



IoT embedded systems network and sensors signal conditioning applied to decentralized photovoltaic plants

Renata I.S. Pereira ^{a,*}, Sandro C.S. Jucá ^b, Paulo C.M. Carvalho ^a

^a Laboratory of Alternative Energies, Electrical Eng. Department, Pici Campus, Federal University of Ceará, 60455-760 Fortaleza, Brazil

^b LAESE Laboratory, Computing Department, Federal Institute of Ceará, Industrial District I, 61939-140 Maracanaú, Brazil



ARTICLE INFO

Article history:

Received 27 November 2018

Received in revised form 16 January 2019

Accepted 29 April 2019

Available online 8 May 2019

Keywords:

Photovoltaic microgeneration

Signal conditioning

IoT

Monitoring network

ABSTRACT

The proposed Internet of Things (IoT) monitoring network aims at reducing costs associated with commercial dataloggers and sensing modules, requiring control and data storage that need proprietary software. Other usual drawbacks of commercial solutions are limited sensor connections with low expansion flexibility, maintenance restricted to the manufacturer and long cable communication distances. Our developed IoT network was implemented and tested in three grid-connected photovoltaic (PV) plants, installed in Fortaleza and Maracanaú, in Brazil, and Cologne, in Germany. The proposed IoT embedded systems are based on free software, allowing online distribution, free usage and communication with a server in the Cloud wirelessly via WiFi. A web page called Web Monitor was developed for online data consultation and for real-time monitoring of the three plants. PV modules temperature monitoring has the objective of providing data for the analysis of electricity generation efficiency and for detecting failures characterized by overheating of the PV plant. Meteorological data such as solar irradiance, ambient temperature, relative humidity and wind speed are also monitored, allowing a more complete analysis of the effect of these variables on the PV module's performance. Hence, we propose the design and development of an IoT modular system to compose a worldwide monitoring network focused on meteorological and PV modules temperature data, available to researchers from partner institutions.

© 2019 Elsevier Ltd. All rights reserved.

1. Introduction

Data acquisition and monitoring systems are important in the evaluation of the solar potential of a region, in the prognosis of failures, in the practical verification of project data, as well as in the optimization of power plants conversion efficiency. With the increasing number of decentralized power generation units, monitoring systems modularization is only possible with reduction of costs and flexibility of installation and maintenance [1]. We focus our research on monitoring micro and mini photovoltaic (PV) grid-connected plants.

Considering the Brazilian context, connected PV plants are increasing their importance in the country electricity matrix. Brazil has an annual average irradiation between 1500 and 2400 kWh/m², higher than most of the European countries, where PV technology has been widely explored and implemented. The country's solar

resource has a low seasonal and interannual variability due to its location in an equatorial region.

Brazilian PV market is increasing in a fast rhythm. In January 2017, Brazilian National Electric Energy Agency (ANEEL) registered more than 7610 distributed generation connections (73.5 MW of installed capacity); PV plants participate with 7528 plants totaling 57 MW of installed capacity, of which almost 6000 are domestic connections [2]. In June 2018, Brazilian Association of Solar Photovoltaic Energy (ABSOLAR) registered a total of 17.1 MW of PV power installed in the state of Ceará (1040 units), placing the state as a leader in the Brazilian Northeast region [3]. In total, Brazil has already 1 GW of installed PV capacity.

Only 13 to 18% of the solar energy is converted into electricity (PV efficiency). Losses in the form of heat in the PV cell cause an increase in temperature, reducing the electrical efficiency by increasing the saturation current of the cell. Therefore, each degree above the Nominal Operating Cell Temperature (NOCT) corresponds to 0.45% decrease in the conversion efficiency [4] for the polycrystalline module used in the proposed project [5]. NOCT corresponds to 45 °C ($\pm 2^{\circ}\text{C}$) for the PV module; however, it is necessary to observe the module datasheet as the specification varies according to the PV technology used by the manufacturer.

* Corresponding author at: Laboratory of Alternative Energies, Federal University of Ceará, Electrical Eng. Department, Pici Campus, Humberto Monte Av., 60455-760 Fortaleza, Brazil.

E-mail address: renata@dee.ufc.br (R.I.S. Pereira).

With the aim of modularizing, reducing costs and making the online monitoring system even more practical and fast in data processing, we propose a free software and hardware monitoring system using Internet of Things (IoT) concept. This proposal represents one of the innovations of this project, considering the launch of the IoT ESP 32 module in 2016 and commercialization in 2017, allowing its application in the monitoring of individual PV modules due to the low cost (US\$ 8) and the embedded WiFi and Bluetooth circuitry and antenna. Hence, the proposed IoT system aims at eliminating the disadvantages of conventional online monitoring units applied to centralized plants: imported and high cost commercial dataloggers and sensor modules, control and data storage via proprietary software, limited sensor connections, maintenance restricted to the manufacturer as well as long communication distances using cable (optic fiber or Ethernet).

In our project, IoT ESP 32 module was configured and implemented to collect PV module temperature and solar irradiance; IoT ESP 8266 module (basic version launched in 2014) was used for wind speed, ambient temperature and relative air humidity monitoring.

Conventional dataloggers usually have a specific input for sensors like the used: anemometers, pyranometers, PT100. As we propose our own IoT hardware and firmware for data monitoring, we needed also to design specific signal conditioning circuits for reading the PT100 temperature sensor and the pyranometer considering AD range and resolution, bit-voltage curve and input and output circuit voltages. The logic implemented for reading the anemometer is innovative because requires no conditioning circuit and is rather connected directly to the AD input of the IoT module ESP 8266. The proposed conditioning circuits and the anemometer reading logic are novel, scientific innovative, specially designed for IoT embedded systems and were developed using simple components available at local market. After implementation, the sensors were then calibrated using commercial equipment and presented measurements within the specific error range of each sensor.

IoT monitoring was developed, implemented and tested in three PV plants:

- Cologne – Germany (Lat.: 50.933730, Long.: 6.988664);
- Fortaleza – Brazil (Lat.: -3.737491, Long.: -38.572781);
- Maracanaú – Brazil (Lat.: -3.872287, Long.: -38.612233).

The design and construction of an IoT monitoring network applied to PV microgeneration plants in distinct sites is the main innovation of the proposed project. The three mentioned PV plants are grid-connected, but only the two PV plants installed in Brazil have the electrical generation data available. Considering that PV DC voltage, current and power data are supplied by the grid-tie inverter, we didn't install redundant sensors to measure these variables; hence, we could reduce costs and eliminate faulty components in a standardized and inspected plant.

2. Literature review

The proposed wireless monitoring network for meteorological data, PV generation and cell temperature analysis was designed for data collection and practical verification of values previously estimated with mathematical models.

An Arduino-based data acquisition and transmission system (DATS) with local control and SD card storage, supporting 16 sensors, for PV systems temperature measurement is proposed by [6].

A low-cost, computer-based open source DATS for general data acquisition is also proposed in [7] using a dedicated PC with USB or Serial RS-232 interface for communication, which requires installation of a driver to generate a virtual COM port. The system is com-

patible with LabVIEW™ (Laboratory Virtual Instrument Engineering Workbench) [8] and the used analog-to-digital (AD) converter has a 16-bit resolution.

The project proposed by [9] consists of a low-cost system to measure an off-grid PV module voltage and current. Several sensor nodes communicate with each other until the final information reaches the central hub to be sent to the Cloud and displayed to the user in an HTML page. The Linux embedded computing system selected to be the hub was Raspberry Pi (RPi) model 3, which communicates with Arduino Uno R3-based nodes via XBee Zigbee, a solution based on the IEEE 802.15.4 standard with straight-line communication distance up to 50 m.

ZigBee is generally applied in external sensor networks where there is no telephony or WiFi coverage and there are many sensor nodes. The development board has 16 channels in the 2.4 GHz band, each with 5 MHz of bandwidth. The transmission rate is 250 Kb/s in spokes from 1 to 100 m. ZigBee also presents low cost, low energy consumption and easy implementation. Routers, coordinators and end devices are required on a ZigBee network. Coordinator initiates the communication by selecting the channel and identifier of the node to be read. Router function is to allow integration of other end devices into the network. ZigBee technology disadvantages are the susceptibility to signal interference from other on-site emitters, low coverage and signal loss in industrial environments [10].

A wireless sensors network with communication via ZigBee and propose the use of Arduino or RPi as interface sensors is described in [11]. When intelligent sensor nodes receive antennas for wireless communication and an Internet network IP, these nodes can synchronize different devices and share data between them or with a central hub, ensuring new possibilities for efficient communication in sensors networks. Aiming to reduce the power consumption of these nodes, which may send rates every minute, sources such as solar PV have been applied to recharge batteries.

The research in [12] addresses monitoring applied to PV plants using the Message Queuing Telemetry Transfer (MQTT) protocol. MQTT is based on the publish/subscribe principle, where data are requested by the client (user/subscriber) to the MQTT broker, which receives data from the publisher (sensor nodes). Each node performs voltage, current and PV data acquisition.

The authors in [13] propose a lightweight IoT service mashup middleware (a way to compose a new service from existing services) based on REST-style architecture for IoT applications using publish/subscribe based messages. The system was applied on a coal mine safety monitoring and control automation.

A situation-aware coordination approach for dynamic IoT services is proposed by [14]. The authors focus on focusing on defining event pattern with event selection and consumption strategy and proposing an automaton-based situational event detection algorithm. The event-condition-action is used to coordinate the IoT services and typical scenarios of IoT services coordination for smart surgery process are also illustrated. Finally, the measurement and analysis of the platform's performance are reported.

A current and voltage monitoring system to obtain PV modules power is developed by [15]. The system is also able to protect the load against overvoltage using relays, which disconnect the power supply. The used embedded system was the ESP 8266 microcontroller, which communicates directly with the Internet via an embedded WiFi antenna. Arduino is used to read the sensors, which communicates with the ESP 8266, responsible for data storing on an SD card; part of the data are sent to the developed Web environment. Database is the free tool Thingspeak. ESP 8266 transmits each reading to the Thingspeak base every 15 s.

In addition to the WiFi network, LoRa has been gaining space as a means of communication in wireless sensor networks. LoRa is a network patented in France, with long-range (2–15 km), low

power consumption (microamperes) and license of use in frequencies below 1 GHz. The authors of the article in [16] implement an Arduino transmitter and a RPi receiver using embedded LoRa antenna for voltage, current, temperature and battery data monitoring in renewable energy plants.

The system proposed by [17] monitors meteorological and inverter data in a grid-connected PV plant. The hardware is based on PCDuino (discontinued) and stores data on a SD card as well as sends over the Internet. PCDuino combines Arduino with RPi, operates on Linux and features HDMI interface, USB, input/output pins, SD card slot, 512 MB of DRAM memory, 1 GHz processor, six channels AD, UART, SPI, I²C. Data from 20 PV plants are monitored from 2016 to 2017. Additionally, authors present a bibliographic review comparing 23 projects. Surveyed projects implement low-cost hardware solutions with PIC microcontrollers and Arduino, but also commercial dataloggers. Data are usually stored locally via serial interface (RS232) or dedicated PC and those who have implemented wireless communication have used ZigBee or Radio Frequency. Authors claim that commercial dataloggers have disadvantages such as high cost, low memory, connection to a PC, low programming flexibility requiring proprietary interface software and offer few sensor inputs. In the research, wind speed and direction, irradiance, PV and ambient temperature sensors were connected to the PCDuino via low cost conditioning circuits. Data is then sent to the Internet via RJ45 wired connection to the Internet modem. The used inverters allow PV voltage and current to be read via Bluetooth, from a tool available with the product.

The proposed project in [18] also implement RPi to read PV voltage and current data from grid-tie inverters. Inverters with Bluetooth communication require a Bluetooth receiver – RPi connection; for those with RS 485 interface, an Ethernet-RS 485 converter must be used together with RPi.

In the configuration proposed by [19], PV plant inverters communicate via ZigBee with a central system, which sends the monitored data to a PC via Ethernet cable. Monitoring is then done via Web browser using HTTP protocol and HTML page embedded in the STM32F407 microcontroller. In Table 1 a comparison between the discussed projects in the literature review is shown.

From the year 2014, embedded systems and wireless data transmission have been increasingly implemented for PV plants and meteorological variables monitoring, instead of using commercial dataloggers with LabVIEW™ interface, frequently applied until 2013. Network systems are the majority, since the modules are inexpensive – ESP 8266: US\$ 4 and RPi: US\$ 35 (quotation in 2018) – and can be implemented in distributed PV plants, sending data to centralized devices – hub or gateway – which store data on a local embedded server and/or send them to a Cloud database. However, most of the researched systems do not use WiFi for sending data to the Cloud. WiFi is widespread and available in residences and urban locations.

