Omnicopter Project
Full Motion-Capture Setup

September 2024

Table of Contents

Optitrack/Motive

Hardware requirements

1. OptiTrack camera and computer system.
2. Retroreflective markers and adhesive
    a. Used Command strips cut into squares

Initial setup

1. Start Motive application.
2. Follow the instructions here (https://docs.optitrack.com/motive/rigid-body-tracking) to attach the markers to the omnicopter body and create a rigid body object in Motive.
    a. Note: name the object something recognizable such as “omnicopter”.

Each run

1. Start Motive application.

Ground Control Station (GCS) Computer

Machine requirements

1. Windows 10 (TODO: and 11?)
2. Ethernet port to connect to the Optitrack system
3. Ability to broadcast a Wifi hotspot from the computer

QGroundControl

Initial setup

1. Follow the instructions in this link to install QGroundControl onto the GCS: https://docs.qgroundcontrol.com/master/en/qgc-user-guide/getting_started/download_and_install.html

Each run

1. Open QGroundControl and wait for the status in the top left corner to change from "Disconnected" to another status (usually "Ready" or "Not Ready").

Motive Connection

Initial setup

1. Set up the ethernet connection:
   a. Connect via ethernet cable to lab computer running Motive and connected to Optitrack camera array.
   b. Open ControlPanel -> Network and Internet -> Network and Sharing Center and click "Change adapter settings" on the left side of the window.
   c. Right-click on the ethernet device being used by the Motive ethernet connection and select "Properties".
   d. Under the Networking" tab, double-click on "Internet Protocol Version 4 (TCP/IPv4)".
   e. Select "Use the following IP address" and enter these values for the following fields:
      - IP address: 192.168.11.3
      - Subnet mask: 255.255.255.0
   f. TODO: Also mention DNS server addresses?
   g. Click "OK", then "OK".
   h. Close the network windows.
2. Install AftrBurner/OptiTrack_Viewer
   a. Follow the instructions in this YouTube video (https://youtu.be/hPGZf2dHSG0) from timestamps 0:00 to 10:32 to install AftrBurner onto the computer.

   b. Extract the “optitrack_viewer-omnicopter.zip” file located in the “GCS” repo directory and copy the whole folder to the directory “C:\repos\aburn\usr\hub”.

Each run

1. Connect to OptiTrack ethernet:
   a. Connect via ethernet cable to lab computer running Motive and connected to OptiTrack camera array.
2. Begin reading OptiTrack data onto the GCS:
   a. Run the “OptiTrack_Viewer.exe” executable located in the directory “C:\repos\aburn\usr\hub\optitrack_viewer-omnicopter\cwin64”.
   b. A window should appear titled “AftrBurner Engine”.
   c. In that window, click “Mocap Tools” in the “Menu” window, then “Show Object Viewer”.
   d. In the “Object Viewer” window, click “CONNECT”.
   e. Ensure that the “omnicopter” object is checked under the list of available models.
   f. Select “omnicopter” relative to “origin”.

Ubuntu/WSL

Installation

We used Windows WSL 1 and Ubuntu 20.04:

1. Install WSL 1/Ubuntu 20.04 from Microsoft Store: https://apps.microsoft.com/detail/9mttcl66cpxj?hl=en-us&gl=US

Additional Packages

Python 3

Used to send commands and/or receive data in the network.

Install using the following command in the Ubuntu terminal:

- `sudo apt install python3`

MAVSDK

High-level Python functions used to create MAVlink messages.

Install following the instructions in the Quick Start Guide.

Quick Start: https://mavsdk.mavlink.io/main/en/python/quickstart.html

API Docs: http://mavsdk-python-docs.s3-website.eu-central-1.amazonaws.com/

Asyncio

Python library used for asynchronous function calls; included natively in standard Python 3.

Reference: https://docs.python.org/3/library/asyncio.html

PyMap3d

Python library for 3D geographic coordinate conversions.

