EAI Endorsed Transactions
on Energy Web

Research Article EAI.EU

Design and Experimental Validation of a Compact Low-Cost Weather Station for Solar Photovoltaic Applications

Jose A. Salgado[1,*], Vitor Monteiro[1], J. G. Pinto[1], Joao L. Afonso[1] and Jose A. Afonso[2]

[1]ALGORITMI Research Centre, University of Minho, 4800-058 Guimarães, Portugal
[2]CMEMS-UMinho Center, University of Minho, 4800-058 Guimarães, Portugal

Abstract

This paper presents a compact low-cost weather station specially dedicated to renewable energy applications based on solar photovoltaic (PV) technologies. The main objective of the weather station is to verify which technology of solar PV modules would be more suitable for the specific location where the weather station is installed. Therefore, the developed weather station includes three technologies of PV modules (polycrystalline, monocrystalline, and amorphous silicon), each one connected to a dedicated DC-DC power converter with a maximum power point control (MPPC) functionality, as well as a set of sensors (solar irradiance, temperature, humidity, wind speed, and wind direction) used to measure the local weather. The acquired data is processed and stored locally in the weather station and, when necessary, the user can download the data to an Android mobile device through a Bluetooth Low Energy (BLE) wireless network connection using the developed mobile app, where the transferred data is stored in a SQLite database and can be visualized in graphs. Throughout the paper, the design of the developed weather station and the associated technologies are described, as well as the details of the mobile app. The developed system comprising the weather station and the mobile app was validated through a set of experimental tests ranging from the data acquisition to its visualization, as well as the achieved wireless data transfer performance.

Keywords: Weather Station, Solar Photovoltaic Modules, Maximum Power Point, Bluetooth Low Energy, Mobile App, Android.

Received on 04 September 2020, accepted on 21 November 2020, published on 02 December 2020

doi: 10.4108/eai.2-12-2020.167290

1. Introduction

As the world population increases, the demand for energy grows with it, where fossil fuels like coal, oil and natural gas still supply 80% of the total energy demand. However, the environmental effects are a major problem associated with these sources of energy, due to the CO2 emissions [1]. Therefore, there is a need to focus on other energy sources as optimal long-term solutions, like renewable energy sources. In this context, solar energy is one of the main renewable energy sources that has been considered to tackle this problem. A total of 1.8×10^{11} MW of solar power is received by the Earth at each instant [2]. Despite not covering all energy consumption needs, solar energy has the potential to cover a significant share of it. Photovoltaics is one of the methods to convert solar energy into electricity, and this technology has had a significantly improvement in the last years in terms of efficiency and cost reduction. It is based on semiconductor materials that exhibit the PV (Photovoltaic) effect to convert the sunlight into electrical energy. These semiconductors are mainly made of silicon, which has a large availability in the Earth's crust. In terms of environmental effects, PV generates no pollution neither greenhouse gas emissions. It also provides great scalability, meaning that its applications can range from a few mW to hundreds of MW. However, the main disadvantage is that the production of electrical energy is restricted to a range of hours during the day, which many times does not match the peak energy demand hours [3].

Concerning solar PV technologies, there is a variety of options. The most suitable choice will depend on the location and the scale of application. To determine the best solar PV

[*]Corresponding author. Email: jose.salgado@algoritmi.uminho.pt

technology to be used in a given location, the information about the weather conditions in that location is very important. The local temperature, solar irradiance, humidity, and even wind speed and direction, are some factors that have impact on the energy produced by solar PV modules, and that impact can be different for different technologies [4].

This paper is a revised and extended version of a previous conference paper [5]. The main contribution of this work is the development and test of a compact and low-cost weather station designed to measure multiple environmental parameters for renewable energy applications based on solar PV. The collected data may be transferred to a mobile device when the user approached the weather station, where it is stored and may be analysed by the user using the developed mobile app. The developed weather station prototype also integrates small solar PV modules of three different technologies: polycrystalline, monocrystalline, and amorphous silicon. The rationale is to allow correlating the collected environmental data to the power production of the tested solar photovoltaic technologies, to assist the user in assessing which technology is the most suitable for utilization in a particular location.

For convenience and easiness of access, the data is transferred to the user's mobile device using a wireless connection, instead of cables, since the weather station may be installed in a hard-to-reach location. The developed system uses Bluetooth Low Energy (BLE) for the wireless data transfer due to native integration in mobile devices, as well as its low energy consumption [6].

As main contributions in this version, it can be highlighted: (a) A new and more efficient power converter is presented to maximize the power extraction from the PV modules installed in the station; (b) The Android application was extended with more options in terms of data visualization; (c) The throughput of the BLE connection was increased by adjusting the connection parameters using the features available in the BLE Android API (Application Program Interface).

The rest of the paper is organized as follows. Section 2 presents a background and related work. Section 3 presents a description of the developed system, including the description of the data acquisition and storage, as well as the wireless data transfer and the mobile application. Section 4 presents the main experimental results obtained to validate the proposed system. Section 5 presents the main conclusions.

2. Background and related work

Crystalline silicon is the most prevalent material for solar PV modules and its working principle is based on the concept of the p-n junction, which uses two types of semiconductors, one p-doped and another n-doped. Various sub-types of crystalline silicon can be used to develop solar photovoltaic modules, where the most popular ones are monocrystalline silicon and polycrystalline silicon. Monocrystalline silicon is more efficient, but generally more expensive than other types. On the other hand, polycrystalline silicon is less efficient, but less expensive than the monocrystalline silicon ones, so they are the most common type used [7]. Thin-film technologies are also commonly used to develop solar photovoltaic modules. Currently, the most used thin-film technology is the amorphous silicon. Despite a lower efficiency when comparing to crystalline technologies, it has the advantage of being flexible, which can be useful in some applications [7].

Concerning related work, in [8] a system to monitor the performance of solar panels using a low-cost Wireless Sensor Network (WSN) is proposed. Solar irradiance and temperature, as well as the current and voltage from the solar panels were the measured variables. In this case, the collected data was continuously transmitted to a nearby personal computer using ZigBee. In [9] is presented a prototype of a low-cost data logger and monitoring system for a PV application. The values of current and voltage data are acquired from a PV system, which uses a MPPT (Maximum Power Point Tracking) controlling unit. The data acquired is stored in a Secure Digital (SD) card. Then, it is transferred to a human-computer interface via a wireless communication, using a 3DR Radio telemetry module. An application was developed using the software Delphi 7.0, to visualize the data acquired. In [10] a remote data logger for weather forecasting based in a General Packet Radio Service (GPRS) connection was developed. The data gathered was stored locally in a (SD) card. The File Transfer Protocol (FTP) was used to transmit the collected data to a remote server. Every time the data logger receives a report with the measured parameters, it creates a new text file in the SD card and sends the file to the remote server. The GPRS connection requires a cellular device with a SIM (Subscriber Identification Module) card, meaning associated costs with the data transmission. In contrast, this paper proposes an alternative solution, where the data is stored locally in the weather station and later transferred to a user device using a short-range wireless network when they approach the weather station, which avoids the costs associated with data transfer. The solution in this paper also integrates three solar photovoltaic modules of small size, based on three different technologies, in which the produced power is measured for comparison purposes.

