Article ECORFAN Journal-Democratic Republic of Congo
June 2022 Vol.8 No.14 19-30

Simulation of a point-to-point wireless communication with XBEE technology for monitoring environmental variables

Simulación de una comunicación inalámbrica punto a punto con tecnología XBEE para el monitoreo de variables ambientales

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

Universidad Politécnica de Juventino Rosas, Ingeniería en Sistemas Automotrices e Ingeniería en Redes y Telecomunicaciones, Hidalgo 102, Comunidad de Valencia, Santa Cruz de Juventino Rosas, Guanajuato, México.

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

ID 1st Co-author: *Joel, Quintanilla-Domingez* / **ORC ID** 0000-0003-2442-2032

ID 2nd Co-author: *Israel, Yañez-Vargas* / **ORC ID:** 0000-0001-5749-8442

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

DOI: 10.35429/EJDRC.2022.14.8.19.30 Received March 10, 2022; Accepted June 30, 2022

Abstract

This article describes the simulation of a unidirectional point-to-point radio frequency (RF) Wireless communication between two electronic circuits with XBEE technology, using the electronic design software PROTEUS V8.8. In the same worksheet in the ISIS graphical interface, the two electronic circuits were implemented, one was called the *transmitter circuit* and the other the *receiver circuit*. A Microchip® PIC16F877A microcontroller and an XBEE module are the main devices of each of the electronic circuits, to establish wireless communication by RF. The PIC microcontroller of the *transmitter circuit* was used as a DAQ data acquisition device, to measure the temperature of two analog sensors, the LM35 and the TMP36, and three digital sensors DS1621. The temperature data was sent from the *transmitter circuit* to the *receiver circuit* in character strings, using asynchronous serial communication. A programming algorithm was implemented for the PIC microcontroller of the *receiver circuit*, to receive and identify the temperature data. The development of this simulation is the basis to implement a wireless sensor network (WSN) with XBEE technology in the future.

XBEE, Acquisition, Asynchronous

Resumen

En este artículo se describe la simulación de una comunicación inalámbrica por radio frecuencia (RF) punto a punto unidireccional, entre dos circuitos electrónicos con tecnología XBEE, utilizando el software de diseño electrónico PROTEUS V8.8. En una misma hoja de trabajo en la interfaz gráfica de ISIS, se implementaron los dos circuitos electrónicos, a uno se le llamo *circuito transmisor* y al otro *circuito receptor*. Un microcontrolador PIC16F877A de Microchip® y un módulo XBEE, son los dispositivos principales de cada uno de los circuitos electrónicos, para establecer la comunicación inalámbrica por RF. El microcontrolador PIC del *circuito transmisor* se utilizó como dispositivo de adquisición de datos DAQ, para medir la temperatura de dos sensores analógicos, el LM35 y el TMP36, y tres sensores digitales DS1621. Los datos de temperatura fueron enviados del *circuito transmisor* hacia el *circuito receptor* en cadenas de caracteres, usando la comunicación serial asíncrona. Se implementó un algoritmo de programación para el microcontrolador PIC del *circuito receptor*, para recibir e identificar los datos de temperatura. El desarrollo de esta simulación es la base para implementar a futuro, una red de sensores inalámbricos (WSN) con tecnología XBEE.

XBEE, Adquisición, Asíncrono

Citation: PEREZ-GARCIA, Víctor, YAÑEZ-VARGAS, Israel, QUINTANILLA-DOMINGUEZ, Joel and AGUILERA-GONZALEZ, José. Simulation of a point-to-point wireless communication with XBEE technology for monitoring environmental variables. ECORFAN Journal-Democratic Republic of Congo. 2022, 8-14: 19.30

† Researcher contributing first author.

Introduction

A greenhouse is a closed enclosure that allows an optimal climate to be generated, to cultivate and protect crops from climatic variations. The development of crops, in their different growth phases, is conditioned by four climatic factors: temperature, relative humidity, lighting and carbon dioxide (CO_2) (Villalobos et al., 2017). There is a wide variety of crops grown under glass, and in most of them the objective is to increase size and weight. To reach this goal, each crop needs to fulfil its life process with the four climatic factors at the most favourable values. The life process of a crop is characterised by the functions of photosynthesis, respiration, growth and transpiration (absorption of water and nutrients) (García-García & Martínez-Tornero, 2016).

Plants must transpire water to transport nutrients and regulate their growth. A low relative humidity level decreases plant growth because the plant stops transpiring by closing its stomata (INTAGRI, n.d.). This is an example of the relationship between the relative humidity factor and the transpiration function of plants in greenhouse crops. Table 1 shows an example of the effect of temperature and CO_2on photosynthesis function.

