

# A Smart Solution for Electrical Power Monitoring Based on MCP39F501 Sensor

Grisha Spasov<sup>1</sup>, Milka Kutseva<sup>1</sup>, Galidiya Petrova<sup>2</sup> and Vasil Tsvetkov<sup>1</sup>

<sup>1</sup>Department of Computer Systems, <sup>2</sup>Department of Electronics, Technical University of Sofia, Plovdiv branch, 25, Tzanko Dustabanov, 4000 Plovdiv, Bulgaria  
{gvs@tu-plovdiv.bg, gip@tu-plovdiv.bg, vaccc@abv.bg}

**Abstract** – This paper presents a smart solution for AC power metering which can be easily integrated with the home monitoring and management systems in smart homes. It employs a sensor MCP39F501 for real-time electrical power measurement and implements the IoT concept based on three-layer architecture. The developed IoT solution is based on an open source hardware ESP32-EVB development board and a Raspberry Pi platform as a home gateway. The measured data could be displayed on a developed web page in two modes - as a dynamic chart for real-time data and as a static chart for the data stored in the database. The paper demonstrates the application of an inexpensive sensor in the development of a smart solution for continuous power metering and monitoring.

**Keywords** – Internet of Things, Smart metering, Real-time power measurement and monitoring.

## I. INTRODUCTION

The residential and commercial buildings are responsible for 40% of global energy use and contribute towards 30% of the total CO<sub>2</sub> emissions [1]. The striving to reduce energy consumption and associated gas emissions is one of the reasons for the development of Smart Home concept as part of Smart Cities and its associated Smart Grid [2]. This is achievable by conducting smart metering and using smart meters to monitor energy usage and indoor environmental conditions in homes and residential buildings. The first step to start with for developing energy management systems in buildings, is energy consumption monitoring using smart meters and smart devices. Smart meters can provide customers with detailed electricity consumption data of every appliance in real time thus helping the users to evaluate the overall power consumption in their home. Installation of the smart metering system is necessary and beneficial for consumers who would like to monitor and track consumed energy with the aim to reduce it [3].

The purpose of this article is to present a smart solution for real-time electrical power measurement and monitoring applicable for development of energy management systems in smart homes and buildings.

This paper is organized as follows: section 2 presents the MCP39F501 sensor for real-time electrical power measurement and its communication protocol. The implementation of a smart solution based on IoT concept and architecture is explained in section 3. Conclusions and the outlines for future work are presented in the last section.

## II. REAL-TIME ELECTRICAL POWER MEASUREMENT WITH MCP39F501 SENSOR

With the technological development many new wired and wireless sensors have become used in smart home systems. Some of the sensors can be connected directly to networks via their own interface or indirectly via other devices (embedded systems) with the capability of receiving and re-transmitting data.

The MCP39F501 is a single-phase power-monitoring IC designed for real-time measurement of input power for AC/DC power supplies, power distribution units and industrial applications [4]. The user-accessible sensor EEPROM memory is used to store the calibration settings and the data from a single measurement. The MCP39F501 sensor measures only on demand. Thanks to the built-in fast 16-bit processing core it provides real-time measurements and calculations on: the true RMS current and voltage, Active, Reactive and Apparent power, Line Frequency and Power Factor.

The sensor has a flexible two-wire interface, which is used to read/write the settings or retrieve the measured data. The communication protocol is based on the Simple Sensor Interface (SSI) protocol, which is used for point-to-point communication between a single master and a single slave at a time. The MCP39F501 is always the slave. However, it is possible to connect multiple slave devices to a single bus but the communication still remains one master to one slave at a time. The data sent via this interface is grouped in frames. As shown on Fig.1, each frame contains a header byte, the number of bytes in the frame, one or more command packets and the checksum. Having a checksum validation provides security of the protocol.

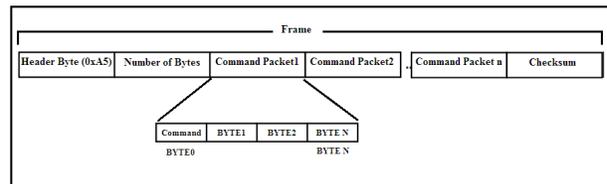


Fig. 1. Communication Frame of MCP39F501 sensor [4].

The command packet in the frame could have different length in bytes. The first byte in a command packet is always the command byte. In order to read the measured data it is necessary to send a frame with a command to select the sensor, followed by a command to set up the

register address to read from, then followed by a command to read number of bytes, and the checksum from the frame. The command for sensor selection is optional because the sensor might be working in the SINGLE\_WIRE mode. The command to set up the register address is also optional because it might have already been set in a previous frame.

