PORTABLE DVOR RECEIVER DESIGN USING RTL-SDR R820T2 WITH PYTHON PROGRAMMING LANGUAGE

Tri Rizki Sajadah*, Lady Silk Moonlight, Teguh Imam Suharto

Politeknik Penerbangan Surabaya, Jalan Jemur Andayani I No 73, Kota Surabaya, 60236

**Corresponding Author. Email: sendtotririzki@gmail.com*

Abstract

The navigation equipment currently in use in Indonesia for flight is DVOR. The PIR (Portable ILS/DVOR Receiver) equipment must be subject to periodic ground check procedures using to ensure that the equipment is in good working condition. The PIR currently in use is manufactured by the manufacturer of DVOR equipment. As a result, the design of a portable DVOR receiver is underway, along with the measurement and evaluation of its performance, using the RTL-SDR R820T2 device in combination with the Python programming language. This design implements the waterfall method as its technique. The output of the portable DVOR receiver that employs RTL-SDR takes the form of 30 Hz amplitude modulation percentage. The portable DVOR receiver using RTL-SDR, which features a positive-intrinsic-negative (PIR) manufacturer, exhibits an average deviation of 0.25% for the 30 Hz amplitude modulation percentage. According to these findings, the design tool functions sufficiently well.

Keywords: DVOR, PIR, RTL-SDR, Python

INTRODUCTION

Air navigation is the process of directing the movement of aircraft from one point to another in a safe and smooth manner to avoid hazards and/or obstacles to flight. The objective of air navigation is to provide air navigation services in accordance with applicable standards, to achieve flight safety, and to achieve an integrated, harmonious, and harmonious air navigation service at the national, regional, and international levels. The navigation equipment utilized for flights in Indonesia is presently DVOR. It requires periodic ground checks.

Ground check is a procedure to ensure the quality and accuracy of DVOR at the transmitter station. The activity is performed periodically using PIR (Portable ILS/VOR Receiver) equipment [1]. The ground check process is very important to ensure that the DVOR is working properly and providing accurate information to pilots. The quality and accuracy of the air navigation system is very important for flight safety, as navigation errors can cause fatal aircraft accidents [2]. Currently, the PIR used is produced by the manufacturer of the DVOR. Therefore, the designs, measures and evaluates the performance of a portable DVOR receiver using RTL-SDR R820T2 with the Python programming language. The results are focused as 30 Hz AM modulation percentages displayed on the design tool.

Based on the background information provided, the design will be conducted using the waterfall method. The waterfall method is one of the techniques in the System Development Life Cycle (SDLC). Its characteristic involves finishing each phase before moving onto the next, ensuring maximum efficiency as no parallel work is implemented [3]. The waterfall research method is employed as it is an appropriate technique for developing software tailored to the requirements of designing a Portable DVOR Receiver Using RTL-SDR R820T2 with the Python programming language.

THEORETICAL FOUNDATION

There are several theories for the development of this device.

DVOR (Doppler Very High Frequency Omnidirectional Range)

DVOR (Doppler Very High Frequency Omnidirectional Range) is a radio navigation system employed in aircraft to ascertain azimuth towards ground stations. DVOR operates within the frequency range of 108 MHz - 118 MHz [4]. DVOR operates within the VHF frequency range, thereby its range is determined by the Line of Sight boundaries. Thus, DVOR is regarded as a short-range navigation tool [5]. The DVOR operates by using the phase difference between two 30 Hz signals which are modulated on the carrier signal. The two signals are known as the reference phase and variable phase [6]. The carrier amplitude is modulated with a 30 Hz sine wave signal to obtain the reference phase signal. The carrier antenna transmits the Amplitude Modulation (AM) signal in the horizontal plane in all directions. This produces a circular radiation pattern that results in a 30 Hz signal with a phase that is independent of azimuth [7].

