IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)
e-ISSN: 2278-1676, p-ISSN: 2320-3331, Volume 16, Issue 5 Ser. II (Sep. – Oct. 2021), PP 07-14
www.iosrjournals.org

Development of A Data Acquisition Subroutine Embedded In Control Program

Ngo Phuong Le, Luong Ngoc Giap,
Nguyen Binh Khanh, Bui Tien Trung, Truong Nguyen Tuong An
Institute of Energy Science, Vietnam Academy of Science and Technology
A9 Building, 18, Hoang Quoc Viet str., Cau Giay dist., Hanoi, Vietnam
Corresponding Author: NGO Phuong Le

Abstract : *This paper presents a solution to embed a data acquisition subroutine in an embedded program of microcontroller in a control board. The solution considers the transmission speed over UART protocol, data frame size, and designs a data packet that fits 4 channels of 100-μs data and 20 channels of 2-ms data. A program also was built on PC to display the data acquired from the microcontroller as graphic charts, data table. The PC program can also send command to the microcontroller to start, stop and set the parameters of the control board. With the capability to view and set parameters of the control board, different experimental scenarios can be created and executed. Implementation was carried out on a maximum power point tracking (MPPT) Control Board.*

KEYWORDS – *UART, embedded program, microcontroller, MPPT, data acquisition, control loop, STM32*

Date of Submission: 08-10-2021 Date of Acceptance: 22-10-2021

I. INTRODUCTION

When testing an electronic control system, there is always a need to collect real-time operational data to analyze system performance. Using an oscilloscope to analyze voltage and current waveforms is one of the most common ways to analyze the operation of control circuits. Digital oscilloscopes allow to capture the waveform as images, to save measurement data as a file [1]. However, the equipment cost is quite expensive, the number of probes is limited, and it is required to reconfigure the experimental system with each different control board. In addition, microcontroller internal values such as control intermediate values, which are useful in debugging control algorithms, cannot be obtained.

From the need for a low-cost data collection method that can be integrated into the microcontroller to control the system, the authors have developed an data acquisition subroutine, which embedded into the control loop routine in the microcontroller to collect sensor data, signal control signals, intermediate values and other operational parameters.

The solution was implemented on a maximum power point tracking (MPPT) control board, which is designed and built by the authors. This is a compact board using a low-cost microcontroller STM32F103C8T6, running simultaneously the MPPT control loop subroutine and the data acquisition subroutine. Data is collected and sent to the computer through the UART protocol using the Direct Access Memory module (DMA). The baud rate is selected at 1152000 bps (~115200 bytes/s). The gathered data is consisted of 4 channels of 100-us high resolution data channels and 20 channels of 2-ms data, occupying a bandwidth of about 106000 bytes/s (92% of selectable bandwidth).

The article will describe the data collection method including: UART protocol, data frame definition, data gathering method, visualization and control program on the computer. The solution is implemented to a MPPT control board to collect, display and analyze its performance.

II. DATA ACQUISITION METHOD

Since data acquisition feature is integrated on an existing control board, the data frame, the transmission method need to be chosen accordingly so that the data acquisition subroutine does not affect the existing control program. The UART protocol was chosen due to its simplicity of implementation, and its availability on most microcontrollers [2].

A. UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER (UART)

This serial port allows the microcontroller to communicate with devices such as other computers, printers, input sensors, and LCD monitors. Serial transmission involves sending bits one at a time, such that the data is spread out over time. The total number of bits transferred in one second is called the baud rate. The inverse of the baud rate is the bit time, which is the time it takes to send a bit. Most microcontrollers have at least one UART.

In UART communication, two UARTs communicate directly with each other. The transmitting UART converts parallel data from a controlling device like a CPU into serial form, transmits it in serial to the receiving UART, which then converts the serial data back into parallel data for the receiving device. Only two wires are needed to transmit data between two UARTs. Data flows from the Tx pin of the transmitting UART to the Rx pin of the receiving UART [3].

