Development of a Data Acquisition and Monitoring System Based on MODBUS RTU Communication Protocol

H.M.K.K.M.B. Herath
Department of Mechanical Engineering
The Open University of Sri Lanka
Nugegoda, Sri Lanka

S.V.A.S.H. Ariyathunge
Department of Mechanical Engineering
The Open University of Sri Lanka
Nugegoda, Sri Lanka

H.D.N.S. Priyankara
Department of Mechanical Engineering
The Open University of Sri Lanka
Nugegoda, Sri Lanka

Abstract:- MODBUS is a serial communication protocol use in industry, which developed by Modicon in 1979. The most common usage of the MODBUS communication protocol is the Programmable Logic Controllers (PLC). The aim of this study is to develop a system for acquiring and analyze measured values from the power measuring device and display them on C# developed GUI. The main objective is to develop an algorithm for MODBUS RTU communication protocol for data acquisition process. In this study, a single power measuring device was used with the aid of MODBUS RTU and RS-485 communication protocols, and data shown in the device was visualized on the C# developed GUI application and analyze them according to the user's requirement. Data acquisitions from the holding registers were made by using the MODBUS Function Code three. The test results were observed and verified accuracy by performing several trials. Data was collected from the power measuring device which was attached to the industrial soft starter panel of a milling machine. Voltage, Current, Total Harmonic Distortion and Power Factor for each phase were retrieved to the C# devolved computer application and verify results with the real-time data of the power measuring device.

Keywords:- *MODBUS, RTU, C#, Serial Port Communication, Microcontroller, PLC, RS-485, GUI, Remote Monitoring.*

I. INTRODUCTION

Energy consumption plays an important role in the growth and socioeconomic development of a country [1]. Ministry of Power and Renewable Energy of Sri Lanka state that demand for electricity is growing at a rate of about 6% per year. Sri Lanka Electricity demand for 2017 consists of 34% of domestic consumers, 29% from Industries, 21% from general purpose consumers and 16% from religious organizations, government institutions, hotels and street lighting. Growth in industrial electrical consumption is recognized as an instant indicator of an economic progress of a country [2]. This is why electrical consumption become monitoring and analyzing is important especially in industries. Most industries in Sri Lanka engage with various projects to monitor and analyze electrical consumption in real-time. This study presents a solution for data acquisition and monitoring system for a tire manufacturing company in Sri Lanka.

Modicon programmable controllers are widely used for communicating with each-other and with the other devices over a variety of networks. Supported networks include the MODBUS Plus industrial networks and standard networks such as MAP and Ethernet. The common communication languages used by all Modicon controllers are the MODBUS protocol [3].

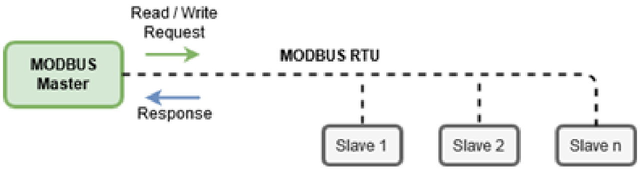


Fig 1:- MODBUS RTU Communication Protocol

Fig. 1 shows the typical MODBUS RTU communication protocol, which use in industry. MODBUS Remote Terminal Unit (RTU) and ASCII are types of protocol that use for the serial communication to create a Master-Slave or Server-Client communication between intelligent devices [4]. It is widely accepted protocol due to its ease of use and reliability. Remote terminal units (RTU's) are connected to sensors and actuators to interface with the physical systems. Industrial control systems comprise a master terminal unit (MTU) that connected to the remote terminal units (RTU) through a communication link [5]. The MODBUS RTU protocol is commonly using for the Building Management Systems (BMS) and Industrial Automation Systems (IAS). This wide application acceptance is due in large part to MODBUS RTU's ease of use.

MODBUS remote terminal unit's messages are a simple 16-bit Cyclic-Redundant Checksum (CRC). The basic 16-bit MODBUS RTU register structure can be used to pack in floating point, tables, ASCII text, queues, and other unrelated data [6]. Modbus RTU communication protocol is associated with the half - duplex communication mode. The mainframe sends the command signals to the terminal devices according to the different slave address. After the corresponding operations, the terminal equipment sends the answering signal to the mainframe. In Remote Terminal Unit mode, the messages are transmitted in a continuous stream format. Each 8-bit (1 byte) is framed by 1 Start bit, 8 Data bits, 0 or 1 Parity bit, 1 or 2 Stop bit.

RS-485 is known as most common wired serial communication protocol in the industrial automation. RS-485 interface is able to support for distance of maximum 1200 meters with multiple devices (up to 32 devices) on the same bus [7]. Table 1 depict the common wired communication protocols that are used in industry.

