Proceedings of the National Power Systems Conference (NPSC) - 2018, December 14-16, NIT Tiruchirappalli, India

Extraction of Data from an RS-485 enabled Multi Function Meter for Building Monitoring Systems

Sandhya CSR*, Sri Vaishnavi Tirunagari†, Shubhasmita Sahoo‡, Pradeep Kumar Yemula§ *MIEEE*,
Institution: *§ Department of Electrical Engineering, Indian Institute of Technology Hyderabad, Kandi, Hyderabad, India,
†‡ Department of Electronics and Instrumentation Engineering, National Institute of Technology Rourkela, Odisha, India.
Email: * ee17resch11015@iith.ac.in, † 115ei0368@nitrkl.ac.in,
‡ 115ei0359@nitrkl.ac.in, § ypradeep@iith.ac.in

***Abstract*—Data analytics on the electrical parameters is needed in order to improve energy consumption efficiency and prevent adverse events such as power outage, short circuit etc.,. It also helps in protecting and increase the longevity of the electrical equipment. The electrical consumption data is measured usually using the multifunction meter(MFM) installed in residential or commercial buildings. In the commercial buildings, these meters are installed in the panel room(centralised) and in all floors/departments(decentralised). The extraction of data is the first step for the process of data analytics. This paper discusses how the extraction of data is possible from these installed multi function meters. In order to demonstrate the data retrieval process, a hardware lab set up is implemented using RS 485 communication interfaced multifunction meter(MFM). In addition to the MFM, certain modules(both hardware and software) used for the data extraction are also explained in a step by step manner. The validity of the method adopted to extract the data is verified using the real time loading instants.**

***Index Terms*—Multi Function Meter, RS-485 Standard, Modbus Protocol, Serial Communication, SCADA, Smart building, Submetering.**

I. Introduction

Smart building is a system of interconnected systems which has high-level control and sensors along with automation, communication, and analytics capabilities [1]. It provides tools to predict and responds to comfort, maintenance, and energy performance issues which leads to better equipment maintenance, higher occupant satisfaction, reduced energy consumption, and cost. In Reference [2] a framework is defined to identify the most convenient strategies for monitoring and data interpretation in smart environments. Several dimensions such as objectives, sampling rate, temporal usage, segmentation, redundancy etc are explored, in the context of smart buildings.

The monitoring system for building energy consumption consists of electrical power system monitoring, environment monitoring, and heating ventilation air conditioning (HVAC) system monitoring. Energy monitoring involves various technologies such as data collection, communication, and databases. These should be advanced enough to grasp the status of energy consumption in real-time [3].

Submetering gives the local electrical consumption pattern. Efficient energy utilisation can be achieved by analysing the recorded data [4]. It could help time-varying electricity price based management of the operation of loads and in getting information regarding the quality of power. Submetering can help in reducing the energy expenses in the large buildings by performing energy auditing.

Most of the energy meters and multi function meters installed for measuring the energy consumption (Energy, Power, Power factor) in a household or any other commercial buildings communicate to the centralised SCADA systems using RS-485 standard of serial communication. Data Analytics is done on the data derived from these meters. Thus the first step is to extract the data from these meters in order to carry out further analytics. In fact, the importance of data and analytics on data has been recognised quite early where non-intrusive identification of electrical loads was made just by analysing the data [5].

While the modern smart meters are coming with inbuilt IP connectivity, the regular multifunction meters have RS485 communication capabilities. However, since the IP based meters are new technology a majority of existing buildings already have the multifunction meters and hence it is needed to examine how to extract data from them. Further there is an advantage of RS485 over Ethernet. Ochai et al., note that [6] RS485 is one of the most optimal communication media for facility networking, as opposed to Ethernet where many UTP cables and switching hubs need to be deployed. A multi function meter installed in electric panel room (of households,industries and practically any other commercial buildings utilizing electrical energy) measures various electric parameters like line voltage, current, frequency, power factor, power consumed, energy consumed of all the three phases.

