

Design y development of a graphical user interface in LabVIEW for acquisition and visualization of climatological data (temperature and relative humidity)

Diseño y desarrollo de una interfaz gráfica de usuario en LabVIEW para la adquisición y visualización de datos climatológicos (temperatura y humedad relativa)

PEREZ-GARCIA, Víctor†*, QUINTANILLA-DOMINGUEZ, Joel, YAÑEZ-VARGAS, Israel and AGUILERA-GONZALEZ, José

Engineering in Networks and Telecommunications, Universidad Politécnica de Juventino Rosas, Hidalgo 102, Community of Valencia, Santa Cruz de Juventino Rosas, Guanajuato, Mexico.

ID 1st Author: Víctor, Perez-García / ORC ID: 0000-0003-3173-6793

ID 1st Coauthor: Joel, Quintanilla-Dominguez / ORC ID 0000-0003-2442-2032

ID 2nd Coauthor: Israel, Yañez-Vargas / ORC ID: 0000-0001-5749-8442

ID 3rd Coauthor: José, Aguilera-González / ORC ID: 0000-0002-4160-448X

DOI: 10.35429/JTP.2020.18.6.18.29

Received July 19, 2020; Accepted December 19, 2020

Abstract

This paper describes the design and development of a Graphical User Interface through the virtual instrumentation software NI LabVIEW using the VISA function, to graphically visualize and store the data of the climatological variables of temperature and relative humidity. The graphical interface offers the option to export the date, time and data of the two variables to text documents with extension “.txt”, which acquires the information of the electronic board wireless monitoring and control, which uses a main device PIC16F877A microcontroller. AMT1001 Precision Analog Sensor was used to sense temperature and relative humidity. The PIC16F877A was programmed using a C programming language in the CCS Compiler compiler, to the data acquisition, and send it via RS232 communication to the computer, using the PL2303 module USB to TTL converter. To check the GUI operation, the electronic wireless monitoring and control card was connected to the computer equipment by wire, however, the monitoring of the climate variables can be done wirelessly by XBEE technology. Future work aims to monitor the climate of a horticultural greenhouse with XBEE technology, so that the data is sent wirelessly to a computer that has the GUI, and is also connected to Ethernet or WIFI, which will have the LabVIEW graphical interface explained in this article, and the data will be displayed / analyzed through the internet.

LabVIEW, PIC16F877A, climatological variables

Resumen

El presente artículo describe el diseño y desarrollo de una Interfaz Gráfica de Usuario a través del software de instrumentación virtual NI LabVIEW mediante la función VISA, para visualizar gráficamente y almacenar los datos de las variables climatológicas de temperatura y la humedad relativa. La interfaz gráfica ofrece la opción de exportar la fecha, hora y los datos de las dos variables a documentos de texto con extensión “.txt”, la cual adquiere la información de una tarjeta electrónica de monitoreo y control inalámbrico, que utiliza como dispositivo principal un microcontrolador PIC16F877A en la adquisición de datos. Se utilizó el sensor analógico de precisión AMT1001 para sensar la temperatura y humedad relativa. El PIC16F877A se programó utilizando un lenguaje de programación en C en el compilador CCS Compiler, para realización de la adquisición de datos, y enviarlos por comunicación RS232 a la computadora, usando el módulo PL2303 convertidor USB a TTL. En trabajos a futuro se pretende monitorear el clima de un invernadero hortícola con tecnología XBEE, para que los datos sean enviados en forma inalámbrica a una computadora conectada a Ethernet o WIFI, la cual tendrá la interfaz gráfica de LabVIEW explicada en el presente artículo, y los datos sean mostrados/analizados a través de internet.

LabVIEW, PIC16F877A, variables climatológicas

Citation: PEREZ-GARCIA, Víctor, QUINTANILLA-DOMINGUEZ, Joel, YAÑEZ-VARGAS, Israel and AGUILERA-GONZALEZ, José. Design y development of a graphical user interface in LabVIEW for acquisition and visualization of climatological data (temperature and relative humidity). Journal of Technological Prototypes. 2020. 6-18:18-29.

* Correspondence to Author (Email: vperez_ptc@upjr.edu.mx)

† Researcher contributing first author

Introduction

Data acquisition is the process of measuring an electrical or physical phenomenon such as voltage, temperature, pressure, among others, with a computer. A Data Acquisition System (SAD) consists of sensors, measurement hardware, and a computer with programmable software (National Instruments, 2020). Figure 1 shows a block diagram of a data acquisition system.