3. Developed system

This section describes the development of the different components of the proposed weather station. Multiple sensors were used to allow the measurement of the following local weather parameters: solar irradiance, temperature, humidity, wind speed and wind direction. The developed system is also able to measure the energy production by three different solar PV modules, to correlate with the measurements of the weather sensors.

The architecture of the developed system, represented in Figure 1, is composed by two main parts: a weather station and an Android client. The data collected by the weather station is processed and stored locally in a SD memory card. When requested by the Android mobile app, the data is transferred to the mobile device using BLE, where it is stored in a local SQLite database and can be visualized by the user.

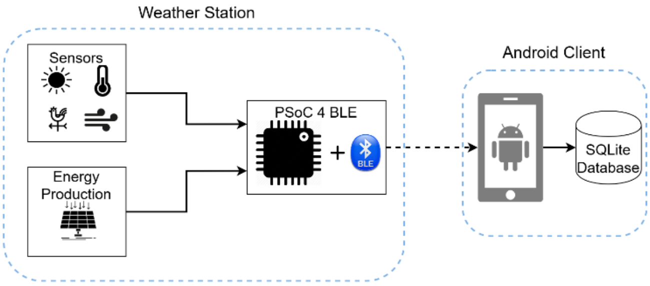


Figure 1. Developed system architecture.

3.1. Data acquisition and storage

The block diagram of the data acquisition hardware implemented in the weather station is presented in Figure 2. The data acquisition and processing (as well as the wireless communication with the Android client) are performed by a PSoC (Programmable System on a Chip) 4 BLE module [11], from Cypress Semiconductor. This module, used in the developed system, includes an ARM Cortex-M0 32-bit microcontroller with integrated Bluetooth 4.2 protocol stack. It contains blocks for serial communication via UART (Universal Asynchronous Receiver/Transmitter), SPI (Serial Peripheral Interface), or I2C (Inter Integrated Circuit). It also provides an ADC (Analog to Digital Converter) with a maximum resolution of 12-bits, four programmable operational amplifiers and four TCPWM (Timer Counter Pulse Width Modulation) blocks, which can be used for timers, counters or PWM (Pulse-Width Modulation) generators. The firmware of the PSoC 4 BLE microcontroller was programmed in C using PSoC Creator IDE (Integrated Development Firmware), which is also provided by Cypress. During the development phase, the PSoC 4 BLE module was attached to a development board for programming and debugging. Both boards are supplied in the CY8CKIT-042-BLE-A kit [11]. In the final prototype only the PSoC 4 BLE module was used, which was attached to the weather station PCB (Printed Circuit Board).

The photodiode BPW34 [12] was used to measure the solar irradiance. This photodiode was chosen taking into consideration the minimum and maximum values for the solar spectral radiation wavelengths and the cost. The signal produced by the photodiode is fed into a signal conditioning circuit to convert the current produced into a voltage that can be read by the ADC of the microcontroller. This circuit was implemented using a transimpedance amplifier, as shown in Figure 3. The value of R1 was determined experimentally so that the output value does not exceed the maximum voltage allowed by the ADC (3.3 V). The capacitor C1 was added to improve stability.

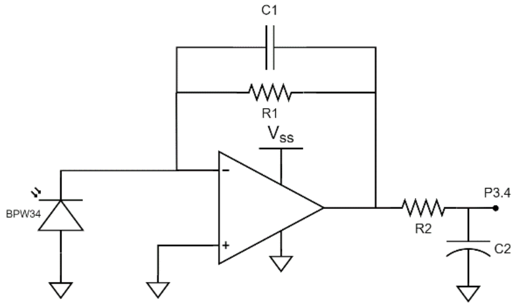


Figure 2. Circuit used for measurement of the solar irradiance, including the photodiode and the signal conditioning circuit.

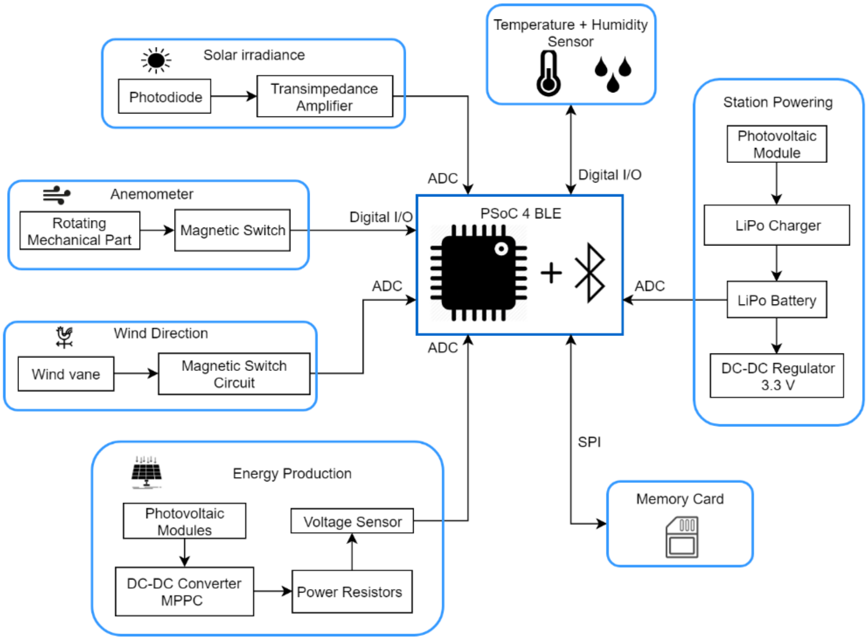


Figure 3. Block diagram of the hardware used in the weather station.

For this purpose, one of the programmable operational amplifiers integrated inside the PSoC 4 BLE module was used in the circuit. The inputs and output of the operational amplifier were programmed to be connected to pins of the microcontroller using the PSoC Creator IDE. A low-pass filter was placed at the output of the amplifier to filter ambient noise, with a cut-off frequency of 10 Hz. A precision pyranometer was used to correlate the output to provide a solar irradiance measurement in W/m^2.

The DHT22/AM2302 [13] sensor was used to measure both the temperature and the humidity. This sensor can measure temperatures from -40°C to 80°C and relative humidity from 0 to 100%. The output of this sensor is a 1-wire bus digital signal that can be read in a digital pin of the microcontroller. The data packet is composed of 40 bits, with 16 bits for the relative humidity data, 16 bits for the temperature data and 8 bits for checksum.

An anemometer is used to measure the wind speed. This device uses a rotational axis with a magnet attached. When rotating, the magnetic force produced by this magnet is detected by a reed switch. The sensor output is connected to a digital pin of the microcontroller, in which an external interrupt is generated. The wind speed is calculated from the time required by the anemometer to complete a turn. A pull-down resistor is used in the pin to ensure that it is not floating.

