Data Acquisition of a Photovoltaic Installation Using ESP8266 card and MQTT Protocol

Yassine Mesbahi, Abdelali Ed-Dahhak, Mohammed Guerbaoui, Abdelouahed Selmani and Abdesalam Lachhab

February 9, 2023

Data Acquisition of a Photovoltaic Installation using ESP8266 card and MQTT Protocol

Yassine Mesbahi
Department of Electrical Engineering
Faculty of sciences
Moulay Ismail University
Meknes, Morocco
y.mesbahi@umi.ac.ma

Abdelali Ed-dahhak
Modeling Systems Control and Telecommunications Team
High School of Technology,
Moulay Ismail University
Meknes, Morocco
e.eddahhak@est-umi.ac.ma

Mohammed Guerbaoui
Modeling Systems Control and Telecommunications Team
High School of Technology,
Moulay Ismail University
Meknes, Morocco
m.guerbaoui@umi.ac.ma

Abdelouahed Selmani
SARS Team
National School of Applied Sciences of Safi (ENSAS),
Cadi Ayyad University
Safi, Morocco
Sel.bdel@gmail.com

Abdesalam Lachhab
Modeling Systems Control and Telecommunications Team
High School of Technology,
Moulay Ismail University
Meknes, Morocco
abd_lachhab@yahoo.fr

***Abstract*—Data acquisition systems are enormously used with photovoltaic installations to acquire all the data of the installation to improve and optimize photovoltaic production. The objective of this work is to show the interest in the Internet of Things (IoT) to facilitate real-time data acquisition using Message Queuing Telemetry Transport (MQTT) protocol which has several advantages over other IoT protocols, by implementing MQTT on the ESP8266 development board which is Wi-Fi based. Sensors are connected to the ESP8266 to capture data. This data is then sent to the ThingSpeak platform as anIoT analytics platform for storage and analysis. We can access it through a browser or through various electronic devices, thus allowing us to monitor the evolution of our photovoltaic installation.**

***Keywords*—Data acquisition, IoT, MQTT,ESP8266, ThingSpeak.**

I. INTRODUCTION

The demand for solar energy is increasing considerably compared to fossil fuels in many countries ofthe world. Many researches is being done on solar photovoltaic (PV) energy to optimize production [1].

Frequent supervision of a PV system is necessary for itto have very high efficiency, especially since there is a probability of the appearance of failures during the production of electrical energy. Most PV systems are installed in remote locations, which makes supervision impossible for human beings.

To solve this problem, we propose a method of remote monitoring that's extensively employed. It is the IoT. This technology allows all devices such as microcontrollers, sensors, etc., to connect to each other via the Internet to provide data to users in real-time. [2, 3].

It also collects several detailed information about the objects to provide new development perspectives. IoT is applied in several fields, such as healthcare, home automation, automotive industries, smart systems, renewable energy systems, and other fields [3, 4].

IoT lately is focusing on solar photovoltaic energy because of the huge use of this energy in distributed generation. This helps to create significant activity between IoT service providers with their customers [5].

In this work we present an application of one of the best communication protocols of the IoT, it is the MQTT which is characterized by lightness, reliability and low power consumption, using the ESP8266 integrated circuit to transfer data to the ThingSpeak platform to visualize and analyze them.

II. COMMUNICATION PROTOCOLS FOR IOT

In recent years, several IoT communication protocols are developed and evolved, such as Hypertext Transfer Protocol (HTTP), MQTT, Constrained Application Protocol (CoAP), and Advanced Message Queuing Protocol (AMQP). Each of them has its characteristics and strengths, either in terms of speed, energy consumption, security, reliability, etc. [6].

However, HTTP and MQTT are still the most used with usual microcontrollers like Arduino, ESP, andRaspberry Pi.

