Home Energy Monitoring System based on Open Source Software and Hardware

Evtim Peytchev, Mihail Lyaskov, Kostadin Popovski, Grisha Spasov

Abstract: *The current paper presents the development and realisation of a home based energy monitoring system that provides detailed information about the electric energy consumption of home appliances. The system is based on WIFI connection and LAN (Local Area Network) preconfigured in the house. It consists of wireless energy monitoring nodes and a Linux based single-board computer, used as a server, which can be accessed from the Internet. The main purpose for the development of this system is to deliver a low-cost and efficient open source solution to determine exactly how is electric energy used in our homes. This work has been carried out under the auspices of the REMOURBAN EU H2020 project.*

Key words: *Energy monitoring, single-board computer, open source, energy efficiency, Smart home.*

INTRODUCTION

The residential, tertiary, and commercial building sector is responsible for over 50% of the electricity consumption in Europe [1]. Homes and working environments are now isolated, energy-consuming units with poor energy efficiency and sustainability. Based on the SH (Smart Home) concept, these units can be transformed into intelligent network connected nodes where a significant part of the energy is locally produced by renewables (typically photovoltaic generators) and the whole, i.e., the generator and the loads, is intelligently managed. This is achievable by performing Smart Metering and using Smart Meters with high Hardware/Software capabilities to run transmission control protocol (TCP) internet protocol (IP) suite and applications on top of TCP or UDP, which enable remote connect/disconnect, real-time pricing, power-quality measurement, load management, and outage notifications. The idea of Smart Home with Smart Metering as presented in [1] is shown on *Figure 1*.

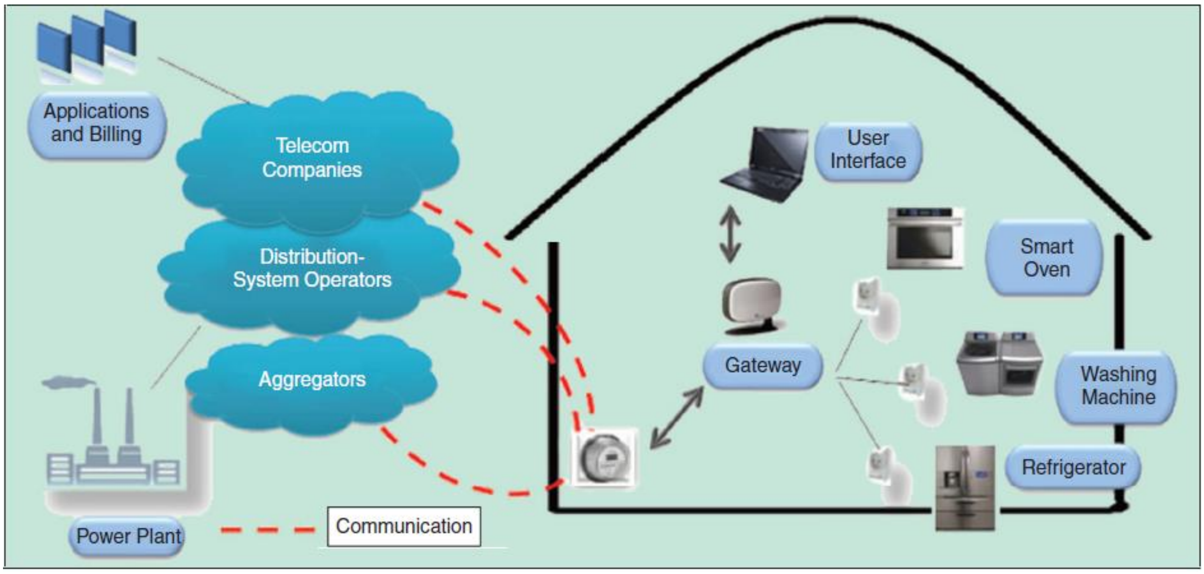

Figure 1. Smart Home with Smart Metering [1].

The employment of Home energy management systems (HEM) [2] in a residential area reduces energy bills for consumers and peak demand. With a normal demand in peak hours the utilities are able to provide power from base plants and hence contribution of Green House Gases (GHG) is less towards environmental pollution.

