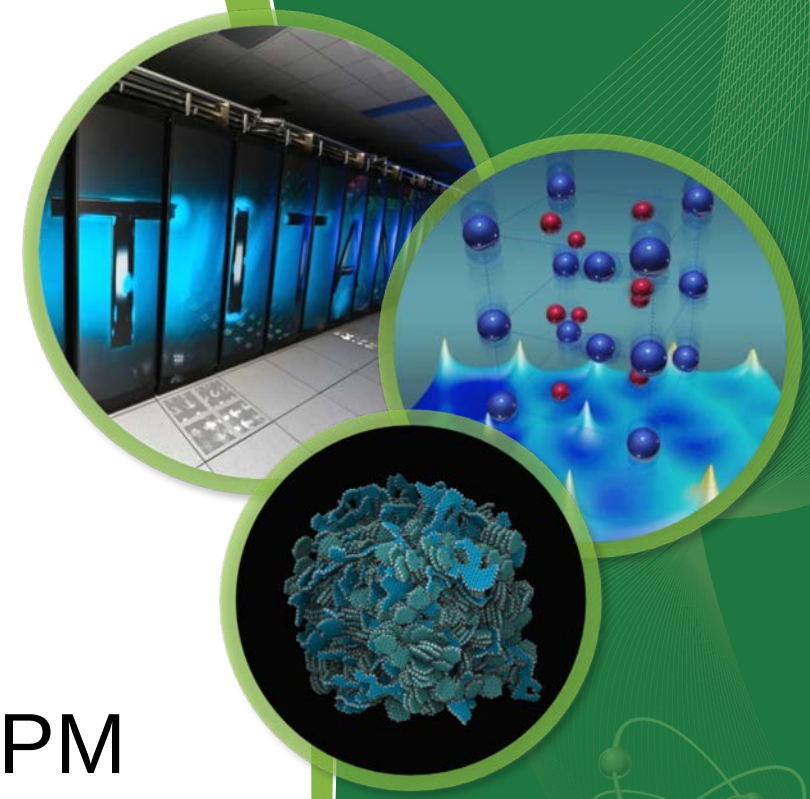
DARPA Bay Area SDR Hackfest Workshop



Toby Flynn

Dr Thomas Rondeau DARPA-MTO PM

July 27, 2017

Agenda

- **Introductions**
- Overview of Hackfest Hardware/Software
  - Installation of required software
- Overview of new gr-uaslink software
- Example use of gr-uaslink with a SITL system
- Example use of gr-uaslink with a serial connected pixHawk 2 controller
- Example OTA test of gr-uaslink to a 3DR-solo
- Topics of interest for the Hacker Space Challenges

Introductions

- Dr Tom Rondeau, DARPA-MTO PM
- Toby Flynn, Senior R&D Staff ORNL

Equipment Needed for best results of the workshop

- Linux based laptop with ability to install software
- GNU Radio installed using PyBOMBS or installation method of choice
  - Must be able to build and install missing OOT modules
- Advanced testing will require 2 SDR units or someone to work with for testing OTA

Planned Results of Workshop

- Leave with a system able to run gr-uaslink with a software-in-the-loop copter emulator
- Improve familiarity with the software which controls UAS systems
- Gain some insights from lessons learned
- See the system operate with a tethered UAS

Agenda

- Introductions
- **Overview of Hackfest Hardware/Software**
- Overview of new gr-uaslink software
- Example use of gr-uaslink with a SITL system
- Example use of gr-uaslink with a serial connected pixHawk 2 controller
- Example OTA test of gr-uaslink to a 3DR-solo
- Topics of interest for the Hacker Space and Challenges

Software

- GNU Radio Latest 3.7 release
  - https://www.gnuradio.org
- Latest UHD stable release
  - https://kb.ettus.com/UHD
- Multiple OOT GNU Radio packages including
  - http://www.cgran.org
  - gr-evenstream, gr-mapper, gr-burst
    - gr-burst requires modifications to remove dependencies on scipy for RPI3 installation

Software Continued

- PyMAVLink to translate MAVLink messages to buffers
  - https://github.com/ArduPilot/pymavlink
  - https://github.com/ArduPilot/pymavlink.git (for looking up commands)
- OpenEmbedded based Linux image for the Raspberry Pi-3
  - 64bit support
  - Cross-compiling SDK
  - meta-hackfest

Optional Software

- MavProxy
  - Will be installed for testing
- Host based ArduPilot for software in the loop testing
- Other Ground control station software
  - QGroundControl for example

MAVProxy

- Install PyMAVLink dependencies
  - sudo apt-get install libxml2-dev libxslt-dev python-dev (Ubuntu)
  - sudo dnf install libxml2-devel libxslt-devel python-devel dnf install redhat-rpm-config (Fedora)
- MavProxy and PyMAVLink can be installed at one time to reduce dependency issues
  - sudo pip install mavproxy

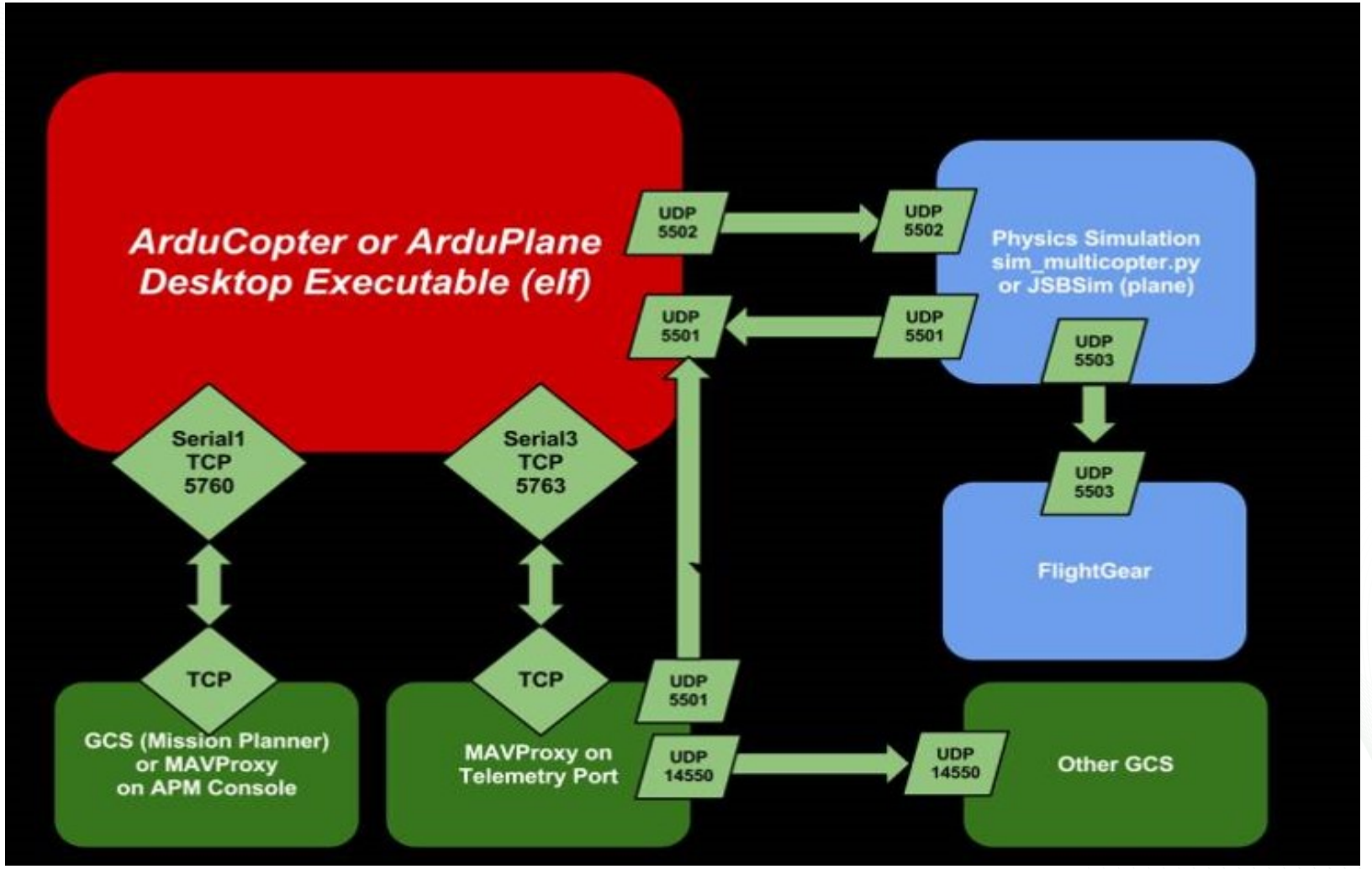
Modes of flight for UAS systems

- Major modes of operation
  - Guided (requires GPS)
  - Loiter (requires GPS or optical flow sensor)
  - ALT_HOLD (stable altitude)
  - STABILIZE (Just enough control to maintain flight, different motor settings from other modes)

