

Design and Realization of Automatic Packet Reporting System (APRS) for Sending Telemetry Data in Nano Satellite Communication System

R. Patmasari^{1*}, I. Wijayanto¹, R.S. Deanto¹, Y.P. Gautama¹, H. Vidyaningtyas¹

¹ School of Electrical Engineering, Telkom University, Bandung, 40287, Indonesia

* raditiana@telkomuniversity.ac.id

Manuscript received February 19, 2018; revised April 10, 2018; accepted June 2, 2018

Abstract

To communicate with the ground station, nano satellite requires a communication system that serves in real time to regulate the procedure with ground station. One of the missions of nano satellites is the retrieval of telemetry data from sensors that transmitted using APRS technology (Automatic Packet Reporting System). We designed a prototype to monitor sensor by utilizing the APRS. Real-time sensor data was transmitted to monitor through radio using AX.25 protocol. On the transmitter side, an APRS Tracker integrated with a microprocessor ATMEGA 1284P to modulate the AFSK signal. The existence of this AFSK signal is used to transmit and receive telemetry data using handy talky. We use a computer with AGWPE software to demodulate the AFSK signal and UI-View32 to show the telemetry data. The results showed that the telemetry data can be displayed on the receiver side and acquired in real time. The accuracy of temperature sensor received by the receiver is 92.97%, and humidity sensor is 90.57%. There was a delay of 1 second for the reception of telemetry data directly.

Keywords: APRS; APRS tracker; Nanosatellite; Telemetry data

DOI: <https://doi.org/10.25124/jmeecs.v4i1.1692>

1. Introduction

Nanosatellite is a satellite that has a size of 10x10x10 cm and weight less than 10 kg [1]. Nanosatellite orbits the Earth at the Low Earth Orbit (LEO) [2]. The development of nanosatellite and microsatellite industry in the past ten years makes researchers competing to conduct research to produce cheap and good satellite. In designing small satellites such as nanosatellite, there are challenges faced by the researchers, including the limited power supply, transponder size, and communication between satellites. Several innovations on nanosatellite communication have been conducted, one of them is on Iridium satellite networks which work on LEO and has a limited line of sight time, which is about 15 minutes [3]. To solve the problem, researchers have designed a transceiver that acts as a communication gateway with an earth station using a microcontroller [4]. Another research conducted by Popescu et al. [5] studied the design of communication systems at CubeSat, the results showed that the proposed method could produce SNR in the range of 10-20 dB which indicates an acceptable bit error result.

A research on robust downlink communication by Mevada et al. [6] produced a satisfactory performance that was only 1dB away from the theory for LEO satellite channels without fading. This result

was obtained without sacrificing channel efficiency by limiting overhead of 15%. Another research on nano-satellite communication systems for weather missions has also been done by Tresvig et al. [7] using a modulated GFSK signal with 9600 bit/s data rate in half-duplex mode. Furthermore, congestion mitigation in nanosatellite network was done to reduce the buffer occupancies in the nanosatellite and to improve the data delivery time [8].

In this study, the planned satellite mission is to deliver telemetry data using Automatic Packet Reporting System (APRS) technology. APRS is commonly used in the process of monitoring sensors [9][10][11]. We use APRS to monitor temperature and humidity. The data is transmitted continuously along with the acquisition of the observed sensor data. Data transmission using APRS can be an alternative means of communication [12]. We use microcontroller to modulate audio frequency shift keying (AFSK) signal, refer to a research by Widada et al [13]. The object monitoring system research by Mahendra et al. [14] was done by sending the position data received by GPS to the receiver side using APRS technology.

2. System Design

The system is designed in two segments as shown in Fig. 1. The first one is space segment. This segment is a nanosatellite prototype consists of a microcontroller, APRS module, and sensors. The second segment is earth segment, consist of personal computer and software to display the received data. Both parts are communicated by using RF module.

