CM3203 Project

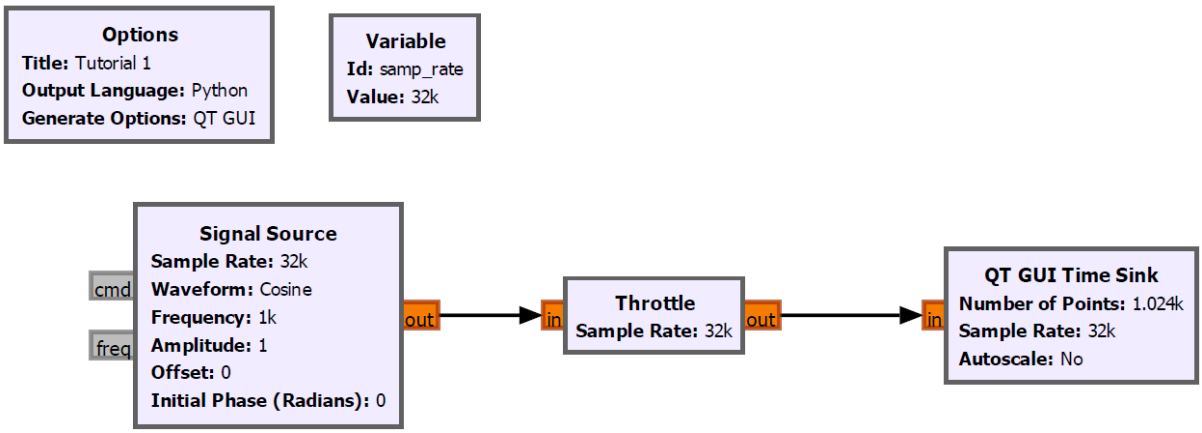
Initial Plan - Implement a GNU Radio driver for the FLEX-6400 SDR Transceiver

Author – Jonny Slim C1634544

Supervisor – David Humphreys
Client – Derek Kozel

Project Description

GNU Radio is an open-source development toolkit, capable of developing and simulating real-world radio systems. It is a modular flow-based framework, in which you piece together predefined functions and operations to develop your own communication system. Its infrastructure is written in C++, but the user-created systems and tools can be written in Python, which is what I'll be doing.



Here is an example of how a flow can be created. In this case, a sinusoidal wave is produced and then plotted by the "Time Sink"[1]. GNU Radio also generates a corresponding Python script with appropriate classes.

The FLEX-6400 is a SDR[2] where components and calibrations usually defined in hardware are instead implemented with software. Most of the RF[3] signal processing is performed with an API/Driver, allowing for greater flexibility in what channels and frequencies the radio is listening to, as well as easier modulation of the signal it's transmitting. SDRs can be reconfigured on the fly, essentially completely transforming its purpose with plug-and-play programs able to run on demand.

As an illustration, the current format the FLEX API uses is shown below (taken from the SmartSDR wiki):

NAME

Set the name of the radio which will show up in broadcasts and optionally the display.

```
C[D]<seq_number>|radio name <name>

< name >    = alphanumeric name of the radio - if left blank, the current name is returned
```

Example:

```
C21|radio name 6M
```

See Response Format for details on the format of the response messages from the radio

Responses

Hex Response	Message	Debug Output / Meaning
00000000	<slice_rx>	OK radio name set
5000002C		Radio name provided is too long (>15 characters)

Response Example:

```
R21|0|OK
```

It's my job to create these commands/responses in GNU Radio and Python to allow control and communication with the FLEX radio.

Currently, The FLEX-6400 radio comes with an API, but it is restrictive due to the fact it is built with the .NET framework. Building a new API with GNU Radio would enable cross-platform uses of the SDR – such as on embedded devices – opening many more opportunities to use its potential. This is what my project will be producing.

Many examples of these uses are research projects that could benefit from the flexibility of GNU Radio and the performance of FLEX. One example could be Oberoi's paper (Oberoi, 2000), in which the authors outline how low frequency radio signals are the last truly unexplored area of Earth's electromagnetic spectrum. This is a common research proposal, and one I believe could benefit from the inclusion of a SDR. For example, SDRs can be quickly and easily upgraded with enhanced features. This update can be delivered over-the-air, meaning the desired radio settings can be adjusted with the satellite already deployed. Furthermore, GNU Radio is an excellent framework to create these updates, so a marrying of the two may prove very beneficial.

Another more specific case is an experiment detecting how radio waves are altered when passing through Earth's ionosphere during a solar eclipse, conducted by William Lloyd (Lloyd, 2019). Here the author streams IQ data[4] from a different make of SDR, but it would benefit from the greater transmit ability of the FLEX-6400. It also provides a nice example of how the FLEX can be plugged straight into an experiment provided it has the drivers I plan to produce in this project.