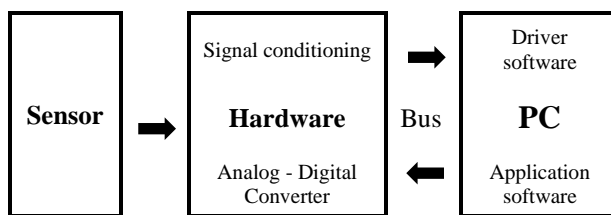


Figure 1 Block diagram of a SAD
Source: self-made

In a SAD the sensor converts a physical variable into an electrical signal that can be measured. The hardware acts as the interface between the computer and the physical variable measured by the sensor, and functions as a device that digitizes analog signals for a computer to interpret. A computer with programmable software is used to process, visualize and store data for the physical variables that have been measured (National Instruments, 2020).

Air temperature, relative humidity (RH), radiation and CO₂ concentration are physical variables that must be measured within a horticultural greenhouse, since they play an important role in product quality, profitability avoiding losses in cultivation and in the use of agrochemical inputs (Martínez & Roca, 2011).

Knowing the development of plant pests and diseases, diagnosing if a good biotic (with bumblebee) or abiotic (forced) pollination is taking place inside the greenhouse, are some of the reasons why horticultural producers monitor with a Graphical Interface User Interface (GUI by its English name, Graphical User Interface) that is interactive and friendly.

A GUI is a fundamental part of any application, to help the user to accomplish tasks quickly, easily and satisfactorily. When the user begins to work with the computer, it interacts with the Interface, either with that of the operating system, that of a particular software or that of any website (Albornoz, Berón, & Montejano, 2017).

LabVIEW is an application software, which facilitates the interaction between the computer and the user, to acquire, analyze and present data of physical variables that have been measured (National Instruments, 2020). Therefore LabVIEW can be used in a SAD and in the design of a custom GUI.

The purpose of this paper is to develop and design a GUI using the VISA functions of LabVIEW, to graphically display and store air temperature and RH data.

The article is organized as follows: The electronic card (hardware) and the analog sensor AMT1001, used in data acquisition, are described in the methodology section. Later in the same section it is mentioned how the data of the electronic card is sent to the port of the computer, for graphic programming in LabVIEW and the design of the GUI. Then, in the tests and results section, the operation of the GUI is presented. Finally, in the last section the conclusions and future works are presented.

Methodology

Electronic data acquisition card

Figure 2 shows a block diagram of the electronic wireless monitoring and control card that was used as hardware in data acquisition, to interact as an interface between the sensor and the computer.

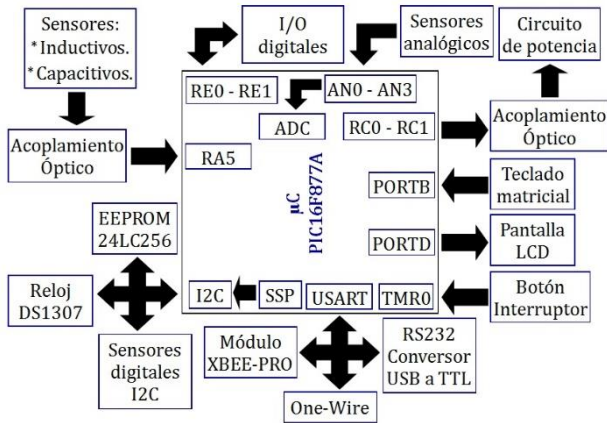


Figure 2 Block diagram of the electronic wireless monitoring and control card
Source: self-made

The electronic wireless monitoring and control card has as its main device a PIC16F877A Microcontroller from Microchip, and female header connectors to connect four analog sensors, an inductive or capacitive digital sensor with 12 Volt output and digital sensors with I2C communication protocol. and One-Wire.

The data obtained during the sampling can be sent to the computer in two ways: wired, using a USB to TTL converter module that connects to two female header connectors on the electronic card, or wirelessly, with an XBEE module integrated in the electronic card. Figure 3 shows an electronic connection diagram of the header connectors and XBEE module to the PIC16F877A.

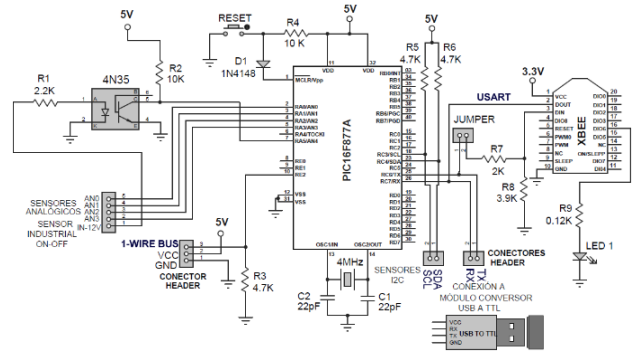


Figure 3 Electronic connection diagram of the XBEE header and module connectors to the PIC16F877A
Source: own elaboration, electronic diagram obtained with the PROTEUS ISIS simulator

Figure 4 shows the physical form of the electronic wireless monitoring and control card, which was used as hardware in data acquisition.