Effect of photosynthesis	Temperature level (°C)	CO_2 level (ppm)
Photosynthesis stop	Under 5	Under 300 or 400
Optimum level	From 20 to 25	From 700 to 900
Photosynthesis stop	Over 45	Over 1000 to 1100

Table 1 Effect of ambient temperature and CO_2 on photosynthesis in greenhouse crops
Source: Own elaboration, based on information from (García-García & Martínez-Tornero, 2016)

Continuous monitoring of climatic factors within a greenhouse with sensors, real-time information, and the computer capacity to store data, are the characteristics used to obtain quality information, make an objective diagnosis, and make an accurate decision in the agronomic management of the crop (AGTECH AMERICA, 2020). The monitoring of the four climatic factors within a greenhouse is important, as it provides information to make an accurate decision in the application of agrochemicals.

Figure 1 shows the stages in the process of making an accurate decision and anticipating the appearance of problems in the vegetative development of the crop.

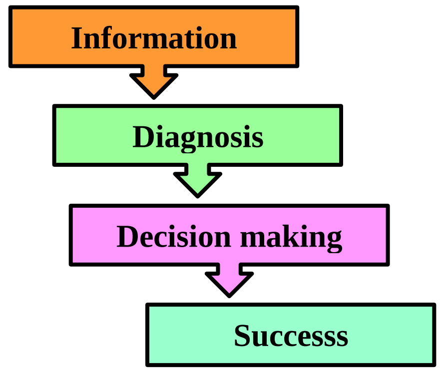

Figure 1 Stages in the process of making a good decision, and anticipating problems in the vegetative development of the crop
Source: Own Elaboration, adapted from (AGTECH AMERICA, 2020)

Wireless technology is a proposal to be implemented in the monitoring of climate variables within a greenhouse, because it provides flexibility in the installation of sensors, robustness in the network, and reduces the cost and complexity of its maintenance (Cama-Pinto et al., 2014). The XBEE radio frequency (RF) module is a small radio device that uses the IEEE 802.15.4 network protocol to create point-to-point and point-to-multipoint communication networks. These modules communicate wirelessly, and can send data from a sensor (DIGI, 2022).

Simulating an electronic circuit consists of emulating it as if it were real, and thus having a middle ground between theoretical concepts and reality. The advantages of simulating an electronic circuit are: reducing the design time without assembling it in a physical system, making the necessary modifications in less time, and anticipating problems in the behaviour of the system and unforeseen results (INDIELEC, 2021). Before physically implementing the communication between wireless devices, in order to test their behaviour between them, it is recommended to use simulation in a software simulator.

PROTEUS is a software simulator that allows designing, testing, and debugging electronic circuits before implementing a physical prototype.

ISSN 2414-4924

This software has a working interface called ISIS, which allows writing and applying firmware to a microcontroller, and has the ability to provide the necessary graphics for the animation of electrical and electronic circuits (Labcenter, 2022). The PROTEUS software has a library to simulate the XBEE module via serial communication. This module has two terminals enabled, the transmission (TX) and reception (RX), which allow communication quite easily with Microchip® microcontrollers, Arduino® modules, among other microcontrollers (Morán-Inga, 2019).

Serial Ports Emulator (VSPE) is a virtual port emulator software that can be used in PROTEUS to simulate wireless communication between different devices (Morán-Inga, 2019). VSPE was designed to help software engineers and developers create, debug and test applications that use serial ports. This software is capable of creating multiple virtual devices to transmit and receive data (Eterlogic, 2022).

This paper describes the simulation of a unidirectional point-to-point RF wireless communication between two electronic circuits with XBEE technology, using PROTEUS V8.8 electronic design software. In the same worksheet in the ISIS graphical interface, the two electronic circuits were implemented, one being called the transmitter circuit and the other the receiver circuit. A PIC16F877A microcontroller from Microchip® and an XBEE module are the main devices in each of the electronic circuits to establish wireless RF communication.

The article is organised as follows: in the methodology, four stages are described; in stage 1, the temperature measurement of two analogue sensors was performed with the PIC microcontroller of the transmitter circuit; in stage 2, in a new ISIS GUI worksheet, the temperature measurement of three digital sensors was simulated with the PIC microcontroller of the transmitter circuit; in stage 3, the temperature measurement of the analogue and digital sensors was performed together with the PIC microcontroller of the transmitter circuit; in stage 3, the temperature measurement of the analogue and digital sensors was performed together with the PIC microcontroller of the transmitter circuit.

