Design and Implementation of an Internet of Things Based Smart Energy Metering

Mohammad Hossein Yaghmaee
Ferdowsi University of Mashhad (FUM)
Mashhad, Iran
e-mail: hyaghmae@um.ac.ir

Hossein Hejazi
Mashhad Electrical Energy Distribution Company (MEEDC),
Mashhad, Iran

***Abstract*—Internet of Things (IoT) can be used to furnish intelligent management of energy distribution and consumption in heterogeneous circumstances. In the recent years, by the growth of IoT and digital technologies, smart grid has been becoming smarter than before. The future power grid needs to be implemented in a distributed topology that can dynamically absorb different energy sources. IoT can be utilized for various applications of the smart grid including distributed power plant monitoring, power generation and consumption prediction, power consumption monitoring, energy storage monitoring, smart meter, electric vehicle charging, power demand side management and various area of energy production. In this paper, by using the IoT capabilities, we have designed and implemented a smart energy metering platform consisting of smart plugs, gateway and cloud server.**

Keywords-Smart Grid, Energy Monitoring, Internet of Things, Internet of Energy

I. INTRODUCTION

In order to develop and implement smart power grid applications in a wide range of countries, the need to design smart energy metering infrastructure is well understood. The utility companies in this area can use the results and data measured by this system to make decisions and move towards widespread implantation of smart grid in the country. Smart buildings are a major element in the optimal use of energy and sustainability. The need to implement smart homes from the point of view of optimal energy consumption and reducing the cost of consuming electricity by subscribers is an important issue. On the other hand, due to the extensive Internet of Things (IoT) technology and the development of various network connectivity sensors, it is possible to send instantaneous power consumption data to the customers. To this end, smart hardware systems must first be designed and built. After that, home network system will be designed and implemented. The home gateway connects to the home sensor nodes and receives their information, and then connects to the Internet servers and transfers the collected information to the remote server. Maintenance, refinement and processing of data is done in the network server that maintains the user information in the required data storage. The system should be designed to meet the existing standards. Advanced Metering Infrastructure (AMI) utilizes the smart grid communication infrastructure to transfer metering data as well as customer consumption-related information [1], [2]. It should be noted that the AMI can only send the total electricity consumed by the customers at a time interval of between 15-60 minutes. Demand Response (DR) is one of the most important applications of smart grid [3-5]. Since we need the customer's power consumption information to develop and implement an efficient demand response program, with the proper implementation of this kind of system, the utility companies can benefit this information to develop proper demand response programs. To balance total demand with the amount of supply, demand response programs can be utilized. For efficient demand response program, the energy consumption and generation information should be tracked in real time. To achieve this goal, we need to deploy more remote sensing equipment capable of measuring, monitoring and communicating. As described in [6], the IoT can be used to furnish intelligent management of energy distribution and consumption in heterogeneous circumstances. In the recent years, by the growth of IoT and digital technologies, smart grid has been becoming smarter than before. The future power grid needs to be implemented in a distributed topology that can dynamically absorb different energy sources.

In this paper, we introduce the Smart Energy Metering (SEM). The main purpose of SEM is to create the necessary infrastructure for collecting information on energy consumption of household appliances and monitor the environmental parameters and provide the necessary services to home users. Using the proposed SEM, the following capabilities will be realized:

- Monitoring of the instantaneous power consumption of each home appliances connected to the system and calculating the power consumption, line voltage and power factor of each home appliances connected to the system.
- Providing timely information to the customers about their current power consumption and the cost of consuming electricity up to now.
- Sending useful information to the customers about their power consumption compared to other customers registered in the system. Each customer continuously observes its power consumption, the maximum, minimum, and average power consumption of other customers in the system.
- Processing the power consumption data to provide useful information for both customers and the utility companies.
- Better forecasts of necessary energy production and energy consumption in each specific area in the grid.
- Continuous monitoring of the power grid.

- Interacting with customers and managing consumption across the network.

