DESIGN AND IMPLEMENTATION OF SMART METER FOR SMART CITY

Ali M.Hasan [1], **Abdulkareem A.Kadhim** [2]

[1,2] College of Information Engineering, Al-Nahrain University, Baghdad, Iraq

{ali.maser, abdulkareem.a}@coie-nahrain.edu.iq [1,2]

Abstract- **Smart city deals with the developments of many services related to energy, environmental, and healthcare. It is also continuously providing new services to all citizens. This paper is concerned with the design and implementation of smart meter system as the core for smart grid in smart city. A system is proposed where the electricity supply is monitored by measuring its related parameters such as voltage, current, power, energy consumption, and consumption bill. Short Message Service (SMS) representing such parameters are issued by the system for notification and control. The designed system used PZEM- 004T, Arduino Mega, Raspberry Pi and Node-Red platforms to implement smart meter system. The data related to the measured parameters are successfully transmitted to the data center using Message Queuing Telemetry Transport (MQTT) protocol, stored in MySQL data base using core python program, and displayed on Node- Red platform. The test and verification of the system are performed using different scenarios showing that successful and accurate operation of the system components are achieved. The implemented system is successfully measured the related parameters (voltage, current, power and energy consumption) and these are sent to data center using MQTT protocol together with SMS containing the monthly bill to the subscribers.**

I. INTRODUCTION

Smart city is not new concept, where urban and city governments have been performing digital or automated solutions to provide improved services for years. Digitization has been emphasized what making services more effective (e.g. , traffic flow, meter reading, online tax, building energy management, court records, digital libraries, and utility payments). Smart city is more than the collection of digital and automation efforts, however; it is about those who work or reside in the environment. However, there are six characteristics to evaluate a city as smart, including smart mobility, smart economics, smart people, smart environment, smart governance and smart living [1]. In addition to these six characteristics, a smart city can be linked using Information Communication Technology (ICT). Smart grids are the backbone of the smart city by providing the integration of: small distributed energy resources in the urban network, increasing the customer's awareness, providing real- time optimization of energy flows at the urban level [2]. A simple smart grid system based on smart meter is introduced by [3] using PZEM- 004T sensor, ESP8266 and an Arduino Nano. The arrangement is used for the measurement of electrical energy within a week and monitoring the consumption of electrical energy in buildings. A design using PZEM-004T sensor also with non-current transformer, SD3004 chip for energy measurement, and microcontroller for measuring current, voltage, accumulated power consumption, and active power [4]. The retrieved data is then are sent to the server using Internet connection so that the users can get information about their energy consumption by accessing the server via web application. A system for reducing energy consumption by switching the energy device on and off at each place depending on the time table is proposed [5]. The system is also used for smart estimation pf power consumption. IoT based smart grid for power monitoring system that analyze and measure the voltage, current, energy consumption and active power is proposed in [6]. ThingSpeak, an IoT platform is used for analysis, storing data, and retrieving them from loads.

In above mentioned works, mainly PZEM- 004t is used as a standalone device to perform smart meter functions without producing electricity monthly bill or notifying the subscribers. These features are introduced in the present paper where the consumed energy is analyzed and its cost is calculated. The data related to the consumed energy for all subscriber are stored in data center and accessible via web application.

II. Smart Meter System

The smart meter system performs a complex set of operations and communications in order to provide a timely and accurate monitoring consumption together with control functions to ease the power managements in the building or any connected facility. The smart grid scenario used in the work consists of several connected smart meter nodes that communicate with a central server (data center) represented by a Raspberry Pi device through the Message Queuing Telemetry Transport (MQTT) protocol using a Global System for Mobile Communications (GSM) modem [7]. The diagram of the smart meter's arrangement is shown in Fig. 1. Each node can operate successfully as a standalone smart meter. In practice, the nodes communicate with the data center remotely using GSM network. The system is able to monitor the smart meter measurements locally with remote control of the power supply for each building. The application also provides an instant cost estimation feature that is based on the regulations of the Iraqi Ministry of Electricity that set different cost categories for each consumption group. In case of GSM network disconnection, the smart meter keeps running as standalone device regarding the storage of readings and control. In either case (stand alone or connected) the readings are displayed on the device Liquid Crystal Display (LCD).
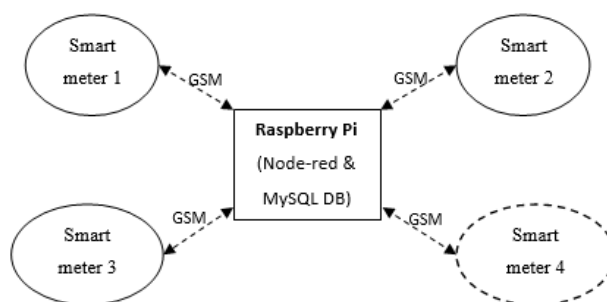