Characteristics	**Protocol**		
	RS-232	**RS-485**	**GB-Ethernet**
Comms mode	Full-Duplex	Full/Half Duplex	Full/Half Duplex
Max. Distance	15 – 20 m	1200 m	100 m
Max. Transmission	20 Kb.s^{-1}	20 Mb.s^{-1}	1 Gb.s^{-1}
Typical Logic Level	± 5 to ± 15 V	± 1.5 to ± 5 V	± 0.5 to ± 2 V

Table 1:- Common Wired Communication Protocols [8]

In RS-485 system, the master device polls each slave device, waits for the responses, and then polls the next slave device. This phenomenon allows a deterministic behavior by avoiding collisions of data packets. Most measuring devices in industry are compatible with RS-485 communication protocol. Due to these advantages, this study used RS-485 wired communication protocol for the communication between master and slave devices.

II. METHODOLOGY

The method of the development of this system was comprised with three main steps. For the communication with the master unit, RS-485 system protocol intelligent converter was used.

- Communication settings of the measuring device were configured for MODBUS communication Protocol.
- A data acquisition algorithm was developed for the request and respond function of the holding registers.
- Data monitoring Graphical User Interface (GUI) was developed using the C# computer language.

A. Configuring the Data Measuring Device

Before starting the communication process with the PC and measuring device, it is required to configure the measuring device for MODBUS communication protocol. Therefore, measuring device was configured for the MODBUS communication requirements [9]. Table II depicts the communication configurations for Peripheral Id, Transmission Speed, Parity, Data Bit and Stop Bit. Once the measuring device was configured, data acquisition algorithm was used to make a serial connection with the device using RS-485.

B. Communication Protocol

When the measuring device is communicating on a MODBUS serial line using the RTU (Remote Terminal Unit) mode, each 8–bit (1 byte) in a message comprised with two 4–bit hexadecimal characters. Table 3 shows the data frame format of the MODBUS RTU. The main advantage of MODBUS RTU is that its greatest character density allows better data throughput than ASCII mode for the same baud rate. Each message must be transmitted in a continuous stream of characters.

Communication Parameter	**Setting**
Device Id	001
Data Bit	8 bit
Baud Rate	19200 bps
Parity Bit	None
Stop Bit	1 bit

Table 2:- Communication Configurations

Address Code	**Function Code**	**Data**	**CRC-Check**
8 bit	8 bit	n × 8 bit	16 bit

Table 3:- Data Frame Format of Modbus RTU

Connection of the master device with slave devices was made by using the RS-485 communication protocol (Fig. 2). The composition of the RS-485 cabling must be carried out with a meshed screen cable with a maximum distance of 1,200 meters between the device and the master unit. This bus was able to connect to a maximum of 32 measuring devices.

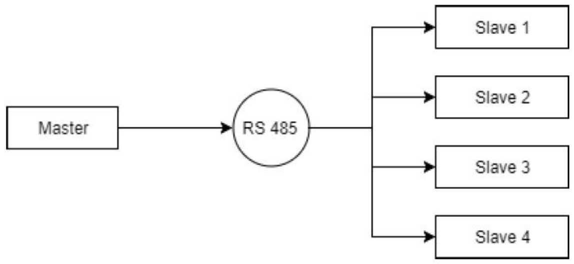


Fig 2:- Master-Slave Communication in MODBUS RTU Protocol.

MODBUS Function Code 03 was used to read the holding register of the power measuring device. Table IV shows the RTU frame description related to the MODBUS communication protocol. It has been responsible to obtain the present value in one or more holding registers of the measuring device. There are several functions that can be found in the MODBUS protocol [10]. For this study, only MODBUS Function Code 03 was obligated for acquiring data from the measuring device to the computer program.

Parameter	**Size**	**Setting**	
Slave Address	1 byte	001 – Peripheral ID	
Function Code	1 byte	03 – Read Holding	
Data	0 – 256 byte	00 00	Starting
		00 03	No of Address
CRC Error Check	2 byte	CRC Low	
		CRC Hi	

Table 4:- MODBUS RTU Frame Description

➢ *Request code*

- Peripheral Address - Slave id.
- Functions Code - Read holding registers (Function Code 03).
- Starting Address - MODBUS register at which to begin reading.
- Register Quantity - amount of registers to read.
- CRC - Cyclical Redundancy Check checksum for the outgoing message.

➢ *Response code*

- Peripheral Address - Slave id.
- Function Code - Function to which the slave is responding.
- Byte Count - Quantity of bytes being read.
- Register Values - 2 × N bytes representing the values read.
- CRC - Cyclical Redundancy Check, checksum for the incoming message.