In stage 4, a unidirectional point-to-point wired communication between the transmitter circuit and the receiver circuit was executed, carrying out in this process the necessary debugging to the program codes of the PIC16F877A microcontrollers, to send, convert, receive, and display the data of the analogue and digital temperature sensors. The tests and results section describes the implementation of stage 5, which consisted of using the XBEE RF modules, and verifying the unidirectional point-to-point communication between the *transmitter circuit* and the *receiver circuit*. Finally, the last section presents the conclusions and future work, mentioning what needs to be improved and implemented in the future.

Methodology

The ISIS graphical environment of the PROTEUS V8.8 electronic design automation software was used to simulate unidirectional point-to-point wireless communication with XBEE technology, using the UART module of the two Microchip® PIC16F877A microcontrollers. According to Castaño-Giraldo (2022), asynchronous serial communication or UART is one of the simplest communications used by PIC microcontrollers, and only uses three terminals: one to transmit TX data, another to receive RX data and the common GND line.

The simulation was carried out in stages, with the aim of quickly identifying and correcting syntax and semantic errors in the structure of the C language program codes of the two microcontrollers. Simulating in stages in PROTEUS helped to find connection errors between components as they were added to the ISIS graphical environment. If, when running the simulation, the electronic circuit did not behave as expected, a direct approach was given to correct the connection errors to the newly added components.

Stage 1: Temperature measurement with analogue sensors

We started by simulating the temperature measurement of two analogue sensors with linear output, the LM35 and the TMP36, using the PIC16F877A microcontroller as the data acquisition device (DAQ).

Figure 2 shows the connection of the analogue sensors to the PIC microcontroller, and a temperature value of the TMP36 sensor displayed on the LCD (Liquid Crystal Display).

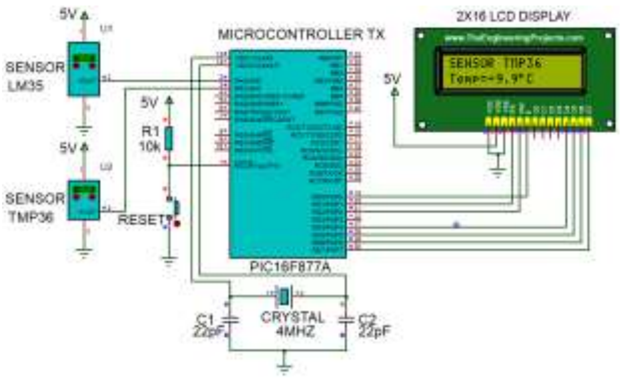

Figure 2 Temperature measurement using LM35 and TMP36 analogue sensors and a PIC16F877A microcontroller
Source: Own Elaboration, electronic diagram obtained with PROTEUS ISIS environment

The connections of the input and output devices to the PIC microcontroller were as follows: the LM35 sensor was connected to the RA0/AN0 analogue terminal, the TMP36 sensor was connected to the RA1/AN1 analogue terminal, and to display the temperature values of the analogue sensors, an LCD display was connected to the C port terminals.

C programming language was used to write the program code for the PIC microcontroller. The program code was edited and compiled in the CCS Compiler, according to Garcia-Breijo (2008) this compiler was developed specifically for Microchip microcontrollers.

Stage 2: Temperature measurement with digital sensors

In a new graphical work environment in ISIS, the acquisition of temperature data from three DS1621 digital sensors with I2C (Inter-Integrated Circuit) serial communication protocol was carried out, using a PIC16F877A microcontroller as DAQ device. Figure 3 shows the connection of the three DS1621 digital sensors to the PIC16F877A microcontroller, and a temperature value displayed on the LCD screen.

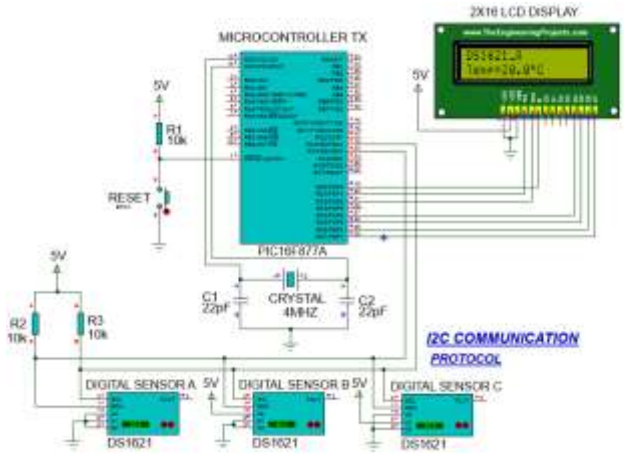

Figure 3 Temperature measurement using three DS1621 digital sensors, and a PIC16F877A microcontroller
Source: Own Elaboration, electronic diagram obtained with PROTEUS ISIS environment

In the I2C serial communication, the three DS1621 digital sensors were used as slaves, while the PIC16F877A microcontroller was used as the master. Table 2 shows the addresses of the digital sensors in slave mode, used to communicate the DS1621 digital sensors with the PIC microcontroller.

