Implementation of Satellite Payload Program Reconfiguration on Low-Cost Software-Defined Radio

Nurul Fadilah[1], Bina Pratomo [1], Nurul Muhtadin[1], Moedji Soedjarwo[1], Reyza Pratama[1]

[1]Satellite Technology Center, National Institute of Aeronautics and Space (LAPAN), Indonesia

e-mail: bina.pratomo@gmail.com

Received: 10-07-2022. Accepted: 16-01-2023. Published: 31-01-2023
DOI: 10.30536/j.jtd.2022.v20.a3938

Abstract

A software-defined radio (SDR) is a radio communication system that uses reconfigurable software-based components for digital signal processing and conversion. These radio devices, in contrast to typical radio communication systems, are extremely adaptable and versatile. This technology has recently been implemented in CubeSat payloads for connecting the continuously expanding wireless world. SDR devices provide flexibility and versatility by allowing the payload configuration and application to be modified in orbit. This study presents the concept and implementation of an SDR ground simulator utilizing inexpensive hardware rather than a typical computer. The simulator runs on Raspberry Pi hardware with Linux OS and is written in the Phyton programming language. This work presents a model and algorithm of satellite software reconfiguration implemented in the SDR hardware ground simulator. The concept is simple and easy to implement and is potentially useful to be used in future satellite missions.

Keywords*: SDR, reconfigurable payload, satellite, on-the-air program update.*

1. Introduction

Indonesia is the largest archipelago country located at the confluence of four tectonic plates, namely the Asian Continent plate, the Australian Continent plate, the Indian Ocean plate, and the Pacific Ocean. Most of Indonesia's region is an unstable area where frequent earthquakes and volcanic eruptions occur. In addition, other natural disasters like floods, tsunamis, hurricanes, landslides, and forest fires are also common in Indonesia which result in the loss of life, and property, economic problems, and psychological problems(Ali et al. 2019). In an attempt to mitigate the disaster, Indonesia planned to develop an early warning system and telecommunications infrastructure that can reach remote areas that are difficult for terrestrial communications to reach. As part of the disaster mitigation effort, LAPAN planned a low-orbit constellation satellite mission that acts as a disaster-related data-collecting platform for an early warning system(Fitrianingsih et al. 2020). In addition to that, each satellite will also carry multiple payloads based on SDR devices for various applications.

The satellites must have the flexibility to accommodate the necessary needs. These satellites will carry an AIS receiver, ADS-B receiver, amateur communications, and data collection platforms to speed up the delivery of the sensor data needed. Based on BNPB data, there have been many natural disasters such as earthquakes, volcanic eruptions, flash floods, and landslides in the last decade. A total of 19737 disasters occurred in that period (Pratomo et al. 2021). In addition to natural disasters, Indonesia, with its geographical location and topography, often also causes transportation accidents, both by land (railway), sea, and air.

An SDR device is a wireless communication system whose tasks are carried out using software on a computer rather than specific hardware. In its development as a transceiver device, SDR can transmit, receive, modulate, and demodulate a signal by changing its parameters according to its environment and propagation needs (Mitola 1995). The cognitive properties in SDR, namely the ability of a device to execute a command based on the newly inputted parameter or the previous parameter were introduced in 1999 (Mitola 1999).

Software-defined radio SDR has been widely used for many applications in terrestrial communication and signal reception(Maheshwarappa, Bowyer, and Bridges 2015), including data from satellite imagery. In recent years, devices based on software-defined radio (SDR) have been implemented as satellite on-board payloads. SDR size is small enough to fit in a CubeSat, and many satellite component developers provide SDR-based payloads for various applications. Automatic Identification System (AIS), air traffic monitoring (ADS-B), the Internet of Things (IoT), and machine-to-machine (M2M) communications are some of the applications that are feasible to implement on an SDR device.

Satellite payloads that use SDR devices benefit from their flexibility in terms of software updates, which allows the software to be reconfigured once in orbit. The reconfigurable payload model was suggested, with an emphasis on developing the intermediate frequency. By reprogramming the payload application once the satellites are in orbit, SDR offers greater flexibility in many ways including space applications. SDR allows the system to operate in several frequency bands, allowing numerous applications to be implemented on the same hardware, such as the Automatic Identification System (AIS), air traffic monitoring (ADS-B), IoT, M2M communications, and many more. A concept of reconfigurable SDR as a satellite payload was proposed (Perlaza, Hoyos, and Vera 2006), in which a model for reconfigurable payload was designed to replace the traditional transponder system for a Geostationary satellite payload.

