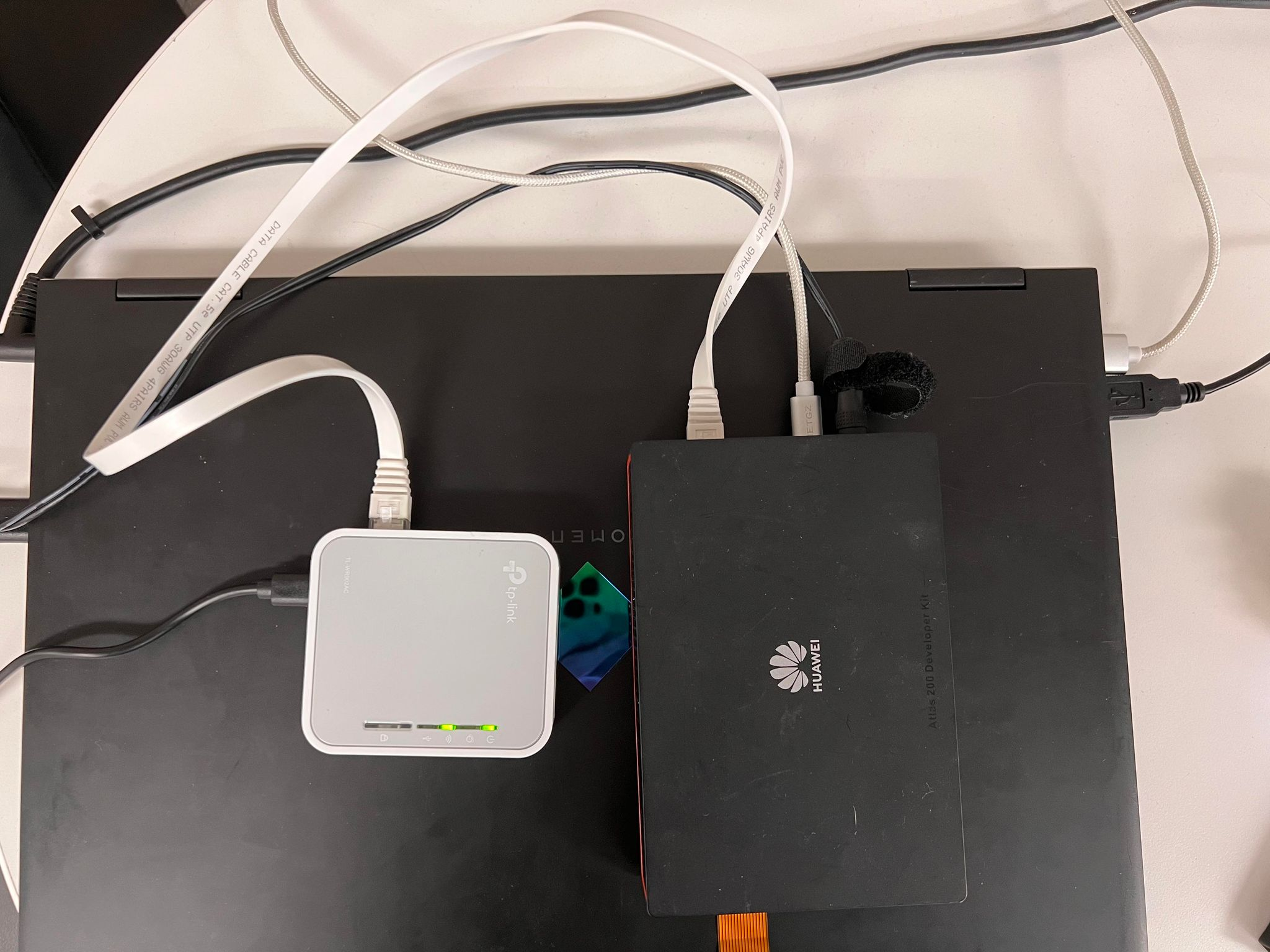
The case holding the sensor boards and arduino broke on the very last day of the project. Please make sure to design and 3D print a new case before attempting to run the code.