Figure 1: The diagram of smart meters arrangement

III. The Hardware and Software of the System

The proposed smart meter system consists of four smart meter nodes in order to build a wireless sensor network. Each smart meter node consists of Arduino Mega 2560 Rev3 as a kind of microcontroller board depending on ATmega2650 [8]. It has 54 digital pins for input and output among these14 can be used for Pulse Width Modulation (PWM), 16 analog inputs, Universal Serial Bus (USB) connection slot, 4 serial ports, reset button, power jack, and In Circuit Serial Programming (ICSP) header [8]. SIM808 EVB V3.2 GSM Sim Card Module is used for sending Short Message Service (SMS) notification to subscriber and making internet connection [9]. PZEM-004T Serial Multimeter which is predominantly

utilized for estimating AC voltage, current, active power, frequency, and power factor. There is no Graphical User Interface (GUI) in the module, the information is read out using the Time to live (TTL) interface [10]. Raspberry Pi 3 Model B+ is used as a mini- computer produced by the Raspberry Pi foundation. It is flaunting a 64- piece quad center processor running at 1.4GHz, double bands of 2.4GHz and 5GHz WiFi module, ethernet interface, and Bluetooth 4.2/BLE [11]. Data transmission between smart meter node and Raspberry Pi is performed by using MQTT protocol. The circuit diagram of the smart meter node is shown in Fig. 2. The programing software used in smart meter system consists of Arduino C used to write the core software for the smart meter node including Arduino Mega, PZEM- 004T Sensor and SIM808 GSM Module [12]. Python is used in the Raspberry Pi to handle the MQTT requests of the smart grid and to pass the data to the data base for storage [13]. JavaScript is used in Node- Red IoT Platform to program the GUI in the data center [14]. Node- Red IoT Platform is used to handle MQTT requests of the smart grid and to visualize the data on the screen for easy monitoring and management [15].
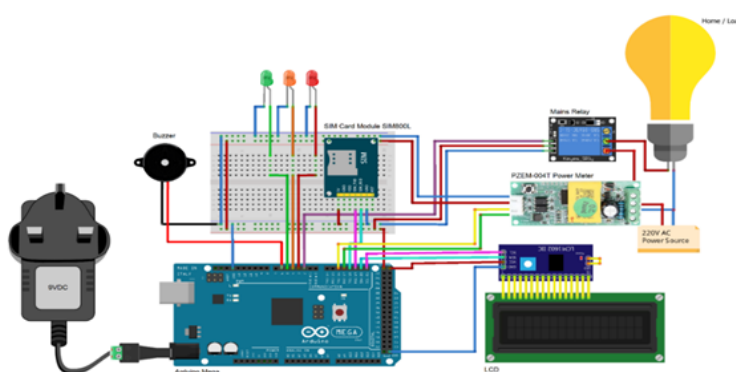