The main purpose of this paper is to present the realization of Home Energy Monitoring System based on Open Source Software and Hardware which in a later stage of development will represent a fully capable Home Automation Network (HAN) and will be part of the Smart Grid Metering infrastructure. [3]

SYSTEM OVERVIEW

The proposed system shown on *Figure 2* aims to deliver an easy and flexible way to monitor home appliances power consumption through time. This is achieved by connecting sensor devices (current and voltage sensors) to the sockets where energy consumers are connected and send regular data about the energy consumption via WIFI home network to the server, located in the house. After storing the data, devices like smart phones, tablets and personal computers, can access this database through the server by sending HTTP requests and a token for authentication. The server will also be available outside of the home network, after setting up the port forwarding on the gateway router.

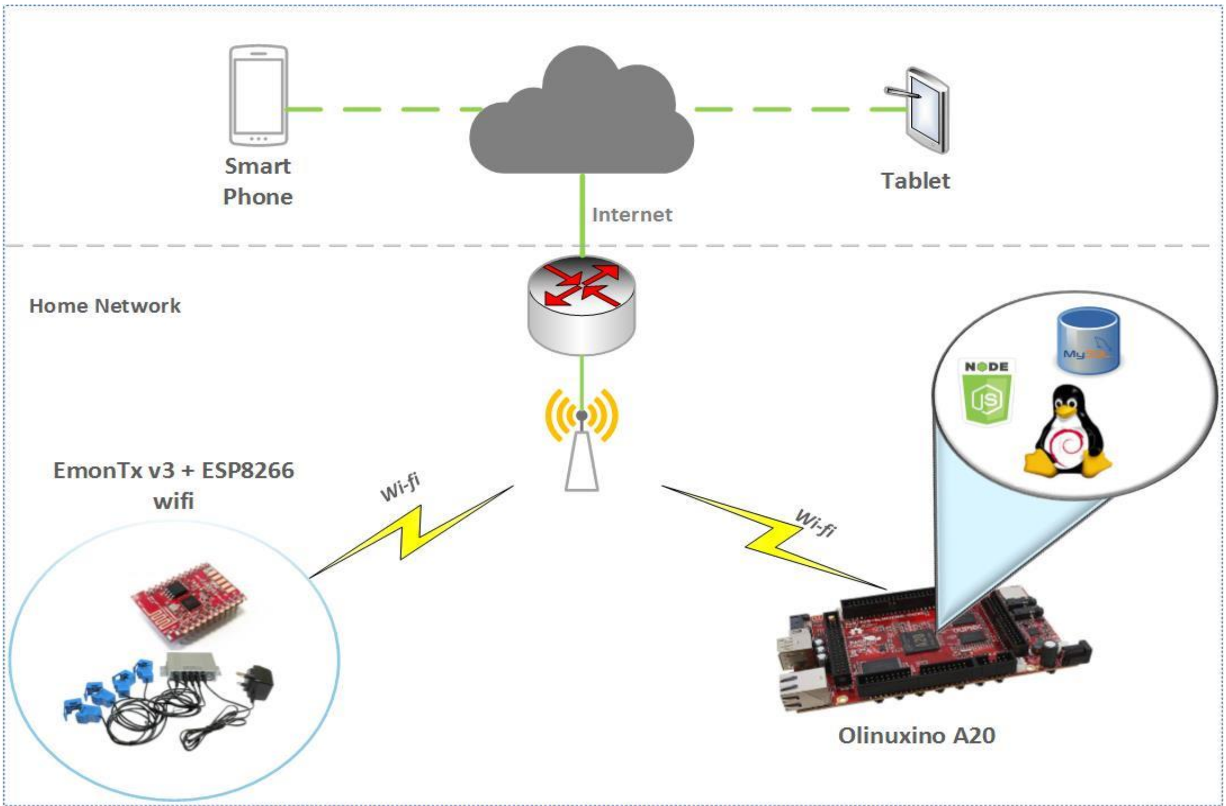

Figure 2. System overview scheme

This Energy Monitoring system is developed entirely with Open Source Hardware modules and Software platforms. The first reason for doing this is the already existing community of specialists working on those projects and providing adequate support and faster temps of development. The second reason is the level of flexibility that those platforms provide and their easy adaptation to different scenarios in real time systems.