2.1 Space segment

For the space segment, we have designed a series of payloads from the APRS nanosatellite prototype that serves to measure the temperature and humidity. In the space segment, there are APRS tracker components, microcontrollers as data processors, sensors and RF modules.

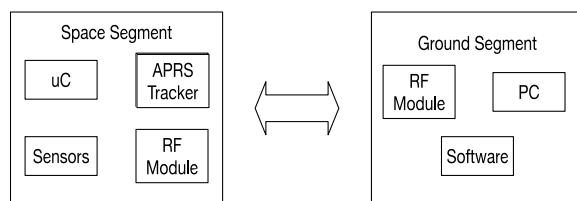


Fig. 1. System design.

APRS tracker is used to parsing the incoming data, then converted into APRS format data encapsulation. In the system, the telemetry data generated by the sensors is received by APRS tracker. The tracker which integrated by a microcontroller that generates AFSK signal. The data then reprocessed into a signal emitted by the transmitter. In this research, we use TinyTrack4 as the APRS module. TinyTrack4 is a radio-based interface that can transmit and receive the data position and the result of digital pieces of information [15]. The firmware version used is tt4_alpha_v072. It connects to the computer and handy talky using RS-232 connector. The radio transmission used by APRS utilizes the AX.25 protocol which is a standard protocol used in radio amateurs [16], nano/pico satellite communication [17], [18] and remote monitoring [6], [19][20]. Radio packet transmitted on the Link Layer is sent in small blocks of data called frames. Several types of frames exist in the AX.25 protocol. In this research, we use Un-numbered Information (UI) frames. UI frames are transmitted spreadingly to all existing recipients. If there is any packet which is not received, the packets will not be resent. This is not a problem, because in APRS there is periodic data packet transmission [16].

Table 1 shows the UI frame format used, in which there are several fields containing data in binary format. These fields are:

1. Flag characters used in this research are 01111110 (0x7E).
2. The destination address contains APRS destination callsign or APRS data which is six characters combined with the SSID.

3. The source address includes callsign and SSID of the sender.
4. Digipeater address carries callsign digipeater, or if not using digipeater can be filled with APRS path (WIDE n-N).
5. The control field and the protocol ID always set to 0x03 and 0xF0 value respectively.
6. The information field contains APRS data.
7. For the frame check sequence, 16-bit value calculated from all data in the packet, except the flag characters. The sender generates the FCS and used by the recipient to ensure that the transmitted data does not experience a sending error.

The parameters need to be configured on APRS are callsign, smart-beaconing which active if set to True, and telemetry rate that contains every few seconds of telemetry data needs to be transmitted.

Table 1. AX.25 UI Frame Format [12]

AX.25 UI Frame Format	
	bytes
Flag	1
Destination Address	7
Source Address	7
Digipeater Addresses (0-8)	0-56
Control Field	1
Protocol ID	1
Information Field	1-256
FCS	2
Flag	1

In the space segment, the sensor used is the LM35 sensors to sense the temperature inside the nanosatellite. This sensor uses analog voltage to communicate with the microcontroller [21]. Another sensor used in the nanosatellite is the soil moisture sensor, the sensor is used to monitor the moisture outside the nanosatellite. The output of the sensor is an analog value of the moisture level.

The two sensors are powered alongside with microcontroller. The microcontroller is used to calculate the output of the sensor and then compared with the data issued by APRS tracker. The communication media used is through VHF radio frequency. The frequency used is 144.390 MHz amateur radio frequency which is allocated for APRS data traffic.

2.2 Ground segment

We use AGWPE software to convert analog data to digital then decode the received data packets. We also use UI-View software as a data packet processing to be displayed in the form of digital maps to facilitate the reading of the received

telemetry data packets. The calibrated Indonesian map is used for telemetry data retrieval. An APRS data format displayed in UI-View32 is shown as:

```
16:50:52RNOCALL>APTT4,WIDE1-1Port=2
<UI Len=34>:
T#021,065,999,794,311,999,00000011.
```

The data format is explained in Table 2.