Information models and communication systems for various microgrid categories are standardized by International Electrotechnical Commission (IEC) in IEC 61850-7-420: Basic communication structure – Distributed energy resources logical nodes [20]. Considering installation costs and data rate requirements, wireless Ethernet (WLAN) technology can be considered as one of the potential communication technologies for the microgrid applications [21].

The wireless technology is less expensive compared to wired, easy to install (no cables needed) and more portable. However, it suffers interference from the surroundings, has lower data rates and lower data security. Hence, it is important to check for reliability when implementing WLAN in microgrids.

Table 2 shows a comparison between the wireless technology used in the present project (WiFi 802.11n) and other already available options for IoT. The specified values for coverage, data rate, QoS, bandwidth and latency depend on system construction, protocol used and local of installation [22,23]. To analyse deployment costs more deeply, price for hardware and software, data transfer charges for licensed technologies, installation and maintenance costs should be considered. Additionally, for the end-device final cost, battery cost, case, cables and extra modules such as GPS and sensors should also be considered.

Although Sigfox and LoRa employ unlicensed spectra and asynchronous communication protocols and can bounce interference, they cannot offer the same Quality of Service (QoS) provided by

Table 1
Comparison between DATS and IoT monitoring systems from the literature review.

Reference	[19]	[6]	[16]	[12]	[15]	[18]	[9]	[17]
Monitoring Network	Yes	No	No	Yes	Yes	Yes	Yes	Yes
Data Acquisition	Grid-tie PV inverters	Arduino	RPI	RPI	Arduino	Grid-tie PV inverters	Arduino UNO	Grid-tie PV inverter and PCDuino
Data processing and estimated cost	STM32F407 CC2530ZNP (US\$ 48)	Arduino and PIC (US\$ 25)	Arduino, RPI, LoRa (US\$ 56)	RPI (US\$ 35)	Arduino and ESP 8266 (US\$ 8)	RPI (US\$ 38)	RPI, Arduino, XBee (US\$ 99)	Pcduino (US\$ 62)
Communication between embedded systems	ZigBee	Serial	LoRa	Smart Utility Network (SUN)	Serial	Bluetooth or RS485/Ethernet	XBee Zigbee	Bluetooth
Number of sensors	4 per node	16	4	2 per node	2 per node	–	2 per node	4 per node
Measured variables	DC and AC voltage and current	Temperature	Voltage, current and temperature	PV string DC voltage and current	PV string DC voltage and current	PV string DC voltage and current	PV module DC voltage and current	Temperature, irradiance, wind speed and direction, DC and AC voltage and current
PV monitoring	Grid-tie PV string	Not applied	PV and wind measuring station	Off-grid PV string	Grid-tie PV string	Grid-tie PV string	Off-grid PV module	Grid-tie PV string and weather station
Sampling time	–	5 min	–	–	–	–	10 sec	5 min
Data storage	Dedicated PC	Local SD card	MongoDB database	MQTT local server on RPI	Local SD card and ThingSpeak database	Inverter memory and local database	Local database	Local SD card and database
Internet data transmission	Ethernet	–	LoRa	MQTT via WiFi	Ethernet	–	–	Ethernet
Data visualization	Local server on STM32	Excel on dedicated PC	Local server on RPI	Android App	ThingSpeak Web	Local server	HTML page on RPI	Local server on PCDuino

Table 2

Comparison between wireless technologies for IoT.

Wireless technology	Coverage	Data rate	QoS	Bandwidth	Latency	Deployment cost	End-device cost
WLAN-WiFi (IEEE802.11n)	<100 m	<300 Mbps	High	20–40 MHz	75 ms	> \$ 20/WiFi router	~ \$ 3
LoRa LoRaWAN (LPWAN)	2–10 km (urban); 40 km (rural); <400 km (line of sight)	EU: 300 bps – 50 kbps US: 900– 100 kbps	Low	125 kHz and 250 kHz	82 ms	> \$ 100/gateway > \$ 1000/base station	> \$ 10 (module + antenna)
SigFox (LPWAN)	3–10 km (urban); 30–50 km (rural); 1000 km (line of sight)	10–100 bps	Low	100 Hz	82 ms	~ \$ 200 (Gateway and base station needed) Data transfer charged (<\$1/month)	\$ 5–10 (microcontroller + radio)
WPAN-ZigBee (IEEE802.15.4)	10–100 m	250 kbps	Medium	2 MHz	50 ms	~ \$ 100 (coordinator, router module)	> \$ 20
NB-IoT (LPWAN)	1 km (urban); 10 km (rural)	200 kbps	High	200 kHz	75 ms	Antenna coverage needed; Licensed LTE; Data transfer charged (<\$ 1/month for 100 kb)	\$7–12
LTE-M (Cellular)	<35 km antenna (worldwide)	<1 Gbps	High	1.4– 20 MHz	50 ms	Antenna coverage needed; Licensed LTE; Data transfer charged (<\$ 3–5/month for 1 MB)	\$10–15

Narrowband IoT (NB-IoT). NB-IoT employs an LTE-based synchronous protocol, which is optimal for QoS at the expense of cost [24]. Using the MQTT protocol via WiFi, high QoS (2) can be achieved.

With the technological evolution, tendency is IoT modules price and size to decrease and processing power and number of available embedded peripherals to increase.

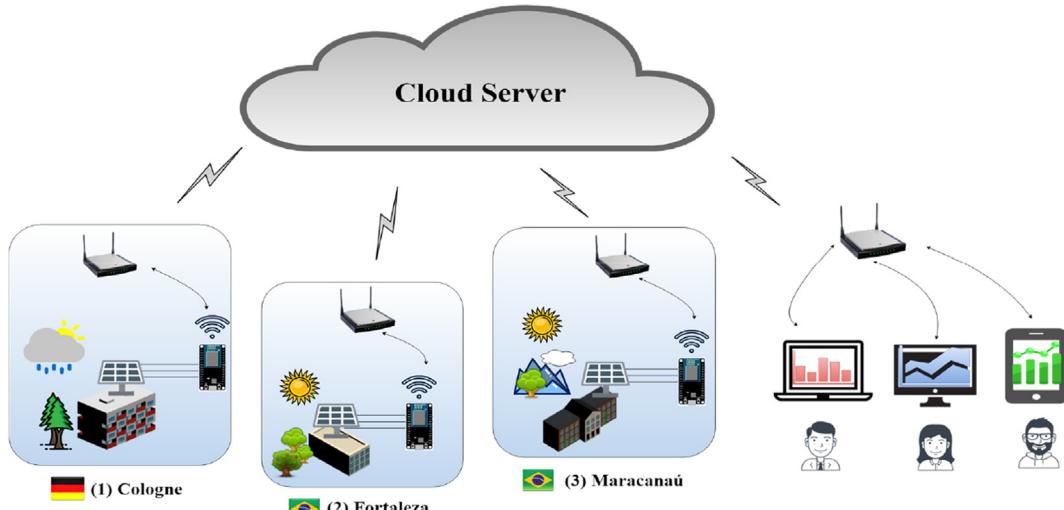
Hence, we propose an IoT monitoring network WiFi connected to the Cloud to provide important data on PV analysis and fault identification worldwide accessible. We present also a complete monitoring system: monitoring of all the required variables, data transmission to the Cloud and Web Monitor for remote and real-time data visualization. The system provides meteorological data and PV module temperature; both analyzed together with electric generation data from the PV inverters. We apply only the ESP 32 board (US\$ 8) for data sensoring, processing and transmission. Considering the literature review, the systems fail in meeting all these points together.

Thus, literature review has shown absence of database of PV modules temperature measurements worldwide, an important variable for the PV electricity generation efficiency. As a conse-

quence, usually researchers have applied statistical techniques to estimate these data from other parameters such as solar irradiance, wind speed, ambient temperature, PV cell thermal coefficient [25], depending on the applied model. The Faiman model is compared to others and validated for five different regions of the world [26]. There is a vast literature available on research on efficiency modelling of PV modules. However, after applying the models of Duffie and Beckman [27], Ross [28] and Chenni [29] with data measured at the Laboratory of Alternative Energies, Federal University of Ceará (LEA-UFC), it was verified that none fits the meteorological conditions of Fortaleza, Brazil [30]. Hence, PV module temperature monitoring to obtain measured data is necessary.

In the literature review, we could also not find a paper similar to ours, which has the following characteristics:

- Addresses design, development, implementation and testing of a multiple site IoT network along with sensors signal conditioning circuits for:
 - Individualized PV module temperature monitoring;
 - Meteorological data (irradiance, wind speed, relative air humidity and ambient temperature);

**Fig. 1.** Developed IoT monitoring network.

- Free software and hardware;
 - Real-time data available in the Cloud for remote access;
 - Web Monitor for real-time and daily charts visualization from any computational device Internet connected.

3. Monitored PV plants

PV modules temperature measurements have been usually neglected in Brazil, where 12 h of sun per day are available and average daily irradiation is 5.5 kWh/m². In the country's rainy season (January to June), there is a reduction in the solar irradiance, which reduces PV modules temperature but also the electricity generation.

With such motivation, we propose the design and development of an IoT modular system to compose a worldwide monitoring network ([Fig. 1](#)) focused on meteorological and PV modules temperature data, available to researchers from partner institutions

Another motivation for monitoring individual PV modules is for fault detection. This concept is based on cost reduction, modularity and practicality of installations, since PV plants can have hundreds of modules. In our project, we use IoT ESP 32 and ESP 8266 modules in Fortaleza and Maracanaú, in Brazil (Fig. 1 (2) and (3)); and the first international unit was installed in Cologne-Germany (Fig. 1 (1)). The IoT PV network is open and ready to be expanded.

IoT monitoring network operation follows the following steps:

1. Data acquisition with ESP 32/ESP 8266;
 2. Data sending from ESP to the Cloud server database via embedded WiFi;
 3. Data visualization via a developed PHP Web Monitor.

Each of the three PV plants under investigation has a specific table, reserved in the same MySQL database of a Cloud server for the storage of incoming data. Hence, the three plants work in parallel sending data from all the sensors every minute, for 24 h per day. IEC 61724:1998 standard specifies that the sampling interval for parameters that vary directly with the irradiance should be 1 min or less. For parameters having larger time constants, an arbitrary interval can be specified between 1 and 10 min (British Standard, 1998).

Table 3 shows the implemented PV plant sensors and the respective embedded systems responsible for data acquisition and transmission. Maracanaú's plant monitors temperature data at the center and at the edge of the PV module and solar irradiance using ESP 32; ESP 8266 is responsible for ambient temperature, relative humidity and wind speed data. Fortaleza's plant monitors the same parameters, except the temperature at the edges of the PV modules. In Cologne, only temperature monitoring at the center of the PV modules was implemented.

In the following subsections the three mentioned plants are described; section 4 describes components and methodology for the IoT monitoring network development.

3.1. PV plant in Cologne – Germany

The first international IoT PV network unit was installed in Cologne-Germany at the Technical University of Cologne.

Table 3

Table 3 IoT network embedded systems and sensors.

(Technische Hochschule Köln) and developed during doctoral research of the first author. IoT concept with ESP 32 and ESP 8266 was chosen aiming modularizing, costs reduction and increase of the data processing speed. For this first plant, IoT ESP 32 module was set up and implemented to collect temperature data at the center of five PV modules (Fig. 2).

Cologne plant consists of five polycrystalline silicon modules. For PV modules temperature monitoring, five PT100 sensors are connected to the IoT ESP 32 embedded system (Fig. 3). For ensure fixation, a bi-component adhesive resistant to temperatures of -20°C to 90°C was used. PT100 sensor developed conditioning circuit is detailed in section 4.

ESP 32 is then powered by a 220 V_{ac}/5 V_{dc} source and temperature sensor cables are distributed from the ESP 32 to the respective PV modules.

3.2. PV plant in Fortaleza – Brazil

ESP 32 was implemented in Fortaleza for temperature monitoring of six grid-connected PV modules of LEA-UFC. PT100 sensors are attached at the center of each module, as shown in Fig. 4. A diagram illustrating the IoT monitoring system of this second plant can be seen in Fig. 5. A LP02 pyranometer was also connected to the ESP 32; the developed sensor conditioning circuit is detailed in section 4.

DHT11 sensor connected to an ESP 8266 IoT module was also implemented at the Fortaleza's plant to measure ambient temper-

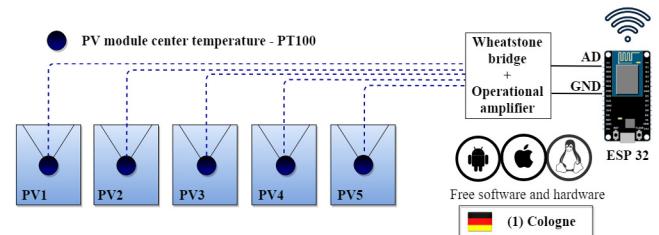


Fig. 2. IoT PV monitoring in Cologne-Germany (1).

