TARGET DETECTION AND RANGING USING SOFTWARE DEFINED FMCW RADAR

A project report submitted

in partial fulfilment of the requirement for the degree of

Bachelor of Technology

in

Electrical Engineering

by

Urooj Akhlaq (2018UEE0110)

Shashwat Singh (2018UEE0112)

Under the Guidance of

Ankit Dubey

Amit Kumar Singh

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY JAMMU

(JUNE 2022)

CERTIFICATE

This is to certify that the work contained in this report entitled **"Target Detection and Ranging Using Software Defined FMCW Radar"** is a bonafide work of **Urooj Akhlaq & Shashwat Singh (2018UEE0110, 2018UEE0112)**, carried out in the **Department of Electrical Engineering, Indian Institute of Technology Jammu** under my supervision and that it has not been submitted elsewhere for the award of any degree.

June, 2022 Supervisor: Ankit Dubey & Amit Kumar Singh

Jammu Assistant Professors

Department of Electrical Engineering

Indian Institute of Technology Jammu

DECLARATION

I declare that this report presents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when required.

Urooj Akhlaq(2018UEE0110)

Shashwat Singh (2018UEE0112)

ACKNOWLEDGEMENT

We would like to thank our supervisors Dr. Ankit Dubey, Assistant Professor, Electrical Engineering Department, IIT Jammu and Dr. Amit Kumar Singh, Assistant Professor, Electrical Engineering Department, IIT Jammu for their continuous support and motivating us to complete our project. Their comments, discussions, constant reminders and interactions helped us in numerous ways throughout the project.

We would also like to thank our friends in our batch who provided valuable input whenever we faced any hurdle.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ADI	Analog Devices Inc.
CW	Continuous Wave
FMCW	Frequency Modulated Continuous Wave
GRC	GNU Radio Companion
GUI	Graphical User Interface
IIO	Industrial I/O
LNA	Low Noise Amplifier
SDR	Software Defined Radio
SFCW	Stepped Frequency Continuous Wave
UAV	Unmanned Aerial Vehicle
USRP	Universal Software Radio Peripheral
VCO	Voltage Controlled Oscillator

ABSTRACT

The design and implementation of a software defined frequency-modulated continuous wave (FMCW) radar operating in S-band for detection and ranging will be implemented in this project. The FMCW radar model will be developed using GNU radio open source software. The ADALM-PLUTO will be used as software defined radio(SDR). A hardware prototype of the above designed FMCW radar is made using ADALM-PLUTO, Power amplifier , wide band horn antennas (S-band) will be used as transmitter and receiver to implement the radar. After modifying FMCW radar, it can be used to detect UAV , in autonomous vehicle detection and in healthcare.

Chapter 1

Introduction

1.1 Motivation

Radars have been in use in a variety of applications since they were developed by the British military during world war II [1]. Modern radar is in use in diverse fields such as air-defence systems, marine radars, self driving vehicles, etc. Due to this a need for direct and highly accurate measurements of target is needed. Thus if one can make a lightweight, cost effective and accurate radar it can be implemented on a much larger scale to provide services for people. FMCW radars are one such light weight radar system that are simple to manufacture and operate and can be utilized in applications such as early warning systems, IoT and vehicle detection. The Btech Project consists of simulation and development of such FMCW radar using GNU-Radio, a software development toolkit to implement software defined radio(SDR).

1.2 Background

A Doppler Radar is a specialized radar that uses the doppler effect to produce velocity data about objects at a distance. The main principle of doppler radar is change in frequency of the wave with respect to the observer.

FMCW radar is a type of doppler radar that uses Frequency modulated continuous waves to detect targets. It sends out a frequency modulated signal and after receiving it back examines the phase delay between both waves and determines the range.

1.3 Problem Statement

In this project we aim to develop a low cost and portable FMCW radar for detection and ranging of target. ADALM PLUTO a software defined radio would be utilised in building this project. GNU Radio Companion will be used to implement this project. Further Custom Blocks will be built in Python Language for GNU radio Companion to detect and calculate range of the target.

1.4 Organization of the Report

The Project report has the following template:

Chapter 2: This Chapter gives the literature survey for the project

Chapter 3: This Chapter Introduces to the basics of FMCW radar and equations used

Chapter 4: This Chapter tells how the Implementation of the project took place

Chapter 5: This Chapter Concludes the report and give ideas that can be implemented in future in the project

Chapter 2

Literature Review

The following academic papers greatly contributed to the development of the project.

1. *An Overview of FMCW Radar Systems in MATLAB* [2]

   This paper lays the foundation for the FMCW basics and simulations performed

2. *Signal processing of range detection for SFCW radars using Matlab and GNU radio* [3]

   This conference paper deals with the simulation of SFCW radar on MATLAB and GNU Radio Companion. The use of GNU Radio Companion was helpful in implementation of the project.

3. *FMCW Software Defined Radar for Range and Speed Estimation* [4]

   This conference paper helped us with the calculations needed to be performed as well as a background on GNU radio Companion.