The reconfigurable payload concept is implemented on the Indonesian disaster constellation mission that is equipped with multiple SDR devices to provide flexibility in terms of application. For LEO orbit where the payload system is typically small in size, the reconfigurable SDR needs to be small and simple to operate. This study presents a simple system and algorithm to reconfigure the payload application based on the SDR module. The system includes some hardware that represents the satellite subsystem's function as well as the ground station. The system simplifies the integration process and manages multiple parameters. It also provides advantages such as ease of development, reconfiguration flexibility, and cost-effectiveness. To communicate with the satellite, a UHF-band communication link is utilized.

2. Method

SDR-based devices have been widely used for ground communication applications however only recently it started to be implemented on satellites. The reconfigurable satellite payload concept was proposed by (Mast 2011,), implementing the concept of full or partial reconfiguration of the software using FPGA-based SDR. The reconfiguration changes the purpose of the communication payload with the initial working frequency Ka-Band to X-Band for radar applications. The principle of IoT software update was proposed by (Bauwens et al. 2020). This demonstrates the flexibility of SDR to be reconfigured once it is in orbit.

In SDR devices, the processing of a signal is not carried out continuously (such as the voltage on the antenna). The received periodic samples will be processed by a personal computer. These samples were obtained from wireless devices connected using USB or Ethernet. Instead, the samples generated in the computer are transmitted through this device. there are many types of peripheral devices such as USRP by Ettus Research, Blade RF, Hack RF, RTL-SDR, and Adalm-PLUTO.

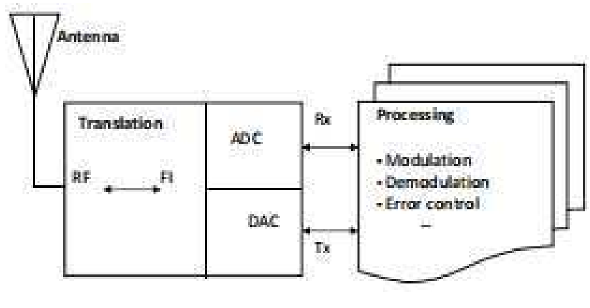

Figure 2-1: General SDR Architecture(Mitola 1999)

SDR consists of hardware and software that receives the RF signal using an antenna as shown in Fig. 2-1. Then the RF signal is processed with a local oscillator frequency, resulting in an intermediate frequency (FI) signal. The resulting signal is then digitized by the converter (DAC) and sent to the processing unit. In this system, the signal is processed in-phase (I) and quadrature (Q) (Hosking, 2016).

To simulate the process of updating SDR software on the air, we built a model that represents the satellite and ground system that are connected via wireless utilities as shown in the model diagram in Fig. 2-2. The satellite subsystems included in the model are TTC which is represented by Pluto SDR, OBDH using Rasberry Pi, and the payload which uses Raspberry Pi and LimeSDR. In the simulator, the GS represented by LimeNet Micro and Computer will send commands to the TTC, and the command will be received and passed on to the OBDH for the verification process. Once the commands are verified, the OBDH passes the command to the payload to update the software and the payload executes the process and sends notification whether the process was successfully executed or not.

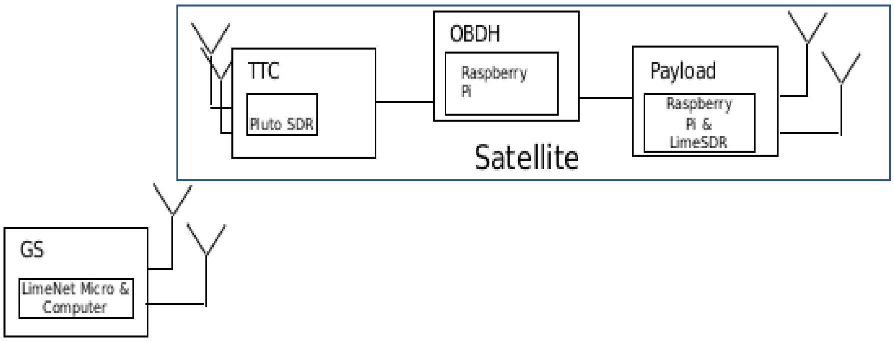

Figure 2-2: Model Diagram of SDR-based Satellite and GS Simulator.

The space segment components in this simulation are simplified into TTC, OBDH, and Payload. Each block is described as follows:

1. Ground Station (GS)

The ground station used a computer device as a user interface that contained software with a GUI display. It has a feature to select the SDR program to be uploaded. Once it is selected, the program is sent to LimeNet Micro and transmitted to the satellite using the UHF band via TTC.