AdruPilot for Software in the Loop Testing

- Online step-step documentation is available at http://ardupilot.org/dev/docs/setting-up-sitl-on-linux.html#setting-up-sitl-on-linux

ArduPilot SITL data flow



http://ardupilot.org/dev/docs/sitl-simulator-software-in-the-loop.html

Test MAVProxy with SITL

- Follow instructions on previous link to make sure MAVProxy and Ardupilot SITL are working correctly
- Test with mode alt_hold ,arm throttle, rc 3 1700, other rc commands

Install GNU Radio and Needed OOT modules

- Only required if GNU Radio is not installed on your computer
- Recommend installation using PyBOMBS
  - https://github.com/gnuradio/pybombs
  - Follow instructions for installation of GNU Radio
- www.cgran.org provides information on OOT modules
  - gr-eventstream
  - gr-mapper
  - gr-burst

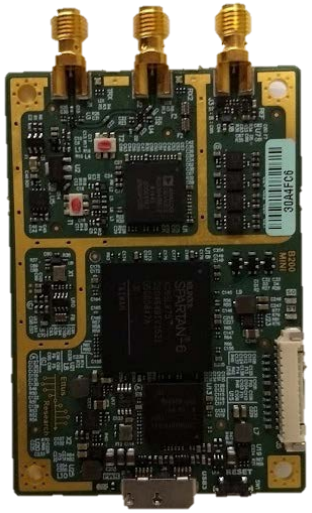
UAS Hardware

- 3DR Solo
- 2 TurboAce Matrix-S available

- Pixhawk-2 Controller
  - ArduPilot (open-source flight control software for Matrix-S)
  - OpenSolo (open-source flight control software fro 3DR Solo)

- Backup Controller
  - (to take over flight controls to avoid crashes)

SDR Hardware

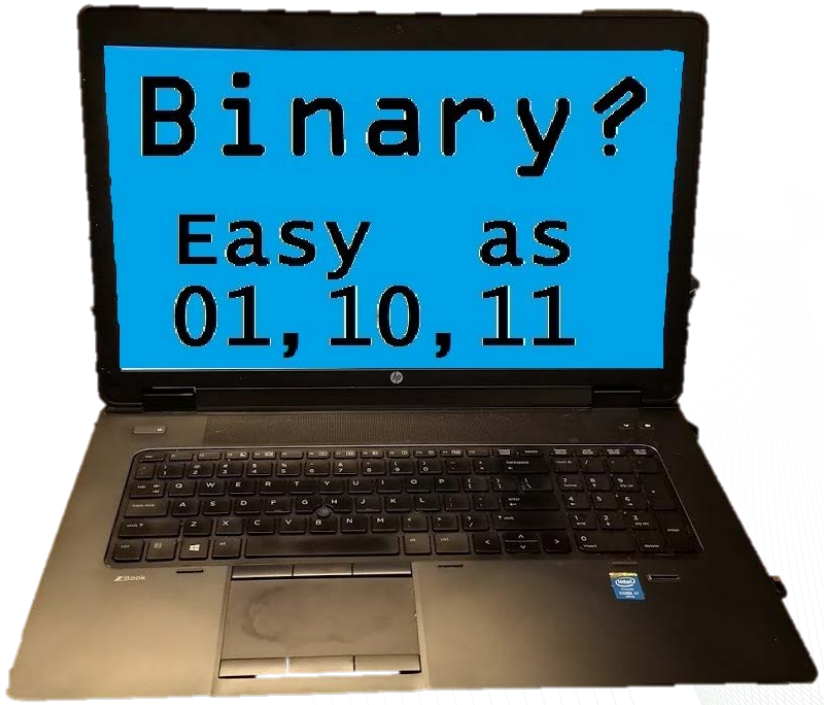


- For the UAS
  - Raspberry Pi-3 with a connected Ettus USRP B200-mini
  - Lightweight omnidirectional antenna
  - Powered by the system battery

- Ground Control
  - Ettus USRP B210
  - Four (4) omni directional antennas to provide MIMO options

Control Hardware

- HP ZBook Mobile Workstation
  - Ubuntu 16.04.2 LTS Operating System
  - 16 GB Memory
  - 512GB SSD Storage
  - Intel 4 core i7-7700 HQ processor
  - Intel Dual Band Wireless AC 8265
  - 17” Display



Agenda

- Introductions
- Overview of Hackfest Hardware/Software
- **Overview of new gr-uaslink software**
- Example use of gr-uaslink with a SITL system
- Example use of gr-uaslink with a serial connected pixHawk 2 controller
- Example OTA test of gr-uaslink to a 3DR-solo
- Topics of interest for the Hacker Space and Challenges

Overview of New GR-UASLink Software

- 7 New blocks within the OOT module
  - All written in Python
  - External dependencies on pymavlink
  - All blocks use Async messages
- 1 External Python Application which originates with GRC
  - Used to generate messages from a GUI controller
  - Messages generation logic added to GRC generated python
- Using multiple methods answers a common GNU Radio question, “How to work with a real system?”
  - Serial device interface
  - UDP is used for Data passing
  - ZMQ Messages allow another path of external data inputs

Overview of Software Continued

- The apps directory contains the grc files developed as part of testing
  - All files have key words to help understand what they are
    - uhd_ means the file uses a SDR for communication
    - _sitl means the system is designed to use the software in the loop test system
    - _serial means the system is configured to use a serial connection
    - test_ is used to test the blocks. Some of these require SITL
    - packet_ uses the packet processing which is part of GNU Radio
    - psk_ uses gr-eventstream, gr-mapper, and gr-burst for packet communications