4. *Real time implementation of FMCW radar for target detection using GNU radio and USRP* [5]

   This conference paper implements FMCW radar using GNU radio and USRP. USRP is a SDR similar to ADALM Pluto.

5. *A Radar System for Sensing and Tracking (LH01)* [6]

   This Project report helped us on how to proceed with our FMCW radar implementation. The report contains detailed methodology which helped us in implementing the radar.

6. *Software Defined Radio Based Frequency Modulated Continuous Wave Ground Penetrating Radar* [7]

   Results from this thesis were helpful in choosing certain values for radar in GRC. The chip used in this thesis is the same as the one used by ADALM PLUTO SDR.

7. *Implementation of FMCW Radar by Using SDR* [8]

This conference paper implemented FMCW radar using PLUTO SDR on MATLAB.

Chapter 3

FMCW Radar

3.1 Introduction

A continuous-wave(CW) Radar is one which transmits and receives signals simultaneously [9]. The CW radar is limited in its transmitted power due to interference between transmitted and received signal. And since there is no way to measure time difference, range cannot be calculated. In a FMCW radar a frequency modulated signal is transmitted and received. When the echo signal is received , change in frequency is measured and range is calculated [10].

3.2 Basic Block Diagram

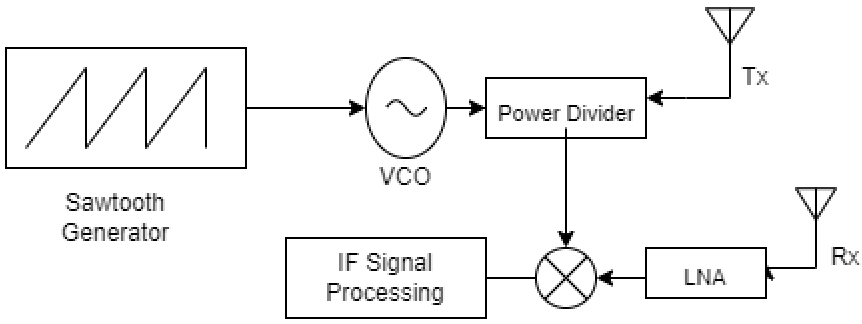


Fig 3.1: Block diagram of FMCW radar

A Basic FMCW radar has following components as shown in Figure 3.1:

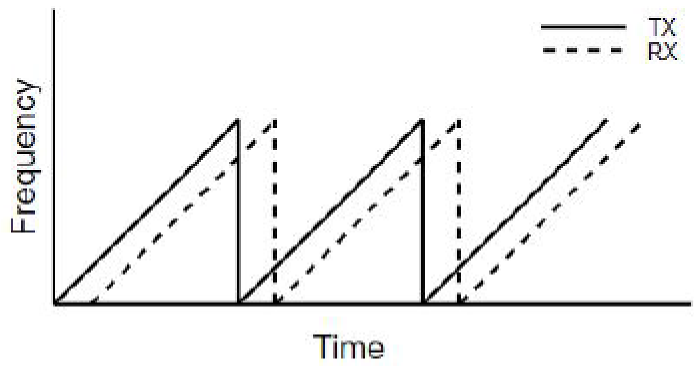
1. **Signal Generator:** A signal generator is a device that generates electronic signals such as sawtooth, sine, ramp, cosine and feeds the signal to VCO
2. **Voltage Controlled Oscillator:** Voltage Controlled Oscillator(VCO) is an oscillator circuit whose frequency can be controlled by a DC input voltage. VCO is used to modulate the frequency of our input signal.
3. **Power divider:** A power divider divides an incoming signal into two or more signals.

4. **Mixer:** A mixer is a device that inputs two signals and creates new signals equal to difference and sum of input signals.
5. **Low Noise Amplifier:** A Low Noise Amplifier(LNA) amplifies a very low power signal without degrading its signal to noise ratio.
6. **Antennas:** An antenna is a transducer that converts electrical power to electromagnetic signal and vice versa. It can work in either transmitting or receiving mode.

3.3 Working

Figure 3.2 shows the transmitted (Tx) and Received (Rx) that is reflected from an object located at a distance R. The received signal is the echo of the transmitted signal. The beat frequency f_b of a signal is a measure of the target's range when there is no doppler shift

Fig 3.2: Frequency vs Time for FMCW



in the signal.

A signal which has a linear frequency increase in time domain is called a chirp signal. Now the slope of the chirp signal sent by FMCW radar can be determined by

$$s = \frac{B}{T_c} \quad (1)$$

where B is the Bandwidth of the signal and T_c is the sweep time or chirp period. The delay between transmitted and received signal is given by

$$\tau_d = \frac{2R}{c} \quad (2)$$

where is the R is the distance/Range of the object and c is the speed of light in free space. The beat frequency is the difference between the frequency of two waves and it is given by

$$f_b = s\tau_d \tag{3}$$

Thus Range in equation (2) can be calculated by putting values of equation (1) and (2) in equation (3).

$$R = \frac{cT_c}{2B} f_b \tag{4}$$

To accurately measure the distance between two objects, some minimum distance is required. This minimum distance is called Range Resolution and it is given by

$$R_{res} = \frac{c}{2B} \tag{5}$$

3.4 Conclusion

In this chapter we gave the basics of an FMCW radar. We also explained the mathematical equations used for range calculations. This knowledge would help us design our FMCW radar model.