Figure 2: Circuit diagram of smart meter nod

IV. Installation and Operation of the Smart Meter System

A. The measured parameters

The PZEM- 004T multimeter module is solely responsible for all parameter's measurements (voltage, current, power consumption, energy consumption, frequency, and power factor) in a smart meter node [10]. The PZEM module is connected to the Arduino Mega 2560 using the Universal Asynchronous Receiver- Transmitter (UART) serial communication port (Tx/Rx) from one side, and to the load line from the other side. The neutral and line are connected directly to the PZEM module in order to provide the voltage measurement. The current transformer is looped through the load line in order to provide current measurement. The PZEM module internally measure and computes the mentioned parameters. These are output as a string to the Arduino via the UART port. Fig. 3 clarifies the connection pins of PZEM- 004T multimeter module. After reading and measuring the parameters, data acquisition is applied to display these parameters on the LCD and to transmit them to the system data center using MQTT protocol.
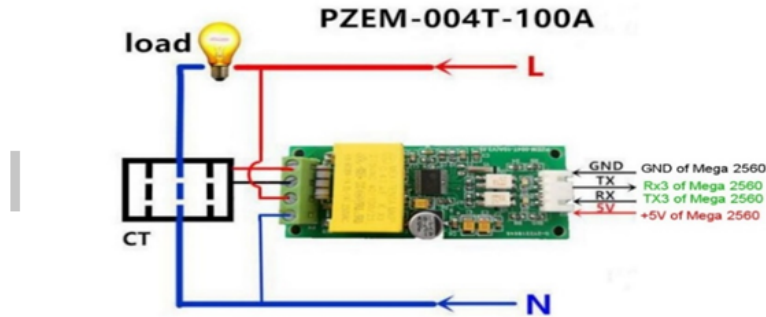
Figure 3: Connection pins of PZEM- 004T multimeter module [10]

B. Voltage protection

An internal voltage protection algorithm is implemented in the smart meter node to provide protection against undervoltage and overvoltage. The smart meter continuously measures the voltage in order to protect the system and its connected devices. The undervoltage is set to be below 180V and overvoltage is set to be above 240V. If the voltage reading is out of such limits, the main relay will be turned OFF until acceptable voltage level is restored. Current limiting feature is also included in the smart grid system design in order to allow the operator to manually increase or decrease the current limit. The current limit can also be changed remotely by the administrator. The smart meter continuously monitors the load current. When the current exceeds the pre-set current limit, the main relay will be turned OFF for 10 seconds and then turned ON to check again whether the current passed the limit or not.

C. Consumption cost estimation

Energy consumption cost estimation is very useful feature in order to publish the electricity monthly bill. Consider P as the AC power in Watts and E_T as the total consumed energy in kilo- Watt-hour (kWh) as given in the following equations:

$$P = V.I.PF \tag{1}$$

V is the voltage reading in Volts, I is the current reading in Amperes, and PF is the power factor. Considering T as the time interval of the power consumption, then:

$$E_T = P.T/1000 \tag{2}$$

Using the energy limits in Table I, then the consumed energy in each category is given by Equation 3. Only the positive values of are considered in the subtractions involved this equation.

$$
\begin{aligned}
E_{c4} &= E_T - 4000 & (3.a)\\
E_{c3} &= E_T - E_{c4} - 3000 & (3.b)\\
E_{c2} &= E_T - E_{c4} - E_{c3} - 1500 & (3.c)\\
E_{c1} &= E_T - E_{c4} - E_{c3} - E_{c2} & (3.d)
\end{aligned}
\tag{3}
$$

Using C_i of the given unit category as given by Table I, the total cost of consumed energy C_T, in Iraqi Dinars, is given by:

$$C_T = \sum_{n=1}^{4} C_i.E_{ci} \tag{4}$$

D. Data handling

The smart meters in the smart grid system are Internet- enabled via SIM808 GSM Module that provides a GSM cellular network connectivity. The smart meter transmits the measured data to the Raspberry Pi using the MQTT protocol. The Python script in the Raspberry Pi subscribes to the MQTT channel of the smart meter and receives the data published by the smart meter nodes via MQTT protocol. The measured data is then stored in MySQL data base to provide reference for data history and to help the administrator to perform data analysis and reports generation. At the same time, the Node- Red GUI is already subscribed to the same MQTT channel thus received the same data published by the smart meter nodes. The data is represented and viewed as graphical elements in the smart meter GUI. Fig. 4.A and Fig. 4.B show the data handling process within the smart meter application.

E. Notification of monthly bill

At the end of each month, the system will send an SMS to the subscriber via mobile phone containing the monthly bill to remind him of the payment date and the amount due in Iraqi Dinars. Further, the system administrator can turn ON or OFF the electricity supplied to a specific subscriber manually by toggling the ON/ OFF button on the Node- Red smart meter GUI. This will send a command to the intended smart meter and toggle the mains relay state, thus turning ON or OFF the power supply to the given building.