2. Telemetry and Telecommand (TTC)

TTC receives analog signal (RF) from GS and converts it to digital data.

3. On-Board Data Handling (OBDH)

As the brain of the satellite, it interprets all the data coming from the TTC and processes the data. In this paper, the Raspberry Pi is used as OBDH.

4. Payload

The satellite payload functions to carry out the mission that has been assigned by OBDH. The payload sends the status of the SDR program replacement process to OBDH to be forwarded to GS via TTC. In this paper, Raspberry Pi and LimeSDR are used as payloads.

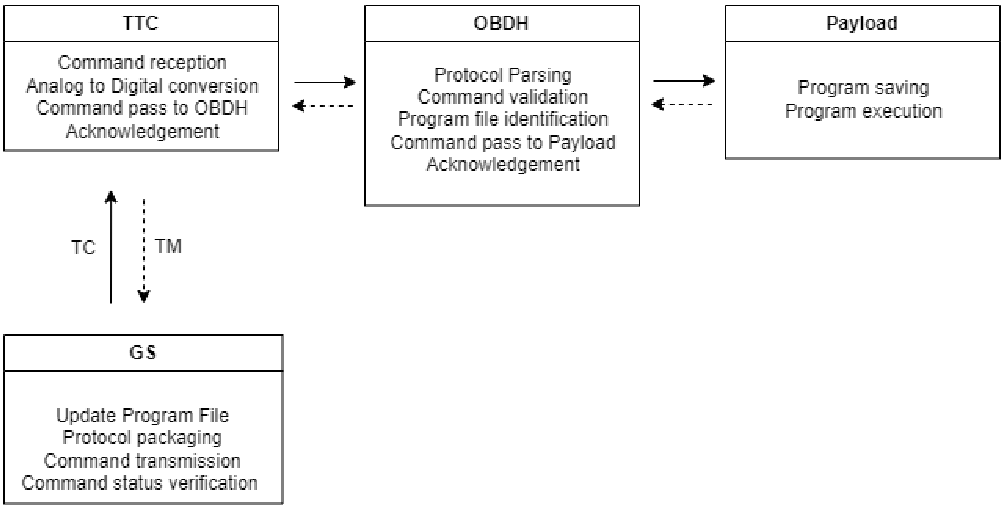

Figure 2-3: Functional Diagram of SDR Reconfiguration Model.

Fig. Figure 2-3 shows the functional diagram of each hardware in the model. The model can work with any program file whether it is a parameter configuration or an application program. To perform validation of the model and the algorithm quickly, we used the SDR frequency reconfiguration program to shift the frequency from 402.825 MHz to 440.651 MHz. A handheld device (HT) and spectrum analyzer were used to verify that the frequency change command was successfully sent to the payload and executed, in addition to verification via telemetry.

3. SDR Reconfiguration Algorithm

The design and simulation of the on-the-air SDR program update follow the following flowchart presented in Fig. 3.1 which shows the program delivery algorithm. The process starts by configuring the LimeNet Micro parameters and is followed by program file selection. The number of the SDR program CRC bytes was determined then the header, CRC, and footer were added to the SDR program so that OBDH can identify the type of command received from GS. The package is then sent to TTC which converts it to digital data and then sends it to OBDH to be interpreted. Upon receiving the command, OBDH checks the CRC to verify the command validity and sends ACK if it is successfully received. If not, NACK is sent to GS and the GS will try another attempt to deliver the program up to 3 times. If the program still can not be received by the satellite successfully after the third attempt, the satellite no longer sends NACK, and the process is terminated.