Chapter 4

Implementation

4.1 Introduction

This chapter will discuss Software-defined radio(SDR) and how it is used for real time detection and ranging. This will be followed by implementing a FMCW radar using ADALM pluto SDR and GNU-Radio software.

4.2 Software Defined Radio

Software-defined radio(SDR), is a radio communication system in which the properties of carrier frequency, bandwidth, etc can be modified through software [11]. ADALM-PLUTO is one such SDR developed by Analog Devices Inc. [12]. The module contains an independent transmission and receiving channel that can be operated in full duplex mode. The Specification of ADALM Pluto are:

- Operating frequency: 325MHz to 3.8GHz
- Sample rates: 65.2 kSPS to 61.44 MSPS
- Resolution: 12 bits
- Tx power output: 7 dBm
- Rx and Tx modulation accuracy: -34 dB (2%)

Fig 4.1: A picture of ADALM-PLUTO SDR

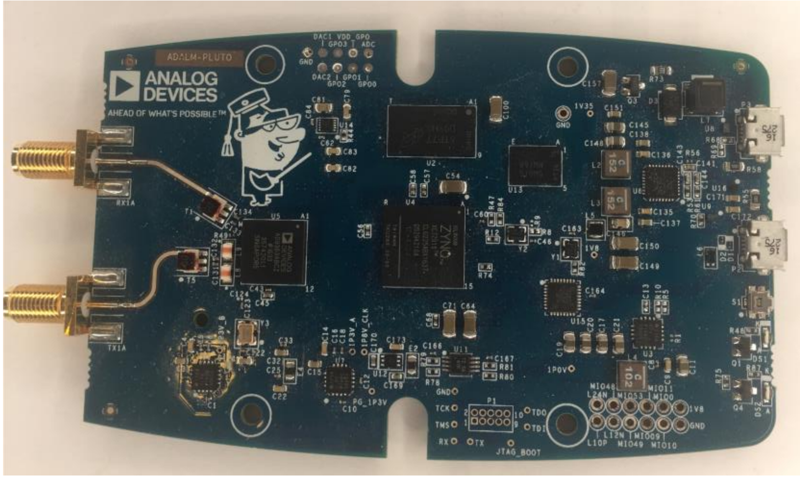


Fig 4.2: Printed Circuit Board of ADALM-PLUTO SDR

4.3 GNU-Radio companion

GNU-Radio companion is a free and open source graphical user interface tool that provides signal processing blocks that help to implement software radios. It was developed by Eric Blossom in 2001 [13]. GNU radio can be used with any software defined radio such as HackRF, RTL-SDR , etc including our device ADALM PLUTO SDR. In GNU Radio Companion (GRC), the signal processing blocks are made using Python and C++ programming languages.

4.4 Installing GNU Radio on Linux

We will be using Ubuntu Linux as our choice of operating system. To install GNU radio open terminal on Ubuntu and type the following code [14]:

```
sudo apt-get-repository ppa:gnuradio/gnuradio-releases-3.9
sudo apt-get update
sudo apt-get install gnuradio python3-packaging
```

After installing GNU Radio you can open GNU Radio companion as shown in figure 4.3.

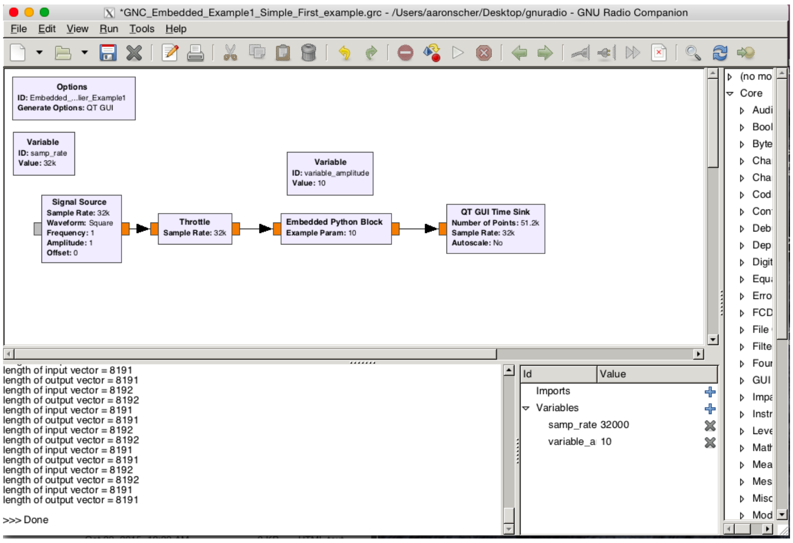


Fig 4.3: Example of GNU Radio Companion window

4.5 Installing necessary drivers

Now to run ADALM-PLUTO on our system we need to install certain drivers [15]. First driver to install is libiio. Libiio is a driver that has been developed by Analog Devices Inc. to ease development of software software interfacing Linux Industrial I/O (IIO) devices [16]. To install it following code was run in the terminal:

```
sudo apt-get install git libxml2 libxml2-dev bison flex libcdk5-dev cmake
python3-pip
cd ~
git clone --branch v0.19 https://github.com/analogdevicesinc/libiio.git
cd libiio
cmake ./
make all -j4
sudo make install
sudo ldconfig
cd bindings/python/
sudo python3 setup.py.cmakein install
```