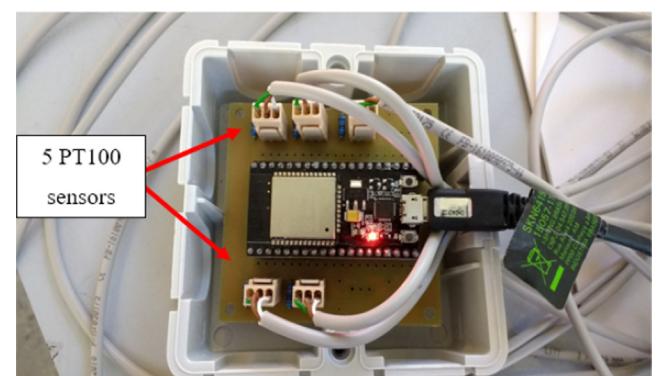


Fig. 3. 5 PT100 connected to ESP 32.

	Temperature at the center of the PV module	Temperature at the edge of the PV module	Solar Irradiance	Ambient Temperature	Relative humidity	Wind speed
Embedded system	ESP 32	ESP 32	ESP 32	ESP 8266	ESP 8266	ESP 8266
Fortaleza	6 PT 100	-	Pyranometer LP 02	DHT11	DHT11	Anemometer NRG #40C
Maracanaú	5 PT 100	3 PT 100	Pyranometer LP 02	DHT11	DHT11	Anemometer NRG #40C
Cologne	5 PT 100	-	-	-	-	-

ature and relative air humidity. ESP 8266 also measures wind speed from a cup anemometer connected to the available AD converter pin.

3.3. PV plant in Maracanaú – Brazil

The main difference between Maracanaú's and Fortaleza's plant is the peak power: Maracanaú has 1.35 kW_p , while Fortaleza has 1.5 kW_p . Additionally, PV modules are oriented South with slope of 10° in Maracanaú and North with slope of 5° in Fortaleza. An important innovation in Maracanaú was the use of sensors to measure the temperature of the PV modules edges. According to the literature review, studies have verified a difference of $1\text{--}2^\circ\text{C}$ between center and edge of PV modules.

Fig. 6 shows an illustrative diagram of Maracanaú's plant. ESP 32 was programmed for reading the pyranometer in the inclined plane and for monitoring the temperature of the five PV modules (five sensors at the center and three sensors at the edges: left, right and center of the PV string).

The implemented PT100 sensors are shown in Fig. 7. The sensors are made from platinum, have two pins, are TO 92 encapsulated and measure the range from -50 to $+150^\circ\text{C}$. After soldering on cables, PT100 were fixed at the back of the PV modules with temperature-resistant adhesive and insulated with silicone.

Maracanaú's IoT monitoring system consists also of the IoT ESP 32 and ESP 8266 embedded systems. ESP 32 is responsible for monitoring PV modules temperature and solar irradiance data, while the ESP 8266 monitors ambient temperature and relative humidity using DHT11 sensor, and wind speed. Fig. 8 shows the grid-tie PV inverter, string box for power protection and ESP 32 enclosure, installed at the Laboratory of Electronics and Embedded Systems (LAESE), Federal Institute of Ceará (IFCE), Maracanaú Campus, 20 m away from the PV plant. ESP 8266 was installed externally, close to the PV string, to reduce the distance to the anemometer, minimizing cable resistance losses, and to perform environment variables measurements close to the PV plant.

Section 4 describes the embedded systems, conditioning circuits for temperature, wind speed and solar irradiance sensors, as well as the online Web Monitor page; these components are part of the proposed IoT monitoring network.

4. IoT monitoring embedded systems and sensors signal conditioning

4.1. Grid-tie PV inverter

ANEEL regulates that grid-connected inverters must be certified by the National Institute of Metrology, Quality and Technology (INMETRO). The single-phase inverters used in Fortaleza and Mar-



Fig. 4. PT100 sensors at the center of PV modules and anemometer at the modules plane.

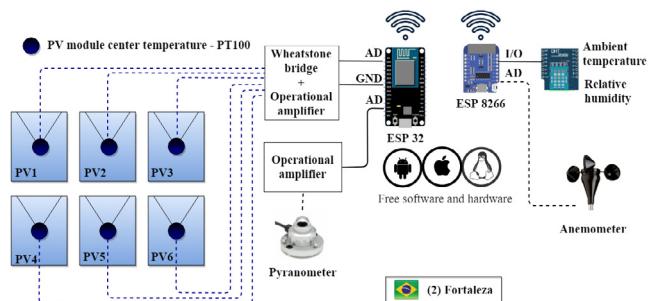


Fig. 5. IoT PV monitoring in Fortaleza-Brazil (2).

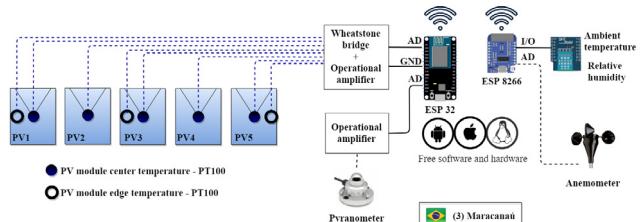


Fig. 6. IoT PV monitoring in Maracanaú-Brazil (3).



Fig. 7. PT100 sensors for PV modules temperature measurement.



Fig. 8. Grid-tie PV inverter, String Box and ESP 32 box.

canaú are model 1500-NS [31], produced by a Brazilian manufacturer with the following characteristics:

- 97.8% efficiency;
- Harmonic distortion rate less than 5%;
- WiFi communication system homologated by the Brazilian National Telecommunications Agency (ANATEL);
- Maximum Power Point Tracker (MPPT) efficiency higher than 99.9%;

- Compliance with standards:
 - ABNT-NBR-16149: characteristics of the interface with the grid;
 - ABNT-NBR-16150: conformity test at the interface with the power grid;
 - ABNT-NBR-IEC-62116: anti-islanding test.
- Five-year guarantee for manufacturing defects.

In addition to PV voltage, current, power and electricity generation data, information on inverter status and error logging can also be downloaded in a data history worksheet from PHB online web page. Considering that the mentioned data are supplied by the inverter, we decided not to install redundant sensors in the IoT monitoring network plants, reducing costs and eliminating failure on faulty electronic components. Furthermore, to acquire voltage and current data, it would be necessary to install sensors at the PV string output, which before being connected to the inverter, goes through the protection box (string box). As the system is inspected by the local distribution company, which requires installation standards, this also avoids possible technical problems.

To measure the other variables – PV module temperature, solar irradiance, ambient temperature, relative humidity and wind speed – the embedded systems ESP 32 and ESP 8266 were installed, and auxiliary circuits were designed for sensors conditioning. Hardware and software development are described in the following subsections.

4.2. IoT embedded system ESP 32

IoT ESP 32 Cloud on Chip [32] module with free software and hardware was launched in September 2016. ESP 32 is a low-power, dual-core, dual-mode WiFi/Bluetooth microcontroller (MCU). The MCU has 240 MHz frequency and processing power of 600 Dhrystone Million Instructions per Second (DMIPS).

ESP 32 implementation in our project was done together with the launch of this module in the market. Hence, at this time, documentation was still incomplete and being constantly updated. New Application Programming Interface (API) continue to be released and the user community publishes questions, enhancements and code fixes on the GitHub online platform daily (Espressif Systems, 2017b). Due to the recent launch of this module, the implementation of ESP 32 in a real-time IoT monitoring network of PV modules can be considered as one of the innovations proposed in this project.

ESP 32 is designed for mobile applications, wearable electronics and IoT, featuring low-power chips, including multiple power modes and dynamic power scaling. For example, in a low power sensor hub application scenario, ESP 32 is periodically awakened only when a programmed condition is identified. In Table 4 ESP 32 current consumption for various forms of operation is shown.

To start a project with ESP 32 (Fig. 9), it is necessary:

- PC with Windows®, Linux or Mac OSX operating system;
- Toolchain to create application for ESP 32;
- Espressif IoT Development Framework (ESP-IDF) that essentially contains APIs for ESP 32 and scripts to operate the Toolchain;
- A text editor for writing C programs, for example, Eclipse;
- ESP 32 development board and a USB-micro to USB cable to connect it to the PC.

The user can also compile and write ESP 32 from a Shell environment with the ‘make’ command group, without the need for a development interface. However, Eclipse has proved to be very practical, reducing project development time.

A tutorial on how to configure the toolchain on the Windows® environment as well as the initial step-by-step to program ESP 32 can be found in [34,35]. ESP 32 programming allows the use of several languages and development interfaces. For being easier,

Table 4
ESP 32 current consumption [33].

Parameter	Typical current consumption
WiFi data transmission	180–240 mA
WiFi data reception	95–100 mA
Bluetooth BLE data transmission	130 mA
Bluetooth BLE data reception	95–100 mA
Sleep mode (active CPU, active WiFi connection but WiFi circuitry off)	240 MHz: 30–50 mA 160 MHz: 27–34 mA 80 MHz: 20–25 mA (conventional) 2 MHz: 2–4 mA (low speed)
Light-sleep mode (standby CPU, active RTC, WiFi circuitry off)	0.8 mA
Deep-sleep mode (active RTC, inactive WiFi connection and WiFi circuitry off)	10–150 µA
Power-off mode	0.1 µA

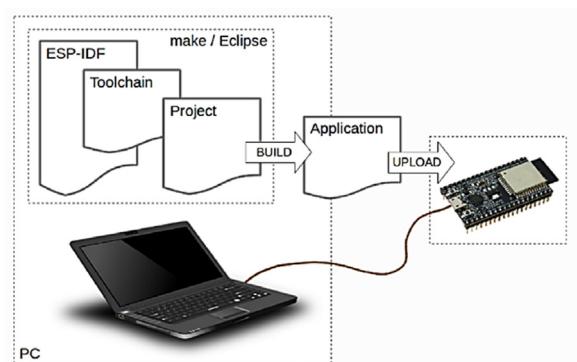


Fig. 9. ESP 32 project development [34].

the most commonly used is the Arduino interface along with libraries developed for this environment.

Because both ESP hardware and software are open, user community can easily implement design enhancements, contributing to the evolving design and support material for new users. IoT module manufacturer provides Espressif IoT Development Framework (ESP IDF), which is a set of APIs to support ESP 32 embedded peripherals [32].

The following subsections describe the developed conditioning circuits for reading the temperature and irradiance sensors.

4.2.1. PT100 temperature sensor conditioning circuit

PT100 sensor is a resistance thermistor, also called Resistance Temperature Detector (RTD). Table 5 shows the resistance values according to temperature. The circuit was designed for temperature reading from -30°C to 160°C .

For the conversion of the resistance value into voltage, necessary for the ESP 32 AD, it was necessary to develop a signal conditioning circuit. The sensor can be connected in three ways (Fig. 10): two (a), three (b) or four cables (c). Three-wire is the most commonly used connection and has better accuracy than the two-wire connection. Four-wire connection is usually used in laboratory and standard sensors.

In our project, the three-wire connection was implemented. For voltage measurement according to the PT100 resistance variation, we used a Wheatstone bridge (Fig. 11 (a)), which allows the measurement of the value of an unknown electrical resistance. The circuit is then balanced following Eq. (2). Hence, when the PT100

Table 5
PT100 resistance and temperature.

Temperature (°C)	Resistance (Ω)	Temperature (°C)	Resistance (Ω)
-30	88.22	70	127.08
-20	92.16	80	130.90
-10	96.09	90	134.71
0	100.0	100	138.51
10	103.9	110	142.29
20	107.79	120	146.07
30	111.67	130	149.83
40	115.54	140	153.58
50	119.40	150	157.33
60	123.24	160	161.05

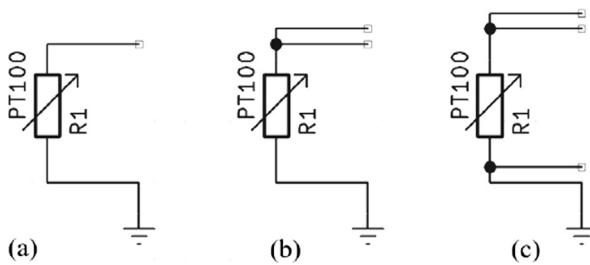


Fig. 10. PT100 connection possibilities.