In the present scenario many existing buildings are aiming to install energy monitoring system so as to move towards the concept of smarter buildings. Implementers are facing hurdle at the point of data extraction from the meters. In some cases, the meters themselves are not communicable and need to be replaced. Even when communicable meters are available, knowledge on how to extract data from them is not common, resulting in dependence on suppliers who charge heavily. The objective of this paper is to explain this crucial first step of data extraction from multi-function meters assuming they are communicable and have RS485 protocol support. The system can be built by using easily available general purpose devices and open source software as explained in this paper. We believe that this will result in reducing the

978-1-5386-6159-8/18/$31.00 © 2018 IEEE

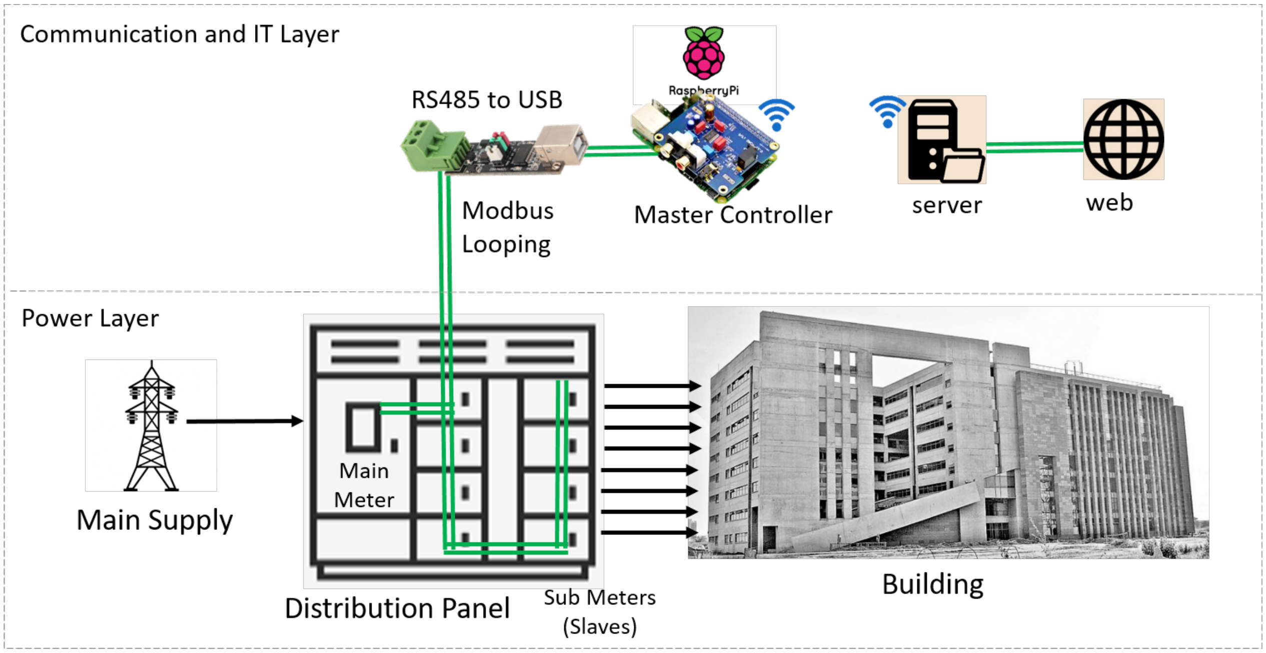

Fig. 1. Block Diagram of Building Monitoring System

overall cost of building monitoring system and increasing the general understanding of the process.

The paper is organized as follows: Section II presents a block diagram of the building monitoring system. Section III presents the step-by-step procedure of executing the system. Section IV presents the hardware lab set-up and case studies. Conclusion and future work follows in Section V.