Next driver to install libad9361-iio. AD9361 is the chip that is used in ADALM-PLUTO device. To install the library following code was run:

```
cd ~
git clone https://github.com/analogdevicesinc/libad9361-iio.git
```

```
cd libad9361-iio
cmake ./
make -j3
sudo make install
```

Final driver to install is pyadi-iio. This is a python abstraction module for ADI Hardware with IIO Drivers to make them easy to use [17]. To install this driver following code was run:

```
cd ~
git clone https://github.com/analogdevicesinc/pyadi-iio.git
cd pyadi-ii
sudo python3 setup.pyo install
```

If everything is installed correctly running the command `iio_info -s` would give you the available devices as given below:

```
Library version: 0.23 (git tag: 00d9621)
Compiled with backends: local xml ip usb
Available contexts:
0: 0456:b673 (Analog Devices Inc. PlutoSDR (ADALM-PLUTO)), serial=
[usb:1.3.5]
1: 192.168.6.1 (Analog Devices PlutoSDR Rev.C (Z7010-AD9364)), serial=
[ip:pluto.local]
```

If you get the error `Unable to create Local IIO context : No such file or directory (2)` you can install pylibiio [18] using command `pip install pylibiio`

This ends our installation of drivers.

4.6 Making GNU-Radio flowgraph

The following flowgraph is built in GNU Radio Companion. Two custom blocks are also built in python to detect peak in the mixed signal and calculate range of the target.

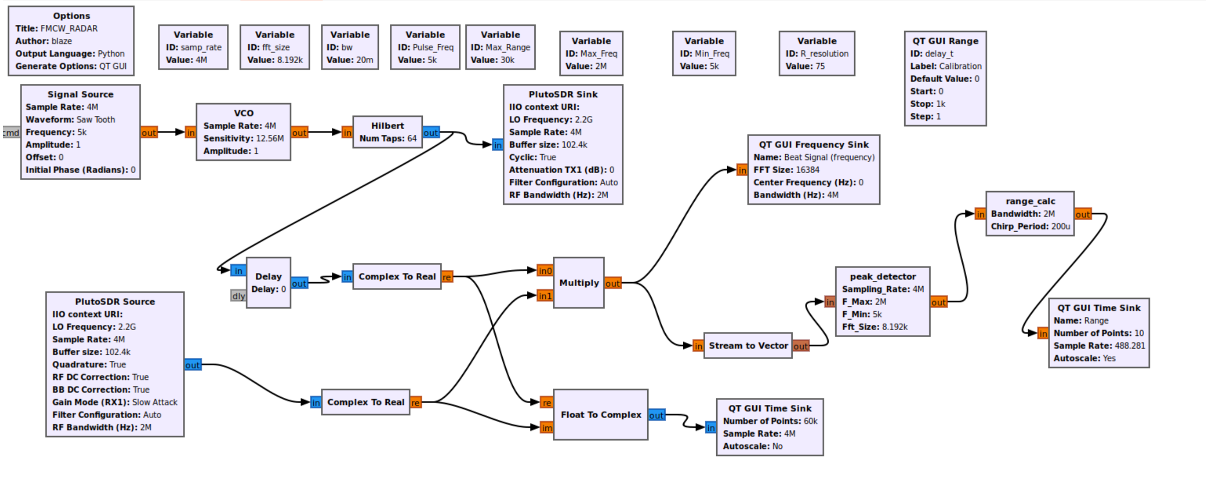


Fig. 4.4: Designed flowgraph in GNU Radio

Following Variables and their values are used:

- Sampling rate: samp_rate = 4M or 4e6
- Fourier Transform length: fft_size = 8.192k or 8192
- Bandwidth: bw = 20m or 20e-3
- Pulse Frequency: Pulse_Freq = 5k or 5000
- Maximum Range: Max_Range = 30k or 30e3
- Maximum Frequency: Max_Freq = 2M or 2e6
- Minimum Frequency: Min_Freq = 5k or 5e3
- Range Resolution: R_resolution = 75

4.7 Flowgraph explanation

Following is the explanation of the flowgraph given in Figure 4.4. First the signal source generates a sawtooth signal with a given sampling rate and pulse frequency. Then the signal is sent to the VCO block to help with frequency modulation.