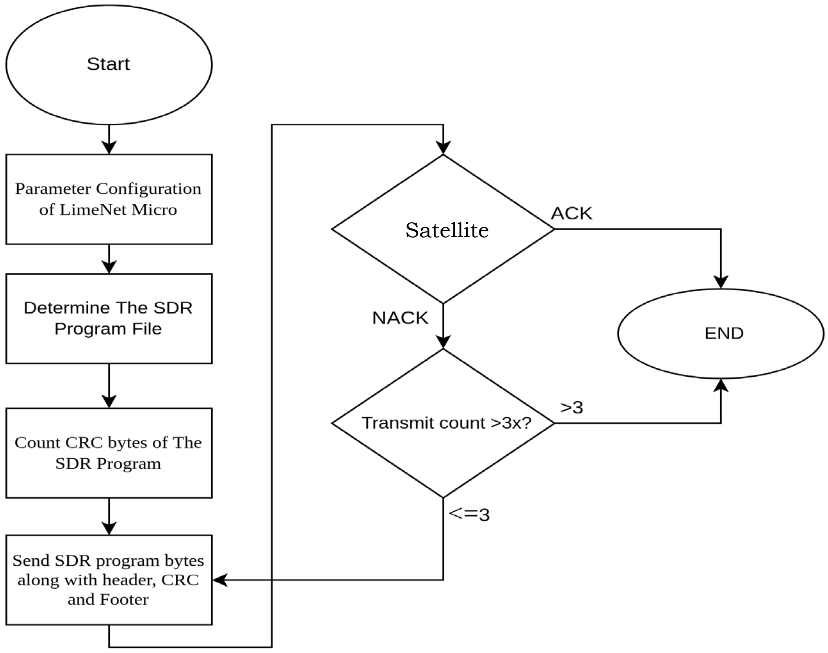

Figure 3-1: Program Delivery Algorithm from GS.

The detailed process of program update and command acceptance in the OBDH model is shown in Fig. 3-2. The process starts after the OBDH receives the command from TTC. Initialization is conducted by configuring the SDR parameters, followed by an input byte check. The process continues by checking the header and footer to see if the command is the program reconfiguration file. If the header and footer are not verified, the OBDH checks the time out and sends NACK to GS via TTC. If the header and footer are verified that the command contains an updated program file, it sends a command to the payload to update the program file for later execution or execute a file that is already stored in the payload memory.