RTL-SDR R820T2

Realtek Software Defined Radio (RTL-SDR) is a technology in constant development that is currently making significant strides in the telecommunications industry. The purpose of Software Defined Radio (SDR) technology is to make maximum use of programmable hardware that allows the construction of software-based radios [8]. To receive and decode radio signals, RTL-SDR utilises a digital TV tuner chip [9]. RTL-SDR is a radio device that utilises open-source hardware and software to convert radio signals into digital data that is computer-processable [10]. RTL-SDR comprises of multiple variants, with RTL-SDR R820T2 being one of them. The RTL-SDR R820T2 constitutes a highly popular and cost-effective Software-Defined Radio (SDR) device. To receive and convert radio signals into digital signals capable of processing by a computer, this device operates using the Realtek RTL2832U chipset and R820T2 tuner [11].

Python

In the age of information technology, programming has become an essential activity. Applications that operate on desktop computers, laptops, mobile phones, PDAs, and so on are not spontaneous but rather created using a systematic process of analysis and design [12]. Python is a programming language that is both simple and reliable. It implements a simple but powerful approach to object-oriented programming, and includes an efficient high-level data structure[13]. Python is equipped with object-oriented programming algorithms, functional programming, as well as procedural programming. Furthermore, Python comprises numerous modules and features that expedite software development, such as modules for data science, image and video processing, web programming, and game development [14].

RESEARCH METHOD

The research methodology used in this study is the waterfall method. The Waterfall method is a conventional software development approach. This technique follows a sequential process commencing with planning, analysis, design, and concluding with system implementation. Each phase in this method requires completion before proceeding to the next phase [3]. The waterfall method outlines a number of sequential phases that must be completed one at a time, with the next phase only occurring after the previous one has been completed [15]. The stages in the waterfall model are outlined in Figure 1, which includes five phases: These phases include Analysis, Design, Implementation, Testing, and Maintenance.

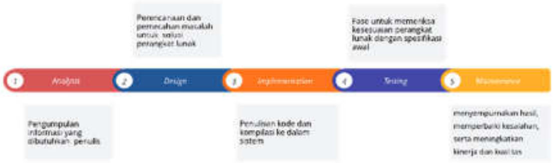
Perencanaan dan pemecahan masalah untuk solusi perangkat lunak
Fase untuk memeriksa kesesuaian perangkat lunak dengan spesifikasi awal
1
Analysis
2
Design
3
Implementation
4
Testing
5
Maintenance
Pengumpulan informasi yang dibutuhkan penulis
Penulisan kode dan kompilasi ke dalam sistem
menyempurnakan hasil, memperbaiki kesalahan, serta meningkatkan kinerja dan kualitas

Figure 1 Waterfall Development Flow

RESULT AND DISCUSSION

Analysis

The devices required for designing the tool are discovered through analysis. According to the analysis, the required devices are:

a) Dipole Antenna

It is used to receive electromagnetic signals emitted by DVOR.

b) RTL-SDR R820T2

They are used to convert radio signals into digital data, which can be processed by a computer.

c) Computer or laptop

This device is used to run the program that has been created using the Python programming language.

d) Visual Studio Code

Visual Studio Code is used as a code editor for writing source code using the Python programming language.

e) Python

The Python programming language is used to write source code that processes data received from RTL-SDR.

Design

The design stage represents the planning and problem-solving phases for developing device solutions. The devises the overall functionality of the tool, as illustrated below:

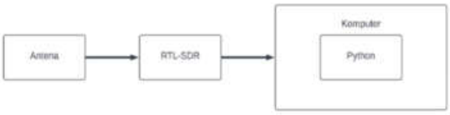

Figure 2 Design Schematic

Implementation of the above block diagram as shown below:

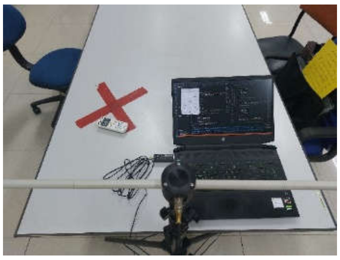

Figure 3 Implementation of The Schematic

Figure 3 shows the implementation of the tool design. To receive the signal beam from the DVOR transmitter station, the tool operates from the antenna linked to the RTL-SDR. The received signal is transmitted to a computer or laptop by an RTL-SDR device, which changes radio signals to digital signals. The signal changed into the computer is processed using Python programming language in order to show the percentage modulation of 30 HZ AM.