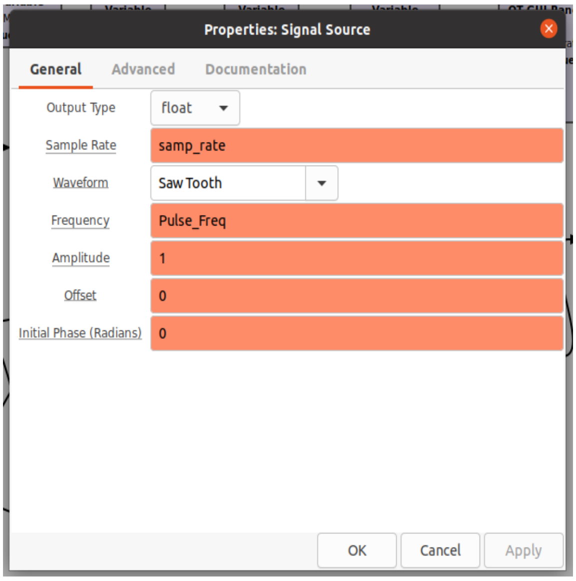


Fig. 4.5: Signal Source Block

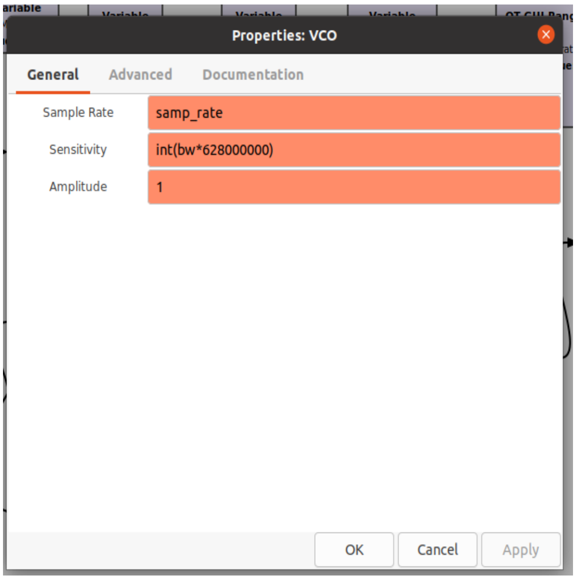


Fig. 4.6: VCO Block

The sensitivity of VCO block is chosen as Bandwidth *bw* times 6.28e8. The signal is then passed through a hilbert block to perform hilbert transformation on the signal. After performing hilbert transformation the signal is split into two parts. The first part is fed to the Pluto SDR sink block and the other part is taken as is to compare with the received signal.

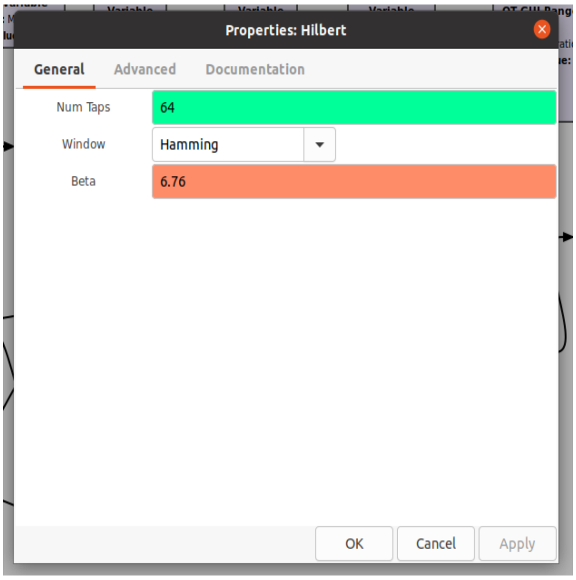


Fig. 4.7: Hilbert block

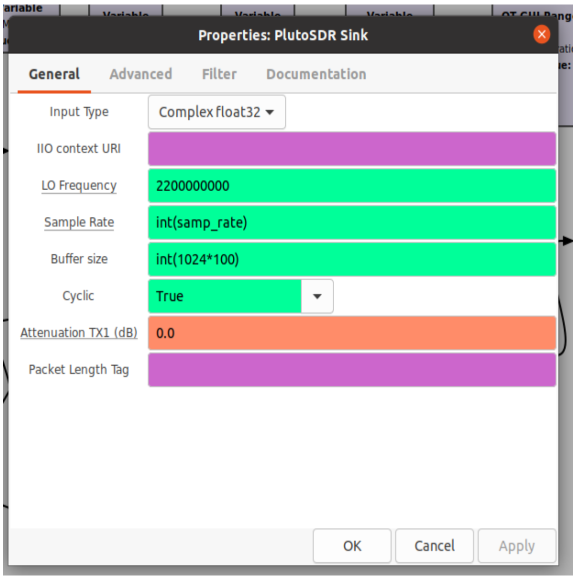


Fig. 4.8: PlutoSDR sink block

The carrier frequency for the signal is chosen as 2.2GHz. Buffer size is taken as 1024k. The echo is received through the PlutoSDR Source block. After receiving the echo both signals are multiplied. The multiplied signal is split into two where one part is used to display beat signal frequency whereas the other is passed on to detect peak using Fast Fourier Transform (FFT). The peak is detected through our custom peak detector block and then range is calculated through a custom made range calculation block. After that range is fed into the GUI time sink for display.

