Indonesian Journal of Aerospace Vol. 22 No. 1 Juni 2024 : pp 45 – 56 (Rohmah et al.)
DOI: 10.59981/ijoa.2024.2553
https://ejournal.brin.go.id/ijoa/

LabVIEW based Ground Monitoring System for RX200

Nurul Fahrizatul Rohmah, Faisa Lailiyul Mutho'Affifah, Fikana Mahardika Cantri, Ardian Wardhana, Mirza Zulfikar Rahmat

Research Center for Rocket Technology
National Research and Innovation Agency (BRIN), Indonesia
e-mail: nuru035@brin.go.id

Received: 17-12-2023. Accepted: 03-12-2024. Published: 31-12-2024

Abstract

The implementation and deployment of a comprehensive ground monitoring system for rockets is essential to ensure the safety, efficiency, and success of space missions. Ground monitoring systems for rockets play a crucial role in providing real-time information and analysis of critical parameters during rocket operations, including ground testing and launch operations. The ground monitoring system encompasses various functionalities including the acquisition, processing, visualization, recording and storage of data. The data acquisition process in LabVIEW involves the use of TCP/IP Visa function to collect information. The processing involves various functions such as parsing, flight time calculation, and queue management for efficient data handling and calculations. The data can be recorded and stored using the TDMS function. The system efficiently managed all the received data. This study achieved 90,94% accurate data in processing the entire dataset and obtaining accurate values for calculating flight time. This combination yields precise and dependable data that can be utilized for making real-time decisions, analyzing performance, and evaluating missions after they have been completed.

Keywords: *real time monitoring; data acquisition; flight time calculation; queue management.*

Nomenclature

x	=	Value of Accelerometer X
i	=	Index
U	=	Upper limit
L	=	Lower limit

1. Introduction

Surveillance of unmanned aerial vehicles is vital in maintaining flight stability and regulating attitude control, particularly when dealing with high-speed rockets. Utilizing multiple sensory channels to present information offers a dual advantage of efficiently addressing high amounts of information and providing flexibility in presenting the information to operators under various environmental conditions (Maza et al., 2010). Ground-based monitoring systems are constructed with antennas and network architecture (Liu et al., 2020), as well as interface software tailored to the rocket's specifications. This customization ensures the captured sensor data is appropriately accommodated and accurately presented.

Traditionally, monitoring systems heavily relied on text-based terminals, which posed various limitations regarding user-friendliness and real-time visualization. However, advancements in technology have paved the way for more effective solutions addressing these challenges. The LabVIEW software was utilized to develop a graphical user interface for monitoring the experimental rocket. This user interface provides a comprehensive monitoring solution during rocket experiments. The graphical user interface integrates multiple components, such as charts, map tracking, data tables, a configuration section, and

e-ISSN 3032-0895 (online)

a three-dimensional rocket animation. These elements enable operators to comprehensively understand the vehicle's real-time status, location, and performance.

Various types of antennas are utilized to enhance coverage and ensure continuous communication with the rocket. These antennas are strategically placed at different locations to ensure continuous and efficient data transmission between the ground monitoring system and the rocket. The monitoring system's software operates on the computer to process all the data received from the antenna. The software utilizes the communication protocol to establish communication between the monitoring computer and the antenna receiver. The selection of a communication protocol is influenced by various conditions. Factors such as distance, data security, and speed play a crucial role in determining which communication protocol to use (Zou et al., 2021).

In developing a software monitoring system, it is crucial to incorporate networking functionalities for communication with antenna systems. Additionally, the system should include modules for data acquisition and processing, as well as logging data for future reference. The process of acquiring and processing data involves collecting, interpreting, and visualizing it in a format that is easily comprehensible and analyzable by the operator. It is important to consider the time process of each function while processing the data and ensure that real-time data is displayed. Multi-threading is one of the techniques that can be employed to achieve real-time data processing in the monitoring system. LabVIEW applies multi-thread programming to handle the concurrent execution of multiple tasks and ensure efficient processing of real-time data (Bitter, R., Mohiuddin, T., & Nawrocki, M., 2000).

2. Methodology

2.1. Payload Specification

The payload on the RX200 has 2 main sub-systems, i.e. the Power Management System (PMS) and the Onboard Flight Control System (OBFCS). PMS has a board system to manage the power requirements of the OBFCS. The PMS on this system has batteries with specifications of 4 cells with a capacity of 5000mAh. PMS has the function of supplying power, starting from power requirements for sensors and processing to data transmission. This PMS is also equipped with current and voltage sensors, which are processed by OBFCS to determine the condition of the batteries and PMS for the success of the payload system. Another payload sub-system is OBFCS, which consists of data acquiring, processing, and transmitting blocks. Data acquisition in this system is equipped with an Inertial Navigation System (INS) and various other sensors, such as a 40G accelerometer, pressure sensor, and temperature sensor, to determine the real-time behavior and conditions of the rocket. Data processing on this system uses a high-speed MyRIO-1950 series processor, which then sends all real-time data via radio data transmitting. The payload's block diagram is shown in Figure 2-1 below.