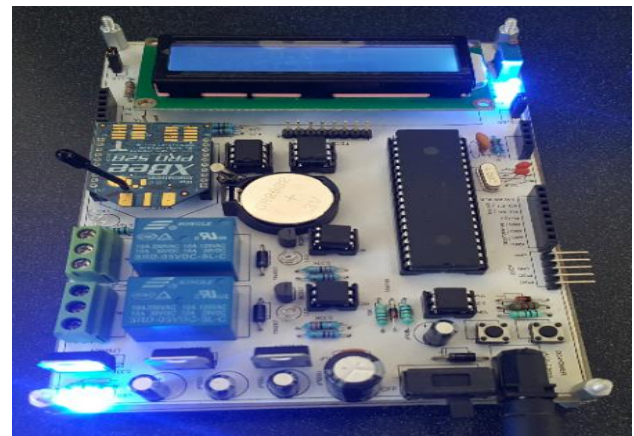


Figure 4 Physical form of the electronic card for wireless monitoring and control
Source: self-made

Temperature and relative humidity measurement

The AMT1001 sensor was used to measure the weather temperature and RH. Figure 5 shows and describes the AMT1001 sensor terminals.

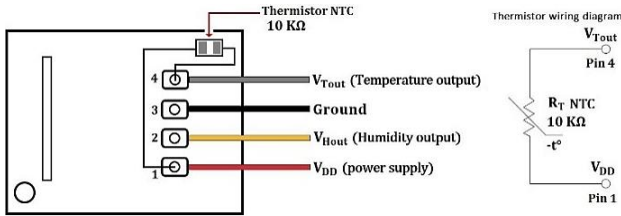


Figure 5 AMT1001 Sensor Terminals Description
Source: self-made

The output voltage V_{Hout} is a linear voltage in the range of 0 to 3 Volts (Temperature and Humidity Module AM1001 / AMT1001 Product Manual, 2020), which was measured with analog channel AN1 of the PIC16F877A Digital Analog Converter (DAC). The relationship between HR and V_{Hout} is shown in equation 1.

$$HR = \frac{V_{Hout}}{0.03} \quad (1)$$

To measure temperature, the AMT1001 sensor has a 10KΩ NTC negative coefficient thermistor, between terminals 1 and 4. Table 1 shows some parameters of the NTC thermistor.

R_o reference temperature resistance (298.15 K) (25 °C)	β Thermistor constant	Temperature operating range (°C)
10KΩ	3435	-40~125

Table 1 Some parameters of AMT1001 sensor NTC thermistor
Source: self-made

A precision Trimpot potentiometer set to 10 KΩ was connected between terminal 4 and ground, to obtain the output voltage V_{Tout} . The V_{Tout} voltage was measured with the analog channel AN0 of the DAC of the PIC16F877A. Figure 6 shows an electrical connection diagram between the NTC thermistor and the Trimpot.

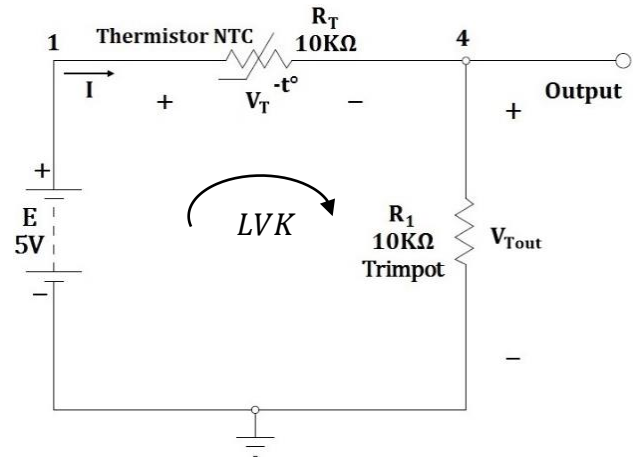


Figure 6 Electrical connection diagram between the NTC thermistor of the AMT1001 sensor and the 10KΩ Trimpot
Source: own elaboration, electronic diagram obtained with the PROTEUS ISIS simulator

A circuit analysis was performed to the electrical diagram of Figure 6, to obtain the relationship between the temperature in degrees centigrade T (°C) and the volatility V_{Tout} . Applying Kirchhoff's Voltage Law (KVL) clockwise to the closed path of the electrical diagram in Figure 6, equation 2 was obtained.

$$-E + V_T + V_{Tout} = 0 \quad (2)$$

By applying Ohm's law in the NTC thermistor to obtain V_T , and by substituting in (2) it was possible to obtain equation 3.

$$-E + IR_T + V_{Tout} = 0 \quad (3)$$

Using Ohm's law in R_1 , we obtain equation 4 that calculates the electric current I .

$$I = \frac{V_{Tout}}{R_1} \quad (4)$$

If we substitute (4) in (3) and solve for R_T , we obtain equation 5.

$$R_T = \frac{ER_1}{V_{Tout}} - R_1 \quad (5)$$

Equation 6 relates some NTC thermistor parameters to R_T .

$$R_T = R_0 e^{\beta\left(\frac{1}{T_K} - \frac{1}{T_0}\right)} \quad (6)$$

Where:

- T_0 , is the reference temperature 298.15 expressed in degrees Kelvin (25 °C).
- R_0 , is the resistance of the thermistor at the reference temperature.
- β , is a constant of the NTC thermistor.
- T_K , is the temperature expressed in degrees Kelvin.