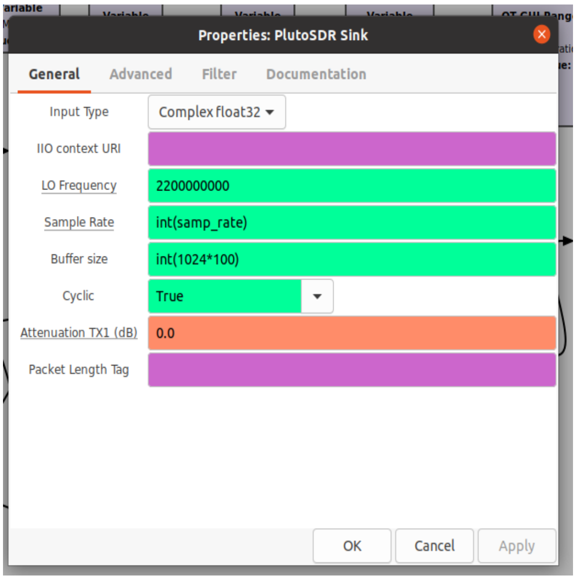


Fig. 4.9: PlutoSDR source block

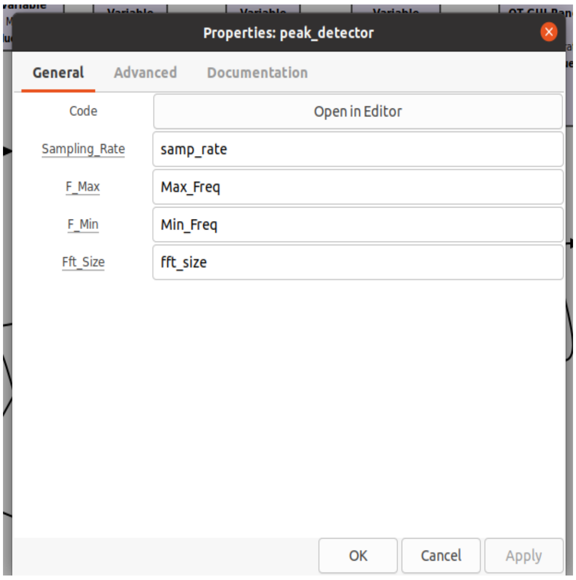


Fig. 4.10: Peak detector Block

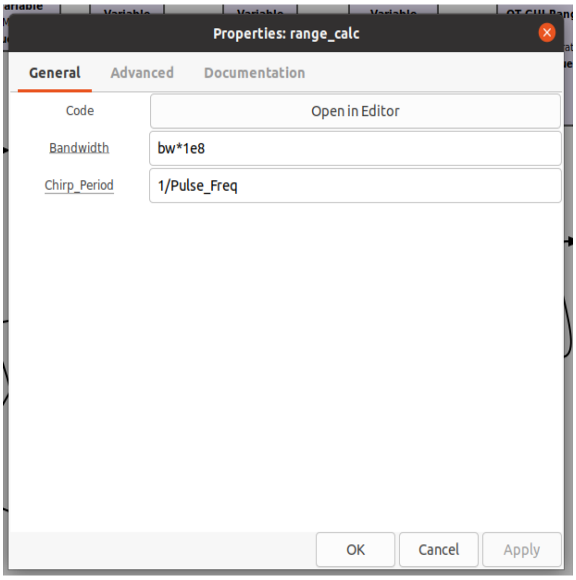


Fig. 4.11: Range Calculation Block

4.8 Building Custom blocks

To detect peak of the combined signal and calculating range, custom signal processing blocks needed to be made in GNU Radio Companion. First the peak detector was built. The input of the peak detector was a stream of vector that included both transmitted and received signal multiplied. After that FFT is performed on the combined signal. Meanwhile a low pass filter is made using a variable mask. The FFT signal is passed through the mask and the signal is filtered out. Then the peak is calculated by checking which signal block has the maximum frequency. Following is the code for peak detector:

```
import numpy as np
from gnuradio import gr

class peak_detector(gr.sync_block):
    def __init__(self, sampling_rate=4e6, f_max=2e6, f_min=5e3,
fft_size=8192):
        gr.sync_block.__init__(self,
            name="peak_detector",
            in_sig=[(np.float32, fft_size)],
            out_sig=[np.float32])
        self.fft_size = fft_size
        self.sampling_rate = sampling_rate
        self.f_max = f_max
```

```
        self.f_min = f_min

        #setting up frequency interval
        T_samp = 1/self.sampling_rate
        self.f = np.fft.fftfreq(self.fft_size, d=T_samp)
        self.mask = np.logical_or(np.abs(self.f) > f_max,
np.abs(self.f) < f_min)


    def work(self, input_items, output_items):
        in0 = input_items[0][0]

        #input data conditions check
        if(len(in0) != self.fft_size):
            raise ValueError('Input vector size does not match fft
size')

        # applying fourier transform and lowpass filter
        fft_signal = np.fft.fft(in0)
        fft_signal[self.mask] = 0

        # retrieving frequency peak and set belonging frequency to
output
        idx = np.argmax(np.abs(fft_signal))
        if(idx >= len(self.f)):
            idx = len(self.f) - 1
        output_items[0][0] = abs(self.f[idx])

        return len(output_items)
```