A. HTTP protocol:

HTTP is a client-server protocol, based on Transmission Control Protocol (TCP), it allows transferring of various data such as images and texts on the Internet in a fast, easy and stable way, this transfer is done from the server to the users, such as browsers. HTTP ensures the integrity of the transmitted data. The HTTP protocol works as follows: requests are sent from the client to the server, which processes them and provides a response, this is the request-response method [7].

B. MQTT protocol:

MQTT is an open-source, Ethernet TCP/IP-based publish-subscribe messaging protocol widely used in IoT applications. It is characterized by its lightweight, low network bandwidth usage and low power consumption [8].

Since it is a lightweight protocol, it is also suitable for machine-to-machine (M2M) communication and wireless sensor networks (WSN) [9].

It is a protocol that can transmit messages between several clients through a broker that manages the transfer of messages, as shown in Fig.1. It is especially suitable for real-time applications.

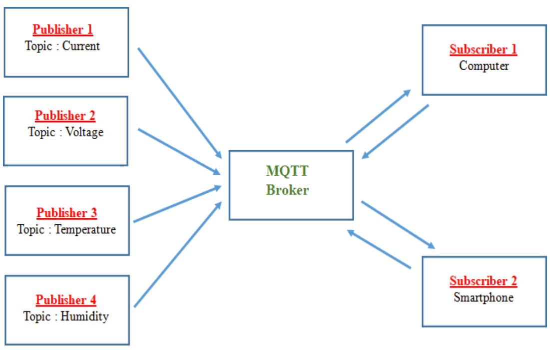

Fig.1. Publication/Subscription structure

The subscriber will subscribe to a topic among the published topics or all the topics, and when the publisher will make a publication, the broker will transfer to the subscriber, and so the subscriber will receive the data.

C. Comparison of HTTP and MQTT.

At the delivery level, the MQTT protocol can set the quality of service (QoS) to publish the topics to the client. It supports three qualities of service. QoS 0 (at most one), this is the default mode. In this case, the information is transmitted only once without storing it by the broker and without acknowledgment. Qos 1 (at last one), the message is transmitted once or more with acknowledgment of receipt. QoS 2 (exactly one), No matter how many times the message is transmitted, the recipient only receives it once [8]. By default, this service does not exist in HTTP.

At the level of speed, an experiment is made, of the sending of data and its complete transmissions, showing that the ability to send data through MQTT is six times faster than HTTP [7].

In terms of size, in HTTP the messages consist of long headers to help with message readability. While in MQTT, the message has a very small header [10].

At the energy level, when the connection time is higher, the energy consumption by MQTT is less than that consumed by HTTP [11].

III. ARCHITECTURE OF DATA ACQUISITION SYSTEM OFA PV INSTALLATION

In this part we present the complete architecture of acquisition of the parameters of a photovoltaic installation, as it is shown in Fig.2. We make the acquisition of current, voltage, temperature, and humidity assuming that the solar irradiation is fixed.

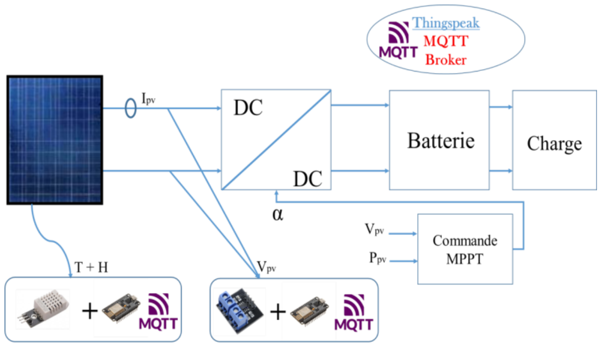

Fig.2. Data acquisition system of a photovoltaic installation

In this experiment, we used a photovoltaic generator, its characteristics are presented in TABLE 1. With a BOOST type DC_DC converter, a Lithium battery, and a resistive load.