Solving for T_K from (6) and applying the properties of logarithms, equation 7 was obtained.

$$T_K = \frac{1}{\frac{1}{\beta} \ln \frac{R_T}{R_0} + \frac{1}{T_0}} \quad (7)$$

To convert from degrees Kelvin to degrees centigrade, equation 8 is used.

$$T_{\circ C} = T_K - 273.15 \quad (8)$$

If (7) is substituted in (8) we obtain equation 9 that is used to calculate the temperature in degrees centigrade.

$$T_{\circ C} = \frac{1}{\frac{1}{\beta} \ln \frac{R_T}{R_0} + \frac{1}{T_0}} - 273.15 \quad (9)$$

Equations (1) and (9) were used to calculate RH and temperature in degrees centigrade respectively. Within the CCS Compiler, these equations were implemented in the C language program code for the PIC16F877A.

Using the description of the AMT1001 sensor terminals in Figure 5, and the electrical diagram in Figure 6, a single-sided Printed Circuit Board (PCB) (Bottom Copper) was made with the ARES tool from PROTEUS. Figure 7 shows the PCB made for the AMT1001 sensor.

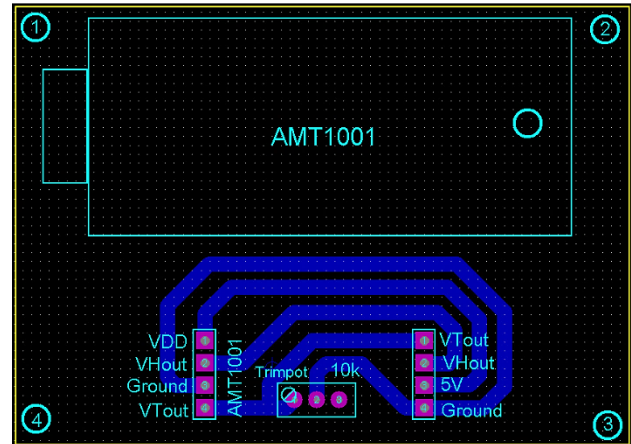


Figure 7 Bottom Copper face of the PCB in ARES
Source: own elaboration, PCB design obtained with the PROTEUS ARES tool

Figure 8 shows in physical form the top view of the PCB made for the AMT1001 sensor.

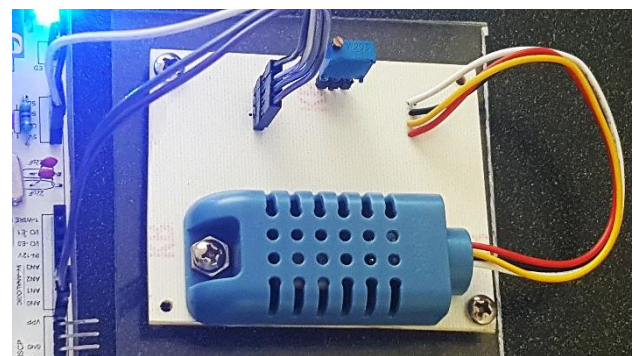


Figure 8 Top view of the PCB made for the AMT1001 sensor
Source: self-made

Sending the climatological data to the computer by RS232 communication

The PL2303 USB to TTL serial converter module from Prolific was used to communicate the PIC16F877A of the wireless monitoring and control electronic card with the NI LabVIEW virtual instrumentation software, using an asynchronous RS232 serial communication protocol.

In the serial transmission of data to the computer, the PIC16F877A sends the temperature or RH in a seven-byte ASCII character string format. Figure 9 shows the format used to send and identify the value of the climatological variables.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
I	E1	E2	.	D1	D2	LF

Figure 9 Format of ASCII characters that are sent to the computer port
Source: self-made

The first ASCII character sent is Byte 0 (I) with the value of '0' or '1', which indicates whether it is the temperature or HR. The characters from Byte 1 to Byte 5 is the reading made by the PIC16F877A of the AMT1001 sensor, in two whole digits and two digits after the decimal point. Finally, Byte 6 (LF) is the newline character (10 in ASCII code) that is sent to the end of the string. An example of the ASCII character format is shown in Figure 10, used to send the HR value to the LabVIEW software.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
0	6	5	.	3	0	LF

Figure 10 Example of the format for sending the HR value to the LabVIEW software
Source: self-made

The CCS Compiler has functions to send data through the USART module (Henao & Duque, 2009). Figure 11 shows the functions used in the C language program code to send the HR value to LabVIEW.

```
printf("0"); //Se envia '0'
printf("%0.2f",HR); //Se envía el valor de HR
putc(10); //Se envía carácter de nueva línea, fin del dato
```

Figure 11 Functions used in CCS Compiler to send HR ASCII character format to LabVIEW
Source: self-made

The function *printf("0")* writes and sends the character '0', indicating that the characters that will be sent after this character correspond to the climatological variable HR. The *printf("%0.2f", HR)* function sends the value of the climate variable HR, with two whole digits and two after the decimal point. The *putc(10)* function writes the newline character 'LF' in ASCII code.