3. Results and Discussion

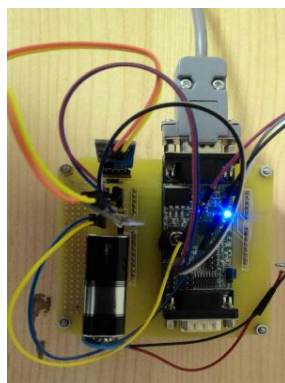
The APRS prototype structure of nanosatellite is divided into upper and lower layer, as shown in Fig. 2 (a) and (b). The top layer consists of APRS module, Soil Moisture sensor as moisture level sensor, LM35 as a temperature sensor, and 9 Volt battery as power supply for the system. The lower layer consists of the microcontroller.

3.1 Sensor data integration test

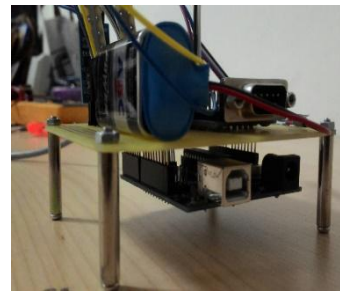
Testing is performed on every sensor data to check their validity. The result will be displayed on Teraterm and microcontroller compiler. Microcontroller compiler is used as a reference while Teraterm software is used as the terminal tested to check whether the sensor data is valid or not after inserted in the APRS system.

Table 2. APRS Data Format in Receiver

Data Formats	Explanations
16:50:52R	Western Indonesian times
NOCALL	Callsign and SSID
APTT4	APRS Tracker Firmware
WIDE1-1	APRS Path
Port=2	Port AGWPE
<UI Len=34>	Numbers of transmitted characters
T#021,065,999,794,311,999,00000011	Telemetry data received



(a)



(b)

Fig. 2 The nanosatellite APRS prototype, (a) upper- and (b) lower layer

The result of sensor data testing on microcontroller compiler can be seen in Table 3. It shows that nanosatellite has been able to take the measurement value of each sensor. The sensor data retrieval continues even though there is no request from the earth station to send the data.

Sensor data obtained from measurement, Telemetry Data in Table 3, is then translated by the microcontroller compiler. The data T52H999 means that the numbers 52 and 999 represent data obtained from sampling analog signal generated by the temperature (T) and moisture levels/humidity (H) sensor, respectively.

By utilizing the concept of analog to digital converter that applies to temperature sensors and microcontroller, it is found that the result of analog signal sampling 52 is the temperature which is worth 25^o C. While, the result of analog signal sampling 999 is humidity which indicates “Dry Soil” meaning that the environment being measured is less water.

The reading result of the analog sensor follows the conditions [22]:

If the analog result reading is greater than 700, it indicates “In Water” means that the soil is lack of water. If the analog result reading is less than 300, it indicates “Dry Soil” means that the soil is too watery or too moist. If the analog result reading is between 300 and 700, the humidity of the environment is “Humid Soil” means that the soil is in proper moisture.

3.2 Sensor accuracy rate

The test was carried out by taking five samples reference data of temperature and humidity sensor then compared with output data of LM35 and Soil Moisture in microcontroller compiler and APRS module. After doing the above test methods, mathematical calculations will be done to the data obtained in the test to get the error and accuracy value. The result of the data acquisition is shown in Table 4 and 5.

The tables of the testing result show that the sensor data released on the Microcontroller compiler has a temperature accuracy rate of 92.97% and the humidity accuracy rate of 90.57%. Meanwhile, the

sensor data published on the APRS module has a temperature accuracy rate of 91.08% and the humidity accuracy rate of 90.57%.

