Journal of Electrical Technology UMY (JET-UMY), Vol. 2, No. 3, September 2018
ISSN 2550-1186 e-ISSN 2580-6823

Development of Data Acquisition System Using LabVIEW-based Interface for Aircraft Application

Faaris Mujaahid[*1], Fuad Hammaminata Arya Anjasmara[1], Rama Okta Wiyagi[1], Karisma Trinanda Putra[2]

[1]Department of Electrical Engineering, Faculty of Engineering, Universitas Muhammadiyah Yogyakarta, Indonesia
Bantul 55183 Daerah Istimewa Yogyakarta, Indonesia

[2]College of Information and Electrical Engineering, Asia University
Liufeng Road 500, Wufeng District, Taichung City, Taiwan 41354
*Corresponding author, e-mail: f.mujaahid@umy.ac.id

Abstract – *This paper discusses an interface data acquisition (DAQ) of a rocket using LabVIEW. The software development is meant to improve the quality and reliability of the interface that has been built by the UMY rocket team. The data used on the experiments is a set of dummy data created for simulation purpose. This data is generated by Arduino UNO, in which it was acquired by LabVIEW through VISA serial communication platform. The test results show that there is 10% error during the data retrieval. The DAQ average speed is 270 milliseconds. This new approach has more stable and reliable system when compared to previous algorithm.*

Keywords: *DAQ, LabVIEW, Arduino, Rocket*

I. Introduction

Along with the development of technology, aeronautics and astronautics become one of the parameters to measure the level of technology and science of a country, including Indonesia. The Indonesian government believes that the research development on these fields will have a significant impact on the scientific and economic sectors. Therefore, the government, through LAPAN (National Aeronautics and Space Institute) and Kemenristekdikti (Ministry of Research, Technology, and Higher Education of Indonesia) holds an annual competition called KOMURINDO (Indonesian Rocket Contest Competition) with the aim to spur rocket research in higher education, such as universities and polytechnics. Since 2008, this competition has been held ten times and every year the participants are increasingly competitive [1][8]. Universitas Muhammadiyah Yogyakarta rocket team participated in Komurindo 2018 with improvements in the communication sector. This is due to the technology used in the previous competitions had a stability problem in the interface system to display the rocket sensor readings [2].

In a research article published in 2012, LabVIEW is used in control system engineering education [3]. The data acquisition is used to provide better performance of model predictive control (MPC). The three methodologies to differentiate MPC for a simple model, simple model with time delay, and MPC versus PID controller are using LabVIEW to simulate the results and explain the performance of the MPC [3].

This paper contains an important development for the system communication interface and involves the signal acquisition from the sensors part to the ground control system (LabVIEW-PC). Therefore, the scope of this paper is data acquisition conducted using the LabVIEW interface with dummy data as a substitute for the original data. Unlike a text-based programming language interface or command-line interface (CLI), this research uses LabVIEW which is based on graphical-based programming. The reminder of this paper is organized as follows; the method is described in section 2, experiments and results are presented in section 3, and section 4 concludes the paper.

Manuscript received July 2018, revised August 2018

II. Tools and Methods

In this section, we focus on the design methodology of interface that is used for data acquisition. User interface is a visual part of a system, software application or hardware device that ensures how a user interacts with the application and how information is displayed on the screen. The user interface combines the concepts of visual design, design interaction, and information infrastructure. This data acquisition process uses dummy signal generated by Arduino board, thus no sensor signals from rocket is used. Therefore, only few supporting tools are used, including:

- Toshiba Satellite C50-B Laptop, Intel Core i3-3217U 1.8 GHz RAM 4 GB
- Operating System: Windows 8.1 SP 1
- Arduino UNO Module
- Software: Arduino IDE 1.8.5 64-bit
- Software: LabVIEW 2017 32-bit

There are two processes in the design, i.e. design of dummy data and design of interface (LabVIEW). Dummy data is used as a raw data to be processed by the interface. The interface will carry out data acquisition through data communication link with the Arduino UNO module.