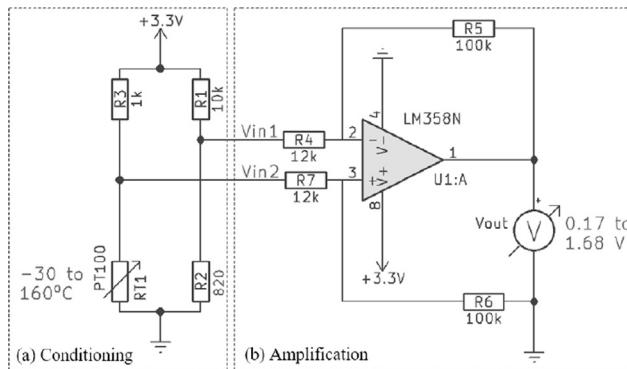


Fig. 11. Input and output ranges of the PT100 conditioning circuit.

(RT1) is 82 Ω, the output voltage between points V_{in1} and V_{in2} is 0 V.

$$R1.RT1 = R2.R3 \quad (2)$$

$$10k.82 = 820.1k$$

As the resistance increases with temperature, a small voltage variation can be measured between V_{in1} and V_{in2}. Considering that for each degree the corresponding variation is only 0.385 Ω, it is necessary to amplify the signal to be read by the ESP 32 AD. The differential amplification circuit was developed with the IC LM358N (Fig. 11 (b)). According to the theory, if R4 = R7 and R5 = R6, the output voltage can be calculated according to Eq. (3).

$$V_{out} = (V_{IN2} - V_{IN1})(R5/R4) \quad (3)$$

When the resistance is 88.22 Ω, equivalent to a temperature of -30 °C, output voltage is 0.17 V, which is the minimum circuit voltage (Fig. 11). If the temperature is 160 °C, PT100 resistance is 161.05 Ω and the maximum output voltage is 1.68 V.

PT100 sensor circuit output values from 0.17 to 1.68 V are designed to meet the measurement range of the analog-to-digital

converter embedded in the ESP 32; in this range the voltage-bit graph is approximately linear. The PT100 conditioning circuit has the sensitivity of 8 mV/°C.

For PT100 sensors reading, a multiplexing circuit has been developed, in which eight inputs are read by the same AD channel through the 74HC4051 or 4051 IC (Fig. 12). This strategy allows to expand the number of sensors connected to the ESP 32. The multiplexer IC is powered with 3.3 V supplied by the ESP 32.

4.2.2. Pyranometer sensor conditioning circuit

After manufacturer calibration, each sensor has a different output signal. In the case of the used LP02, the value is in $\mu\text{V}/\text{W}/\text{m}^2$. Hence, Fortaleza's sensor has a sensitivity of $18.56 \mu\text{V}/\text{W}/\text{m}^2$, as shown in Fig. 13, and Maracanaú's pyranometer has output of $15.07 \mu\text{V}/\text{W}/\text{m}^2$. Second-class LP02 pyranometer specifications according to ISO 9060 standard are described in Table 6.

Since LP02 pyranometer output signal is too low to be read by the IoT embedded system AD, which interprets values with 1 mV

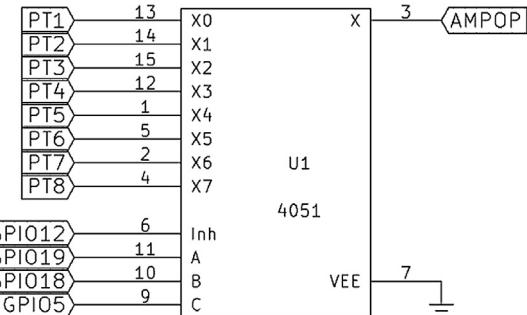


Fig. 12. 8 channel AD multiplexer 8 – 74HC4051.



Fig. 13. Pyranometer in inclined plane at Fortaleza's plant.

Table 6
LP02 pyranometer specifications.

Description	Specification
Response time (95%)	18 s
Non-linearity error	±1%
Measuring range	0–2000 W/m ²
Sensitivity	7–25 $\mu\text{V}/\text{W}/\text{m}^2$
Operating temperature	15 $\mu\text{V}/\text{W}/\text{m}^2$ nominal -40 to 80 °C
Power supply	Not required, passive sensor
Field of View Angle	180°
Protection class	IP67 (submersion up to one meter per 30 min)

Table 7

Connection pins of the AD pre-amplifier capacitors. Adapted from [34].

VDD3P3_RTC			
SENSOR_VP	5	I	GPIO36, ADC_PRE_AMP, ADC1_CH0, RTC_GPIO0
Note: Connects 270 pF capacitor from SENSOR_VP to SENSOR_CAPP when used as ADC_PRE_AMP.			
Name	No.	Type	Function
SENSOR_CAPP	6	I	GPIO37, ADC_PRE_AMP, ADC1_CH1, RTC_GPIO1
Note: Connects 270 pF capacitor from SENSOR_VP to SENSOR_CAPP when used as ADC_PRE_AMP.			
SENSOR_CAPN	7	I	GPIO38, ADC1_CH2, ADC_PRE_AMP, RTC_GPIO2
Note: Connects 270 pF capacitor from SENSOR_VN to SENSOR_CAPN when used as ADC_PRE_AMP.			
SENSOR_VN	8	I	GPIO39, ADC1_CH3, ADC_PRE_AMP, RTC_GPIO3
Note: Connects 270 pF capacitor from SENSOR_VN to SENSOR_CAPN when used as ADC_PRE_AMP.			

accuracy, we developed a signal conditioning based on an operational amplifier (AMPOP) to be applied in both PV plants. To delimit the amplification range, it is necessary to consider the irradiance limit values at the plant site. For Fortaleza and Maracanaú, for example, the irradiance can have peaks up to 1200 W/m²; in this case, the output signal of the pyranometer is 22.3 mV.

Considering that the AD has an offset of 11 mV and can be configured to read values up to 3.3 V, the circuit was designed so that the minimum and maximum amplifier output voltages were in the range of 11 mV–3.3 V. To avoid offset imprecision errors, we decided not to use the minimum limit value (close to 11 mV). The maximum amplification limit is designed for an irradiance value of 1200 W/m², that is, the circuit output signal is 22.3 mV (pyranometer value) multiplied by the gain of the amplifier circuit (10 1), resulting 2.23 V. However, the LM358N limits the amplified voltage in Vin minus 1.5 V (datasheet specified). Since the supply voltage is 3.3 V, the maximum amplified value is around 1.8 V. In Fig. 14 the developed circuit and the output amplified voltage (14 mV–1.81 V) for the pyranometer input range (22 μV–22 mV) are shown. The pyranometer conditioning circuit has the sensitivity of 2 mV/W/m².

4.2.3. ESP 32 AD converter

One of the properties of the analog channels of AD converters is attenuation, which is a voltage scale factor. Usually in ESP, the input range is 0–1 V, but with different attenuations. Hence, via firmware, the user can configure the input voltage in the four ranges available from the following parameters:

- ADC_ATTEN_0db – 0 to 1 V
- ADC_ATTEN_2_5db – 0 to 1.34 V
- ADC_ATTEN_6db – 0 to 2 V
- ADC_ATTEN_11db – 0 to 3.6 V

ESP 32 integrates a very low noise analog pre-amplifier connected to the AD converter. Amplification rate is given by the size of a pair of sampling capacitors that are placed externally. Using larger capacitors, sampling noise is reduced, but the stabilization time is increased. Table 7 shows the pins for the two 270 pF capacitors connection; ‘ADC_PRE_AMP’ function is used to enable the pre-amplifier on these pins. Hence, the General-Purpose Input/Output pins GPIO 36, 37, 38 and 39 are reserved for this function, unlike other GPIOs, which can be configured via firmware to perform other functions.

The mentioned 270 pF capacitors are part of the Espressif development board original circuit (Fig. 15).

Amplification rate is also limited by the amplifier reaching the peak with a gain of about 60 dB (decibels). 60 dB means, considering Eq. (4):

$$Lx = 20 \cdot \log_{10} \left(\frac{V_{out}}{V_{in}} \right) dB \quad (4)$$

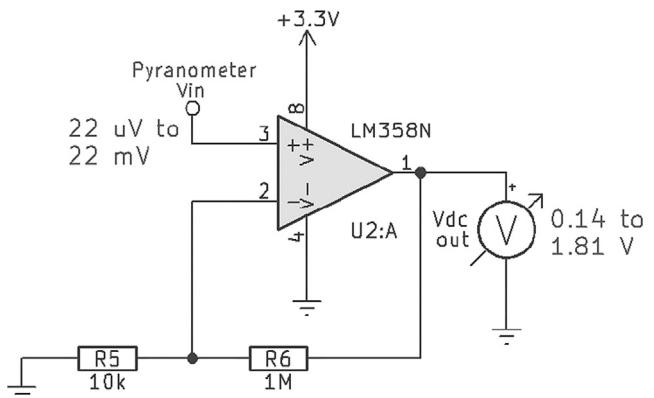


Fig. 14. Input and output ranges of the pyranometer amplifier circuit.

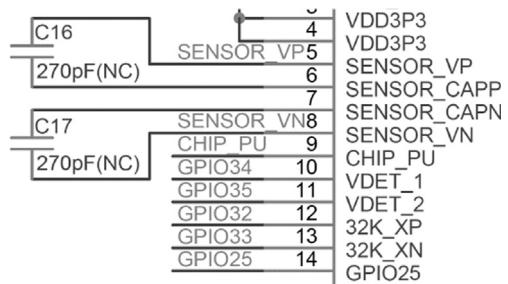


Fig. 15. Low noise pre-amplifier pins connected to ESP 32 AD [34].

wherein, if $Lx = 60dB$, solving the equation, the embedded pre-amplifier has a gain of 1000 times. If the analog input voltage on the AD pin is 0.001 V, the “pre-amp” amplifies the value to 1 V. The Ultra Low Power (ULP) ESP 32 coprocessor is also designed to measure voltages while operating in standby mode to allow low power consumption; the CPU can be awakened by interruption or by other triggers.

ESP 32's ADs have 12-bit resolution and support measurements on 18 channels (firmware-enabled pins). APIs released until 2017 support only nine channels (GPIOs 25–27, 32–36 and 39) and there is an expectancy for new APIs having other nine channels available only if the embedded WiFi module is disconnected, because some of these pins are used simultaneously by the embedded WiFi module and the second AD converter. It is also recommended that reading on both AD converters is not performed while the WiFi module is transmitting data, as it may cause changes in voltage [36].

To check the linearity of the AD converter voltage-bit curve and identify the scale that best fits the proposed PT100 circuit, it was necessary to calibrate the channel. Calibration is necessary to minimize measurement errors and because the voltage-bit curve varies according to the AD converter due to the constructive characteristics of the material. However, it is not necessary to calibrate whenever using a new ESP 32; the user can consider the linear curve with an error of ± 1 LSB (least significant bit), that is ± 1 bit in the range of 4096 bits to the 12-bit AD. This value seems negligible, however, when looking the curve on Fig. 16, it is verified with the help of two dashed lines that the curve is approximately linear from 0.11 V (approximate offset) to 2.5 V; from 2.5 V to approximately 3.1 V would be necessary to represent this interval by a second equation.

It is possible to select the resolution of 10 or 12 bits, besides being able to configure the attenuation of 0; 2.5; 6 or 10 dB. As the PT100 circuit was designed to measure 190 different temperature values (−30 to 160 °C), it was decided to implement the 12-bit

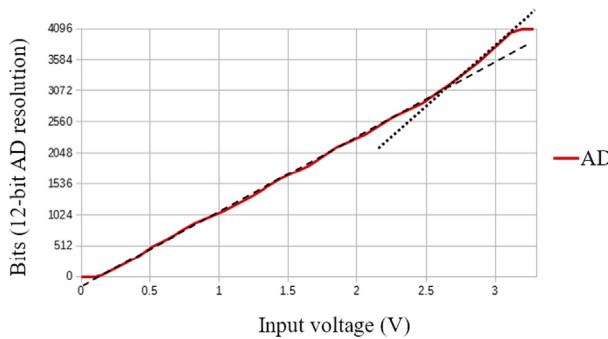


Fig. 16. Linearity of the ESP 32 AD converter - voltage-bit curve [37].

resolution (4096 bits) to obtain better accuracy for each measurement. To calibrate the AD, the following steps and procedures were followed:

1. Generation of AD graph with desired number of bits and attenuation;
 - a. Methodology: with stable and accurate power source, list bits read according to applied voltage (reading three decimal places for better precision);
2. Use linear part of the line and define the initial and final value; if necessary, generate more than one line;
3. Define the operational amplifier gain resistor so that the minimum value of the output voltage is within the linear range of the AD; as $V_{out} = \text{Gain} \cdot (V_{in2} - V_{in1})$, when modifying the resistors of the Wheatstone bridge or the amplifier, the output voltage is multiplied either by voltage difference or gain, respectively;
 - a. Observe minimum value of circuit voltage output according to the minimum limit of the temperature range (min: -30°C);
4. Verify that the output voltage for the maximum value with defined gain does not exceed AD linear range:
 - a. 0 dB: 1.037 V – 4070 bits;
 - b. 11 dB: 3.03 V – 3766 bits (used).

