TEACHING TOOLS: PORTABLE SPECTRUM ANALYZER

MUHAMMAD HELMI BIN OTHMAN

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

BORANG PENGESAHAN STATUS TESIS

JUDUL: TEACHING TOOLS: PORTABLE SPECTRUM ANALYZER

SESI PENGAJIAN: 2015/2016

Saya MUHAMMAD HELMI BIN OTHMAN

mengaku membenarkan tesis (PSM/Sarjana/Doktor Falsafah) ini disimpan di Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dengan syarat-syarat kegunaan seperti berikut:

1. Tesis dan projek adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. ** Sila tandakan (/)

____	SULIT	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
/	TERHAD	(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)
____	TIDAK TERHAD	

(TANDATANGAN PENULIS)	(TANDATANGAN PENYELIA)
Alamat tetap : 145-B JALAN BANJAR MASJID TINGGI 34300 BAGAN SERAI PERAK	Nama Penyelia : Dr. Norharyati binti Harum
Tarikh : 24-8-16	Tarikh : 25-8-16

TEACHING TOOLS: PORTABLE SPECTRUM ANALYZER

MUHAMMAD HELMI BIN OTHMAN

This report is submitted in partial fulfillment of the requirement for the
Bachelor of Computer Science (Computer Networking)

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA
2016

DECLARATION

I hereby declare that this project report entitled

TEACHING TOOLS: PORTABLE SPECTRUM ANALYZER

is written by me and is my own effort and that no part has been plagiarized without citations.

STUDENT : ______________ DATE: 24-8-16

(MUHAMMAD HELMI BIN OTHMAN)

I hereby declare that I have read this project report and found this project report is sufficient in term of the scope and quality for the award of Bachelor of Computer Science (Computer Networking) With Honor.

SUPERVISOR : ______________ DATE: 25-8-16

(DR.NORHARYATI BINTI HARUM)

DEDICATION

To my beloved and parent and family

ACKNOWLEDGEMENT

Firstly, I am grateful to the God for the good health and wellbeing that were necessary to complete this project.

I wish to express my sincere thanks to Dr. Norharyati binti Harum, my supervisor for this project, for providing me with all the necessary information and resources to complete this project.

I take this opportunity to express gratitude to all of my friends and family members for their help and support. I also thank my parents for the unceasing encouragement, support and attention.

I also place on record, my sense of gratitude to one and all, who directly or indirectly, have lent their hand in this project.

ABSTRACT

Spectrum analyzer is a device used to measure the frequency of a signal but is a device that has a steep price on the market. To help reducing the cost of making a spectrum analyzer, a Raspberry Pi, which is known as a cheap small computer used for many projects can be used to make a portable spectrum analyzer. The objective of the project is to develop a radio spectrum analyzer using a Raspberry Pi and RTL-SDR. This project will use only 5v power supply to power it. An external battery bank can be used to power the project on the go. This characteristic makes this project an energy efficient and mobile solution. This project is targeted to be used as a teaching tool for secondary school student. The result as expected is to have a fully functional and widely documented portable spectrum analyzer done by using cheap and efficient solution.

ABSTRAK

Penganalisis Spektrum adalah alat yang digunakan untuk mengukur kekerapan isyarat tetapi merupakan sebuah alat yang mempunyai harga yang tinggi di pasaran. Untuk membantu mengurangkan kos membuat penganalisis spektrum , Raspberry Pi , sebuah lomputer kecil yang murah yang banyak digunakan untuk pelbagai projek , boleh digunakan untuk membuat penganalisis spektrum mudah alih. Objektif projek ini adalah untuk membangunkan penganalisis spektrum radio menggunakan Raspberry Pi dan RTL-SDR . Projek ini akan menggunakan hanya 5V bekalan kuasa untuk menghidupkannya . Sebuah bank bateri luaran boleh digunakan untuk menghidupkan projek ini di mana sahaja. Ciri-ciri ini menjadikan projek ini satu penyelesaian yang cekap tenaga dan mudah alih. Projek ini dijangka akan digunakan sebagai alat pengajaran untuk pelajar sekolah menengah. Hasil yang diharapkan adalah untuk mempunyai penganalisis spektrum mudah alih yang berfungsi sepenuhnya dan didokumenkan secara meluas dengan menggunakan penyelesaian yang murah dan cekap.

LIST OF TABLES

LIST OF FIGURE

DIAGRAM	TITLE	PAGE

CONTENTS

CHAPTER I

INTRODUCTION

1.1 Project Background

This project is about making a portable spectrum analyser using a Raspberry Pi and an RTL-SDR tuner. The goal of this project is to develop a low cost and portable spectrum analyser which is suitable for secondary school teaching tool. The requirement for this project is a Raspberry Pi Model B+, a RTL-SDR DVB-T TV tuner, and a touchscreen display.

Common spectrum analyser nowadays is very expensive and have a big bulky size. It is not suitable to be used for amateur user such as student who would like to explore the world of frequency spectrum. Added with its bulkiness in size, making the device more likely not used by the user due to its condition and the difficulties in using it. With this project, both the problem of expensive price and big bulky size can be resolved.

Raspberry Pi is a micro size computer that can do many things. The main purpose for the Raspberry Pi project inventor is to give the chance to school student to learn on how to code and make simple project using low cost hardware. The Raspberry Pi Model B+ used in this project has a processor that clocked at 700MHz backed up with a 512Mb of RAM. This neat specification is great for basic computing especially in the making of this portable spectrum analyser.

RTL-SDR is a software defined radio using the RTL2832U chipset. The developer of RTL-SDR has found that the signal I/Q data can be accessed directly from the common TV tuner which allow the device to be used as a wideband software defined radio. The usage of DVB-T sticks in this project let the tuner to scan the radio frequency in the air and convert the frequency into a frequency data to be shown on the portable spectrum analyser.

Another requirement in this project is a touchscreen display. To make the device as portable as it can, we need to eliminate those many knob and button found on the common spectrum analyser. To do this, we use a touchscreen display as the display can be used to interact with the user to make any selection and configuration to the analyser. The 4DPi-24-HAT display is very suitable for this project because of its small size and the touchscreen function it offers

1.2 Problem Statement

Table 1.1 shows the research problems are occurring. That also has been described in previous section.

Table 1.1 Summary of Problem Statement

PS	Problem Statement
1	Common spectrum analyser is very expensive and has a big bulky size making it not suitable for amateur such as school student to use

1.3 Project Question

Project Questions are found based on Problem Statements. Each project question is created to identify each problem statements as described in Table 1.2

Table 1.2: Summary of Project Questions

PS	PQ	Project Question
1	1	How to develop a fully functional and cheap spectrum analyzer?
	2	What are the available parameters of the developed spectrum analyzer?

1.4 Project Objective

Project Objective is found based on Problem Statements and Project Question. Each project objective is created as described in Table 1.3

Table 1.3: Summary of Project Objective

PS	PQ	PO	Project Objective
1	1	1	To study about performance parameters used in analyzing spectrum
		2	To develop a portable radio spectrum analyser using a Raspberry Pi and RTL-SDR tuner.
	2	3	To analyze radio frequency spectrum using the developed device, based on studied parameters

There are three project objectives that can be identified from this project which are listed as below:

PO 1: To study about performance parameters used in analysing spectrum

The main purpose of this project is to study about the spectrum analyser.

PO2: To develop a portable radio spectrum analyser using a Raspberry Pi and RTL-SDR tuner.

This project will develop a low cost portable spectrum analyser to overcome the problem of expensive and big common spectrum analyser.

PO3: To analyze radio frequency spectrum using the developed device, based on studied parameters

By the end of this project, the device created should be able to analyse radio frequency just like the common spectrum analyser can while having a small size and low cost.

1.5 Scope

The scope of this research will focus on some issues as stated below:

1. A Raspberry Pi is needed to be used as the central processing unit of the developed device.
2. A DVB-T TV tuner dongle based on the RTL2832U chipset to be used as the device that capture the frequency that need to be analysed.
3. A small touch screen to view the output of the spectrum analyzed as we would like to develop a portable device.

1.6 Project Significant

This project is about a developing a cheap and small portable spectrum analyser using a Raspberry Pi and RTL-SDR tuner. This developed device should be very useful for school use and as a learning device because of its simple and easy to use interface.

1.7 Conclusion

This chapter helps to understand what the project background, scope of the project, objective that need to be achieve and problem occurred before started this project. This project will create a low cost portable spectrum analyser using a Raspberry Pi and RTL-SDR tuner

CHAPTER II

LITERATURE REVIEW

2.1 Introduction

In this chapter, we will make some reviews on a topic that has the same research field as the spectrum analyser and Raspberry Pi. Hence, this chapter will expose the related work, critical review of current project and justification, proposed solution and conclusion. After reviewing them, we will analyse the current problem with the common spectrum analyser and its disadvantages. Lastly, a solution for the problem will be proposed.

2.2 Related Topic

2.2.1 Spectrum Analyser Basics

Based on an article titled spectrum analyser basics by Agilent Technologies [1], the article write about the fundamental of a spectrum analyser. It gives an overview of what is a spectrum analyser, the measurement that it can make, the theory of operation of the device, and the specific hardware of the spectrum analyser.

A spectrum analyser is a device that can analyse the electrical signal that are passing through or being transmitted into the device. Spectrum analyser can show signal information such as voltage, power, period, wave shape, sideband, and frequency of a signal. In order to make the signal shown unaltered, we need a passive receiver that doesn't do anything to the signal and make it display the signal as it is.

The basics measurement that a spectrum analyser can make are: modulation, distortion, and noise. Measurement of modulation is important to make sure that a system is working properly and the information is being transmitted correctly. Distortion is a measurement to make sure that the signal is free of intermodulation distortion to prevent signal crosstalk. Lastly is noise which is the most often signal to be measured because by measuring the noise, the performance of a device can be measured.

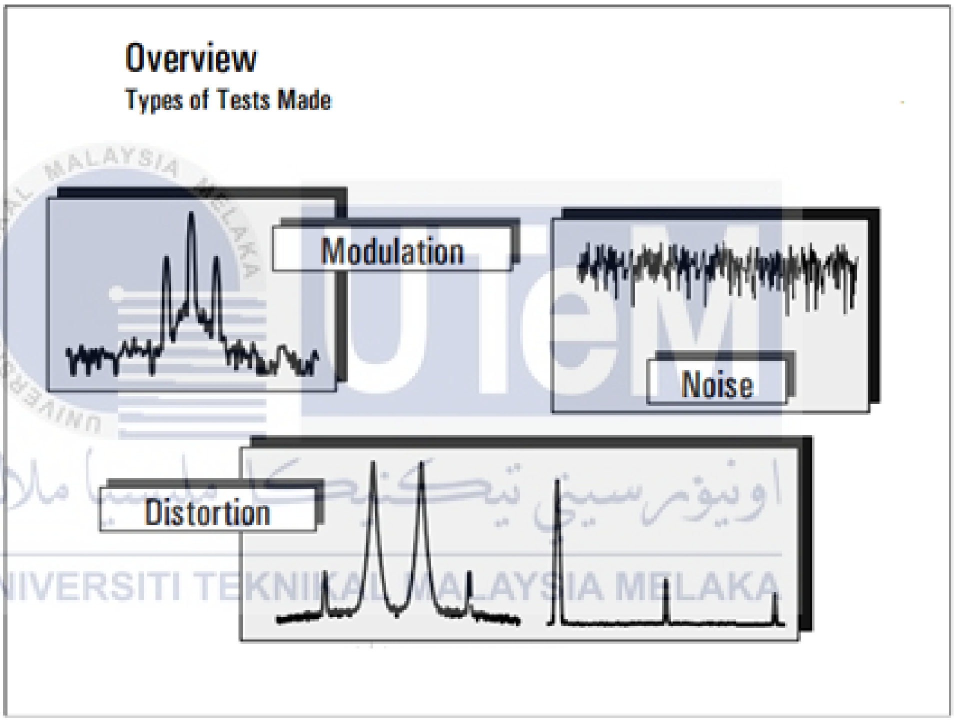


Figure 2.1: Basic Measurement of a Spectrum Analyser

The major component in a spectrum analyser are the RF input attenuator, mixer, IF (intermediate frequency) gain, IF filter, video filter, sweep generator, detector, local oscillator, and lastly an LCD display or any type of display. A mixer is a three port device that converts a signal from one frequency to another. A detector converts the IF signal to a baseband or video signal so it can be digitized and then viewed on the display. The display does what it should do which is displaying the converted and processed signal.