Digital Sensor DS1621	Address in hexadecimal format
Sensor A	$0 \times 00h$
Sensor B	$0 \times 01h$
Sensor C	$0 \times 02h$

Table 2 Hexadecimal addresses used to communicate the digital sensors with the PIC16F877A microcontroller
Source: Own Elaboration

The PIC microcontroller terminals used in the I2C serial communication were RC3/SCL and RC4/SDA. With the idea of connecting the LM35 and TMP36 analogue sensors and the three DS1621 digital sensors to the same microcontroller at a later stage, an LCD display was connected to port C of the PIC microcontroller as shown in figure 3.

Stage 3: Temperature measurement with the analogue and digital sensors in one electronic circuit

In this stage, the analogue and digital sensors used in the electronic circuits of figures 2 and 3 were connected to a single PIC microcontroller. In another new graphical work environment in ISIS, the electronic circuit in figure 4 was simulated.

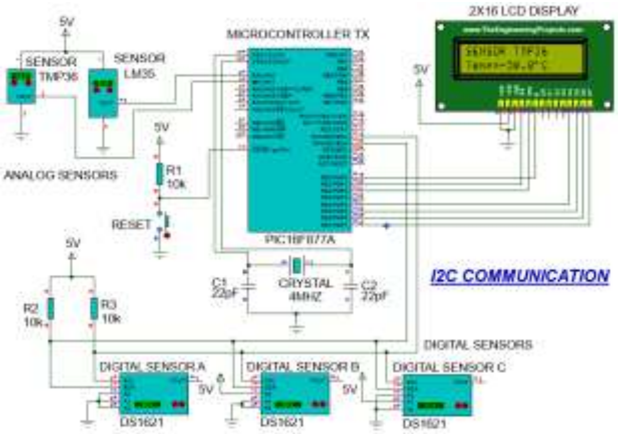

Figure 4 Temperature measurement with DS1621 digital sensors, LM35 and TMP36 analogue sensors, and the PIC16F877A microcontroller.
Source: Own Elaboration, electronic diagram obtained with the PROTEUS ISIS graphical environment

Figure 5 shows the variables used in the C language program code of the PIC microcontroller to store the temperature values of the analogue and digital sensors.

```
char string[6];
float Temperature1, Teemperature2;
float Temperature3, Temperature4;
float Temperature5;
```

Figure 5 Floating type variables used in the CCS Compiler to store temperature values obtained from analogue and digital sensors
Source. Own Elaboration

The variables Temperature1 and Temperature2 were used to store the temperature values in degrees Celsius (°C) of the analogue sensors. The Temperature1 variable was used to store the temperature values of the LM35 and Temperature2 for the TMP36. The variables Temperature3, Temperature4 and Temperature5 were used to store the temperature values in degrees Celsius (°C) of the DS1621 digital sensors. The variable Temperature3 was used to store the temperature values for digital sensor A, Temperature4 for digital sensor B and finally Temperature5 for digital sensor C.

Stage 4: Unidirectional point-to-point wired communication

Before simulating wireless communication with the XBEE modules in PROTEUS, tests were performed to send point-to-point data in one direction only in a wired fashion.

The electronic circuit in figure 4 was used as the transmitter circuit and another electronic circuit with a PIC16F877A microcontroller and an LCD display was used as the receiver circuit. In figure 6, a wire is connected from the RC6/TX terminal of the transmitter circuit's PIC microcontroller to the RC7/RX terminal of the receiver circuit's PIC microcontroller to perform the wired communication.