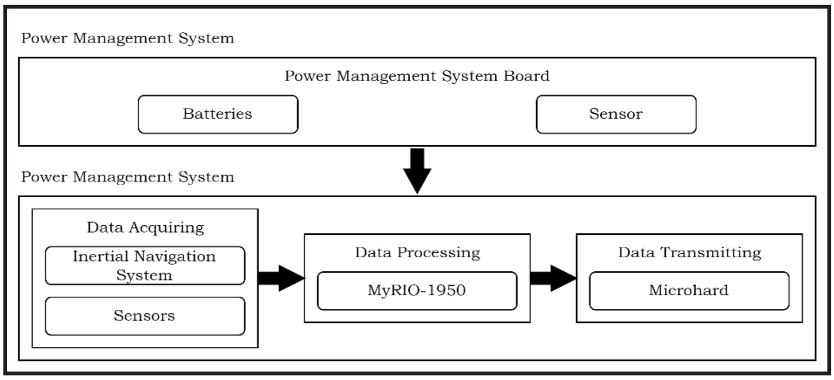

Figure 2-1. Payload block diagram.

2.2. Antenna Receiver Specifications

The infrastructure for the ground monitoring system has been carefully designed to meet the specific requirements and conditions of the operational site. The infrastructure diagram is shown in Figure 2-2. To enhance signal coverage and reception, multiple antenna receivers are strategically placed in different areas (Y. Huo et al., 2020), (Roger et al., 2022). The number of computers is adjusted accordingly to ensure efficient control and management. These receivers operate using a serial protocol, which is then converted to TCP/IP protocol through a control box. This conversion allows for seamless communication between the antennas and the computers that handle monitoring tasks. Reliable data transmission from the control box to the computers is facilitated by Power over Ethernet technology along with a switch network setup. With this customized antenna infrastructure in place, reliable signal reception, efficient data transfer, and effective component communication all contribute towards enhanced monitoring and control capabilities of aerial vehicles under supervision.

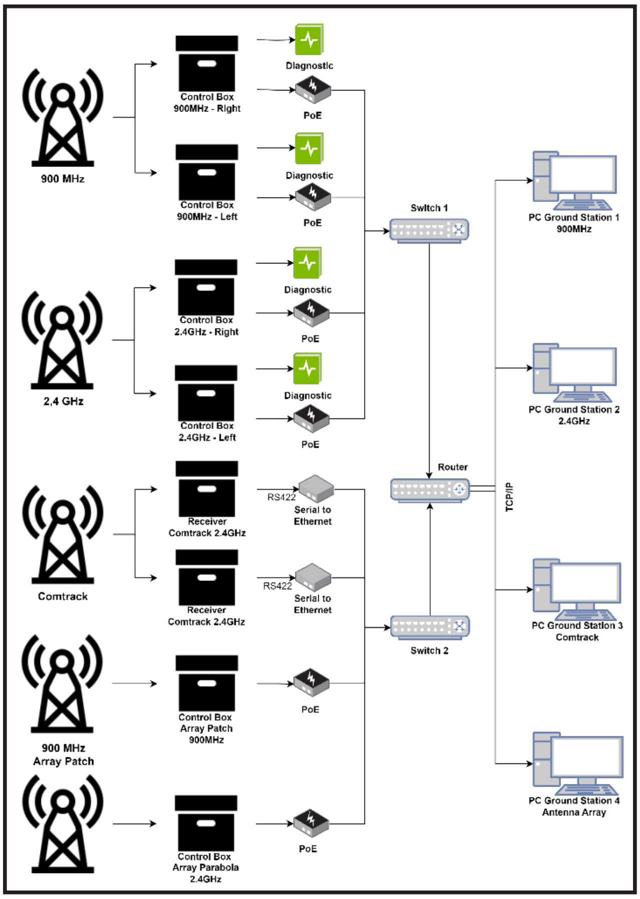

Figure 2-2. The infrastructure diagram of the ground monitoring system.

2.3. Architecture of Software Design

The ground control system was developed using LabVIEW, a visual programming environment. LabVIEW programming language is utilized for the user interface and data acquisition system. LabVIEW was chosen as the programming language because of its user-friendly and graphical-based nature, enabling the creation of an easily operable user interface (Pranowo, I. D. and Artanto, D., 2021). LabVIEW's extensive graphical programming capability enhances the flexibility and power of data logging (Allah et al., 2017). LabVIEW allows us to optimize performance on multithreaded operating systems and multiprocessor computers without adding complexity or extending development time (Fareeza et al., 2018). This software greatly enhances the efficiency of monitoring and controlling the rocket to maximize its effectiveness.