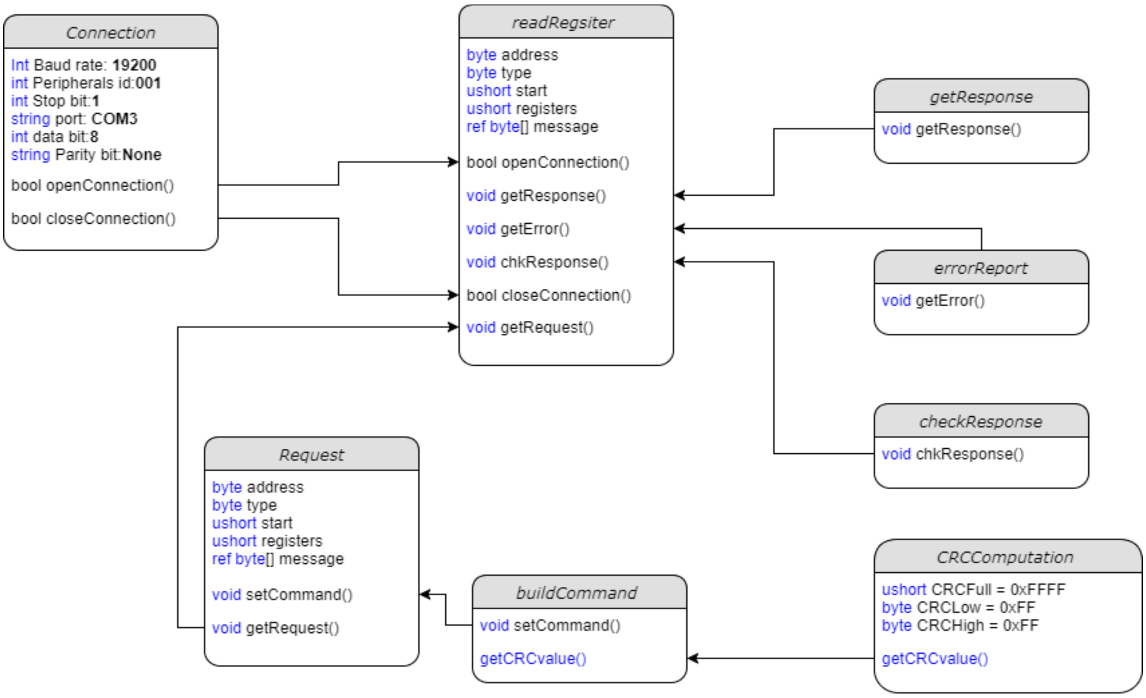


Fig 3:- Data Flow Diagram of the Program

C. *Registers Identification for the Requirements*

For the development of data acquisition algorithm, it was highly required to identify the register's address of the device. Table V represents the MODBUS register's address and its default SI unit for the required parameters to the development of the Data Monitoring System. To achieve the objective of this study, the following set of parameters was identified.

➢ Voltage L1, L2, and L3

➢ Current L1, L2, and L3

➢ Total Harmonic Distortion V1, V2 and V3

➢ Power Factor PF1,PF2 and PF3

D. *Development of Data Acquisition Algorithm*

For this study, C# Dot Net development environment was used to develop the Data Monitoring and Acquisition System. The *"readRegister"* class contains several functions and this class was worked with the support of *"SerialPort"* class which can be found in the *"System.IO.Ports"* namespace of the Dot NET framework [11]. The *"SerialPort"* class was represented the serial port resources for the serial communications.

Parameter	**Units**	**MODBUS Registers**	
		Hexadecimal	
Phase voltage - VL1	V x 10	00	01
Phase voltage - VL2	V x 10	0A	0B
Phase voltage - VL3	V x 10	14	15
Current - AL1	mA	02	03
Current - AL2	mA	0C	0D
Current - AL3	mA	16	17
%THD VL1	% x 10	30	31
%THD VL2	% x 10	32	33

	Units	MODBUS Registers	
%THD VL3	% x 10	34	35
%THD AL1	% x 10	36	37
%THD AL2	% x 10	38	39
%THD AL3	% x 10	3A	3B
Active energy	kWh	3C	3D
Power factor - PF1	P.F. x 100	08	09
Power factor – PF2	P.F. x 100	12	13
Power factor – PF3	P.F. x 100	1C	1D