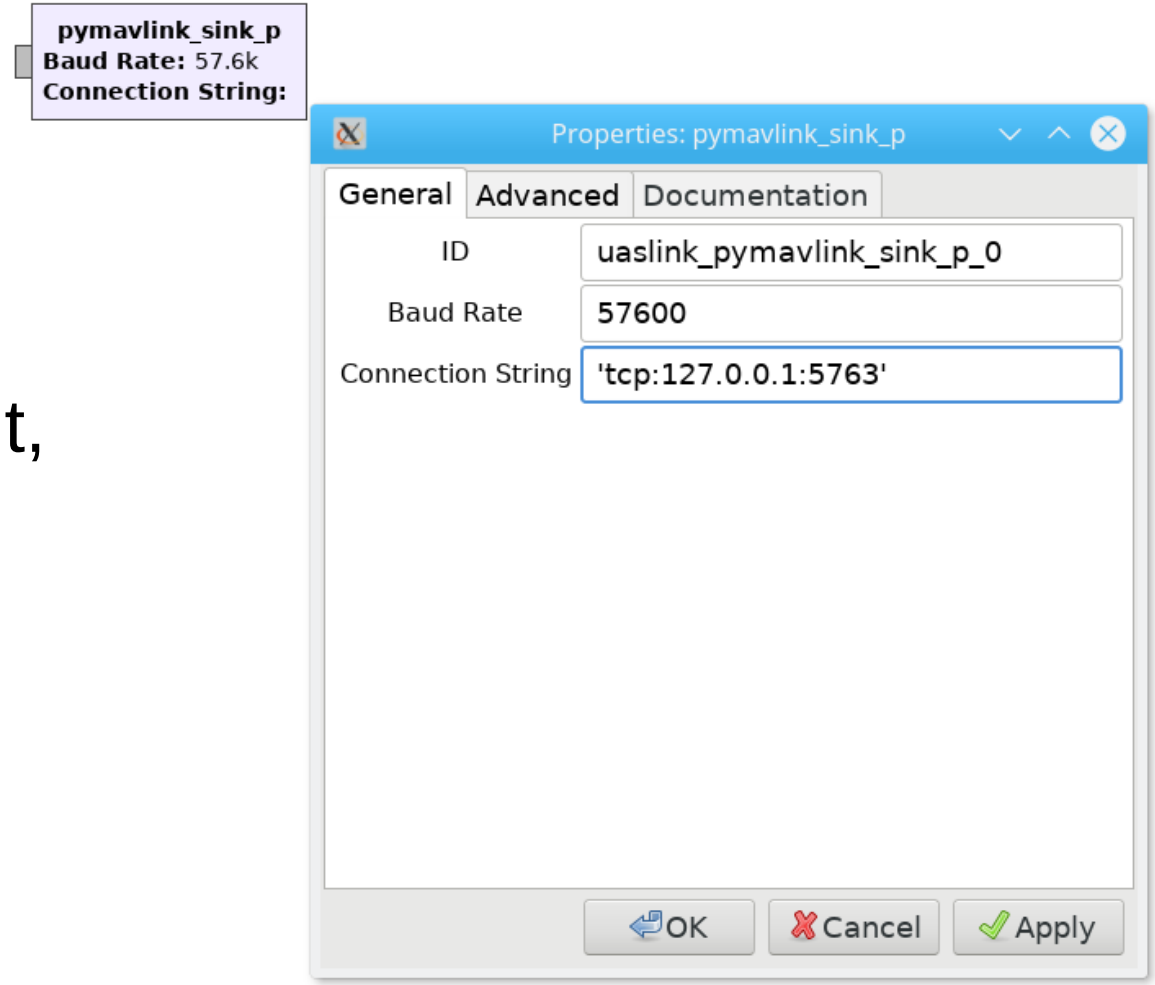
Control GUI

- Main GUI for control of the UAS
- When a button is pressed the motor commands are updated to generate the desired results
- When a button is released default motor commands are sent
- This is located in the apps directory and called: control_gui_override_control.py

Control GUI Continued

- ‘Take Off’ is used for take-off
  - If takeoff fails, ‘Land’ must be sent before ‘Take Off’ Will operate again
- ‘speed’ is the amount the motors will be increased or decreased for direction control
- ‘RC Steady Engine Power’ is the default speed of the motors
- ‘mode’ can currently be either ALT_HOLD or STABLIZE
  - Preferred flight mode is ALT_HOLD

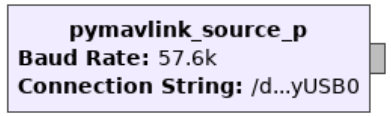
PyMAVLink sink

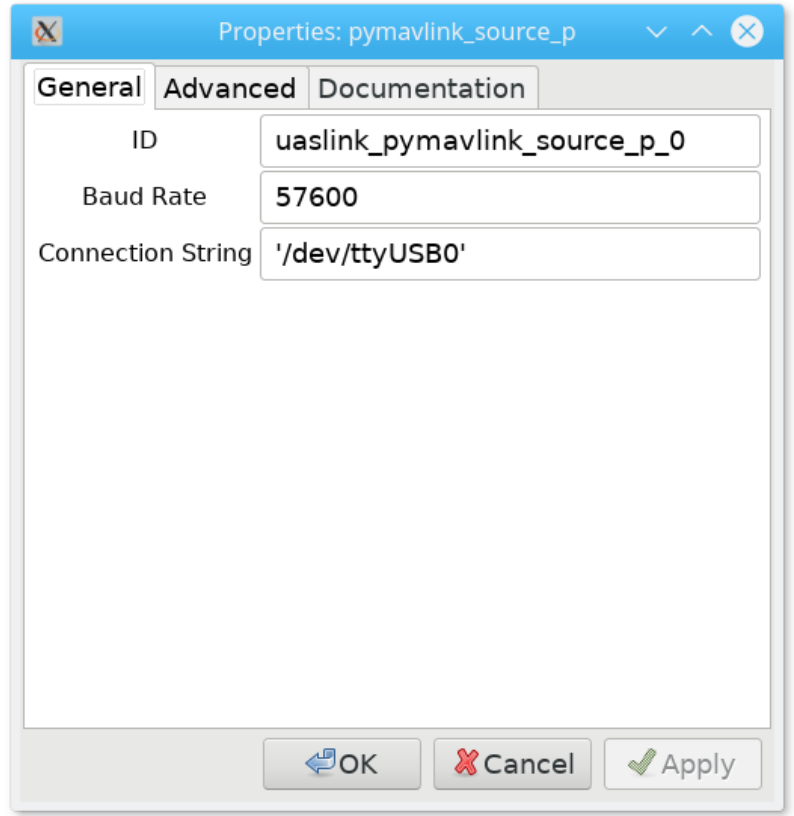


- Simple test case as a sink for MAVLink Messages
- Connection String can be tcp, udpout, or serial string
- Only useful for passing a MAVLink Message from a source to a device
  - Breaks the required 2-way communication of MAVLINK
  - Works well in SITL test cases

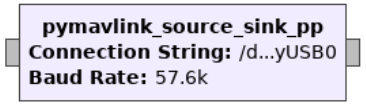
PyMAVLink Source

- Simple test case as a source for MAVLink Messages
- Connection String can be tcp, udpin, or serial string
- Only useful for passing a MAVLink Message from a device to the system
  - Breaks the required 2-way communication of MAVLINK