The system's structure consists of interconnected subroutines responsible for managing crucial functions, as shown in Figure 2-3. The data communication subroutine establishes a secure and reliable TCP/IP connection between the antenna. This connection ensures that important data can be transmitted accurately and efficiently during mission operations. The process of parsing data involves extracting and organizing the received information into packets that can be meaningfully processed. The subroutine for detecting the flight time uses of different algorithms, such as the average method, to analyze data from the accelerometer and identify when the rocket's flight begins. LabVIEW's graphical capabilities are utilized to visually represent attitude, trajectory, and other pertinent information through charts, graphs, and maps. These data visualization subroutines present the collected data in a visual format that is easily comprehensible. The data logging subroutine employs the fast TDMS file format to store processed data for subsequent analysis. The data queue subroutine implemented in this system ensures seamless data flow, effectively preventing any loss of information during the processing stage. These subroutines are implemented within the main VI, establishing an integrated interface system that facilitates effective monitoring and control of the rocket's mission.

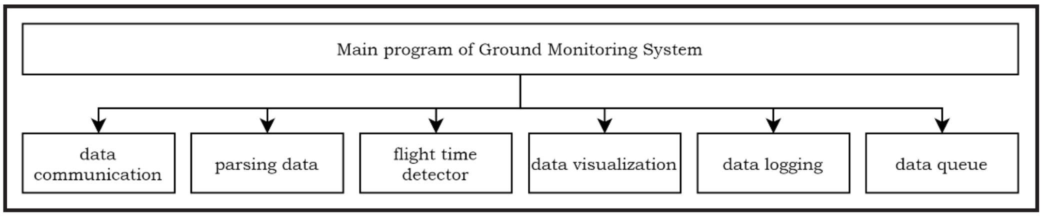

Figure 2-3. The system subroutine.

3. Main Program

3.1. Data Communication

The data transmission between the rocket and ground station involves the utilization of two separate frequencies: 2.4GHz and 900MHz. To facilitate the transmission of data, a Radio Microhard is utilized to relay information from the rocket. On the ground station, a variety of antennas are strategically positioned to maximize signal reception and effectively capture transmitted data. These include parabolic, patch, and COMTRACK antennas that have been specifically selected for their ability to optimize communication with the satellite. The antennas are crucial for establishing reliable and effective communication between the rocket and the ground station throughout its entire flight path.

This conversion becomes especially important when the distance between the processing PC and the antenna exceeds the 20-meter limit (Li et al., 2020). The TCP is designed to ensure reliable and resilient data transfer over long distances (Fisher et al., 2020), (Park, M & Chung, S., 2010). By adopting the TCP protocol, the communication system can circumvent the cable length limitation (Bayilmiş et al., 2022), ensuring a stable connection between the antenna and the processing PC. This enhancement improves the overall performance and dependability of the communication system by enabling efficient and uninterrupted transfer of data.

3.2. Parsing Data

The workflow in a rocket's payload involves gathering and sending data from various sensors, including the inertial navigation system, pressure sensor, current/voltage sensor, temperature sensor, and accelerometer. This data is used to determine the attitude of the rocket. The data is processed by the MyRIO-1950 and transmitted to the ground monitoring system using a hexadecimal data format. The payload data is divided into sub-packets for systematic transmission, with each sub-packet containing data collected from a specific sensor. At the ground system, the data is parsed thoroughly to extract and interpret the data from the sensors from each subpacket using specialized algorithms and techniques. This payload operation allows for precise monitoring and analysis of the rocket's orientation, facilitating real-time decision-making throughout its flight path.

Figure 3-1. Data protocol structure.

The hex characters of each transmitted packet data are read sequentially by the ground monitoring system. A type-cast function is utilized to convert hexadecimal data into specific data types. Before parsing, the data obtained from the TCP Read function goes through a filtering procedure. Figure 3-1 presents an illustration of the data protocol structure. To ensure proper alignment, synchronization is achieved by comparing the first character with the designated sync bytes “FC”. If synchronization is verified, the following character is compared to the “A5” as the start byte. When the sync byte and start byte align, the system assumes that the next subsequent byte represents pertinent information and proceeds to examine the third character, which signifies the message’s length. This systematic methodology allows for precise extraction and interpretation of the payload data, facilitating its subsequent analysis and processing within the monitoring system.