Due to the saturation voltage limitation of the LM358N Operational Amplifier, the maximum output voltage is equal to: $V_{in} - 1.5\text{ V}$.

The voltage-bit graph for 0 dB (0–1 V) attenuation would require a higher gain circuit to improve accuracy out of the 3.3 V amplifier voltage range. Since the amplifier saturation voltage is subtracted by 1.5 V and the circuit supply voltage from ESP 32 development board is 3.3 V, the maximum amplified voltage should be 1.8 V. The attenuation of 11 dB is the largest available allowing the longest range of measurements. The voltage-bit plot was approximately linear from 0.165 V (35 bits) to 3.03 V (3766 bits), as shown in Fig. 17. However, considering some imperfections in the curve, we decided to divide the graph into several straight lines for AD calibration during project development.

4.3. Over the air (OTA) updates

Over the Air (OTA) update tool allows a device to update its own firmware based on received data while the main firmware is running, for example via WiFi or Bluetooth. OTA requires the configuration of the device internal Partition Table with at least two partitions “OTA app slot” (that is, ‘ota_0’ and ‘ota_1’) and an “OTA Data Partition”.

OTA operation functions write a new application firmware image to any OTA application slot that is not currently being used for Startup. After the image is checked, OTA data partition is

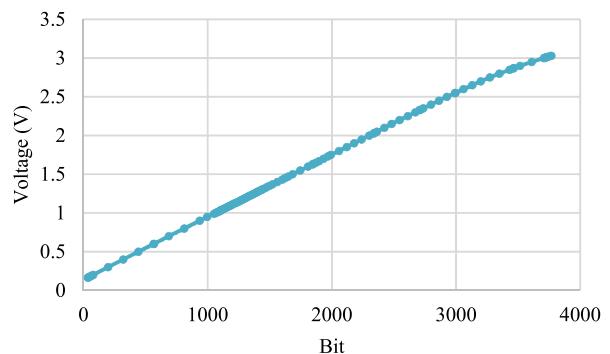


Fig. 17. ESP 32 AD calibration performed.

updated to specify that this image should be used for the next Startup.

An OTA data partition ('data' type, 'ota' subtype) must be included in the Partition Table of any project that uses OTA functions. For factory-boot settings, OTA data partition must not contain data (all bytes cleared to 0xFF). In this case, ESP-IDF software bootloader initializes the factory application, if present in the partition table. If no factory application is included in the partition table, the first available OTA slot (usually 'ota_0') is initialized.

After the first OTA update, OTA data partition is updated to specify which OTA application slot partition must be initialized next. OTA data partition has two flash sectors with 0x2000 bytes in size, to avoid problems if there is power failure while flash is being written. Sectors are erased and written independently with matching data, and if they mismatch, a counting field is used to determine which sector was most recently written.

A single flash of ESP 32 can contain multiple applications as well as many different types of data (calibration data, file systems, parameter storage). Each entry in the partition table has a name (label), type (application, data), subtype, and flash offset where the partition loads.

The simplest way to use the partition table is to use “make menuconfig” via SSH (MinGW32 interface – Minimalist GNU for Windows®) and choose one of the predefined single partition tables. Initially select “Partition table” and then select one of the three options: “Single factory app, no OTA”, “Factory app, two OTA definitions” or “Custom partition table CSV”.

MinGW is a ported version for Microsoft Windows® of the GNU/Linux kernel toolkit. This software includes a set of directories files for Windows® that allows users to use the file system without requiring real-time emulation of a Unix-like system. MSYS (minimal system acronym) is a package that needs to be installed to allow the use of a simple standard Shell scripting language, supporting the POSIX standard to allow the execution of “autoconf” scripts. Both configurations were originally released on Cygwin, which provide greater Unix-like support for Windows®. Both are free software packages and available in English as public licenses in the GNU – General Public License.

In both cases, the factory application is displayed at flash offset 0x10000. With the ‘partition_table’ command, a summary of the partition table is printed on the serial monitor [38]. Table 8 shows the printed summary of the specifications for the three forms of partition table configuration. In the “otadata” space, data is saved for OTA updates. The bootloader queries this data to know which application to run. If “otadata” is empty, the factory application runs. If the “Custom Partition Table CSV” option is chosen, the user can also enter the name of a CSV file (in the project directory) to use as the partition table. CSV file can describe any number of settings for the table that the scheduler needs.

Table 8
Partition table specifications [38].

Partition Configuration	Specifications
Single factory app, no OTA	# Espressif ESP 32 Partition Table # Name, Type, SubType, Offset, Size nvs, data, nvs, 0x9000, 0x6000 phy_init, data, phy, 0xf000, 0x1000 factory, app, factory, 0x10000, 1M
Factory app, two OTA definitions	# Espressif ESP 32 Partition Table # Name, Type, SubType, Offset, Size nvs, data, nvs, 0x9000, 0x4000 otadata , data, ota, 0xd000, 0x2000 phy_init, data, phy, 0xf000, 0x1000 factory, 0, 0, 0x10000, 1M ota_0, 0, ota_0, , 1M ota_1, 0, ota_1, , 1M
Custom Partition table CSV	# Name, Type, SubType, Offset, Size nvs, data, nvs, 0x9000, 0x4000 otadata , data, ota, 0xd000, 0x2000 phy_init, data, phy, 0xf000, 0x1000 factory, app, factory, 0x10000, 1M ota_0, app, ota_0, , 1M ota_1, app, ota_1, , 1M

4.4. Offline storage using SPI flash file system – SPIFFS

SPI Flash File System (SPIFFS) is a file system tool intended for embedded devices. To configure SPIFFS, the user needs to determine the physical page size and then the physical block size. After this, one decides on the size of the logical block, which is a multi-

plier of the size of the physical block. The entire SPIFFS file system must be multiple of the logical block size.

A common scaling for the ESP 32 is 64 K for the logical block size and 256 for the logical page size. SPIFFS implementation does not directly access flash memory. Instead, a functional area called hardware abstraction (“hal”) provides this service. Integration with SPIFFS requires three functions that have the following signatures:

- s32_t (* spiffs_read)(u32_t addr, u32_t size, u8_t * dst)
- s32_t (* spiffs_write)(u32_t addr, u32_t size, u8_t * src)
- s32_t (* spiffs_erase)(u32_t addr, u32_t size)

If the functions succeed, the return code should be “SPIFFS_OK (0)”. In an ESP 32, these map to the flash SPI APIs. To use a SPIFFS file system, the user must execute a call to the “SPIFFS_mount ()” function. This function sets the flash space for SPIFFS, as well as other parameters. An example configuration to create a SPIFFS is shown in Table 9.

Table 9
SPIFFS configuration example.

```
#define LOG_PAGE_SIZE 256
static uint8_t spiffs_work_buf [LOG_PAGE_SIZE * 2];
static uint8_t spiffs_fds [32 * size of (uint32_t)];
static uint8_t spiffs_cache_buf [(LOG_PAGE_SIZE + 32) * 4];
```

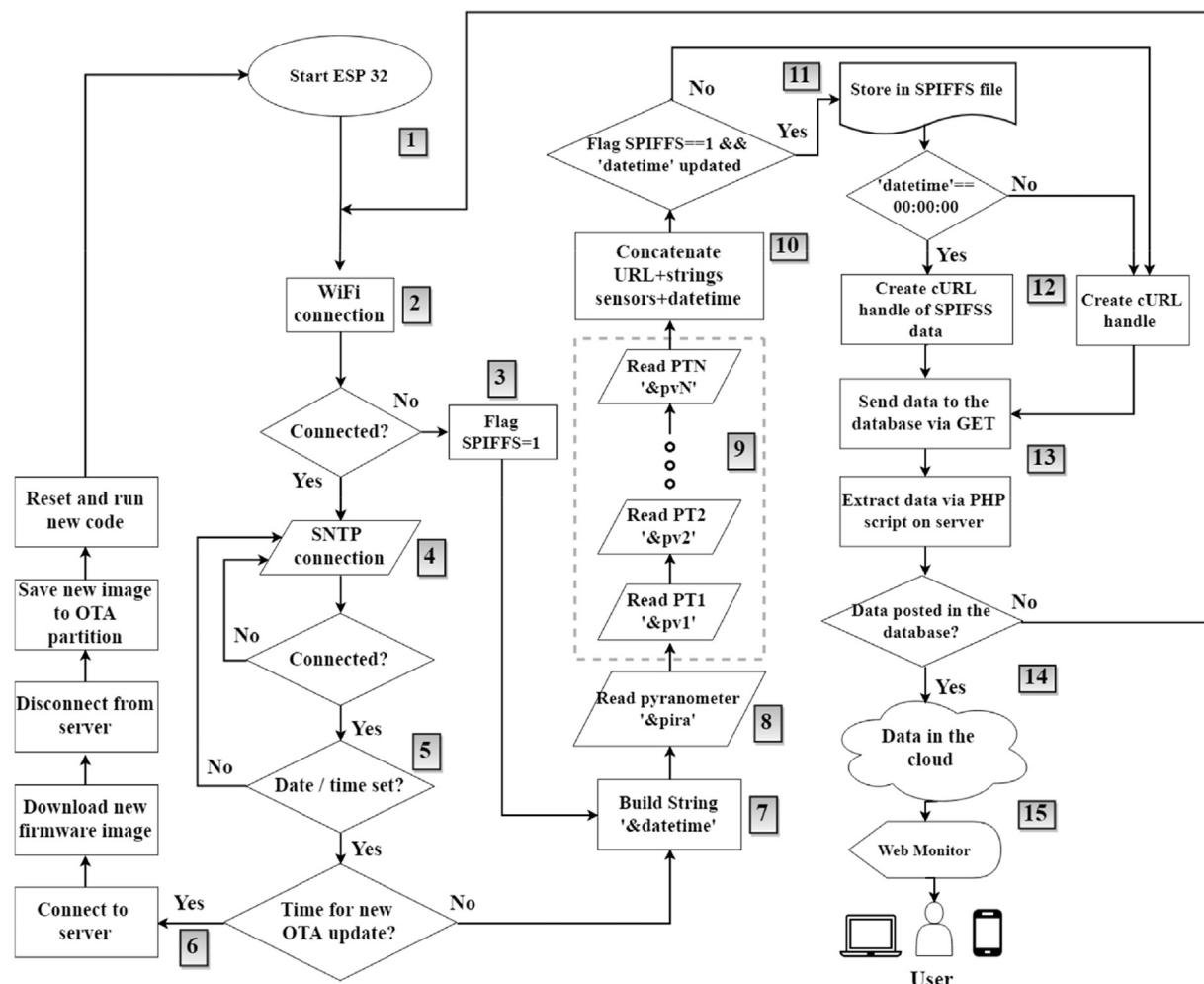


Fig. 18. ESP 32 code flowchart.

4.5. ESP 32 programming

ESP 32 code has been programmed in C Language using Espressif's framework ESP-IDF. The initial project settings were made with the Windows® toolchain in the MinGW SSH environment. Then, for code development and debugging, Eclipse software was used. Fig. 18 shows the flowchart of the code programmed on ESP 32.