An example of the character format is shown in Figure 12, used to send the temperature value to the computer port.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
1	2	5	.	2	0	LF

Figure 12 Example of the format for sending the temperature value to LabVIEW
Source: self-made

Figure 13 shows the functions used in the C language program code to send the temperature value to LabVIEW.

```
printf("1"); //Se envia '1'
printf("%0.2f",Tc); //Se envía el valor de Temperatura
putc(10); //Se envía carácter de nueva línea, fin del dato
```

Figure 13 Functions used in CCS Compiler to send the ASCII temperature character format to LabVIEW
Source: self-made

Graphical Programming in LabVIEW for the Graphical User Interface (GUI)

A virtual instrument (VI) was developed in NI LabVIEW using the NI VISA functions, based on the methodology of (Flores Flores, 2014), to establish RS232 serial communication with the wireless monitoring and control card, read the data that arrives to the serial communication port, visualize, graph and export the data of the weather variables. Figure 14 shows the program code in graphic language (block diagram) of the VI developed for the GUI.

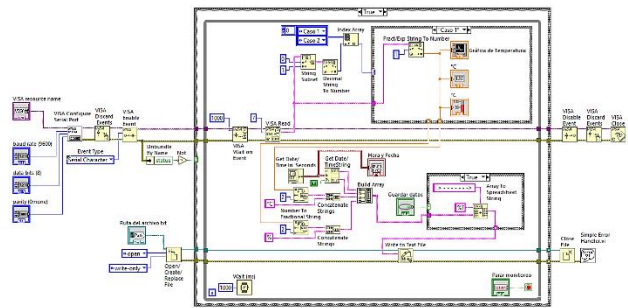


Figure 14 Graphical language program code developed in LabVIEW for the GUI.

Source: own elaboration, block diagram obtained from NI LabVIEW

Figure 15 shows the blocks used to configure and enable asynchronous serial communication and the computer port.

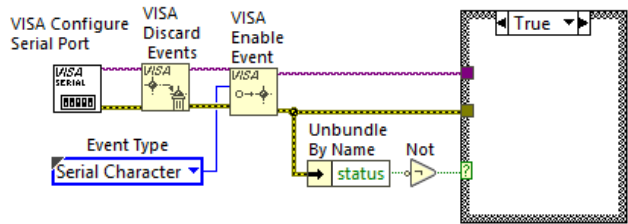


Figure 15 Blocks to configure and enable serial communication and computer port.

Source: own elaboration, obtained from the LabVIEW block diagram

The VISA Configure Serial Port block configures the speed parameters (bits per second), data bits, and parity. The VISA Discard Events block clears the characters that were left in the port from previous events. The VISA Enable Event function enables the serial port to receive a character or byte and place it on hold. Since the wireless control and monitoring card sends ASCII characters to LabVIEW, the Serial Character option is selected in VISA Enable Event. The first case (True) of the Case Structure control structure is used to process the characters that arrive through the port, and the second case (False) is used to rule out errors during serial communication. The Unbundle By Name element was connected with the not operator, to convert the serial port error status to Boolean values.

The VISA Wait on Event and VISA read functions are used to wait and receive respectively an ASCII character (Figure 16).

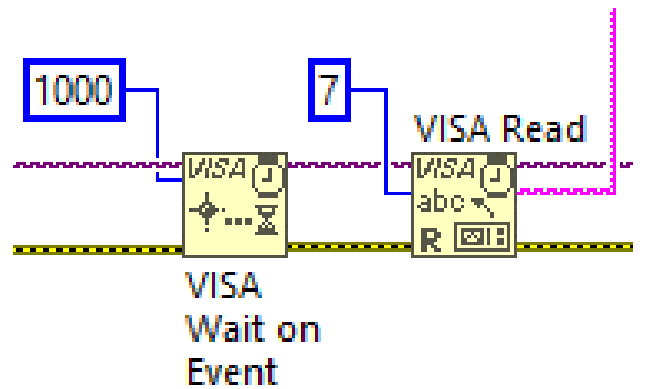


Figure 16 Functions to wait and receive an ASCII character

Source: own elaboration, obtained from the LabVIEW block diagram

The VISA Wait on Event function waits for an ASCII character before 1000 ms, if this does not happen, an error is generated by VISA Discard Events and VISA Enable Event. The VISA read function receives the format of the seven bytes of the temperature or HR before reading them. The VISA read byte count parameter was assigned the value of 7, to indicate the bytes to expect.