The measured data in bytes does not strictly correspond to an actual value. It is up to the user to find out how exactly to calculate the actual measured value from the raw data received by the sensor because it depends on the calibration of the sensor.

### III. IMPLEMENTATION OF IOT BASED SOLUTION WITH WIRELESS ESP32-EVB BOARD

The developed smart solution for real-time power monitoring is based on integration of open source hardware platforms and MCP39F501 sensor. The functional structure is shown on Fig.2. It is based on the conventional IoT architecture which has been divided into three layers, i.e. perception layer, network layer, and application layer [5, 6]. The first layer is implemented by the MCP39F501 and the programming modules for its initialization, data transfer from the sensor and the measurement results.

The wireless module ESP32-EVB was used to realize a part of the first and a part of the second layer. It is open source hardware board having WiFi and Bluetooth LE connectivity, but supporting also wired 100Mb Ethernet Interface, which making it perfect for IoT applications [7]. It supports the needed communication protocols and the necessary ports, and is easy for programming. MCP39F501 is connected to ESP32-EVB via UART interface – TX (GPIO\_NUM\_4) and RX (GPIO\_NUM\_36) at 4800 bauds/s. The ESP32-EVB module receives raw data from the sensor EEPROM memory in which the measured and calculated results are stored (24 consecutive bytes starting at address 0x0004). The Description of the meanings of the results from a measurement and their size are shown on TABLE 1.

The ESP32-EVB board provides the wireless WiFi communication to the gateway. It has complete embedded TCP/IP stack (IPv4, TCP/UDP/HTTP/FTP). The firmware that is built into ESP32-EVB development board allows wireless communication to be managed via APIs.

TABLE 1. MEASURED AND CALCULATED PARAMETERS WITH MCP39F501 SENSOR

| Description of the meanings of the results from a measurement | Size in bits (u – unsigned; s – signed) |
|---|---|
| Current RMS   | u32                                     |
| Voltage RMS   | u16                                     |
| Active Power  | u32                                     |
| Reactive Power  | u32                                     |
| Apparent Power  | u32                                     |
| Power Factor  | s16                                     |
| Line Frequency  | u16                                     |
| Thermistor Voltage  | u16                                     |

The WiFi module supports three modes for connection:

- Station (STA) – connects to existing network;
- Access point (AP) – creates a network which the other devices connect to;
- Mixed.

In our implementation the STA mode is chosen because in order to provide high scalability the better way is to have a single wireless network created by the home gateway, which all sensor devices to connect to. This approach will allow realization of architectures with two gateways [8] in which the ESP32-EVB module could perform as a sensor gateway.

The communication with the gateway starts with gateway discovery through UDP and then switching to TCP persistent connection between them. After the establishment of TCP socket, ESP32-EVB module sends the name of the sensor and its settings to the server. Then the ESP32-EVB starts sending measurement data periodically to the gateway. The period can be changed by the gateway and recorded in the memory of ESP32-EVB module. The data from a measurement represents 24 bytes received from MCP39F501 sensor without any modifications or computations made by ESP32-EVB module.

The received 24 bytes of data shown in TABLE 1 is decoded by the gateway and converted to JSON string because it is the most common format used by the HTTP clients. Each integer in the 24 data bytes is stored with the most significant byte first.

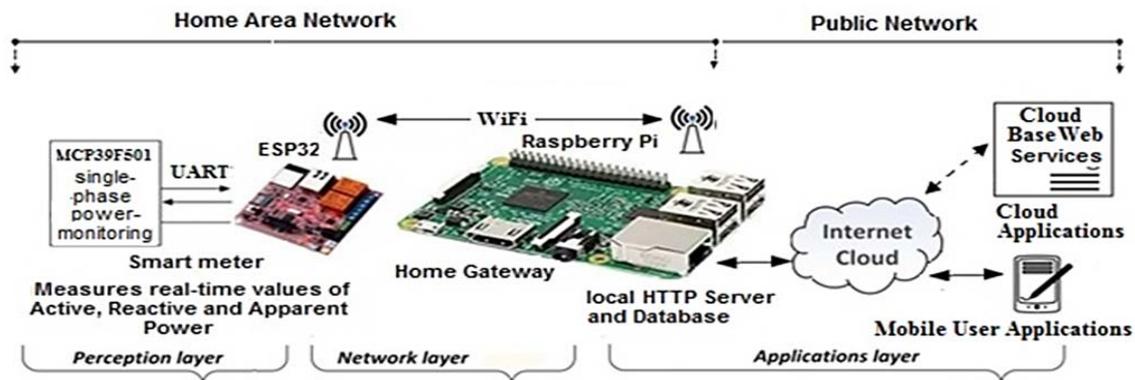


Fig.2. Functional structure of the developed IoT based solution for electrical power monitoring.