Table 5:- Register Address for the Required Parameters

The *“readRegsiter”* class was used to retrieve the data from the device through the RS-485. The MODBUS communication protocol was written in the *“readRegsiter”* class (Fig. 3). The *“readRegister”* class was required to use the *“Connection”* class in order to made connection with the measuring device. In here, as for the communication configuration, 19200 bps baud rates, COM 3 communication port, Stop bit - 1, Data bit - 8 and None-parity was used. The *“Connection”* class was responsible for open and closes the serial port connection between the measuring device and PC. Once the connection was made, it is required to send the request code to the device. The request was made by the *“Request”* class. The *“Request”* class builds the message with the aid of *“buildCommand”* class. The *“buildCommand”* class was used for sending MODBUS message to read the holding register of the measuring device. Function Code 03 (read holding registers) was used and *“CRCComputation”* class was used to return the CRC value to the MODBUS message. After the request code was sent to the device, the program will receive the response code. Response code was analyzed using the *“getResponse”* class. The *“getResponse”* class used to return the values of the holding registers. For this study, the data for Voltage (V), Current (A), Power Loss (W), Total Harmonic Distortion (%THD) and Energy were returned to the C# developed GUI (Fig. 4). Following pseudo code represents the procedure of the setting communication with the measuring device.

```
BEGIN
   IF connection not open THEN
        PortName = COM3;
        BaudRate = 19200;
        DataBits = 8;
        Parity = NONE;
      StopBits = 1;
        TRY
          CALL Open connection
        END TRY
        CATCH
               PRINT error opening port
        END CATCH
               PRINT port open successfully
   ELSE
        PRINT port already opened
  END IF
END
```

Building of the outgoing MODBUS message for request data can be done by following process.

```
BEGIN
Initialize byte address, type
Initialize ushort start, registers
Initialize ref byte [] msg
Initialize byte Array CRC[2]
Initialize int l = message length
msg[0] = address
msg[1] = type
msg[2] = start >> 8
msg[3] = start
msg[4] = registers >> 8
msg[5] = registers
CALL CRCComputation
msg[l - 2] = CRC[0]
msg[l - 1] = CRC[1]
END
```

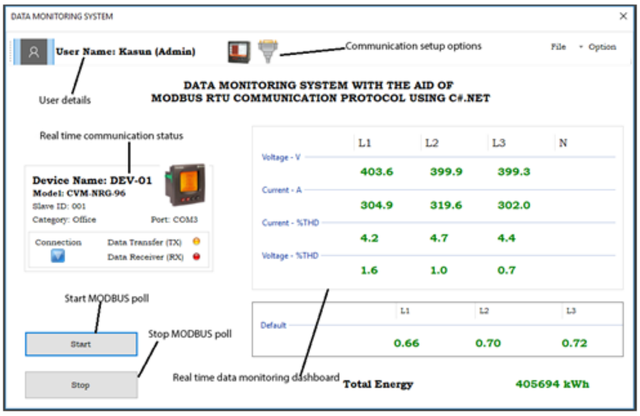
Following pseudo code represent the functionality of the MODBUS Function Code 03 for reading holding registers of the measuring device,

```
BEGIN
Initialize byte address
Initialize ushort start, registers
Initialize ref short [] values
Initialize int i
IF connection open THEN
Discards receive buffer
Discards transmit buffer
CALL buildCommand
TRY
          Writes message to the serial port
                  CALL getResponse
END TRY
          CATCH
                 PRINT Error read the event
          END CATCH
IF response THEN
FOR i=0 TO ((response.Length - 5) / 2) THEN
values[i] = response[2 * i + 3]
values[i] <<= 8
values[i] = values[i] + response[2 * i + 4]
END FOR
```

```
PRINT Data read successful
ELSE
    PRINT CRC error
END IF
ELSE
PRINT Serial port is not open
END IF
END
```

E. Graphical User Interface Design

Graphical User Interface (GUI) (Fig. 4) was designed using the Visual C# programming language. Communication monitor was indicated the state of the serial communication with the power measuring device. When the communication was established, the *"blue"* connection bulb on the screen was getting bright. Data transfer and receiver were displayed by blinking of the RX/TX bulb on the screen. When the communication was interrupted, the RX/TX bulb was red colored and the connection bulb was got darken.


DATA MONITORING SYSTEM

User Name: Kasun (Admin)

DATA MONITORING SYSTEM WITH THE AID OF
MODBUS RTU COMMUNICATION PROTOCOL USING C#.NET

Device Name: DEV-01
Model: CVM-NRG-96
Slave ID: 001
Category: Office
Port: COM3

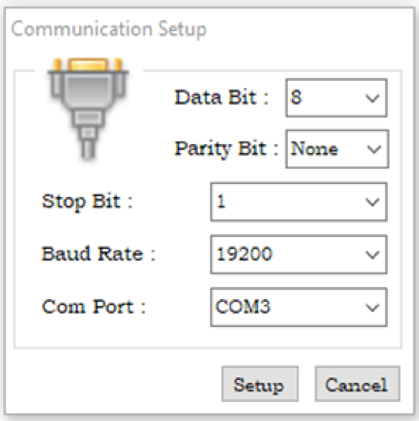
	L1	L2	L3	N
Voltage - V	403.6	399.9	399.3	
Current - A	304.9	319.6	302.0	
Current - %THD	4.2	4.7	4.4	
Voltage - %THD	1.6	1.0	0.7	