Code main steps, identified from 1 to 15 in the flowchart, are described below:

1. ESP 32 Initialization;
2. WiFi mode station (STA) API is initialized and ESP 32 assumes an IP on the network previously configured via SSID and password;
3. In the event of a WiFi connection failure, a flag variable is set so that the data is later stored in the offline SPIFFS file;
4. Network Time Protocol (NTP) server connection API is initialized and connects to the server preprogrammed in the code;
5. The time of the NTP server is then extracted and stored in a variable that is incremented every second via internal timer. The timed NTP function is also responsible for ensuring the synchronism of the number of days of the month and time zones;
6. With date and time set, OTA function is executed in case the programmed condition is satisfied. In this code, the pull strategy was implemented, in which the client (ESP 32) requests data from the server (Cloud) from time to time (for example, it is checked every week if there is new code to be downloaded). Step 6 is subdivided into the following steps:
 - Connection to the server where the code is stored;
 - New code is downloaded in system image format;
 - Connection to server is terminated;
 - New downloaded image is saved on the previously configured OTA partition;
 - ESP 32 is reset and returns to step 1.
7. String 'datetime' is constructed from the time obtained from the NTP server in the format: "&date = 2018-07-21-18:25:12";
8. String 'pira' is built from the average obtained from reading the pyranometer connected to the ESP 32 AD in the format: "&pira = 850";
9. The string of temperature values is constructed from the concatenation of the mean of each PT100 sensor connected to the ESP 32 AD in the format: "&temp1 = 55&temp2 = 54.5&temp3 = 55.5&temp4 = 55.1&temp5 = 55.1&temp6 = 55.7";
10. The final string is built from the concatenation of the previous strings: http://serveraddress/getESP_leagrid.php?action=send1&pira=850&temp1=55&temp2=54.5&temp3=55.5&temp4=55.1&temp5=55.1&temp6=55.7&date=2018-07-21-18:25:12";
11. In the condition of step 11, if the SPIFFS flag is equal to 1, that is, there is no WiFi connection; and if the variable 'datetime' is updated, that is, it has already been fed by the NTP server time ('year > 2017'), the sensor values are then stored in the SPIFFS file ("SPIFFS_mount ()") and sent at midnight. Two points were considered for the offline storage implementation:
 - SPIFFS file space: In case of more than five hours without Internet, it would not be possible to store all strings concatenated per minute (64 k SPIFFS, 200 bytes per URL, 320 URLs, 5.3 h). Therefore, we decided to take averages every 10 min to take up less storage space, allowing up to 53 h of offline backup.
 - Time of sending to the server: in case of no internet connection for 20 min, for example, two URL strings would be stored in SPIFFS. It would not be possible to post them to the database as soon as the Internet connection is re-established, as new values would be measured and posted every minute. Hence, we decided that previous offline stored values would be posted only at midnight. This does not cause monitoring losses, because for a real-time system, what matters is the visualization of the current values at the time of measurement.
12. If data is stored offline and is midnight, the cURL API makes the GET on the server using the in SPIFFS saved URLs and post to the database. After posting the content, the SPIFFS file is dismounted ("SPIFFS_unmount ()"). Throughout the day and with Internet connection, the code continues the main programming, posting current data every minute.
13. Data is received and extracted on the server by the code 'getESP_leagrid.php' and then allocated to the database table for the given plant.
14. If the data has been correctly posted, it is immediately queried by the PHP page and displayed in the Web Monitor. In case of a post error, the WiFi connection is verified in step 2; if it is disconnected, the code is resumed from step 3.
15. The user can then access both the database remotely and monitor the plant via Web Monitor in the form of graphs or logs and verify the operation of the three plants through computer devices with Internet access from anywhere in the world.

For sensor reading, a digital filter was implemented to minimize AD converter instability errors. The sensors values are measured in a cycle of 700 times every 10 ms and then the average of these 700 values is calculated. That is, the sampling rate of each value is 7 s. The averages of all six sensors are then made in 42 s adding up with the server response time, which can range from 2 to 5 s. The sampling rate can be adjusted according to the user need. In the present project, we decided to use most of the available minute for measuring the variables, always considering that when adding to the server response time, the total time does not exceed 60 s.

Steps 2, 3, 4, 5, 6, 10, 11 and 12 use error checking functions ("event_handler ()") and resume the code in case of command failures. ESP 32 is also reset to step 1 in case of code freeze by using Watchdog Timer (WDT), which is configured by default in ESP-IDF. WDT is a timer that triggers a system reset in case of an error.

Regarding step 6 on OTA flashing, 'pull' method was implemented. The ideal is the application of 'push' technology, in which the client (ESP 32) is only updated in case of new code on the server (Cloud). That is, the communication is initiated by the server. This methodology reduces the number of requests from the customer and simplifies the data transfer. 'Push' can be implemented using MQTT protocol.

The measured variables units are defined in the database. Each plant has two tables. One of the tables, named "plantX_monit" consists of the PHP file from step 13 which extracts the values from the URL concatenated in step 10. The second table, named "plantX_sens", consists of the variable description, the corresponding index and the unit: °C for temperature, W/m² for irradiance, % for relative humidity and m/s for wind speed.

4.6. ESP 8266 programming

Compared to ESP 32, the IoT ESP 8266 NodeMCU microcontroller [39] is an earlier and more simplified version, launched in

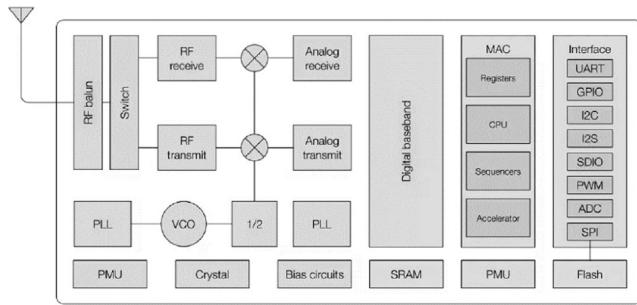


Fig. 19. ESP 8266 block diagram [39].

Table 10
ESP 8266 current consumption. Adapted from [39].

Parameter	Typical current
WiFi data transmission	120–170 mA
WiFi data reception	50–56 mA
Sleep mode (RTC on, circuit in standby mode ready to be woken up)	15 mA
Wake up mode (External crystal enabled to wake up when requested)	0.9 µA
Deep-sleep mode (RTC on, circuitry off)	20 µA
Power-off mode (RTC off)	0.5 µA

2014. ESP 8266 functional block diagram is shown in Fig. 19. The AD channel has 10 bits of resolution with 0–1 V range. ESP 8266 current consumption can be seen in Table 10. The selected development board was mini D1 board from the Wemos manufacturer with 11 GPIO and 4 MHz flash.

The selected model is compact and provides plug-in sensor modules, such as the DHT11 sensor shield. Although only one AD channel is available, this was enough to read the anemometer, since DHT11 is a digital sensor and communicates with the ESP 8266 through only one GPIO. DHT11 sensor specifications are listed in Table 11. A protective enclosure was necessary because the system is installed outdoors. However, for the humidity and ambient temperature sensor DHT11 to operate correctly, it was necessary to weld a cable so that the sensor was outside the circuit protection box.

The #40C NRG anemometer is a low-level AC sine wave sensor with frequency proportional to wind speed. The sensor has a scale factor of 0.765 m/s/Hz and offset of 0.35 m/s. Hence, it is necessary to develop a logic for reading the values, if the user does not have a conversion circuit for digital wave. Conventional dataloggers usually have a specific input for this type of sensor. Considering that ESP 8266 is a microcontroller that has an AD converter, it is

possible to connect the sensor directly to the AD without the need of conditioning circuit for reading the sine wave values in bits.

The output amplitude is not proportional to the wind speed, but rather to the frequency [40]. The anemometer has a four-pole magnet that induces a sine wave voltage in a coil, producing an output signal with frequency proportional to the wind speed. The highest recorded value by the manufacturer was 96 m/s, with frequency of 125 Hz.

To calculate the sine wave frequency, aiming to obtain the wind speed value, we established some programming conditions and parameters. One of the conditions is the identification of the rising edge for the start of the wave point count using the ESP 32 AD converter. From the identification of the number of wave points, it is possible to calculate the period and consequently the frequency. However, the AD does not read negative values, so the negative half-cycle points are considered ‘0’. The wave period is then obtained from the quantity of values in the positive and negative half-cycles.

Considering the anemometer threshold of 0.78 m/s, approximately 1 Hz, and from AD readings, the value of 27 bits was defined as the offset of the rising edge and not the value 0. However, regardless of the selected point, this does not interfere with the count, since the wave profile repeats in the next cycle, that is, the values are counted until the value (V_{AD}) is higher than 27 again and the other conditions are satisfied.

Hence, initially a flag was set at ‘1’ when the value was less than 27 to identify transition from the falling edge to the leading edge. In addition, AD current value must be greater than the previous value. For this, the previous value is stored in one variable (V_{prev}) and is compared with the current value, stored in another variable (V_{actual}). Finally, if V_{prev} is less than 27, V_{actual} is greater than V_{prev} and flag is equal to 1, the leading edge is identified and the counting of V_{pos} variable starts. An interval of 10 ms was defined for counting points in V_{pos} , since for a wind speed of 6 m/s, the frequency is 7.5 Hz and the period is 130 ms. Therefore, 10 ms is ideal to identify several points in the wave, besides filtering out any high frequency noises. After calculating V_{pos} , we obtain the wave period from Eq. (5):

$$\text{period} = (V_{pos} + 1) * 0.01 \quad (5)$$

where 0.01 is the 10 ms of the reading range. The frequency is then the inverse of the period. Finally, the wind speed is calculated according to Eq. (6):

$$\text{windspeed} = (\text{freq} * 0.765) + 0.35 \quad (6)$$

where 0.765 m/s/Hz is the scaling factor and 0.35 m/s the offset.

Fig. 20 shows the conditions necessary for the construction of the AD sine wave reading logic.

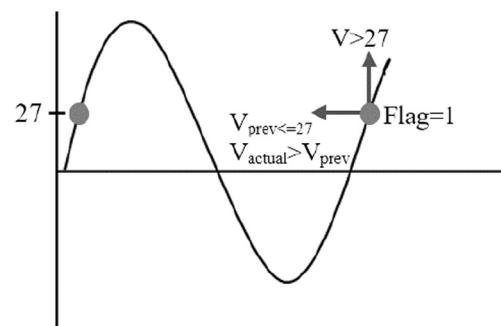


Fig. 20. Anemometer waveform and conditions for AD reading logic construction.

Table 11
DHT11 specifications.

Description	Specification
Power supply	3–5 V _{dc} (5.5 V maximum)
Current consumption	200 µA–500 mA; standby: 100 µA a 150 µA
Relative humidity measuring range	20–90%
Accuracy of relative humidity measurement	±5.0%
Temperature measuring range	0°–50 °C
Accuracy of temperature measurement	±2.0 °C
Response Time	<5 s
Dimensions	23 × 12 × 5 mm (including pins)

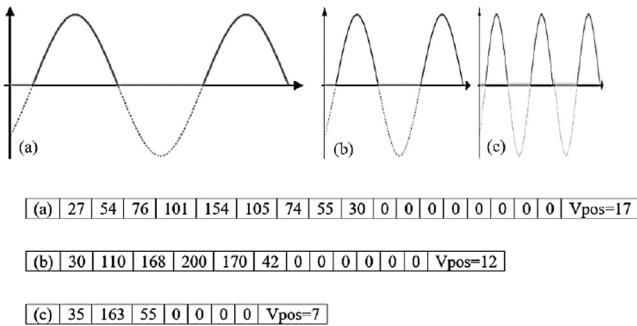


Fig. 21. Examples of anemometer output waveforms.

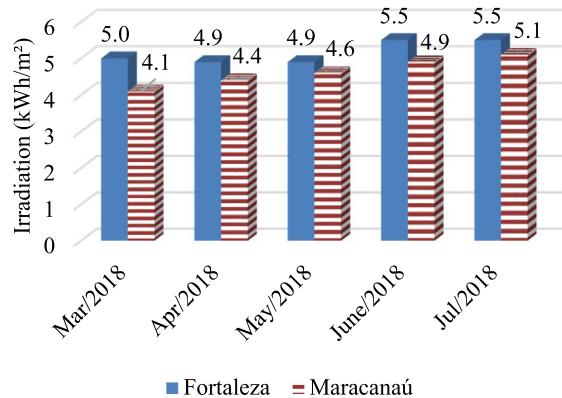


Fig. 23. Average daily irradiation values in Fortaleza and Maracanaú – Brazil.

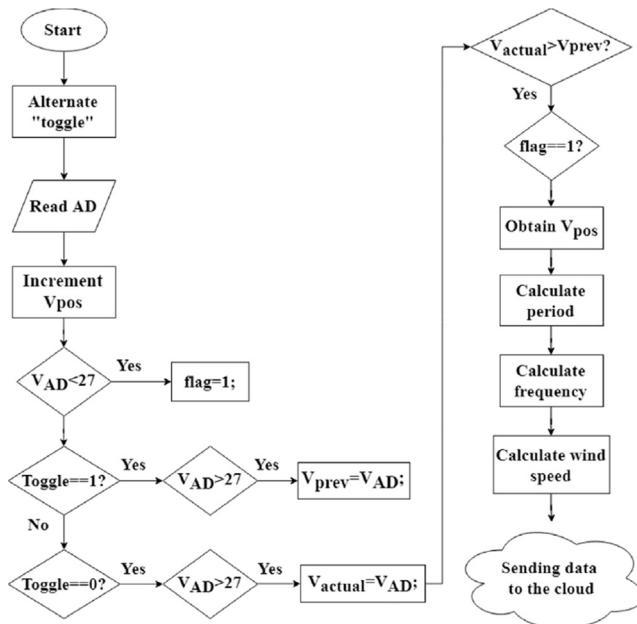


Fig. 22. Anemometer reading code flowchart with ESP 8266.

In Fig. 21 three examples of anemometer output waveforms are illustrated: a low frequency waveform in (a), medium frequency in (b) and high frequency in (c).