Figure 17 shows the functions that were used to convert the seven-byte ASCII character format to a floating point number, and identify whether it is the temperature or HR value.

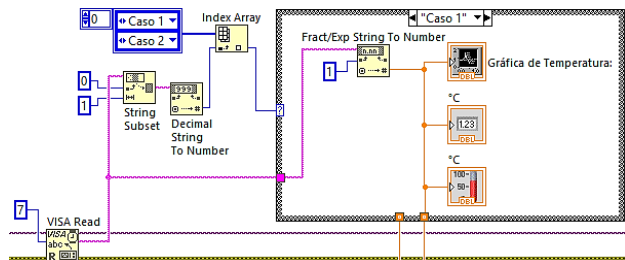


Figure 17 Functions to convert a character string to a floating point number and identify whether it is temperature or HR
Source: own elaboration, obtained from the LabVIEW block diagram

The String Subset block separates byte 0 of the string from the seven bytes that VISA read delivers in the read buffer, and sends it to its output as a substring. The Decimal String To Number block converts the substring (byte 0) to an integer decimal number. To identify if the character string corresponds to temperature or RH, a Case Structure case control structure (case 1 and case 2) was used. The assignment of cases is carried out through the Index Array, which in its row 0 has case 1 and in its row 1 has case 2. The rows of the array must match the output of the Decimal String To Number ('0' or 'one'). The Fract / Exp String To Number block separates byte 0 from the character string that VISA Read sends, and converts the rest of the string to a floating point number.

In case 1 the numerical indicators and graphs of the temperature were placed, and in case 2 the numerical indicators and graphs of HR.

The functions used to export a Table with the date, time and the data of the two climate variables to a text document ".txt", are shown in Figure 18.

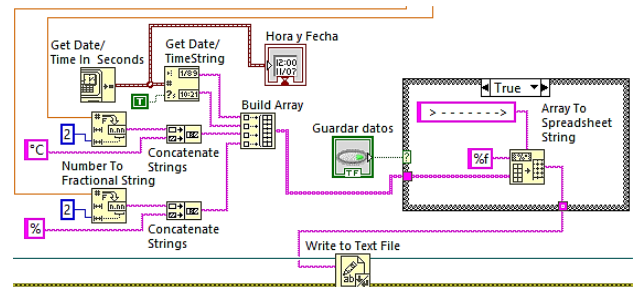


Figure 18 Functions to export a Table with the date, time and the measured variables of the climate to a text document ".txt"
Source: own elaboration, obtained from the LabVIEW block diagram

The functions Get Date / Time in Second and Get Date / Time String obtain the date and time from the computer clock in two strings, one for the date and the other for the time. The Number To Fractional String functions were used to convert temperature and RH to character strings. The Concatenate Strings function was used to concatenate the symbol “°C” with the temperature character string, in the same way, another similar function was used to concatenate the symbol “%” with the HR character string. With the Build Array function, the date, time, temperature and RH strings were concatenated. Within a Case Structure case control structure, the Array To Spreadsheet String function was used to generate a Table with the character strings concatenated by the Build Array function. The case control structure is conditioned by a logical button, when True the Table is generated with the characters. Finally, the function Write Text File prints the Table in the text file ".txt".

Design of the Graphical User Interface

In the front panel of LabVIEW the GUI was designed mainly to graphically display and store the data of the climatological variables of temperature and RH. Figure 19 shows the GUI designed on the LabVIEW front panel.

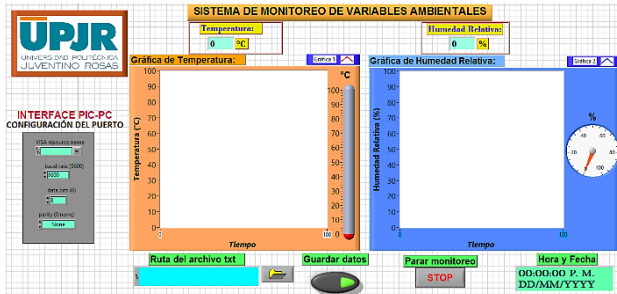


Figure 19 GUI developed on the front panel of LabVIEW

Source: own elaboration, obtained from the LabVIEW control panel

In the port configuration window, select the computer port where the device is connected, the communication speed, the data bits, and the parity. The device connected to the computer port can be any electronic development board that handles the USART asynchronous serial communication protocol.

To visualize the data of the variables as they arrive at the computer port, the GUI has two graphic windows, one to graph temperature and another to graph RH. At the top of each graph, there are numerical indicators where the value of each variable is displayed in a number. A thermometer as a graphical indicator is displayed to the right of the temperature graph, and a hygrometer is displayed to the right of the RH graph.