	L1	L2	L3
Default	0.66	0.70	0.72

Total Energy 405694 kWh

Fig 4:- Real Time Data Monitoring GUI

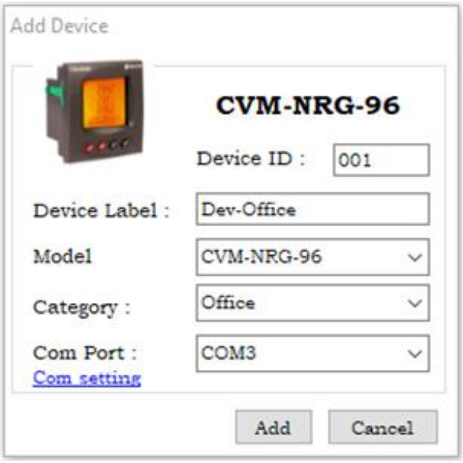
Real-time data monitoring section of the screen was displayed the real-time data which were retrieved from the power measuring device. For this study, Voltage, Current and the Total Harmonic Distortion for the three phases were acquired. Energy record displays the active and apparent consumed power of the power measuring device which attached to the milling machine.


Communication Setup

Data Bit : 8
Parity Bit : None
Stop Bit : 1
Baud Rate : 19200
Com Port : COM3

Fig 5:- Communication Configuration GUI

Total Active energy and Apparent Energy also displayed in the energy record which was used for calculating the electricity bill and other calculations [12] [13] [14]. Before the main program start it was necessary to initialize the serial communication on the developed computer application. Setting up communication was made by using the communication setup GUI (Fig. 5). Data Bit, Parity Bit, Stop Bit and Baud rate was added in the communication setup GUI. Once the measuring device was attached to the RS-485 intelligent converter, COM Port was initialized automatically by the system. By default, the program was configured to baud rate at 19200 bps.


Add Device

CVM-NRG-96

Device ID : 001
Device Label : Dev-Office
Model : CVM-NRG-96
Category : Office
Com Port : COM3

Fig 6:- Device Configuration GUI

When a new device attached to the RS-485 intelligent converter, system needs to configure for the new device. A new device was configured using the Add Device GUI (Fig. 6). After the installation of a new device, Device ID and COM Port were automatically detected and setup by the system itself.

III. TEST RESULTS

The results were observed and verify accuracy by obtaining several trials. All the results were associated with the industrial soft starter panel which attached to the milling machine of a tire manufacturing company. Data was recorded once user pressed the start button on GUI.

Parameter	Test Results	
	Trial 1	Trial 2
Phase voltage - VL1	403.6 V	412.6 V
Phase voltage - VL2	399.9 V	410.5 V
Phase voltage - VL3	399.3 V	411.7 V
Current - AL1	304.9 A	288.9 A
Current - AL2	319.6 A	300.1 A
Current - AL3	302.0 A	284.1 A
%THD VL1	1.6	1.8
%THD VL2	1.0	1.0
%THD VL3	0.7	1.0
%THD AL1	4.2	4.3

	Test Results	
%THD AL2	4.7	4.1
%THD AL3	4.4	4.4
Power factor - PF1	0.66	0.60
Power factor – PF2	0.70	0.64
Power factor – PF3	0.72	0.68

Table 6:- Test Results Display on GUI

Table 6 show the acquired data (Voltage, Current, THD and PF) on the GUI Dashboard for first and second trials. Fig. 7 to Fig. 11 shows the data in the power measuring device for first trial.