Implementation

This is the implementation stage and is concerned with writing code and compiling it into the system. Code writing involves transforming design requirements and specifications into programs that can be executed through programming. This stage comprises several processes:

Interface Development

The interface was developed using the Python programming language and the tkinter library. Numeric buttons are used to enter a numeric input for adjusting the frequency received by RTL-SDR. The interface that resulted is displayed below:

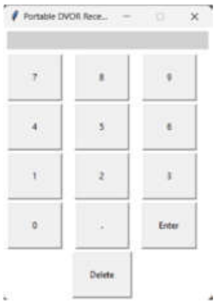

Figure 4 Portable DVOR Receiver Interface

The interface for writing the source code is represented in Figure 4. This interface is utilised to input the frequency of DVOR in MHz units and exhibit the percentage modulation outcome of 30 Hz AM.

Signal Processing Source Code Development

The source code is used to convert the received signal into an 8-bit form to facilitate signal processing. The result of the processing is a signal that is ready for transmission to the next stage.

Decimate Signal Source Code Development

The Decimate Signal Source Code is used to remove undesired signals. This source code aims to decrease the number of signals that other source codes must process.

Lowpass Filter Source Code Development

The source code for the low pass filter is used to allow low frequencies to pass through the filter. The use of this source code will speed up the processing of the signal in the following process. The main purpose of using this filter is to preserve the 30 Hz frequency in the DVOR carrier signal.

Development of RTL-SDR Source Code for Frequency Setting

The RTL-SDR frequency is set to receive the correct frequency from the DVOR transmitter. To set the frequency on RTL-SDR, the value for the centre frequency can be entered through the sdr.center_freq program. Additionally, the gain can also be adjusted.

4.4 Testing

The design of the device has been tested using the Avionic Tester, which is acting as a DVOR transmitter. The device, which displays the results of the modulation of the 30 Hz AM signal with the carrier signal, receives the DVOR signal transmitted by the Avionic Tester. A PIR (Portable ILS/VOR Receiver) NORMARC 7710 NAV Analyzer is used for comparison.

Table 1. Navigation Laboratory Test Result Data

No	Design Device	PIR	Deviation	Frequency
1.	30.2 %	30.8 %	0.6 %	108.0 MHz
2.	33.2 %	32.6 %	0.6 %	108.0 MHz
3.	31.8 %	31.7 %	0.1 %	108.0 MHz
4.	31.7 %	31.8 %	0.1 %	108.0 MHz
5.	32.3 %	31.9 %	0.4 %	112.4 MHz
6.	30.1 %	30.8 %	0.7 %	112.4 MHz
7.	32.0 %	31.9 %	0.1 %	112.4 MHz
8.	33.1 %	32.4 %	0.7 %	113.6 MHz

9.	30.1 %	30.7 %	0.6 %	113.6 MHz
10.	31.9 %	31.5 %	0.4 %	113.6 MHz
Deviation Average			**0.43 %**	

Tests were carried out in the navigation laboratory. The 30 Hz AM modulation percentage reading on the Portable DVOR receiver design was compared with the PIR NORMARC 7710 NAV Analyzer. The test result showed that the average 30Hz AM modulation percentage deviation was 0.43%.

Table 2. Classroom Test Result Data

No	Design Device	PIR	Deviation	Frequency
1.	31.2 %	30.9 %	0.3 %	108.0 MHz
2.	32.0 %	32.1 %	0.1 %	108.0 MHz
3.	32.1 %	31.8 %	0.3 %	108.0 MHz
4.	32.0 %	31.8 %	0.2 %	112.4 MHz
5.	31.5 %	31.2 %	0.3 %	112.4 MHz
6.	31.9 %	32.1 %	0.2 %	112.4 MHz
7.	31.4 %	31.8 %	0.4 %	113.6 MHz
8.	32.3 %	32.1 %	0.2 %	113.6 MHz
9.	31.2 %	31.2 %	0 %	113.6 MHz
10.	31.7 %	31.5 %	0.2 %	113.6 MHz
Deviation Average			**0.22 %**	

The test was carried out in the classroom. The results of the 30 Hz AM modulation percentage reading on the portable DVOR receiver design were compared with the PIR NORMARC 7710 NAV Analyser. The obtained average deviation of 30 Hz AM modulation percentage, from the test results, was 0.22%.

