Webology (ISSN: 1735-188X)
Volume 19, Number 5, 2022

Real-Time Monitoring Of An Electrical Prototype Using A Microcontroller For A Photovoltaic Microgrid

Javier E. Sierra[1], Alejandro S. Guerrero Hernández[1] and José L. López Prado[1]

[1] Faculty of Engineering, Universidad de Sucre, Sincelejo, Sucre- Colombia.

Correspondence: Author: Alejandro S. Guerrero Hernández

Abstract

Electrical energy is linked to two signals, current and voltage; before feeding a load, both are sinusoidal signals with equal frequencies but different amplitudes. Micrograms commonly include a communication system that connects all parts of the system. This study builds a prototype of an electrical system with a PIC 18F4550 microcontroller to monitor loads of the University of Sucre microgrid in real-time. They are also designed in MATLAB, a platform to monitor voltage and current variables. The proposed system will be considered an essential solar energy exploitation project component.

Keywords: microgrid, microcontroller, prototype, communications device class.

I. INTRODUCTION

Electrical energy is one of the agents that have promoted the comfort and well-being of humanity. But at the same time that it provides quality of life, its demand increases exponentially while conventional generation strategies have some restrictions that do not facilitate their expansion. This last dilemma has motivated the study of alternative sources of generation, which are based on the use of inexhaustible energy resources such as solar radiation, and wind currents, among others.

The electrical energy is linked to two signals, current, and voltage, before feeding a load; both are sinusoidal signals with equal frequencies but different amplitudes. The current, the amplitude, is a function of the load and the voltage is generally constant. This can be considered an ideal case since, in reality, on the one hand, the elements that makeup loads of electrical systems are not linear.

A microgrid is a small power grid containing controllable loads, distributed generation sources, and energy storage (Kim & Kinoshita, 2010; Zhang, 2013). Microgrids commonly include a communication system that connects all parts of the microgrid. This system transfers and monitors the information collected from all parts of the microgrid to ensure the best management and control (Ray, 2016).

The design and implementation of supervisory control and data acquisition (SCADA) system accompanied by the internet of things (IoT) is a case study in microgrids where the microcontroller is used as a remora terminal unit to obtain data from sensors and communicate with the master terminals (Duair, Jaafar J and Majeed, Ammar I and Ali, 2022). In Kaya, Ibrahim Berkin and Kalayci, (2022) Presents a microgrid that includes the battery, solar energy, and wind energy, designing a SCADA system that seeks through an Arduino microcontroller to control each of the elements that make up the hybrid system. In Correia et al., (2022), an intelligent thermostat with a microcontroller was implemented to monitor and control air quality and thermal comfort within classrooms, integrate into the building's microgrid, and seek energy efficiency and demand control measures. In Badar et al., (2022) a non-linear control with fuzzy logic and neural networks was designed to manage the energy demand of a microgrid made up of wind energy, photovoltaic, fuel cell, battery, and Supercapacitor. The experimental part was worked with the F28397D microcontroller. Mohammed & Selman (2021) proposed a DC electrical system for exchanging energy between two houses and the primary grid, controlled by Arduino Uno and remote viewing by the Ubidots platform. In addition, the system's power management was done to reduce the cost of electricity as much as possible.

This article focuses on building a prototype electrical system with a PIC 18F4550 microcontroller to monitor loads of the University of Sucre microgrid in real-time. The system allows real-time voltage and electric current monitoring through a MATLAB Guide platform.

II. METHODOLOGY

This project is based on the USB connection between a computer and a PIC 18F4550. Using a programming code written in PIC C will execute orders for supervision of the variables. This will read the ports in which four sensors are connected (2 voltage and two current), quantify, encode, and using CDC (Communications Device Class) communication will

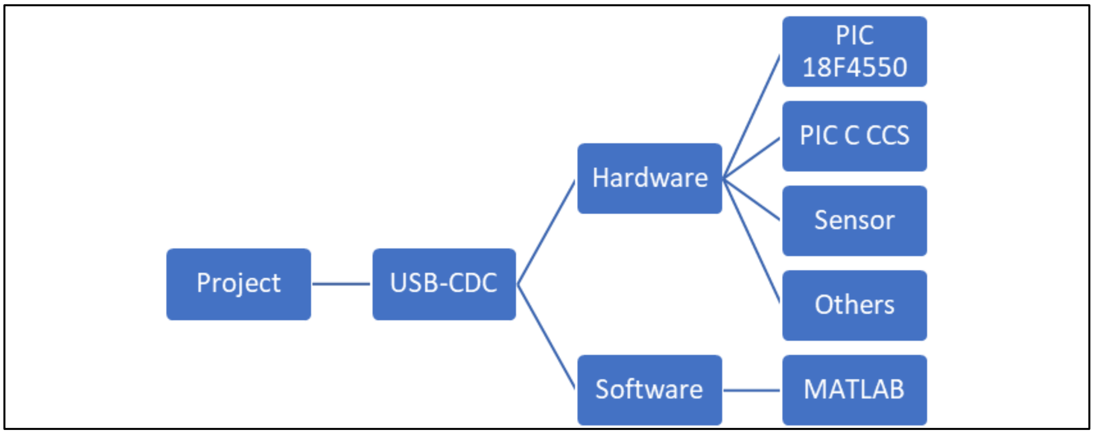

Figure 1. Block diagram of the project

transmit the data to the computer, an interface designed and created in MATLAB with its GUIDE application environment will be graphically displayed.