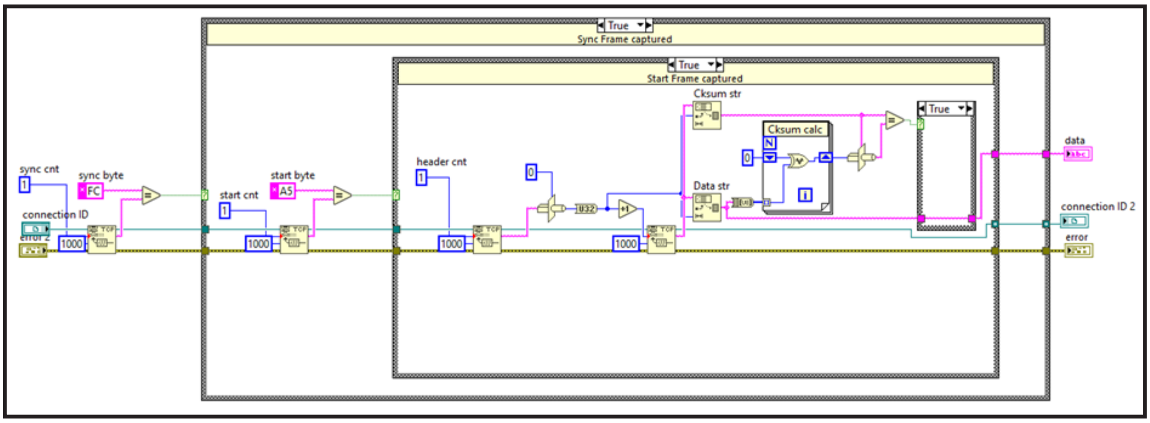

Figure 3-2. Block diagram of parsing data.

After determining the length of the message, the system changes its operation from processing individual characters to handling a specific amount of data that corresponds to the value of the message length. This method eliminates the necessity for reading each character individually and enables more efficient management of data. In this phase, the system parses and organizes the received data into subpackets according to a predetermined protocol. By following the prescribed protocol, the system guarantees consistent and precise arrangement of the payload information, enabling further analysis and understanding of the sensor data. Figure 3-2 illustrates the block diagram of the process of data parsing. The process of systematically organizing data into subpackets enhances both the efficiency and reliability of the monitoring procedure.

3.3. Flight Time

Throughout the ignition and launch of the rocket, an accelerometer sensor identifies and records the acceleration encountered by the spacecraft. The initiation of the rocket’s flight time commenced of the rocket can be identified by examining the mean acceleration measurements recorded by the accelerometer sensor. Observing the graphical representation of the acceleration data, specifically from the X-axis of the accelerometer, noticeable significant changes in acceleration can be observed during the ignition and launch phases. These distinct fluctuations in acceleration serve as important indicators for determining the exact duration of the rocket’s flight. By conducting a meticulous examination of the acceleration data, it is possible to precisely ascertain the exact moment when ignition takes place and subsequently launch the object (Tian et al., 2022). This analysis yields invaluable information concerning both the performance and trajectory of the rocket.

The rocket’s flight initiation time is determined using the average method shown in Figure 3-3. The system gathers a series of current accelerometer data points and computes their mean Eq. (3-1), which functions as the central value for subsequent computations. A threshold value is established to determine the duration of the flight. The upper and lower boundaries are calculated by adding Eq. (3-2) and subtracting Eq. (3-3) the threshold value from the average number, respectively. Identifying the start of the flight time relies heavily on three important values: the median number, upper boundary, and lower boundary.

$$\bar{x} = \frac{1}{15}\sum_{n}^{i} x_n \text{, with } n = i - 15 \quad (3\text{-}1)$$

$$U = \bar{x} + r \quad (3\text{-}2)$$

$$L = \bar{x} - r \quad (3\text{-}3)$$

The system continuously monitors the accelerometer data in order to compare it with predetermined upper and lower thresholds. If the X data exceeds the predefined threshold or deviates from the expected range, it suggests that there is an acceleration anomaly affecting the rocket's flight. Consequently, this triggers the activation of the flight time indicator. This method enables precise calculation of the initiation of the rocket's flight duration using average acceleration data and predetermined thresholds. Utilizing the average technique, in conjunction with upper and lower thresholds, offers a dependable and sturdy approach to identifying the commencement of the rocket's upward trajectory (Patoz et al., 2022). Furthermore, it aids in accurately determining the duration of flight.

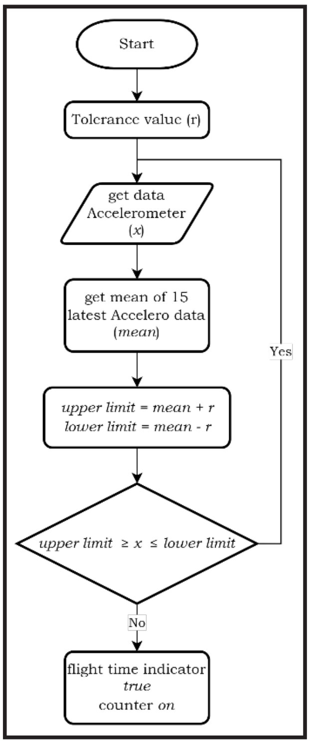

Figure 3-3. Flight time initiation flowchart.