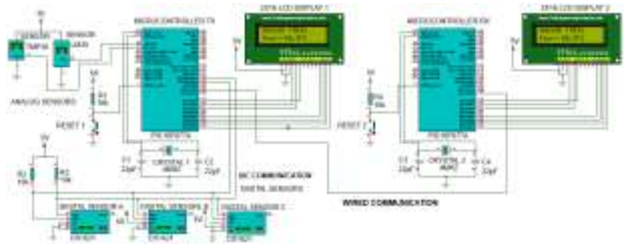

Figure 6 Unidirectional point-to-point wired communication between two electronic circuits with PIC16F877A microcontrollers
Source: Own Elaboration, electronic diagram obtained with PROTEUS ISIS environment

On the LCD displays connected to the PIC microcontrollers, the temperature value of the analogue sensor TMP36 is displayed. In this communication, the PIC microcontroller in the transmitter circuit sends the temperature data via asynchronous serial communication and displays it on the LCD, and the PIC microcontroller in the receiver circuit receives the temperature data and displays it on the LCD.

Focusing exclusively on debugging the program codes of the microcontrollers, so that they communicate properly with each other, was the reason for simulating the wired communication before simulating the wireless communication with the XBEE modules in PROTEUS.

In the wired simulation, before sending the temperature data from the transmitter circuit to the receiver circuit, each temperature data in floating format (number with decimal point) was converted to a six-element character string. As shown in Figure 7, the first five positions of the string were used to store the temperature data in character form, and the last position was used to assign an identifier character.

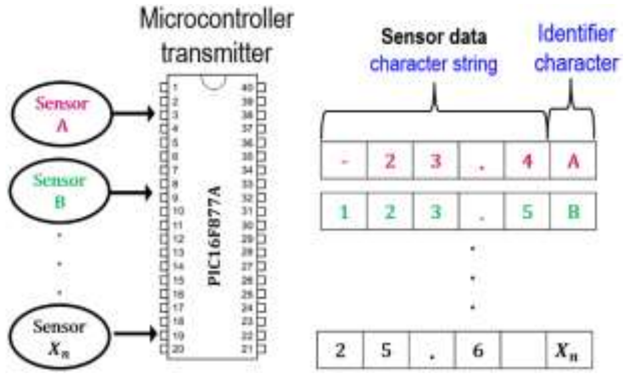

Figure 7 Storage of temperature data from analogue and digital sensors in a six-character string
Source: Own Elaboration

Each of the sensors connected to the transmitter circuit's PIC microcontroller was assigned an identifier character, which was assigned in the fifth position of each sensor's temperature data character string. Table 3 shows the identifier character assigned to each sensor.

LM35 sensor characters					*Identifier*
C_0	C_1	C_2	C_3	C_4	*A*
TMP36 sensor characters					***Identifier***
C_0	C_1	C_2	C_3	C_4	*B*
DS1621A sensor characters					***Identifier***
C_0	C_1	C_2	C_3	C_4	*C*
DS1621B sensor characters					***Identifier***
C_0	C_1	C_2	C_3	C_4	*D*
DS1621C sensor characters					***Identifier***
C_0	C_1	C_2	C_3	C_4	*E*

Table 3 Assignment of the identifier character for each of the digital and analogue sensors
Source: Own Elaboration

Figure 8 shows the instructions that were used in the program code of the transmitter circuit's PIC microcontroller to form the character string for each of the temperature data.

```
sprintf(string,"%1.1f",Temperature1);
string[5] = 'A';
SEND_SERIAL_DATA();
```

Figure 8 Instructions of the program code in C language of the PIC microcontroller of the transmitter circuit, to form the string of characters of each temperature data
Source: Own Elaboration

The instruction sprint *sprintf(string,"%1.1f",Temperature1)* was used to convert a temperature data into character format. With the instruction string[5] = 'A' the identifier character was assigned to the fifth position of the character string.

UART asynchronous serial communication was used to send the temperature data in character string format from the *transmitter circuit* to the *receiver circuit*. Figure 9 shows the transmission of a character string by asynchronous serial communication of a temperature data.

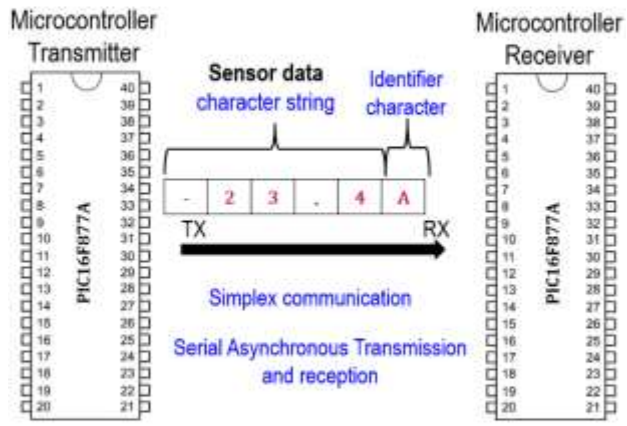

Figure 9 Temperature data sent asynchronously serially from the transmitting circuit's PIC microcontroller to the receiving circuit's PIC microcontroller
Source: Own Elaboration

The RC6/TX terminal of the physical UART port of the transmitter circuit's PIC microcontroller was used to send the characters of each temperature data. In the PIC microcontroller of the receiver circuit, the *RC7/RX* terminal was used to receive the characters of each temperature data.