The hardware is made up of the following elements:

- **Zero crossing detector**. At this stage, the voltage of the network is captured and rectified in half a wave and then turned on the diode of the optocoupler PC817, which in its data sheet has the safe operating voltage of the emitting diode is VF = 1.2v, and an IF = 20mA.
- **Current sensor.** The census of the two current lines, one AC and one DC was carried out through the ACS712-30A sensor. The measuring range of the current sensor ranges from -30A to 30A. In addition, this sensor requires a power supply of 5Vcd, and at its output, it delivers a voltage proportional to the input current in a range of 0 to 5Vcd (nominally 2.5Vdc to a current of 0A). This voltage range fits perfectly into the dynamic range of the PIC analog-to-digital converter used in this project.
- **Voltage Sensor.** The ZMPT101B AC Voltage Sensor is a module used to measure the phase voltage of the Alternating Current. The AC voltage sensor is designed from a transformer, so it can be used to measure the AC voltage.
- **Phase-out circuit.** It was necessary to implement it because observing the signal at the sensor output with the oscilloscope was approximately 51,1° out of phase concerning the network signal (see figure 2).

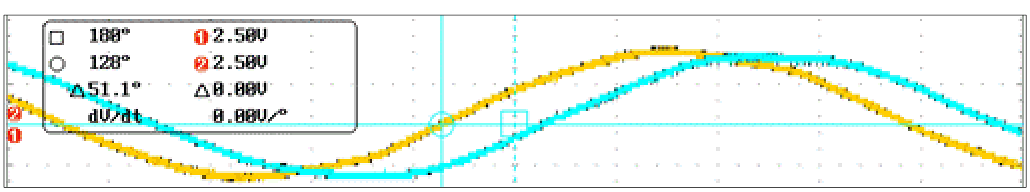

Figure 2. *Sensor lag from the network*

The methodology of the software was as follows: The speed of work of the PIC is defined. A feature that must be taken into account when using a PIC with a USB interface, for this module to work, the clock frequency at its input must be 48 MHz. To achieve their frequency, there is a multiplier with pre-scale and post-scale. At the input of the multiplier, we must have a fixed frequency of 4 MHz.

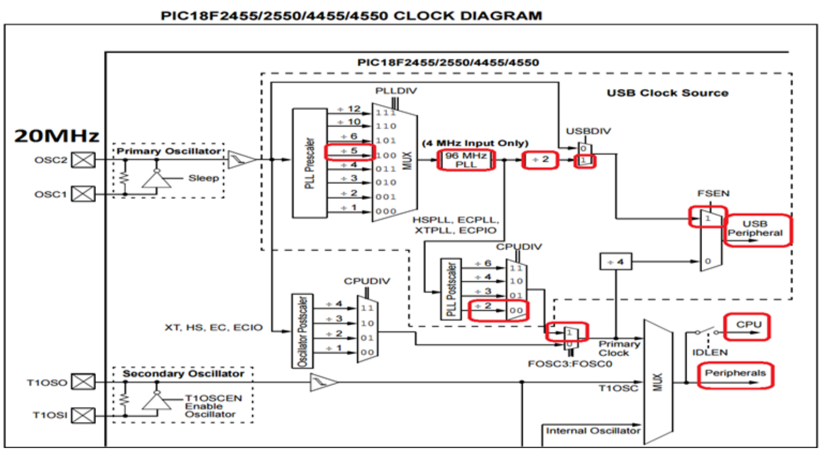

Figure 3. Prescaler configuration of the PLL with a 20 MHz crystal

As seen in figure 3, in this case, the pre-scaler divides the 20 MHz by 5 to obtain the 4MHz required at the input of the PLL (Phase Lock Loop); this, in turn, produces 96 MHz in its output, which is distributed on one side to the USB module, previously dividing the frequency by 2 to obtain the 48 MHz at the input of this and on the other hand feeds the post-scale of the PLL, so that you can choose the working frequency of the PIC core, in this case, it divides by two, so we will have the same 48 MHz to feed the "core" of the PIC.

Communication USB CDC.