In the lower left part of the GUI there is an indicator with the time and date. These data are obtained from the computer clock, and it is very important that they are updated, since these will be part of the historical data in the measurements of the climatological variables.

The GUI offers the option of storing the date, time and the data of the weather variables in a text document with the extension ".txt". The user must create the text document, give it a name and save it in some path on the computer's hard drive. In the Path of the txt file window, you must select the path of the text document where the data of the variables will be saved. At the bottom is the Save data button, when pressed it shows an appearance in orange, indicating that the data of the variables will begin to be stored in the text document.

Tests and results

To test the GUI operation, the wireless monitoring and control electronic card was connected to the computer by wire, using the PL2303 USB to TTL converter module (Figure 20).

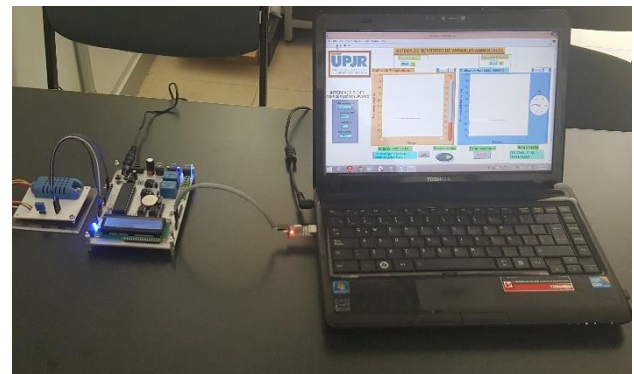


Figure 20 How to connect the wireless monitoring and control electronic card to the computer

Source: self-made

In the computer equipment where the GUI operation was tested, the PL2303 driver for the USB to TTL converter module was installed, and the NI-VISA driver that allows communication between NI LabVIEW and a device connected to the port. For data acquisition (measurement and reading), the AMT1001 sensor was connected to the electronic wireless monitoring and control card. Figure 21 shows the sensor and electronic card connected in a test performed for the GUI.

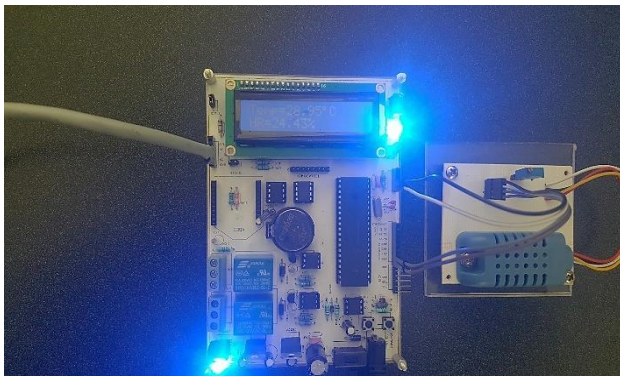


Figure 21 Connection of the AMT1001 sensor and electronic card for data acquisition
Source: self-made

In the design and development of the GUI several performance tests were performed. Figure 22 shows how the GUI worked on the LabVIEW front panel, for a test run performed.

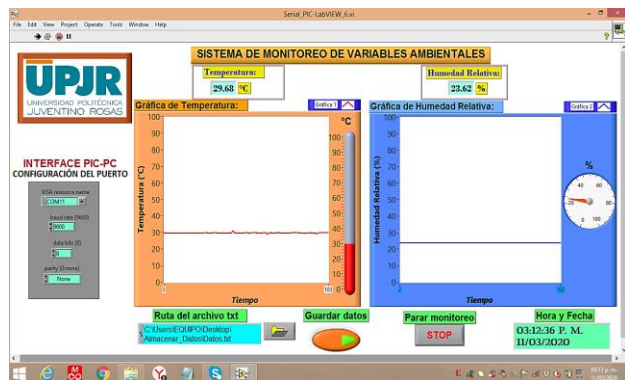


Figure 22 GUI Operation on the LabVIEW Front Panel
Source: own elaboration, obtained from the LabVIEW control panel

Before running the test shown in Figure 22, the document path named Data.txt was selected. The path of this document was a folder named Store_Data (Figure 23).

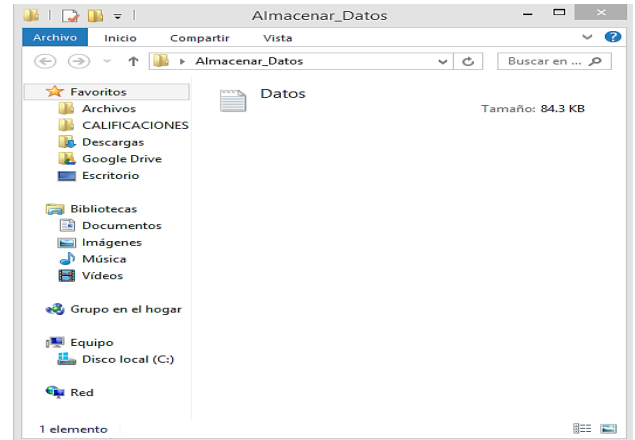


Figure 23 Folder where the document Data.txt was saved
Source: self-made

During the test run, the Save Data button was pressed, only to store some temperature and RH values in the Data.tx document. Figure 24 shows a printed table with the date, time and data of the two climate variables in the document Datos.txt.