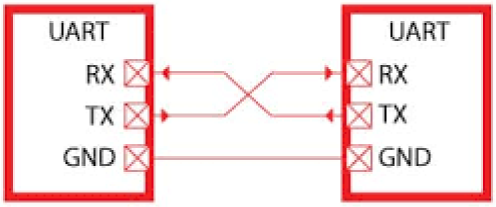


Figure 1. UART Wiring

Each device is capable of creating its own serial clock with a transmission frequency approximately equal to the serial clock in the computer with which it is communicating. A frame is the smallest complete unit of serial transmission. Figure 1 plots the signal versus time on a serial port, showing a single frame, which includes [3]:

– a start bit (which is 0): The UART data transmission line is normally held at a high voltage level when it's not transmitting data. To start the transfer of data, the transmitting UART pulls the transmission line from high to low for one clock cycle. When the receiving UART detects the high to low voltage transition, it begins reading the bits in the data frame at the frequency of the baud rate.

– 5-9 bits of data: The data frame contains the actual data being transferred. It can be 5 bits up to 8 bits long if a parity bit is used. If no parity bit is used, the data frame can be 9 bits long. In most cases, the data is sent with the least significant bit first. Normally in most application, 8 bits of data is used.

– 0-1 parity bit: It is a way to ensure data integrity. UART counts the number of bits with value 1 and compares with parity bit to detect if any bit has changed during transmission.

– 1-2 stop bit (which is 1): To signal the end of the data packet, the sending UART drives the data transmission line from a low voltage to a high voltage for at least two bit durations.

–

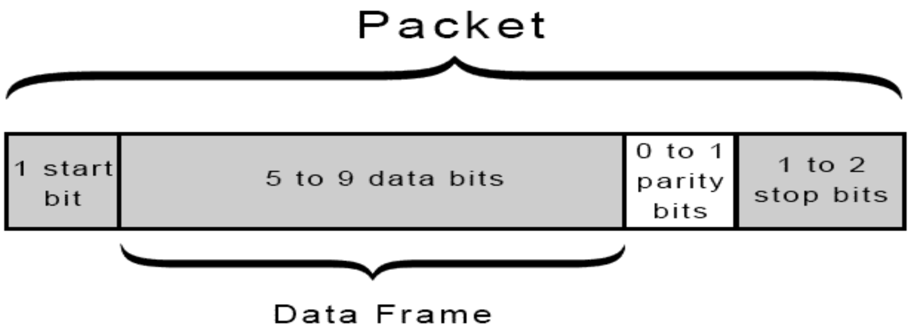


Figure 2. UART data packet [4]

UART is simple, use only 2 wires, no clock signal required, and is very well documented and supported by every microcontroller. Therefore this protocol is used for data acquisition in microcontroller in our project.

In our project, UART configuration was set up as following:

Item	Value	Note
Baudrate	1152000 bps	~1 Mbps
Stopbit	1	
Parity	None	
Databit	8	

B. DATA FRAME

Data acquisition subprogram is embedded with the main control program in the control loop. The control loop is executed with interval of 100 microseconds (frequency 10 kHz). Data acquisition subprogram needs to do the following in this 100-microsecond time window:

– Convert data

– Put data into the payload

– Send data via UART

Data frame must be selected to match with the 100-microsecond time window, so that all data is sent without overlap. If data frame size is A (byte), transfer with speed S (byte/s) the time to transfer the data is :

$$T_{df} = \frac{A}{S}$$

The transmission speed $S = 1152000\ bps = 115200\ \text{byte/s}$

To match $T_w = 100\mu s$ time windows:

$$T_{df} \le T_w$$

$$\Leftrightarrow \frac{A}{S} \le T_w$$

$$\Leftrightarrow A \le T_w S$$

With $T_w = 100\mu s = 100.10^{-6}s$, the maximum data to be sent in 100 µs is

$$A \le 100 * 10^{-6} * 115200 = 11{,}52 \text{ bytes}$$

With that low amount of data to be sent each 100 μs, it is not beneficial, so we proposed a 2ms time window method:

– In the control loop, which is excuted every 100 µs, the 100-µs data is stored in a queue

– After 20 control loops (or 2ms time window), the 2-ms data also is recorded.

– The 2-ms data, along with the 100-µs data in the queue, is sent via UART using DMA

With $T_w = 2ms = 2.10^{-3}s$; the maximum data to be sent in 2 ms is

$$A \le 2 * 10^{-3} * 115200 = 230{,}4 \text{ bytes}$$

Restricted by the maximum data to be sent in 2ms, the data packet is defined by the authors as follows:

– Packet header:

  - o Signature (2 bytes): Start of the data packet
  - o Size (2 bytes): Size of the payload
  - o Type (2 bytes): Type of the data packet
  - o Reserved (2 bytes): For future use
  - o Timestamp (4 bytes): Timestamp of the datapacket, in an unit of 100-µs

– Payload:

  - o 100-us data series with 4 channel, each channel has 20 point of data, each data is 2 bytes
  - o 2-ms data series with 20 channel, each channel has 1 point of data, each data is 2 bytes

Total size of the data packet is 212 byte (about 92% of maximum allowed size).