TABLE I
TARIFF CATEGORIES OF CONSUMMED ENERGY IN IRAQ

Category	Consumed Energy E (kWh)	Unit Cost C_i (IQD)
1	$E \leq 1500$	10
2	$1500 < E \leq 3000$	35
3	$3000 < E \leq 4000$	80
4	$E < 4000$	120

V. RESULTS AND VERIFICATION OF THE SYSTEM

A. Calibration of smart meter

The PZEM- 004T V3.0 multimeter module is first calibrated by comparing its readings to an external hand- held multimeter. The results show that the relative percentage error of one percent (1%) in the measured parameters is occurred.

B. Standalone single node tests

Standalone operation of each node is then checked by considering household devices with different load conditions being connected to the electricity mains for each smart meter node to set a load scenario for system testing. The voltage, current, active power, and power factor for each device are recorded to provide data for the purpose of later analysis as arranged in Table II.

TABLE II
THE LOG FOR VARIOUS HOUSEHOLD DEVICES

Device	Voltage (V)	Current (A)	Active Power (W)	Power Factor
Ceiling fan	216.10	0.99	188.27	0.86
Laptop charger	216.10	0.27	59.20	0.89
Uninterrupted power supply (UPS) (no load)	222.40	0.08	8.70	0.51
LDC TV / monitor	221.70	0.38	75.70	0.91
Air cooler ♯ 1	216.30	1.17	213.82	0.85
Air cooler ♯ 2	214.70	0.74	145.40	0.92
Blender (mixer)	215.70	1.57	332.42	0.99
LED light (one unit or 1x)	222.90	0.18	26.50	0.66
Refrigerator	216.40	1.31	220.54	0.80
Oven (double heaters)	212.20	6.43	1365.61	1.00

To consider the calculation of the consumed energy in the smart meter node, another test scenario is considered to measure the consumed energy for different loads over assumed time intervals. The scenario also allowed for later analysis to cover the electricity consumption and the production of the electricity bill. Table III shows the load conditions considered in the scenario. The calculated parameters by the smart meter for the different load conditions of Table III are shown in Table IV. The last column of Table IV is determined according to section IV-C.

TABLE III
DETAILED OF HOUSEHOLD DEVICES LOAD SCENARIO

Time Interval	Ceiling fan	Laptop charger	Uninterrupted power supply (UPS) (no load)	LDC TV / monitor	Air cooler ♯ 1	Air cooler ♯ 2	Blender (mixer)	LED light *	Refrigerator	Oven (double heaters)
01:00 to 01:05 PM	ON	ON	ON	OFF	ON	ON	OFF	1x	ON	OFF
01:05 to 01:10 PM	ON	ON	ON	OFF	ON	ON	OFF	2x	ON	OFF
01:10 to 01:15 PM	ON	ON	ON	ON	OFF	ON	OFF	3x	ON	OFF
01:15 to 01:20 PM	OFF	ON	ON	ON	OFF	ON	OFF	3x	ON	OFF
01:20 to 01:25 PM	OFF	ON	ON	ON	OFF	ON	OFF	3x	ON	ON
01:25 to 01:30 PM	OFF	ON	ON	ON	OFF	ON	ON	3x	ON	ON
01:30 to 01:35 PM	OFF	ON	ON	ON	OFF	ON	OFF	4x	ON	OFF

∗ LED light is always on with the number preceding x represents the number of lighted elements

TABLE IV
MEASURED PARAMETERS FOR THE LOAD SCENARIO IN TABLE III

Time Interval	Voltage (V)	Current (A)	Power (W)	Frequency (Hz)	Power Factor	Energy (kWh)
01:00 to 01:05 PM	212.90	3.17	647.50	50.50	0.96	0.05
01:05 to 01:10 PM	213.70	3.22	654.60	50.60	0.95	0.11
01:10 to 01:15 PM	213.20	2.32	471.00	50.40	0.95	0.15
01:15 to 01:20 PM	209.50	2.18	426.00	50.10	0.93	0.18
01:20 to 01:25 PM	20.70	8.38	1740.50	50.10	1.00	0.33
01:25 to 01:30 PM	214.90	10.27	2175.90	50.20	0.99	0.51
01:30 to 01:35 PM	215.80	2.38	462.80	50.20	0.91	0.55