The wind direction is determined using an electronic wind vane. This device also uses magnets attached to the rotational axis, together with a circuit consisting of eight reed switches, each one corresponding to a possible direction. Depending on the position of the rotational axis and magnet, up to two switches are activated. Each switch is connected to a resistor of different value. Another resistor at the output of the circuit forms a voltage divider with the resistor, or resistors, associated with the activated switches. The output of the voltage divider is fed into the ADC of the microcontroller and its value is associated with a direction.

In the weather station, three PV modules of different technologies were installed: monocrystalline, polycrystalline and amorphous silicon. For comparison purposes, in principle, all the PV modules need to have the same area, but it was not possible to acquire modules that satisfied this condition. Nevertheless, this problem can be easily solved by calculating the average power per 1000 mm^2. Table 1 presents the characteristics of the acquired PV modules.

In order to maximize the power extraction from the PV modules and according to the characteristics of each module, the DC-DC converter LTC3129 [14] was chosen. This converter has a buck-boost topology and integrates the MPPC (Maximum Power Point Control) functionality. It provides a maximum output current of 200 mA and its quiescent current is very low, which is suitable for systems that run on a battery. The converter allows an input voltage from 2.42 V to 15 V.

Figure 4 presents the circuit adopted for the DC-DC converter LTC3129. The datasheet was followed in order to select the components. With the MPPC functionality, the converter will try to have the input voltage close to a set reference voltage. This reference voltage can be set using an external voltage divider connected to the converter pin associated with the MPPC functionality (R5 and R6). The value of this voltage depends on the characteristics of the PV modules. The output voltage can also be set if needed, in this case with resistors R1 and R2 and its value can be from 1.4 V to 15.75 V.

Table 1. Characteristics of the PV modules installed in the weather station.

	Mono	Poly	Amorphous
Maximum Power	1 W	1 W	0.6 W
Short circuit current	250 mA	250 mA	300 mA
Maximum power point current	200 mA	200 mA	200 mA
Open circuit voltage	6 V	6 V	4.15 V
Maximum power point voltage	5 V	5 V	3 V
Total area	5100 mm^2	5580 mm^2	6696 mm^2

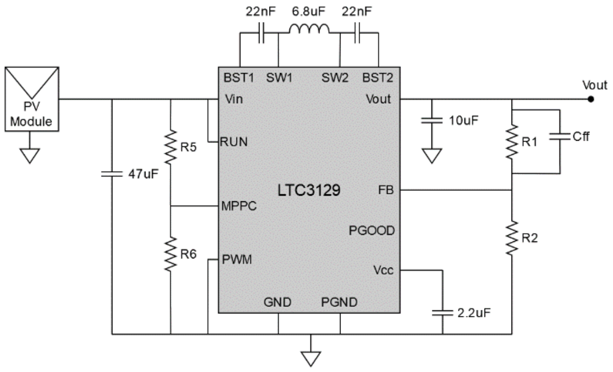


Figure 4. Schematic of the circuit designed using the converter LTC3129 to maximize power extraction from the PV modules.

Each converter has a power resistor (R) as load, with the same value, for comparison purposes. The power (P) in each PV module is calculated according to equation (1) and will depend on the output voltage (V).

$$P = \frac{V^2}{R} \tag{1}$$

A station powering system was developed to recharge the battery of the weather station using a PV module. This system uses an extra PV module not related to the previously mentioned modules used for energy production measurement. This PV module is directly connected to a LiPo (Lithium Polymer) battery charger, which is based on the TP4056 [15]. The LiPo battery is charged when the weather conditions allow it and the battery is not full or close to fully charged.

The voltage of the battery is regulated to 3.3 V, which is the voltage used to power all the weather station components.

To acquire the data from all the sensors and the PV modules, various peripherals were programmed in the PSoC module. A timer was programmed to establish the polling rate at which the data from the sensors are acquired. This value is set by the user in the Android mobile app. This timer functions in interrupt mode, activating a flag inside the ISR (Interrupt Service Routine). This flag is then verified in the main loop and, in case of being activated, the values are acquired from the sensors. The flowchart of the data acquisition process is presented in Figure 5.

To measure the solar irradiance and wind direction, the ADC of the microcontroller was programmed with the maximum resolution of 12 bits, a sample frequency of 4 kHz and a voltage reference of 3.3 V. For measurement of the wind speed, a counter was programmed in interrupt mode to count the amount of time for a rotation to be completed and then the value is converted to RPM (Revolutions Per Minute).

The data from all the sensors is periodically saved in a text file corresponding to the day of acquisition. This process is done until it reaches the daily end time, defined by the user in the mobile application.

An SD card is used to store the measured data locally in the weather station. For this purpose, a module to interface the SD card to the microcontroller was used. This module communicates with the microcontroller using the SPI protocol.

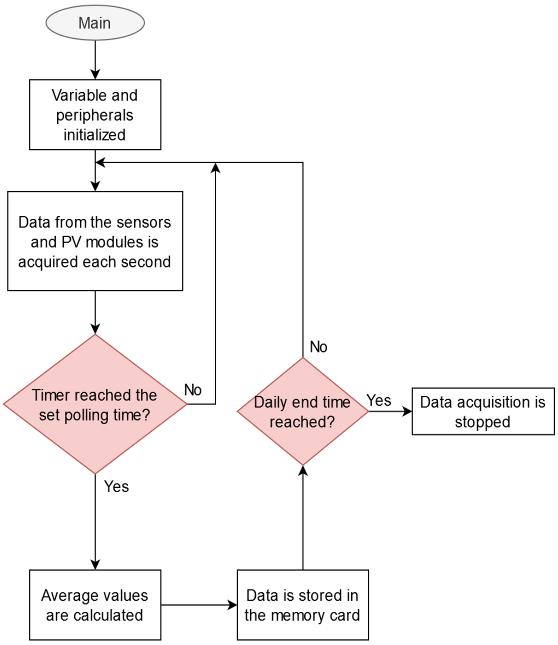

Main
Variable and peripherals initialized
Data from the sensors and PV modules is acquired each second
Timer reached the set polling time?
No
Yes
Average values are calculated
Data is stored in the memory card
Daily end time reached?
No
Yes
Data acquisition is stopped

Figure 5. Flowchart of the data acquisition process in the main function of the program.

A communication block in the microcontroller was programmed as SPI master, whereas the SD card module works as SPI slave. The open source file system FatFs [16] is used to read and write the text files in the SD card. A new text file is created to store all the data acquired on each day. Another text file, called dates.txt, is used to store the starting and end times of acquisition for all the days, as well as the sampling rate. Every time there is new data to save in the SD card, it checks if it is a new day or is it still the same day and acts accordingly.