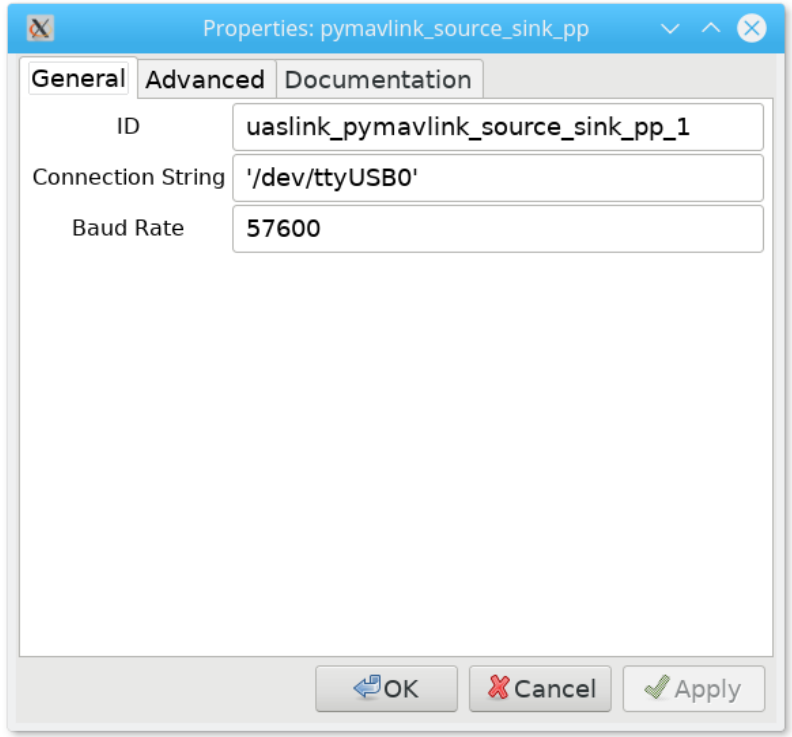




PyMAVLink Source Sink



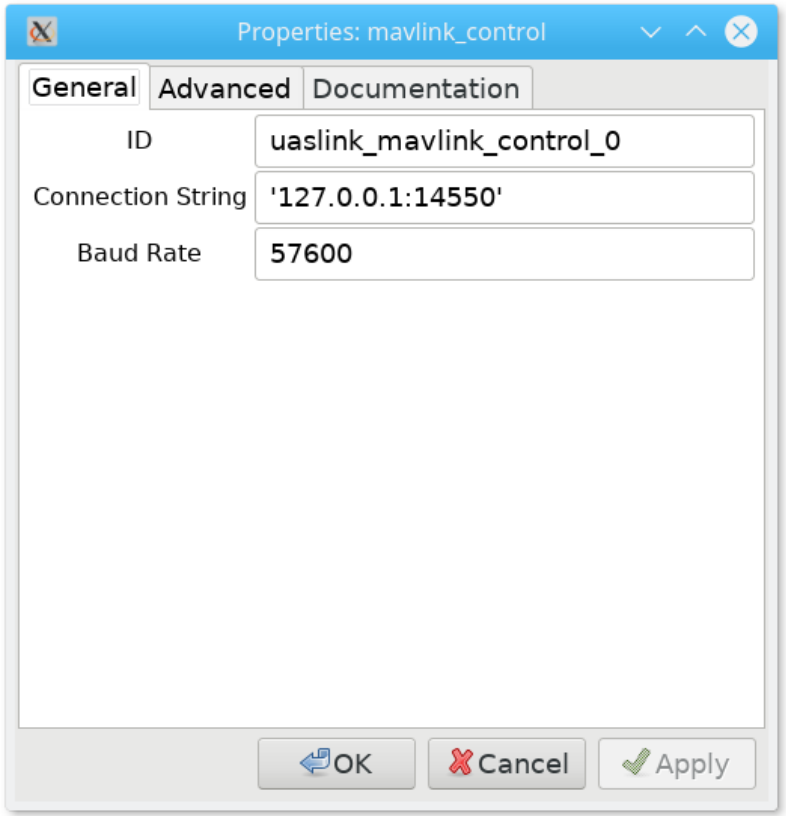
- Acts as a GNURadio async message source and sink
- Sends MAVLink commands to a device and device MAVLink messages into GNURadio
- Should always run on the computer connected to the UAS (normally the RPi3)



OAK RIDGE National Laboratory

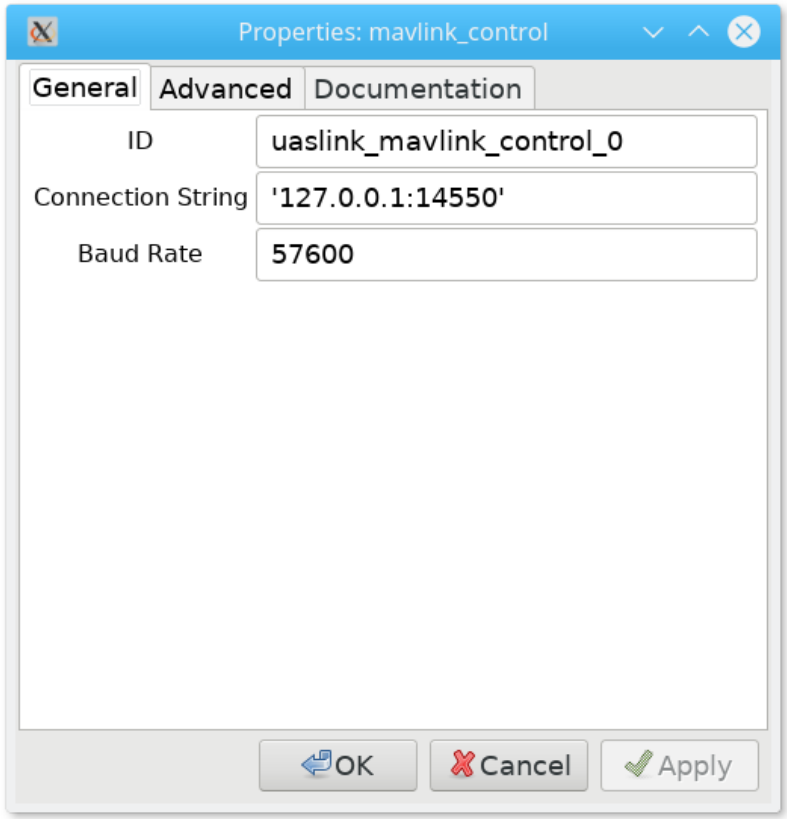
MAVLink Controller

- Creates MAVLink Messages from the control messages
- Generates heartbeat and RC override commands every second
- Processes incoming messages from the PMT Async messages into MAVLink messages to pass to an internal PyMAVLink state



MAVLink Controller Continued

- Passes data from PMT to UDP internal to the block (PyMAVLink maintains state and requires messages to be sent to them.)
- Can be configured to interface with MAVProxy
- Connection String should always be a UDP address, but without udpin or udpout



Burst Verification

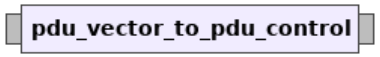
- Used to verify a received burst is valid at a limited level
- Verifies that part of the data contains data within expectations
- May require modification if data vectors and information passed changes
- Has the same functionality as PDU vector to PDU Control

PDU Control to PDU Vector

- Transfers Control messages to vectors for burst transmission
- Currently uses a lookup table to transform some meta data to numerical values
- Output is a vector of bytes

PDU Vector to PDU Control

- Transfers vector messages to control messages after receiving a burst transmission
- Currently uses a lookup table to transform some numerical values to meta data
- Output data array and meta data



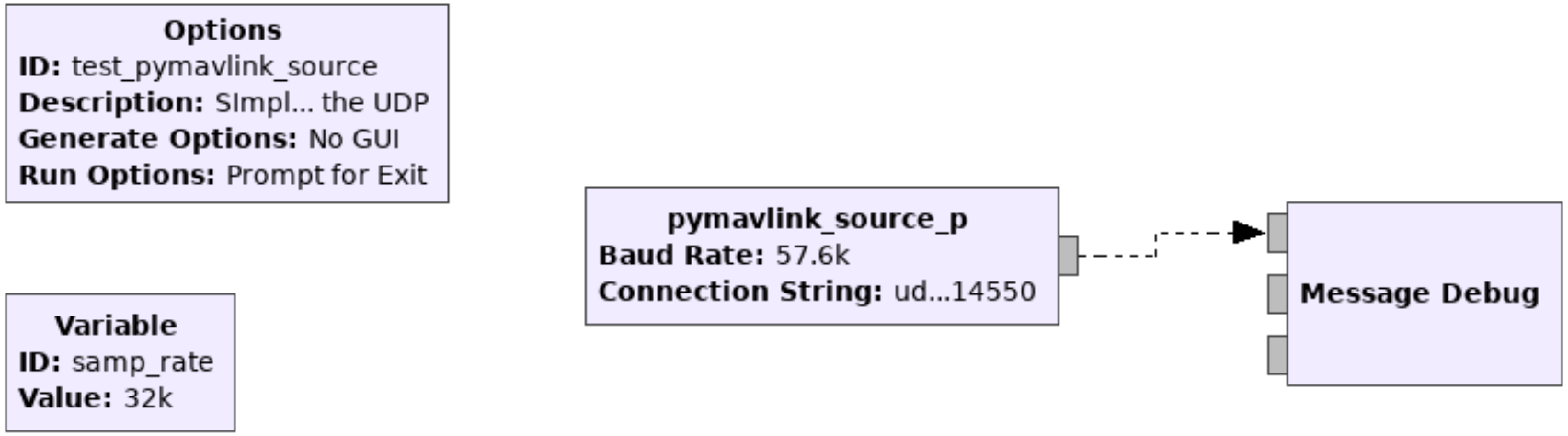
Install gr-uaslink

- gr-uaslink
- git clone git://github.com/deptofdefense/gr-uaslink
- cd gr-uaslink
- mkdir build
- cd build
- cmake ../
- make
- make install