C. EMBEDED SUBPROGRAM FOR DATA ACQUISITION

The block diagram of data acquisition subprogram and main control program is shown on Figure 3.

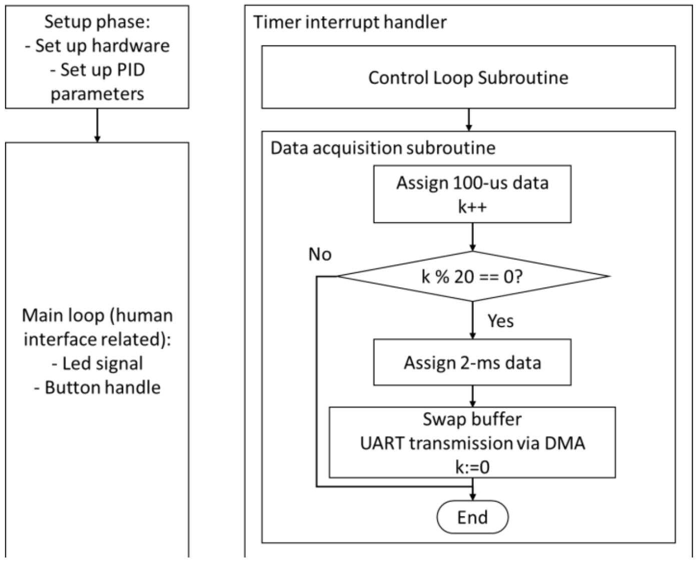


Figure 3. Block diagram of data acquisition program embedded along with control loop

On Figure 3, there are 2 parts:

– Main program:
  - o Setup phase: setup hardware (PWM, Timer, GPIO, UART, etc.), setup software (PID parameters)
  - o Main loop: human interface related functionalities like LED signal, Button handling and control signal from the PC (via UART)

– Timer interrupt handler: this is subprogram called by the timer every 100 µs. It is consisted of 2 blocks:
  - o Control loop routine
  - o Data acquisition routine: handling data acquisition and data transmission via UART

When powered on, main program will set up the hardware with the required parameters. Timer interrupt handler also is set in this phase. After set up, program enter main loop, in which LED signal and buttons are handled to interract with human user. Despite of its name, main loop is not the main part of the program. The main loop doen't require strict real-time feature, because the interraction with human user doesn't require very fast response.

Timer interrupt handler will interrupt the main loop in every 100 µs to do the control loop routine and data acquisition routine. The timer interrupt handler must be executed exactly every 100 µs. The control loop routine and data acquisition routine must be programmed so that it doesn't execute too long (not exceed 100 µs limit).

UART transmission is handled by DMA (Direct Memory Access) unit, which is a digital logic element in computer architecture that can be used in conjunction with the main microprocessor on the same chip in order to offload the memory transfer operations. This significantly reduces the CPU load. As the DMA controller can perform memory to memory data transfers as well as peripheral to memory data transfers or vice versa.

There are 2 buffers to store the data, one as back buffer for data acquisition routine to assign values to, the other is for DMA to send data via UART. When the back buffer is full (every 2 ms, when k % 20 == 0), the buffers are swapped.

D. DATA VISUALIZING PROGRAM ON PC

A program on Windows PC was developed for controlling the embedded program on microcontroller and for visualizing the acquired data. The graphic user interface (GUI) of the program shown on Figure 4 is consisted of the following parts:

(1) UART: select, open, close UART port; start, stop data streaming and logging to disk.

(2) Control the embedded program: the control program in the microcontroller can be control, change parameters, start, stop experiments, etc.

(3) Axis configuration for 4 high resolution channels (100-µs channel)

(4) Data table of 20 low resolution channels (2-ms channel)

(5) Real-time graphic chart of 4 high resolution channels

(6), (7) YT or XY graph type of one or two variables in the data table in (4). YT is time-based graph of the variable by time, XY graph is the polar coordinate graph for visualization of relationship between two variables.