Install using the following command:

- `pip install pymap3d`

Reference: https://pypi.org/project/pymap3d/

ODROID Connection

Initial setup

1. Set up Mobile Hotspot:
   a. Either search "Mobile Hotspot" in the Windows search bar or navigate to Settings -> Network and Internet -> Mobile Hotspot.
   b. Edit the username and password to match what will be used on the ODROID.

Each run

1. Turn on hotspot:
   a. Either search "Mobile Hotspot" in the Windows search bar or navigate to Settings -> Network and Internet -> Mobile Hotspot.
   b. Toggle the switch to "ON".
2. Power on ODROID and wait for it to connect to the hotspot.
3. Send mocap data to ODROID:
   a. Open Ubuntu to the "scripts" repo directory containing "forwardmocap.py".
   b. Run the command:
      - `python3 forwardmocap.py`

   c. Once the connection is established between Motive, the ground station computer, and the ODROID, the script will begin printing position and attitude updates to the Ubuntu console, indicating that the data is being sent to the ODROID.

ODROID

Hardware requirements

1. Monitor with HDMI cable
2. USB keyboard and mouse
3. USB Wi-Fi module (ODROID brand or otherwise)
4. FTDI USB to TTL Serial Adapter

Login Info

- Admin username: odroid
- Password: odroid

Serial adapter

References

1. https://docs.px4.io/main/en/companion_computer/pixhawk_companion.html#serial-port-setup
2. Amazon link for the adapter we purchased: https://a.co/d/iM7mWNy

Initial setup

The FTDI serial adapter is used to establish a connection between the ODROID and the flight controller.

1. Follow the instructions in the PX4 docs above to map the wiring between the flight controller's TELEM2 port and the serial adapter.
2. Plug the adapter into one of the ODROID's USB ports.
3. Run the command "`dmesg`" in the ODROID terminal.
4. In the resulting output, look for the phrase "`FTDI Serial Device converter now attached to`…" and take note of the file path that follows (ours was "`ttyUSB0`", so the resulting path is "`/dev/ttyUSB0`").
5. Check to make sure the device is listed in the "`/dev/`" directory with the adapter connected.
6. The flight controller should already be configured to use MAVlink messages over TELEM2, but if not, make that change following the instructions in the PX4 docs above.

Internet Connection

Set to connect to GCS Wi-Fi hotspot automatically

1. Click the Wi-Fi symbol in the top right corner of the ODROID desktop.
2. Click "Edit Connections".
3. Double-click on the hotspot network under the list of available Wi-Fi networks.
4. Under "General", check the box for "Connect automatically with priority" and use the plus sign to set the priority to 1 or greater (gives preference to connecting to the hotspot over other known networks).
5. Under "Wi-Fi", select the appropriate option for "Band" for the device (the ODROID module uses 5 GHz) and the device under the options for "Device" (ours was wlan0).
6. Under "Wi-Fi Security", set the "Security" to "WPA & WPA2 Personal" and the "Password" to the hotspot's password if not already set.

Potential issues

Missing GUI for Wi-Fi connections

Source: https://www.cyberciti.biz/faq/change-netplan-renderer-from-networkd-to-networkmanager/

If netplan (the backend program that manages network connections) is not configured to use NetworkManager, the GUI for Wi-Fi connections will not be useful for viewing available networks and connecting automatically.

To fix this problem, NetworkManager needs to be set as the "renderer" for netplan in the config file "`/etc/netplan/01-netcfg.yaml`" following these steps:

1. Open the ODROID terminal.
2. Backup the current config file by copying the file to another location.
3. Open the config file for editing by right-clicking on the file or entering the following command with your editor of choice:

   ```
   sudo editor /etc/netplan/01-netcfg.yaml
   ```

4. Set the renderer to NetworkManager by editing the file to appear as follows (a copy is saved in the "ODROID" directory in this repo for reference):