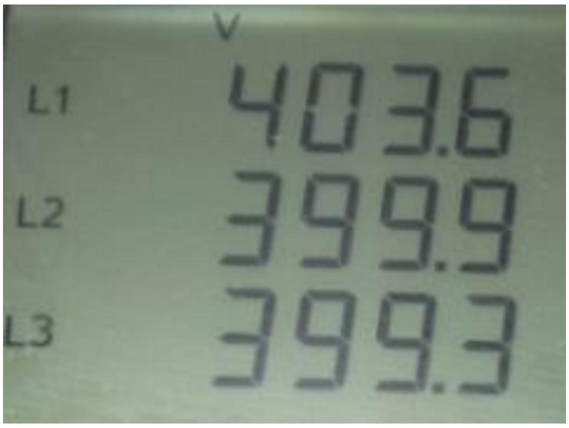


Fig 7:- Voltage of L1, L2, and L3 on Trial-1

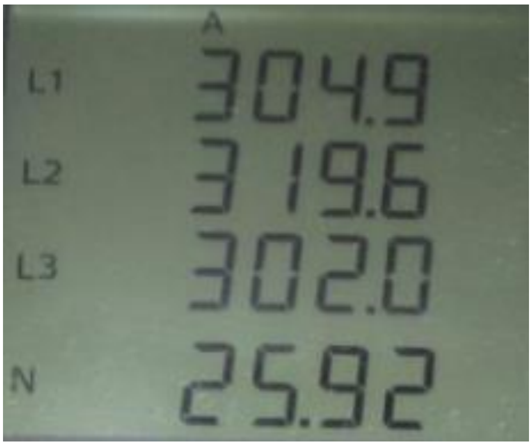


Fig 8:- Current of L1, L2, and L3 on Trial-1

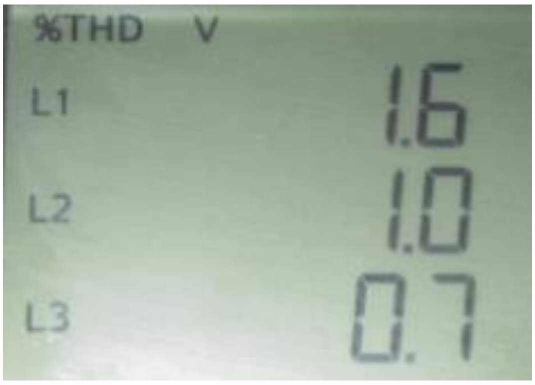


Fig 9:- THD Voltage of L1, L2, and L3 on Trial-1

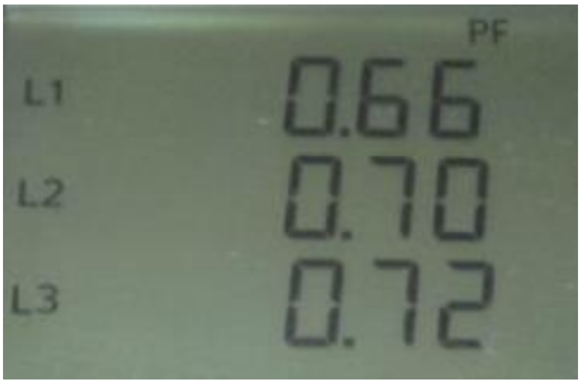


Fig 10:- Power Factor of L1, L2, and L3 on Trial-1

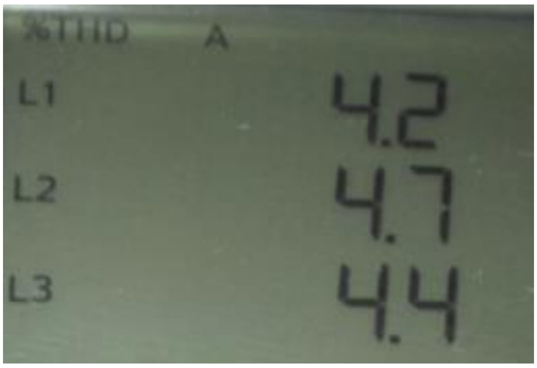


Fig 11:- THD Current of L1, L2, and L3 on Trial-1

Fig. 12 to Fig. 16 associate with the data shown in measuring device for second trial. All the parameters are represented for phase L1, L2 and L3.