Table 3. Sensor Data Test Result

Telemetry Data	Temperature	Moisture Levels/ Humidity
T52H999	24 ⁰ C	Dry Soil
T52H999	25 ⁰ C	Dry Soil
T52H999	25 ⁰ C	Dry Soil
T52H998	25 ⁰ C	Dry Soil
T52H998	25 ⁰ C	Dry Soil
T52H998	25 ⁰ C	Dry Soil
T52H999	25 ⁰ C	Dry Soil
T52H999	25 ⁰ C	Dry Soil
T52H999	25 ⁰ C	Dry Soil
T52H999	25 ⁰ C	Dry Soil

Table 4 Temperature Data Acquisition

Temperature Sensor Reference	Temperature Sensor Results in APRS module	Temperature Sensor Results in Simulator
26 C	28,32 C	27,83 C
26 C	28,32 C	27,83 C
26 C	28,32 C	28,32 C
26 C	28,32 C	28,32 C
26 C	28,32 C	27,83 C

Table 5 Humidity Data Retrieval

Humidity Sensor Reference	Humidity Sensor Results in APRS module	Humidity Sensor Results in Compiler
87%	976 (95,40%)	974 (95,21%)
87%	975 (95,30%)	975 (95,30%)
87%	974 (95,21%)	979 (95,69%)
87%	977 (95,50%)	982 (95,99%)
87%	978 (95,60%)	989 (96,67%)

Table 6 Accuracy Value of APRS Module and Compiler

Accuracy Value of APRS Module		Accuracy Value of Microcontroller compiler	
Temp	Hum	Temp	Hum
91,08%	90,35%	92,97%	90,57%
91,08%	90,46%	92,97%	90,46%
91,08%	90,57%	91,08%	90,02%
91,08%	90,23%	91,08%	89,67%
91,08%	90,12%	92,97%	88,89%

The error value occurs because the temperature and humidity sensor is in one load with APRS module and other components so that if there is a heated load like a battery, the temperature and humidity sensors will be sensitive to take data on the actual state of capacity.

3.3 Telemetry data reception delay and error

In the telemetry data reception test, the configuration in the APRS module should be noted. Configuration of APRS module includes Callsign used, Smartbeaconing, Configure, and Telemetry Rate which are already explained previously.

Table 7 shows that the first telemetry data was obtained at 07:27:07 with the result of temperature 054 and humidity of 999. So if converted into actual unit the result of temperature obtained is 26,36 C and the measured humidity shows result that the environment is less water.

The table shows the reception on each data packets is approximately 2 seconds compared to the initial configuration which is set every seconds. The results show that there is a delay of 1 second, which is occurs because the APRS tracker perform processing data needed to send data, not because of the data reception from the transmitter to the receiver.

Table 7 Telemetry Data Testing

Times	Received Telemetry Data
07:27:07	T#001,999,883,369,054,999,00000011
07:27:09	T#002,999,882,376,054,999,00000011
07:27:11	T#003,999,882,382,055,999,00000011
07:27:13	T#004,999,882,387,055,999,00000011
07:27:15	T#005,999,882,389,055,999,00000011
07:27:17	T#006,999,882,391,056,999,00000011
07:27:19	T#007,999,882,393,056,999,00000011
07:27:21	T#008,999,882,395,056,999,00000011
07:27:23	T#009,999,883,396,056,999,00000011
07:27:25	T#010,999,883,397,057,999,00000011
07:27:27	T#011,999,883,397,057,999,00000011
07:27:29	T#012,999,883,399,057,999,00000011
07:27:31	T#013,999,883,401,057,999,00000011
07:27:33	T#014,999,884,403,057,999,00000011
07:27:35	T#015,999,883,404,057,999,00000011
07:27:37	T#016,999,883,405,057,999,00000011
07:27:39	T#017,999,883,406,057,999,00000011
07:27:41	T#018,999,883,408,057,999,00000011
07:27:43	T#019,999,883,408,057,999,00000011
07:27:45	T#020,999,884,410,057,999,00000011

3.4 Received signal

The telemetry data retrieval at the near and far distance is conducted for testing the signal receptions in the receiver. This test aims to take the data in near and far distance whether the received signal is damaged or not. The telemetry data of the received signal output from the receiver at the near and far

distance is taken, which each has a form of a sine wave signal and the received frequency is an AFSK signal.