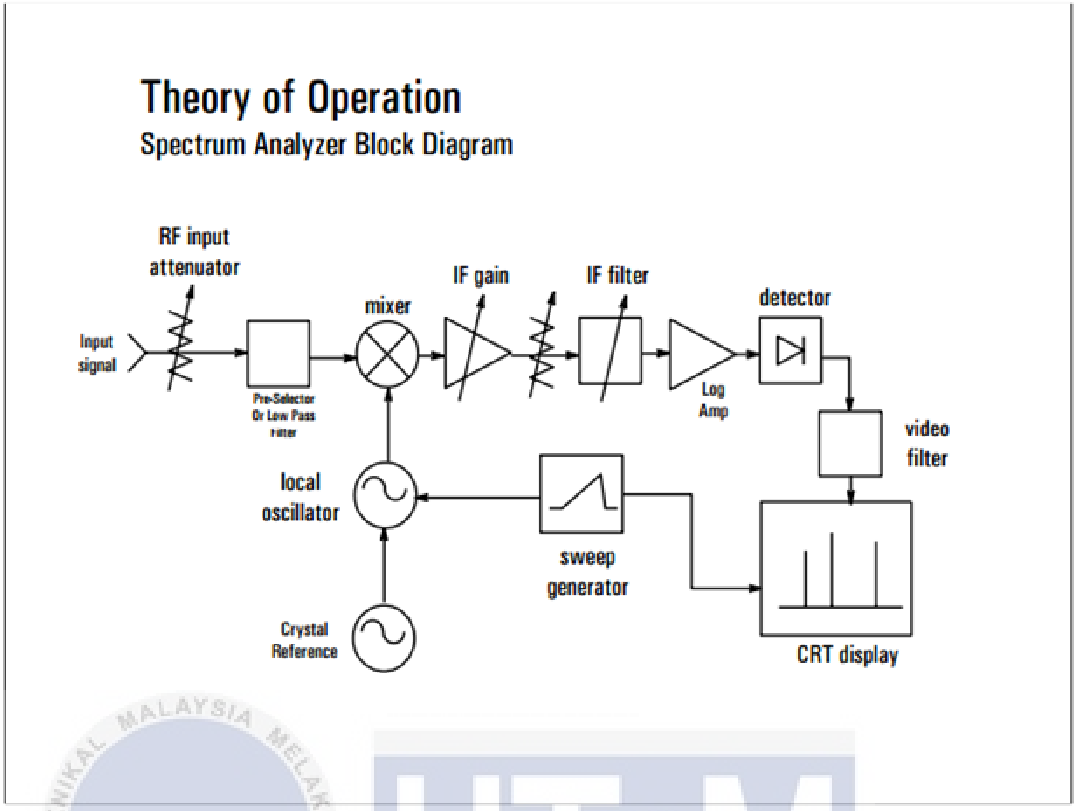


Figure 2.2: Theory of Operation of a Spectrum Analyser

2.2.2 About RTL-SDR

RTL-SDR is a very cheap software defined radio that uses a DVB-T television tuner dongle that are based on the RTL2832U chipset made by Realtek. Antti Palosaari, Eric Fry and Osmocom combined efforts found that the signal I/Q data could be accessed directly, which allowed the DVB-T TV tuner to be converted into a wideband software defined radio via a new software driver[2].

A software defined radio is a software based hardware. Modulators, demodulators, and amplifiers can be implemented into software. This enables easy signal processing and thus make a cheap wide band scanner radios to be produced.

RTL-SDR radio scanner can be applied to so many applications. Some of the most famous include listening to unencrypted police, ambulance, or fire conversations.

Another one is as simple as listening to fm radio signal. A more complex application is using the RTL-SDR to listen to satellites and the ISS signal from up above.

The specification of RTL-SDR may vary depending on the type and manufacturer of the DVB-T Tuner itself but most of the tuner scan on the same range of frequency. Most tuner frequency range is between 25MHz to 1500MHz. The sample rate is 2.4 MS/s at stable but can be crancked up to 3.2 MS/s with a possibilities of dropping samples. The native resolution of the tuner is 8 bits. The impedence of the tuner is about 75 ohms

2.2.3 What is Raspberry Pi

Raspberry Pi is a computer with a credit-card size designed for education created by Eben Upton which he says is inspired by the BBC Microcomputer [3]. The main goal of the device is to create a low-cost device that would improve programming skills and understanding at school level. Because of its low-cost, many people tinker with it and make a creative and unique electronic project based on it. If compared side by side with a modern laptop hardware, it underperforms very much but the application of the computer is only for small project so it gets the job done as it should do.

The Raspberry Pi is an open hardware, meaning that many of the project made using the Raspberry Pi are open and free to be used by anyone. Many of the project is also well documented so building and modifying it should be a very easy step.

As of today, there are 4 different model of Raspberry Pi. The lowest will be the Raspberry Pi 1 which sport a 700 MHz single-core ARM11 processor paired with a 256MB for model A and 512MB for model B of random access memory. Next is the Raspberry Pi 2 which have a 900MHz quad-core ARM cortex-A7 processor paired with a 1GB of RAM. The latest one is Raspberry Pi 3 with a 1200MHz quad-core ARM cortex-A53 and 1GB of RAM. All the 3 version comes with 4 USB port, an Ethernet port, HDMI display output, and 40 pin GPIO pin. The last one is Raspberry Pi Zero which target at embedded system and have smaller size compared to the other Raspberry Pi. It lacks full size USB and HDMI port and has no Ethernet port at all [4].

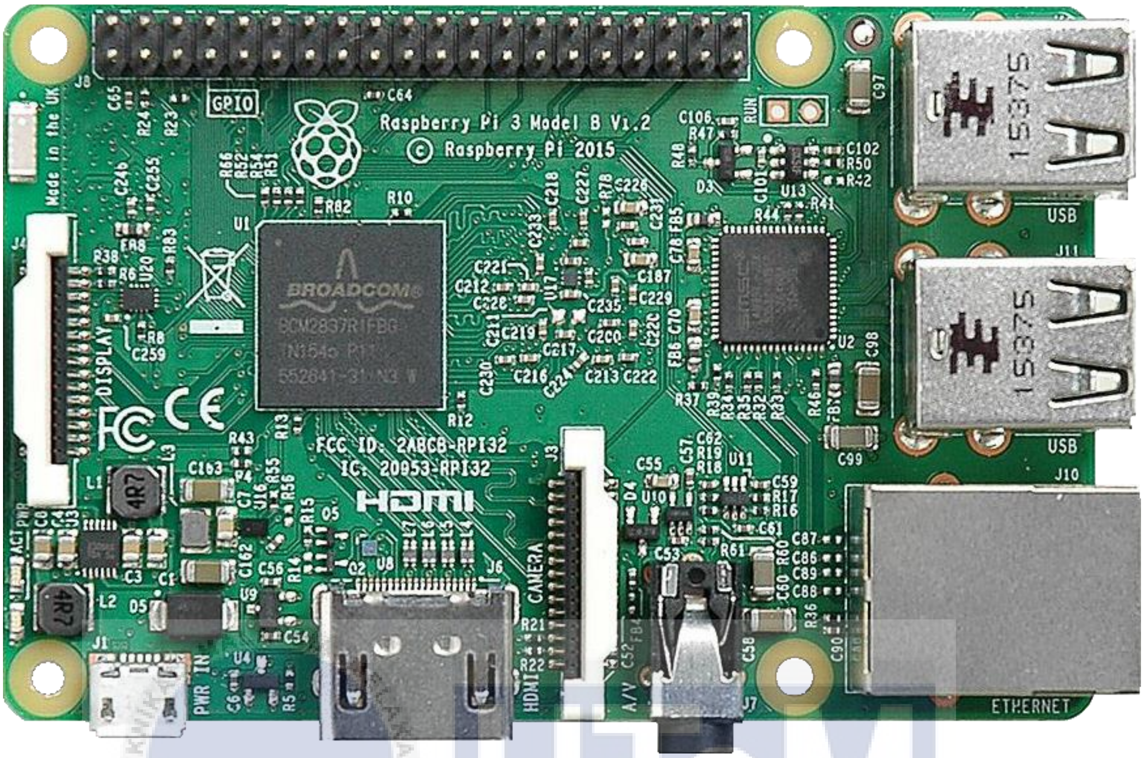


Figure 2.3: Raspberry Pi 3 model B

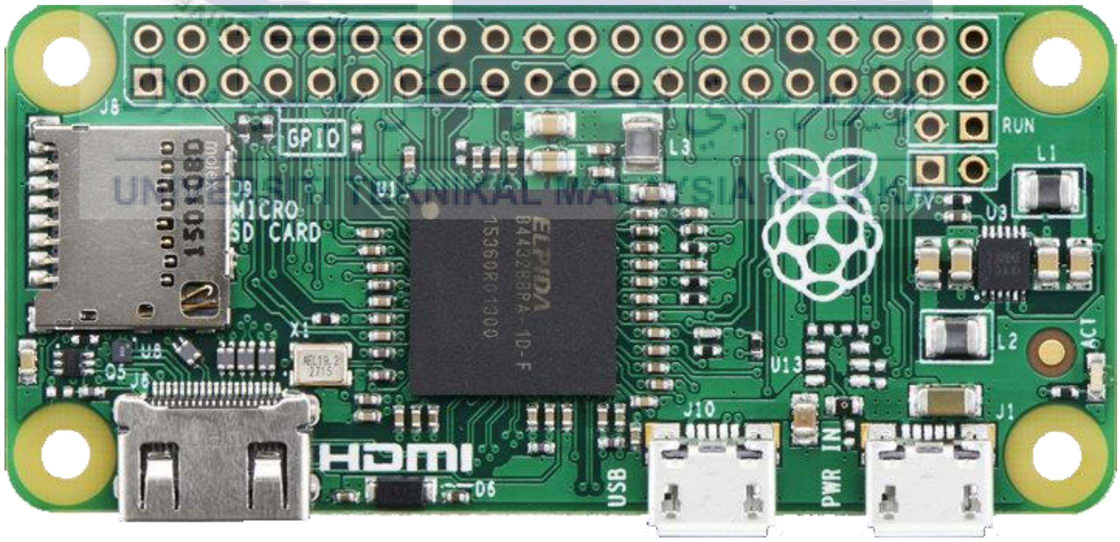


Figure 2.4: Raspberry Pi Zero

Another neat addition to the Raspberry Pi is the display. An example of a Raspberry Pi supported display is 4DPi-24-HAT. This display can output a resolution of 320x240. What more attractive is the display also include touchscreen and only connect to the Raspberry Pi via the 40 pin GPIO pin. This make the display a very ideal solution for a touch based project.

Figure 2.5: a 4DPi-24-HAT attached to Raspberry Pi

2.2.4 Analysed parameters

Radio frequency(RF) is electromagnetic wave frequencies that lie in the range extending from around 3kHz to 300GHz [8]. This include the frequency used for communications and radar signals. Unit for radio frequency is hertz(Hz). Radio frequency is usually refers to electrical oscillation rather than mechanical oscillations.

Frequency bandwidth is the difference between upper and lower frequencies in a set of frequencies [9]. Usually, it is measured in hertz(Hz). Bandwidth using the unit hertz is used in many fields, such as electronics, information theory, digital communication, and radio communications. In our spectrum analyser, the bandwidth is the x axis of the graph produced by the device.

Frequency gain can be described as how well the antenna converts input power into radio waves[10]. Antenna gain is usually described as the ratio of power to the power produced by the antenna. The unit used for gain is decibels(dB). In spectrum analyser, the gain is the ratio of conversion of radio frequency to radio waves.

2.3 Related Research

Based on the article "**Freq Show: Raspberry Pi RTL-SDR Scanner"** by Tony DiCola[5]. The article explains about how the writer setup and create his own frequency scanner using a Raspberry Pi and RTL-SDR scanner. The author put a step by step guide on the setup process of his project. He first installs the operating system on the Raspberry Pi and then create the applications using python language. Some configuration is made to make the application run such as the screen size and touchscreen input. The last chapter describe the usage of the application created. The application can show an intensity versus frequency graph on the display. It also can show a waterfall plot of the intensity of the frequency versus time. On the configuration tab, you can control the centre frequency to be shown, the sample rate to control how wide the range of the frequency to be shown on the graph, the gain of the tuner to remove the noise and make weak signal more easily visible, and the minimum and maximum intensity value shown on the graph. The project uses python as its programming language.

Slaven Krilic in his article "**Monitoring Network Manager for RTL-SDR**" [6] writes about how he create a software that allows you to set up a remote Raspberry Pi embedded computer with an RTL-SDR dongle attached and access it remotely through a Windows PC GUI. The article shows very in depth features of the software created such as listening to audio from Remote Raspberry Pi by streaming back from station to desktop. Another features included is the ability to scan for RF bands spectrum and shows it in spectrum and waterfall diagram. The system consists of a management station and a monitoring station communicating in real-time over the network. The monitoring system is a networked Raspberry Pi running Raspbian operating system and a RTL-SDR adapter while the management station is a windows PC installed with a MNM4SDR software created.

The article "**ViewRF software – RTL-SDR Spectrum Analyser for BeagleBone Black**" by Stephen Ong are very interesting [7]. The article briefly described about how the author create a RTL-SDR based spectrum analyser running on the BeagleBone Black microcomputer with a 7-inch touchscreen LCD. The software created named ViewRF which is licensed under GPL licence can be downloaded on GitHub. The software has been designed to run on an 800x480 resolution screen which is the default resolution of the 7-inch LCD used in the project. The spectrum analyser can shows the RF spectrum of various sources such as car remote transmitter, Analog PAL signal, and even GSM 900 frequency spectrum. The project uses C as the programming language.

2.4 Critical Review of Current Problem and Justification

Table 2.1 Summary of Critical Review of Current Problem and Justification

Article Title	Purpose	Description	Problems
Freq Show:Raspberry Pi RTL-SDR Scanner Author: Tony DiCola	Explains about the project and the software used by the author	• Step by step guide on how the project is done. • List of the features and usage of the application • Brief specification of the software created and its limitation	• The project only uses a RTL-SDR adapter so the frequency range of the spectrum shown is limited to the range of the tuner used which is 24 MHz to 1,850 Mhz.

Monitoring Network Manager for RTL-SDR Author: Slaven Krilic	Explains about the project and the software used by the author	• Step by step guide on how the project is done. • List of the features and usage of the application • Brief specification of the software created and its limitation	• The project uses a Raspberry Pi and RTL-SDR but also need a full sized Windows PC to operate the MNM4SDR software. This make the project less portable
ViewRF software – RTL-SDR Spectrum Analyser for BeagleBone Black Author: Stephen Ong	Explains about the project and the software used by the author	• Step by step guide on how the project is done. • List of the features and usage of the application • Brief specification of the software created and its limitation	• The project has the same function as FreqShow but uses a BeagleBone Black microcomputer which has a higher price compared to Raspberry Pi

2.5 Proposed Solution/ Further Project

Based on the critical review, a portable spectrum analyzer can be made using a Raspberry Pi, RTL-SDR receiver, and a touchscreen display. The rationale of using a Raspberry Pi is because it is the cheapest solution for a microcomputer aside from its ease of using it. A touchscreen display is selected because it should make the portable and easy to handle. A BeagleBone Black microcomputer is not chosen because it is pricier that Raspberry Pi. This project of a portable spectrum analyzer should be useful for use in school as a teaching tool for school student.