3.2 Wireless data transfer

The BLE network was programmed in the PSoC module using the BLE component, where is placed the design diagram of the PSoC Creator IDE. Though the BLE component, various parameters of the BLE network can be configured, such as the connection interval, the advertising interval and the MTU (Maximum Transmission Unit) size. This component also allows the creation of the GATT (Generic Attribute Profile) services and characteristics that are used to transfer the data through the BLE network. The GAP (Generic Access Profile) role of this component was defined as peripheral device, since it will advertise its presence to the central device (Android smartphone), which will be responsible for establishing the BLE connection. The peripheral device component was also defined as GATT server, since it will provide the data requested by the GATT client (central device).

The data throughput of the BLE connection was maximized in order to minimize the data transfer time through the configuration of two main BLE parameters: the maximum packet length (MTU size) and the packet transmission interval (BLE connection interval). The MTU size can be configured to a value between 23 bytes and 512 bytes, whereas the connection interval can be set to a value between 7.5 ms and 4 s, with higher energy consumption for lower values. Since the Android device acts as the GAP central device in the developed system, it decides the values used for the connection interval and MTU size. The MTU size chosen was 500 bytes. It is not possible to configure the value of the connection interval on the Android device, but it allows defining the priority of the connection to balanced (default) or high. Although not explicit in the documentation of the BLE Android API [17], choosing a connection with high priority allows higher data rate, which implies the use of a lower connection interval.

A GATT service and its characteristics were then created to accommodate all the data transfer needs. Since the data to be transferred can take up to a few dozens of kilobytes per text file, a characteristic called *DataArray*, with a type of 16-bit integer array and a length of 250 values, was created. With a maximum of 500 bytes to be transferred at once, this characteristic needs to be read several times. For that purpose, an algorithm to transfer the data was created, as shown in Figure 6.

After reading the data from the text file, it is saved in a buffer in the RAM (Random Access Memory). It was also created a characteristic called NumBytes, used to notify the Android device of the amount of data about to be received per packet. Another characteristic, called Next, was also created

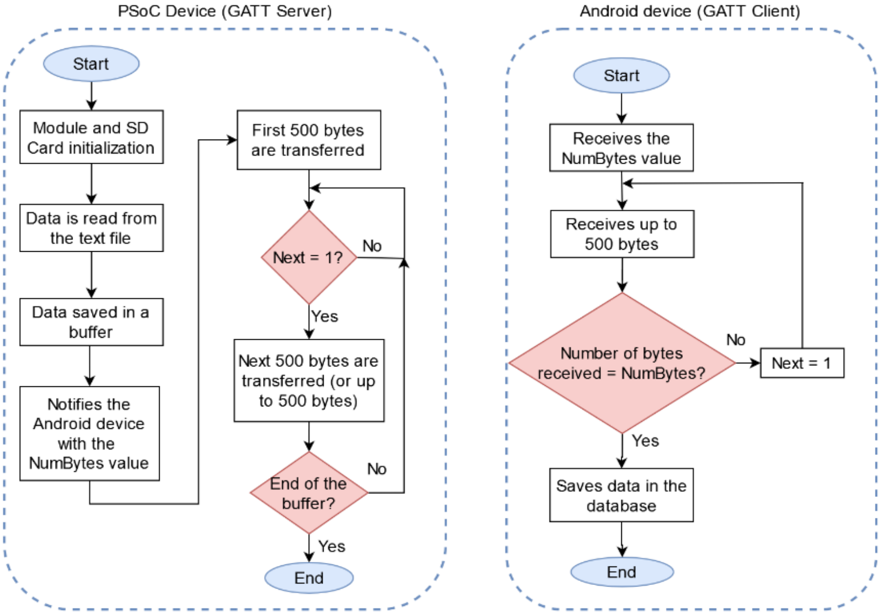

PSoC Device (GATT Server)
Start
Module and SD Card initialization
Data is read from the text file
Data saved in a buffer
Notifies the Android device with the NumBytes value
First 500 bytes are transferred
Next = 1?
No
Yes
Next 500 bytes are transferred (or up to 500 bytes)
End of the buffer?
No
Yes
End
Android device (GATT Client)
Start
Receives the NumBytes value
Receives up to 500 bytes
Number of bytes received = NumBytes?
No
Next = 1
Yes
Saves data in the database
End

Figure 6. Developed algorithm for data transfer using a BLE connection.

to inform the GATT server (PSoC device) when the DataArray characteristic was read. After receiving each 500-byte data segment from the buffer, the Android device writes the value ‘1’ in the Next characteristic, meaning that it is ready to receive the next 500 bytes from the buffer (or up to 500 bytes, in the case of the last data packet). This process is done multiple times until the end of the buffer.

3.3 Mobile application

An Android mobile app was developed in order to handle the BLE data transfer and to allow the storage and visualization of the data received from the weather station. The mobile app also allows the configuration of some parameters related to the data acquisition performed by the weather station. The mobile app was developed using the Android Studio IDE. To use the Bluetooth hardware of the Android device, it was necessary to require some permissions in the Android manifest. The API available on the Android Developers website was followed [18], as well as the Cypress Semiconductor examples [19].

Figure 7 presents a flowchart of the developed mobile application. An activity (*Scan Activity*) was implemented to scan for BLE devices. This activity displays the device names or addresses, as well as their RSSI (Received Signal Strength Indication). The user can select the device in the list correspondent to the weather station.

Another activity, called *Control Activity*, was created. This activity is divided in two fragments. The first fragment, *Home Fragment*, allows the user to visualize the current data measured by the station as well as setting the daily start and end times of acquisition and the sampling period. The second fragment, *Data Fragment*, is dedicated to data transfer and data visualization. The user can choose a date of the day they want to download the data or download all the data stored in the weather station. A list is updated with all the dates of the downloaded data, which is stored in a SQLite database.

Upon selecting a date in the list, a new activity is opened. This activity, called *Graph Activity*, allows the user to visualize the data of that day in graphs, being able to choose a variable to be visualized. The development of this activity used the open source library MPAndroidChart [20].

A service called *PSoCStationService* was created to deal with the connection and transfer of data via BLE. The connection to the GATT server of the BLE device is established within this service. After a successful connection to the server, the process of service discovery is started. The data transfer process is also handled inside this service. The results regarding the BLE connection, as well as the characteristic that was read or written, are returned in a callback function called BluetoothGattCallback.

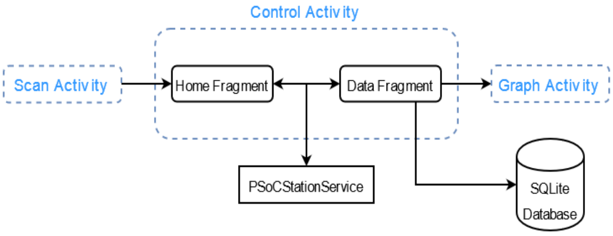


Figure 7. Architecture of the developed Android application.

4 Experimental results

This section presents the main results obtained from experimental tests, which are used to validate the implementation of the whole system, ranging from the data acquisition in the weather station to the visualization in a smartphone as well as the performance analysis of the data transfer process. Figure 8 shows the prototype of the weather station that was developed.