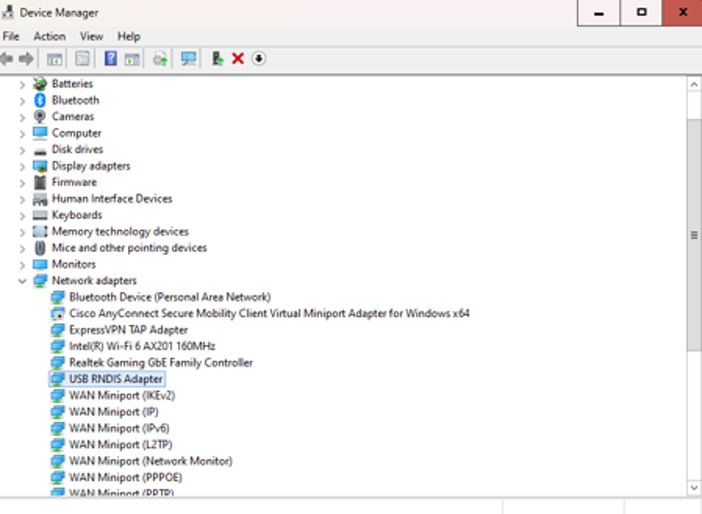
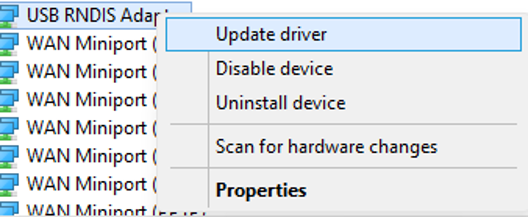
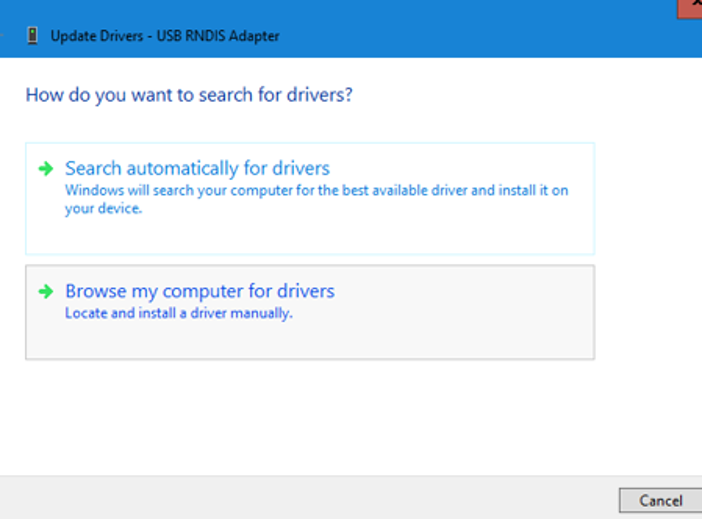
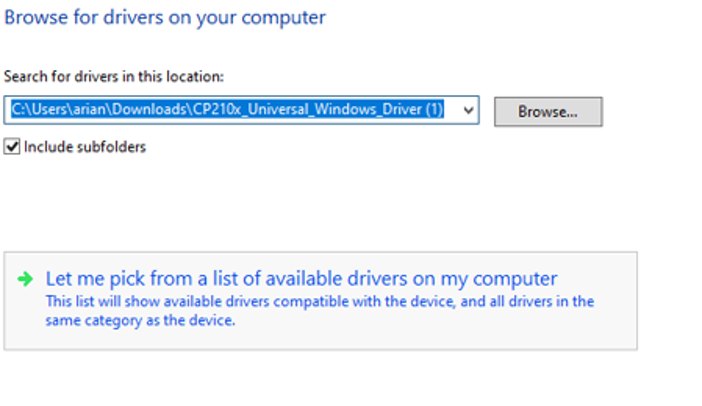