For range calculation, the peak signal is sent as input. After that the equation (4) is used to calculate range. Following is the code used to calculate range:

```
import numpy as np
from gnuradio import gr

class range_calc(gr.sync_block):
    def __init__(self, bandwidth=2e6, chirp_period=200e-6):
        gr.sync_block.__init__(self,
```

```
            name="range_calc",
            in_sig=[np.float32],
            out_sig=[np.float32])
        self.bandwidth = bandwidth
        self.chirp_period = chirp_period
        self.speed_of_light = 300000000


    def work(self, input_items, output_items):
        in0 = input_items[0]

     # calculating range
        output_items[0][:] = (
in0[:]*self.speed_of_light*self.chirp_period ) /
(2*self.bandwidth)

        return len(output_items[0])
```

4.9 Results

After completing the flowgraph, the PlutoSDR is connected to the Linux system through usb and FMCW radar is tested. The radar gives the range of the object as per the calculation.

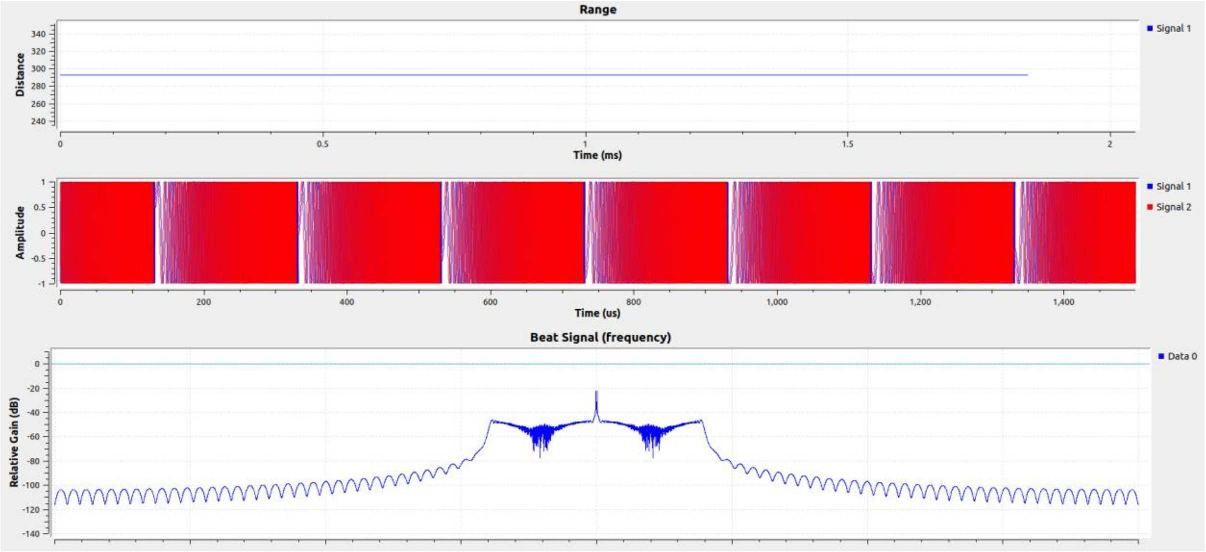


Fig. 4.12: Range observed for an object at 295m

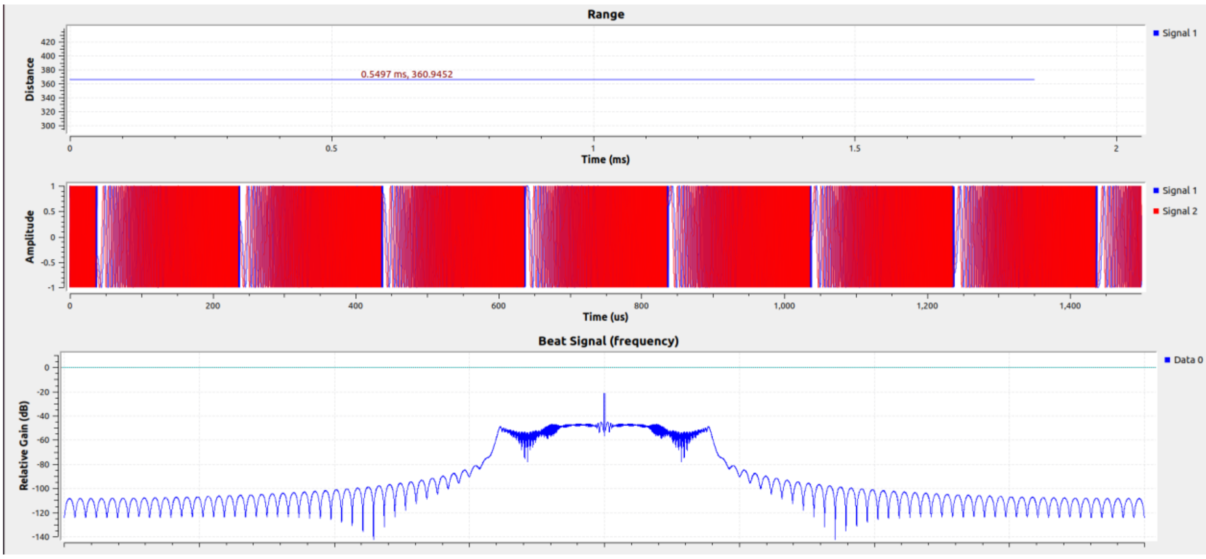


Fig. 4.13: Range observed for an object at 365m

Chapter 5

Conclusion and Future Work

5.1. Conclusion

In this Btech Project we achieved, successful demonstration of FMCW radar using ADALM PLUTO SDR and GNU Radio Companion. The report shows the relevant theory utilised in building and working of radar and how the Btech Project was completed.