2.6 Conclusion

Basically by reviewing literature review, we will get more information from the previous research and it will help to find the solution for problem that we may face. A portable spectrum analyser is suitable for use as a teaching tool for school student and as a Portable Alternatives for the big and bulky common spectrum analyser. The next chapter will be focusing about project methodology. Which will be covering about the method use to complete the project.

CHAPTER III

PROJECT METHODOLOGY

3.1 Introduction

This chapter explains the methodology of approach used in the making of this project. The main important thing in developing a project is the selection of methodology, methods, technique and tools to make sure that the project work as planned and finished according to the time. In this project, I will be using the waterfall model as the methodology in order to complete the project.

3.2 Methodology

This project will be using Waterfall methodology because it fit with the development process of the project. This methodology has each phase with a specific deliverables and a review process which give us time to focus on each of the phases. This methodology also works well for a project where requirements are very well understood. The process for this project is show in the figure 3.1.

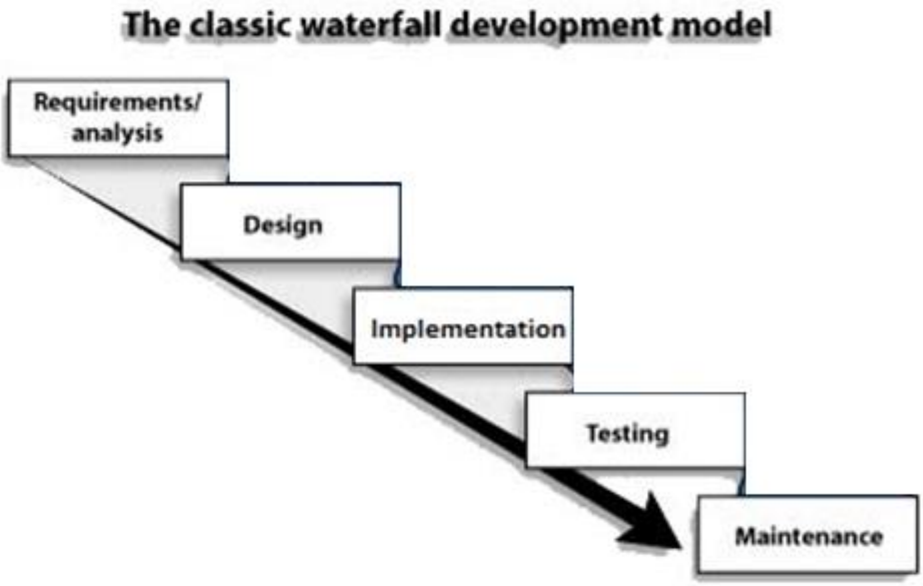


Figure 3.1: Waterfall Methodology Architecture

3.2.1 Requirement/Analysis

Firstly, we need to make a thorough analysis on how to make the project works. In this process, the major information about the project should has been identified. As early as making the proposal, those are included as the analysis process because in the proposal we already specify the purposes, the hardware and software requirement, and the objective which is the most important part of the project. Listed is the needed hardware and software needed to be used in this project.

1. Hardware
    - Raspberry Pi Model B+ board
    - DVB-T TV Tuner
    - 4DPi-24-HAT 2.4” touchscreen display
    - Portable Battery Bank

2. Software

    - Raspbian Operating System
    - Python Programming Language

3.2.2 Design

In this process we need to design the project to make sure is work with the information gained from analysis part before. This project purpose is to develop and build a portable spectrum analyser using a Raspberry Pi and RTL-SDR tuner. The design process will show the design of the project and where the component used in this project is placed.

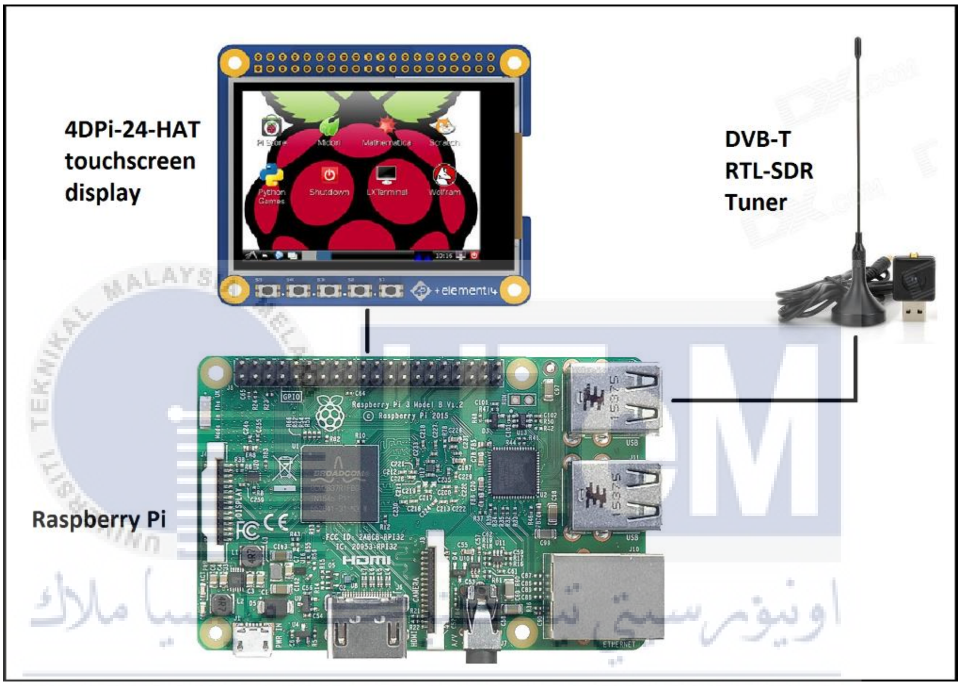


Figure 3.2: Portable Spectrum Analyser System using Raspberry Pi

3.2.3 Implementation

This level of process will implement all the early process to build the project. Firstly, we need to connect all the equipment. Connect the Raspberry Pi to the RTL-SDR tuner and hook a monitor and keyboard. The Raspberry Pi need to be connected to monitor first to do all the programming because it easy to make and troubleshoot the programming error in a bigger display. After all is done, then we connect the touchscreen display and the program should run on the small touchscreen display.

3.2.4 Testing

This process will conduct a series of testing of what we have done in implementation process. The testing process will be conducted phase by phase. We will test all the program to make sure it works as it should. Then only we will try to connect it to the touchscreen display and test the interface in a touchscreen environment. If all the assets in the program is loaded according to the right size, then only we can finalize the developed device.

Figure 3.3: The Illustration of Testing process phase

3.2.5 Maintenance

This process is the last process in the methodology. It should include the maintenance of the device such as making sure the program load correctly and making sure the touchscreen work correctly. If the program does not load correctly, we can read the source code again to check for any problem that may occur or any impersistence in the code. If the touchscreen does not work correctly, we can try to calibrate the touchscreen to make sure it works flawlessly.