Once the flight time indicator is activated, a counter starts counting. If there are no incoming characters detected by the TCP Read function, the counter will pause to allow for any potential interruptions or losses in data transmission. Suppose incoming characters are detected after a specified interval. In that case, the flight time system will determine the difference between the previously recorded stop time of the counter and the timestamp of the latest incoming characte The time interval between data inputs accurately represents the period of inactivity when the counter received no incoming data. By incorporating a time delay into the calculation of flight time, the system guarantees accurate tracking and measurement of the duration of the rocket's flight while considering any interruptions in data transmission. This approach ensures precise and continuous recording of the duration of the rocket's flight, even in situations where data transmission experiences temporary interruptions.

3.4. Data Visualization

After analyzing and categorizing the collected data, the system proceeds to present the information using a variety of visual representations such as graphs, maps, bar charts, and 3D animations. These visuals can be seen in Figure 3-4. These graphical representations are intricately crafted to improve the user's understanding of the rocket's orientation. Additionally, there is a designated section on the interface where users can access numerical data for those who prefer a more quantitative analysis. The rocket's orientation is visually represented through graphics and 3D animations, allowing for a thorough comprehension of its attitude. Moreover, velocity data is represented using graphical displays, allowing users to effectively analyze and interpret speed information. A graph showing the altitude of the rocket as it travels downrange is utilized to visually represent its trajectory. The integration of a central map view offers users an intuitive way to visualize the trajectory of the rocket. This comprehensive display, which combines both visual and numerical representations, enables users to derive meaningful insights into the behavior and performance of the rocket. As a result, users can develop a more profound comprehension of its flight characteristics.

Figure 3-4. User interface of ground monitoring system.

The depicted ground monitoring system in Figure 3-4 encompasses distinct sections meticulously devised to offer exhaustive monitoring and control functionalities, such as:

1. Time Area prominently displays the current time as well as the flight time, allowing operators to track the duration of the mission accurately. (1)
2. The attitude information is displayed in two formats: a dynamic chart that visually depicts the rocket's pitch, roll, and yaw parameters and a numeric display that provides precise numerical values for these parameters. (2)(3)
3. The Battery Bar Area provides a concise and visual representation of the remaining capacity of the payload battery, allowing for easy assessment of its power status. (4)
4. Operators can visualize the flight route and geographical position of the rocket using the Maps Area, aiding in trajectory monitoring. (5)
5. All data in the Data Section is presented in decimal format, which allows for easier analysis and interpretation of the parsed information. (6)
6. The Hex Data Area provides operators with a comprehensive display of the incoming data stream in hexadecimal format, allowing for detailed observation and analysis. (7)
7. The Raw Data Area presents the data in hexadecimal format, filtering and displaying only the information that includes valid sync bytes and start bytes. This ensures that the integrity and accuracy of the data are maintained. (8)
8. The 3D Animation Area offers an interactive and illustrative depiction of the rocket's spatial orientation, helping operators gain a better comprehension.
9. The Configuration Area provides operators with the ability to configure and personalize different parameters such as antenna connections, log file locations, launch pad coordinates, and map views. This feature allows for adaptability and flexibility in customizing the GCS interface based on specific operational needs. (10)

3.5. Data Logging

In order to retain raw data in its original hex format, it is common practice to store the data in a .txt file. This preservation technique guarantees data integrity and authenticity, enabling future reference or analysis while preserving the originality of the data. The processed data has been analyzed and interpreted by the established methods that are stored in the TDMS format for its efficient data processing capabilities [15] that meet the demands of the ground monitoring system with high volumes of data. The TDMS format, when utilized with Excel, provides efficient data reading and writing capabilities that are necessary for effectively managing a large amount of incoming data in real time (Czerwinski, F., & Oddershede, L. B., 2011). This approach offers the benefit of convenient access and analysis of the processed data. Importantly, using a single TDMS file enables efficient organization of multiple groups and channels, reducing the reliance on excessive log files. Optimizing data storage, retrieval, and analysis methods facilitates effective data management for comprehensive insights into rocket monitoring and performance.

3.6. Data Queue

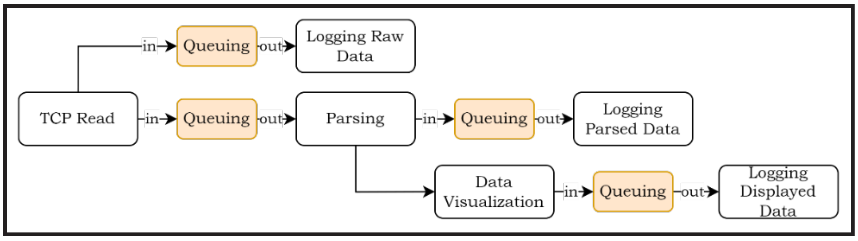

Figure 3-5. Data flow pipeline.