The weather station includes the temperature and humidity, solar irradiance, wind speed and wind direction sensors, as well as the photovoltaic modules. The electronics developed to acquire and transfer all the data, as well as the battery, were placed in a dedicated enclosure. A 40º inclination in relation to the vertical axis was used for the photovoltaic modules, which is the recommended value in the location where the weather station was placed for the month when it was tested. Figure 9 shows the inside of the electronics enclosure.

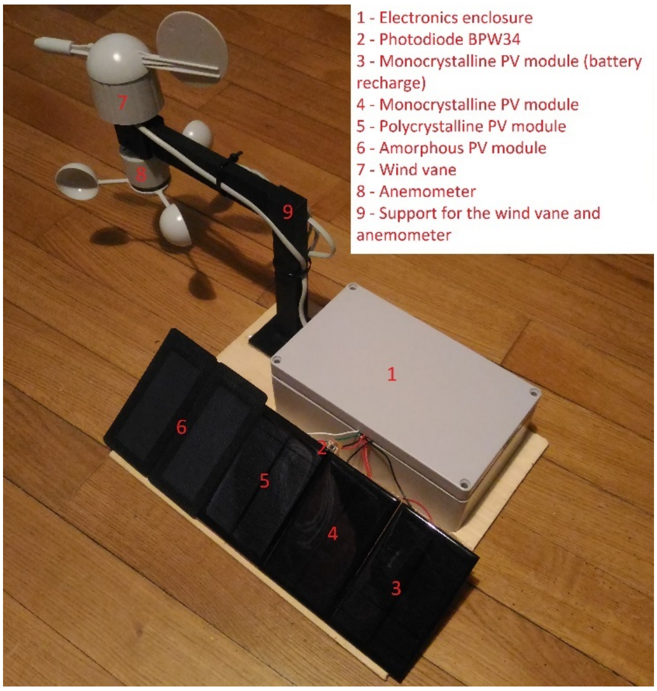

1 - Electronics enclosure
2 - Photodiode BPW34
3 - Monocrystalline PV module (battery recharge)
4 - Monocrystalline PV module
5 - Polycrystalline PV module
6 - Amorphous PV module
7 - Wind vane
8 - Anemometer
9 - Support for the wind vane and anemometer
1
2
3
4
5
6
7
8
9

Figure 8. Developed prototype of the weather station.

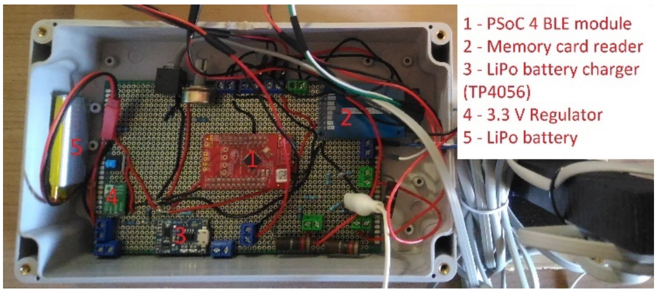
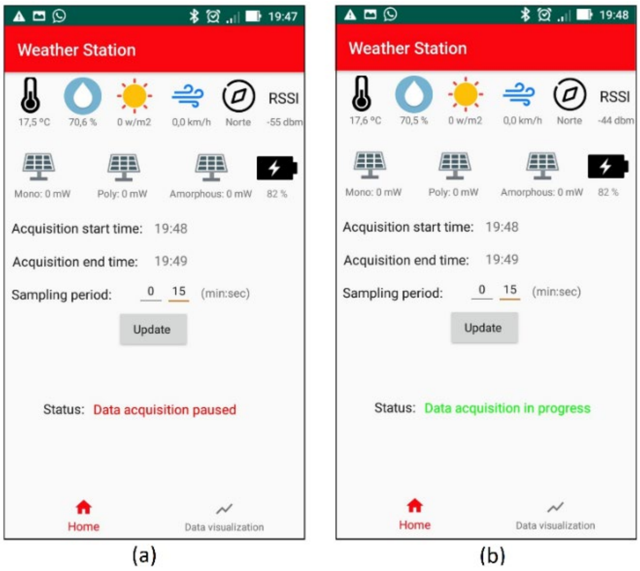


Figure 9. Detail of the developed electronics included in the prototype of the weather station.

4.1 Data acquisition and transfer

The developed mobile app has a bottom navigation menu that allows the user to transition between its screens. After connecting to the weather station, the default screen is the home screen, as shown in Figure 10.



(a)

17,5 °C 70,6 % 0 w/m2 0,0 km/h Norte -55 dbm

Mono: 0 mW Poly: 0 mW Amorphous: 0 mW 82 %

Acquisition start time: 19:48

Acquisition end time: 19:49

Sampling period: 0 15 (min:sec)

Update

Status: Data acquisition paused

(b)

17,6 °C 70,5 % 0 w/m2 0,0 km/h Norte -44 dbm

Mono: 0 mW Poly: 0 mW Amorphous: 0 mW 82 %

Acquisition start time: 19:48

Acquisition end time: 19:49

Sampling period: 0 15 (min:sec)

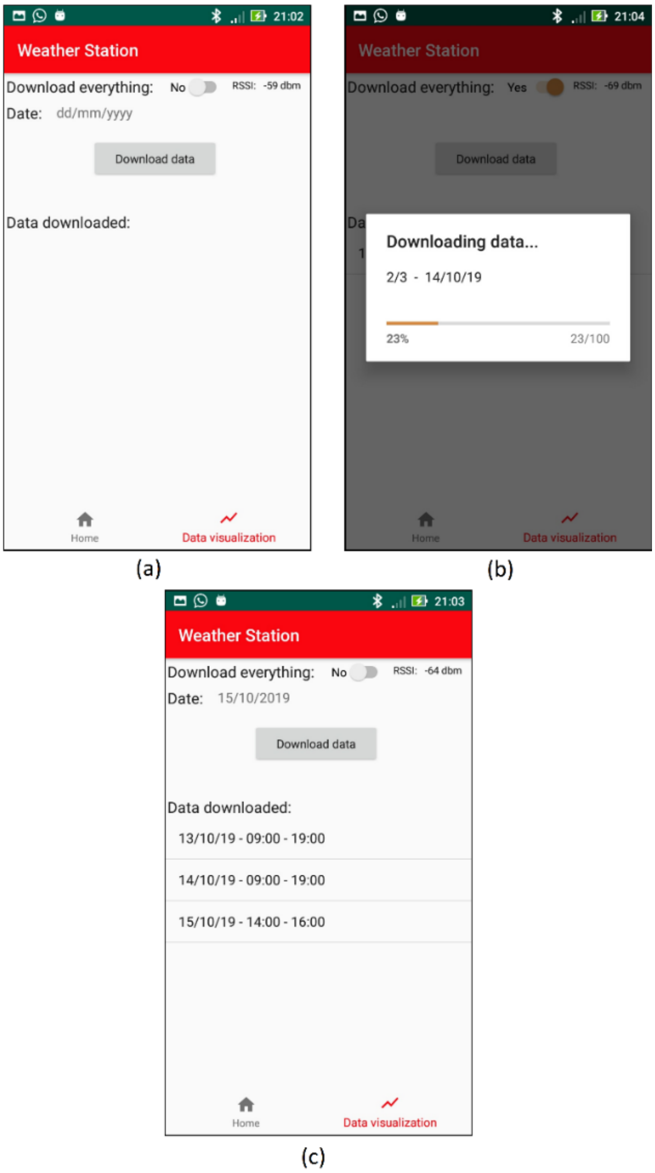
Update

Status: Data acquisition in progress

Figure 10. *Home Fragment* screen of the developed mobile app. (a) Before starting the acquisition. (b) After starting the acquisition.