TABLE 1. Characteristics of the solar panel

Power rating	50 W
Type of cell	Silicium monocristallin
Voltage Mpp	18.62 V
Current Mpp	2.69 A
Short-circuit current	2.92 A
Open-circuit voltage	22.6 V
Size (Length x Width x Height)	670 x 550 x 30 mm

For the acquisition elements, we used the MAX471 sensor to measure the current (Ipv) and voltage (Vpv) at the solar panel output. Its characteristics are presented in TABLE 2. And the DHT22 sensor to capture the temperature (T) and humidity (H) at the solar panel. Its characteristics are presented in TABLE 3.

TABLE 2. Electrical characteristics of the MAX471

Voltage Range	3V to 36V
Current Range	0A to 3A
Accuracy	±2%
Output type	Analog Output

TABLE 3. Electrical characteristics of the DHT22 sensor

Operating Voltage	3.5 V to 5.5 V
Operating Current	0.3mA (measuring) 60µA (standby)
Temperature Range	-40°C to 80°C
Humidity Range	0% to 100%
Accuracy	±0.5°C and ±1%
Output	Serial data

And for control, we proposed the ESP8266 board model ESP-12. The ESP8266 is a system-on-chip (SOC) that contains a microcontroller with a Wi-Fi module. It hasTCP/IP protocol. Its processor Tensilica Xtensa LX106 is of the 32-bit Reduced Instruction Set Computer (RSIC) architecture, which allows for very low power consumption and a maximum clock speed of 160 MHz.

ESP8266 can be programmed with several languages, but the most used languages are C++ with (Development Environment) the Arduino IDE and MicroPython with the Micropython firmware. It integrates General Purpose Input/Output (GPIO), a CP2102 TTL to USB chip for programming and debugging, a Wi-Fi antenna, Flash memory of 4 Megabits, and a micro USB connector, as shown in Fig.3.

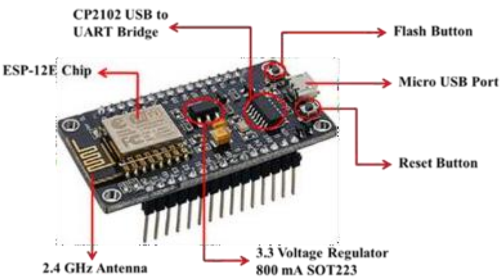

Fig.3. ESP8266 Borad

The ESP-12 model has 17 General Purpose Input/Output (GPIO), many of which are shared or multiplexed with other features of the board such as Inter-Integrated Circuit (I2C) bus, Serial Peripheral Interface (SPI), Pulse Width Modulation (PWM), a 10bits analog- to-digital converter, etc. as shown in Fig.4.

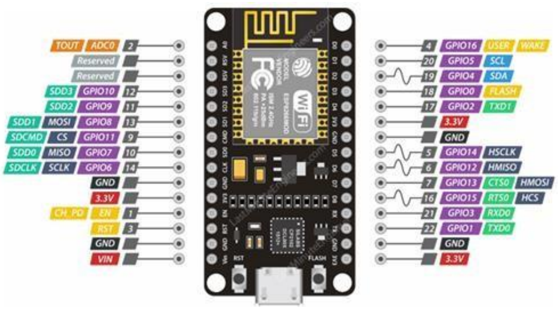

Fig.4. ESP8266 Pinout

Pins from D0 to D8 are digital inputs/outputs. RX and TX pins are used for serial communications and loading programs on the device, they can be used also as digitalI/O.

The ESP8266 has three operating modes, active mode, sleep mode, and deep sleep mode, which makes this board very convenient for IoT and wireless sensor network applications [12].

The MAX471 module needs two analog inputs, one for current and one for voltage, and as a solution, we have added an analog-to-digital converter type MCP3002.

The MCP3002 is a 10-bits analog-to-digital converter that combines low power consumption (5nA typical in standby and 550µA typical in active) with high performance.

MCP3002 can operate over a voltage range between 2.7V and 5.5V. It contains two input channels. It can communicate with the ESP board by using a simple serial interface compatible with the SPI protocol.