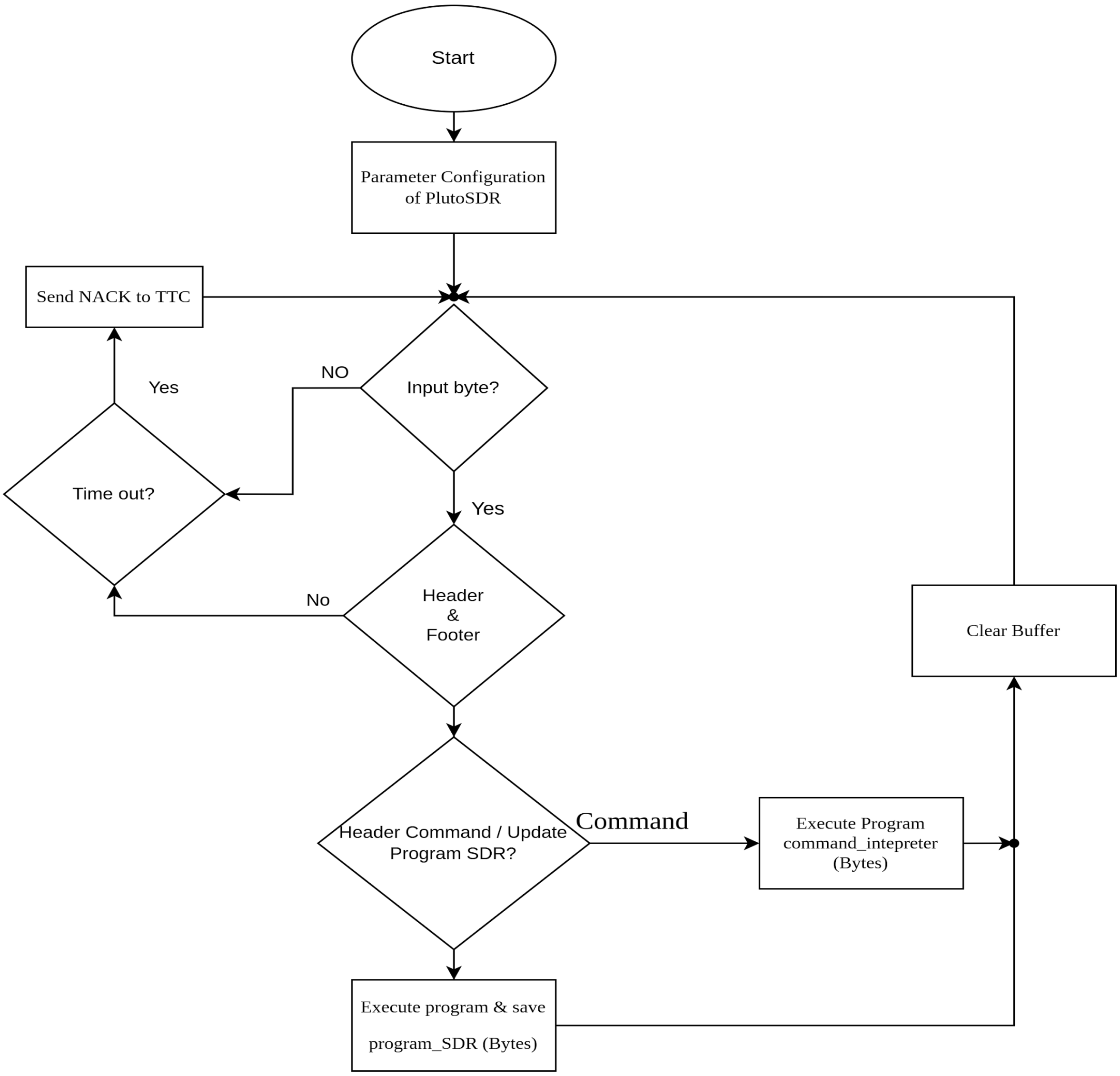

Figure 3-2: Updated Program and Command Acceptance Algorithm in OBDH Model.

In Fig. 3-3, the algorithm for saving the update program file is presented. The procedure begins with checking the CRCR which is the last 2 bytes of CRC received then determines the data byte which is the first bytes up to the bytes before CRCR. Then calculate CRCB which is CRC count and compare it with CRCR. If the CRCR and CRCB are equal, save the updated program and send ACK to GS, and if the process then the process is terminated.

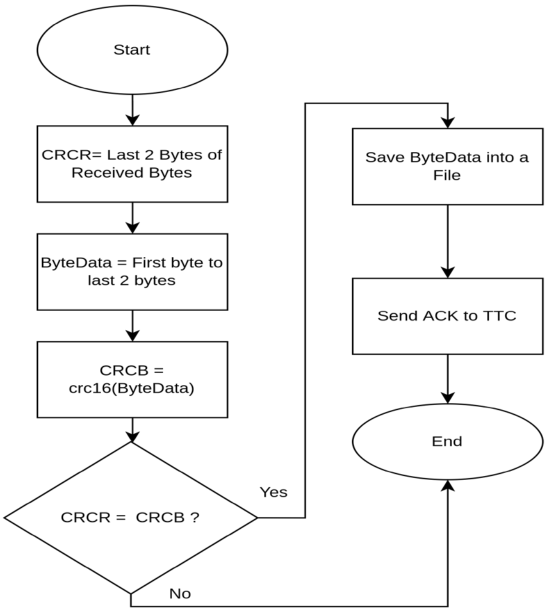

Figure 3-3: Save Update Program File Algorithm.

The command execution algorithm is almost similar to the saving update program algorithm. The only difference is instead of saving the incoming program file, it executes the program which already exists or was previously stored in the payload memory. The flow diagram is shown in Fig. 3-4.

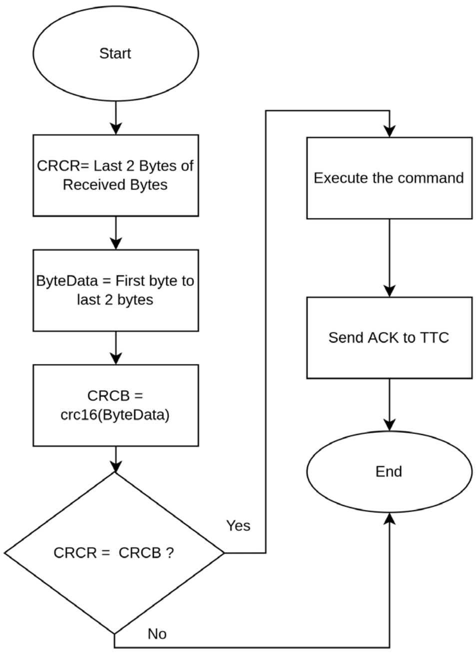

Figure 3-4: Reconfiguration Command Execution Algorithm.

4. Result and Analysis

This paper applies the SDR device to simulate the mechanism for updating the SDR program on payload when the satellite is in orbit. Updating the SDR program on the air is proposed to replace the existing payload program in the satellite according to the specified needs.