Agenda

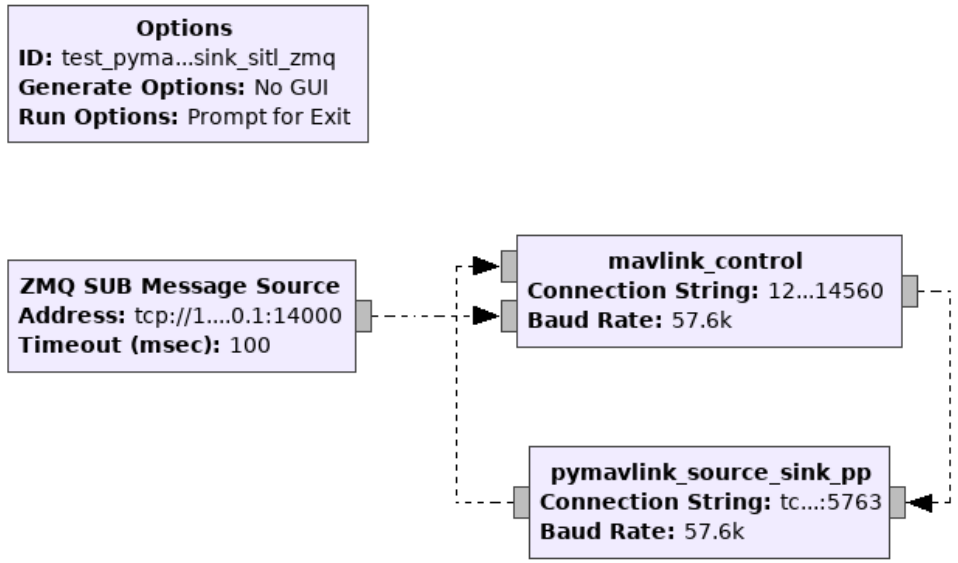
- Introductions
- Overview of Hackfest Hardware/Software
- Overview of new gr-uaslink software
- **Example use of gr-uaslink with a SITL system**
- Example use of gr-uaslink with a serial connected pixHawk 2 controller
- Example OTA test of gr-uaslink to a 3DR-solo
- Topics of interest for the Hacker Space and Challenges

Test pymavink source



- Very simple test case to verify the ability to read in MAVLink messages
- Can be changed to read from multiple sources

Test PyMAVLink Source Sink SITL

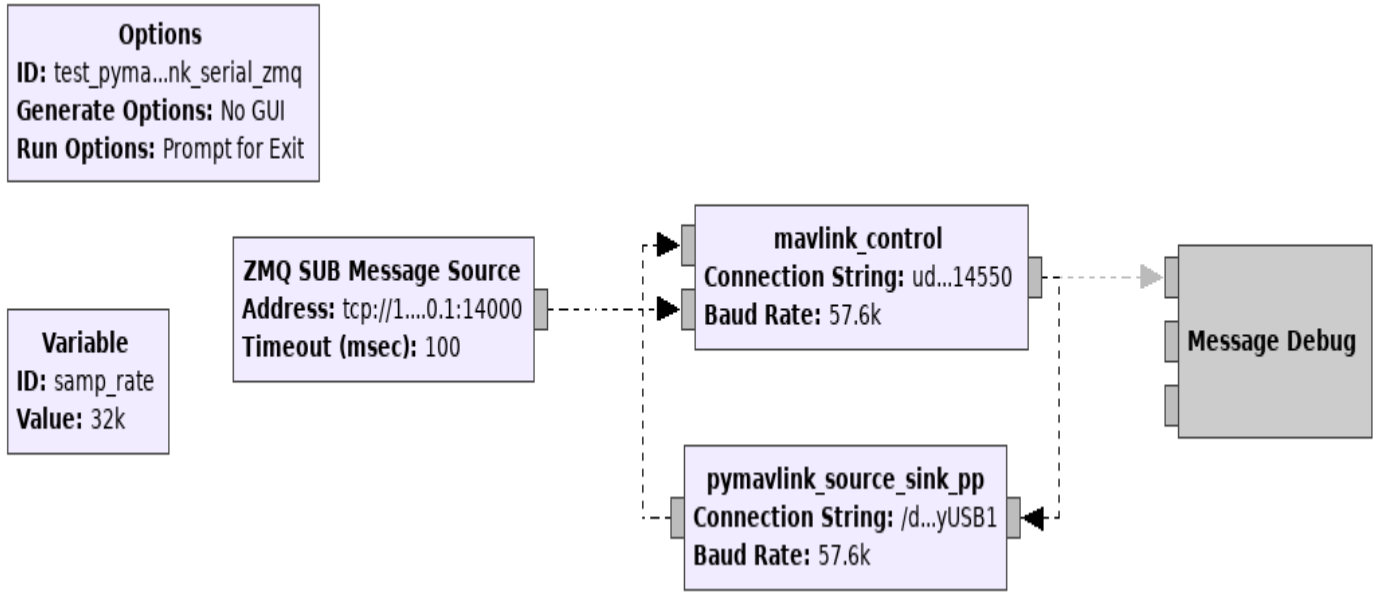


- Very simple test case to verify the ability to read in MAVLink messages
- Can be changed to read from multiple sources
- Can be configured to work with MAVProxy

Agenda

- Introductions
- Overview of Hackfest Hardware/Software
- Overview of new gr-uaslink software
- Example use of gr-uaslink with a SITL system
- **Example use of gr-uaslink with a serial connected pixHawk 2 controller**
- Example OTA test of gr-uaslink to a 3DR-solo
- Topics of interest for the Hacker Space and Challenges