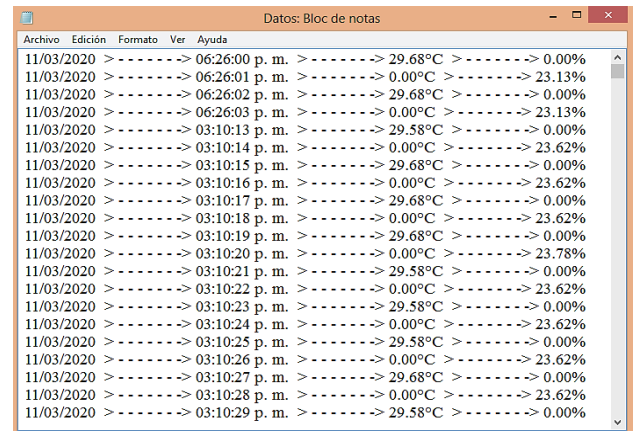


Figure 24 Date, time and temperature and HR data in the document Datos.txt.
Source: self-made

Conclusions and future work

The purpose of the work was to develop and design a GUI using the LabVIEW VISA functions, to graphically display and store air temperature and HR data. In the functional tests, the GUI worked properly. In future work it is intended to add control buttons to the GUI, to control and activate electric actuators.

Figure 14 shows that the Operation parameter of the Open / Create / Replace File block is set to “open”, this causes the data to be overwritten in the text document each time the Save data button is pressed. To generate a new text document each time the Save data button is pressed and the data is not overwritten, the open or create option must be chosen in the Operation parameter.

The GUI stores the temperature and RH data in a single “.txt” text document. If you want to save a variable per text document, in the LabVIEW graphical programming, you will have to add an Open / Create / Replace File, Write to Text File and Close File block for each variable.

The program code for the PIC16F877A and graphical programming in LabVIEW were developed to work with temperatures from 0 to 99 ° C. To perform tests below 0 ° C both codes would have to be modified. In the PIC16F877A an 8-byte character string must be sent, where one byte must be allocated to assign it an ASCII character, indicating whether it is a negative (less than 0 ° C) or positive (greater than 0 ° C) temperature.). In the LabVIEW graphical programming, more blocks must be added, cases in the control structures, and some parameters of the functions must be modified.

The GUI was designed to graphically display and store data for two climatological variables. If you want to add another variable, such as Carbon Dioxide $[[CO]]_2$, the Microcontroller will have to send three strings of ASCII characters instead of two. In the LabVIEW block diagram, more blocks and cases should be added to the control structures.

In this work, the analog sensor AMT1001 was used, however, with the electronic wireless monitoring and control card, digital sensors with I2C and One Wire communication protocol can be used.

To check the operation of the GUI, the electronic wireless monitoring and control card was connected to the computer equipment by wire, however, the monitoring of the climate variables can be done wirelessly by XBEE technology. Future work is intended to monitor the climate of a horticultural greenhouse with XBEE technology, so that the data is sent wirelessly to a computer that has the GUI and is also connected to Ethernet or WIFI.

References

- Albornoz, M., Berón , M., & Montejano, G. (2017). Interfaz gráfica de usuario: el usuario como protagonista del diseño. *XIX Workshop de Investigadores en Ciencias de la Computación (WICC 2017, ITBA, Buenos Aires)*, (págs. 570-574). Buenos Aires.
- Flores Flores, E. (2014). Sistema inalámbrico de medición, control y monitoreo de flujo de líquidos no viscosos. *Universidad Autónoma de México*. México D.F.
- Henao, C., & Duque, E. (2009). PROGRAMANDO MICROCONTROLADORES PIC EN LENGUAJE C . *Scientia et Technica*, 37-42.

Martínez, P., & Roca, D. (2011). El control del clima de los invernaderos de plástico. Un enfoque actualizado. *ReDivia*, 181-247.

National Instruments. (Marzo de 2020). Obtenido de ¿Qué es Adquisición de Datos?: [http://www.ni.com/data-acquisition/what-is/esa/Temperature and Humidity Module AM1001/AMT1001 Product Manual](http://www.ni.com/data-acquisition/what-is/esa/Temperature%20and%20Humidity%20Module%20AM1001/AMT1001%20Product%20Manual). (Marzo de 2020). Obtenido de Temperature and Humidity Module AM1001/AMT1001 Product Manual: [https://datasheetspdf.com/pdf/1298922/AOSO NG/AM1001/1](https://datasheetspdf.com/pdf/1298922/AOSO/NG/AM1001/1)