Table 3. Outdoor Test Result Data

No	Design Device	PIR	Deviation	Frequency
1.	31.0 %	31.1 %	0.1 %	108.0 MHz
2.	31.7 %	31.5 %	0.2 %	108.0 MHz
3.	32.4 %	32.5 %	0.1 %	108.0 MHz
4.	33.0 %	32.8 %	0.2 %	108.0 MHz
5.	32.0 %	32.0 %	0 %	112.4 MHz
6.	32.6 %	32.7 %	0.1 %	112.4 MHz
7.	31.7 %	31.8 %	0.1 %	112.4 MHz
8.	31.1 %	31.1 %	0 %	113.6 MHz
9.	31.7 %	31.5 %	0.2 %	113.6 MHz
10.	32.4 %	32.5 %	0.1 %	113.6 MHz
Deviation Average			**0.11 %**	

The testing was conducted outdoors by comparing the 30 Hz AM modulation percent reading results on the portable DVOR receiver design with those of the PIR NORMARC 7710 NAV Analyser. According to the test results, the average deviation of 30 Hz AM modulation percent was determined to be 0.11%.

Table 4. Average Data Deviation

No	Location	Deviation Average
1.	Navigation Laboratory	0.43 %
2.	Classroom	0.22 %
3.	Outdoor	0.11 %
Overall Average Deviation		**0.25 %**

Tests conducted in navigation laboratories, classrooms, and outside areas reveal average deviation values of 0.43%, 0.22%, and 0.11% respectively. Based on these deviation data, the overall average deviation is 0.25%. The average value of the overall deviation indicates that the design device functions well.

Implementation

The final stage of the waterfall method is the maintenance stage. During this stage, software modifications are made following implementation to improve results, correct errors, and enhance performance and quality. At this point, the design tool software has source code added to it. This addition aims to enhance the stability of the results displayed in the software. The added source code is "rate = abs(rate)".

Discussion

The Portable DVOR receiver is designed to display a range of parameters, including the azimuth and 30 Hz AM modulation percentage, using RTL-SDR R820T2 and Python programming language. Unfortunately, during the testing phase, the azimuth parameter could not be displayed, thus only showing the modulation percentage on the interface of the Portable DVOR receiver. The cause of the failure was found to be inaccurate source code writing in the azimuth reading section and the failure to split the AM and FM signals.

The average deviation of the percent modulation reading on the Portable DVOR receiver design against the PIR NORMARC 7710 NAV Analyser is 0.25%. The deviation value was obtained by comparing the results of the 30 Hz AM modulation percentage readings between the Portable DVOR receiver design and the PIR NORMARC 7710 NAV Analyser. Based on the average deviation results, the design device is deemed to be functioning accurately.

CONCLUSION

The RTL-SDR R820T2 device is employed as a radio signal receiver in the portable DVOR receiver design. The radio signal data is processed on a computer or laptop utilizing the Python programming language. Programming is carried out in a text editor app known as Visual Studio Code that utilizes the Python programming language. The interface of the equipment displays results in terms of the percent modulation of 30 Hz AM. Measurements of the portable DVOR receiver design, utilizing the RTL-SDR R820T2 and programmed with Python, are taken and analysed by comparing them with readings from the PIR NORMARC 7710 NAV Analyser. There was a deviation of 0.25% found between reading results of the design device and the PIR NORMARC

7710 NAV Analyser. These outcomes demonstrate good performance of the design device.

REFERENCES

[1] D. J. P. U. Departemen Perhubungan, "PERATURAN DIREKTUR JENDERAL PERHUBUNGAN UDARA NOMOR: SKEP/83/VI/2005 TENTANG PROSEDUR PENGUJIAN DI DARAT (GROUND INSPECTION) PERALATAN FASILITAS ELEKTRONIKAL DAN LISTRIK PENERBANGAN," 2005.