Software Defined Radio (SDR) is a radio transceiver electronic device that can be configured using software parameters. SDR is the latest technological development that makes it easier for developers of radio-based electronic devices without physically changing the hardware configuration. SDR parameter configuration settings are done using the software. An example of radio parameters set via software include working frequency, bandwidth, sample rate, filter, and power gain. The parameters can vary depending on the type of SDR device. In this paper, the working frequency is set with different values to verify that the program can be updated remotely. The software used to update the existing payload program runs on a computer device. Physically, the SDR device is connected to the mini-computer via a USB port. In the configuration of SDR parameters, SDR devices can be applied for various purposes, such as a satellite payload transceiver. With the SDR device used in the satellite payload, the satellite operator can configure it for various desired missions. Operators can configure SDR on satellite payloads remotely using ground station infrastructure via radio waves. The SDR parameter configuration is packaged in a file and sent to the satellite directly.

Table 4-1: The Hardware Used in the Research

Device	Quantity	Function	Device
SDR	3	Transceiver for TTC, Payload, Ground Station	SDR
Raspberry Pi	3	Computer mini for OBDH, Payload, Ground Station	Raspberry Pi
Computer	1	Simulation control	Computer

Table 4-1 shows the list of hardware used in this research. Those hardware are set into a configuration that is shown in Figure 3-4 and they were set up as seen in Fig. 4-1.

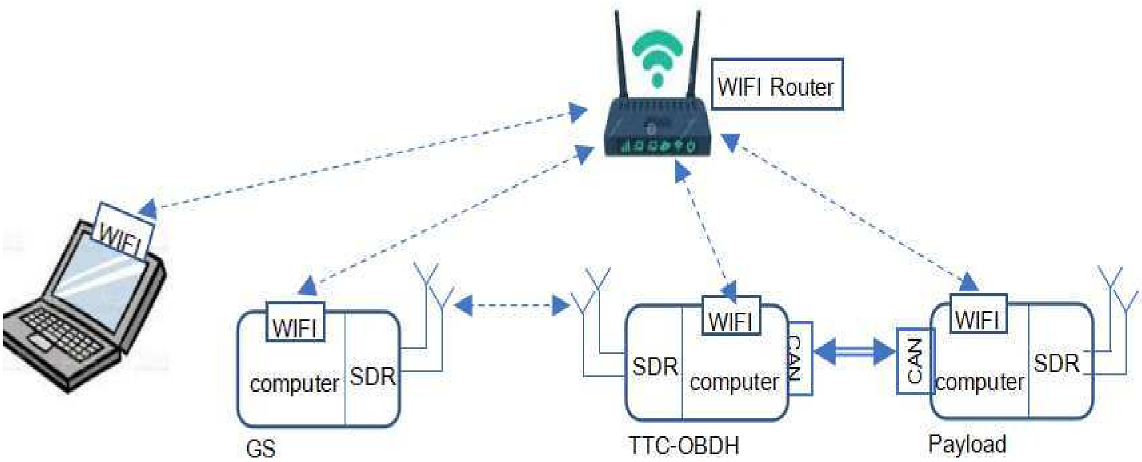

Figure 4-1: The Hardware Configuration.

Figure 4-2: The Hardware Set Up.

In this simulation, there are five Python programs to verify that the system can work properly.

- send_program.py: to send the SDR program contained in the ground station computer
- Exec-redundan.py: to send the command
- pitonSC.py: to receive the updated program from the ground station
- Update_redundan_pyld.py: to save the updated program
- Command_interpreter.py: to execute the sent command

The ground station uses a computer device as a user interface that contains software that has a feature to select the SDR program to be uploaded. Once selected, the software is sent to LimeNet Micro, and then, transmitted to TTC. The software also has a feature to activate the SDR program on the payload by sending a command.

TTC, by using PlutoSDR mini converts electromagnetic waves into discrete signals with SDR program data output being transmitted and commands. OBDH receives the SDR program and commands to detect whether it is valid; if not, OBDH sends the status to GS via TTC. If it is valid, then it is processed. Data containing the SDR program is temporarily stored in OBDH, while for commands, it is necessary to reinterpret the types of commands. Suppose the command contains a command to activate the SDR program on the payload. The SDR program stored in OBDH is sent to the payload while instructing the payload to disable the currently running SDR program and activate the new SDR program. The payload will send the SDR program replacement process status to OBDH to be forwarded to GS via TTC.