In the main program design, each subroutine functions autonomously while maintaining interconnectivity with other subroutines. As the processing time of each subroutine varies, there is a concern about potential data loss during program execution. To prevent this, a queue system is introduced at the transition points of every subroutine. This queuing system efficiently handles incoming data by collecting it in a First-In-First-Out manner. This ensures that the data is processed in the same order it was received once the preceding subroutine finishes its processing. A queue system is utilized to enhance the overall robustness and reliability of the ground monitoring system program (Sivaraman et al., 2016). The flow of data in the pipeline is illustrated in Figure 3-5. This ensures data integrity and prevents data loss even when multiple data inputs coincide with ongoing subroutine processes. As a result, seamless and uninterrupted operation is achieved, thereby enhancing the effectiveness of the system.

4. Result and Analysis

This section explains the results achieved from the testing of completed work in this project. The software operates on a computer that is equipped with an 8th Generation Intel Core i7 processor clocked at 3.7GHz, along with 16GB of RAM, a 128GB SSD storage drive, and an NVIDIA GeForce GTX 1060 6GB graphics card. The developed system underwent multiple laboratory tests and one field test during a rocket launch. The interface displayed in Figure 4-1 enables manual input of configuration details, including IP address, port, server address, and data location for saving log files. When the connect button is clicked, the program will commence receiving and processing data and automatically generate a plot of the data. For additional data not shown in this interface, users can access it in the log file.

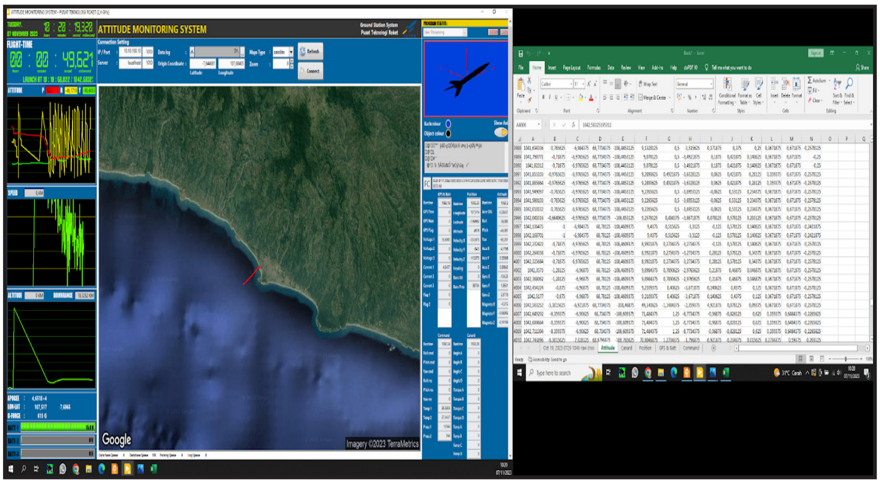

Figure 4-1. Interface of Ground Monitoring System during operation.

4.1. Receive and parsing data

During the activation of the payload and subsequent transmission of data to the ground station, the system effectively received and processed the data in real time. The received data in hexadecimal format will be decoded by the system to extract information according to a pre-defined format. Throughout the experiment, the system received a cumulative count of 31.802 lines of hexadecimal data within a time span ranging from 852.794 to 1113.69 seconds on the processor clock or approximately 260.895 seconds. Extract a total of 28.922 parsed data lines from this dataset, accounting for approximately 90.94397% of the entire dataset. The residual data comprises incomplete bytes that have a specific structure according to the data protocol. This prevents it from successfully passing through the filter. The reason for an incomplete byte is the loss of data during transmission. As demonstrated in Figure 4-2, a sample of clearly hexadecimal data along with their corresponding data protocol structure.

	Sync byte	Start byte	Messages length	Component ID	Data	Checksum
GPS & Battery	FC	A5	1C	01	4090 20DE F420 2020 2020 2020 203F 7320 2020 2016 3720 2020 2020 20	B6
Position	FC	A5	1D	02	4090 2063 1A20 2020 35D7 92FC 2D79 2011 2020 2020 2020 2020 2020 C5DF	58
Attitude	FC	A5	23	03	4090 214A F9A0 2020 FF9E FC75 2258 C9BD 0490 2040 FE40 200C 2019 2005 202F 2056 FFDF	24
Command	FC	A5	17	04	4090 204D 48C0 2020 2020 2020 2020 06A9 0671 1C81 01FB	8E
Canard	FC	A5	15	05	4090 2033 0940 2020 2020 2020 2020 2020 2020 2020	8F

(a)

GPS & Battery

Runtime	GPS Time	GPS Number	GPS Flag	Battery V1	Battery V2	Battery V3	Battery Curr 1	Battery Curr 2	Battery Curr 3	Batt Flag 1	Batt Flag 2
1032,22	2105376	32	32	15,8623	8,03125	8,03125	5,55371	8,03125	8,03125	32	32

Position

Runtime	Long	Lat	Alt	VX	VY	VZ	Head	BAlt	BPres
1032,1	107,684	-7,64474	16418	1028	1028	1028	64,25	16448	101310