3.3 Project Milestone

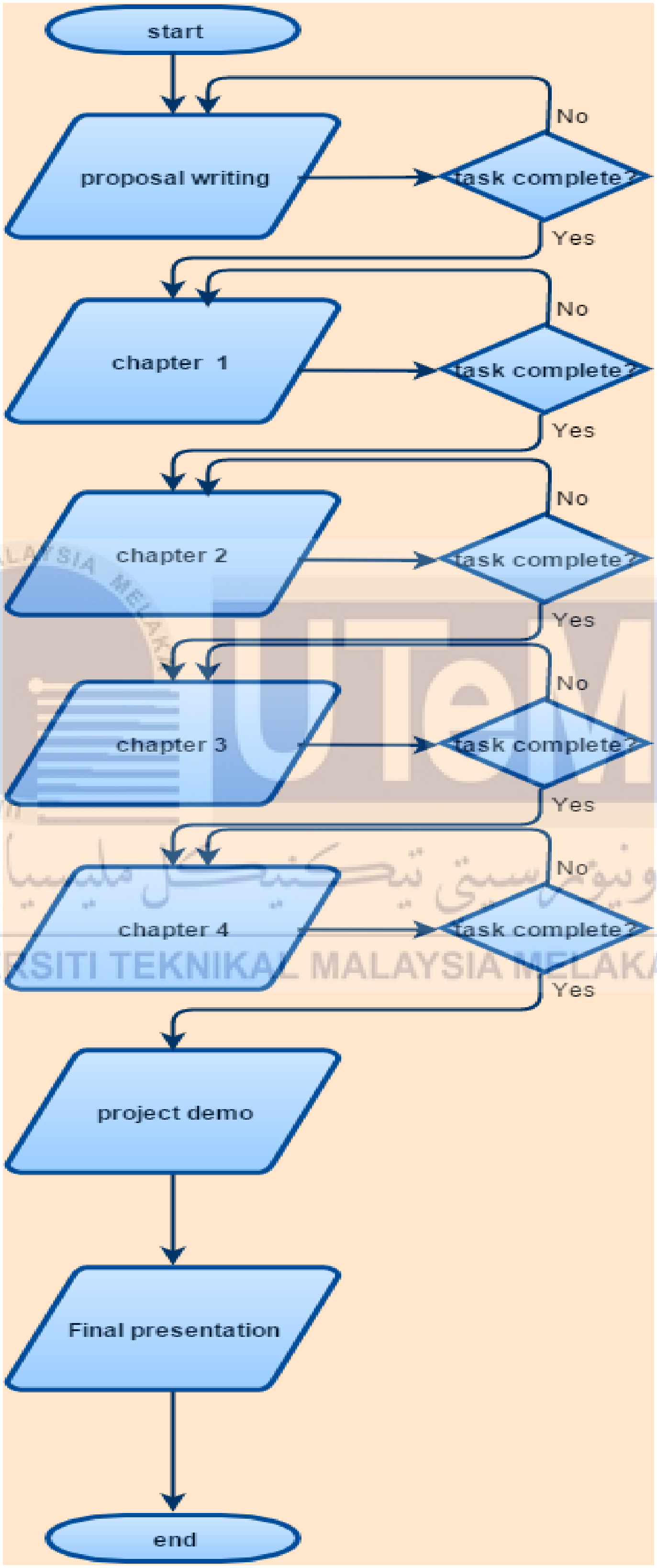


Figure 3.4: Flow Chart of Project Activities

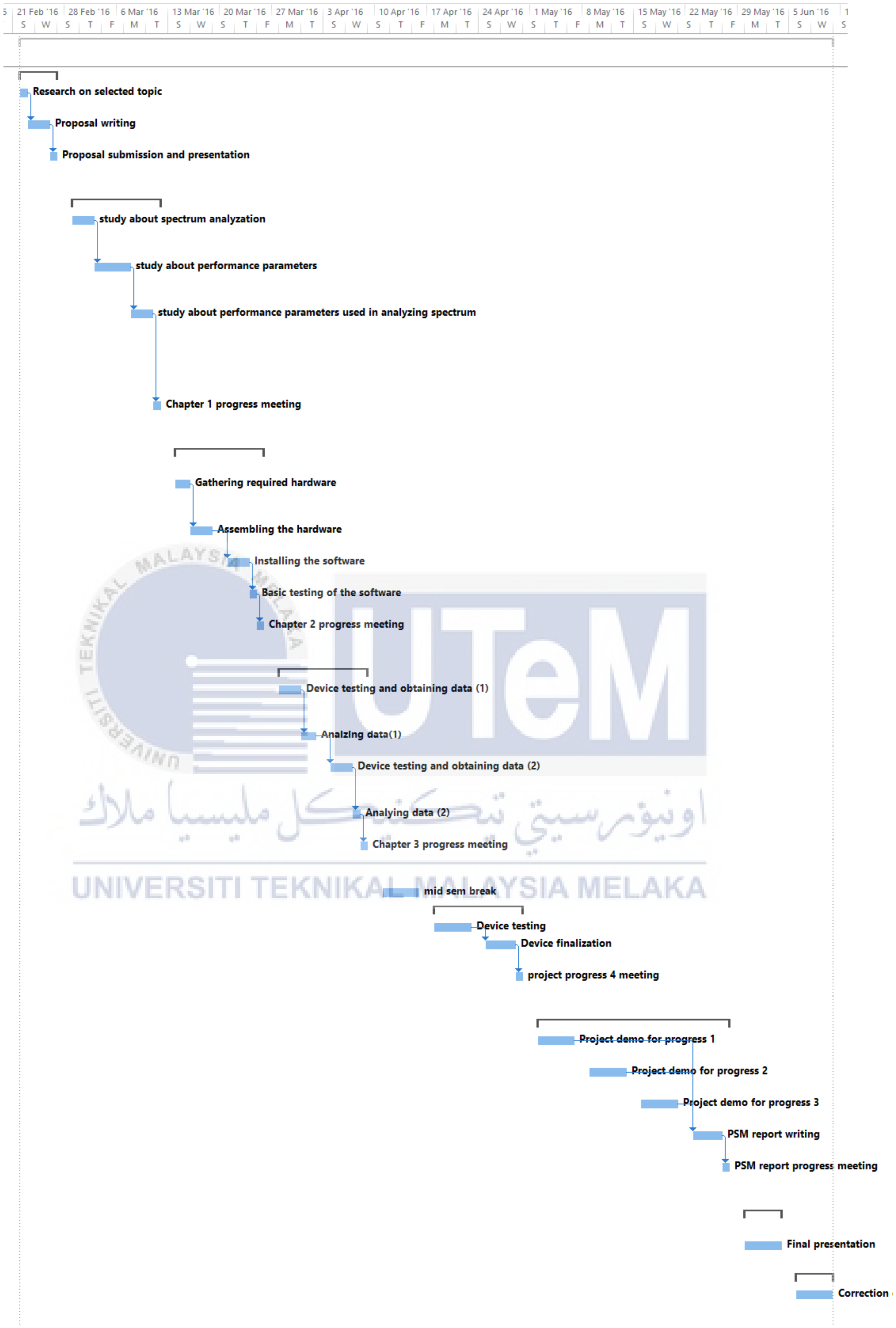


Figure 3.5: Gantt chart of Project Activities

Table 3.1: The Milestone

Task Name	Duration	Start	Finish
Final Year Project Development	**80 days**	**Mon 22/2/16**	**Fri 10/6/16**
Proposal	**5 days**	**Mon 22/2/16**	**Fri 26/2/16**
Research on selected topic	1 day	Mon 22/2/16	Mon 22/2/16
Proposal writing	3 days	Tue 23/2/16	Thu 25/2/16
Proposal submission and presentation	1 day	Fri 26/2/16	Fri 26/2/16
Chapter 1	**10 days**	**Mon 29/2/16**	**Fri 11/3/16**
study about spectrum analyzation	3 days	Mon 29/2/16	Wed 2/3/16
study about performance parameters	3 days	Thu 3/3/16	Mon 7/3/16
study about performance parameters used in analysing spectrum	3 days	Tue 8/3/16	Thu 10/3/16
Chapter 1 progress meeting	1 day	Fri 11/3/16	Fri 11/3/16
Chapter 2	**10 days**	**Mon 14/3/16**	**Fri 25/3/16**
Gathering required hardware	2 days	Mon 14/3/16	Tue 15/3/16
Assembling the hardware	3 days	Wed 16/3/16	Fri 18/3/16
Installing the software	3 days	Mon 21/3/16	Wed 23/3/16
Basic testing of the software	1 day	Thu 24/3/16	Thu 24/3/16
Chapter 2 progress meeting	1 day	Fri 25/3/16	Fri 25/3/16
Chapter 3	**10 days**	**Mon 28/3/16**	**Fri 8/4/16**
Device testing and obtaining data (1)	3 days	Mon 28/3/16	Wed 30/3/16

Analysing data(1)	2 days	Thu 31/3/16	Fri 1/4/16
Device testing and obtaining data (2)	3 days	Mon 4/4/16	Wed 6/4/16
Analysing data (2)	1 day	Thu 7/4/16	Thu 7/4/16
Chapter 3 progress meeting	1 day	Fri 8/4/16	Fri 8/4/16
mid semester break	5 days	Mon 11/4/16	Fri 15/4/16
Chapter 4	**10 days**	**Mon 18/4/16**	**Fri 29/4/16**
Device testing	5 days	Mon 18/4/16	Fri 22/4/16
Device finalization	4 days	Mon 25/4/16	Thu 28/4/16
Chapter 4 progress meeting	1 day	Fri 29/4/16	Fri 29/4/16
Project demo	**20 days**	**Mon 2/5/16**	**Fri 27/5/16**
Project demo for progress 1	5 days	Mon 2/5/16	Fri 6/5/16
Project demo for progress 2	5 days	Mon 9/5/16	Fri 13/5/16
Project demo for progress 3	5 days	Mon 16/5/16	Fri 20/5/16
PSM report writing	4 days	Mon 23/5/16	Thu 26/5/16
PSM report progress meeting	1 day	Fri 27/5/16	Fri 27/5/16
Final presentation	**5 days**	**Mon 30/5/16**	**Fri 3/6/16**
Final presentation	5 days	Mon 30/5/16	Fri 3/6/16
Correction	**5 days**	**Mon 6/6/16**	**Fri 10/6/16**
Correction draft report based on supervisor's and evaluator's comments during the final presentation	5 days	Mon 6/6/16	Fri 10/6/16

3.4 Conclusion

This chapter explained all the processes and activities that involved in this project. In this project the methodology that are used is Waterfall methodology which represent five process cycle which is analysis, design, implementation, testing, and maintenance. Each process need to be done before continue to other processes. All the processes are dependent to it process before and the chain continue until this project achieve its objectives.

CHAPTER IV

DESIGN

4.1. Introduction

This chapter explains about elements that needed to design this project. This project main objective is to develop and build a portable spectrum analyser. In this chapter, the design of the component used in this build will be specified. Also specified is how the hardware and software is combined together to make the whole project work as it should. The possible scenarios that needed to define two possible scenarios and make justification are also included.

4.2. System Architecture

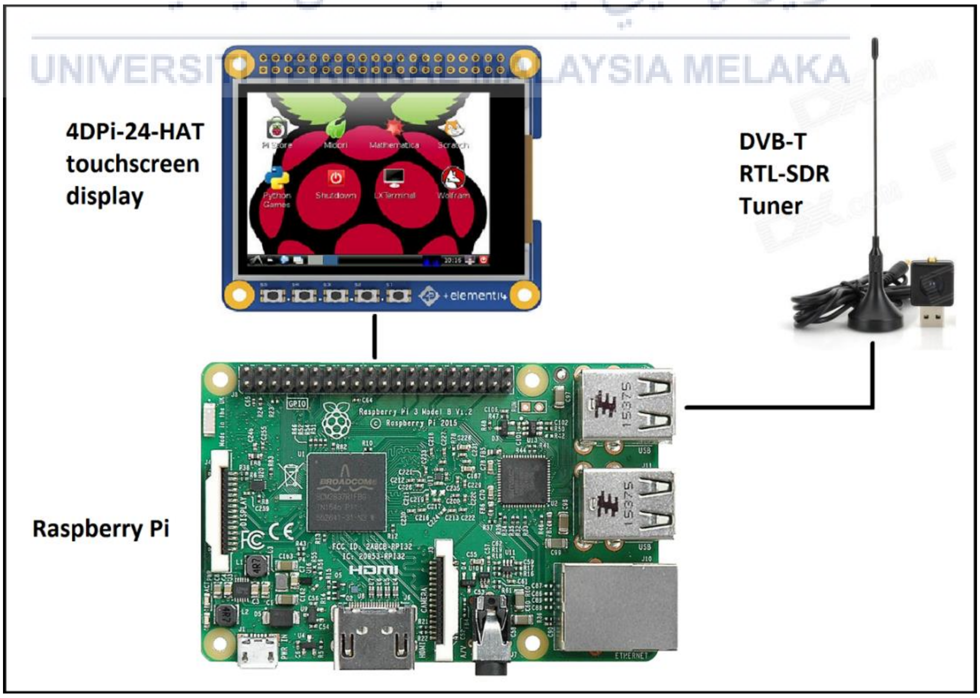


Figure 4.1: The Illustration of the project hardware

Figure 4.1 shows how the hardware is connected. The 4DPi-24-HAT touchscreen display is connected to the Raspberry Pi board via the 40 pin GPIO pin while the RTL-SDR DVB-T TV tuner is connected to the Raspberry Pi through the USB port.

4.2.1 Software Requirement

4.2.1.1 Raspbian operating System

After setting up and connecting the hardware, we need to install the operating system to make the system boot. The preferred operating system for a Raspberry Pi is Raspbian. Although there are many other operating system that can be installed on the Raspberry Pi, Raspbian is the most favourable because it is the officially supported operating system by the Raspberry Pi foundation. Raspbian is also very well documented so if any problem encountered, the solution may be found easily.

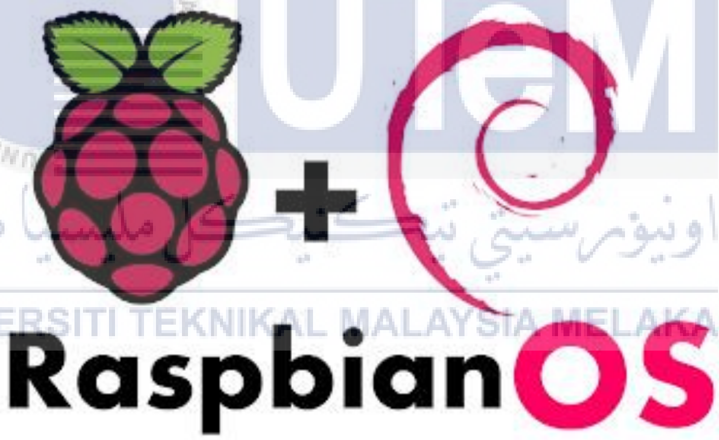


Figure 4.2: Raspbian Logo

4.2.1.2 Python Programming Language

After the operating system installation is completed, we can start to code on the Raspberry Pi. As for the programming language, this project will use python because it is easy to use and understand. The codes will be saved into the SD card so we don't need to rewrite the code every time.

Figure 4.3: Python Logo

4.2.2 Hardware Requirement

4.2.2.1 Raspberry Pi

Raspberry Pi is the main board and the main processing unit for this project. It serves as the central processing unit for the device developed. It will be connected to the touchscreen display via the 40 pin GPIO connecter on the board. The RTL-SDR tuner will be connected through USB cable to the board

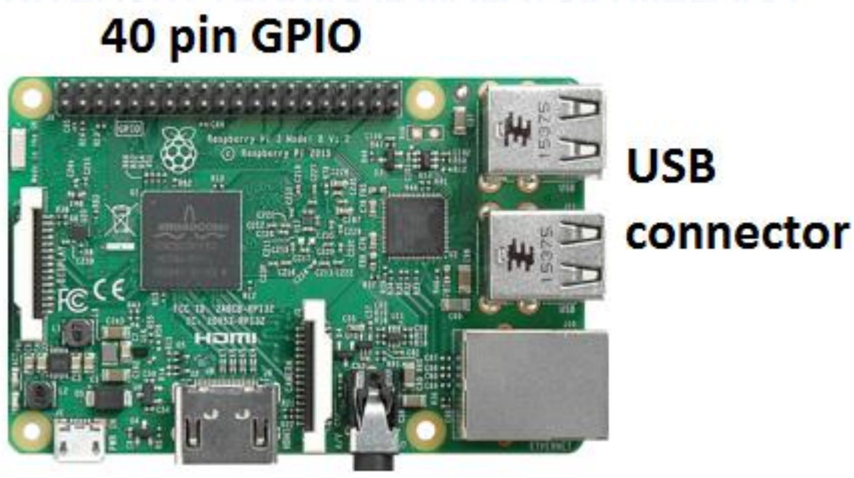


Figure 4.4: Raspberry Pi Model B+ for this project

4.2.2.2 RTL-SDR DVB-T Tuner

The DVB-T stick Tuner is used in this project as the device that capture the wireless signal frequency. The device used the RTL2832U chipset. Initially, this device is used to watch TV and radio but we can repurpose the signal to be shown as a frequency versus intensity graph.

Figure 4.5: RTL-SDR DVB-T TV Tuner

4.2.2.3 4DPi-24-HAT Touchscreen display

For this project, the 2.4-inch touchscreen display is used as the input and also the output device for the portable spectrum analyser created. The touchscreen act as the input device by receiving touch input from the user and also show the frequency versus intensity graph plotted on the screen.

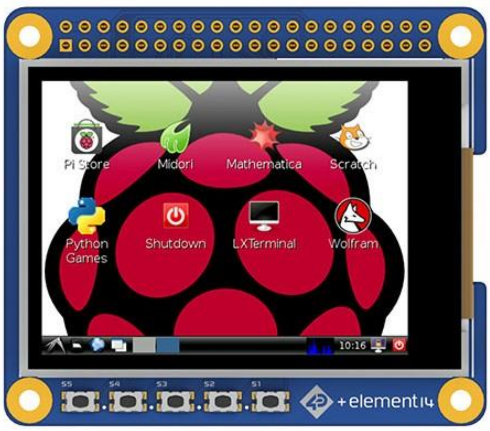


Figure 4.6: The 4DPi-24-HAT touchscreen display

4.3. Flow Chart and Physical Design

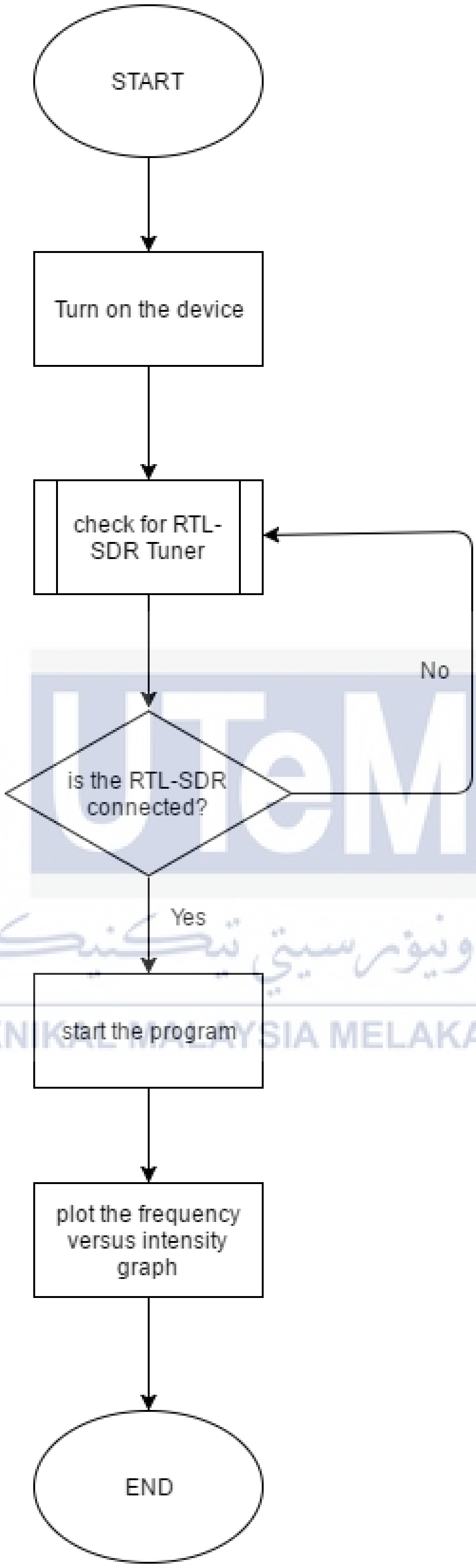


Figure 4.7: Flow Chart for the Portable spectrum analyser

Figure 4.7 show the flowchart for the project. When the device is turned on, it will check for the presence of the RTL-SDR tuner. If it is not connected, the program will not start. Only when the tuner is connected, then the program will run and start plotting graph of frequency versus intensity based on the captured wireless signal by the RTL-SDR tuner.