II. Block Diagram of the Building Monitoring System

The power is distributed from the main power supply to the entire building. The building power consumption is recorded by the main meter installed at the main supply. The purpose of this meter is for billing and accounting. This meter is usually installed by the utility for the above mentioned purpose. In addition to this main meter, the building users can install several meters dedicated for individual floors/departments of the building. The purpose of these meters is to provide valuable data and feedback on consumption pattern to the building users. This concept is called submetering. A system built using all the data from submeters for visualisation, analytics and decision making is referred to as building monitoring system(BMS).

The Block diagram of the building monitoring system is shown in the Fig. 1. In the power layer, the main supply is given to the distribution panel where the main meter is used for metering and billing for the entire building. However, the distribution panel in turn has dedicated circuits for distribution of power to different parts of the building. Each of these circuits are also fitted with submeters which are used for monitoring power flows in respective parts of the building.

In the communication layer, a modbus looping wire is connected to all the multifunction meters. These meters act as slaves where as the on the other end, a master controller is connected through an RS485 to USB converter. A general purpose micro controller, namely, Raspberry pi has been used. On the micro controller software is installed which can communicate with the meters using the RS485 Modbus protocol in RTU mode and read the data from the slave meters. The overall objective is to extract the data from this meter and write it to a file and store it in a desired server location using wifi. This requires a microcontroller to control the data being requested and sent to and from the meter. Microcontroller also has to have a wifi module embedded in it. A centralised server is needed, so a dedicated PC is to be used.

The physical connections of meters(looping) is shown in the Fig. 2. Each meter is identified using a unique slave ID. In case of Modbus, each slave device must have a unique assigned identifier ranging from 1 to 247. Programming software is used to set the Modbus slave id; it will be permanently stored in the memory of the device. Whenever a master sends a message on the network, every device receives the signal. Master embeds the value of the Modbus unique id of the slave in the messages. All of the slaves will receive the slave id and will compare the received identifier with their internal Modbus id. If it matches then, they will respond to the message. Slave IDs can be set by configuring the meter.D+ Pin and D- Pin refer to the RS485 communication port of the meter. All the real time values that are measured by the meter will be communicated to any server using these pins. The mode of communication is serial communication and the standard used is RS-485. The RS-485 standard interface in this meter uses MODBUS Protocol in RTU mode. All the data coming from these meters is pooled

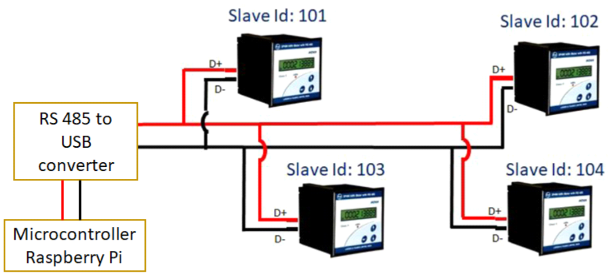

Fig. 2. Connection of Meters to the modbus loop

28 bits	8 bits	8 bits	N*8 bits	16 bits	28 bits
START	ADDRESS	FUNCTION	DATA	CRC CHECK	STOP

Fig. 3. String processed as Modbus Protocol

into a database or written into a file of a centralised server and thus further analysis of the data takes place. The serial port of the RS 485 is configured to run at baud rate or bits per second. It differs from device to device. It is to be taken care that all the devices connected to the same bus should have same baud rate. Were the maximum baud rate of the RS 485 protocol goes up to is 115200 bits per second but not be much longer than 2500 feet. At a longer distance, the baud rate can be 76800 bits per second.

RS 485 (EIA 4850) is the most common wired serial communication protocol. The noise immunity and multi dropping ability made it be the preferred connection for the industrial or pc controllers for data collection. It supports the electrical signal cancelling by using the twisted pair cable for the transmission, and it also supports multiple devices to communicate on the same bus through long distance about 4000 m [7].