For example, if V_{pos} equals 17, in (a), the period would be 180 ms, the frequency 5.55 Hz and the speed 4.6 m/s. The positive half-cycle is read by the AD in bits and the negative half-cycle of the wave is read as zero. All these points are summed and used to calculate the wind speed. The flowchart shown in Fig. 22 details how the logic was implemented.

5. Results and discussion

Daily average irradiation values in Fortaleza and Maracanaú were calculated using data from the proposed IoT monitoring network, as shown in Fig. 23. Irradiation values are higher in Fortaleza, with averages varying from 4.9 to 5.5 kWh/m², from March to July 2018. In Maracanaú, values range from 4.1 to 5.1 kWh/m² in the same period. As the local rainy season (first semester of the year) passes, irradiation values increase at the two localities.

In Table 12 we show irradiance and PV module temperature behaviour, monitored in the days of lowest, average and highest PV electricity generation in July 2018 at the two plants.

On July 10th the maximum PV module temperature peak was approximately 40 °C at 2 pm in Fortaleza. On this day a drop of irradiance was observed around noon due to rainfall. The irradiance in Maracanaú was also low and oscillating, presenting values below 500 W/m² throughout the day and only a peak of 558 W/m² at 1:50 pm. PV modules temperature remained low during the day and did not exceed 39 °C.

Considering the average day, irradiance oscillation is observed between 10 am and 3 pm, which reflects in the module temperature curve. On the same day, the modules maximum temperature reached 62 °C in Fortaleza and 48 °C in Maracanaú.

Considering the day of greatest PV electricity generation, data collected by the IoT monitoring network show that the PV modules maximum temperature values reached approximately 60 °C in Fortaleza while in Maracanaú the modules temperature is 12 °C lower. One of the reasons is that Maracanaú is windier than Fortaleza and the PV plant support structure favours wind circulation under and over the modules. In Fortaleza, the PV string is installed on a concrete ceiling.

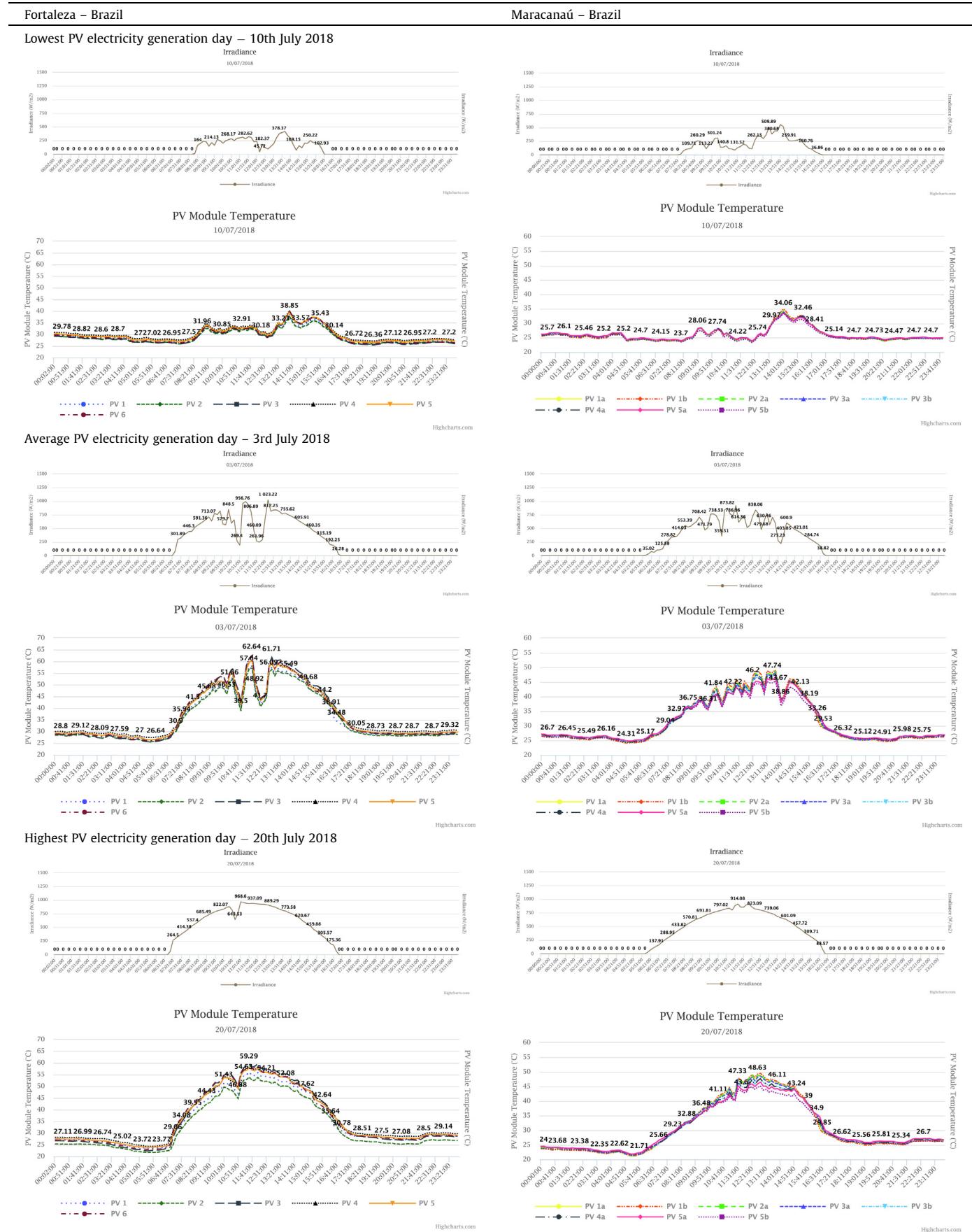
Compared to Fortaleza, Maracanaú's irradiance graph is more stable on the highest generation day but shows lower values. It is worth emphasizing the influence of the irradiance curve shape on the module's temperature curve. The peaks are approximately 850 W/m². The PV module temperature on this highest generation day of July reached 48 °C at 12:30p.m. Due to thermal inertia, temperature has a slow response time to irradiance variation.

In both sites, July was the month with highest PV electricity generation. Fortaleza's PV plant generated 235.8 kWh and Maracanaú's plant 161.4 kWh. The July 10th was the lowest generation day with only 3.6 kWh in Fortaleza and 2.2 kWh in Maracanaú. The chosen average day was the 3rd of July. On this day, Fortaleza's generation was 7.3 kWh and Maracanaú's 5.1 kWh. The highest generation day was the 20th of July, with a total of 8.8 kWh in Fortaleza, while in Maracanaú only 6.1 kWh was generated. According to Table 13, Fortaleza was the site with the highest electricity yield (relation between electricity generated and installed power capacity) in July 2018. Using electricity yield, both PV plants used, which have different installed power capacity can be compared (Fortaleza: 1.5 kW_p; Maracanaú 1.35 kW_p).

In July, summer in Germany, the average PV module temperature at Cologne's plant reached a maximum value of 60 °C. Sun hours totalize 16 h in this period (05:40 am to 9:40 pm), as shown in Fig. 24.

Table 12

Lowest, average and highest PV electricity generation in July 2018 in Fortaleza and Maracanaú – Brazil.



In Fortaleza and Maracanaú, although July starts the dry period in the Brazilian northeast region, there was still rainfall on some days of the month in 2018 in both sites. In Maracanaú, PV modules temperature is in average 10 °C lower than Fortaleza.

All the applied sensors meet the requirements of IEC 61724:1998 standard (British Standard, 1998), which specifies the required accuracy of sensors for PV system performance monitoring. ESP32's AD has 1 mV resolution, which is satisfactory for reading the pyranometer (2 mV/W/m²) and the PT100 (8 mV/°C) developed conditioning circuits. The anemometer has 0.2 m/s precision, which is better than the 0.5 m/s specified. For ambient temperature the applied DHT11 has ±2 °C accuracy. A sensor with ±1 °C accuracy should be used.

Our proposed system is validated using a Programmable Logic Controller (PLC) with 12-bit resolution AD channel. Fig. 25 shows the monthly average curve of the PV module temperature measured from 1st to 31st, July. For temperature values above 52 °C, the error between the proposed IoT monitoring using ESP and the PLC increases, about 2 °C; for lower temperature values, the measurement follows the reference value.

Fig. 26 shows the correlation between the PV module temperature measured by the proposed IoT monitoring and PLC reference value for the average measurements from 1st to 31st, July. Using the Root Mean Square Error (RMSE), the correlation is about 0.9989.

Table 13
PV electricity yield in July 2018.

PV Plant	Maracanaú (kWh/kW _p)	Fortaleza (kWh/kW _p)
Day of highest yield	4.5	5.87
Day of average yield	3.78	4.93
Day of lowest yield	1.63	2.4
Total yield in July 2018	119.6	157.2

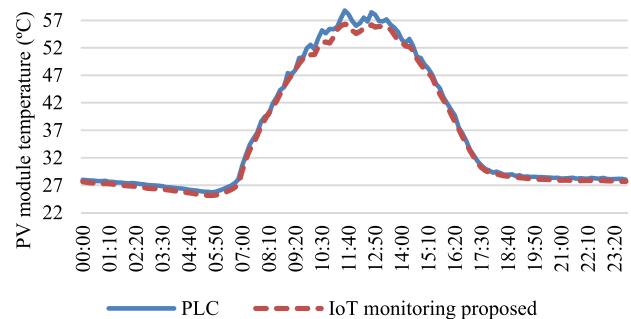


Fig. 25. PV module temperature measured by the proposed IoT monitoring and PLC.

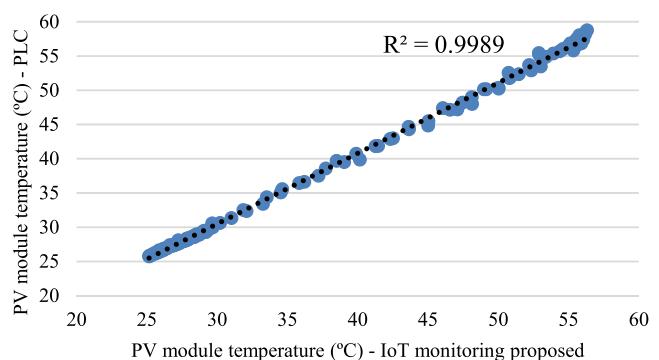


Fig. 26. Correlation between PV module temperature measured by the proposed IoT monitoring and PLC reference value.

6. Conclusion

The designed, implemented and tested IoT monitoring network proposes low cost embedded systems, based on free hardware and

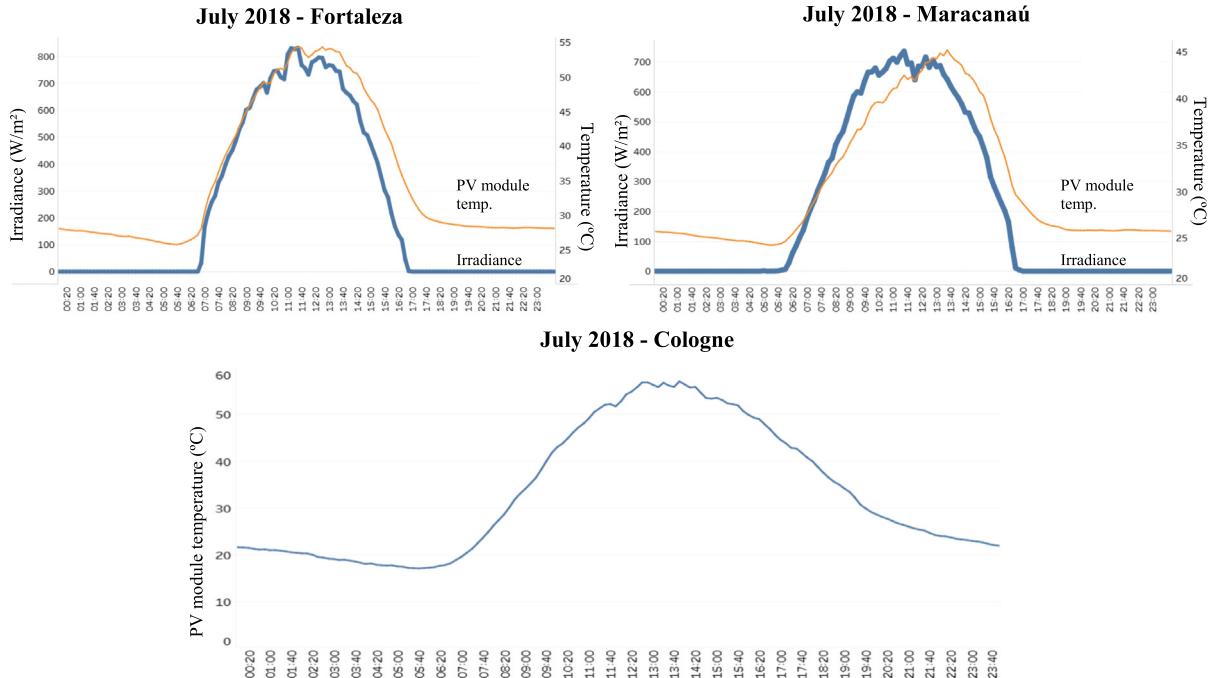


Fig. 24. PV module temperature – Fortaleza × Maracanaú × Cologne – July 2018.

software, allowing online distribution and free use for project development and research. ESP 8266 and ESP 32 communicate with the Cloud via WiFi. A limitation for the WiFi use in decentralized plants is the need of a WiFi router. As an additional advantage, the proposed monitoring systems can be expanded to collect data from other types of analog or digital sensors as well as for other types of applications using renewable energy sources.