A. Preparation

This stage is intended to prepare a tool that will be used to design dummy data and the LabVIEW interface. Dummy data can be designed by making sure the Arduino IDE program is installed properly on a laptop and can be connected to the Arduino UNO module through the available ports. Then, the new interface can be designed when LabVIEW has been installed and can be connected to Arduino UNO through data communication called VISA (Virtual Instrument Software Architecture). With this condition, then the system design can be run.

B. Dummy Data

Dummy data can be generated using programming in Arduino IDE. The output of the programming becomes dummy data, where the data generated is random and considered to be similar to real data in the field so that it can be used as an ideal replacement data. The format of the dummy data is adjusted to the needs of the rocket payload. The following is the data format and its explanation.

```
t%fp%fa%faccx%faccy%faccz%flo%fla
%fcx%fcy%fcz%fgx%fgy%fgz%fhd%f
```

Fig. 1. Format of the dummy data

Before explaining the design stages, the following is the design scheme of the dummy data.

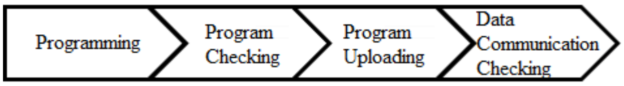

Fig. 2. Dummy data design scheme. The process starts from the left and ends on the right

TABLE I
DESCRIPTION OF THE DUMMY DATA FORMAT

Data Format	Function
t	Prefix for temperature data
p	Prefix for pressure data
a	Prefix for height/altitude data
accx	Prefix for accelerometer data (x axis)
accy	Prefix for accelerometer data (y axis)
accz	Prefix for accelerometer data (z axis)
lo	Prefix for latitude data
la	Prefix for longitude data
cx	Prefix for compass data (x axis)
cy	Prefix for compass data (y axis)
cz	Prefix for compass data (z axis)
gx	Prefix for gyroscope data (x axis)
gy	Prefix for gyroscope data (y axis)
gz	Prefix for gyroscope data (z axis)
hd	Prefix for head angle data
%f	Float numbers type

First, programming is carried out using *randomSeed ()* function to get results in the form of random numbers with integer types. The function becomes the initial setup function located in *setup ()*, which later on *loop ()* will be derived using the *random ()* function.

Second, checking the program is done by clicking on the icon in the Arduino IDE called *Verify*. Basically, this process is used to check for syntax errors in the program, not the logic because the machine only follows the program writing procedure. If no error is found, then proceed to the next stage.

After the program checking phase is complete, the third step is to upload the program to the Arduino UNO module. This is done by clicking the *Upload* icon. If successful, then proceed to the final stage of design.

The data communication checking stage is done by opening the *Serial Monitor* to see whether the data has been transferred by the hardware or not. This is verified by the data displayed on the *Serial Monitor*. The purpose is to ensure that Arduino data communication works accordingly. After the data is observable in the serial monitor, then the design of interface can be started. This interface will parse the serial data in accordance with pre-determined variables.

C. Design of Interface using LabVIEW

The software tool used in this work is LabVIEW 2017 32-bit (Version 17.0). This program is a graphical-based programming software produced by National Instruments. LabVIEW program is known as VI or *Virtual Instruments* because its appearance and operation can mimic an instrument. In LabVIEW, there are two parts, i.e. Front Panel and Block Diagram. The interface is the Front Panel, while the source code is worked out in the backend layer, namely Block Diagram. Figure 3 shows the design of the Front Panel from LabVIEW. From this window design, there are six smaller sections available to separate difference functions. Explanations of each section are in Table 2.