The multi-functional meter works on Rs 485 Modbus protocol, which gets data from the device only when it sends a request, which helps in avoiding unnecessary communication on a bus. It works on half-duplex data transmission [8].

Modbus protocol supports a maximum of 247 devices to connect without a repeater (with only a Master initiating the communication). It can send max 32-node information at a time [9]. Multiple devices on one network are called Multidrop. The looping connection is most preferred for Modbus. The positive terminals (D+) are connected in chain pattern and similarly negative terminals (D-). The data from the MFM is a string of 34 register values. The string consists of communication parameters like start call, Address, function code, data, CRC, stop. This can be seen as shown in Fig.3

Start call: Initiates the start of the device, which has 28 bits or 3.5 characters. Address: It stores the slave address. Function code: It specifies the type of data to read or write.

Data: Its length depends on the type of message type. CRC: It checks the error in data during the data transmission. Stop call: It terminates the communicate the request for the device which has 28 bits or 3.5 characters.

TABLE I
FUNCTION CODE VALUES

Code	Description
01	Read Coil Status
02	Read Input Status
03	Read Holding Register
04	Read Input Register
05	Write Single Coil Register
06	Write Single Register
15	Multiple Coil Write
16	Multiple Register Write

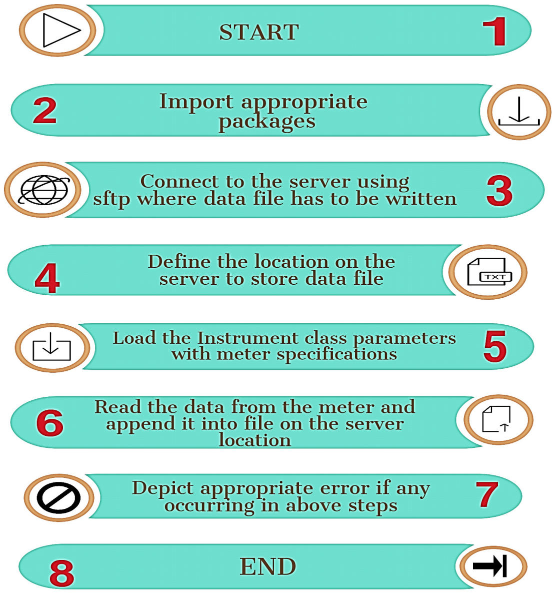
START
1
2
Import appropriate packages
Connect to the server using sftp where data file has to be written
3
4
Define the location on the server to store data file
TXT
Load the Instrument class parameters with meter specifications
5
6
Read the data from the meter and append it into file on the server location
Depict appropriate error if any occurring in above steps
7
8
END

Fig. 4. Flowchart of the code implemented

In this section the various components of the BMS system are described. In the next section the procedure for data extraction will be presented.

III. Procedure for Data Extraction

Step 1: First the meters have to be configured by assigning unique slave IDs. The other parameters such as baud rate, number of data bits, number of start bits, parity should also be set. This can be done manually via the interface of the meter using the navigation buttons. Alternately, the configuration can also be done through software via the data port, by writing the parameters in the relevant registers specified in the user manual of the meter.It should be ensured that all the devices which are on the modbus network must have same settings (except the slave IDs) of the configuration parameters.

Step 2: Ensure that the meters are looped correctly and they are connected to the micro-controller via RS485/USB interface.