C. Connected smart meter nodes test

After testing each node separately, test of another scenario considering connected smart meter nodes is performed. In this test, the four smart meter devices are connected to different loads to demonstrate the basic smart meter functionality while being connected to the Internet via GSM data connection. Fig. 5 shows that the load was connected to the smart meter node ♯1.
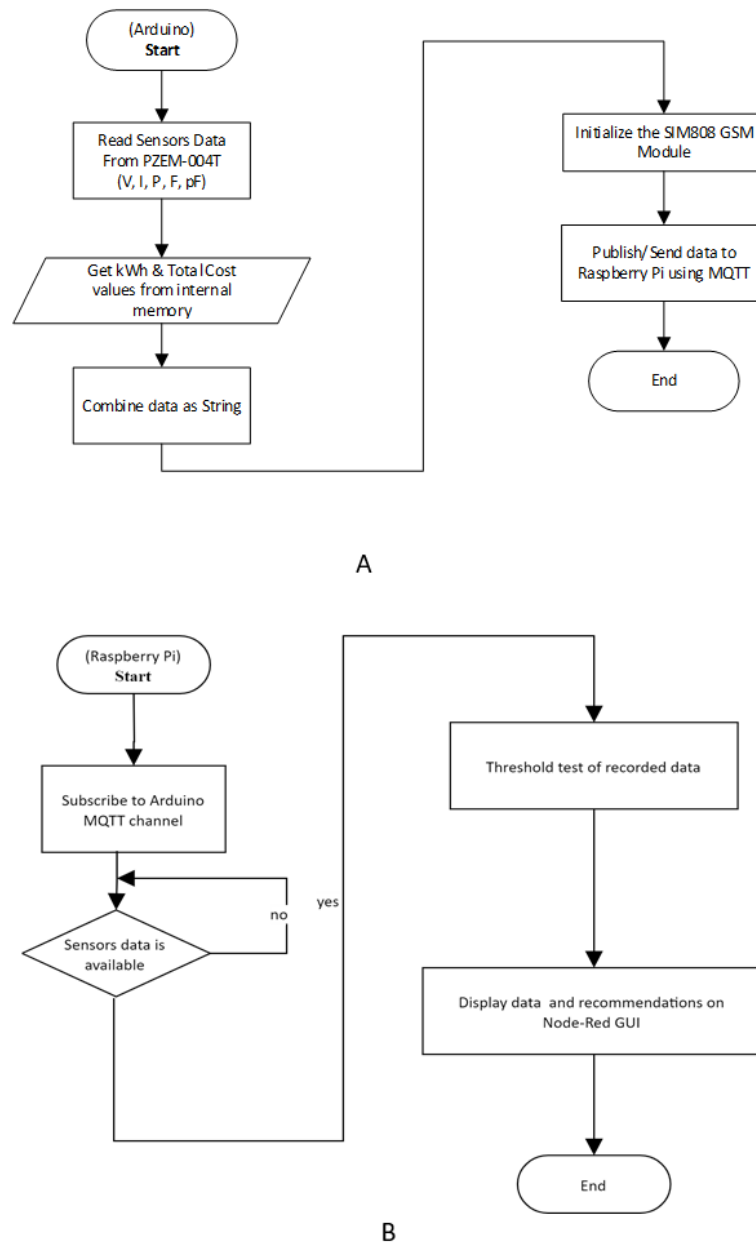
Figure 4: Flow charts for data handling for: smart meter node side (A) and data center side (B)