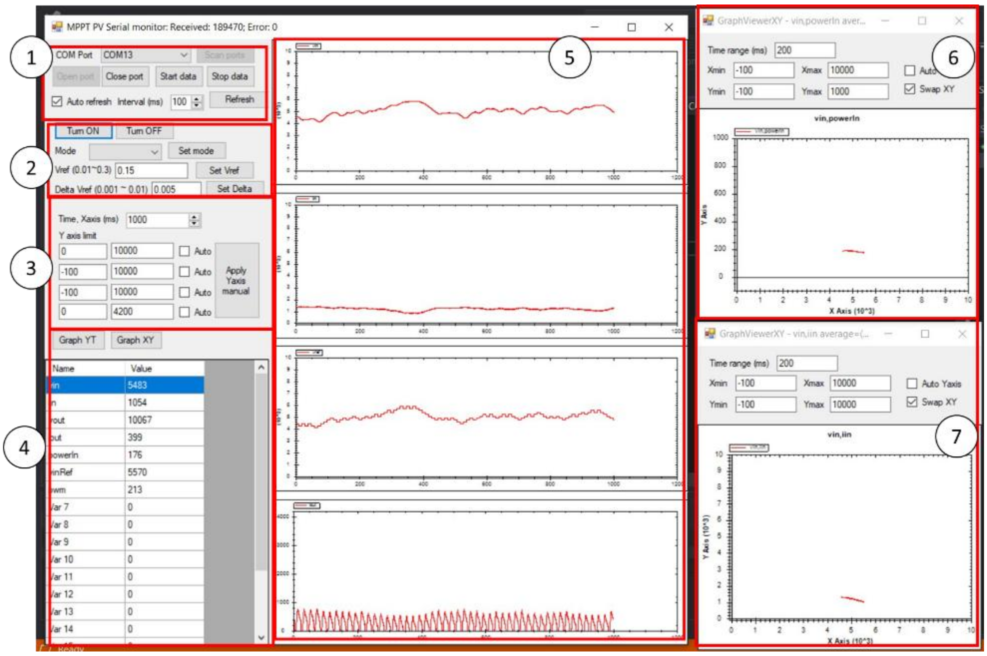


Figure 4. Graphic user interface of the data visualizing program on PC

The data acquired (data from 100-µs channels and 2ms channels) from microcontroller is recorded to the hard-disk when the start/stop data streaming buttons are clicked.

III. IMPLEMENTATION

The solution was implemented to acquire and visualize operational data of a maximum power point tracking (MPPT) control board, which is designed and built by the authors. The data is very useful in analyzing the operation of the system in general and the controller in particular. The data can also helps to illustrate the operating principle of the controller, helping to train students and researchers.

A. SYSTEM STRUCTURE

The system consists of a control board that integrates the data acquisition subroutine, a solar panel, a resistive load and a computer for data acquisition, visualization and configuration. The block diagram of the system is shown on Figure 5. The photo of system and the control bỏad is on Figure 6, showing the control board, DC/DC converter and the PC. Solar panel is not in the photo, but outside.