This screen allows the user to set the start and end times of the acquisition, as well as the sampling period (in this case, a fragment dedicated to choosing the hours and minutes will pop up). Even if there is data being acquired, the new values of start and end times, as well as the sampling period, will be updated for the next day. The default value of the sampling period is 15 seconds. When the data acquisition starts, the *TextView* elements that present the last acquired values from all the weather station sensors are continuously updated in real-time (top of the screen).

The second screen, shown in Figure 11, is dedicated to download and visualize the weather station data. Here, the user can choose a day to download the respective acquired data or enable an option to download the data acquired from all the available days. A progress bar is updated as the data is transferred. The data is stored in a SQLite database. After finishing the download, the list of available dates and respective start and end times is updated in the screen, as shown in Figure 11 (c). If the user selects one of the dates by tapping on it, the activity dedicated to the visualization of data in graphs is opened.



(a)

Weather Station

Download everything: No

RSSI: -59 dbm

Date: dd/mm/yyyy

Data downloaded:

(b)

Weather Station

Download everything: Yes

RSSI: -69 dbm

Downloading data...

2/3 - 14/10/19

23%

23/100

(c)

Weather Station

Download everything: No

RSSI: -64 dbm

Date: 15/10/2019

Data downloaded:

13/10/19 - 09:00 - 19:00

14/10/19 - 09:00 - 19:00

15/10/19 - 14:00 - 16:00

Figure 11. *Data Fragment* screen of the developed application. (a) Options to download the data. (b) Dialog that appears when downloading. (c) After downloading the data.

4.2 Data transfer performance

This section presents experimental results regarding the total time needed to transfer data from the weather station to the Android mobile device, as well as the corresponding throughput. The total transfer time can be divided in two phases, as shown in Figure 12. When the user requests data using the mobile app, the PSoC microcontroller in the weather station first reads the data from the SD card into a buffer. The buffer is located in the SRAM (Static Random-Access Memory) memory of the microcontroller and its size (12960 bytes) was set taking into consideration the total SRAM available (32 KB). In the second phase the data is read from the buffer 500 bytes at a time (corresponding to the configured MTU value), and transferred to the Android device via BLE.

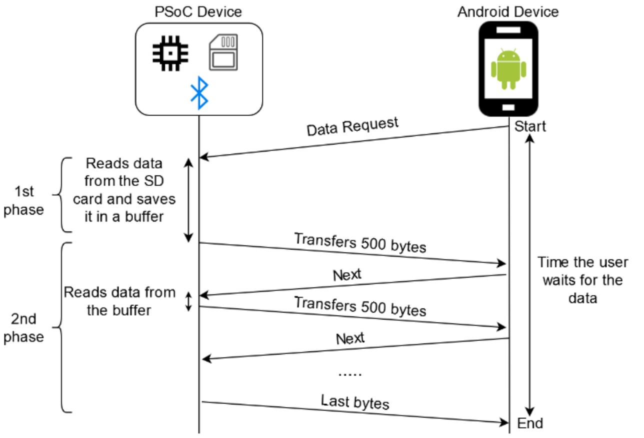


Figure 12. Test setup for measurement of the total transfer time from the weather station to the Android mobile device.

The total transfer time was measured on the Android device, since the user presses the download button until all the requested data is transferred. In this test scenario, the total data to be transferred was configured to 10800 bytes, which corresponds to the amount of data collected by the weather station for a period of 9 hours on a given day. The priority of the connection was set to high, which, according to the documentation [17], guarantees a fast and low latency connection. The Android device used in the tests presented in this section was the Asus Zenfone 2, with version 6.0 of the Android operating system. The data transfer results are presented in Table 2. This test was performed ten times.

Table 2. Measured data transfer time and corresponding average throughput between the weather station and the Android device.

Bytes	Minimum time (s)	Average time (s)	Maximum time (s)	Throughput (kbps)
10800	10.69	11.06	11.45	7.81

The data transfer time for a single transfer through the BLE wireless network was also measured, using the test scenario presented in Figure 13. The measurement was performed in the Android device, from the instant the function that requests the reading of the BLE characteristic that contains the data from the GATT server is executed until the onCharacteristicRead() function is called in the callback function corresponding to the read characteristic.

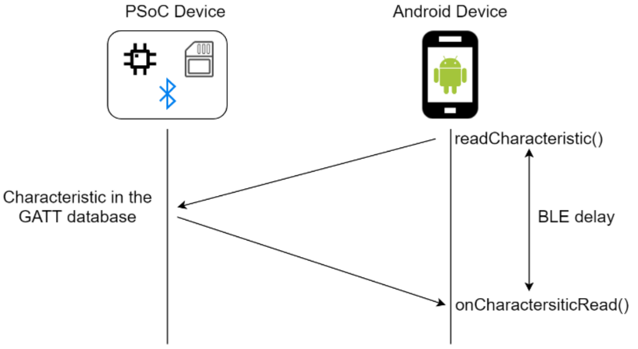


Figure 13. Test scenario to measure the BLE delay.

Table 3 presents the results from this test. The amount of data transferred was 500 bytes and the test was repeated 22 times. This measurement was also executed using two different priorities in the connection (configurable through the BLE Android API): balanced and high priority, to assess its impact on the performance.

With a high priority connection, the data rate increases to more than double compared to the default balanced connection. This is probably because the connection interval is decreased or the number of packets sent per BLE connection interval is increased, although it was not possible to verify because the behaviour for each connection priority is not specified in the documentation of the BLE Android API.

Given that the throughput obtained in Table 2 (7.81 kbps), performed with high connection priority, is much lower than the corresponding value obtained in Table 3 (25.32 kbps), it can be concluded that the throughput is severely affected by the extra processing time required for the PSoC device to read the data from the SD card and from the buffer before transmission.

The time necessary to read the data from the SD card in the weather station was measured, corresponding to the first phase of the test scenario mentioned in Figure 12. The time is measured since the function to read the data is called until all the data is transferred to the buffer. To transfer 10800 bytes, an average time of 5.036 seconds was measured to read the data from SD card to the buffer. This is almost half of the time required to transfer the data to the Android device, as shown in Table 2 (11.06 s), which confirms the negative impact of this processing time in the achieved throughput.