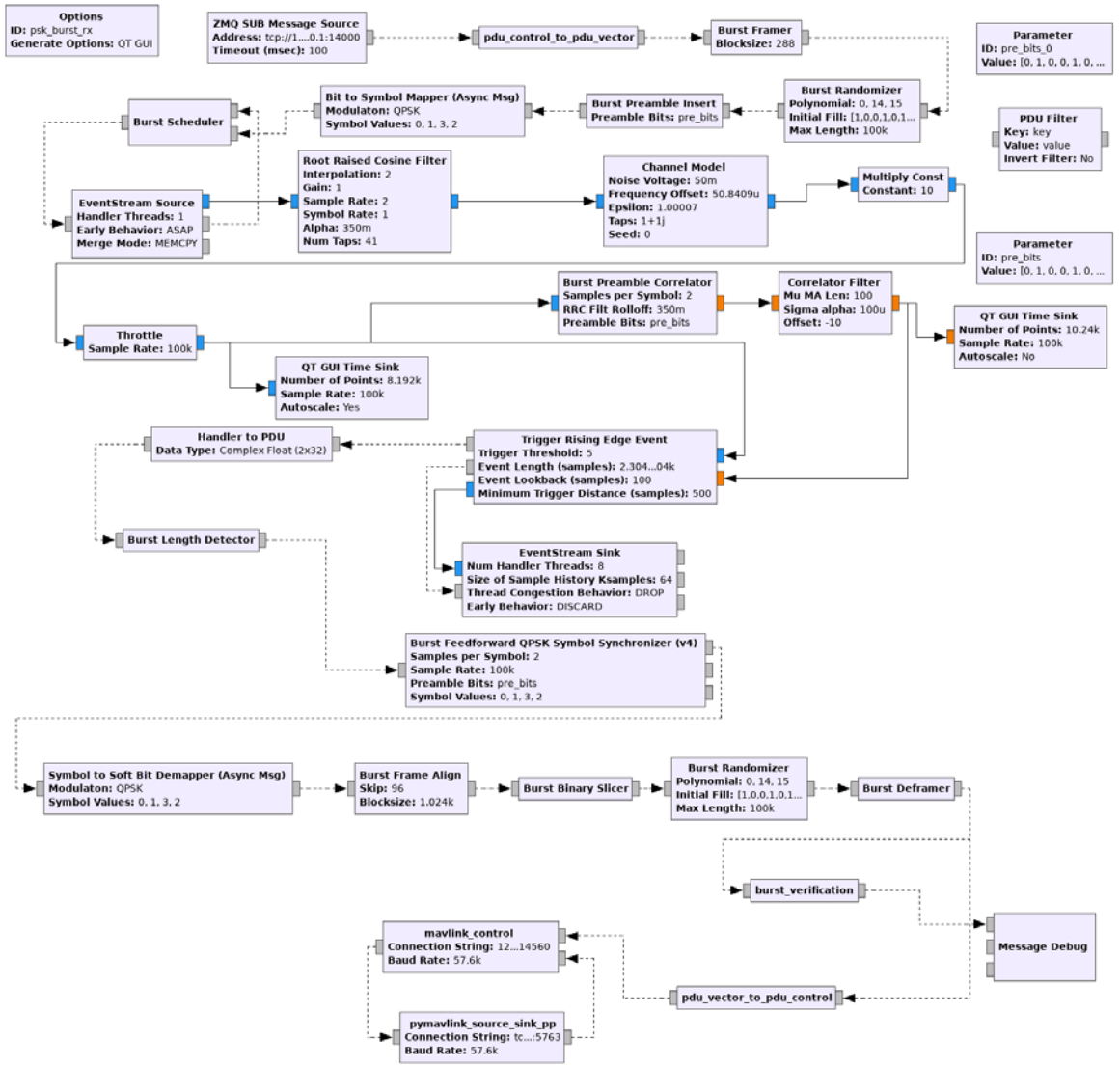
Test PyMAVLink Serial



- Serial connections are used to connect to physical hardware
  - Can be a cable connection to a UAS
  - Can be a radio connection using a USB/Serial radio
  - Can be a USB/Serial connection to a standalone flight controller

PSK Burst

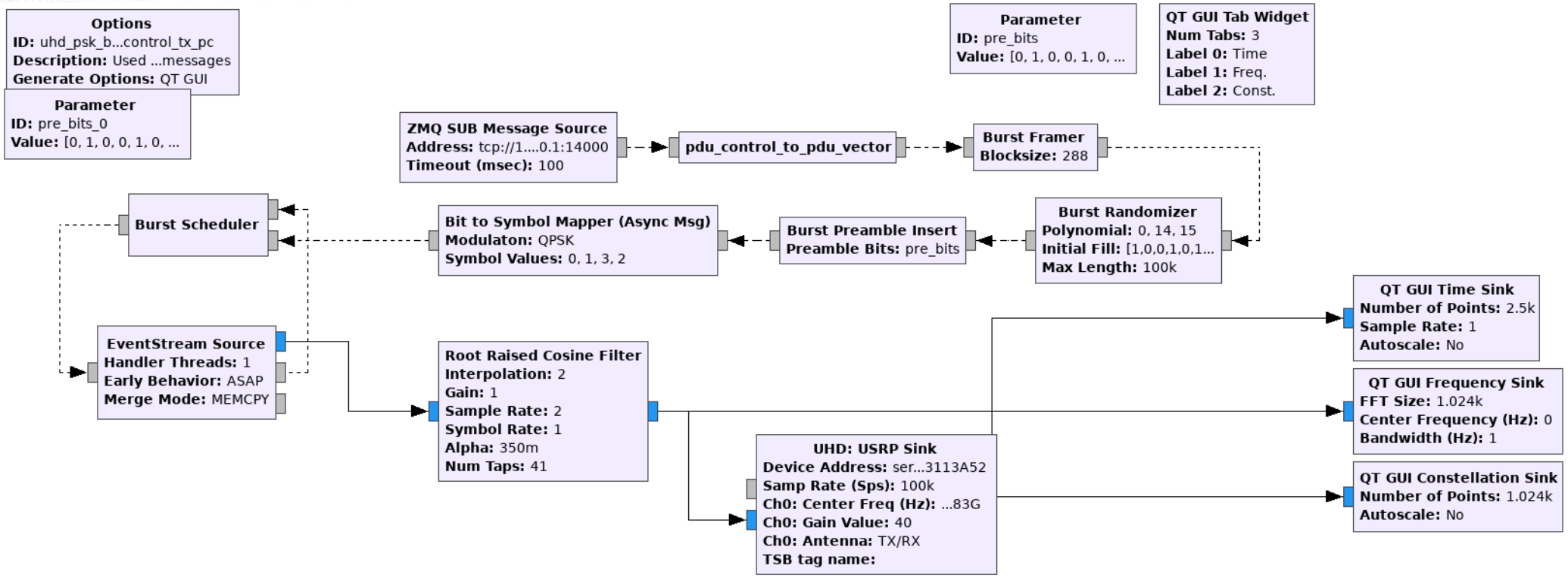
- Only sends simple control messages
- Uses several OOT modules



Agenda

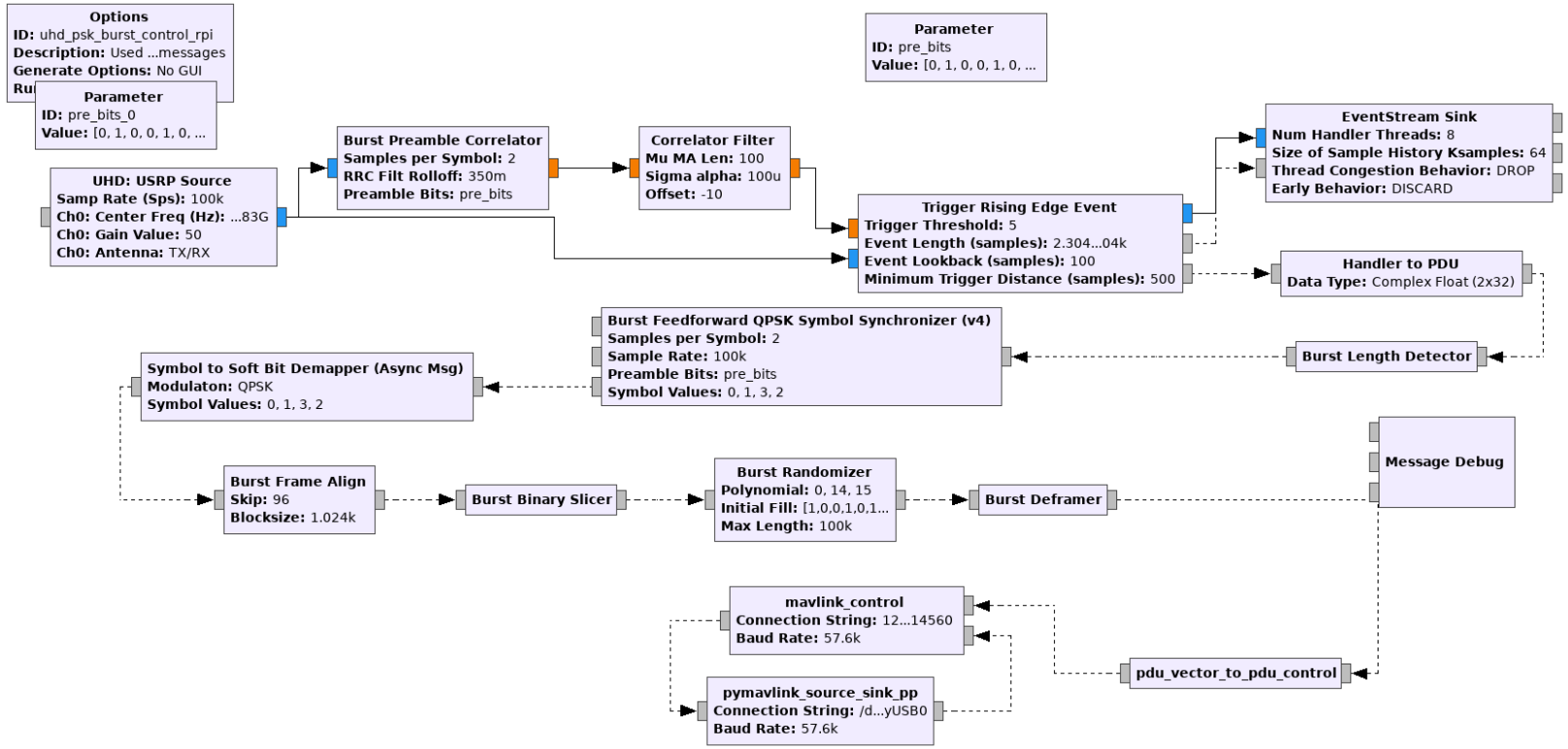
- Introductions
- Overview of Hackfest Hardware/Software
- Overview of new gr-uaslink software
- Example use of gr-uaslink with a SITL system
- Example use of gr-uaslink with a serial connected pixHawk 2 controller
- **Example OTA test of gr-uaslink to a 3DR-solo**
- Topics of interest for the Hacker Space and Challenges