IV. RESULTS AND DISCUSSION

After data acquisition, we visualize the data in the ThingSpeak platform. ThingSpeak is an IoT analytics platform dedicated to the Internet of Things, it was made by MathWorks Company, the principle of this platform is visualize, analyze and store data from sensors of various IoT applications in the cloud in real-time.

ThingSpeak communicates with the Internet connection to transport data packets between the connected objects and the ThingSpeak cloud using HTTP or MQTT protocol.

The REST and MQTT APIs are used for data access. The REST API communicates via HTTP and is based on the request-response principle. The MQTT API communicates via TCP/IP sockets or WebSockets (in this case, it can be secured with SSL) and is based on publish- subscribe method. While the MQTT Publish method only permits data update, the MQTT Subscribe is required for data retrieval, whereas the REST request supports both data update and retrieval.

ThingSpeak can work with MATLAB and Simulink, Arduino, Raspberry, ESP32, and ESP8266. To use it, you just need to create a MathWorks account, define the number of fields to use, and have a Write API Key.

Each ThingSpeak channel has eight fields of 255 characters of data, which can display the measurements of a sensor. ThingSpeak has two types of views. Private View, only the user can see the information in this case. And a Public View to make public access to the channel.

This platform allows us to have a database of our project under the formats JavaScript Object Notation (JSON), Extensible Markup Language (XML), and Comma-Separated Values (CSV) for integration intoapplications.

Each parameter is in a separate field based on the date and time. The current Ipv in milliamp (mA), and voltage Vpv in volts (V), as shown in Fig.5 and Fig.6. The temperature in degree Celsius and humidity in percent as shown in Fig.7 and Fig.8.

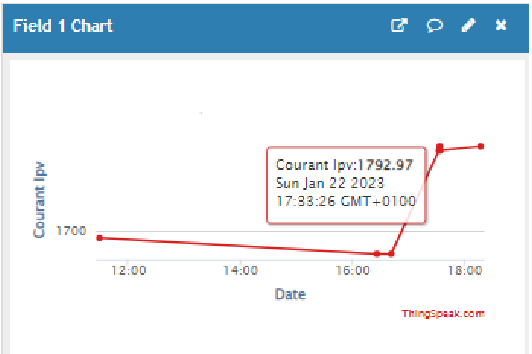

Fig.5. Real-time evolution of Current

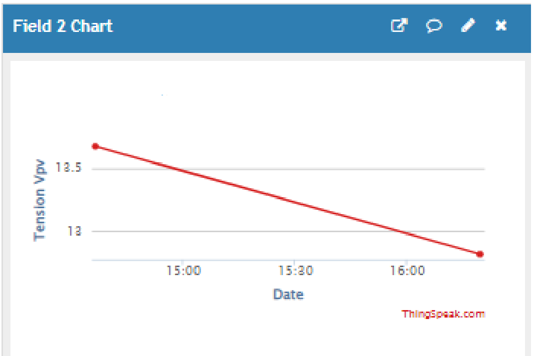

Fig.6. Real-time evolution of Voltage

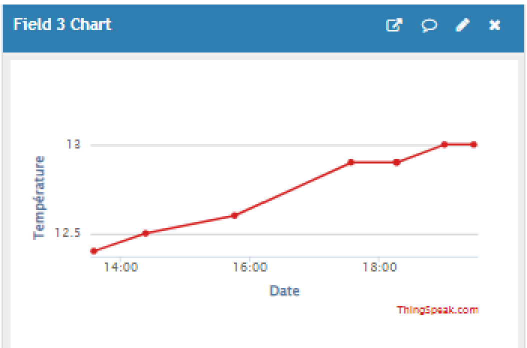

Fig.7. Real-time evolution of Temperature

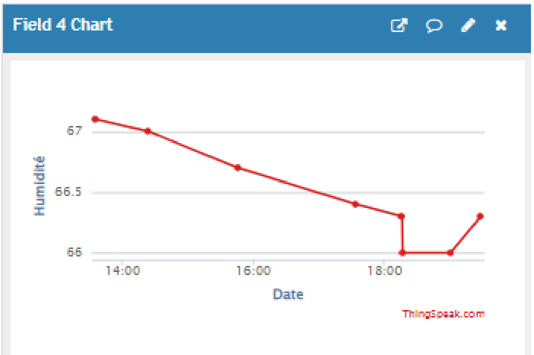

Fig.8. Real-time evolution of Humidity

The visualization can be either in the form of curves or numerical values.