Step 3: Since Raspberry pi is general purpose controller and it is wifi enabled, we will consider it as the controller for the following discussion. The python code for extracting the data from meters is given below.

```
import time
import minimalmodbus
import serial
import pigpio
import datetime
import paramiko
try:
client = paramiko.SSHClient()
client.load_system_host_keys()
client.set_missing_host_key_policy(paramiko.WarningPolicy())
client.connect(”ip”,port,”username”,”password”)
dirname=”/home/sglab/Dropbox/meterdata/new”
sftp = client.open_sftp()
sftp.mkdir(dirname)
except IOError:
print(’connection␣with␣server␣failed␣!!!’)
pass
slave_id_of_meters=[101]
instrument=[]
files=[]
i=0
for x in range(0,len(slave_id_of_meters)):
    instrument.append(minimalmodbus.Instrument...
            (’/dev/ttyUSB0’,slave_id_of_meters[x]))
    instrument[x].serial.baudrate=9600
    instrument[x].serial.bytesize=8
    instrument[x].serial.parity=serial.PARITY_EVEN
    instrument[x].serial.stopbits=1
    instrument[x].serial.timeout=1
    instrument[x].mode = minimalmodbus.MODE_RTU
    files.append(sftp.open(dirname+ ’/’ + ...
            str(slave_id_of_meters[x])+’test’, ’a’))
while 1:
    try:
        i+=1
        for x in range(0,len(slave_id_of_meters)):
            before_read = datetime.datetime.now()
            test_reg=instrument[x].read_registers(0,2,4)
            after_read = datetime.datetime.now()
            files[x].write(str(time_now)[:23]+...
                                    str(test_reg)+’\n’)

    except:
        print(”exception␣occured␣after␣” +str(i)...
                    +”␣lines␣hence␣restarting”)
        i=0
        time.sleep(1)
```

Step 4: The structure of the code is as follows: The initial few lines are used to import the necessary packages. The most important package is *minimalmodbus* [10]. This package is freely available at the link mentioned. It is an easy-to-use Modbus RTU and Modbus ASCII implementation for Python. Another important package to note is *paramiko*, which is a Python implementation of the SSHv2 protocol, providing both client and server functionality [11]. This package is used for opening connection with the server by providing its IP address, port, username and password. This connection is required as later we will be writing the data as strings into files on this server. The later sections of code include lines to establish connection with server, followed by setting of communication parameters to match with that of the meter. Then an infinite while loop is started where the registers of the meter are read one by one and the data is received in strings for writing into the files on the server. To summarize, Fig. 4, shows the main steps involved in the code.

Step 5: Since the code has been written and stored in the controller it can now be executed.

Step 6: If everything goes well, the files in the server will be written with the data measured by the meters in CSV form. From here, we can use any method to store / analyse the data.

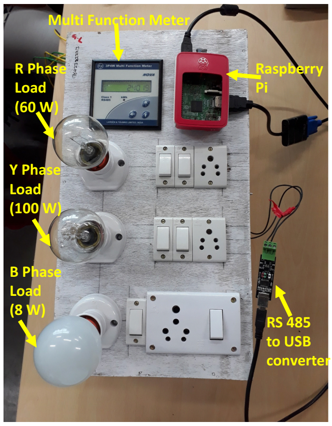

Fig. 5. Lab Set Up

Thus, in this section the steps for extracting data from meters is explained in detailed, including the complete python code.

IV. Lab Hardware Setup for Data Extraction

In this section the test setup for data extraction is described.

Before implementing the above building monitoring system, we have tested the same with a single meter in our lab. The cost of lab setup is as follows - MFM:Rs.4,250; R-pi:Rs.2,750; Miscellaneous:Rs.550; RS485 to USB converter:Rs.450. Total cost is about Rs.8,000 (All amounts in Indian Rupees). The overall lab set up is depicted in the Fig. 5. The internal connection of the MFM can be understood from the Fig.6.

The multi-function meter used is of L&T make Nova. A Raspberry Pi is used as a microcontroller. To connect the meter to the Raspberry Pi, a RS 485 to USB converter is used. The protocol used by this meter is MODBUS protocol in RTU mode. MFM is connected to the three phase supply. The communication ports (here pin numbers 7 and 8) of the MFM are connected to the RS 485 to USB converter module. This in turn is connected to the USB port of the Raspberry Pi.