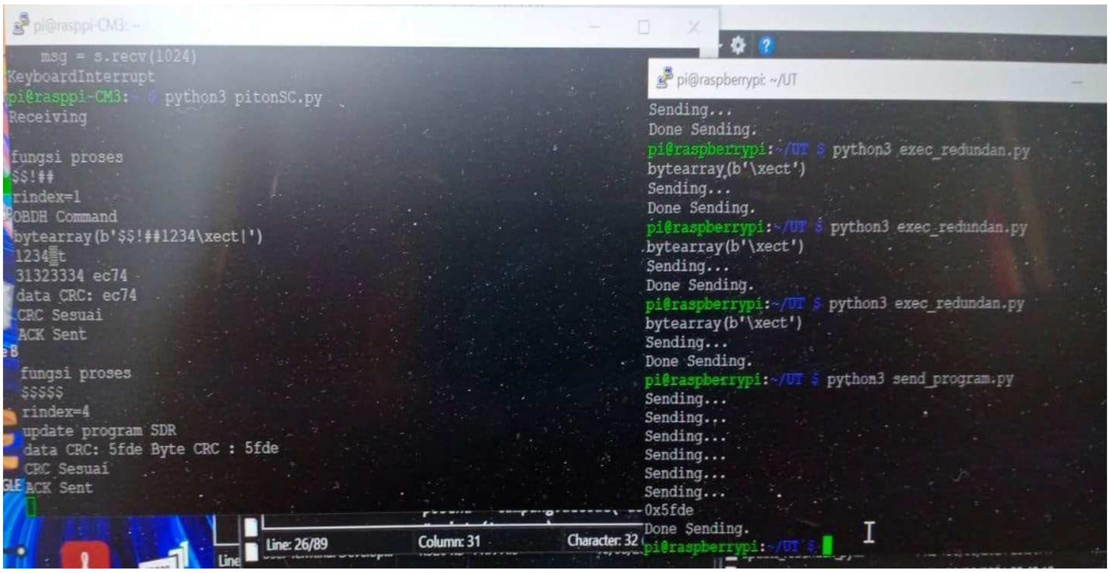

Figure 4-3: Updated program and command send process.

On the right side of Fig. 4-3, it shows the process of sending commands for payload program updates on the GS model. On the left side of Fig. 4-3, the first few lines show the process when the OBDH received the command from GS via TTC. The OBDH performed protocol parsing, and command validation by calculating the CRC count and then sent ACK to GS via TTC. The lower part is the process to command the payload to update the program.