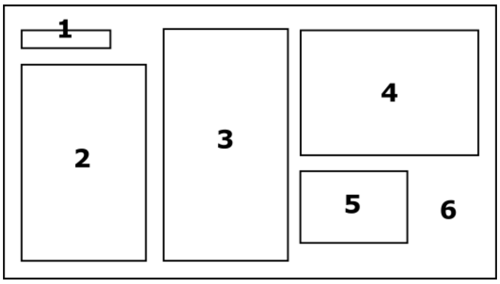

Fig. 3. Design of LabVIEW interface

TABLE II
DESCRIPTION FOR EVERY FIELDS SHOWN IN FIGURE 3

No.	Function
1	Determine the serial communication port
2	Output indicator after processing
3	Initial data indicators that are still in one package
4	Indicator data after fragmentation and processing time
5	Output error indicator
6	Front panel window

LabVIEW is used as a tool for data acquisition. This process requires a feature called VISA (Virtual Instrument Software Architecture). VISA or often also called NI-VISA can use many types of data communications such as GPIB, USB, Serial, and Ethernet [4]. In this project, VISA is used for Serial communication.

Figure 4 shows a flowchart that was designed using LabVIEW to run the data acquisition mechanism. This process also involves *Global Variables*, where it acts as a means of virtual access and filter data through several VIs running simultaneously. This feature is built-in within LabVIEW and only has block diagrams to design with.

The algorithm flow of DAQ in LabVIEW is as shown in figure 4 which includes the data reading, scanning and filtering of unwanted data (*ovf, nan*), data parsing, filtering out the error messages, and data display on Front Panel.

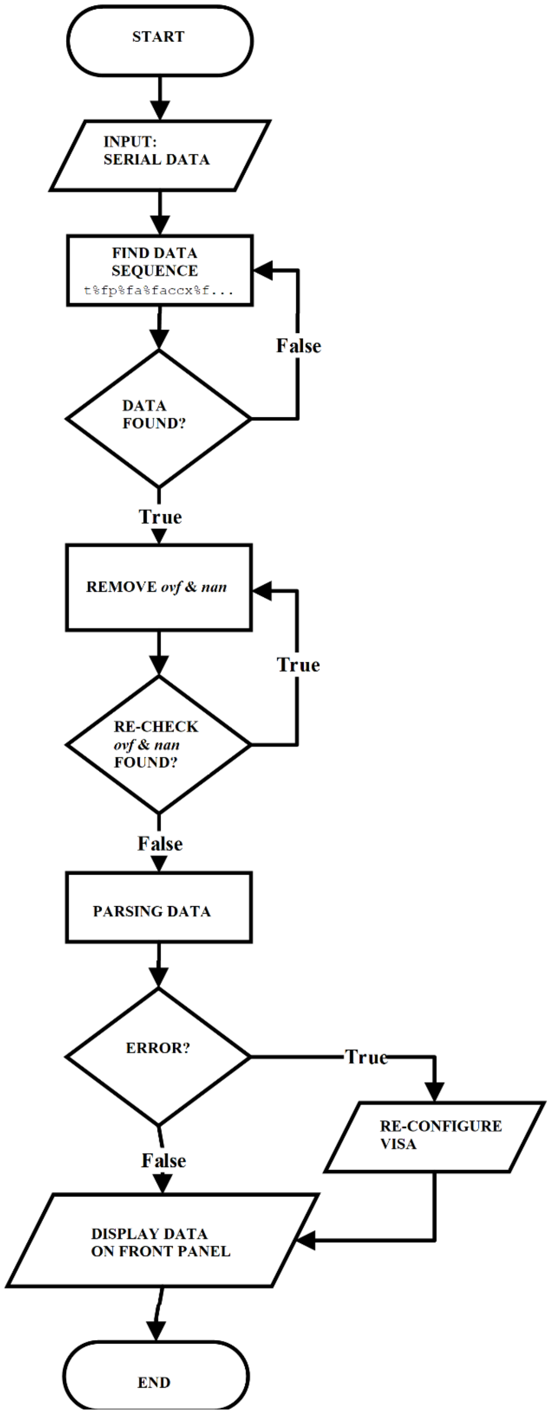

Fig. 4. Flowchart of DAQ algorithm

III. Experiments and Results

The simulation is carried out using two approaches. The first approach is to check the quality of data communication, while the second is to see the reliability of data communication.