II. Smart Energy Monitoring (SEM)

In this section, we introduce the proposed Smart Energy Monitoring (SEM) in details. The proposed system can be used in residential and industrial centers to measure the amount of energy consumed by each electrical device and to apply various controls on electrical appliances. The system can measure power consumption and power line parameters and send them to a central server in a different way. The data obtained can be used to predict customer consumption and consumption schedules. The system can control the electrical equipment and, if necessary, turn it off in the peak hours and turn it on at non-peak hours. The system is also able to monitor and store other environmental parameters including temperature, humidity, brightness and possible gas leakage. It is possible to design and implement various smart applications on this infrastructure. If the utility companies move forward to dynamic and instantaneous pricing and implementation of demand-side management services, this system can be well-designed to instantly inform users of their consumption and the current price in the network.

As shown in Fig. 1, the proposed SEM system consists of three major components including sensor nodes, Gateway and Server. The sensor node sends its information to the Gateway, and the Gateway connects to the Internet through a communication technology such as an ADSL modem or 3G/4G/LTE network. Each component is introduced in the following section in details.

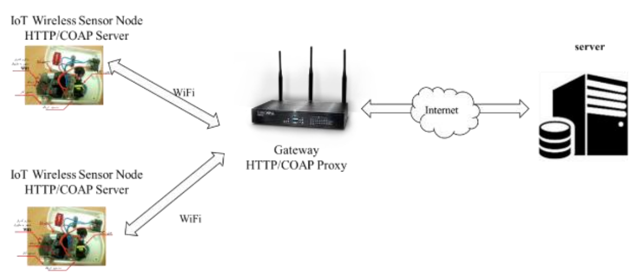
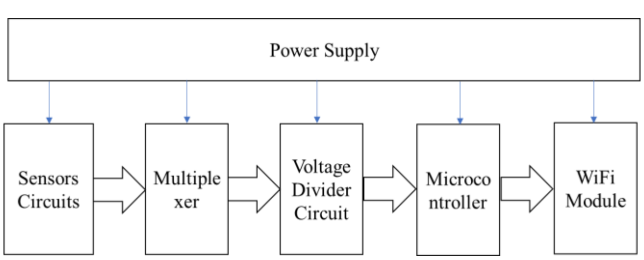


Figure 1. The proposed smart energy metering system

A. Sensor Node

To collect the customer's power consumption information, an IoT based smart plug has been implemented. The hardware is able to collect the other environmental parameters such as temperature/humidity, gas and light sensor.



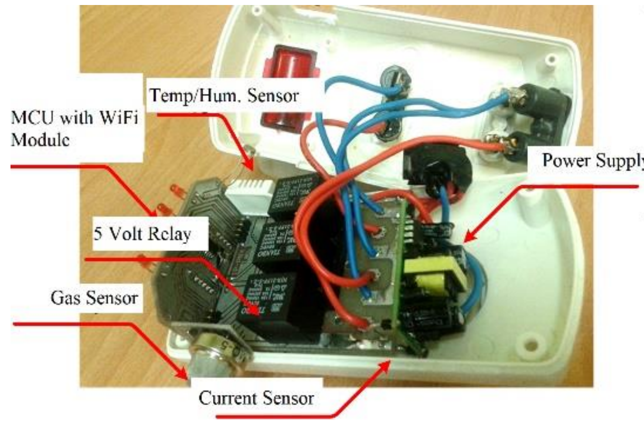


Figure 2. (a) The structure of sensor node (b) a sample sensor node

As shown in Fig. 2, the sensor node hardware consists of the different parts including Current sensors, Voltage sensor, Temperature and humidity sensor, Brightness sensor, Gas leak sensor, Relay, Microcontroller unit and WiFi module. In addition to the HTTP, for efficient data gathering and transmission, the COAP [7] protocol over WiFi network which is a lightweight protocol and follows the REST has been implemented.

B. Gateway

We have developed an Android HTTP Gateway. The Gateway is responsible for config the sensor nodes and collects the sensor values by using the standard API RESTful commands. The collected sensor data is transferred to the specific server for permanent storage. Each Customer has a unique API key which is used for communication between the Gateway and Server. This API key is used for providing required security and authentication process. Each time which Gateway wants to establish a connection with the Server uses this API key in the messages.