Although the PSoC device was configured for maximum performance, the maximum throughput obtained in these experimental tests was much lower than the BLE bit rate (1 Mbps). Part of the justification is due to the overheads imposed by the BLE protocol, such as the data packet headers, the guard times between packets and the control packets, but a substantial part may be due to additional limitations in the implementation of the software stacks in both the PSoC device and the Android smartphone, since the maximization of the throughput sometimes is overlooked in the implementation of BLE devices [21]. In this sense, the connection interval used by the BLE Android API (not referred in the documentation) is probably much higher than the minimum value defined in the BLE specification (7.5 ms), even for the high priority option.

As an alternative to BLE that may be suitable to achieve better communication performance, we propose the use of an IEEE 802.11/Wi-Fi wireless network [22]. These networks provide much higher bit rates (which have been increasing with each new version). Although it tends to have higher energy consumption than BLE, this is not a concern for the application considered in this paper, given that the data transfer times tend to be short. On the other hand, other wireless network alternatives, like LoRa [23] and ZigBee [24], would not be adequate due to their lack of native support in smartphones and lower data rates than BLE.

Table 3. Data transfer delay (in seconds) and throughput from the PSoC device to the Android smartphone using BLE and different connection priorities.

Connection Priority	Bytes	Minimum time (s)	Average time (s)	Maximum Time (s)	Throughput (kbps)
Balanced (default)	500	0.186	0.329	0.542	12.16
High	500	0.105	0.158	0.442	25.32

4.3 Data visualization

Figure 14, Figure 15 and Figure 16 show examples of the activity *Graph Activity*, which is opened after the user taps in one of the dates in the list provided by the *Data Fragment*. Through a set of checkboxes, the user can select the variable to display in the graph. The x-axis represents the time from the start to end of data acquisition of that day. It is possible to zoom in the graph to view a certain value at a specific time if necessary.

The data shown in Figure 14 corresponds to the measurements of temperature and humidity acquired in October 26, 2019 between 16:00 and 19:00. In this case, there was a negative correlation between the temperature and the humidity, with the temperature decreasing along the time as the night approaches, while the humidity increases.

Figure 15 presents the results of the battery recharging and its relationship with the current solar irradiance. The data was acquired on November 18, 2019 between 13:20 and 17:30.

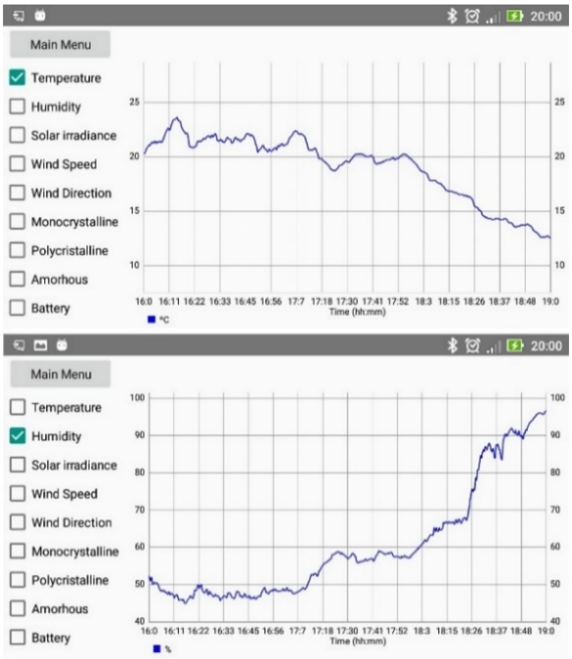


Figure 14. Graphs of the temperature and humidity measured by the weather station along the time provided by the developed mobile app.

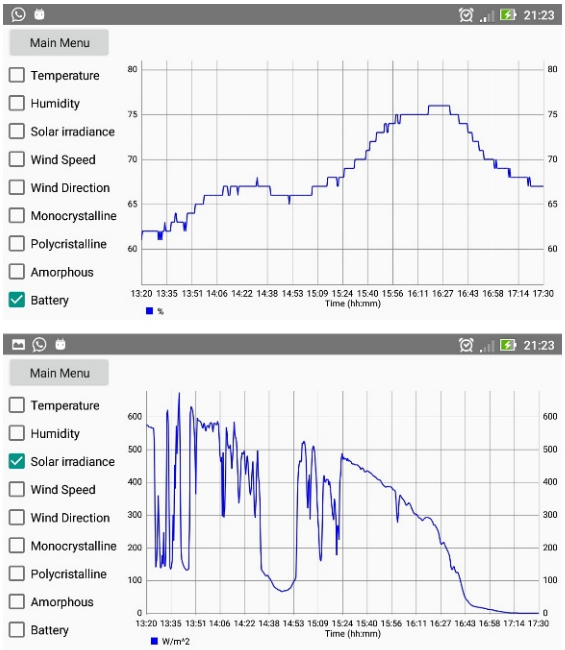


Figure 15. Measurements of the weather station battery capacity (with recharging though a PV module) and the solar irradiance along the time.

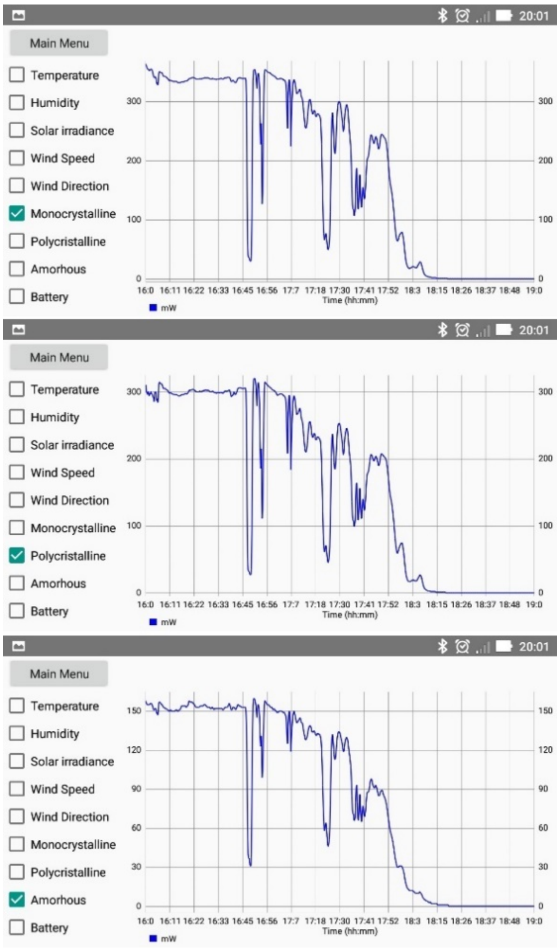


Figure 16. Power produced by each PV module installed in the weather station for 3 hours.

Figure 16 shows the measured power in all the three PV modules installed in the weather station. It can be seen that the monocrystalline module produced the most amount of power and, as expected, the amorphous module produced the least amount of power.

