Alex Simpkins, Ph.D.
Prepared for RIIS LLC
3/21/2022

Basic install instructions for gazebo or Jmavsim on Linux or Mac and connecting the QGC on Herelink together

1 - Installing the simulator

There is a new shell script that does most of the setup, but it may not be perfect. Follow the most up to date instructions to the letter. If the script completes, and you follow the instructions to the point that the simulation runs, you can ***skip to part 2*** of this document.

Follow the installation instructions in the px4 documentation here carefully but do not ignore the part about the shell profile before installing for the Mac, it is critical as it sets up a few details on the path. Also as you go take note of any messages from home brew - for example of it says something is not in the path add to the ~/.zshrc file then call

`source ~/.zshrc` to reload the paths

In summary for the shell:

`nano ~/.zshrc`
that opens it. Copy these things in, don't replace anything in there
save using control-O and then get out with ctrl-x
then type **`source ~/.zshrc`**

Read each page completely before doing the installations, there is important information and some relevant parts are not in sequence

Mac: https://docs.px4.io/master/en/dev_setup/dev_env_mac.html (requires osx 10.15.7 or later)

Additional paths to add are included below to paste into the ~/.zshrc file

Linux: basically run the shell script it's fairly automated, https://docs.px4.io/master/en/dev_setup/dev_env_linux_ubuntu.html (ignore the nuttx installation, just run the ubuntu.sh script)

Next move to build instructions:
https://docs.px4.io/master/en/dev_setup/building_px4.html

It is recommended on either system after install before the px4 code is installed to run

```
Git submodule update –recursive
Make distclean
```

commands in the command line.

below is a starting point for a .zshrc setup file:

```
#start of file————————————————————————————————-
ulimit -S -n 2048

# Point pip3 to MacOS system python 3 pip
alias pip3=/usr/bin/pip3

export PATH="/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin"

export PATH="$PATH:"/Users/<yourusername>/Library/Python/3.8/bin""

export PATH="$PATH:"/Users/<yourusername>/Library/Python/3.8/lib/
python/site-packages""

export PATH="$PATH:"/usr/local/Cellar/bullet/3.21/lib""

export PATH="$PATH:"/usr/local/Cellar""

export PKG_CONFIG_PATH="/usr/local/lib/pkgconfig"

export PATH=“$PATH:"/Users/<yourusername>/Library/Python/3.9/bin""

export PATH=$PATH:/Library/Frameworks/GStreamer.framework/Commands

#end of file————————————————————————————————-
```

At the end of the setup process try running

```
make px4_sitl jmavsim
```

when the instruction say, and if all is well a window will open with a simulated drone in a basic environment

or for gazebo

```
make px4_sitl gazebo
```

2 - Setting up mavlink to broadcast

If in a recent install of Linux, the driver for an FTDI usb-serial adapter is included and no other installation is needed. If on mac you will need to install the most recent VCP driver from www.ftdi.com

Then on Mac or Linux find the driver identifying name, we will need that. Type

```
ls /dev
```

and locate the item that says something along the lines of tty.usb0 on linux or on mac it will be a bit longer, ending in a combination of letters and numbers. Copy that to a text document or write it down, be sure of the precise spelling and capitalization.

Now cd to the following directory:

/Users/<your user name>/Documents/riis/code/PX4-autopilot/build/px4_sitl_default/etc/init.d-posix

nano to open the file,

```
nano init.d-posix
```

then comment out this line with a #

Works over wifi no problem after

```
# API/Offboard link - comment this out
#mavlink start -x -u $udp_offboard_port_local -r 4000000 -f -m onboard -o $udp_offboard_port_remote

#use this one to broadcast over the network - useful for #debugging to desktop QGC or if serial connection not working
mavlink start -x -u $udp_offboard_port_local -r 4000000 -f -p

#serial broadcast - 57600 is the standard telemetry rate, insert your serial device id instead of  tty.usb0 if different
mavlink start -d /dev/tty.usb0 -b 57600 -f -p -Z -m onboard
```

then type control-o to save and press return, then control-x and return to quit. cd back to your main px4-autopilot directory (cd ../ until you get there)

Now when you launch the simulator (either) mavlink will broadcast, allowing QGC on other computers on your wifi network to automatically connect, including the Herelink. It will also connect to a serial connection and send telemetry through there as well.

3-Herelink

WiFi method: Finally we must set up your wifi network and the herelink to connect, and install the custom .apk file.

Your wifi network must be altered - it must be a 5G network, and you have to fix the channel to either 20, 40, or 80. This is a herelink requirement.

Set the herelink to connect to the same wifi the computer is on running the simulation.

Serial method: Now make sure the Herelink is paired with its receiver. Connect the receiver to the usb-serial adapter with a custom wire that has the following connections.

Pin1 - black (GND) Herelink ——>	P1 - GND
Pin2 - Yellow (Tx) Herelink ——>	P5 - RX FTDI
Pin3 - Red (Rx) Herelink ———>	P4 - TX FTDI

Set the herelink to enable usb debugging and install the app according to

https://docs.cubepilot.org/user-guides/herelink/herelink-user-guides/installing-a-custom-app

The old app is called **`org.mavlink.qgroundcontrolcustom`** - you will need this to remove an older version if it exists from prior installations

4 - Test!

So now the system should all work if all goes well. Launch the custom app in the herelink, then open the simulator using the make command

```
make px4_sitl gazebo
```

or

```
make px4_sitl jmavsim
```

If you have errors, check the px4 dev documentation for reference and at this point a possible google search, or we can provide support.

The hardware sticks are not enabled for this simulation at this time. Use the virtual joystick, necessary for orbit function anyway.

If you want to include the hardware sticks it requires a second serial connection and connecting the RC output from the hireling's SBUS1. This is untested but documented at the px4 developer site. It may need to be inverted with a three piece circuit.