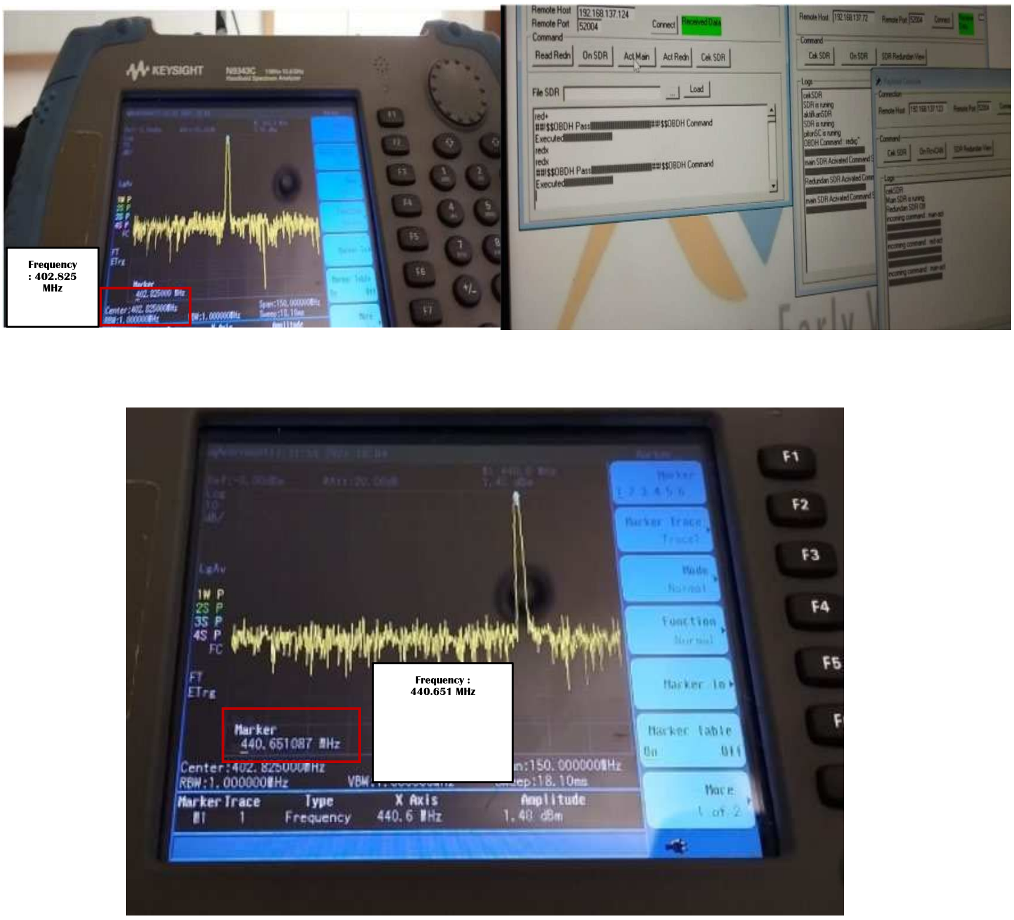

Figure 4-4: Frequency Shift before Program Update (top), After Program Update (bottom).

The transmission testing software for the SDR on the air configuration update application has been tested. It was tested by sending an SDR configuration file with a carrier signal transmission of 402.825MHz. This test uses the final primary function payload, which transmits a signal with a carrier frequency of 440.425 MHz. As a visual indicator, a spectrum analyzer shows the signal. The test results show that the payload frequency changes from 402.825 MHz to 440.651 MHz as in Fig. 4-4.

5. Conclusion

The objective of this work was to develop a model and algorithm to simulate the process of updating or reconfiguring the program on a satellite payload based on SDR and to implement it using low-cost SDR devices. The research demonstrates the capability of SDR to be reconfigured via telecommand. This positive result shows that it is possible to update the payload program once the satellite is operated in orbit. The concept is easy to implement and it works with any type of program file whether it is a configuration file or application file, adding flexibility to the satellite mission.

Acknowledgments

The authors sincerely appreciate the Indonesian National Institute of Aeronautics and Space LAPAN and the LAPAN A5 team for supporting this research.

Contributorship Statement

NF and BP are the main contributors to this research. Other authors contributed as follows: NM developed the simulation; MS designed the method, analyzed the results, and prepared the manuscript; RP prepared the manuscript.

References

Ali, M. S. S., M. Arsyad, A. Kamaluddin, N. Busthanul, and A. Dirpan. 2019. "Community-Based Disaster Management: Indonesian Experience." *IOP Conference Series: Earth and Environmental Science* 235(1):012012. doi: 10.1088/1755-1315/235/1/012012.

Bauwens, J., P. Ruckebusch, S. Giannoulis, I. Moerman, and E. D. Poorter. 2020. "Over-the-Air Software Updates in the Internet of Things: An Overview of Key Principles." *IEEE Communications Magazine* 58(2):35–41. doi: 10.1109/MCOM.001.1900125.

Fitrianingsih, E., Dwiyanto, P. A. Budiantoro, B. Pratomo, and H. Mayditia. 2020. "Mission Analysis of Indonesia Low Earth Micro Satellite Constellation." in *Proceedings of the International Astronautical Congress, IAC*. Vols. 2020-Octob.

Hosking, Rodger H. 2016. *Software Defined Radio Handbook*. 12th ed. New Jersey: Pentek, Inc.

Maheshwarappa, M., M. Bowyer, and C. Bridges. 2015. "Software Defined Radio (SDR) for Parallel Satellite Reception in Mobile/Deployable Ground Segments." in *Small Satellite Conference 2015, AIAA/Utah State University 29th Annual AAIA/USU Conference on Small Satellites*.

Mast, A. W. 2011. "Reconfigurable Software Defined Payload Architecture That Reduces Cost and Risk for Various Missions." Pp. 1–5 in *2011 Aerospace Conference*.

Mitola, J. 1995. "IEEE Communications Magazine." *IEEE Communications Magazine, Vol. 33*, 26–38.

Mitola, J. 1999. "Cognitive Radio for Flexible Mobile Multimedia Communications." Pp. 3–10 in *1999 IEEE International Workshop on Mobile Multimedia Communications (MoMuC'99) (Cat. No.99EX384)*.

Perlaza, Samir M., Evelio A. Hoyos, and Pedro V. Vera. 2006. "Reconfigurable Satellite Payload Model Based on Software Radio Technologies." in *3RD IEEE INTERNATIONAL CONGRESS OF THE ANDEAN REGION – ANDESCON*. Andescon.

Pratomo, B., E. N. Nasser, A. H. Qonita, N. Muhtadin, M. A. Arifin, and M. Soedjarwo. 2021. "User Terminal Prototype Development for LAPAN's Low Orbit Constellation Satellite." Pp. 1–7 in *2021 IEEE International Conference on Aerospace Electronics and Remote Sensing Technology (ICARES)*.