HARDWARE REALISATION

The main components of this system are the sensor devices and the server. They are based on Open Source Hardware and Software.

The sensor devices are being realized with ATMEGA 328P based circuit EmonTx v3 [4] shown on *Figure 3 a*.) and MOD-WIFI-ESP8266 WIFI module [5] shown on *Figure 3 b*.). A current sensor (SCT-013-000) is connected to the EmonTx board and performs non-invasive sensing of current on the live wire in the electricity circuit. It supports a maximum current of 100A and it has no internal burden resistor but has zener diodes that limit the output voltage in the event of accidental disconnection from the burden. It is capable of providing sufficient voltage to fully drive a 5V input. Voltage is being sensed from the system by using AC-AC power supply adapter that has input of 240V 50Hz and output of 9V AC. Both sensors are connected to the chip's ADC inputs.

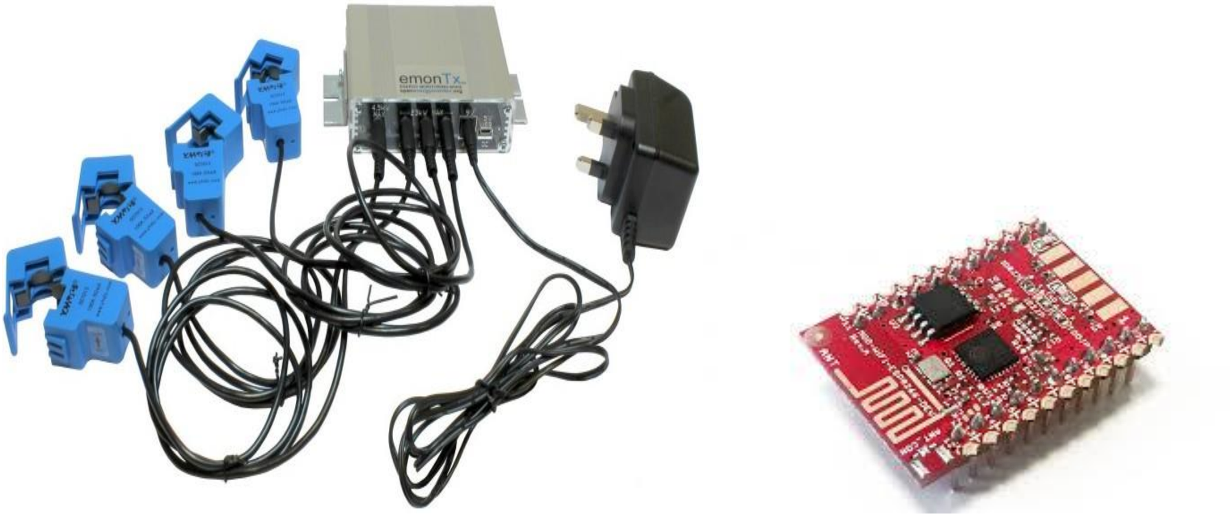

Figure 3. a) EmonTX v3 [4] b) MOD-WIFI-ESP8266 [5]

Arduino IDE [10] and C++ were used to program the ATMEGA 328P. The chip comes with pre-installed Arduino bootloader. The creators of this circuit (Open Energy Monitor [4]) provide easy to use library for calibrating the sensors and performing calculations. After that, the data from the calculations is being sent out to the MOD-WIFI-ESP8266 through UART.

MOD-WIFI-ESP8266 is a tiny development board based on ESP8266EX [6], a highly integrated SoC. It has a number of interfaces like I2C, SPI, SDIO and a couple of GPIOs. This chip allows implementation of WIFI TCP/IP stack and is mainly used to connect other boards with the Internet. Its software is written in C++ using the provided SDK for ESP8266 by ESPRESSIF [6]. At first power up, the chip is in Access Point mode and it runs a simple web server that provides a web page that gives the connected user an opportunity to configure the home network SSID, password and authentication token, and IP address for the server. After the initial configuration it connects to the home network and starts sending periodic HTTP POST requests to the server.