In figure 10, the instructions inside the *void* function *SEND_SERIAL_DATA()* were used in the program code of the transmitter circuit's PIC microcontroller to send by asynchronous serial communication the characters of each temperature data.

```
void SEND_SERIAL_DATA()
{
 for (i = 0; i <= 5; i++)
    {
       putc(string[i]);
       delay_ms(100);
    }
}
```

Figure 10 Instructions for sending the six characters of each temperature data via asynchronous serial communication
Source: Own Elaboration

The instruction putc *putc(string[i])*, was used to send through the *RC6/TX* terminal of the transmitter circuit's PIC microcontroller, the six characters of each temperature data. The characters were sent one by one with a delay of 100 milliseconds.

When receiving the temperature data with the receiving circuit's PIC microcontroller, the six characters were received one by one via the UART interrupt, and then transformed to floating-point data, as shown in figure 11.

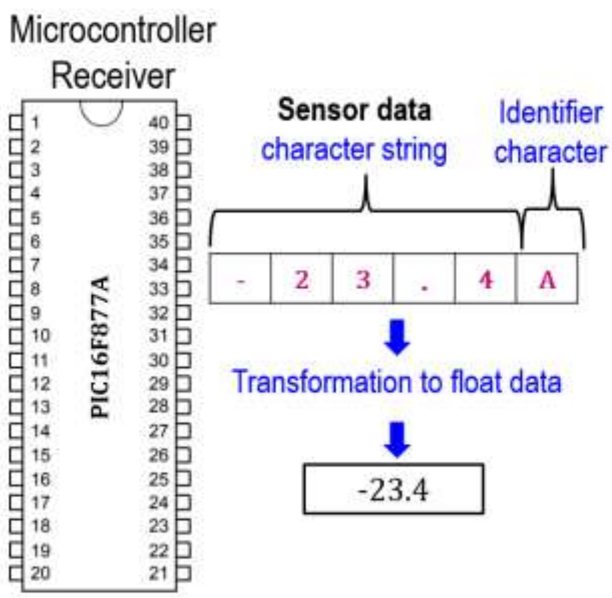

Figure 11 Representation of the reception and transformation of the character string of a temperature data to a floating point number
Source: Own Elaboration

In the program code of the receiving circuit's PIC microcontroller, the variable *string[i]* was declared to store the characters received. The variable i was declared as a global variable and assigned an initial value of zero (i = 0). Figure 12 shows the flowchart of the programming algorithm for character reception, conversion to floating data and display on the LCD screen.