Attitude

Runtime	Acc 50G	Roll	Pitch	Yaw	GX	GY	GZ	AX	AY	AZ	MX	MY	MZ
1032,32	-0,765625	-7,08594	68,6875	-108,523	9,125	64,5	-3,5	128,187	128,391	128,078	64,3672	64,6719	-0,257812

Command

Runtime	Roll cmd	Pitch cmd	Yaw cmd	Roll res	Pitch res	Yaw res	Temp 1	Temp 2	Press 1	Press 2
1032,08	0	0	0	0	0	0	26,6406	25,7656	14594	1014

Canard

Runtime	C Angle A	C Angle B	C Angle C	C Angle D	C Torque A	C Torque B	C Torque C	C Torque D	C Current A	C Current B	C Current C	C Current D
1032,05	0	0	0	0	0	0	0	0	0	0	0	0

(b)

Figure 4-2. Data sample in hexadecimal format (a) and parsed result (b).

4.2. Flight Time

Table 4-1. Data of Accelerometer X and Flight Time ignition.

Processor Time (Runtime)	Accelerometer X (x)	Mean ($\overline{x}$)	Upper Limit (U)	Lower Limit (L)	Flight Time Status ($U \geq x \geq L$)
second	g	g	g	g	
1042,04102	9,25781	8,54948	12,54948	4,54948	OFF
1042,13647	9,43750	8,55938	12,55938	4,55938	OFF
1042,16870	9,43750	8,58125	12,58125	4,58125	OFF
1042,23242	8,99219	8,60313	12,60313	4,60313	OFF
1042,26404	8,99219	8,59427	12,59427	4,59427	OFF
1042,32568	8,99219	8,58542	12,58542	4,58542	OFF
1042,35730	9,89844	8,57708	12,57708	4,57708	OFF
1042,38806	9,89844	8,62813	12,62813	4,62813	OFF
1042,45422	9,21094	8,68281	12,68281	4,68281	OFF
1042,51770	9,21094	8,69167	12,69167	4,69167	OFF
1042,58325	**49,14063**	**8,68646**	**12,68646**	4,68646	ON
1042,64929	71,48438	11,34323	15,34323	7,34323	ON
1042,68066	71,48438	15,49375	19,49375	11,49375	ON

The X value from the accelerometer is utilized to determine the commencement of the rocket launch duration. Table 4-1 displays the evolution of the Accelerometer X value over time and highlights a significant peak in its variation at a particular moment. Showcased in the table is the rise in value of accelerometer X from 9.21 g to 49.142 g, observed during a runtime of 1042.5832 seconds. The runtime value is also presented on the interface of the Ground Monitoring System, as illustrated in Figure 4-3. The flight ignition was activated when the accelerometer X value surpassed 49,142, which had previously been determined to be beyond the acceptable range. The experiment conducted in the laboratory yielded an r value of 4 for this process. The research on determining the optimal value of r for ignition flight time using accelerometer x is unavailable. It depends on the sensitivity of the sensor, where a lower r value defines a more sensitive ignition system based on this calculation method.

Figure 4-3. Display of flight timer on Ground Monitoring System interface.

4.3. Data Logging and Queue

It is advisable to utilize a queue system in this program to prevent data loss caused by variations in the time processing of each loop. In a different experiment using the same data, a program without the queue system encountered a notable data loss. When the program with a queue system is used, 31,802 lines of raw data are saved. In contrast, the program without a queue system only save 26,768 data lines. The condition affects the next loop process, which only clears data 18.056 times or 67,45%. Every piece of data in this experiment holds significance for subsequent analysis. Therefore, incorporating a queue system into this program is an excellent measure to minimize further data loss.

5. Conclusions

A ground monitoring system is utilized to efficiently monitor and analyze a rocket's orientation during payload experiments. Prior to utilizing this system, an emulator program is used to save received data and manually decode and parse it. It offers real-time data visualization and effective data storage to enable swift issue detection and troubleshooting, as well as identification of potential enhancements in the rocket's performance. The adopting of a data queuing system also enhances the effectiveness of processing and retaining important data.

In future work, a software monitoring system with the ability to process data from various sources within a unified interface will be developed. The software monitoring system can process data from multiple sources within a single interface with customizable data protocols. It provides more meaningful and engaging data visualization to support efficient monitoring and analysis. This also simplifies the devices used, thereby minimizing installation and setup time.

Acknowledgements

This research is supported by the Research Center for Rocket Technology by the National Research and Innovation Agency (BRIN).

Contributorship Statement

NFR developed the program, analyzed the result and prepared the manuscript; FLM developed the payload and prepared the manuscript; FMC developed the program; AW developed the program; MZR developed the program.