The system's database is located on an Olinuxino A20 single-board computer [5] shown on *Figure 4* which also runs a NodeJS [7] based web server which provides RESTful services. This board is fully capable of running operating systems like Debian and Android. It is based on ARM Cortex-A7 dual-core CPU with 1GB of DDR3 RAM memory and it is assembled with industrial grade components. It can be connected to the network with either its 100Mbit native Ethernet port or via WIFI dongle connected to one of its USB ports.

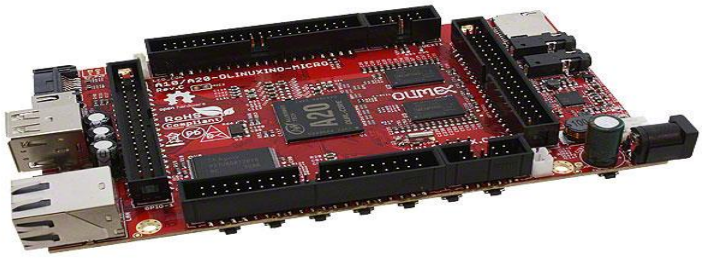

Figure 4. Olinuxino A20 single-board computer [5]

SOFTWARE REALISATION AND DATAFLOW

The dataflow diagram is shown on *Figure 5*. The software system comprises the modules in the following three main parts: sensor device(s), Monitoring application and server - based on the Olinuxino A20 OSHW (Open Source Hardware) single-board computer.