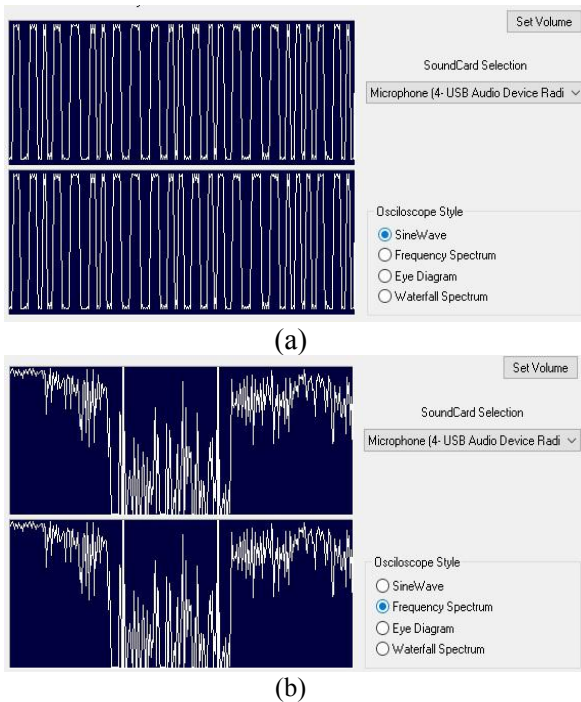


Fig. 3 Received signals in near distance testing, (a) in the sinusoidal waves form and (b) spectrum frequency

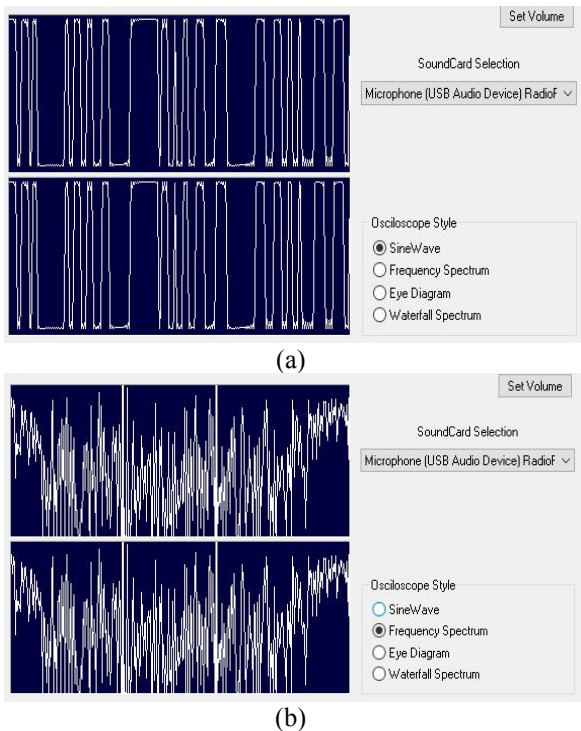


Fig. 4 Received signals in far distance testing, (a) in the sinusoidal waves form and (b) spectrum frequency

Figure 3 is the received telemetry results of the near distance testing in the format of:

```
14:59:47RNOCALL>APTT4 Port=2 <UI
Len=34>:
T#021,999,832,539,053,793,00000011
```

The tests show that the received data and signal in the receiver did not get damaged and run well. It is seen in sine wave signals that have stretched and received frequencies indicate the existence of information obtained on the side of the receiver in the form of tone sound produced.