From the developed PHP Web Monitor, it is possible to check the data online and monitor in real time the three PV plants that make up the IoT PV network. Open source and cross-platform software (Linux, Windows® and Mac OSX) use for online microgeneration monitoring systems allows greater user interaction and accessibility due to the possibility of free distribution. One advantage of the developed Web Monitor is that there is no need to unlock ports and/or firewall when using the multi-user Cloud service.

The proposed IoT network focus on PV modules temperature monitoring to provide data for the analysis of the electric generation efficiency and for the detection of failures in case of overheating of PV cells. Meteorological data such as solar irradiance, ambient temperature, relative humidity and wind speed were also monitored, allowing a more complete analysis of the effect of these variables on the modules' temperature.

Signal conditioning circuits of PT100, pyranometer and anemometer were calibrated using commercial equipment and presented measurements within the specific error range of each sensor. ESP 32 AD converter was calibrated with a precision power source to avoid measurement errors, considering the nonlinearity of the bit-voltage curve.

It was possible to program ESP 32 remotely via OTA using 'Pull' method, where the client (ESP 32) makes requests to the server to check if there is a new code to be downloaded and saved. This method makes the system inefficient as it increases the client's processing and generates discomfort, since in most cases there is not an exact time for the information to be updated. 'Push' method via MQTT should better fit to this purpose and is going to be implemented for new plants. Data sending from ESP 32 and 8266 is done successfully every minute via GET to the MySQL database. In case of server or Internet disconnection, data is stored offline using SPIFFS.

Results show that the average daily irradiation from March to July 2018 in Fortaleza (4.9–5.5 kWh/m²) is higher than in Maracanaú (4.1–5.1 kWh/m²), resulting in a greater PV electricity generation in Fortaleza.

Considering that Maracanaú's plant is in a windier site with a support structure that favours wind circulation, it was observed that the maximum module temperatures reached 50 °C; Fortaleza's plant, installed on a concrete ceiling and having more obstacles to the wind, reached values around 62 °C. In Cologne, PV modules temperature data in July resemble the values of the Brazilian cities, however, there are more hours of sunshine in Germany in this period.

PT100 temperature data were compared with data from another PT100 sensor connected to a PLC. Using RMSE, the calculated dispersion was about 0.9989. To analyze the overall energy consumption of the IoT embedded systems and sensors, we intend to calculate in future papers, the current consumption curves of both ESP 8266 and ESP 32 and compare their performance with other applicable modules, like ZigBee and LoRa.

Our Web Monitor is password protected and only authorized people can access the data stored on the database. However, if the handled data need to be kept private, a Cloud service for data security should be applied or data encryption could be implemented in the firmware.

Acknowledgements

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brazil (CAPES) – Finance Code 001. The authors would like to thank the Deutscher Akademischer Austauschdienst (DAAD) and CAPES for exchange scholarship awarded to the first author; CNPq for project financial support (420133/2016-0 Universal 01/2016) and for researcher scholarship awarded to the third author; and UFC and IFCE for the availability of laboratories and equipment.

References

- [1] R.I.S. Pereira, I.M. Dupont, P.C.M. Carvalho, S.C.S. Jucá, IoT embedded Linux system based on raspberry Pi applied to real-time cloud monitoring of a decentralized photovoltaic plant, *Measurement* 114 (2017) 286–297, <https://doi.org/10.1016/j.measurement.2017.09.033>.
- [2] ANEEL, Aneel registers more than 7,600 distributed generation connections in Brazil, (2017). <http://www.brasil.gov.br/infraestrutura/2017/01/aneel-registra-mais-de-7-6-mil-conexoes-de-geracao-distribuida>.
- [3] H.R. do Nascimento, Installed solar power of Ceará jumps 57%; State is leader in the Northeast Brazil, (2018). <http://diariodonordeste.verdesmares.com.br/mobile/cadernos/negocios/potencia-solar-instalada-do-ceara-salta-57-estado-e-lider-no-nordeste-1.1956879> (accessed June 19, 2018).
- [4] L. Zhu, A. Raman, K.X. Wang, M.A. Anoma, S. Fan, Radiative cooling of solar cells, *Optica* 1 (2014) 32–38, <https://doi.org/10.1364/OPTICA.1.000032>.
- [5] JINKO SOLAR, Jkm270Pp-60, (2008) 5–6. <https://www.jinkosolar.com/ftp/US-Eagles-270PP-60.pdf>.
- [6] H.E. Gad, H.E. Gad, Development of a new temperature data acquisition system for solar energy applications, *Renew. Energy* 74 (2015) 337–343, <https://doi.org/10.1016/j.renene.2014.08.006>.
- [7] F.J. Ferrero Martín, M. Valledor Llopis, J.C. Campo Rodríguez, J.R. Blanco González, J. Menéndez Blanco, Low-cost open-source multifunction data acquisition system for accurate measurements, *Meas. J. Int. Meas. Confed.* 55 (2014) 265–271, <https://doi.org/10.1016/j.measurement.2014.05.010>.
- [8] N. Instruments, LabVIEW, (1986). <http://www.ni.com/labview/> (accessed January 24, 2017).
- [9] R.M. Zago, F. Fruett, A low-cost solar generation monitoring system suitable for internet of things, *INSCIT 2017 - 2nd Int. Symp. Instrum. Syst. Circuits Transducers Chip Sands, Proc.*, 2017, doi:10.1109/INSCIT.2017.8103509.
- [10] M.M. Rahman, J. Selvaraj, N.A. Rahim, M. Hasanuzzaman, Global modern monitoring systems for PV based power generation: a review, *Renew. Sustain. Energy Rev.* 82 (2018) 4142–4158, <https://doi.org/10.1016/j.rser.2017.10.111>.
- [11] D. Pasalic, D. Bundalo, Z. Bundalo, B. Cvijic, ZigBee-based data transmission and monitoring wireless smart sensor network integrated with the Internet, in: 2015 4th Mediterr. Conf. Embed. Comput., 2015, pp. 240–243, doi:10.1109/MECO.2015.7181913.
- [12] C.S. Choi, J.D. Jeong, J. Han, W.K. Park, I.W. Lee, Implementation of IoT based PV monitoring system with message queuing telemetry transfer protocol and smart utility network, in: Int. Conf. Inf. Commun. Technol. Converg. ICT Converg. Technol. Lead. Fourth Ind. Revolution, *ICTC 2017. 2017-Decem.* 2017, pp. 1077–1079, doi:10.1109/ICTC.2017.8190859.
- [13] B. Cheng, S. Zhao, J. Qian, Z. Zhai, J. Chen, Lightweight service mashup middleware with REST style architecture for IoT applications, *IEEE Trans. Netw. Serv. Manag.* 15 (2018) 1063–1075, <https://doi.org/10.1109/TNSM.2018.2827933>.
- [14] B. Cheng, M. Wang, S. Zhao, Z. Zhai, D. Zhu, J. Chen, Situation-Aware Dynamic Service Coordination in an IoT Environment, 25 (2017) 2082–2095.
- [15] S.H. Pramono, Internet-based monitoring and protection on PV Smart grid system, in: Int. Conf. Sustain. Inf. Eng. Technol., 2017, pp. 448–453.
- [16] C. Choi, J. Jeong, I. Lee, W. Park, LoRa based renewable energy monitoring system with open IoT platform 2. LoRa based Energy IoT Monitor. (2015).
- [17] N. Erraissi, M. Raoufi, N. Aarich, M. Akhsassi, A. Bennouna, Implementation of a low-cost data acquisition system for "PROPRE.MA" project, *Meas. J. Int. Meas. Confed.* 117 (2018) 21–40, <https://doi.org/10.1016/j.measurement.2017.11.058>.
- [18] S. Pulipaka, A. Ramesh, R. Kumar, B. Bora, S. Kumar, S.K. Singh, Non-intrusive real-time monitoring of PV generation at inverters using Internet of photovoltaics, *Electron. Lett.* 53 (2017) 1137–1138, <https://doi.org/10.1049/el.2017.0694>.
- [19] P.Q. Dzung, D.N. Dat, N.B. Anh, L.C. Hiep, H. Lee, Real-time Communication Network Solution based on Zigbee and Ethernet for Photovoltaic Systems, (2014) 197–202.
- [20] International, S.I. 61850-4-720, Communication networks and systems for power utility automation – Part 7-420: Basic communication structure – Distributed energy resources logical nodes, (2003).
- [21] P.P. Parikh, Investigation of Wireless LAN for IEC 61850 based Smart Distribution Substations, *The University of Western Ontario*, 2012.
- [22] T. Lagkas, V. Argyriou, S. Bibi, P. Sarigiannidis, UAV IoT framework views and challenges: towards protecting drones as "things", *Sensors (Basel)* 18 (2018), <https://doi.org/10.3390/s18114015>.

- [23] P. Barker, M. Hammoudeh, A survey on low power network protocols for the internet of things and wireless sensor networks, Proc. Int. Conf. Futur. Networks Distrib. Syst. - ICFNDS '17. (2017) 1–8, <https://doi.org/10.1371/journal.pcbi.1003573>.
- [24] K. Mekki, E. Bajic, F. Chaxel, F. Meyer, A comparative study of LPWAN technologies for large-scale IoT deployment, ICT Exp. (2018), <https://doi.org/10.1016/j.icte.2017.12.005>.
- [25] T. Simioni, O impacto da temperatura para o aproveitamento do potencial solar fotovoltaico do Brasil, (2017).
- [26] E. Barykina, A. Hammer, Modeling of photovoltaic module temperature using Faiman model: sensitivity analysis for different climates, Sol. Energy 146 (2017) 401–416, <https://doi.org/10.1016/j.solener.2017.03.002>.
- [27] J.A. Duffie, W.A. Beckman, Solar Engineering of Thermal Processes, 4th ed., 2013.
- [28] R.G. Ross, Interface design considerations for terrestrial solar cell modules, in: Proc. 12th IEEE Photovolt. Spec. Conf., Baton Rouge, LA, 1976, pp. 801–806.
- [29] R. Chenni, M. Makhlof, T. Kerbache, A. Bouzid, A detailed modeling method for photovoltaic cells, Energy (2007) 1724–1730.
- [30] B.D.O. Busson, P. Hassan, M. Campos, P. Cesar, M. De Carvalho, Validação de modelos de comportamento térmico de painéis fotovoltaicos para o semiárido brasileiro, (2018).
- [31] PHB, INVERSOR SOLAR FOTOVOLTAICO PHB 1500-NS, (2017).
- [32] Espressif Systems, Read the Docs Template Documentation, (2017). <https://esp-idf.readthedocs.io/en/latest/>.
- [33] Espressif Systems, ESP32 Datasheet, (2018) 58. http://espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf.
- [34] Espressif Systems, ESP 32 Get Started, READ DOCS. (2017). <http://esp-idf.readthedocs.io/en/latest/get-started/>.
- [35] Espressif Systems, Set up of Toolchain for Windows, READ DOCS. (2017). <https://esp-idf.readthedocs.io/en/v2.0/windows-setup.html>.
- [36] Espressif Systems, Espressif GitHub, (2017). <https://github.com/espressif>.
- [37] Espressif, What are the ADC input ranges? Mnemonix, 2017, <https://wwwesp32.com/viewtopic.php?t=1045> (accessed July 20, 2018).
- [38] Espressif, Partition Tables - OTA, READ DOCS. (2018). <http://esp-idf.readthedocs.io/en/latest/api-guides/partition-tables.html> (accessed July 15, 2018).
- [39] Espressif, ESP8266EX Datasheet, Espr. Syst. Datasheet. (2018) 1–31. https://www.adafruit.com/images/product-files/2471/0A-ESP8266_Datasheet_EN_v4.3.pdf.
- [40] NRG, NRG # 40C Anemometer, (2010) 1–2. <http://www.windandsun.co.uk/media/277060/NRG-40C-Anemometer-data-sheet.pdf>.