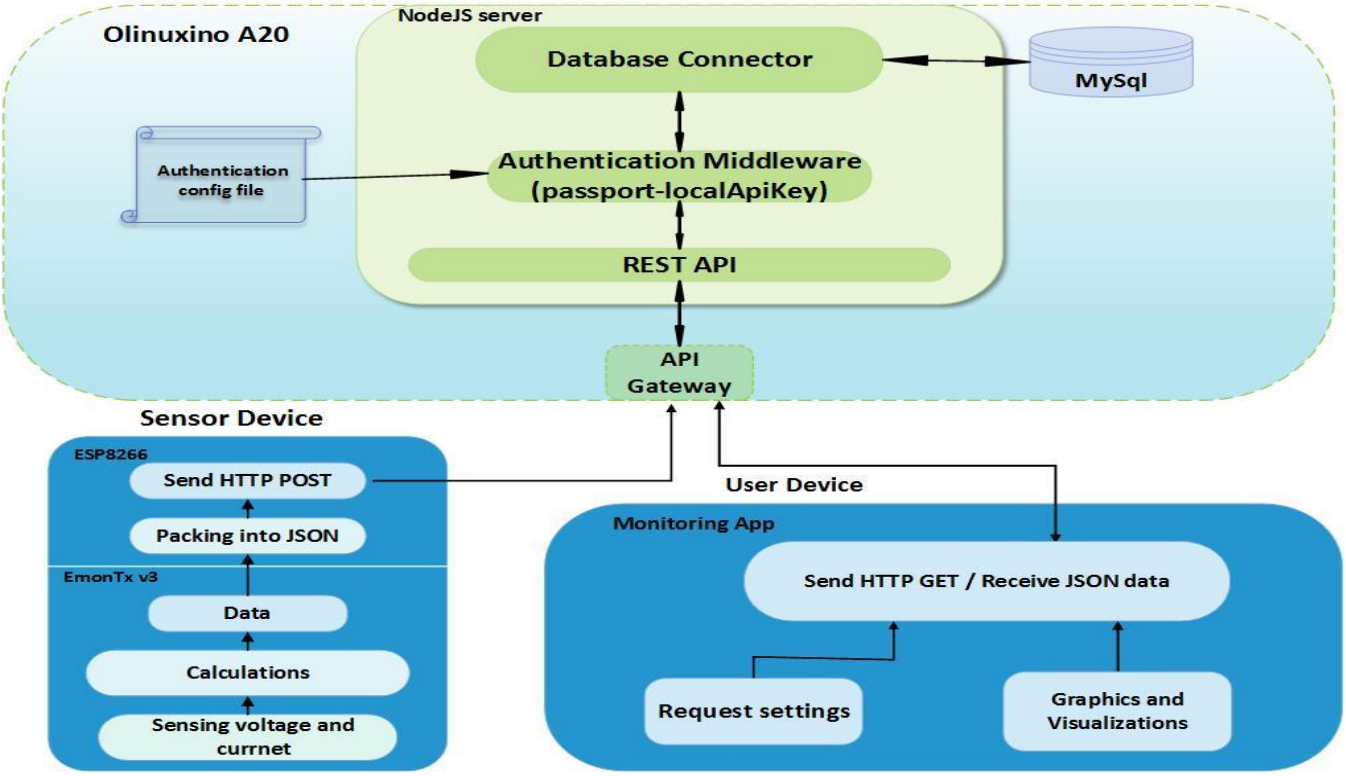

Figure 5. Dataflow diagram

The sensor device (EmonTx v3) collects data about the voltage and current drawn by the consumer that it is attached to, and then calculate Real and Apparent Power consumption, and Power Factor. After that, the calculated data is sent via UART to the ESP8266, and there it is packed into a JSON data format and is sent to the NodeJS web server, located on Olinuxino A20, via HTTP. Measurements are made every 5 seconds and at every minute the average from the collected data is sent to the server.

Example JSON data:

{ "NodeID" : "1" , "realP" : "504.88" , "appP" : "1291.03" , "Vrms" : "249.9" , "Irms" : "5.17" , "powrF" : "0.42" }

The monitoring application presents the collected data in a more user friendly way as shown on *Figure 6*. It allows the user to set what kind of data he would like to see by setting exact time period and the preferred measurements. After that, the data is requested from the NodeJS web server and is then visualized in the form of graphics and data tables.