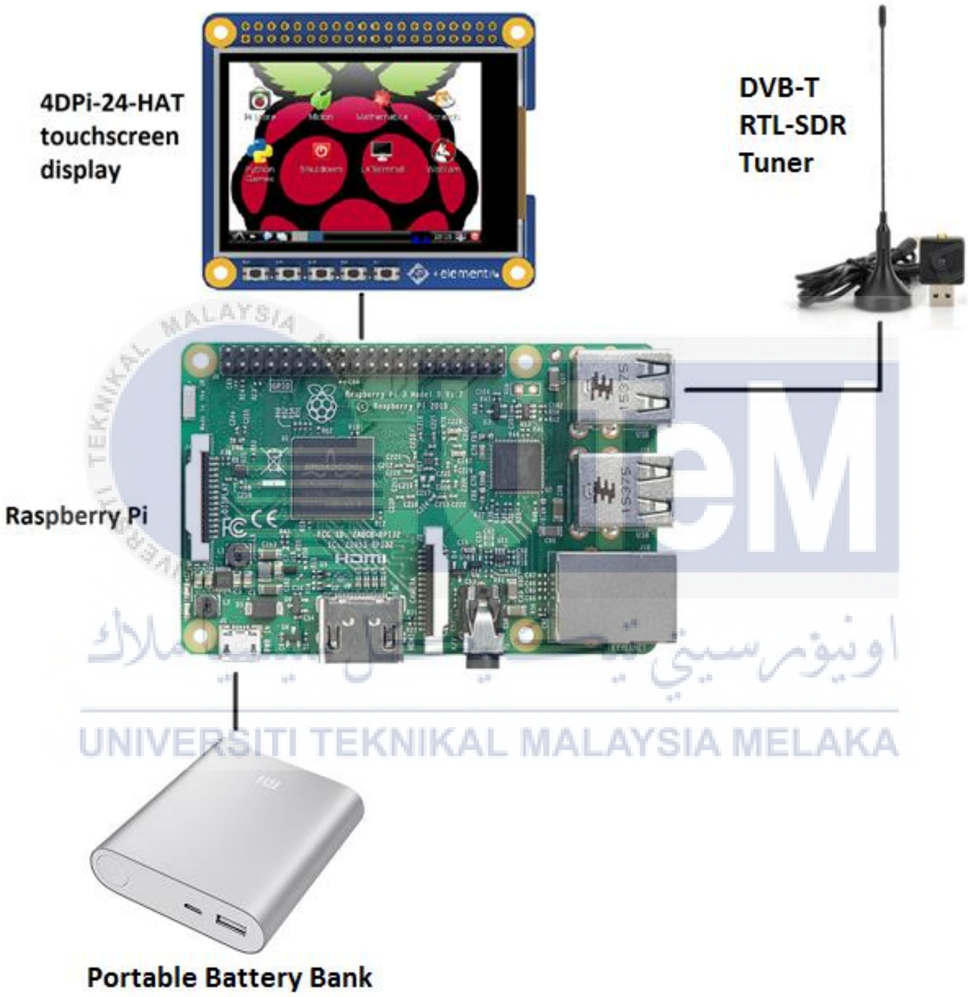


Figure 4.8: Physical Design for Portable spectrum analyser

Figure 4.8 show the physical design for the portable spectrum analyser. The 4DPi-24-HAT display is connected to the board via the 40 pin GPIO port. The RTL-SDR DVB-T Tuner is connected to the board through the USB port. Lastly, a power supply is needed to power the project alive. A portable battery bank can also be used to power the project on-the-go.

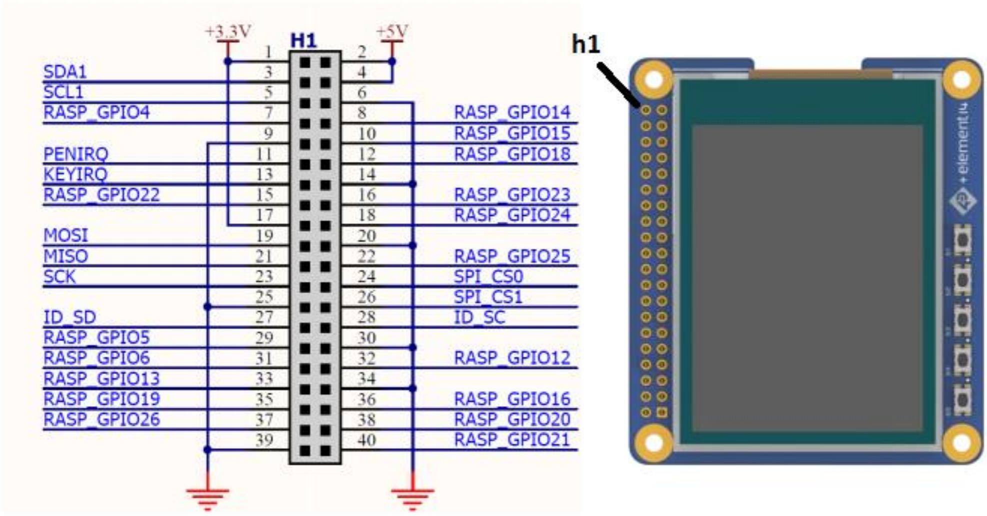


Figure 4.9: GPIO pin Layout

Figure 4.9 show the GPIO pin connection between the 4DPi-24-HAT touchscreen display and the 40 pin GPIO connector on the Raspberry Pi. As the display has a female GPIO port while the Raspberry Pi has a male GPIO port, the installation is as simple as sliding in the connecter. Do note the h1 port on the display and the Raspberry Pi must be the same.

4.4 Conclusion

In conclusion, the design phase is to show the requirements of hardware and software, the overall system of this project, logical and physical diagram, the design of scenario that is possible for this project and the metric measurement that is used for the output result in this project. All the information that has been gathered in this phase will be implements in next process which is implementation.

CHAPTER V

IMPLEMENTATION

5.1. Introduction

This chapter explains about the implementation of this project. This chapter consist of three main topics. Firstly, the hardware setup which is divided to another two subtopic named hardware requirement and hardware installation. The next topic is software setup that contain three part which is operating system installation, Freqshow installation, and 4DPi-24-HAT Driver installation. The last part is Freqshow configuration setup. Figure 5.2 shows the summary of this chapter.

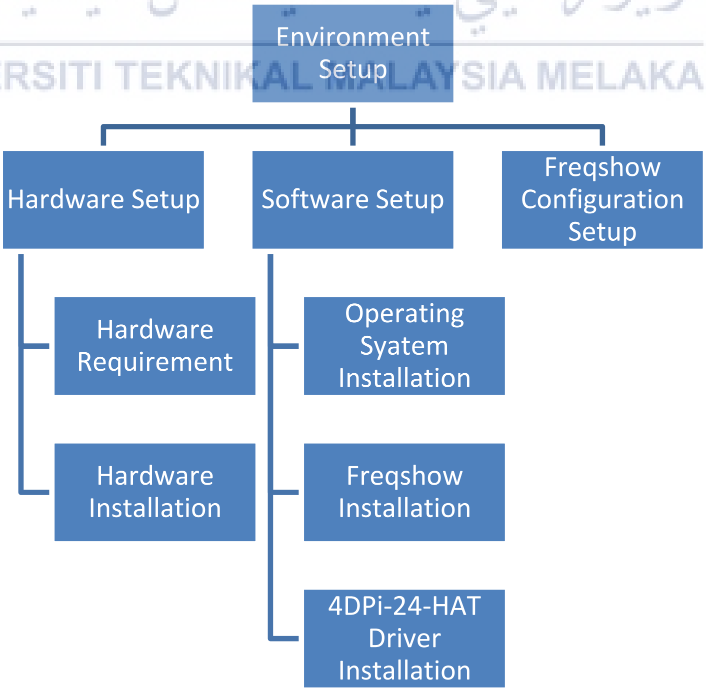


Figure 5.1: Implementation chapter summary

5.2. Environment Setup

5.2.1 Hardware Setup

5.2.1.1 Hardware Requirements

- Raspberry Pi Model B+
- USB Mouse
- USB Keyboard
- Monitor with HDMI input
- Power Supply
- RTL-SDR DVB-T Tuner
- 4DPi-24-HAT Touchscreen display

5.2.1.2 Hardware Installation

i. Connect all the peripheral needed for the setup process which is mouse and keyboard via the USB port and the monitor via the HDMI port.

ii. Connect the RTL-SDR DVB-T tuner to the USB port.

iii. The 4DPi-24-HAT Touchscreen display should be connected to the raspberry pi through the 40 pin GPIO.

iv. Connect the raspberry pi to the internet through the RJ-45 port.

v. Lastly, connect the raspberry pi to power outlet.

Figure 5.2: Physical diagram of the project

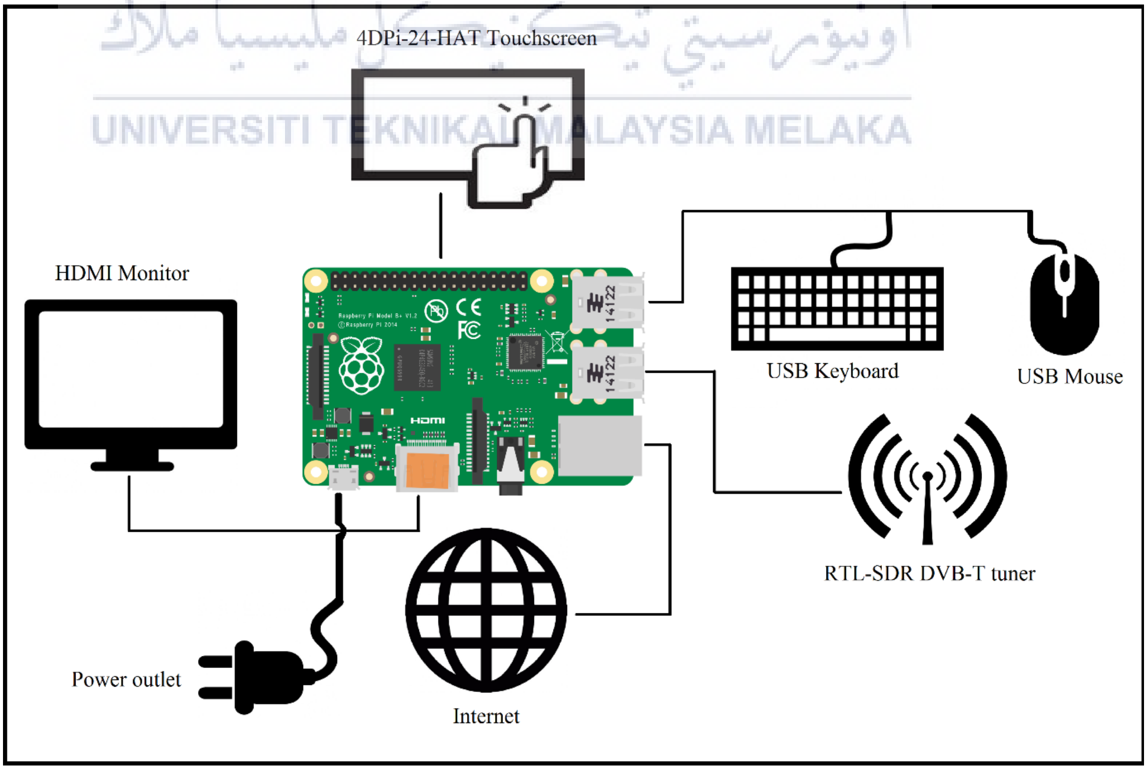


Figure 5.3: Logical diagram of the project

5.2.2 Software Setup

5.2.2.1 Operating System Installation

Raspbian is a Linux based operating system based on Debian Linux. All the installation commands will be based on Linux commands.

i. Once the hardware setup finished, download the latest Raspbian OS from the website *https://www.raspberrypi.org/downloads/noobs/*