   ```
   network:
     version: 2
     renderer: NetworkManager
   ```

5. Apply the change by rebooting the system or entering the command:

- `sudo netplan apply`

ODROID Wi-Fi 5BK Module Driver Bug

Source: https://wiki.odroid.com/odroid-h2/application_note/howto_wifi_driver_rtl8812au

If experiencing issues connecting to Wi-Fi using this USB module, try following these instructions in the ODROID terminal to install the correct driver:

1. Install DKMS:
   - `sudo apt install build-essential dkms git`
2. Clone the driver source from GitHub:
   - `git clone https://github.com/brektrou/rtl8821CU`
   - `cd rtl8821CU`
3. Add the driver to DKMS:
   - `sudo ./dkms-install.sh`
4. Enable Wi-Fi in every boot:
   a. Open usb_modeswitch rules using your editor of choice:
      - `sudo editor /lib/udev/rules.d/40-usb_modeswitch.rules`

   b. Append the following before the end line `LABEL="modeswitch_rules_end"`:
      - `# Realtek 8821CU Wifi AC USB`
      - `ATTR{idVendor}=="0bda", ATTR{idProduct}=="1a2b", RUN+="/usr/sbin/usb_modeswitch '/%k'"`

   c. Reboot the machine or reload udev rules to apply changes without rebooting:
      - `sudo udevadm control -reload-rules && sudo udevadm trigger`

   d. NOTE: If you lost the Wi-Fi module 5B after rebooting your system, you can re-enable that by using the following command. To do this automatically, there're so many ways to do that such as using systemd, an init script, or a Python script:
      - `sudo udevadm control trigger`

MAVlink-Router

References

Source: https://github.com/mavlink-router/mavlink-router

Other references:

1. https://ardupilot.org/dev/docs/raspberry-pi-via-mavlink.html

Multiple MAVlink pathway setup

The Python script "mavserver.py" in the "scripts" directory establishes 3 "paths" between the GCS and the flight controller using specific port numbers:

1. Port 14550: Used by QGroundControl to connect to the flight controller.
2. Port 14560: Used by the "forwardmocap.py" script to send constant motion capture data to the flight controller.
3. Port 14570: Used by the user to send MAVlink commands via Python scripts.

The script can be run on the ODROID by copying it to the ODROID's desktop and running the following command in the terminal:

- `/usr/bin/python3 /home/odroid/Desktop/mavserver.py`

Initial setup

1. Install mavlink-router:

   Run each of these commands in the ODROID's terminal:

   ```
   1. cd
   2. git clone https://github.com/mavlink-router/mavlink-router.git
      ./mavlink-router
   3. cd mavlink-router
   4. git submodule update --init --recursive
   5. sudo apt install git ninja-build pkg-config gcc g++ systemd
   6. sudo pip3 install meson
   7. /usr/local/bin/meson setup build .
   8. ninja -C build
   9. sudo ninja -C build install
   ```

2. Connect to the hotspot if not already connected.
3. Copy/recreate the "mavserver.py" script to the ODROID desktop.
4. Edit the mavserver file so that the string variable `GCS_IP` equals the GCS hotspot IP address and the string variable `FC_path` equals the path of the FTDI serial adapter.

5. Set mavserver as executable using the command:

   - `chmod a+rx ~/Desktop/mavserver.py`

6. Set mavserver to run at device startup:

   The ODROID was configured to automatically run mavserver at startup by following these steps:

   1. Copy/recreate the “StartUpD.service” file located in the “ODROID” directory in the repo to the “`/etc/systemd/system`” directory on the ODROID.

   2. Run the following command in the terminal to enable the service file:

      - `systemctl enable StartUpD.service`

   NOTE: The .service file waits for an internet connection before running, so ensure that the mobile hotspot is turned on when trying to run the script at startup.

   Reference: https://unix.stackexchange.com/questions/47695/how-to-write-startup-script-for-systemd

Future work

1. A config file should be created following the instructions on the mavlink-router GitHub, using the default location “`/etc/mavlink-router/main.conf`” or otherwise, and set to keep logs of all MAVlink messages for post-run analysis.