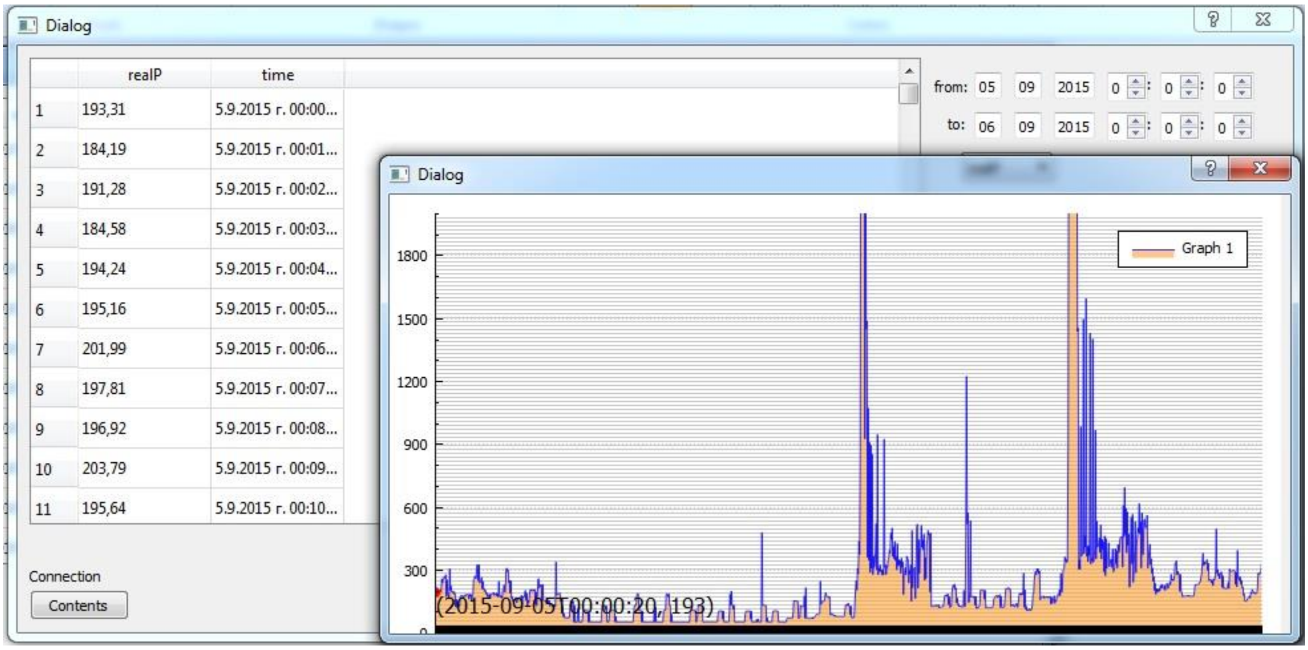

Figure 6. Monitoring application and power readings diagram

The server based on the Olinuxino A20 [5] is the most essential component of the system. It provides RESTful services which allows getting or manipulating specific data in the database. An API gateway is used to receive the HTTP requests and forward them to the REST API. For authentication purposes we propose using Passport – token based authentication middleware for NodeJS. The database connector provides an interface between the NodeJS server and MySql database.

NodeJS [7] is an open-source, cross-platform runtime environment for developing server-side Web applications. It's not a JavaScript framework but many of its basic modules are written in JavaScript. The runtime environment interprets JavaScript using Google's V8 JavaScript engine. NodeJS has an event-driven architecture capable of asynchronous I/O and it aims to optimize throughput and scalability in Web applications with many input/output operations.

CONCLUSIONS AND FUTURE WORK

The test experiments show that the developed system performs well, but it is still a prototype. The authentication and data need to be encrypted. In future we plan on adding ways to perform control over the home appliances so that the usage of electric energy can become more efficient. We envisage to use lightweight machine-to-machine (M2M) publish/subscribe messaging protocols like MQTT [8] or other M2M Cloud based frameworks like DeviceHive [9] that provide higher level of flexibility and security.

ACKNOWLEDGEMENTS

The presented work is supported by the H2020 EU funded "REMOURBAN" project and National Science Fund of Bulgaria project "Investigation of methods and tools for application of cloud technologies in the measurement and control in the power system" under contract NSF E02/12 (http://dsnet.tu-plovdiv.bg/energy/).

REFERENCES

[1] VEHBI C. GUNGOR at al, "Smart Grid and Smart Homes Key Players and Pilot Projects", IEEE industrial electronics magazine, December 2012, pp.18-34, ISSN: 1932 – 4529.

[2] Ihsan Ullah at al, "A Survey of Home Energy Management for Residential Customers", IEEE 29th International Conference on Advanced Information Networking and Applications (AINA), 2015, pp. 666 – 673, Print ISBN: 978-1-4799-7904-2.

[3] Mahmud, Rasel, et al. "A survey on smart grid metering infrastructures: Threats and solutions." Electro/Information Technology (EIT), 2015 IEEE International Conference on. IEEE, 2015.

[4] Open Energy Monitor official site: https://openenergymonitor.org/emon/

[5] OLIMEX official site: https://www.olimex.com/

[6] ESPRESSIF official site: http://espressif.com/

[7] NODEJS official site: https://nodejs.org/en/

[8] MQTT official site: http://mqtt.org/

[9] DEVICEHIVE official site: http://devicehive.com/

[10] ARDUINO official site: https://www.arduino.cc/

ABOUT THE AUTHOR:

Reader Evtim Peytchev PhD, School of Science & Technology, Nottingham Trent University, Phone: +44 (0)115 848 8371, E-mail: evtim.peytchev@ntu.ac.uk.

Mihail Lyaskov, student Bcs, Department of Computer Systems, Technical University of Sofia, Plovdiv branch, E-mail: m.lyaskov@abv.bg

Kostadin Popovski, student Bcs, Department of Computer Systems, Technical University of Sofia, Plovdiv branch, E-mail: k.popovski93@gmail.com

Prof. Grisha Spasova, PhD, Department of Computer Systems, Technical University of Sofia, Plovdiv branch, Phone: +359 32 659557, E-mail: gvs@tu-plovdiv.bg.