The general purpose of this project is a proof of concept through the combination of the FLEX-6400 and GNU Radio, but there are still some clear objectives that should be met at the end of the 13 weeks. These objectives are discussed in the next section.

Project Aims & Objectives

The project will be delivered in the form of a Python file/files with classes for each of the objectives. These files will be a merging between the generated GNU radio files and communication directly with the FLEX radio. The ultimate goal at the end of the project will be able to send and receive IQ data to and from the FLEX and my remote computer. However, there are few steps along the way to achieve this:

- **Discover and Configure FLEX**
  The basic entry point of the project
  - **Locate FLEX over TCP**
    Discover and start communication with the FLEX Radio (Week 1)
  - **Query Status Info and receive a Response**
    Send a request for the SDR's current status information and translate the response (Week 2)
  - **Set SDR parameters with confirmation**
    Adjust SDR's current configuration settings; Request current settings, Post new settings, Request again to confirm change (Week 3)
- **Receive Audio data from SDR**
  The main portion of the project. Requirements of receiving and saving of audio data and possibly visualising it (Week 4-5)
- **Receive IQ Data from SDR**
  Another main requirement. Again data can be streamed and saved locally, this time in the IQ complex form (Week 7-8)
- **Transmit Audio data from PC**
  Transmission of data will be an extension and if achieved it will be in addition to the basic requirements stated above. An audio file can be read and transmitted from local machine and is received by the FLEX radio (Week 9-10)
- **Transmit IQ data from PC**
  As above, transmission of raw IQ data would be a nice extension if it is time-feasible (Week 11-12)

The initial intention is to have a weekly meeting with my supervisor to make sure I'm on the right track and the project is progressing in a timely manner.

Throughout the project, a diary will be kept with the progress made that week and the updated goals for the subsequent week. Code will be version controlled with a GitLab.

(Week 1 starts 8th February)

#	Acronyms or Terms	Definition
[1]	Sink	The end point/output of the signal flow
[2]	SDR	Software Defined Radio
[3]	RF	Radio Frequency
[4]	IQ Data	A cartesian (instead of polar) representation of the RF signal, where both amplitude and phase can be modified simply

Work Plan

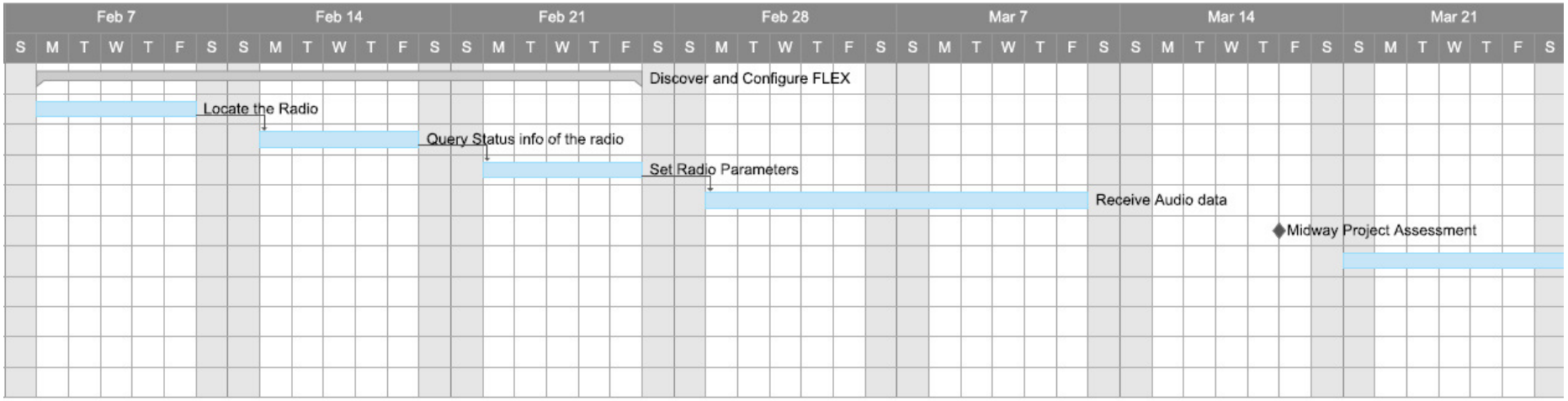
The current Work Plan has two main milestones; The Project Hand-in date on May 14th and a Midway Project Assessment in the week starting March 15th. The midway assessment is to evaluate how much progress the project has made, whether my project is still following my initial plan and how I've developed personally due to the project.