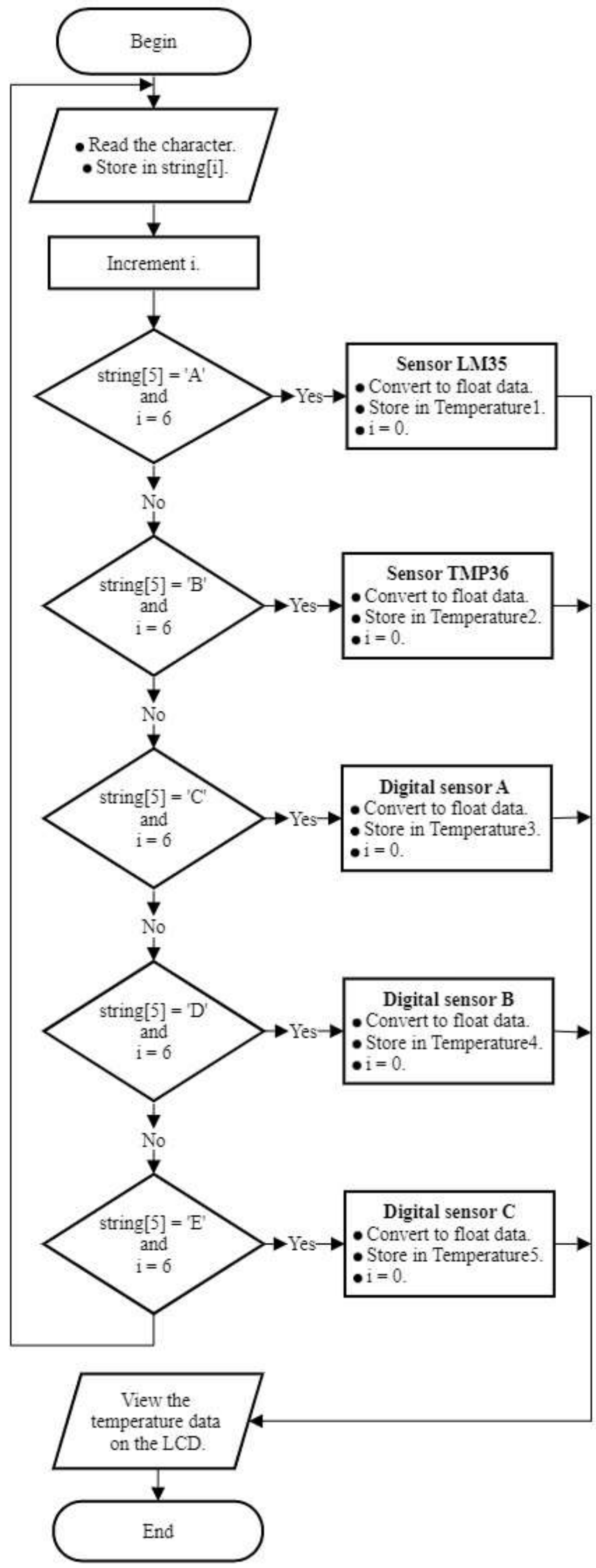

Figure 12 Flowchart of the programming algorithm for the PIC16F877A of the receiver circuit
Source: Own Elaboration, flowchart developed with smartdraw online

In the program code for the PIC16F877A *receiver circuit*, the interrupt function of the UART receiver was used to receive the characters via the *RC7/RX* terminal, and the instructions of the main function of the program code were used to display the temperature data of the analogue and digital sensors on the LCD.

Figure 13 shows the interrupt function *RDA_isr()*, used to receive the characters by the *RC7/RX* terminal. The instructions within this function were used to receive, store, and convert to floating data, and identify which sensor the characters belong to.

```
RDA_isr()
{
   string[i] = getc();
   i++;
     if (string[5] == 'A' && i == 6)
      {
        i = 0;
        Temperature1 = atof(string);
      }
}
```

Figure 13 Interrupt function to receive, store, convert to a floating point number, and identify which sensor the string belongs to
Source: Own Elaboration

The instruction *string[i] = getc()* was used to wait for characters from the *RC7/RX* terminal and store them in the variable *string[i]*. With the control statement and the evaluation of the expression *if(string[i] == 'A' && i == 6*) the identification of the characters belonging only to the LM35 sensor was made. The instruction *Temperature1 = atof(string)* was used to convert the characters of the temperature data from the LM35 sensor to a floating point number and store it in the variable *Temperature*1. The instructions to convert the temperature characters to a floating point number for the TMP36 sensors and the other DS1621 digital sensors are not shown in Figure 13, but if they were implemented in this work, a control statement with its expression evaluation must be added inside the serial interrupt function *RDA_isr()*, to detect the characters of each sensor to be added to the unidirectional point-to-point communication.

Results

Stage 5: Unidirectional point-to-point wireless communication

Once stage 4 was completed, the next step was to connect the XBEE RF modules to the PIC microcontrollers. Figure 14 shows the connections of the XBEE modules to the PIC microcontrollers.

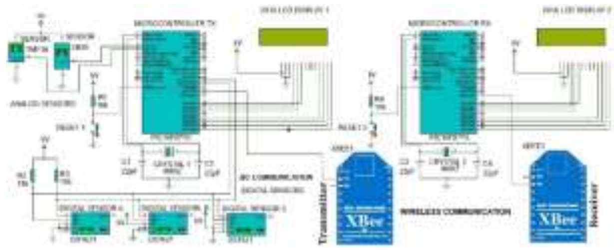

Figure 14 Connecting the XBEE modules to the terminals of the PIC16F877A microcontrollers for wireless communication
Source: Own Elaboration, electronic diagram obtained with the PROTEUS ISIS graphical environment

In the transmitter circuit, the TX terminal of the XBEE module was connected to the RC6/TX terminal of the PIC microcontroller. In the receiver circuit, the RX terminal of the XBEE module was connected to the RC7/RX terminal of the PIC microcontroller.

To achieve wireless communication in PROTEUS, the virtual serial port emulator VSPE version 0.938.4.846 was used to emulate the connection between two virtual ports. Figure 15 shows the component editing window to edit the characteristics of the XBEE module of the *transmitter circuit*.