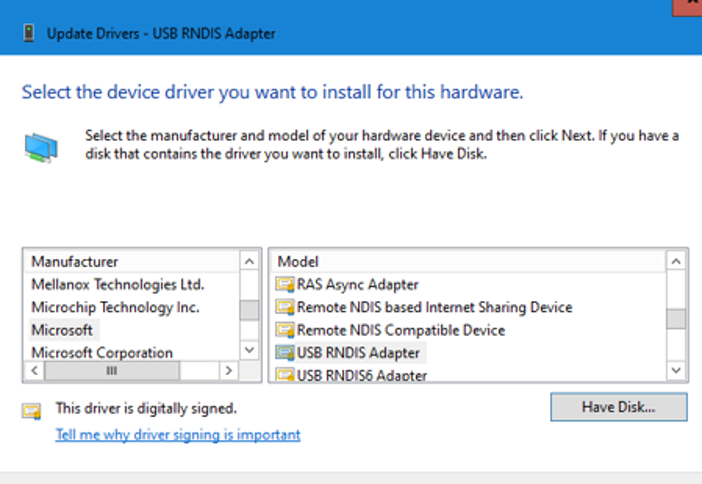
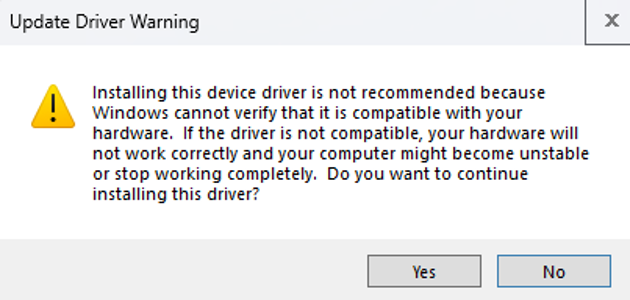
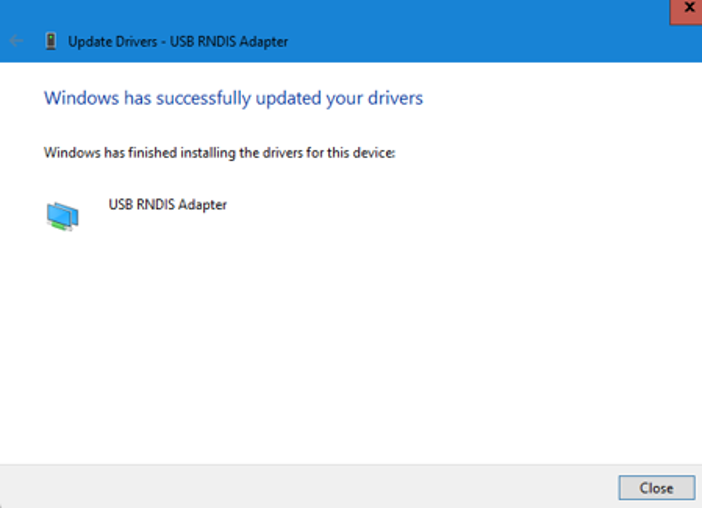
**Section 4 - Atlas board:**