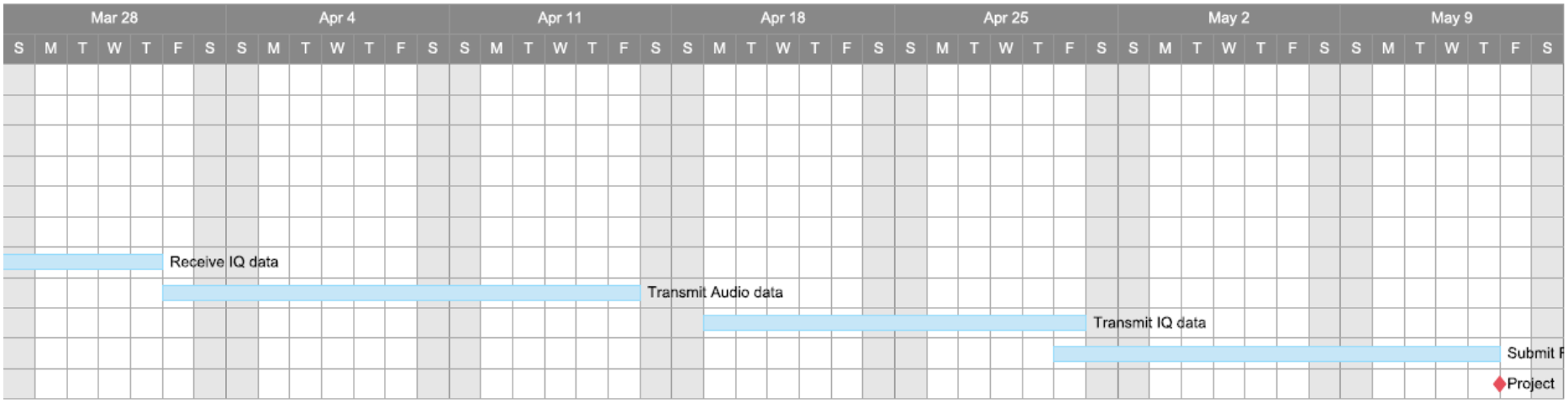
I will also have another two milestones, one for completion of the Discovery and Configuration of the FLEX SDR and one for the completion of Reception of the two data types. The two preliminary dates for these milestones are Feb 26th and April 1st respectively.

Each of these milestones will be accompanied with a meeting with my supervisor and client to determine the success or failure of the task.

FLEX SDR Project

Task Name	Duration	Start	Finish	Comments
Discover and Configure FLEX	15d	Feb 8, 2021	Feb 26, 2021	
Locate the Radio	5d	Feb 8, 2021	Feb 12, 2021	Ensure radio is discoverable and API can send and receive packets
Query Status info of the radio	5d	Feb 15, 2021	Feb 19, 2021	API can request radio's status information and current settings
Set Radio Parameters	5d	Feb 22, 2021	Feb 26, 2021	API can configure radio settings and parameters and receive confirmation of new calibration
Receive Audio data	10d	Mar 1, 2021	Mar 12, 2021	Audio data can be streamed from radio and be saved locally
Midway Project Assessment	0	Mar 19, 2021	Mar 19, 2021	Progress should be documented weekly and be collated here to be discussed during weekly meeting
Receive IQ data	10d	Mar 21, 2021	Apr 1, 2021	IQ data can be streamed from radio and be saved locally
Transmit Audio data	11d	Apr 2, 2021	Apr 16, 2021	Audio file can be read and streamed via API to FLEX radio
Transmit IQ data	10d	Apr 19, 2021	Apr 30, 2021	IQ data can be streamed to FLEX radio
Submit Final Report	10d	Apr 30, 2021	May 13, 2021	2 weeks to discuss project evaluation and to complete final report
Project Hand In	0	May 14, 2021	May 14, 2021	


Feb 7
Feb 14
Feb 21
Feb 28
Mar 7
Mar 14
Mar 21
S M T W T F S
Discover and Configure FLEX
Locate the Radio
Query Status info of the radio
Set Radio Parameters
Receive Audio data
Midway Project Assessment


Mar 28
Apr 4
Apr 11
Apr 18
Apr 25
May 2
May 9
S M T W T F S
Receive IQ data
Transmit Audio data
Transmit IQ data
Submit F
Project

References

Oberoi, D., 2000. A Feasibility Study for a Very Low Frequency Satellite Based Radio Interferometer. [online] Academia.edu. Available at: <https://www.academia.edu/22900369/A_Feasibility_Study_for_a_Very_Low_Frequency_Satellite_Based_Radio_Interferometer?email_work_card=title> [Accessed 30 January 2021].

Lloyd W. 2019. Ionospheric Sounding During a Total Solar Eclipse Virginia Tech Available at: <https://vtechworks.lib.vt.edu/handle/10919/89951> [Accessed 3 February 2021]

Smart SDR API documentation: <http://wiki.flexradio.com/index.php?title=Main_Page#SmartSDR_API>

GNU Radio documentation: <https://wiki.gnuradio.org/index.php/Usage_Manual>