UHD PSK Burst TX



- Uses Ettus Research Hardware to transmit the Burst data
- Should be ran on a PC with a GUI display

UHD Burst PSK RX

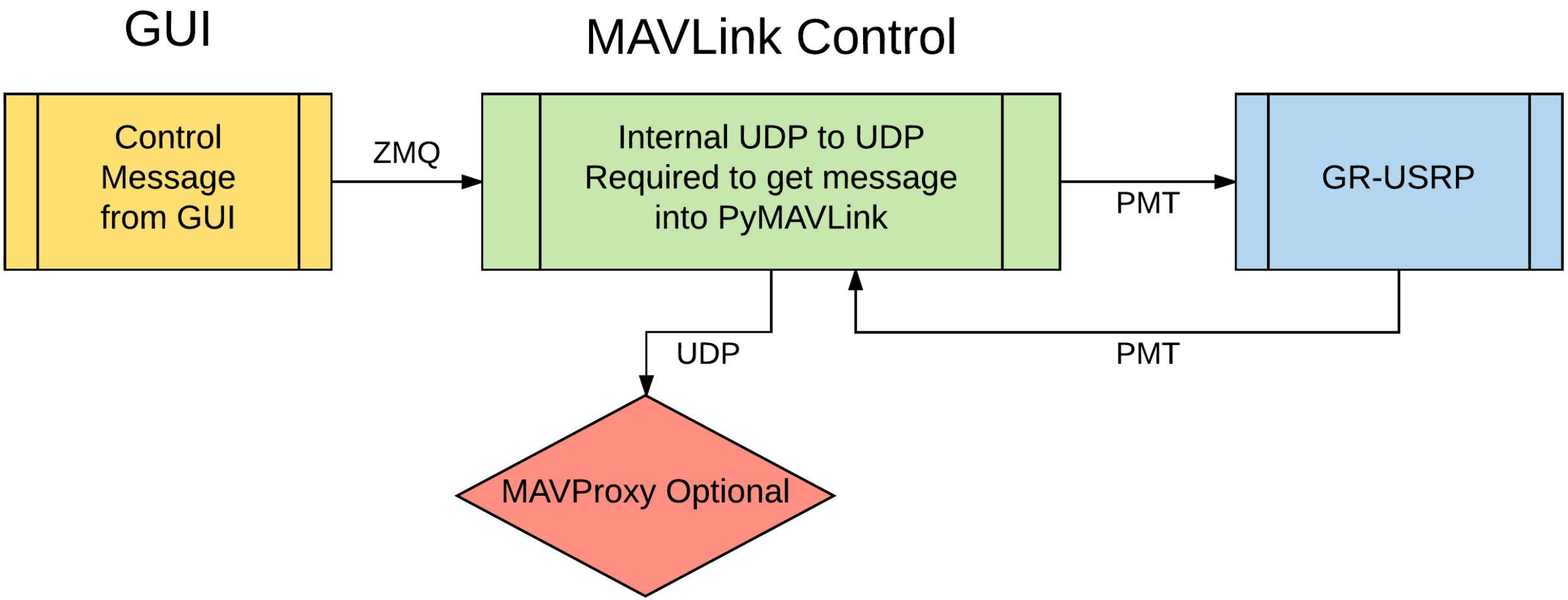


- Uses Ettus Research Hardware to receive the Burst data
- Is designed to be able to run on the RPi3 (No GUI)

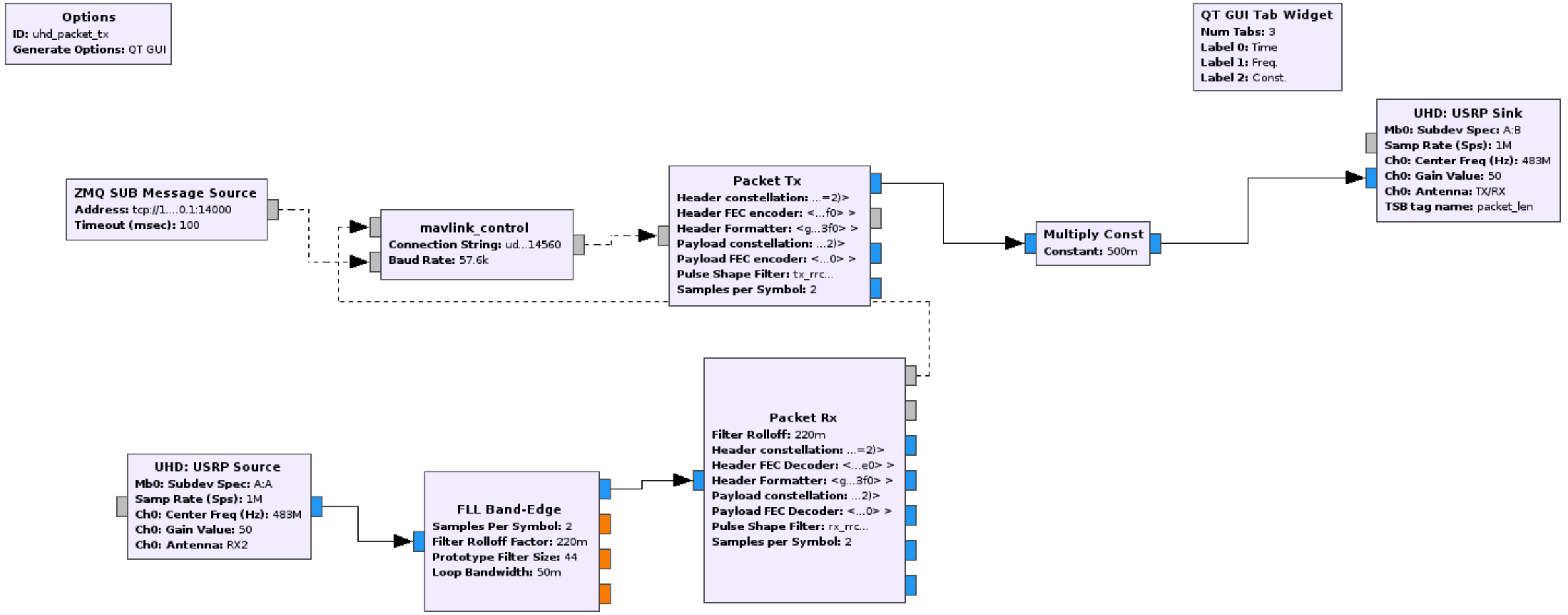
Planned Packet Based Communications

- Uses the packet based communications within GNU Radio
- Is designed for full bi-directional MAVLink message flow
  - MAVLink controller will run on the PC instead of the RPi
  - Requires higher throughput and improved communications reliability to work
- Currently works in simulation and a path forward is planned for hardware