The actual measured values depend on the calibration of MCP39F501 sensor. With the default calibration, the actual measured values are calculated the following way:

- Current RMS—the integer value multiplied by 0.0001;
- Voltage RMS – the integer value multiplied by 0.1;
- Active Power – the integer value multiplied by 0.01;
- Reactive Power – the integer value multiplied by 0.01;
- Apparent Power—the integer value multiplied by 0.01;
- Power Factor – if the integer value is 0x7fff then the actual value is 1.0, otherwise the actual value is calculated by dividing the sent integer value to 32678.
- Line Frequency—the integer value multiplied by 0.001;
- Thermistor Voltage – the same value.

The exact JSON format is as follow:  
`[“%Timestamp%”, %CurrentRMS%, %VotageRMS%, %ActivePower%, %ReactivePower%, %ApparentPower%, %PowerFactor%,%LineFrequency%, %ThermistorVoltage%”]`, where %Timestamp% is a string, representing the gateway time of receiving the data in format “yyyy-MM-ddTHH:mm:ss.fff”. All the others are number values calculated the way just described.

The third layer is implemented using a built-in HTTP Server upon ASP.NET Core 2.2 in the gateway. It is realized with a single-board computer Raspberry Pi. The gateway transmits the received data from all sensors to the HTTP clients (browsers) using real-time communication provided by SignalR (from .NET) using web sockets. Each connected device has its own identifier (Guid) and can be accessed by URI like REST. When data is received from many sensors, the gateway sends that data to all web pages that must show them. For example, if a client is viewing

the page showing the data from Device2, and the data from Device1 is received by the gateway, that data will not be sent to that client because has nothing to do with the page the client is viewing. That way the network traffic between the clients and the gateway is limited only to the data useful by each separate client. Sensor settings (name, period, position on the interface) can be changed from any HTTP client. After the change they are automatically sent to the remaining HTTP clients. The gateway also supports storing data from every sensor into a database. The individual web page for each connected device visualizes measurement data in two charts – as a dynamic chart for the real time data and as a static chart for data stored in the database. An example for visualization of the measured Active, Reactive and Apparent power on static chart is shown on Fig. 3.

The data from sensors, stored in the database, can also be used for analyzing the sensors condition, performance and stability.

Our idea was to simultaneously display data from many sensors in a web page and dynamically discover their connection/disconnection with the gateway plus providing a way to change the settings of each sensor.

Using ASP.NET core provides cross-platform option, so the gateway can run on different operating systems. It can automatically discover connected to the same network sensor even if the sensor type is new and not yet included in the system by default. It is possible to dynamically install “drivers” for new sensors without the need to restart the gateway HTTP server.