Next we examine the 3 phase 4 wire multi function meter which is called Nova of L&T make marked as “Multi Function Meter” in Fig. 5.

The back of the meter shows wiring of both electrical and data cables as shown in Fig. 6. Any multi function meter generally has current and potential transformers embedded in it. The input and output pins for these transformers are made available so that the live wires could be connected to them for measurement of currents, voltages and thus in turn all other electrical parameters like power factor, energy and power(to name a few). As a case study, a Nova 3 Phase 4 Wire Multi Function meter of Larsen and Toubro (L&T) make is taken into consideration. This Meter has a total of 16 Pins. Pin sets {1,2}, {3,4}, {5,6} are for measuring the currents (current transformer connections) in each of the three phases. The green wires in the diagram show current coil connections. Pins 9, 10, 11, 12 are for measuring the voltages (potential transformer connections) of each phase and the neutral. The

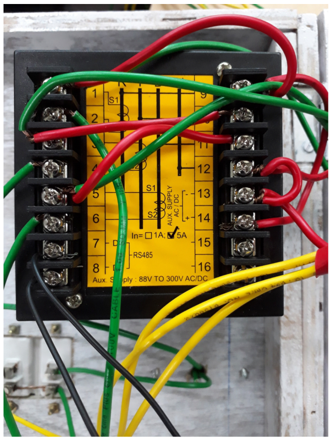

Fig. 6. Back of the Meter

red wires in Fig. 6 are the voltage coil connections. Thus the 3 phase, 4 wire connections are done. Input current and voltage level are very high to measure directly. So a current transformer is used to isolate and step down the current value in the secondary and in turn connected to the meter for measurement. Potential transformer serves the same purpose but for voltage measurement. CT Ratio: It is equal to the ratio of primary current input to the secondary current output of current transformer at full load. Similarly PT Ratio corresponds to voltage ratio between primary and secondary. Pins 13 and 14 are used for powering the meter. The meter extracts power from any one of the phases connected. Pins 15 and 16 are reserved for future functionality by the meter manufacturer, hence they are not used. The meter internally calculates all the other electric parameters using these inputs. Pins 7 and 8 are the RS-485 serial communication pins and it is from these pins that the data is extracted from the meter. A detailed diagram of the internal connection of this meter can be understood from the Fig 6.

A. Connection of Loads

For the test set up, different loads are connected across each phase as mentioned in the table. II. The last two columns also show the turn on and turn off times of these loads on different phases. The objective is to verify whether the data extracted from the meter matches with the load applied to the meter in the pattern mentioned in table. II. Out of the various measurements derived from the meter, power, voltage and frequency are examined.

TABLE II
CONNECTION OF LOAD

Phase	Load Type	Wattage	Turn ON timestamp	Turn OFF timestamp
R	Incandescent	60 W	16:10:00	16:14:00
Y	Incandescent	100 W	16:11:00	16:15:00
B	CFL	8 W	16:12:00	16:16:00

TABLE III
CASE STUDY EVENTS

Time Stamp	R_{ph}	Y_{ph}	B_{ph}	Event
16:09:00	OFF	OFF	OFF	Start of experiment
16:10:00	ON	OFF	OFF	R Phase Load ON
16:11:00	ON	ON	OFF	Y Phase Load ON
16:12:00	ON	ON	ON	B Phase Load ON
16:13:00	ON	ON	ON	All Phase in ON State
16:14:00	OFF	ON	ON	R Phase Load OFF
16:15:00	OFF	OFF	ON	Y Phase Load OFF
16:16:00	OFF	OFF	OFF	B Phase Load OFF
16:17:00	OFF	OFF	OFF	End of experiment