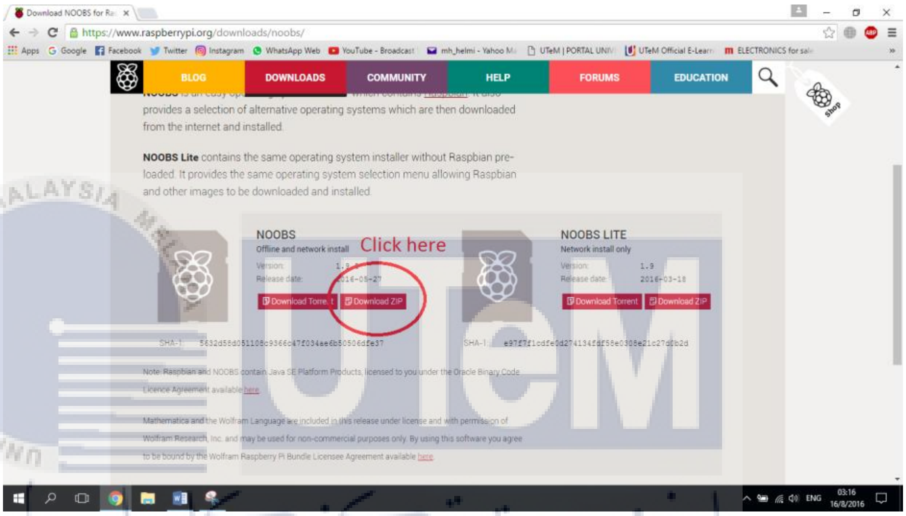


Figure 5.4: Raspbian download page

ii. Extract the downloaded file into the SD card root directory

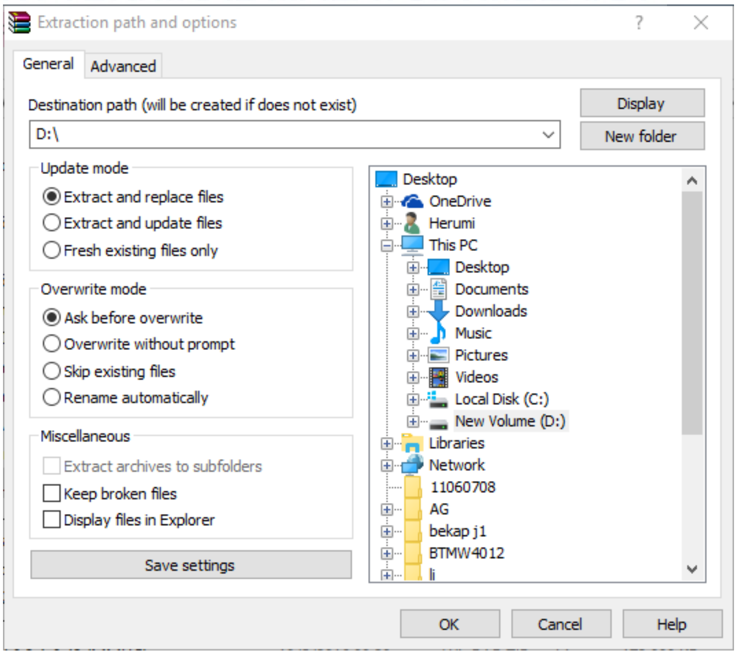


Figure 5.5: Extract the zip content to SD card

iii. Boot the raspberry pi using the SD card created. The installation process will begin. Follow the on screen instruction until the installation process complete.

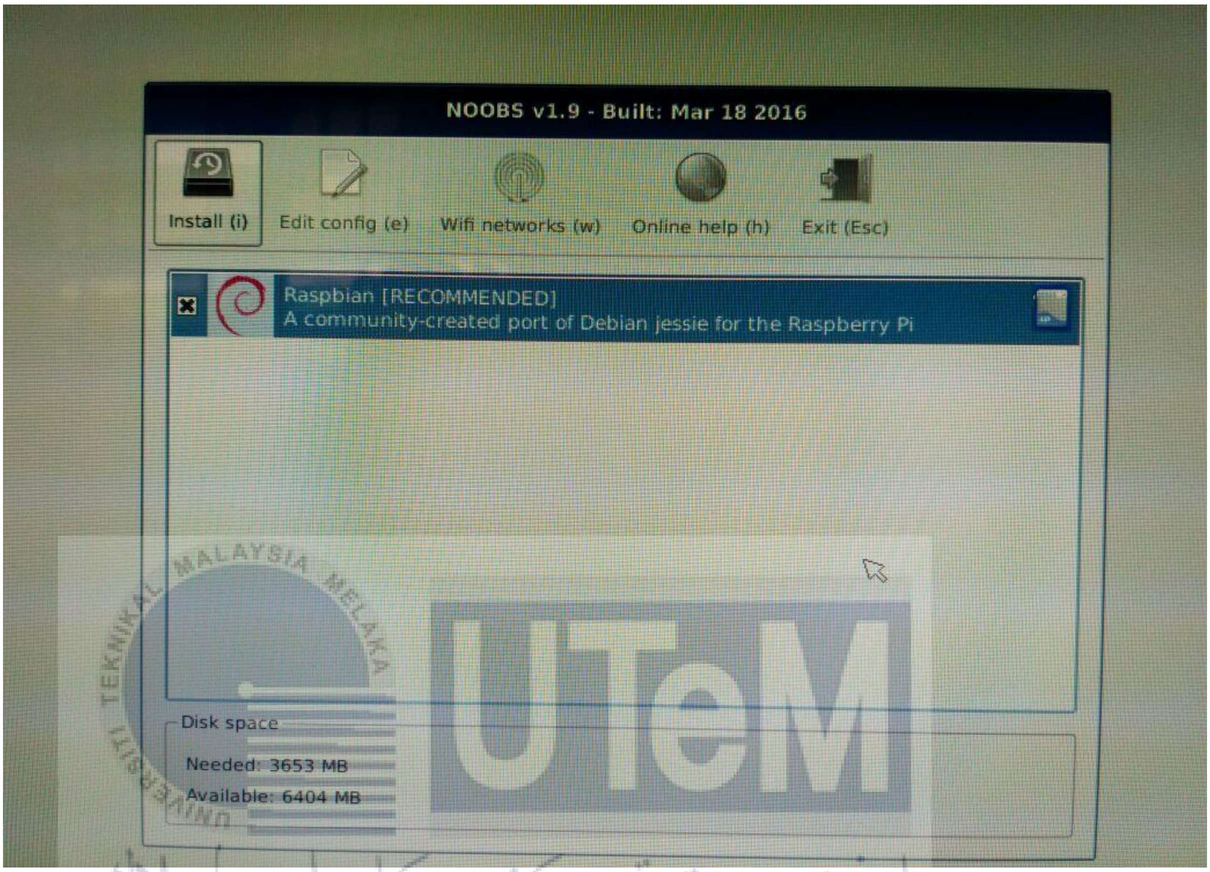


Figure 5.6: operating system installation

5.2.2.2 Freqshow installation

Freqshow is a python application created to view and manipulate frequency signal into graphical form.

i. Open terminal program in the Raspbian operating system.

ii. Run the command "*sudo apt-get install cmake build-essential python-pip libusb-1.0-0-dev python-numpy git*" to install the required package to run the application.

iii. Clone the rtl-sdr source code from the developer github using code "*git clone git://git.osmocom.org/rtl-sdr.git*".

iv. Make a new directories in the rtl-sdr folder named build using command “*mkdir build”.*

v. Go to directory rtl-sdr/build.

vi. Compile the application rtl-sdr

-cmake ../ -DINSTALL_UDEV_RULES=ON -DDETACH_KERNEL_DRIVER=ON

-make

-sudo make install

-sudo ldconfig

-*sudo pip install pyrtlsdr*

vii. Clone the freqshow application source code from the developer github using code *“git clone https://github.com/adafruit/FreqShow.git”.*

5.2.2.3 4DPi-24-HAT driver installation

This driver contains all the necessary files needed to output the display to the 4DPi-24-HAT Touchscreen display.

i. Download the driver package from the manufacturer’s website using code “*wget http://www.4dsystems.com.au/downloads/4DPi/All/4d-hats_4-1-19_v1.0.tar.gz*”

ii. Extract the package using “*sudo tar -xzvf 4DPi-24-HAT/4DPi-24-HAT_kernel_R_1_0.tar.gz -C /”*

iii. Open the configuration file in /boot/config.txt. uncomment the line “*kernel=kernel_hat.img”* because we are running it on raspberry pi 1

```
#uncomment to overclock the arm. 700 MHz is the default.
#arm_freq=800

# Uncomment some or all of these to enable the optional hardware interfaces
#dtparam=i2c_arm=on
#dtparam=i2s=on
#dtparam=spi=on

# Uncomment this to enable the lirc-rpi module
#dtoverlay=lirc-rpi

# Additional overlays and parameters are documented /boot/overlays/README

# Enable audio (loads snd_bcm2835)
dtparam=audio=on

# uncomment one of the ovrlays if loading from EEPROM is not supported
#dtoverlay=4dpi-32
#dtoverlay=4dpi-35
#dtoverlay=24-hat
#dtoverlay=32-hat
#dtoverlay=35-hat

#uncomment to enable HAT EEPROM programming
#dtparam=i2c0

#uncomment if running on RPi2
#kernel=kernel7_hat.img

#uncomment if running on RPi1
kernel=kernel_hat.img
```

Figure 5.7: Uncomment the line necessary to run the display on Raspberry Pi 1

5.2.3. Freqshow Configuration Setup

i. In this project a Python file were created. The python file will be name ***freqshow.py***. the python file contains all the necessary code needed for the application to run.

ii. We need to alter some of the code to suite it to the hardware used in this project, in this case, the 4DPi-24-HAT touchscreen display.

freqshow.py

```
# FreqShow main application and configuration.
# Author: Tony DiCola (tony@tonydicola.com)
#
# The MIT License (MIT)
#
# Copyright (c) 2014 Adafruit Industries
#
# Permission is hereby granted, free of charge, to any person obtaining a copy
# of this software and associated documentation files (the "Software"), to deal
# in the Software without restriction, including without limitation the rights
# to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the Software is
# furnished to do so, subject to the following conditions:
#
# The above copyright notice and this permission notice shall be included in all
# copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
# IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
# FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
# AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
# LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
# OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
# SOFTWARE.
import os
import time

import pygame

import controller
import model
import ui


# Application configuration.
SDR_SAMPLE_SIZE = 1024  # Number of samples to grab from the radio.  Should be
                        # larger than the maximum display width.

CLICK_DEBOUNCE  = 0.4   # Number of seconds to wait between clicks events. Set
                        # to a few hunded milliseconds to prevent accidental
                        # double clicks from hard screen presses.

# Font size configuration.
MAIN_FONT = 23
NUM_FONT  = 23

# Color configuration (RGB tuples, 0 to 255).
MAIN_BG        = (  0,   0,   0) # Black
INPUT_BG       = ( 60, 255, 255) # Cyan-ish
INPUT_FG       = (  0,   0,   0) # Black
CANCEL_BG      = (128,  45,  45) # Dark red
ACCEPT_BG      = ( 45, 128,  45) # Dark green
BUTTON_BG      = ( 60,  60,  60) # Dark gray
BUTTON_FG      = (255, 255, 255) # White
BUTTON_BORDER  = (200, 200, 200) # White/light gray
INSTANT_LINE   = (  0, 255, 128) # Bright yellow green.
```

Figure 5.8: Freqshow.py source code part 1

iii. The code inside box 1 show the number of sample size. We need to set it to 1024 because the value is larger than the maximum display width resolution.

iv. The code in box 2 is the font size configuration. We need to change it according to size 23 to make it fit with the resolution of the display which is 240x320.

```
# Define gradient of colors for the waterfall graph.  Gradient goes from blue to
# yellow to cyan to red.
WATERFALL_GRAD = [(0, 0, 255), (0, 255, 255), (255, 255, 0), (255, 0, 0)]

# Configure default UI and button values.
ui.MAIN_FONT = MAIN_FONT
ui.Button.fg_color     = BUTTON_FG
ui.Button.bg_color     = BUTTON_BG
ui.Button.border_color = BUTTON_BORDER
ui.Button.padding_px   = 2
ui.Button.border_px    = 2


if __name__ == '__main__':
    # Initialize pygame and SDL to use the PiTFT display and touchscreen.
    os.putenv('SDL_VIDEODRIVER', 'fbcon')
    os.putenv('SDL_FBDEV'      , '/dev/fb1')
    os.putenv('SDL_MOUSEDRV'   , 'TSLIB')
    os.putenv('SDL_MOUSEDEV'   , '/dev/input/touchscreen')
    pygame.display.init()
    pygame.font.init()
    pygame.mouse.set_visible(True)
    # Get size of screen and create main rendering surface.
    size = (pygame.display.Info().current_w, pygame.display.Info().current_h)
    screen = pygame.display.set_mode(size, pygame.FULLSCREEN)
    # Display splash screen.
    splash = pygame.image.load('freqshow_splash.png')
    screen.fill(MAIN_BG)
    screen.blit(splash, ui.align(splash.get_rect(), (0, 0, size[0], size[1])))
    pygame.display.update()
    splash_start = time.time()
    # Create model and controller.
    fsmodel = model.FreqShowModel(size[0], size[1])
    fscontroller = controller.FreqShowController(fsmodel)
    time.sleep(2.0)
    # Main loop to process events and render current view.
    lastclick = 0
```

3

Figure 5.9: Freqshow.py source code part 2

v. The code in box 3 control the driver initialise with the application.

-os.putenv('SDL_VIDEODRIVER', 'fbcon')

to channel the output to the screen

-os.putenv('SDL_FBDEV'   , '/dev/fb1')

to select the screen output, which is the 4DPi-24-HAT

-os.putenv('SDL_MOUSEDRV'   , 'TSLIB')

to switch on mouse

-os.putenv('SDL_MOUSEDEV'   , '/dev/input/touchscreen')

to select the touchscreen as the input device

vi. Next, we need to set the application to automatically run upon boot.

vii. Create a new shell script using code “*nano launcher.sh*”. add the following code into the script.

cd /home/pi/FreqShow

sudo python freqshow.py

viii. Change the permission of the shell script created using code "*chmod 755 launcher.sh"*

ix. Cron is a Unix utility that allows tasks to be automatically run in the background at regular intervals by the Cron daemon. Using this in built feature, we will set the application to automatically run upon the device start. Open the Cron utility configuration file using code "*sudo crontab -e". *add the following code

@reboot sh /home/pi/launcher.sh >/home/pi/logs/cronlog 2>&1

x. The application should be automatically launched when the device starts.

5.3. Conclusion

In conclusion, the implementation phase show how to make this project work. By doing this chapter, I had learned how to setup and install the necessary component needed to make the project work. I also learn how to code in python programming language. I also familiarise myself with basic Linux command to navigate the Raspbian operating system.