DHT22 sensor gives good values. For the MAX471 module, it gives almost the same current values as the ammeter, between 1670 and 1800 mA, while for the voltage values, there are small differences compared to the voltmeter, up to 0.7 volts of offset.

V. CONCLUSION

This study presents the advantages of using ESP8266 board that implement the MQTT protocol to acquire data from a photovoltaic installation. The data is displayed in the ThingSpeak platform to supervise this installation either by navigator or by smartphone in real time.

The implementation of MQTT in the ESP8266 allowed us to transmit data in a reliable, lightweight and low-power consumption manner.

The sensors used in this experiment, either DHT22 to measure temperature and humidity, or MAX471 to measure current and voltage, gave good results with a good response on the ThingSpeak platform.

This work is the first step. Afterward, we will try to take advantage of the acquired data to optimize the energy of our photovoltaic installation.

REFERENCES

[1] Renewables 2019 global status report, Renewable Energy Policy Network for the 21st Century, (2019).

[2] L. Atzori, A. Iera, G. Morabito. The internet of things: A survey. Computer Networks, vol. 54, no. 15, pp. 2787–2805, (2010).

[3] A. Zanella, N. Bui, A. Castellani, L. Vangelista, M. Zorzi. Internet of Things for Smart Cities. IEEE Internet of Things Journal, vol. 1, no. 1, pp. 22-32, Feb. (2014). doi: 10.1109/JIOT.2014.2306328

[4] Bellavista, G. Cardone, A. Corradi, L. Foschini. Convergence of MANET and WSN in IoT urban scenarios. IEEE Sensors Journal, vol. 13, no. 10, pp. 3558–3567, Oct. (2013)

[5] Nallapaneni Manoj Kumar, KarthikAtluri, SritejaPalaparthi. Internet of Things (IoT) in Photovoltaic Systems, National Power Engineering Conference (NPEC), (2018).

[6] Performance Comparison of IoT Communication Protocols, Thays Moraes, Bruno Nogueira2, Victor Lira and Eduardo Tavares, IEEE International Conference on Systems, Man and Cybernetics (SMC) Bari, Italy. October 6-9, 2019

[7] R AAtmoko, R Riantini, and M K Hasin. IoT real time data acquisition using MQTT protocol, J. Phys.: Conf. Ser. 853 012003. (2017)

[8] Lee S, Kim H, Hong D K, et al. Correlation analysis of MQTT loss and delay according to QoS level[C]// International Conference on Information NETWORKING. 2013:714-717

[9] Authorization Mechanism for MQTT-based Internet of Things – A.Niruntasukrat, C.Issariyapat, P.Pongpaibool, K.Meesublak, P.Aiumsupucgul, A.Panya [IEEE ICC2016-Workshops: W07- Workshop on Convergent Internet of Things].

[10] Enhancing Security in IoT based Home Automation - Idris Afzal Shah,Faizan Amin Malik, Syed Arshid Ahmad [978-1- 4673- 9338-6/16/$31.00 ©2016 IEEE]

[11] Dr. BharatiWukkadada, Dr. KirtiWankhede Mr. RamithNambiar, Ms. Amala Nair. Comparison with HTTP and MQTT in Internet of Things (IoT), ISBN: 978-1-5386-2456-2. 2018 IEEE.

[12] Tejas Thaker, ESP8266 based Implementation of Wireless Sensor Network with Linux Based Web-Server, 2016 Symposium on Colossal Data Analysis and Networking (CDAN)