[2] R. A. White, R. N. Clarke, and R. W. Yell, "VOR waveform synthesis and calibration," *IEE Proc. F Commun. Radar Signal Process.*, vol. 128, no. 7, pp. 443–450, 1981, doi: 10.1049/ip-f-1.1981.0076.

[3] F. Heriyanti and A. Ishak, "Design of Logistics Information System in the Finished Product Warehouse with the Waterfall Method: Review Literature," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 801, no. 1, 2020, doi: 10.1088/1757-899X/801/1/012100.

[4] Z. Yang, J. Ye, J. Liu, P. Yang, and F. Liang, "Digital Signal Processing Technology of DVOR Navigation Signal," in *Lecture Notes in Electrical Engineering*, vol. 516, 2020, pp. 290–295.

[5] V. A. R. Pratiwi, Y. Suprapto, and M. M. Sukma, "Modifikasi Rancangan Informasi Alarm Dvor (Doppler Very High Frequency Omnidirectional Range) Berbasis Android," *Semin. Nas. Inov. Teknol. Penerbangan*, pp. 1–4, 2020.

[6] S. Sandmann, J. Bredemeyer, T. Schrader, and H. Garbe, "Numerical time domain method for calculation of the indicated DVOR bearing in dynamically disturbed scenarios," *IET Microwaves, Antennas Propag.*, vol. 13, no. 15, pp. 2687–2692, 2019, doi: 10.1049/iet-map.2018.5591.

[7] F. Fatonah, M. Wildan, S. Purnomo, and A. M. Soleh, "Design of High Gain Low Noise Amplifier at Base Station Receiver VOR Equipment for Ground Check Monitoring as Learning Media in Civil Aviation Design of High Gain Low Noise Amplifier at Base Station Receiver VOR Equipment for Ground Check Monitoring as Le," no. October, pp. 397–402, 2020.

[8] B. B. Harianto, Y. Suprapto, L. Winiasri, and M. F. Amansyah, "Studi Ekperimental Penerima Ads-B Menggunakan Rtl 1090 Dan Rtl-Sdr R820T2 Di Bandara Juanda Surabaya," *J. Penelit.*, vol. 4, no. 3, pp. 20–28, 2019, doi: 10.46491/jp.v4e3.346.20-28.

[9] R. Akeela and B. Dezfouli, "Software-defined Radios: Architecture, state-of-the-art, and challenges," *Comput. Commun.*, vol. 128, no. July, pp. 106–125, 2018, doi: 10.1016/j.comcom.2018.07.012.

[10] C. Laufer, *THE HOBBYIST'S GUIDE TO RTL-SDR: REALLY CHEAP SOFTWARE DEFINED RADIO*. 2014.

[11] B. Uengtrakul and D. Bunnjaweht, "A cost efficient software defined radio receiver for demonstrating concepts in communication and signal processing using Python and RTL-SDR," *2014 4th Int. Conf. Digit. Inf. Commun. Technol. Its Appl. DICTAP 2014*, pp. 394–399, 2014, doi: 10.1109/DICTAP.2014.6821718.

[12] R. Munir and L. Lidya, *Algoritma dan Pemrograman Dalam Bahasa Pascal, C, dan C++*, Keenam. Bandung: Penerbit Informatika Bandung, 2016.

[13] G. Van Rossum, *Python Tutorial*, Release 3. Amsterdam: Python Software Foundation, 2022.

[14] W. McKiney, *Python for Data Analysis*. United States of America: O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA95472, 2013.

[15] Y. Bassil, "A Simulation Model for the Waterfall Software Development Life Cycle," *Int. J. Eng. Adv. Technol.*, vol. 02, no. 05, Jun. 2012, [Online]. Available: http://www.ijircce.com/upload/2015/may/13_A_Simulation.pdf.