CHAPTER VI

TESTING

6.1. Introduction

This chapter will explain about testing of the product produced in this project. Based on the phase that has been discussed before, now it is time to do the testing of the project. This process will discuss about how we test and analyse the component used in the project. A brief testing of the completed product is also done in this chapter. Figure 6.1 below show the summary of this chapter.

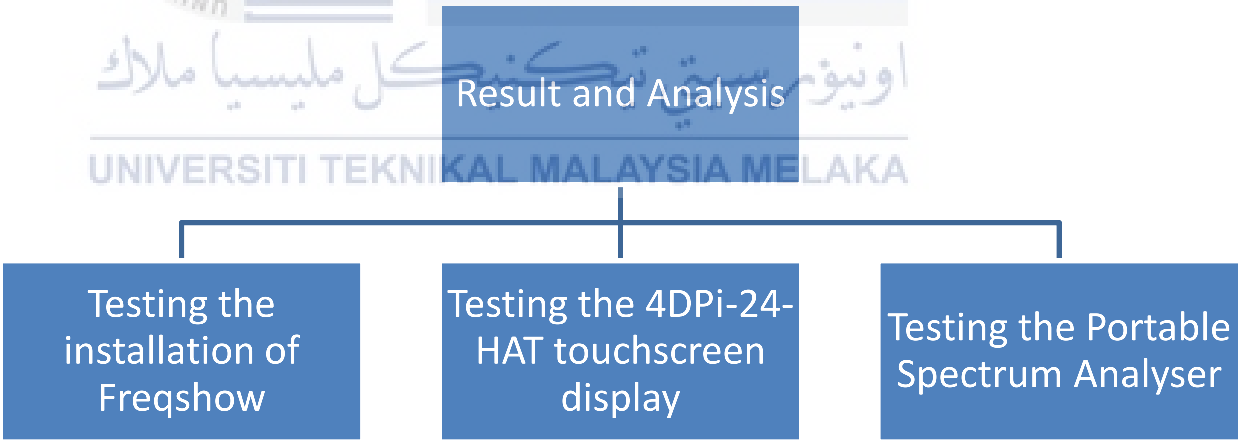


Figure 6.1: Result and analysis chapter summary

6.2. Result and Analysis

6.2.1 Test the installation of Freqshow

i. After the installation process of the software Freqshow is completed, we need to test whether the installation OS working or not. To test this, we run the code using command *"sudo python freqshow.py"*

Figure 6.2: The splash screen will load if the program installed correctly

6.2.2 Test the 4DPi-24-HAT touchscreen display

i. After the installation of the driver package for the touchscreen display, we need to restart the device. After restart, the display should be working as it should be.

Figure 6.3: The display after restart

6.2.3 Test the Portable Spectrum Analyser

i. To test the Portable Spectrum Analyser, we need to complete the implementation phase and the testing phase first. After the device has been completely connected and the software has been setup perfectly, then we can test all the functionality of the device.

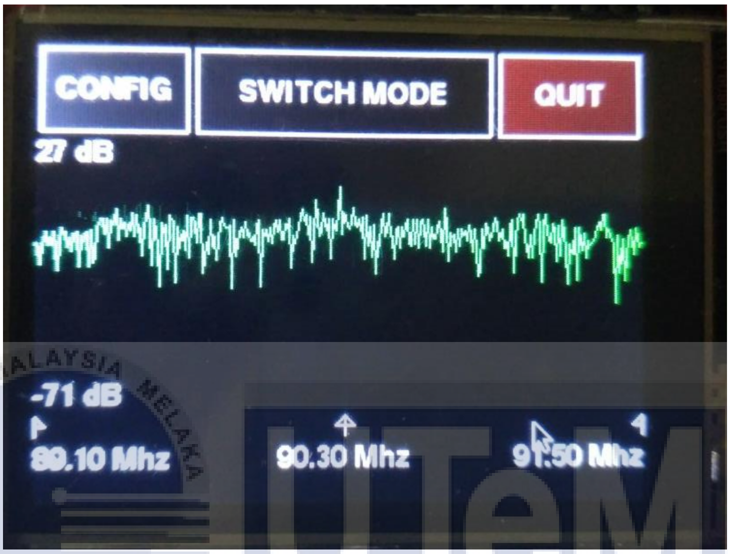


Figure 6.4: Main window of the Portable Spectrum Analyser

ii. Touch on the config button to open the configuration menu. Here, there are four item that can be configured which is the centre frequency, sample rate, gain, and the minimum and maximum value of the graph.

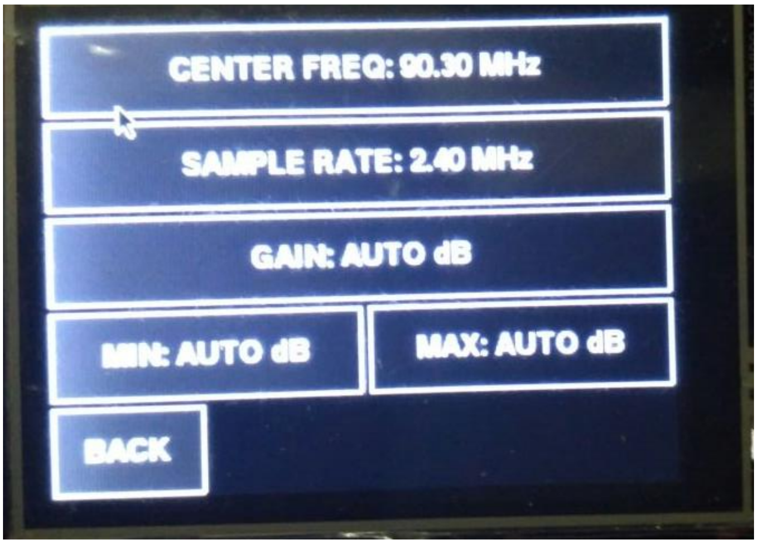


Figure 6.5: Configuration windows

iii. The centre frequency menu control the centre frequency shown on the graph produced. The range of frequency that can be tuned using the RTL-SDR tuner used in this project is from 24MHz – 1700MHz. Any value larger than 1700MHz will cause the program to crash.

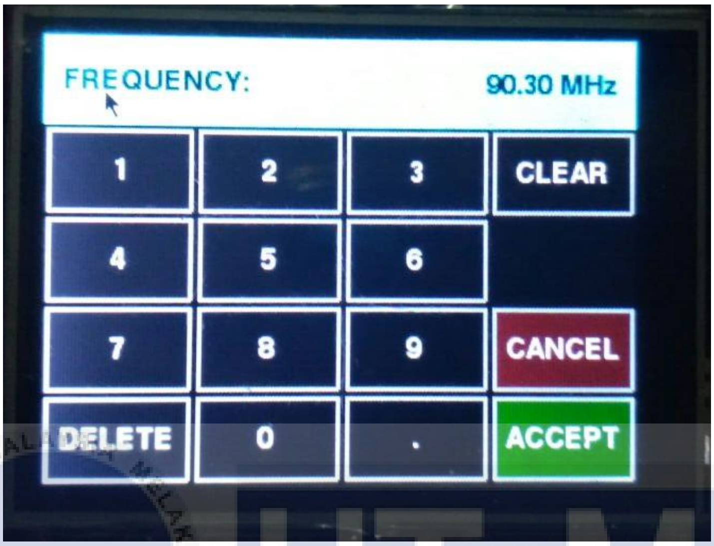


Figure 6.6: Change the centre frequency value by entering the desired value

iv. The next option is sample rate, which control how wide the range of frequencies to be displayed on the graph produced. Do keep in mind that the tuner only support value from 1MHz to 3.2MHz.

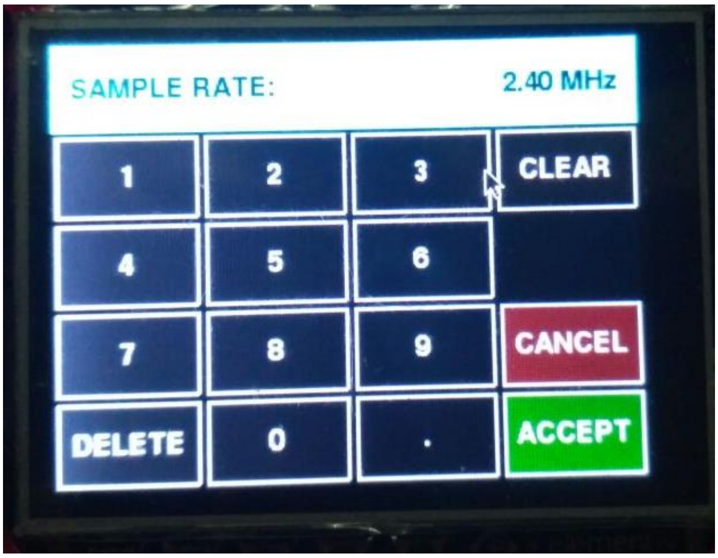


Figure 6.7: Change the sample rate value by entering the desired value

v. The gain setting adjusts the internal gain of the tuner. Increasing the gain will make a weak signal more readable. The range of gain that the tuner can accept is between 0 dB to 3.7 dB. The auto button will select the best gain for the signal received.

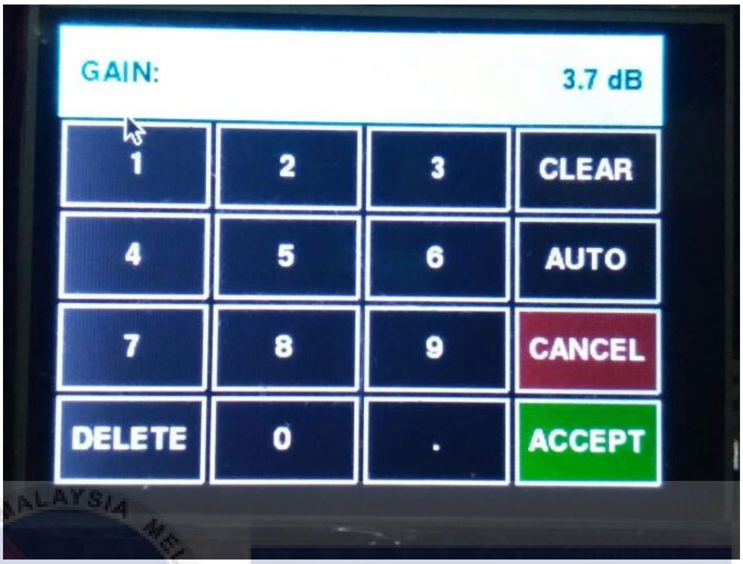


Figure 6.8: Change the gain value by entering the desired value

vi. The min and max button serve x-axis of the graph. The min represents the minimum value while the max represents the maximum value to be shown on the graph plotted. An auto mode is also can be used to make the graph adjust the minimum and maximum value based on the lowest and highest intensity captured by the tuner.

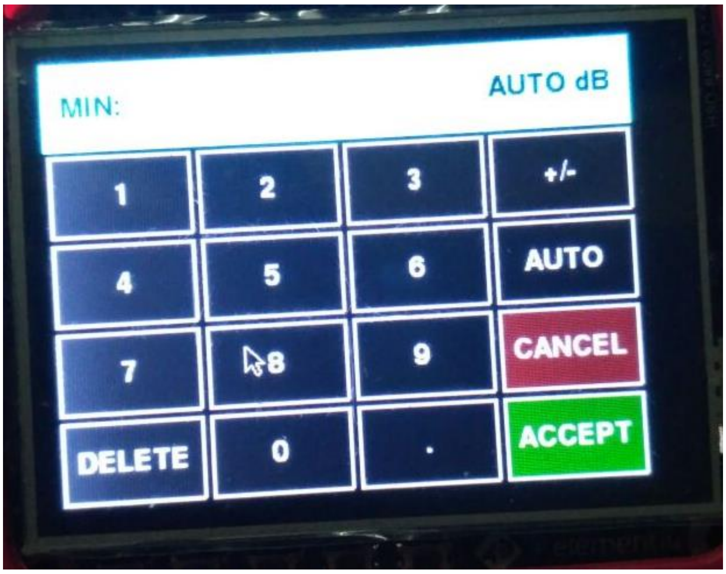


Figure 6.9: Change the minimum and maximum value by entering the desired value

vii. The last function to be explored is the switch mode function. By pressing the switch mode button, the device will display an intensity of the signal over time. This will come handy when we need to see the stability of the signal. A red colour means the maximum intensity while blue colour means minimum intensity.

Figure 6.10: The intensity of signal over time graph

6.2.4 Example spectrum testing

To test the functionality of the device, we need to test the device using a fix frequency. Firstly, we set the centre frequency to 90.3 Mhz. As of the testing is done in Malacca, the frequency set is for Era fm. Here, we can see the spike of graph showing the radio signal is transmitted on the selected frequency.

Figure 6.11: The graph produced at 90.3Mhz

Next, we increase the sample rate to 3.2Mhz which is the highest the tuner can accept. The frequency range should change to a bigger value now.

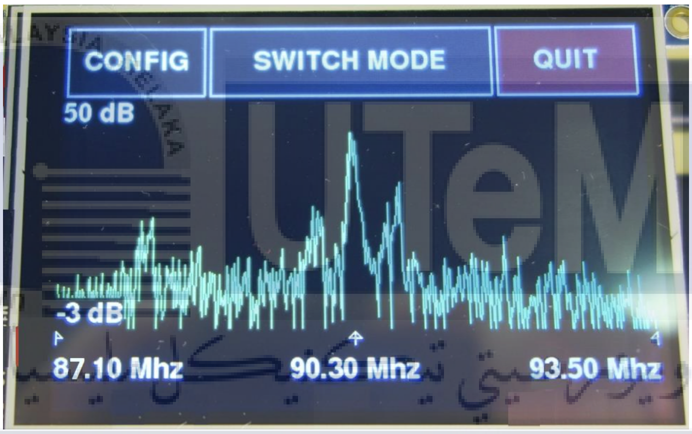


Figure 6.12: The graph produced when the sample rate is increased

Next, we increase the gain to 3.7dB which is the highest gain the tuner can support. The graph should be enlarged and zoomed.