For this project, the CDC class (Communications Device Class) is used; the functions that have been used to send and receive data by the USB bus are:

- usb_cdc_kbhit (): is a function that returns the boolean value actual if one or more characters are waiting in the receive buffer.
- usb_cdc_getc (): Gets the received character in the Receive Buffer.
- usb_cdc_putc (char c): Places the character it receives as a parameter in the transmission buffer to be sent.

Development of the MATLAB guide application.

The first step for the realization of this interface was to organize in the MATLAB GUIDE the different elements that were going to be used for the realization of this, leaving the application in this way (see Figure 3):

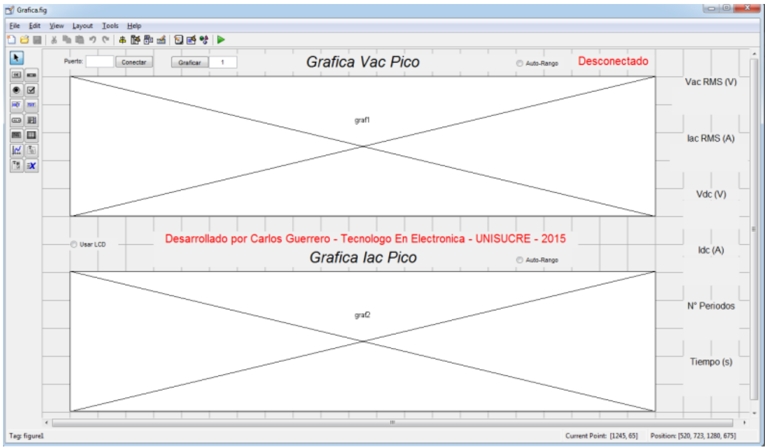

Figure 4. Interface MATLAB

After organizing the elements, we proceed to write the code or function of each element, specifically the buttons and CheckBox that are the ones that govern the operation of all the other components. This interface is responsible for linking the PC with the PIC, so the initial of the code is to establish the walls for such communication; this is done with the following lines of code:

Puerto = serial ('COM'); Save all port settings COM.

Set (puerto,'BaudRate',19200); Speed is set to 19200 Baud.

Set (puerto,'DataBits',8); The data is configured to be 8-bit.

Set (puerto,'Parity','none'); Configures without parity.

Set (puerto,'StopBits',1); Stop bit is set to one.

Set (puerto,'FlowControl','none'); Configures without flow control.

Puerto.InputBufferSize=2000; "n" is the number of bytes to receive.

Puerto.Terminator='LF'; Character with which the reception of data ends.

Fopen (puerto); Open the COM port.

Set (puerto,'Timeout',5); 5 seconds of timeout.

III. SIMULATION AND DISCUSSION OF THE RESULT

For the execution of the monitoring software of the project variables, it is necessary to follow the following steps:

- Have MATLAB installed.
- Connect the circuit to the USB of the PC.

- Install the PIC drivers manually, taking them from the "Driver" folder (CD or download from the WEB).
- Open MATLAB GUIDE and run the interface.
- Go to the Windows device manager and see to which COM port the USB module is connected.
- Enter the COM port number in the MATLAB interface text box and click "CONNECT."
- Write the number of periods you want to graph and click on the "GRAPH" button.

The microcontroller only supports USB 2.0 communication and hence its variants and classes. The results obtained with this data acquisition system are as desired. They are not far from the values obtained with measuring instruments dedicated to such purposes as the multimeter. Figure 4 shows the values obtained without a load connected to the data acquisition system. As can be seen, the voltage value quantified by the ADQ system is 119.704V, compared to the 120.23V measured with the multimeter, obtaining an approximate error of 0.43%.

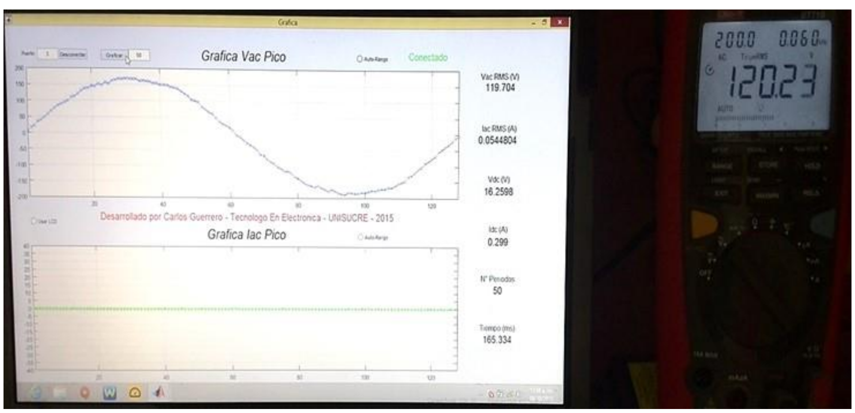

Figure 5. No-load acquisition system

In Figure 5, a load was connected using two multimeters at a time, one to measure the current through the load and one to measure the voltage of the network. As you can see, the voltage and current values thrown by the ADQ system are 118.805V and 7.58A. Compared to the values measured with the multimeters, which were 116.39V and 7.72A, the margin of error between the voltage and current measurements were 2.07% and 1.81%, respectively.