The sensor node is accessible through IP address 192.168.4.1 in the Access Point (AP) mode. In the beginning, the Gateway connects to the AP and programs it. By entering the network SSID and password, the sensor node can connect to the Gateway through the local WiFi network. At the end of this process, the sensor node is connected to the local WiFi network. A spatial web service is designed in the Gateway so that the sensor nodes are able to transfer their information after they connected to the WiFi network

In the implementation of this web service, the service feature of Android has been used. The service is one of the components of the Android apps that can run long behind the scenes, so the web server is constantly running as a service on the Android operating system, and if it stops, Android will automatically start it. We used the NanoHTTPD library to implement the web server in the Android Gateway. With this famous library, you can easily get HTTP messages and extract the information in the body and send the response as an http message to the client if necessary. NanoHTTPD is a light-weight HTTP server designed for embedding in other applications, released under a Modified BSD license. It can be used as a library component in developing other software.

When the Gateway receives sensor node's IP address, it can read the data of each sensor node.

The overall algorithm for the Gateway function is as follows:

1. Connect to the Sensor node AP on 192.168.4.1
2. Choose name for sensor node and send name, IP address of gateway, SSID and password of local network to the sensor node.
3. Receive the sensor node IP address from the POST message and update the database.
4. Sensor node initiates a web server and sends its IP address to the gateway
5. Read sensor node data
6. Collect data from all sensor nodes and send them to the server in a single message
7. Sleep for T seconds and go to step 4

When the Gateway sends the node's data to the server, the server sends customer power consumption, minimum power consumption, maximum power consumption, average power consumption, customer cost, customer rank, and power grid state*,* in response to the gateway. This information is shown on the home gateway screen which provides useful information on the user's power consumption and his/her status in the community. The Gateway can also show this information using a chart on its screen. In this chart, user can see the trends of the power consumption and the minimum, maximum and average power consumption of the community. To implement this graph we use the *androidplot* library. These libraries can be used to plot dynamic or static charts in the gateway screen. The implemented Gateway has the ability to manage and update the names and categories of all nodes connected to it. According to the categories, it is possible to view and analyze the consumption of each electrical appliances on the server side which will be explained in the next subsection.

C. Server

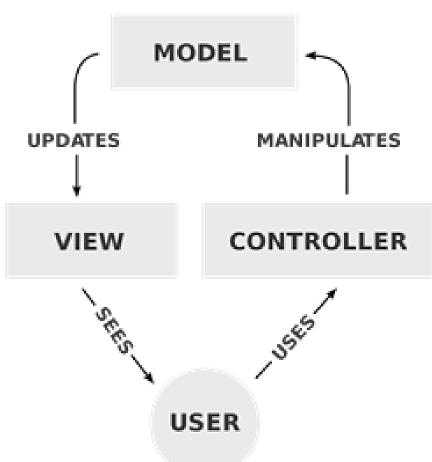


Figure 3. Model view controller architecture

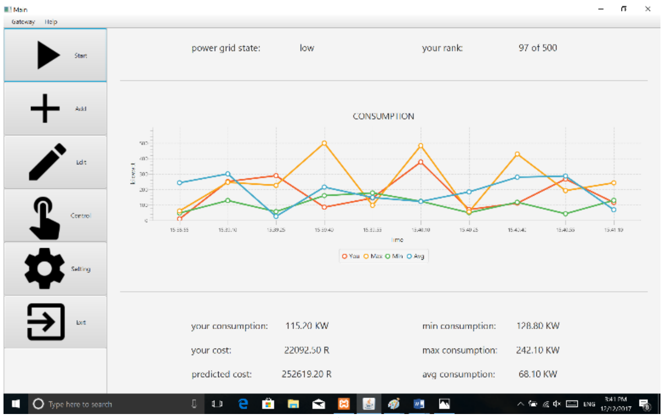
The main task of the Server is to communicate with the Gateways to capture sensor nodes information as well as to send control commands and related data to the Gateways. It also provides a user portal for instant access to the Gateway control commands. To develop the Server, the PHP programming language on the Laravel framework has been used. The Server uses SQL and the MySQL database and uses the Eloquent and ORM (Object Relation Mapping) technology to communicate with the database. For the panel's design the HTML, CSS, Javascript languages, and Bootstrap library on the AdminLTE framework have been used. The HTTP protocol and JSON data structure are used to transfer data between the Server and the Gateway. To secure this connection, each user has an own API-Token that can be generated on its panel and, if disclosed, can quickly remove it from its panel and build a new token. This token is used to communicate between the Server and the gateway for authentication. As shown in Fig. 3, the MVC (Model View Controller) architecture has been used.