Figure 4 is the received telemetry results of the far distance testing in the format of

```
15:00:16R NOCALL>APTT4 Port=2 <UI
Len=34>:
T#252,999,854,495,064,818,00000011
```

The test shows that the received signals in the receiver experience little difference from telemetry data reception at near distance. It is seen in sine wave signals that have a very long stretch and received different frequencies from the frequencies in near distance. In far distance testing, telemetry data can still be collected well although sometimes the time received when data retrieval is delayed. This is due to physical obstacles that make the range of HT limited. Also, the barriers to radio communication such as the influence of reflection, propagation and so forth are also a constraint in amateur radio communication. The sine wave signal that stretches far due to the impact of reflections caused by the delay occurs. The frequency generated should be noted as one of the essential considerations in amateur radio communications for it relates to information data transmitted.

4. Result and Discussion

Based on the testing results, the proposed nanosatellite load in this research, sensor accuracy rate test shown on Arduino compiler has a temperature and humidity accuracy rates of 92.97% and 90.57%, respectively. As for the APRS module, the sensor accuracy rate test shows temperature and humidity accuracy rates of 91,08% and 90.57%, respectively. The error value occurs because the temperature and humidity sensor is in the same load with APRS module and so that the heat from other components affect the sensors. In the next design needs to reconsider the component placement. The telemetry data reception test shows a delay of 1 second, which is acceptable for real-time data transmission. The signal reception test on receiver indicates that the emitted telemetry data is still well received on the receiver side for both near and far distance. There is a constraint in distant testing that causes a time delay in data retrieval. This condition might be happened due to a physical obstacle, an effect of reflection, propagation and other constraints in amateur radio communication.