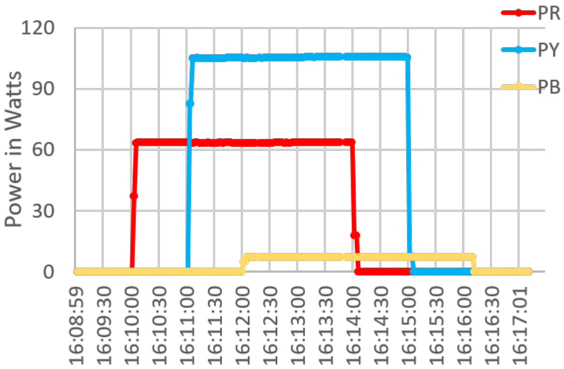

Fig. 7. Plots of Power in each phase

B. Case Studies - Validation

Lab hardware set up data extraction validation is done by switching on/off the loads in each of the phases at the time instants as shown in table. II. Fig. 7 shows the plot of power in each of the phases and the overall power in all the three phases.

Case 1 : Loads in all phases are turned off All the loads are kept in off mode. This can be seen in Fig. 7 as a total of zero power on all the phases.

Case 2 : Phase R load is turned on The 60 watt bulb load on R phase is turned on at the instant 16:10:00.This instant of real time can be seen in table III.In the data retrieved, the same is observed and this is shown as the plot of phase R power in Fig. 7. A spike in the graph is clearly seen at instant 16:10:00.

Case 3 : Phase Y load is turned on in addition to the phase R load The 100 watt bulb load on Y phase is turned on at the instant 16:11:00.This instant of real time can be seen in table III. In the data retrieved, the same is observed and this is shown as the plot of phase Y power in Fig. 7. A spike in the graph in phase Y is clearly seen at instant 16:11:00. The total power is addition of powers on phase R and phase Y. Small spikes seen between is the transition time from off to on condition of phase Y. similar spikes can be seen when other phases are turned on and off gradually.

Case 4 : Phase B load is turned on in addition to the phase

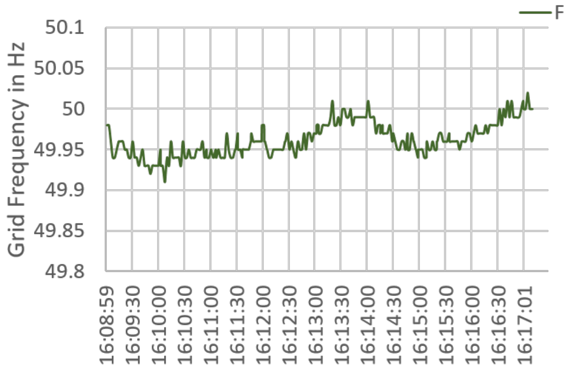

Fig. 8. Plots of frequency during the test

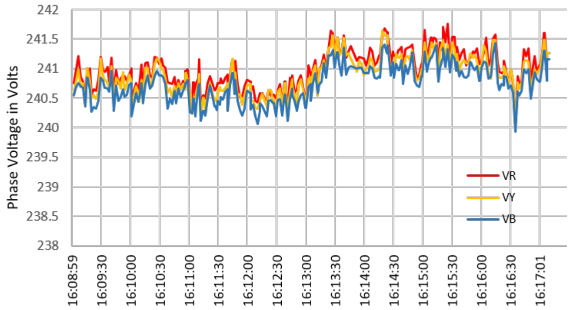

Fig. 9. Plots of voltages in each phase

R and Y loads The 8 watt bulb load on B phase is turned on at the instant 16:12:00.This instant of real time can be seen in table III.In the data retrieved, the same is observed and this is shown as the plot of phase B power in Fig. 7. A spike in the graph is clearly seen at instant 16:12:00. The total power is addition of powers on phase R,Y and B. Thus the total load in all the phases put together is 168 watts.

Case 5 : Phase R load is turned off leaving phases Y and B in turned on The phase R is turned off at the instant 16:14:00.This instant of real time can be seen in table III.The phase Y load of 100 watts and phase B load of 8 watts are in turned on state as they were previously.In the graph, we can see the fall in the graph from 60 watts to 0 watts at this instant of phase R power.