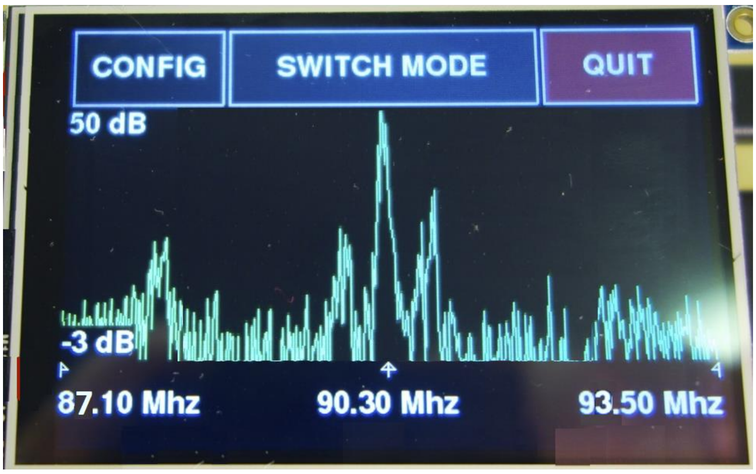


Figure 6.13: The graph produced when the gain is increased

6.3. Conclusion

As the conclusion, the testing phase is to show the result of this project and how to test the project to get the result of the project. By doing the testing phase, we can define the range and limit of the tuner. All the information that has been gathered in this phase will be bring in the next process which is project conclusion.

CHAPTER VII

PROJECT CONCLUSION

7.1. Introduction

This chapter will explain about the project conclusion. This phase will discuss overall aspect of this project, the strength and weakness of the project, the contribution of the project to the society, the limitation of this project and the future works of this project that needed to be improved to make the project better.

7.2. Project Summarization

7.2.1 Project Objective

The objective of this project that has been identified was listed as below:

i. To study about performance parameters used in analysing spectrum.

   To achieve this objective, we have done a research on how a spectrum analyser works, what are the basic component of spectrum analyser, and the type of graph produced by a spectrum analyser.

ii. To develop a portable radio spectrum analyser using a Raspberry Pi and RTL-SDR tuner.

The objective has been done and a fully functional low cost Portable Spectrum Analyser has been developed using a Raspberry Pi and RTL-SDR Tuner. The device developed is small and cheap enough and very suitable to be used as a teaching tool for secondary school student.

iii. To analyse radio frequency spectrum using the developed device, based on studied parameter

After the completion of the development of the device, we had done an extensive test to see the ability of the developed device. Some of the ability of the device is to create a frequency versus intensity graph, frequency versus time graph, and changing the parameters of the graph created.

7.2.2 Project Weakness and Strength.

7.2.2.1 Project Weakness

i. The prototype is not very strong. Because the device is assembled using three different components which is the Raspberry Pi itself, the RTL-SDR tuner, and the battery, we need to handle the device with care to maintain the connection. Any harsh movement may cause the connection to break and disrupt the device operation.

ii. The RTL-SDR tuner used in this project has a range of frequency that is limited from 24MHz to 1700MHz. The tuner also can only view sample rate from 1MHz to 3.5MHz. Any value higher than the range stated will cause the program to crash because the needed data to create the graph is absent.

iii. The display unit used in this project is only 2.4 inch wide. This limit the size of the graph to be plotted and make the graph less readable. The graph is also not very precise because the range to be drawn is not so big due to the small screen size.

7.2.2.2 Project Strength

i. This project only need 5v power supply to power the device. Because raspberry pi is used as the main processing unit, only 5v is needed to operate the device making it suitable for school use or any low powered user.

ii. The device created is small enough to be taken anywhere, making it a very portable and mobile device. The only component of the device is the raspberry pi, the RTL-SDR tuner, and a battery bank making the device easy to be taken and stored.

iii. The cost for the making of this Portable Spectrum Analyser is only RM400. This make the device a very compelling option to be used as a teaching tool as a normal portable spectrum analyser may price up to RM800. The component price can be break down as RM130 for the raspberry pi board, RM130 for the touchscreen display, RM20 for SD card, RM70 for the RTL-SDR tuner, and RM50 for the battery bank.

7.3. Project Contribution

This project was developed to help school student to learn the basic of frequency. Using the device, student can learn how the graph of frequency look like, and see the wireless signal around us in real life compared to learning them just in theory. Due to the portability and low cost of the device, this make a very good device to be placed at school.

By completing this project, I hope that all school student can learn and interact using the device to enhance their knowledge. Also, school will no longer need to provide big budget to buy a spectrum analyser thus save some money to be used for more useful program.

7.4 Project Limitation

The limitation of this project is still based on the prototype produced. Because its only a prototype, it still need some fine adjustment to make it look better and stronger. Also, the whole project depends on the RTL-SDR tuner specification. The limitation of the tuner is also the limitation of the project. We can only see and plot the graph based on the tuner range of frequency only. Another limitation is the display size which is quite small making it a little bit less easy to be readable and causing the graph produced to be not as precise as it should be

7.5 Future Works

This project can be upgrade for a better performance and function. The future works that can be considered are:

i. Use a tuner that can capture a bigger range of frequency and bigger range of sample rate to increase the sensitivity of the device.

ii. Upgrade the display to a bigger and higher pixel density screen to produce a bigger and better graph to be plotted on screen.

iii. Combine all the component into a single device to make it look better and easier to handle. It will also increase the strength of the device.

7.6 Conclusion

As the conclusion, this project successfully meets my objectives that I had identified earlier before the project is started. Hopefully, the Portable Spectrum Analyser can help school student to learn more effectively. School will also benefit from the low price of the device by cutting the cost to buy the device and make it very suitable to be handled by student. I am confident that this project can be very useful for the development of the country.

REFERENCE

[1] Agilent Technology. (2012). "Spectrum Analyser Basics".

[2] About RTL-SDR, accessed on 28 March 2016, <http://www.rtl-sdr.com/about-rtl-sdr/>.

[3] What is Raspberry Pi, accessed on 29 March 2016, <https://opensource.com/resources/what-raspberry-pi>.

[4] Raspberry Pi, accessed on 29 March 2016, <https://en.wikipedia.org/wiki/Raspberry_Pi>.

[5] Dicola, T. (2014). "Freq Show: Raspberry Pi RTL-SDR Scanner" .

[6] Krilic, S. (2015). "Monitoring Network Manager for RTL-SDR" .

[7] Ong, S. (2013). "ViewRF software – RTL-SDR Spectrum Analyser for BeagleBone Black" .

[8] Radio Frequency, accessed on 30 March 2016, <https://en.wikipedia.org/wiki/Radio_frequency>.

[9] Bandwidth, accessed on 30 March 2016, <https://en.wikipedia.org/wiki/Bandwidth_(signal_processing)>.

[10] Antenna gain, accessed on 30 March 2016, <https://en.wikipedia.org/wiki/Antenna_gain>.

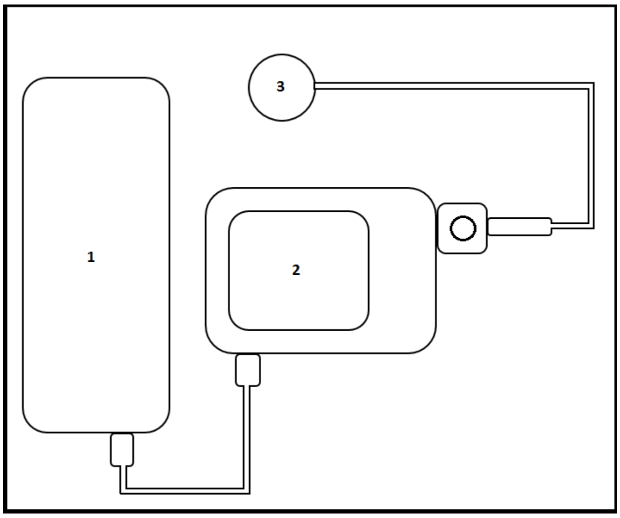
APPENDIX

Product Manual

1.Specification

- Raspberry Pi 1 model B+
  - 700 MHz single-core ARM11 processor
  - 512MB SDRAM
  - 600mA power rating
- DVB-T RTL-SDR USB TV tuner
  - SDR frequency 24MHz-1700MHz
  - SDR bandwidth 1MHz-3.2MHz
  - USB 2.0 connection
- 4DPi-24-HAT Touchscreen display
  - 2.4-inch diagonal size
  - 320x240 QVGA TFT display
  - Resistive touch input
- 10000 mAh battery bank
  - Lithium-ion rechargeable cell
  - 10000mAh rated capacity

2. Overview



[1] Battery bank

[2] Touchscreen display

[3] Antenna

3. How to Use

a. Plug in the battery bank to the device.

b. The device will turn on automatically.

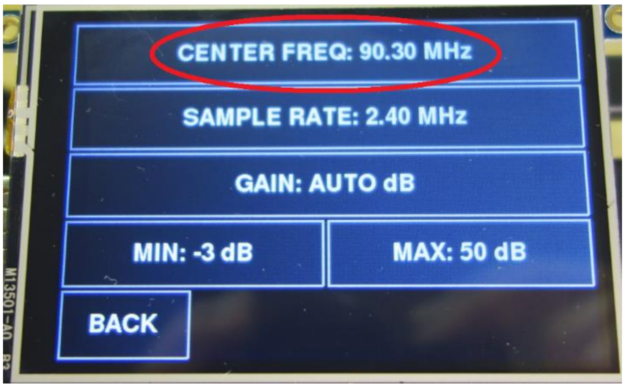
c. When the device is booted, click on the launcher icon.

d. The program will load and the device is ready to use.

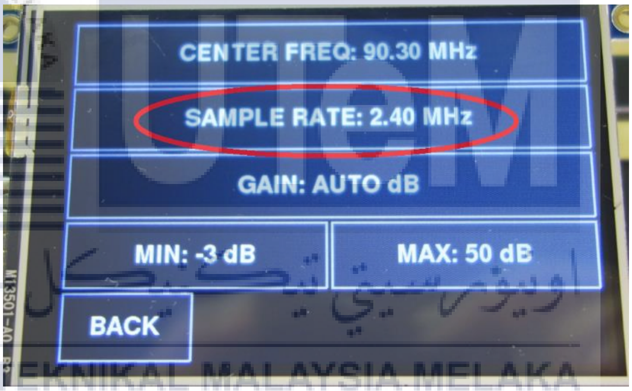
4. Features

a. Touch on the config button to open the configuration menu. Here, there are four item that can be configured which is the centre frequency, sample rate, gain, and the minimum and maximum value of the graph.

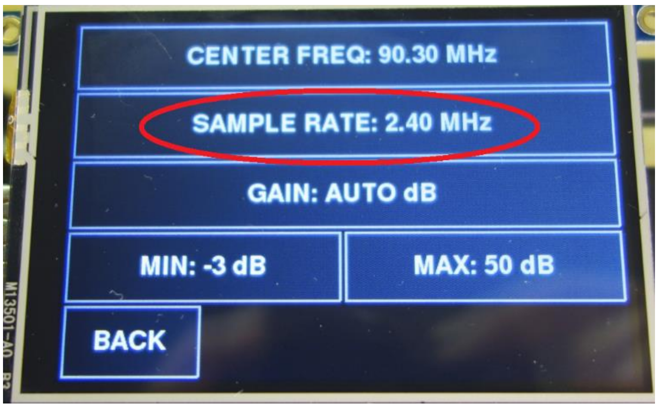
b. The centre frequency menu control the centre frequency shown on the graph produced. The range of frequency that can be tuned using the RTL-SDR tuner used in this project is from 24MHz – 1700MHz.



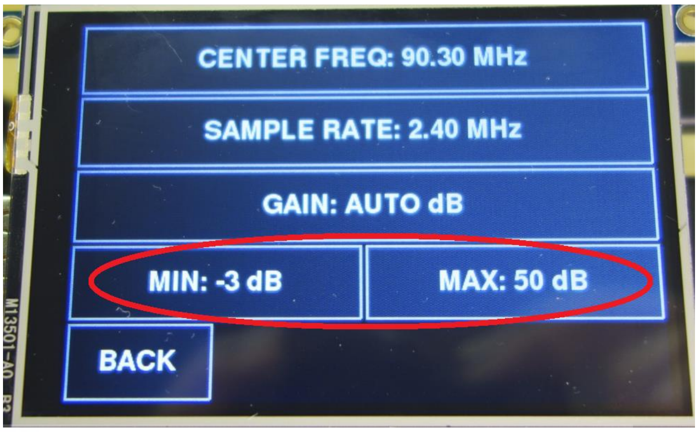
c. The sample rate menu control how wide the range of frequencies to be displayed on the graph produced. The tuner only support range of value from 1MHz to 3.2MHz.



d. The gain setting adjusts the internal gain of the tuner. Increasing the gain will make a weak signal more readable. The range of gain that the tuner can accept is between 0 dB to 3.7 dB. The auto button will select the best gain for the signal received.



e. The min and max button serve x-axis of the graph. The min represents the minimum value while the max represents the maximum value to be shown on the graph plotted. An auto mode is also can be used to make the graph adjust the minimum and maximum value based on the lowest and highest intensity captured by the tuner.



f. The switch mode button stays at the main windows. This option will change the graph to display the intensity of the signal over time. This will come handy when we need to see the stability of the signal. A red colour means the maximum intensity while blue colour means minimum intensity.