Figure 6. Sistema de adquisición con carga

La Figure 6 shows the prototype for acquiring current and voltage variables with the PIC 18F4550 and communication port with the laptop.

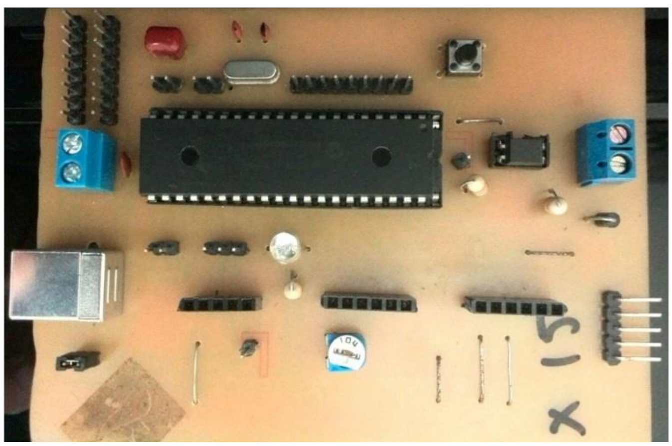

Figure 7. Prototype designed

IV. CONCLUSIONS

For selecting the current sensor, the two most used types of sensors were considered: invasive and non-invasive. Non-invasive sensors were discarded from the beginning due to their high cost. For this reason, it was decided to use the (invasive) ACS712-30A current sensor to acquire the current signal.

It was thought to use a transformer directly when selecting the voltage sensor. Still, it complicated the entire due process that had to be designed and then built said transformer, in addition to a signal conditioning circuit. The PIC18F4550 microcontroller arose to answer how to capture the signals from the sensors. While communicating with the computer through a current communication such as the USB 2.0 protocol to send the data to the computer and process them, this microcontroller was the perfect answer. Through this project, he has approached its practical use, programming, and operation, but above all, it has given him a clear perspective of its possibilities and limitations, a beneficial knowledge.

Finally, we have MATLAB, a set of tools for software development focused mainly on the creation of environments for the development of projects from the most basic as this project; even the most complex, such as aerospace projects

The project developed a prototype with programming in different languages, data acquisition, and digital communication. After making the respective energy conversions, the proposed and designed system will be considered an essential component of the solar energy exploitation project.

REFERENCE

Badar, M., Ahmad, I., Mir, A. A., Ahmed, S., & Waqas, A. (2022). An autonomous hybrid DC microgrid with ANN-fuzzy and adaptive terminal sliding mode multi-level control structure. Control Engineering Practice, 121. https://doi.org/10.1016/j.conengprac.2021.105036

Correia, A., Ferreira, L. M., Coimbra, P., Moura, P., & de Almeida, A. T. (2022). Smart Thermostats for a Campus Microgrid: Demand Control and Improving Air Quality. Energies, 15(4). https://doi.org/10.3390/en15041359

Duair, Jaafar J and Majeed, Ammar I and Ali, G. M. (2022). Design and Implementation of IoT-Based SCADA for a Multi Microgrid System. ECS Transactions, 107, 17345.

Kaya, Ibrahim Berkin and Kalayci, B. and K. (2022). Off-Grid Hybrid (Photovoltaic / Wind / Battery) Microgrid Development : An Experimental Design and Analysis. International Congress on Human-Computer Interaction, Optimization and Robotic Applications, 1–5.

Kim, H. M., & Kinoshita, T. (2010). A new challenge of microgrid operation. Communications in Computer and Information Science, 78 CCIS. https://doi.org/10.1007/978-3-642-16444-6_32

Microchip Technology Inc. (2009). PIC18F2455/2550/4455/4550 Data Sheet. Microchip, 1–438. https://ww1.microchip.com/downloads/en/DeviceDoc/39632e.pdf

Mohammed, N. S., & Selman, N. H. (2021). Real-time monitoring of the prototype design of electric system by the ubidots platform. International Journal of Electrical and Computer Engineering, 11(6). https://doi.org/10.11591/ijece.v11i6.pp5568-5577

Ray, P. P. (2016). A survey of IoT cloud platforms. Future Computing and Informatics Journal, 1(1–2). https://doi.org/10.1016/j.fcij.2017.02.001

Zhang, D. (2013). Optimal Design and Planning of Energy Microgrids. Doctoral Thesis, UCL (University College London)., September.