5.2. Future work

In future work can be done to calculate velocity of the target being measured. Further an object classification system can be developed by collecting necessary data and applying machine learning algorithms to it. This object classification system can be then incorporated into the radar for real time object classification and ranging.

Real time detection can also be highly affected by environment noise, therefore further research in the area can be done.

References

[1]: T. Long, “Feb. 26, 1935: Radar, the Invention That Saved Britain,” *WIRED*, Feb. 26, 2008. Accessed: May 27, 2022. [Online]. Available: https://www.wired.com/2008/02/dayintech-0226/

[2]: K. Parrish, “An Overview of FMCW Systems in MATLAB,” *unknown*, May 10, 2010. https://www.researchgate.net/publication/260286270_An_Overview_of_FMCW_Systems_in_MATLAB (accessed May 27, 2022).

[3]: Azizah, A. B. Suksmono, and A. Munir, “Signal processing of range detection for SFCW radars using Matlab and GNU radio,” Oct. 2014. Accessed: May 27, 2022. [Online]. Available: http://dx.doi.org/10.1109/ic3ina.2014.7042617

[4]: A. M. Hegazy, M. Mosaad, and A. M. hassan, “FMCW Software Defined Radar for Range and Speed Estimation,” *unknown*, Jul. 25, 2016. http://dx.doi.org/10.13140/RG.2.1.2116.3761

[5]: Sundaresan S, Anjana C, T. Zacharia, and Gandhiraj R, “Real time implementation of FMCW radar for target detection using GNU radio and USRP,” Apr. 2015. Accessed: May 27, 2022. [Online]. Available: http://dx.doi.org/10.1109/iccsp.2015.7322772

[6]: AHN, Jaemoo, Yu, and Kijun, “A Radar System for Sensing and Tracking (LH01),” Mar. 2021. [Online]. Available: https://ykijun.github.io/website/assets/FYP.pdf

[7]: P. Fiske, “Software Defined Radio Based Frequency Modulated Continuous Wave Ground Penetrating Radar,” *UVM ScholarWorks*, 2020. https://scholarworks.uvm.edu/graddis/1300/

[8]: S. Kaya, A. C. Yapici, B. Tibikoglu, and S. O. Yazici, “Implementation of FMCW Radar by Using SDR,” Nov. 2021. Accessed: May 27, 2022. [Online]. Available:

http://dx.doi.org/10.23919/eleco54474.2021.9677650

[9]: Electronics Club, “Continuous Wave (CW) Radar - Electronics Club Radar Engineering %,” *Electronics Club*, Jan. 10, 2020. https://electronics-club.com/continuous-wave-cw-radar/ (accessed May 27, 2022).

[10]: Electronics Club, “FMCW Radar - Electronics Club Radar Engineering FMCW Radar Electronics Club,” *Electronics Club*, Feb. 09, 2020. https://electronics-club.com/fmcw-radar/ (accessed May 27, 2022).

[11]: B. A. Fette, *Cognitive Radio Technology*. Academic Press, 2009.

[12]: Analog Devices Inc., “ADALM-PLUTO Evaluation Board,” *Analog Devices*. https://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/adalm-pluto.html (accessed May 27, 2022).

[13]: Contributors to Wikimedia projects, “GNU Radio,” *Wikipedia*, Apr. 01, 2022. https://en.wikipedia.org/wiki/GNU_Radio (accessed May 27, 2022).

[14]: “InstallingGR,” *GNU Radio*. https://wiki.gnuradio.org/index.php/InstallingGR (accessed May 27, 2022).

[15]: M. Lichtman, “5. PlutoSDR Basics — PySDR: A Guide to SDR and DSP using Python 0.1 documentation.” https://pysdr.org/content/pluto_intro.html (accessed May 27, 2022).

[16]: “What is libiio? [Analog Devices Wiki],” *Analog Devices Wiki*. https://wiki.analog.com/resources/tools-software/linux-software/libiio (accessed May 27, 2022).

[17]: Python Software Foundation, “pyadi-iio,” *PyPI*. https://pypi.org/project/pyadi-iio/ (accessed May 27, 2022).

[18]: V. all posts by mtr333, “Setting Up Adalm Pluto on Ubuntu,” *Tech Focus - AlexN*, Apr. 03, 2021. https://techfocusalexn.wordpress.com/2021/04/03/setting-up-

adalm-pluto-on-ubuntu/ (accessed May 27, 2022).