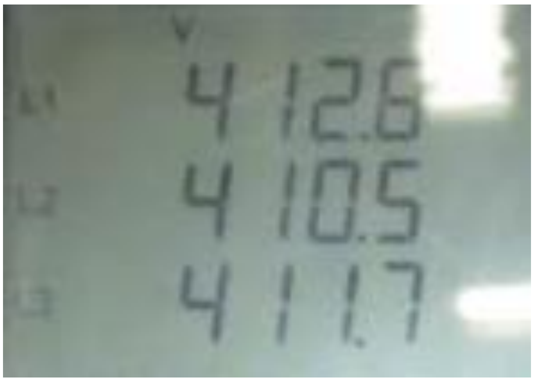


Fig 12:- Voltage of L1, L2, and L3 on Trial-2

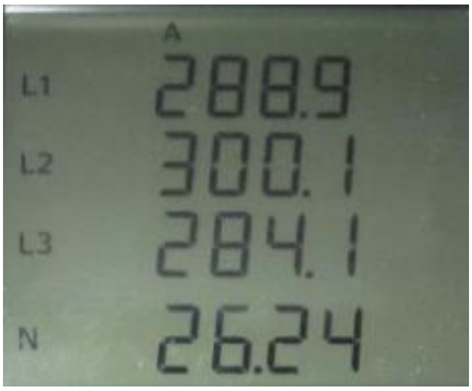


Fig 13:- Current of L1, L2, and L3 on Trial-2

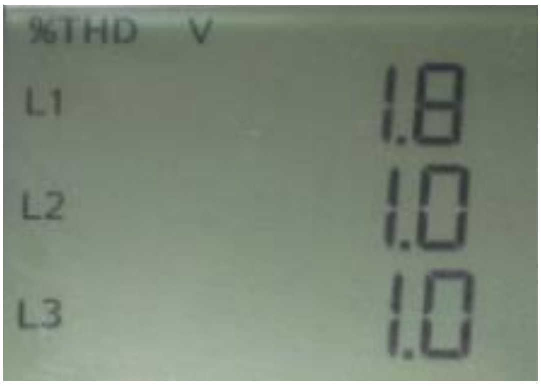


Fig 14:- THD Voltage of L1, L2, and L3 on Trial-2

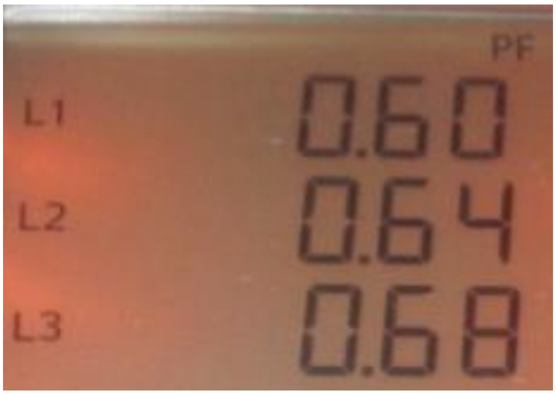


Fig 15:- Power Factor of L1, L2, and L3 on Trial-2

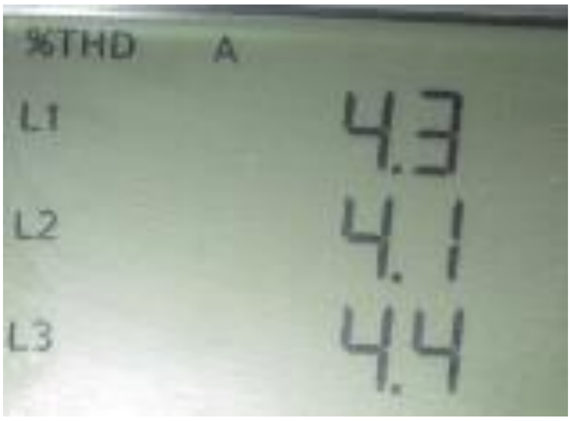
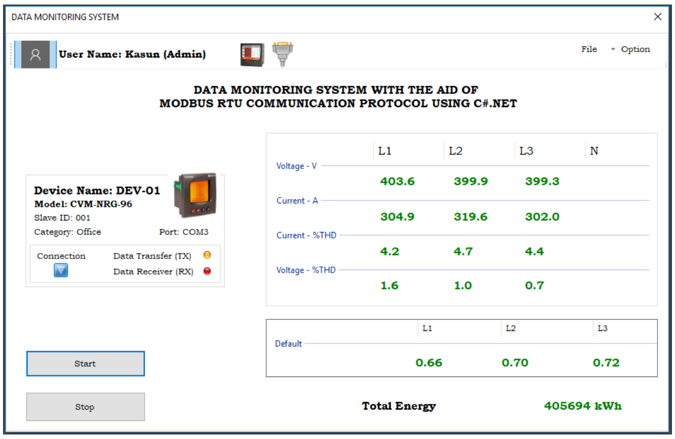


Fig 16:- THD Current of L1, L2, and L3 on Trial-2



DATA MONITORING SYSTEM

User Name: Kasun (Admin)

DATA MONITORING SYSTEM WITH THE AID OF
MODBUS RTU COMMUNICATION PROTOCOL USING C#.NET

Device Name: DEV-01
Model: CVM-NRG-96
Slave ID: 001
Category: Office
Port: COM3

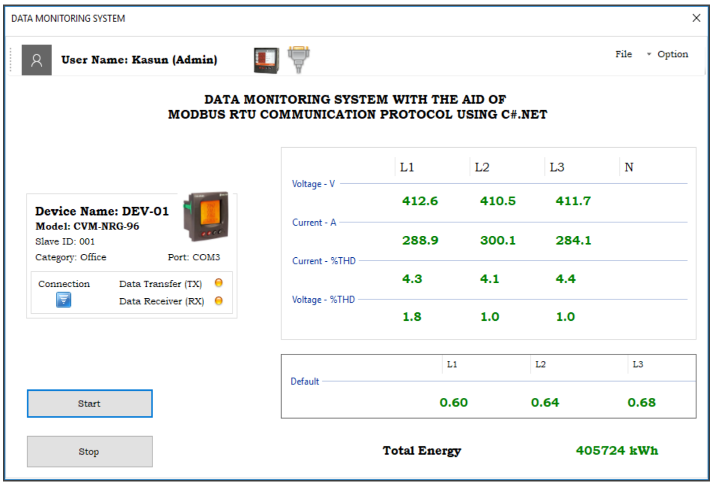
Connection
Data Transfer (TX)
Data Receiver (RX)

	L1	L2	L3	N
Voltage - V	403.6	399.9	399.3	
Current - A	304.9	319.6	302.0	
Current - %THD	4.2	4.7	4.4	
Voltage - %THD	1.6	1.0	0.7	

	L1	L2	L3
Default	0.66	0.70	0.72

Total Energy 405694 kWh

Fig 17:- Real Time Data Visualize on Trial-1



DATA MONITORING SYSTEM

User Name: Kasun (Admin)