The Server to Gateway communication is based on RESTfull architecture. Gateway collects environment information from sensors and sends this information to the Server every T seconds. The Server then responds to Gateway by sending some community statistics data such as maximum consumption, minimum consumption, average consumption, and user consumption and user cost. The Server also sends the commands that the user has configured on their panel for the Gateway, and Gateway implements these commands in the sensor nodes.

III. IMPLEMENTATION RESULTS

To collect the customer's home information, an IoT board containing DHT22 temperature/humidity sensor, MQ-5 gas sensor, light sensor, current sensor, 5 V relay and an MCU/WiFi controller has been implemented. We have developed the home gateway on a Raspberry Pi 3 with the Raspbian operating system using JavaFX platform. An Android gateway has also been developed. We have also implemented an SCT current sensor to measure the total power consumption of the customers. The proposed architecture can be considered as a backup system for the AMI network. Furthermore, as the energy consumption and the other parameters are collected by the system it is possible to developed and implement many smart grid applications such as AMI, Demand Response and Energy Management System and privacy/security applications. Fig. 4 shows the smart plugs and a sample output of the JavaFX gateway. The system was implemented in some customers home in Mashhad city. The collected data of a specific customers is shown in Fig. 5.

(a)



(b)

Figure 4. The system components (a) IoT based smart plug (b) Screenshot of the customer gateway

Figure 5. An example of data collected from the system

IV. CONCLUSION

In this paper, we proposed the smart energy monitoring which can measure power consumption and power line parameters and send them to a central server on the Internet through an intermediate Gateway. The system can control the electrical appliances and, if necessary, turn them off in the peak hours and turn them on at non-peak hours. Some of the benefits of the proposed system are as follows:

- Full design in accordance with the IoT standards and protocols.
- Ability to function both in Web Server/ Web Client
- Supports HTTP, COAP, MQTT and various data formats including JSON, HTML
- Ability to collect power consumption information and environmental parameters
- Ability to support and implement demand response programs.
- Support for various programming platform including Android, Linux, and Java.
- Reducing the cost of electricity for subscribers and reducing the amount of load at peak hours of the electricity grid.

Provide instant information to users about their power consumption and power consumption data of other users defined in the system.

REFERENCES

[1] J. Gao, Y. Xiao, J. Liu, W. Liang, and C. P. Chen, “A survey of communication/networking in smart grids,” Future Generation Computer Systems, vol. 28, no. 2, pp. 391–404, 2012.

[2] V. C. Gungor, D. Sahin, and e. Kocak, “A survey on smart grid potential applications and communication requirements,” IEEE Transactions on Industrial Informatics, vol. 9, no. 1, pp. 28–42, 2013.

[3] A. H. Mohsenian-Rad, V. W. Wong, J. Jatskevich, R. Schober, and A. Leon-Garcia, "Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid," IEEE Trans. Smart Grid, vol. 1, no. 3, pp. 320–331, 2010.

[4] I. Atzeni, L.G. Ordóñez, and G. Scutari,"Demand-Side Management via Distributed Energy Generation and Storage Optimization," IEEE Trans. Smart Grid, vol. 4, no. 2, pp.866-876, 2013.

[5] S. Maharjan, Q. Zhu, Y. Zhang, S. Gjessing, and T. Basar,"Dependable demand response management in the smart grid: A Stackelberg game approach," IEEE Trans. Smart Grid, vol. 4, no. 1, pp. 120–132, 2013.

[6] N. Ruiz, , I. Cobelo, and J. Oyarzabal, "A direct load control model for virtual power plant management," IEEE Trans. Power System, vol. 24, no. 2, pp. 959–966, 2009.

[7] Z. Shelby, K. Hartke, C. Bormann, “The Constrained Application Protocol (CoAP)”, Internet Engineering Task Force (IETF)Request for Comments: 7252, June 2014