Figure 5: Node-Red GUI for smart meter node♯1

D. Logging of power measurements into MySQL database

While the smart meters are connected to the Internet and communicate with the smart grid data center, a core Python program specifically built for data logging is running in the background. This provides subscribing and listening to the MQTT channel for each smart meter node. The data received from MQTT channels are stored into MySQL data base for logging purpose and future analysis such as viewing the power consumption on hourly, daily, weekly, monthly or yearly basis for specific node. Fig. 6 shows the Python program running while power measurement data is being received from smart meter♯1 via MQTT connection. At the same time, the Python program exports the power measurements data to the smart grid MySQL data base to be stored.
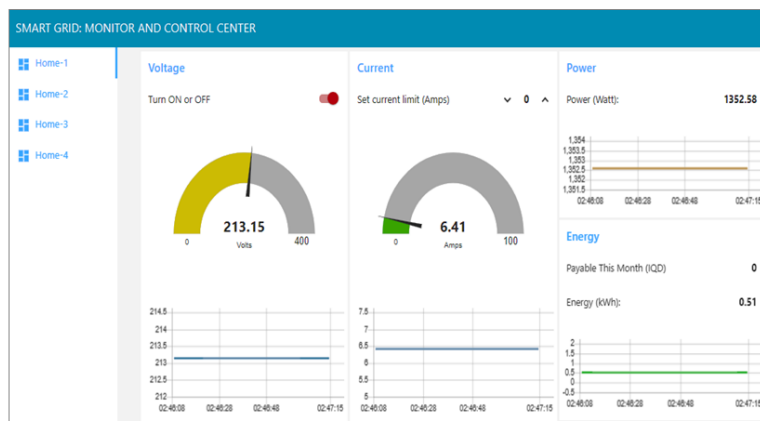

Figure 6: Python program for smart grid data center and MySQL data logging

E. Cost calculation and SMS notification

The last scenario is considered to demonstrate the production of the monthly bill. Assuming a consumed energy of 3000 kWh for certain month that have 30 days. Since it is unpractical to run the smart meter for a long time until reaching an adequate energy value, the smart meter of node ♯1 is programmed to transmit a total energy value of 3000 kWh for

30 days interval to the smart grid data center. The following operations are then automatically performed: Node-Red GUI calculates the cost of energy according to section IV-C as shown in Fig. 7, where the cost for 3000 kWh is calculated as 67,500 Iraqi Dinars. The Node-Red sends an SMS request command to the smart meter node ♯1 containing the total cost. The smart meter responding by combining an automated SMS and sending it to the subscriber mobile phone whose number is already stored in the smart meter's microcontroller. SMS is then received from the smart meter containing the required information for the bill as show in Fig. 8.
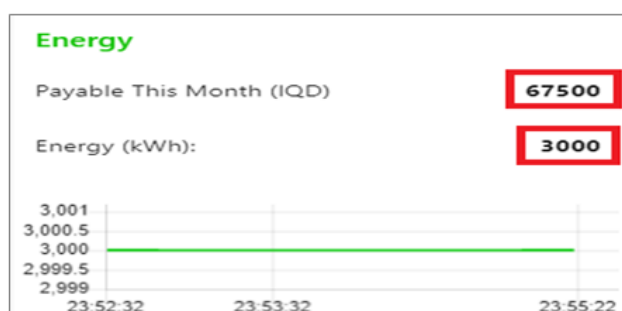

Figure 7: Node- Red GUI - Energy and Cost View for Smart Meter Node ♯1
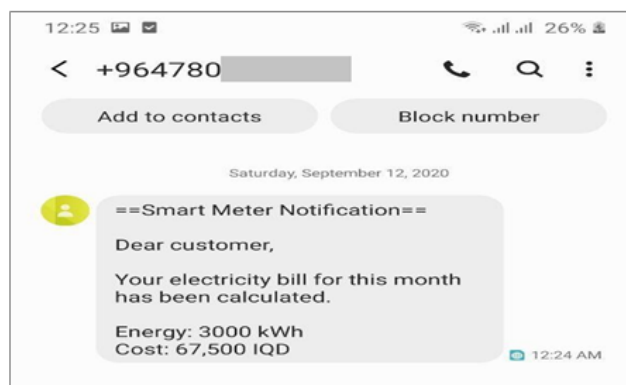

Figure 8: SMS notification for monthly bill

F. Verification of the system

The tests involved in the smart grid system produced results for the different scenarios considered above as expected according to the system features set by the node specifications. The readings of the parameters are highly accurate for both the standalone and connected nodes operations. The GSM connectivity to the Internet and the related hardware and software implementations are successfully operated and verified according to its designed features. The play of smart meter data in Node-Red GUI and their transfer to the data base are correctly accomplished for all nodes.