A. System Quality

Before entering LabVIEW interface testing, it is necessary to check the data through Serial Monitor. This check is carried out to ensure that the data communication from Arduino UNO is still in good quality. The following are the results of the dummy data displayed on the *Serial Monitor*. From the results of serial data monitoring, it shows that the data communication on Arduino UNO is going well, proven by the presence of dummy data on the display. Dummy data is displayed continuously, with no separator in the form of Enter.

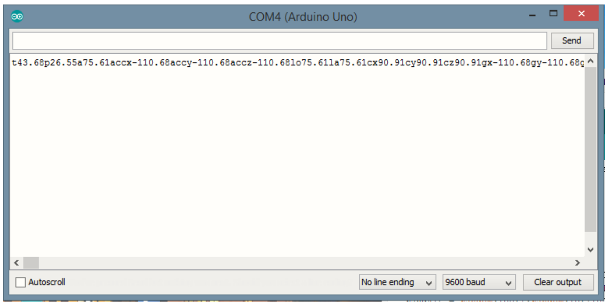

Fig. 5. Result of dummy data in Arduino *Serial Monitor*

After the dummy data is obtained, communication between Arduino UNO and LabVIEW can be done. This is done using the VISA feature, as stated earlier. The following shows the Front Panel interface after running.

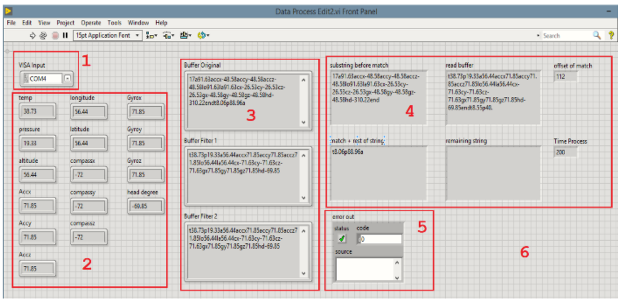

Fig. 6. Display of Front Panel

Figure 6 shows the data processing through indicators in the form of parameters that are processed, starting from raw data to real data. Raw data is in the form of dummy data that has been partially retrieved from serial communication using the VISA feature. As stated in the interface design, the function of LabVIEW above works according to what was planned earlier in the Methodology section. Chronologically, the process order based on Figure 3 and 6, we get the sequence of 3-4-5-2.

B. System Reliability

After checking the work quality of the system, the outputs of the system need to be considered. LabVIEW is run 50 times to see the system's efficiency in data acquisition. The following Table 3 shows the results of data retrieval.

TABLE III
EXPERIMENT OF DATA PARSING MECHANISM

Experiment No.	**Processing Time (milisecond)**	**Error**	**Position of Error (data seq. n-th)**
1	400	No	-
2	300	No	-
3	100	Yes	10
4	200	No	-
5	300	No	-
6	300	No	-
7	200	No	-
8	200	No	-
9	800	No	-
10	300	No	-
11	200	No	-
12	200	No	-
13	500	No	-
14	200	No	-
15	100	No	-
16	100	Yes	4
17	700	No	-
18	200	No	-
19	200	No	-
20	200	No	-
21	200	No	-
22	800	No	-
23	200	No	-
24	600	No	-
25	200	No	-
26	300	No	-
27	200	No	-
28	200	No	-
29	100	Yes	12
30	200	No	-
31	100	No	-
32	200	No	-
33	500	No	-
34	600	No	-
35	200	No	-
36	100	Yes	3
37	300	No	-
38	200	No	-
39	500	No	-
40	200	No	-
41	100	Yes	10
42	200	No	-
43	200	No	-
44	200	No	-
45	100	No	-
46	200	No	-
47	500	No	-
48	200	No	-
49	200	No	-
50	200	No	-

The grey-marked data is the data with error occurred in the n-th data format (refer to table 1). For example, position of error 10 means that the error occurs on data format *cy* or compass data y-axis. In 50 data retrievals, five errors were found. Hence, using the following equation, error possibility can be retrieved.