The board is running Linux 18.04 and most of the files for this project can be found under cd code/gsdk3. Packages are installed in various environments and due to the sheer number of the packages, they are not included in this README file. The bootable SD card that is currently running on the board contains all the packages that are necessary for running the project. The SD card of the board needs to be always in the board for the Atlas 200 DK board to function properly. To investigate the files inside the SD card, you need a computer running LINUX AND that computer needs to have a SD card reader. Windows computers require you to format the disk, which will cause everything to disappear, so don’t do that!

**Section 5 - Quickstart DEMO:**

1. Make the following connections as shown in the picture.
   1. Connect the Atlas 200 DK Board to power and connect the TP-Link Modem to the Atlas 200DK board using a network cable.
   2. Connect the TP-Link modem to power.
   3. Connect the USB\_C cable from the Atlas board to your laptop. Then turn on the drone and the sensor module.



1. Set up the USB\_C (RNDIS).
   1. In device manager find USB RNDIS. This may be in “other devices”
   2. 
   3. 
   4. 
   5. 
   6. 
   7. 
   8. 
2. Go to "View Network Connections" on your PC and right-click on the current Wi-Fi connection and select "Properties''. Then in the "Sharing" tab, check the option "Allow other network users to connect through this computer's...". Then, from the dropdown menu, choose the Ethernet connection that you assigned to the USB-C port, coming from the Atlas 200DK board.
3. Now, navigate back to "View Network Connections", right-click on the Ethernet connection that is RNDIS (coming from the Atlas 200 DK Board). Then, go to "Properties" and click on TCP/IPv4 connection in the dropdown menu. Choose "Use the following IP address:" and specify the IP address as 192.168.1.223 and the subnet mask to 255.255.255.0, then press OK and then make sure everything in view network connection is as specified above. Double-check for sure!!!
4. Install MobaXterm for additional convenience while SSH-ing into the board.
5. In MobaXterm, create an SSH terminal by clicking on Session - SSH, then Remote host is 192.168.1.2 and specify the username as "HwHiAiUser". If there is a VPN running on your computer (e.g. UBC VPN, …) the Atlas Board will not be able to connect through SSH.
6. If everything has been done correctly, you should be prompted with a message asking for the password: Mind@123. Ignore the message about saving the password. If you do the above correctly, you will only need to start from this step when demoing the project.
7. Make sure the drone is on and ping the Drone: ping 192.168.42.1 (if not pingable see (Section 1 - Drone) and (Section 2 - Wi-Fi Modem)).
8. Make sure the sensor is on and ping the Sensor: ping 192.168.0.152 (if not pingable see (Section 3 - Sensor) and (Section 2 - Wi-Fi Modem)).
9. If successfully ping both, then run the following commands exactly as specified:
   * cd code/gsdk3/products/olympe/linux/env
   * source setenv (ignore the error, not finding the library)
   * cd
   * cd /usr/share/zoneinfo
   * sudo chmod +rwx UCT
   * cd
   * cd code/gsdk3/packages/olympe/src/olympe/doc/examples
   * source ~/.bashrc
   * export LD\_PRELOAD=/usr/lib/aarch64-linux-gnu/libgomp.so.1
10. Run the following commands exactly as specified:
    * python3.7 StreamAndFindPerson.py

**Section 6 – Additional Debugging:**

Sometimes if you do not close the UDP port that communicates with the sensor properly, you will get missing events error which can be solved with this code:

* netstat -lnp | grep 12345
* kill -9 <pid>

where <pid> is the process id that the port is open on

To change the python version that the system is currently running on:

* “sudo update-alternatives --config python” or “sudo update-alternatives --config python3”

The python3.7 command for StreamAndFindPerson.py, testfile.py, PARROTML.py and takeoff.py should work as intended. Make sure to DEMO the Parrot Drone in an environment that is free of obstacles as the drone should have some space to maneuver when it first initializes. The UBC Health Sciences building is a prime location! Make sure to fix the motor of the drone before flying it again. Make sure to acquire a new case for the sensor (obstacle avoidance) module.

**Description of python code:**

* ParrotDrone.py: Miscellaneous code for drone connection, etc.
* StreamAndFindPerson.py: Command the drone based on both ML model and the sensor readings
* artificial\_potential\_field.py: Contains the artificial potential field algorithm
* testfile.py: Command the drone based on the sensor readings, do “python3.7 testfile.py” to run the stand alone sensor-command code
* takeoff.py: Sample code to take off and land the drone to see the stability of drone
* PARROTML.py: Code example for running the standalone ML model – Third Version

**Section 7 - Future Work:**

It is advised for a Linux expert to be part of this project so that all the Linux debugging is done faster and more efficiently, without damaging the rest of the environments. Make sure to read through all of the useful links as all the information here comes from these websites and if there is anything missing, there is a high chance that it could be found in one of the provided links!

For starters, the future capstone project can get the presenter server running. Presenter server allows tracking to be viewed from the PC GUI (MobaXTerm). This will require additional packages and will require the Atlas board to be connected to the internet to download and install the packages.

Fixing the internet connection so that the drone simultaneously connects to the board while the internet is active is another goal for the future capstone group.

The Parrot Drone also responds slowly to movements and more modification needs to be done to make it more responsive.

**Useful links:**

<https://github.com/Ascend-Huawei/Obstacle_Avoidance_Stack>

<https://github.com/Ascend-Huawei/HiFly_Drone/wiki/TP-Link-Wireless-Router-Setup>

<https://github.com/Ascend-Huawei/gesture-controlled-drone>

<https://www.hiascend.com/document/detail/en/CANNCommunityEdition/60RC1alphaX/softwareinstall/instg/atlasdeploy_03_0002.html>

<https://www.hiascend.com/document/detail/en/Atlas200DKDeveloperKit/1013/environment/atlased_04_0001.html>

<https://developer.parrot.com/docs/olympe/installation.html>

<https://github.com/Parrot-Developers/olympe/blob/gsdk-1.0-branch/src/olympe/doc/installation.rst>

<https://www.youtube.com/watch?v=cKK8-iR-ERs&t=70s&ab_channel=Parrot>

<http://tplinkwifi.net/>

<https://www.parrot.com/assets/s3fs-public/2021-09/anafi-user-guide.pdf>