VI. Conclusions

A complete design of smart grid based on smart meter system is introduced. The system consists of three main parts: the smart meter, MQTT protocol, and data center. The smart meter components are Arduino Mega, PEZM- 004T, LED, and GSM module. MQTT protocol is used to send data from smart meter to data center and vice versa. The implemented system performed accurate measures of electricity supply and related parameters such as voltage, current, power, power factor, energy consumption in kWh. The latter is used to produce electricity monthly bill. Voltage and current protection also included in the design. The measured parameters are sent from smart meter to data center using MQTT, while the monthly bill is sent from data center to smart meter using MQTT. The designed system can be extended to cover large geographical areas and can be modified to serve for pre- paid arrangement in an effort to assist in reducing the electricity consumption in Iraq with the present continuing crises of electricity supply.

References

[1] D Novita and E Suryani, "Smart City on Public Perception" , IOP Conference Series: Earth and Environmental Science, Vol. 340, Conference- 1, Indonesia, 2019.

[2] F. Gasparin, "Smart Grid Systems" , Ver. 2, Netherlands, 2013, https://eu-smartcities.eu/sites/default/files/2017-10/Smart%20Grid%20Systems %20-%20Smart%20Cities%20Stakeholder%20Platform%20%281%29-0.pdf.

[3] S. Wasoontarajaroen, K. Pawasan and V. Chamnanphrai, "Development of an IoT device for monitoring electrical energy consumption" , 2017 9th International Conference on Information Technology and Electrical Engineering (ICITEE), Phuket, Thailand, 2017.

[4] K. Chooruang and K. Meekul, "Design of an IoT Energy Monitoring System" , 2018 16th International Conference on ICT and Knowledge Engineering (ICT& KE), Bangkok, Thailand, 2018.

[5] A. Frank, R. Asuncion and M. Frank, "Smart Optimization of Energy Consumption Using IoT", 2019 2nd IEEE Middle East and North Africa COMMunications Conference (MENACOMM), Manama, Bahrain, 2019.

[6] F. Khan, M. Siddiqui, A. Rehman, J. Khan, M. Sadiq, and A. Asad, "IoT Based Power Monitoring System for Smart Grid Applications" , 2020 International Conference on Engineering and Emerging Technologies (ICEET), Lahore, Pakistan, 2020.

[7] C Talamo, M R Pinto, S Viola and N Atta, "Smart Cities and Enabling Technologies: Influences on Urban Facility Management Services" , IOP Conference Series: Earth and Environmental Science, Vol. 296, Milan, Italy, 2019.

[8] Arduino Co. , "Arduino Mega 2560" , rev3, Last visited:7/9/2020, https://store.arduino.cc/usa/mega-2560-r3.

[9] Data Sheet, "Usage of Sim808 GPRS GSM GPS Shield" , SimCom, Ver1, USA, 2018, https://www.robotshop.com/media/files/pdf/sim808-gps-gprsgsm-arduino-shield-mkf-datasheet.pdf.

[10] Data Sheet, "PZEM-004T v3.0 user manual" , Innovators Guru Co. , Ver. 3, 2019, https://innovatorsguru.com/wp-content/uploads/2019/ 06/PZEM-004T-V3.0-Datasheet-User-Manual.pdf.

[11] Data Sheet, "Raspberry Pi 3 Model B+", Raspberry Pi Foundation, Ver. 3, 2018, https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf.

[12] J. Bayle, "C programming for Arduino" , 1st ED, Packt publishing, USA, 2013.

[13] R. Barnes, "Raspberry Pi Projects Book" , Liz Upton, Ver. 1, England, 2015.

[14] M. Blackstock and R. Lea, "Toward a Distributed Data Flow Platform for the Web of Things" , IEEE IoT conference: Developing IoT Applications in the Fog: a Distributed Dataflow Approach, South Korea, 2015.

[15] IBM Services, "Node- red" , Last visited:7/8/2020, https://nodered.org/docs/user-guide/.