$$error = \frac{n\ error}{n\ datum} = \frac{5}{50} = \frac{1}{10} = 10\%$$

From this system, an error of 10% is obtained and this is still acceptable or tolerated. This error is caused by data that failed to split due to incomplete format according to the format above. This is indicated in the Error Position column, where there is a possibility of data loss between the first variable up to the fifteen, due to the delay of reading or due to serial communication that is sometimes interrupted. Then, the system takes less than 1 second (1000 milliseconds), which is between 100 - 900 milliseconds. The following is an average calculation.

$$t_{average} = \frac{proccess\ time}{n\ datum} = \frac{13500\ ms}{50} = 270\ ms$$

The system works with an average speed of 270 milliseconds and this is fast enough for the reading of a payload rocket instrument in competition.

IV. Conclusion

From the experimental results, it may be concluded that the design of DAQ interface by using dummy data in LabVIEW accordingly works well. Data generated from the Arduino UNO could act as the original data. The process includes the VISA serial communication, data reading, scanning, filtering, and VISA reconfiguration. For future works, error handling needs to be optimized to get less error and more accurate results.

References

[1] Mujaahid, F., Hizbullah, A. M., Syahfitra, F. D., Dahlan, M. A., Juliansyah, N. D. (2017). Development of User Interface Based on LabVIEW for Unmanned Aircraft Application. Journal of Electrical Technology UMY (JET-UMY), Vol. 1, No. 2, 106.

[2] Balaji, V. (2012). Study of Model Predictive Control Using NI LabVIEW. International Journal of Advanced Research in Engineering and Technology, Vol. 3, Issue 2, pp. 257-266.

[3] National Instruments (2018). NI-VISA Overview. Retrieved January 6, 2018

[4] National Instruments (2018). Global Variable. Retrieved February 1, 2018

[5] Wiyagi, R. O., Danardono, Agus, T. A. (2017). High Altitude Balloon Payload Design for Atmospheric Observations. *Journal of Electrical Technology UMY (JET-UMY), 1*(1). 50-57.

Authors' information

Faaris Mujaahid Hold a bachelor degree in 2010 from Electrical and Electronics Engineering Department, Saxion University of Applied Sciences, the Netherlands. He received a master degree in Sustainable Energy Technologies in 2016 from University of Southampton, UK. ing. Faaris Mujaahid, M.Sc is currently a lecturer in the Department of Electrical Engineering, Faculty of Engineering, Universitas Muhammadiyah Yogyakarta, Indonesia. His main research interest is in LabVIEW and renewable energy (mainly in solar cell material and fabrication technologies).

Fuad Hammaminata Arya Anjasmara is a bachelor student of Electrical Engineering, Faculty of Engineering, Universitas Muhammadiyah Yogyakarta, expected to finish his study in 2019.

Rama Okta Wiyagi Received B.Sc degree from Department of Electrical Engineering Universitas Muhammadiyah Yogyakarta in 2009, M.Eng. degree from Department of Electrical Engineering and Informatics Technology, Universitas Gadjah Mada, Yogyakarta, Indonesia in 2014. Rama Okta Wiyagi, M.Eng. is a Lecturer in Department of Electrical Engineering, Faculty of Engineering, Universitas Muhammadiyah Yogyakarta, Indonesia. His research interests are in robotics, robotics vision and instrumentation.

Karisma T. Putra born in Bondowoso on June 19, 1990. Graduated from elementary to senior high school in Bondowoso until 2008. Studied bachelor degree program in Surabaya, precisely in Electronics Engineering Polytechnic Institute of Surabaya (EEPIS) until 2012. He got scholarship program to continue master degree in Institut Teknologi Sepuluh Nopember (ITS) Surabaya. Karisma is also a lecturer at Electrical Engineering, Faculty of Engineering, Universitas Muhammadiyah Yogyakarta. The main focus of research is the intelligent systems and controls. He engaged in joint research related to the development of food commodity tracking systems and integrated intelligent systems. He was involved in several competitions in developing smart devices. Pursue the field of electronics and software development since college. Mr. Putra joined in Indonesian's engineer union organization (PII) in 2016. Mr. Putra is also active in writing publications on IEEE society.