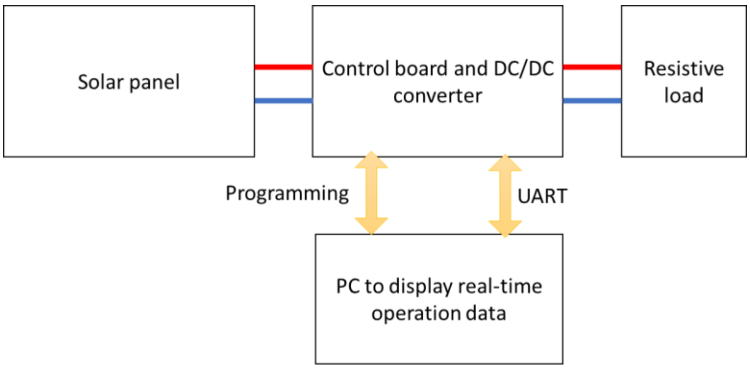
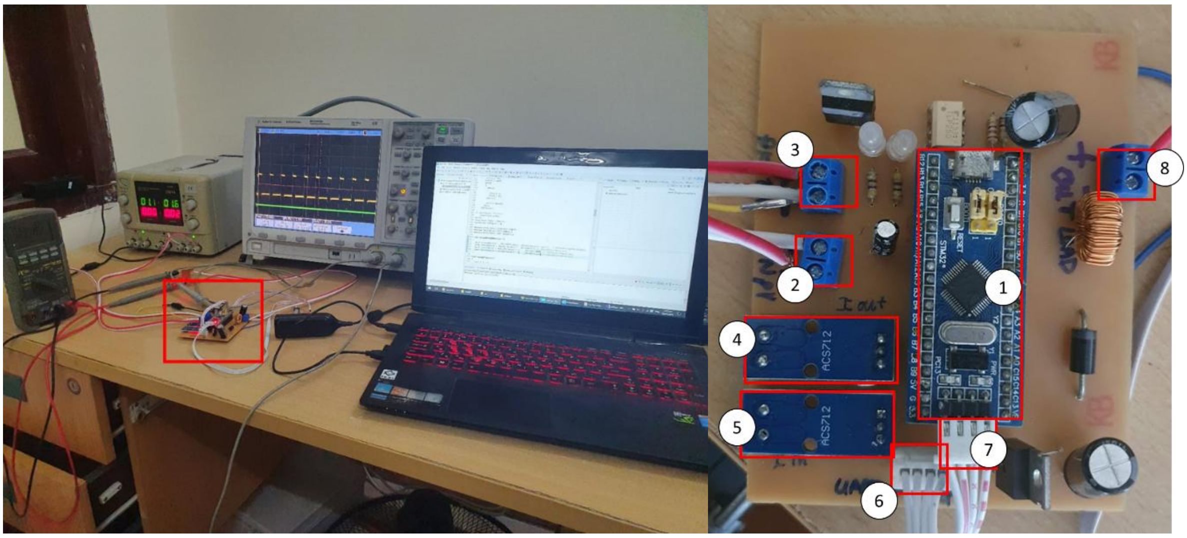


Figure 5. Experimental system



1 –STM32F103C8T6 Kit; 2 – Input from solar panel; 3 – Power supply for control board; 4, 5 – Current sensors ACS 712; 6 –UART port to PC; 7 – Programming port; 8 – Output to resistive load;

Figure 6. Photo of the experimental system and control board

B. SYSTEM FUNCTIONS

For the specific application as MPPT control, the data acquisition subroutine and the graphic user interface (GUI) on the computer are tailored accordingly. The data acquisition subroutine retrieves and sends the following data:

– 100-µs resolution data: current, voltage, voltage set value of the solar panel, load output current value

– 2-ms resolution data: current, voltage, voltage set value of the solar panel, load output current value, control output value, control mode, instruction execution time in control loop as cycle count

The GUI of control panel on PC program has the following buttons:

– Turn ON/Turn OFF: turn on/off the MPPT control loop

– Mode, Set Mode: select the MPPT control mode, including: voltage scan mode (for plotting the characteristics of the solar panel), static voltage mode (to test the PID control), P&O mode (MPPT control using Perturbation and Observation algorithm), ANN mode (MPPT control using Artificial Neural Network)

– Vref: voltage setting value when static voltage mode is selected

– Delta Vref: value of voltage change when P&O mode is selected

C. EXPERIMENTS

With visualization and control program on the computer, experiments can be carried out very conveniently. The following experiments with the system were performed:

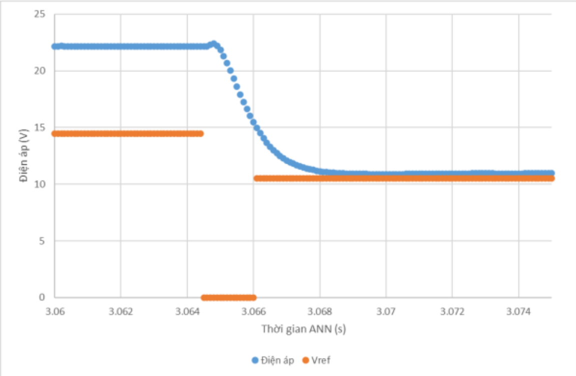
– Test the operation of the PID controller