References

Allah, O. A., Abdalla, M., Abdalla, S., Babiker, A., & Allah, A. A. (2017). Universal data logger system for environmental monitoring applications. Indonesian Journal of Electrical Engineering and Informatics, 5(2), 131–136. https://doi.org/10.11591/ijeei.v5i2.289

Bayılmış, C., Ebleme, M. A., Çavuşoğlu, Ü., Küçük, K., & Sevin, A. (2022). A Survey on Communication Protocols and Performance Evaluations for Internet of Things. Digital Communications and Networks, 8(6), 1094–1104.

Bitter, R., Mohiuddin, T., & Nawrocki, M. (2000). LabVIEW: Advanced Programming Techniques. LabVIEW: Advanced Programming Techniques (pp. 1–441). CRC Press.

Czerwinski, F., & Oddershede, L. B. (2011). TimeSeriesStreaming.vi: LabVIEW Program for Reliable Data Streaming of Large Analog Time Series. Computer Physics Communications, 182(2), 485–489.

Fareeza, F., Rambabu, C., Krishnaveni, S., & Kabiso, A. C. (2018). Automation of DMPS manufacturing by Using LabView and PLC. International Journal of Electrical and Computer Engineering, 8(6), 5484–5494. https://doi.org/10.11591/ijece.v8i6.pp5484-5494

Fisher, E. M. D., Benoy, T., Humphries, G., Wilson, D., Lengden, M., Johnstone, W., Ozanyan, K. (2020). A Custom, High-Channel Count Data Acquisition System for Chemical Species Tomography of Aero-Jet Engine Exhaust Plumes. IEEE Transactions on Instrumentation and Measurement, 69(2), 549–558. https://doi.org/10.1109/TIM.2019.2895932

Li, R., Lu, J., Liu, P., Min, H., & Liu, J. (2020). Design of The VLRA Battery Real-Time Monitoring System Based on Wireless Communication. Sensors, 20(15), 4350. https://doi.org/10.3390/s20154350

Lin, J., & Lin, J. (2022). Design of High Speed Relational Database Based on Labview TDMS Format. In 2022 7th International Conference on Intelligent Computing and Signal Processing, ICSP 2022 (pp. 1421–1425). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ICSP54964.2022.9778835

Liu, Y., Chen, Y., Jiao, Y., Ma, H., & Wu, T. (2020). A Shared Satellite Ground Station Using User-Oriented Virtualization Technology. IEEE Access, 8, 63923–63934.

Maza, I., Caballero, F., Molina, R., Peña, N., & Ollero, A. (2010). Multimodal Interface Technologies for UAV Ground Control Stations: A Comparative Analysis. Journal of Intelligent and Robotic Systems: Theory and Applications, 57(1–4), 371–391.

Park, M., & Chung, S. (2010, January 1). Detecting TCP Retransmission Timeouts Non-related to Congestion in Multi-Hop Wireless Networks.

Patoz, A., Lussiana, T., Breine, B., Gindre, C., & Malatesta, D. (2022). A Single Sacral-Mounted Inertial Measurement Unit to Estimate Peak Vertical Ground Reaction Force, Contact Time, and Flight Time in Running. Sensors, 22(3).

Pranowo, I. D. and Artanto, D. (2021). Improved control and monitor two different plc using labview and ni-opc server. International Journal of Electrical and Computer Engineering (IJECE), 11(4), 3003. https://doi.org/10.11591/ijece.v11i4.pp3003-3012

Roger, S., Martin-Sacristan, D., Garcia-Roger, D., Monserrat, J. F., Kousaridas, A., Spapis, P., & Ayaz, S. (2022). 5G V2V Communication with Antenna Selection Based on Context Awareness: Signaling and Performance Study. IEEE Transactions on Intelligent Transportation Systems, 23(2), 1044–1057. https://doi.org/10.1109/TITS.2020.3019530

Sivaraman, A., Cheung, A., Budiu, M., Kim, C., Alizadeh, M., Balakrishnan, H., Licking, S. (2016). Packet Transactions: High-Level Programming for Line-Rate Switches. In SIGCOMM 2016 - Proceedings of the 2016 ACM Conference on Special Interest Group on Data Communication (pp. 15–28). Association for Computing Machinery, Inc.

Tian, Z., Yang, K., Danino, M., Bar-Shalom, Y., & Milgrom, B. (2022). Launch Point Estimation with A Single Passive Sensor without Trajectory State Estimation. IEEE Transactions on Aerospace and Electronic Systems, 58(1), 318–327.

Y. Huo, L. Zhao, Q. Hu, E. Ding, Z. Xin, & Z. Sun, “Optimal Deployment of Antenna for Field Coverage in Coal Mine Tunnels”, IEEE Access, vol. 8, p. 51954-51963, 2020. https://doi.org/10.1109/access.2020.2979504

Zou, L., Wang, Z., Hu, J., Liu, Y., & Liu, X. (2021). Communication-Protocol-based Analysis and Synthesis of Networked Systems: Progress, Prospects And Challenges. International Journal of Systems Science, 52(14), 3013–303