Reference

- [1] NASA, "Small Spacecraft Technology State of the Art," no. February, pp. 1–197, 2014.
- [2] G. Maral and M. Bousquet, *Satellite communications systems: systems, techniques and technology*. Wiley, 2013.
- [3] H. Boiardt and C. Rodriguez, "The use of Iridium's satellite network for nanosatellite communications in Low Earth Orbit," in *2009 IEEE Aerospace conference*, 2009, pp. 1–5.
- [4] H. Boiardt and C. Rodriguez, "Low Earth Orbit nanosatellite communications using Iridium's network," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 25, no. 9, pp. 35–39, Sep. 2010.
- [5] O. Popescu, J. S. Harris, and D. C. Popescu, "Designing the communication sub-system for nanosatellite CubeSat missions: Operational and implementation perspectives," in *SoutheastCon 2016*, 2016, pp. 1–5.
- [6] J. Mevada, J. Samuel, S. Bhadane, A. K. Gulati, and R. D. Koilpillai, "Design and implementation of a robust downlink communication system for nanosatellites," in *2015 International Conference on Space Science and Communication (IconSpace)*, 2015, pp. 164–169.
- [7] J. L. Tresvig and T. Lindem, "Design of a Communication System for a Nanosatellite Space Weather Mission," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 29, no. 11, pp. 22–29, Nov. 2014.
- [8] M. Cello, M. Marchese, and F. Patrone, "ColdSel: A Selection Algorithm to Mitigate Congestion Situations over Nanosatellite Networks," in *2016 IEEE Global Communications Conference (GLOBECOM)*, 2016, pp. 1–6.
- [9] A. Goeritno, R. Yatim, and D. J. Nugroho, "Implementasi Automatic Packet Reporting System (APRS) Untuk Paket Data Pemantauan dan Pengukuran," in *The 3rd National Conference on Industrial Electrical and Electronics (NCIEE)*, 2013, pp. 27–34.
- [10] R. T. Prodoningrum, H. Wijanto, and A. D. Prasetyo, "Antenna deployment for Automatic Packet Reporting System of nanosatellite using Global Positioning System as a height sensor," *14th Int. Conf. QiR (Quality Res. QiR 2015 - conjunction with 4th Asian Symp. Mater. Process. ASMP 2015 Int. Conf. Sav. Energy Refrig. Air Cond. ICSERA 2015)*, pp. 8–11, 2016.
- [11] S. Chaiyasoonthorn, N. Hongyim, and S. Mitatha, "Building Automatic Packet Report System to report position and radiation data for autonomous robot in the disaster area," *ICCAS 2015 - 2015 15th Int. Conf. Control. Autom. Syst. Proc.*, no. Iccas, pp. 85–88, 2015.
- [12] APRS Working Group, *APRS Protocol Reference*, no. 29. 2000.
- [13] W. Wldada, "Pengembangan Modem AFSK Untuk Telemetri Muatan Roket UHF," *Lapan*.
- [14] R. A. Mahendra, "Sistem Monitoring Objek Memanfaatkan Teknologi GPS dan APRS." Universitas Telkom, 2012.
- [15] BYONICS, "BYONICS TinyTrak4 Quick-Start Guide," 2009.
- [16] W. Beech, D. Nielsen, and J. Taylor, "AX . 25 Link Access Protocol for Amateur Packet Radio," no. July, 1998.
- [17] S. M. Dilek, A. Ayranci, A. Seker, O. Ceylan, and H. B. Yagci, "AX.25 protocol compatible reconfigurable 2/4 FSK modulator design for nano/micro-satellites," in *2012 20th Telecommunications Forum (TELFOR)*, 2012, pp. 416–419.
- [18] J. J. Paternina-Anaya, J. E. Salamanca-Cespedes, and M. A. Avila-Angulo, "Communication system design and implementation for one pico-satellite and four earth stations using AX.25," in *2014 IEEE Central America and Panama Convention (CONCAPAN XXXIV)*, 2014, pp. 1–6.
- [19] A. SETIAWAN, "Perancangan Dan Implementasi Sistem Monitoring Jarak Jauh Berbasis Protokol AX.25 Dengan Menggunakan Mikrokontroler." Universitas Telkom, 2007.
- [20] A. Addaim, A. Kherras, and B. Zantou, "Design of Store and Forward Data Collection Low-cost Nanosatellite," *IEEE Aerosp. Conf. Proc.*, 2007.
- [21] L. Datasheet, "LM35 Precision centigrade temperature sensors," *Retrieved Sept. 13th*, no. November, pp. 1–13, 2017.
- [22] Octopus Brick, "Octopus Soil Moisture Sensor Brick [OBSoil-01]."



Raditiana Patmasari is currently a Lecturer in School of Electrical Engineering Telkom University, Bandung, Indonesia. She received the bachelor and master degree from Telecommunication Engineering, Telkom Institute of Technology, Bandung. Her research interest is information signal processing, especially in biomedical engineering, embedded system/ hardware system development, and information coding theory.



Inung Wijayanto received the B.S. and M.S. degree from Telecommunication Engineering, Telkom Institute of Technology, Bandung, Indonesia in 2008 and 2011 respectively. He is currently a Lecturer in School of Electrical Engineering Telkom University. His research interest includes, image processing

essing, audio processing, image analysis, biomedical engineering and embedded system development.



Yusuf Pradana Gautama received a bachelor's degree in telecommunication engineering from Telkom University.

He is passionate in IT and engineering with some research in Software Development, Nanosatellite Technology, On-board Computer and Internet of things. These days he is working as

a software engineer on backend side of big data company powered by AI technology.



Raditia Saputra Deanto received a bachelor's degree in telecommunication engineering from Telkom University. He is passionate in Nanosatellite Technology, embedded system/hardware system development



Hurianti Vidyningtyas received the B.S and M.S degree from Telecommunication Engineering, Telkom Institute of Technology, Bandung, Indonesia in 2008 and 2011 respectively. She is currently a Lecturer in School of Electrical Engineering, Telkom University.

Her research interest includes wireless communication, digital video broadcasting (DVB), embedded system development.