– Test the characteristics of solar panel

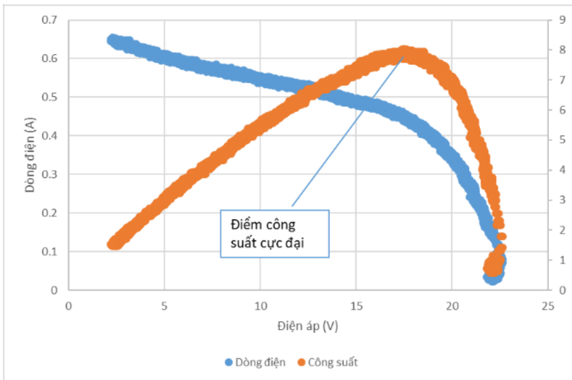
– Test P&O algorithm for MPPT control

– Test the ANN algorithm for MPPT control

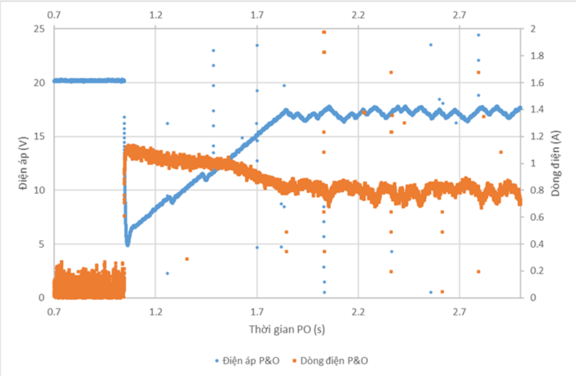
The data is recorded to hard-disk, processed, analyzed and constructed as charts in Excel (Figure 7).



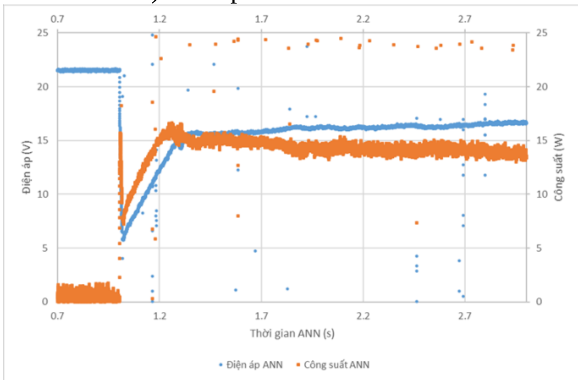
a) Step response of PID controller



b) Solar panel characteristics



c) Voltage and power from solar panel in P&O control mode



d) Voltage and power from solar panel in ANN control mode

Figure 7. Example results of the data acquisition system

Among the collected data, the number of instruction execution cycles in the control loop is also collected. The test results show that the number of cycles is about 448 to 830. The STM32F103C8T6 microcontroller can run at maximum frequency of 72MHz [5], therefore the control subroutine and data acquisition subroutine take the following amount of time to complete:

$$\frac{830}{72 * 10^6} \approx 11.53 * 10^{-6}\, s = 11.53\, \mu s$$

The control loop has a 10kHz sampling period, allowing for a maximum processing time of 100 µs. As such, the embedded data acquisition subroutine does not affect the control loop quality.

IV. CONCLUSION

The article has presented the results of integrating a data acquisition subroutine, which collecting the operational data of the control board, along side with the control loop program. Data is gathered and sent to computer via UART port. The data acquisition subroutine can get the operational data, including not only the input and output of the control board, but also the intermediate data inside the microcontroller, which cannot be obtained just by the external measuring device alone. Besides, the control board can be controlled from the computer, allowing the user to design and run different experimental scenarios.

A computer program has also been developed to display the data collected from the control board and to control the control board. Operational data can be displayed in the form of data tables, graphs over time and graphs of correlation between two variables; control the operation of the control board (ex. on/off, change parameters) and create different experimental scenarios.

Authors has used this data acquisition solution in a MPPT control system. The data is very useful in analyzing the operation of the system in general and the controller in particular. The data can also helps to illustrate the operating principle of the controller, helping to train students and researchers.

The experiment results show that the data acquisition subroutine does not affect the control quality of the system.

REFERENCES

[1]. Hughes, Electrical and electronic technology, Pearson Education, 2015.
[2]. J. Axelson, Serial Port Complete: COM Ports, USB Virtual COM Ports, and Ports for Embedded Systems, Lakeview Research, 2007.

[3]. G. Niemirowski, Serial port and Microcontrollers: Principles, Circuits, and Source Codes, CreateSpace, 2013.
[4]. S. C. |. D. Electronics, “BASICS OF UART COMMUNICATION,” [Online]. Available: https://www.circuitbasics.com/basics-uart-communication/. [Accessed 12 October 2021].
[5]. STMicroelectronics, “STMF103x8 datasheet,” STMicroelectronics, 2015.

Ngo Phuong Le, et. al. “Development of A Data Acquisition Subroutine Embedded In Control Program.” *IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)*, 16(5), (2021): pp. 07-14.