DATA MONITORING SYSTEM WITH THE AID OF
MODBUS RTU COMMUNICATION PROTOCOL USING C#.NET

Device Name: DEV-01
Model: CVM-NRG-96
Slave ID: 001
Category: Office
Port: COM3

Connection
Data Transfer (TX)
Data Receiver (RX)

	L1	L2	L3	N
Voltage - V	412.6	410.5	411.7	
Current - A	288.9	300.1	284.1	
Current - %THD	4.3	4.1	4.4	
Voltage - %THD	1.8	1.0	1.0	

	L1	L2	L3
Default	0.60	0.64	0.68

Total Energy 405724 kWh

Fig 18:- Real Time Data Visualize on Trial-2

Acquired data in GUI for both Trial-1 and Trial-2 shows respectively in Fig. 17 and Fig. 18. Total energy was displayed on the GUI for both trials. Real-time data was visualized until user press the stop button on the GUI.

IV. CONCLUSION

According to the test results, acquired data for electrical parameters (Voltage, Current, Total Harmonic Distortion and Power factor) of the GUI application on each trial was same as the data displayed on the power measuring device. Moreover, when the connection established "Connection" indicator on GUI was turned to blue color and observed that measuring device was connected to COM Port 03. Therefore, it is possible to use C# serial communication environment for development of the data acquisition algorithm for power measuring devices that supported by the MODBUS communication protocol.

REFERENCES

[1]. J. Mawejje and D. N. Mawejje, "Electricity consumption and sectoral output in Uganda: an empirical investigation," Journal of Economic Structures, vol. 5, no. 1, 2016.

[2]. C. Zhang, K. Zhou, S. Yang, and Z. Shao, "On electricity consumption and economic growth in China," Renewable and Sustainable Energy Reviews, vol. 76, pp. 353–368, 2017.

[3]. "Modbus Organization," Modbus Technical Resources. [Online]. Available: http://modbus.org/tech.php. [Accessed: 13-Jun-2020].

[4]. G. B. M. Guarese, F. G. Sieben, T. Webber, M. R. Dillenburg, and C. Marcon, "Exploiting Modbus Protocol in Wired and Wireless Multilevel Communication Architecture," 2012 Brazilian Symposium on Computing System Engineering, 2012.

[5]. T. Morris, R. Vaughn, and Y. Dandass, "A Retrofit Network Intrusion Detection System for MODBUS RTU and ASCII Industrial Control Systems," 2012 45th Hawaii International Conference on System Sciences, 2012.

[6]. “Modbus RTU Protocol Overview,” Real Time Automation, Inc., 25-Nov-2019. [Online]. Available: https://www.rtaautomation.com/technologies/modbus-rtu/. [Accessed: 13-Jun-2020].

[7]. S. Jian, L. Peng, F. Yaqiong, L. Ruixiang, and C. Le, “The implementation and application of programming port communication between industry PC and Mitsubishi FX series PLC,” 2008 3rd International Conference on Intelligent System and Knowledge Engineering, 2008.

[8]. R. Capocci, G. Dooly, E. Omerdić, J. Coleman, T. Newe, and D. Toal, “Inspection-Class Remotely Operated Vehicles—A Review,” Journal of Marine Science and Engineering, vol. 5, no. 1, p. 13, 2017.

[9]. “Samey.is,” CVM Power analyzers. [Online]. Available: http://www.samey.is/_pdf/_circutor/Circutor_CVM_Power_Analyzers_Catalog.pdf. [Accessed: 13-Jun-2020].

[10]. Distantcity, “Simple Modbus Protocol in C# / .NET 2.0,” CodeProject, 18-Oct-2007. [Online]. Available: https://www.codeproject.com/Articles/20929/Simple-Modbus-Protocol-in-C-NET. [Accessed: 13-Jun-2020].

[11]. Dotnet-Bot, “SerialPort Class (System.IO.Ports),” (System.IO.Ports) | Microsoft Docs. [Online]. Available: https://msdn.microsoft.com/en-us/library/system.io.ports.serialport(v=vs.110).aspx. [Accessed: 13-Jun-2020].

[12]. J. Wang and P. Wang, “Power Monitoring System Based on LabView in the Application of Intelligent Building,” Applied Mechanics and Materials, vol. 333-335, pp. 2475–2478, 2013.

[13]. S. Kumar, R. Poonia, A. Kumar and V. Kumar, "SCADA IMPLEMENTATION BASED ON RF TECHNOLOGY", International Journal of Information Research and Review, vol. 03, no. 04, pp. 2253-2257, 2016.

[14]. S. Belekar, A. Desai, M. Parit and A. Dakre, "PLC SCADA based Distribution Monitoring & Control", Multidisciplinary Journal of Research in Engineering and Technology, vol. 1, no. 1, pp. 105-110, 2014.