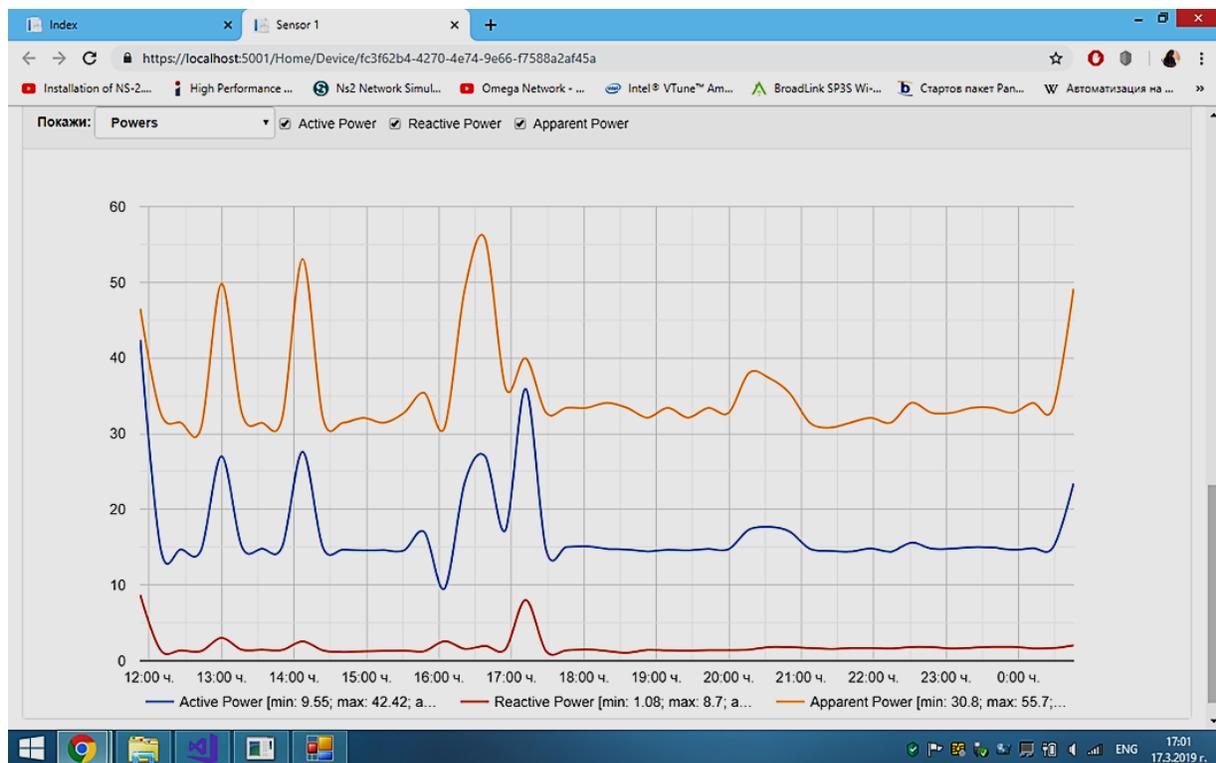


Fig.3. Static chart of the electrical power measured from Sensor 1 in a Smart home.

#### IV. CONCLUSION

In this article a smart solution for electrical power monitoring applicable in smart home metering systems is described. It employs an inexpensive real-time power measurement sensor and implements the IoT concept and architecture.

In the future work some additional options for extracting information about the peak power consumption in real-time and identifying which appliances in home are used will be added. This will be followed by automatic switching on and off of electrical appliances in Smart Home, depending on their actual power consumption.

The data stored in the database further will be used to analyze daily, monthly or yearly energy consumption, identify energy usage pattern and predict future energy demand. Thus way energy consumption forecasts will be generated which will be used in the development of smart home energy management systems.

#### ACKNOWLEDGMENT

This work was supported by the European Regional Development Fund within the Operational Programme "Science and Education for Smart Growth 2014 - 2020" under the Project Center of Competence "Intelligent mechatronics, eco- and energy-saving systems and technologies" BG05M2OP001-1.002-0023-C0.

#### REFERENCES

- [1] Muhammad Waseem Ahmad, Monjur Mourshed, David Munday, Mario Sisinni, Yacine Rezgui, "Building energy metering and environmental monitoring – A state-of-the-art review and directions for future research", Elsevier Energy and Buildings 2016 Vol. 120, pp. 85-102, ISSN: 0378-7788.
- [2] Bahram Shakerighadi, Amjad Anvari-Moghaddam, Juan C. Vasquez, and Josep M. Guerrero, "Internet of Things for Modern Energy Systems: State-of-the-Art, Challenges, and Open Issues", Energies 2018, 11(5), 1252; <https://doi.org/10.3390/en11051252>
- [3] Sun Q, Li H, Ma Z, Wang C, Campillo J, Zhang Q, et al. A "Comprehensive Review of Smart Energy Meters in Intelligent Energy Networks", IEEE Internet of Things Journal, 2016; 3: p. 464-79.
- [4] <https://ww1.microchip.com/downloads/en/DeviceDoc/20005256A.pdf>
- [5] Ala Al-Fuqaha et al., "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications", IEEE Communication Surveys & Tutorials, Vol. 17, No 4, 2015, Page(s): 2347–2376, ISSN: 1553-877X, DOI: 10.1109/COMST.2015.2444095
- [6] Peter Yakimov, Atanas Iovov and Nikolay Tuliev, "Implementation of Internet of Things based Solution of Universal Power Transducer", IEEE, 2017, XXVI International Scientific Conference Electronics - ET2017, ISBN: 978-1-5386-1753-3, DOI: 10.1109/ET.2017.8124410, Page(s): 1 – 4.
- [7] <https://www.olimex.com/Products/IoT/ESP32/ESP32-EVB/open-source-hardware>
- [8] Galidiya Petrova, Grisha Spasov and Georgi Pazhev, "Realization of a Home Automation Network with Two Gateways Based on Open Source Hardware", Annual Journal of Electronics, 2015, ISSN 1314-0078, pp. 116-119.