5. Conclusions

This paper presents the development of a weather station dedicated to renewable energy applications, which also includes a mobile app that provides a user interface and allows data transfer, storage and visualization. The developed system is composed by various sensors used to measure the state of the weather, as well as the power production from the three different technologies of solar photovoltaic modules, installed in the weather station (a module of each type: polycrystalline, monocrystalline and amorphous silicon). A solution to maximize the power extraction from the PV modules in the weather station was presented. The key purpose of using these different solar photovoltaic modules is measuring the produced power for each one of them in order to gauge which technology would be more suitable for the specific location where the weather location is installed.

All the data acquired is processed and stored locally in the weather station. When necessary, the user can download the data to an Android mobile device through a BLE wireless network connection using the developed mobile app, presented in this paper. This transferred data is stored in a SQLite database and allows the user to visualize it, with the help of dedicated charts for the measured variables.

The system was validated through experimental tests, by acquiring data using all the sensors and the PV modules, storing the information locally in the weather station, transferring the data to an Android smartphone via BLE, storing it in the smartphone, and presenting the data to the user.

The data transfer performance was evaluated in different scenarios, through measuring the time to download data from the weather station, as well as measuring the throughput from the BLE connection under different connection priorities. It was concluded that the data transfer time is severely affected by the extra processing time required at the PSoC device, as well as by limitations in the implementation of the BLE Android API.

Acknowledgements.
This work was supported by FCT national funds, under the national support to R&D units grant, through the reference project UIDB/04436/2020 and UIDP/04436/2020.

References

[1] F. R. Pazheri, M. F. Othman, and N. H. Malik, “A review on global renewable electricity scenario,” Renew. Sustain. Energy Rev., vol. 31, pp. 835–845, 2014.

[2] Q. Raza, M. Nadarajah, and C. Ekanayake, “On recent advances in PV output power forecast,” Sol. Energy, vol. 136, pp. 125–144, 2016.

[3] S. Lo Piano and K. Mayumi, “Toward an integrated assessment of the performance of photovoltaic power stations for electricity generation,” Appl. Energy, vol. 186, pp. 167–174, 2017.

[4] V. Perraki and P. Kounavis, “Effect of temperature and radiation on the parameters of photovoltaic modules,” Journal of Renewable and Sustainable Energy, vol. 8, 2016.

[5] J. A. Salgado, J. A. Afonso, V. Monteiro, J. L. Afonso, M. C. Feio, L. M. Silva, “Development of a Compact and Low-Cost Weather Station for Renewable Energy Applications”, EAI International Conference on Sustainable Energy for Smart Cities, December 2019, Braga, Portugal.

[6] D. Hortelano, T. Olivares, M. C. Ruiz, C. Garrido-Hidalgo, and V. López, “From sensor networks to internet of things. Bluetooth low energy, a standard for this evolution,” Sensors (Switzerland), vol. 17, no. 2, pp. 1–31, 2017.

[7] P. K. Nayak, S. Mahesh, H. J. Snaith, and D. Cahen, “Photovoltaic solar cell technologies: analysing the state of the art,” Nat. Rev. Mater., vol. 4, no. 4, pp. 269–285, 2019.

[8] C. Ranhotigamage and S. C. Mukhopadhyay, “Field trials and performance monitoring of distributed solar panels using a low-cost wireless sensors network for domestic applications,” IEEE Sens. J., vol. 11, no. 10, pp. 2583–2590, 2011.

[9] H. R. Iskandar, A. Purwadi, A. Rizqiawan, N. Heryana, “Prototype development of a low cost data logger and monitoring system for PV application,” 3rd Conference on Power Engineering and Renewable Energy, November 2016, Yogyakarta, Indonesia.

[10] A. Kovacs, A. Nicolcioiu, J. Arhip, and G. Caşu, “Design and implementation of a GPRS remote data logger for weather forecasting,” IEEE Int. Conf. Commun., 2014.

[11] Cypress Semiconductor, “CY8CKIT-042-BLE-A Bluetooth Low Energy 4.2 Compliant Pioneer Kit.” [Online]. Available: https://www.cypress.com/documentation/development-kitsboards/cy8ckit-042-ble-bluetooth-low-energy-42-compliant-pioneer-kit. [Accessed: 09-Jul-2019].

[12] Vishay Semiconductors, “bpw34.pdf.” [Online]. Available: https://www.vishay.com/docs/81521/bpw34.pdf. [Accessed: 09-Jul-2019].

[13] WaveShare, “DHT22 Temperature-Humidity Sensor.” [Online]. Available: https://www.waveshare.com/wiki/DHT22_Temperature-Humidity_Sensor.

[14] Linear Technology, “LTC3129: 15V, 200mA Synchronous Buck-Boost DC/DC Converter with 1.3μA Quiescent Current”. [Online]. Available at: https://www.analog.com/media/en/technical-documentation/data-sheets/3129fc.pdf. [Accessed: 25-Aug-2020].

[15] NanJing Top Power ASIC Corp., “TP4056”. [Online]. Available at: https://dlnmh9ip6v2uc.cloudfront.net/datasheets/Prototyping/TP4056.pdf. [Accessed: 27-Jul-2019].

[16] Elm-chan.org, “FatFs - Generic FAT Filesystem Module”. [Online]. Available at: http://elmchan.org/fsw/ff/00index_e.html. [Accessed: 26-Jul-2019].

[17] Android Developers, “BluetoothGatt | Android developers “. [Online]. Available at: https://developer.android.com/reference/android/bluetooth/BluetoothGatt.html#requestConnectionPriority(int). [Accessed: 18-Set-2019].

[18] Android Developers, “Bluetooth Low Energy Overview | Android Developers”. [Online]. Available at: https://developer.android.com/guide/topics/connectivity/bluetooth-le. [Accessed: 18-Set-2019].

[19] Cypresssemiconductor, “GitHub - cypresssemiconductor/bleapp”. [Online]. Available at: https://github.com/cypresssemiconductorco/bleapp. [Accessed: 17-Nov-2019].

[20] P. Jahoda, “MPAndroidChart”. [Online]. Available at: https://github.com/PhilJay/MPAndroidChart. [Accessed: 16-Set-2019].

[21] J. A. Afonso, A. J. F. Maio, e R. Simoes, “Performance Evaluation of Bluetooth Low Energy for High Data Rate Body Area Networks”, Wireless Personal Communications, vol. 90, n. 1, pp. 121–141, 2016.

[22] G. R. Hiertz, D. Denteneer, L. Stibor, Y. Zang, X. P. Costa, and B. Walke, “The IEEE 802.11 universe,” IEEE Commun. Mag., vol. 48, no. 1, pp. 62–70, 2010.

[23] U. Raza, P. Kulkarni, and M. Sooriyabandara, “Low Power Wide Area Networks: An Overview,” IEEE Commun. Surv. Tutorials, vol. 19, no. 2, pp. 855–873, 2017.

[24] P. Baronti, P. Pillai, V. W. C. Chook, S. Chessa, A. Gotta, and Y. F. Hu, “Wireless sensor networks: A survey on the

state of the art and the 802.15.4 and ZigBee standards," Comput. Commun., vol. 30, no. 7, pp. 1655–1695, 2007.