Sam Church
RIIS, LLC
April 21, 2022

Yaw Offsets in Orbit Flight Mode - QGroundControl (Windows)

A how-to guide with code samples on configuring and building your own custom QGroundControl application. We'll be modifying the existing yaw control to allow a pilot to input a yaw offset (in degrees) while in orbit flight mode.

Note if you are looking for a version on Herelink, see the separate documentation labeled "Yaw Offsets in Orbit Flight Mode - QGroundControl (Herelink)"

Environment Set-Up

If you already have the required software installed, double-check that you have the correct versions. Then you may continue to the section titled 'Configuring the Project.'

Required Software

- Microsoft Visual Studio 2019
- Qt/Qt Creator
- PX4 Toolchain

Installing Visual Studio 2019

1. Follow this link to download the last version of Visual Studio 2019:
    a. https://my.visualstudio.com/Downloads?q=visual%20studio%202019&wt.mc_id=o~msft~vscom~older-downloads
2. Once downloaded, run the installer and check the box labeled, "Desktop development with C++"
3. Finish installing by checking the Install box

Installing Qt Creator

1. Follow this link, scroll down to the section titled "Looking for Qt binaries?"
    a. https://www.qt.io/download-open-source
2. Click the button labeled "Download the Qt Online Installer"
3. A page will show up with a green button labeled "Download". Click it and the QT Online Installer will download
4. After downloading, launch the installer
5. Log in / Create an account for Qt
6. Check the box claiming you are an individual
7. Hit next until you reach installation folder

8. Check the box labeled “Custom Installation” and hit next
9. Select the dropdown for Qt and check the box labeled “5.15.2”
10. Finish installation by hitting next until it is complete

Installing PX4 Toolchain

1. Follow this guide for the latest installation tutorial:
    a. https://docs.px4.io/v1.12/en/dev_setup/dev_env_windows_cygwin.html
    b. Make sure to check the box that clones the PX4 repository at the end of the installer

Configuring the Project

1. Clone the QGroundControl repository found here:
    a. https://github.com/mavlink/qgroundcontrol
2. Open Qt Creator
3. Select “File” and then “Open File or Project...”
4. Navigate to the repository
5. Select the file “qgroundcontrol.pro”
6. Select “Configure Project”
7. In the terminal window navigate to the [QGroundControl Repository]/custom
8. Update sub dependencies with this command:
    a. git submodule update --recursive
9. Restart Qt Creator
10. The project will now be able to build without any modifications

Build & Deployment

1. In Qt Creator, select the hammer button in the bottom right corner to build the project
2. Select the green run button to run the project. The default configuration will be on Windows
    a. It will create a folder outside of the project folder (default is [user]/Documents) containing the flight logs, missions, and other miscellaneous items
    b. It will also create a build folder next to the project folder containing a staging folder with an executable. This executable will run the whole application on a Windows device. The staging folder can be dragged and dropped anywhere and the executable will run

Sam Church
RIIS, LLC
April 21, 2022

Development

All the source code used can be found at https://github.com/riis/herelink/

Creating Custom Input Sliders

There are 2 input sliders that we will be creating; A horizontal slider to control the yaw offset (in degrees), and the vertical slider which controls altitude.

1. Create 2 new files under src/QmlControls:
    a. QGCVerticalSlider.qml
    b. QGCHorizontalSlider.qml
    c. Copy the sliders from the riis/herelink repository to QGCVerticalSlider.qml and QGCHorizontalSlider.qml, respectively
2. In custom/qgroundcontrol.qrc, add these lines after the line where it defines QGCSlider.qml like so:

```
<file
alias="QGroundControl/Controls/QGCSlider.qml">../src/QmlControls/QGCSlider.qml</fil
e>
<file
alias="QGroundControl/Controls/QGCHorizontalSlider.qml">../src/QmlControls/QGCHoriz
ontalSlider.qml</file>
<file
alias="QGroundControl/Controls/QGCVerticalSlider.qml">../src/QmlControls/QGCVertica
lSlider.qml</file>
```

3. In qgroundcontrol.qrc follow the same as step 3.
4. In src/QmlControls/QGroundControl/Controls/qmldir, add these two lines below QGCSlider like so:

```
QGCSlider                               1.0 QGCSlider.qml
QGCHorizontalSlider                     1.0 QGCHorizontalSlider.qml
QGCVerticalSlider                       1.0 QGCVerticalSlider.qml
```

5. Copy the VirtualJoystick from the riis/herelink repository to src/FlightDisplay/VirtualJoystick.qml
6. Navigate to custom/updateqrc.py and execute the file in a command prompt

Modifications the User Interface

Making small changes to the existing user interface for quality of life improvements.

1. Copy the FlyViewMap from the riis/herelink repository to src/FlightDisplay/FlyViewMap.qml

2. Copy the QGCMapCircleVisuals from the riis/herelink repository to src/MissionManager/QGCMapCircleVisuals.qml
3. Copy the MissionItemIndexLabel from the riis/herelink repository to src/QmlControls/MissionItemIndexLabel.qml

Testing Using a Simulator

1. After setting up the PX4 Toolchain, in command prompt, navigate to the PX4 folder created from the PX4 Toolchain installer
2. Once there, run this command:

```
$ ./run-console.bat
```

3. Once running, navigate to where the PX4-Autopilot was cloned to from the installer and run this command:

```
$ make px4_sitl jmavsim
```

4. Open QGroundControl after building the application and the simulator should now be connected to QGroundControl