Case 6 : Phase Y is turned off leaving only phase B load turned on The phases Y is turned off at the instant 16:15:00.This instant of real time can be seen in table III. The phase B load of 8 watts is in turned on state as it was previously.In the graph, we can see the fall in the graph from 100 to 0 watts of phase Y power at this instant.

Case 7 : Loads in all phases are turned off At the instant 16:16:00,all loads are turned off.The phase B power in the graph also falls to zero.

Amongst all the parameters measured by the meter and thus extracted, the frequency of the grid supply is depicted in Fig. 8 and voltages of all the three phases is depicted in Fig. 9.

V. Conclusions and Future Work

The three phase voltage plot Fig. 9 shows that all the values are more or less equal and the data extraction is validated as it matches the voltages measured. It can be seen in the Fig. 7 that indeed the power measured on each phase and the total power matches with the expected load demand. Thus it is shown that the data extraction process implemented is matching the exact loading instants. Using the RS 485 to USB converter and Raspberry Pi as micro controller which are usually available in electrical labs has thus made the data extraction process implemented to be a easy approach solution. The data presently is being written to a file in the server location. In future, the data can also be written directly into the data base if needed. The data thus can be made available to the engineers through a common web portal. This web portal can have the data segregated according to time period(daily/weekly/monthly) for building energy monitoring purposes. The data analytics on this data can be done to prevent any electrical mishaps and also to improve the consumption efficiency.

Acknowledgement

The authors here by acknowledge the financial support provided by Department of Science and Technology (DST), Government of India towards India-UK “JUICE” research project no: DST/CHY/F018/2017-18/G69.

References

[1] J. King and C. Perry, *Smart Buildings: Using Smart Technology to Save Energy in Existing Buildings*. Amercian Council for an Energy-Efficient Economy, 2017.

[2] C. Bolchini, A. Geronazzo, and E. Quintarelli, “Smart buildings: A monitoring and data analysis methodological framework,” *Building and Environment*, vol. 121, pp. 93–105, 2017.

[3] L. Zhao, J.-l. Zhang, and R.-b. Liang, “Development of an energy monitoring system for large public buildings,” *Energy and Buildings*, vol. 66, pp. 41–48, 2013.

[4] K. Krishnanand, H. D. Chinh, M. Gupta, and S. Panda, “Electrical submetering with repurposable applications for the built environment,” in *Industrial Electronics Society, IECON 2016-42nd Annual Conference of the IEEE*. IEEE, 2016, pp. 3948–3953.

[5] A. I. Cole and A. Albicki, “Data extraction for effective non-intrusive identification of residential power loads,” in *Instrumentation and Measurement Technology Conference, 1998. IMTC/98. Conference Proceedings. IEEE*, vol. 2. IEEE, 1998, pp. 812–815.

[6] H. Ochiai, H. Nakagami, Y. Teranishi, and H. Esaki, “Facility networking with ip over rs485: Packet control for master-slave cascaded networks,” in *Smart Grid Communications (SmartGridComm), 2014 IEEE International Conference on*. IEEE, 2014, pp. 410–415.

[7] D. M. Strumpf and D. A. Drolet, “Communication module and process for networking within and between powered communication devices over a multi-phase power distribution system or subsystem,” Jan. 23 2007, uS Patent 7,167,081.

[8] T. J. Coombs, “Half-duplex communication system,” Feb. 7 2008, uS Patent App. 11/878,636.

[9] A. W. Blackett, B. J. Gilbert, and M. A. Hancock, “Method and system for master slave protocol communication in an intelligent electronic device,” Sep. 14 2004, uS Patent 6,792,337.

[10] *Minimal Modbus documentation*. [Online]. Available: http://minimalmodbus.readthedocs.io/en/master/

[11] *Paramiko Package documentation*. [Online]. Available: http://www.paramiko.org/