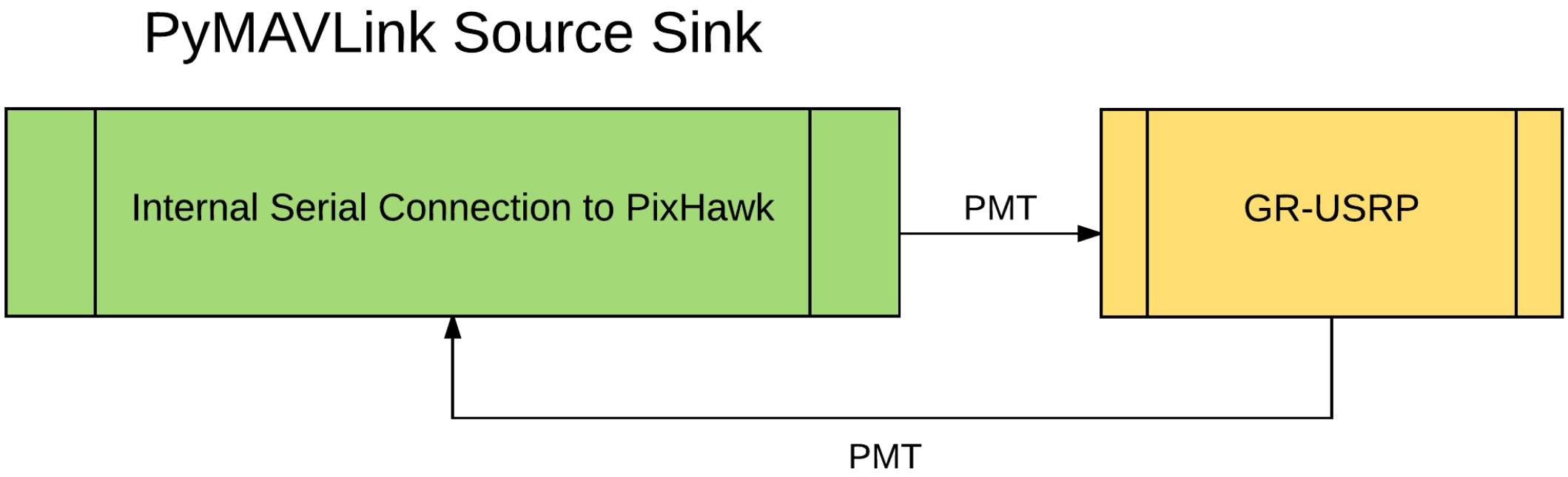
Message Flow for PC side



PC TX Flowgraph

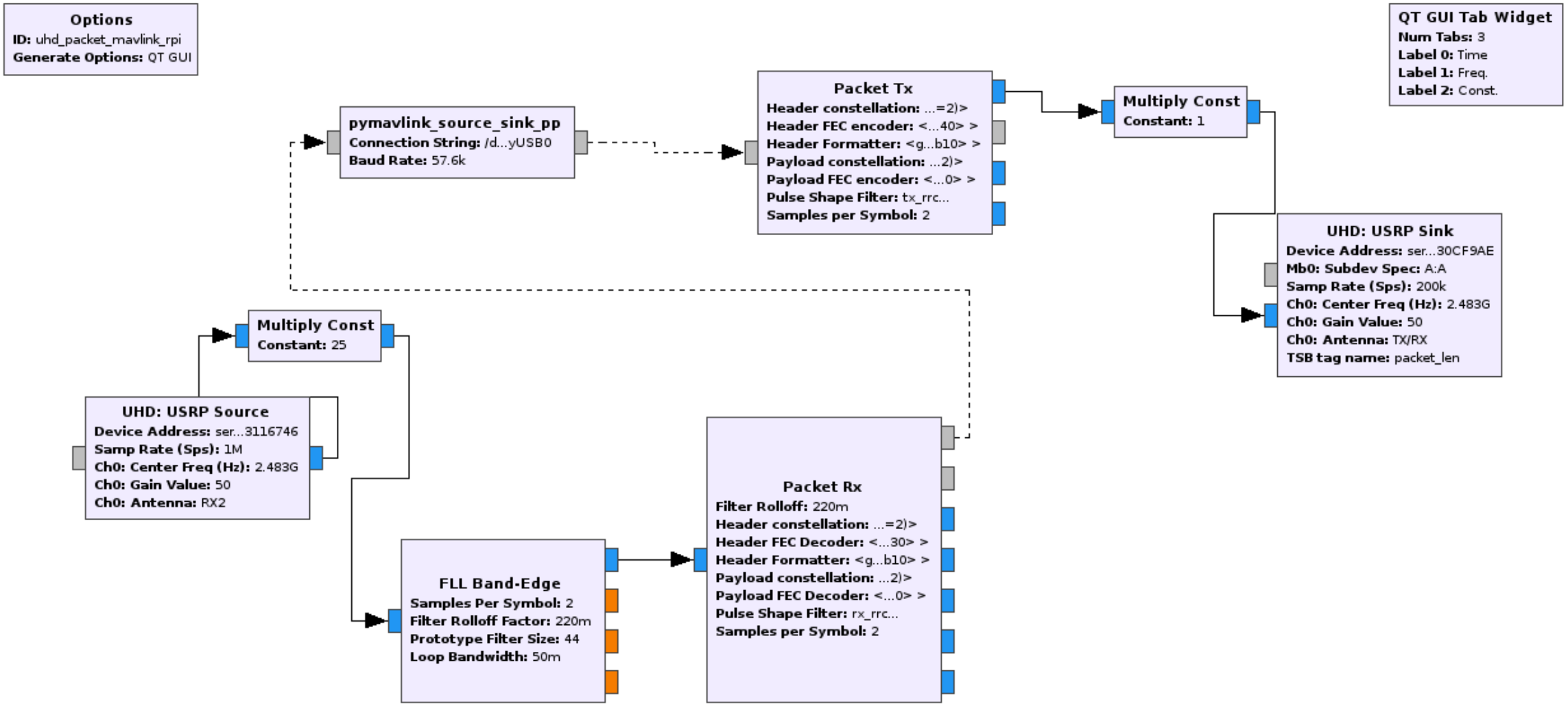


Message Flow for RPI



The GNU Radio blocks are simple, but allows all MAVLink Messages to be passed.

Raspberry Pi3 Flowgraph



Agenda

- Introductions
- Overview of Hackfest Hardware/Software
- Overview of new gr-uaslink software
- Example use of gr-uaslink with a SITL system
- Example use of gr-uaslink with a serial connected pixHawk 2 controller
- Example OTA test of gr-uaslink to a 3DR-solo
- **Topics of interest for the Hacker Space and Challenges**

Hacker Space Brainstorming Ideas

- Improve GNU Radio Packet performance and resiliency
- Improve the AGC blocks for GNU Radio

Limitations of indoor system

- Lack of GPS data limits may limit the commands which can be used
  - We are currently using channel override commands to control the UAS system
- Requires the addition of extra hardware to overcome control issues
  - Fake GPS
  - Optical Flow Control

Limitations of flight control system

- Issues for tethered flights
  - System waits to reach a height before returning from takeoff. The tethered system will have to allow the UAS to reach a known height.
  - Commands to move a specified distance will not work with a tethered system. RC override and Velocity commands should work in the tethered environment.

Discussion?

Additional backup slides

MAVLink

- Variable frame based system
  - Frames are defined using XML
  - PyMAVLink builds a python interface for the message sets based on the XML files
  - Large set of common commands
  - http://mavlink.org/messages/common

ArduPilot

- Software operates on the Pixhawk
  - Open-source flight control
  - Understands the MAVLink command set
  - Requires handshakes between ground station and flight unit every 3 seconds to continue flight
  - Most popular flight control software
  - Can be used on rovers, planes, water vehicles and copters

Optional Software

- QGroundControl – Ground control software
  - http://qgroundcontrol.com/
- MAVProxy
  - Routes MAVLink to different UDP, Serial, TCP
  - http://ardupilot.github.io/MAVProxy/html/index.html
- SITL
  - Software in the Loop for testing and development
  - http://ardupilot.org/dev/docs/sitl-simulator-software-in-the-loop.html