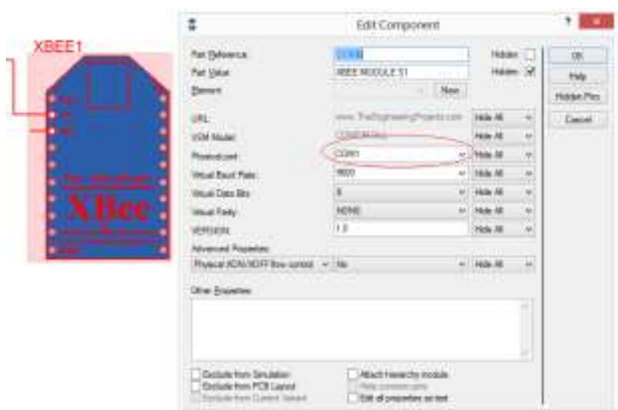

Figure 15 Editing the XBEE module characteristics of the transmitter circuit in the component edit window
Source: Own Elaboration, figure obtained from PROTEUS ISIS environment

For the XBEE module of the *transmitter circuit*, the virtual COM1 port was chosen, with a transmission speed of 9600 bits/second.

Figure 16 shows the component editing window for editing the characteristics of the XBEE module of the *receiver circuit*.

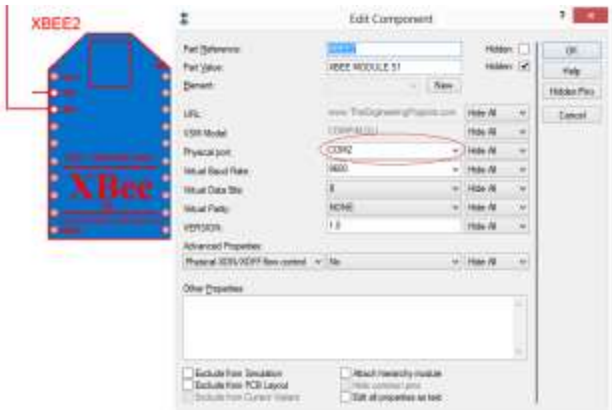

Figure 16 Editing the characteristics of the XBEE module of the receiver circuit in the component editing window
Source: Own Elaboration, figure obtained from PROTEUS ISIS environment.

In the component edit window, the virtual COM2 port with a baud rate of 9600 bits/second was chosen. The baud rate between the transmitter circuit and the *receiver circuit* must be equal for communication between them.

To emulate the virtual ports COM1 and COM2, in the main window of the virtual serial port emulator VSPE, a new device was selected to enter the device creation wizard, as shown in Figure 17.

Figure 17 Creating a new device in the virtual serial port emulator VSPE
Source: Own Elaboration, figure obtained from the VSPE emulator.

In the specify device type window, shown in Figure 18, the *pair* option was chosen, which is to connect two virtual serial ports.

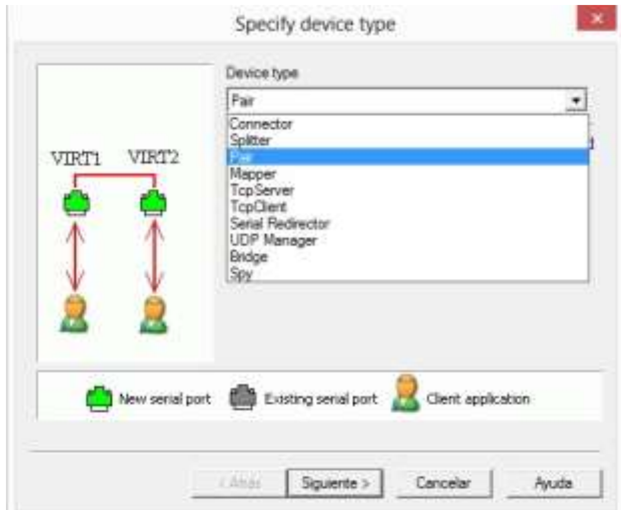

Figure 18 Creation of two virtual ports in the VSPE emulator
Source: Own Elaboration, figure obtained from the VSPE emulator.

In the window to specify the device characteristics, the COM3 and COM4 ports were chosen, taking into account that the same ports selected in the computer equipment should not be used (figure 19).

Figure 19 Selection of ports in VSPE to carry out emulation
Source: Own Elaboration, figure obtained from the VSPE emulator.

Finally, in the *emulation started* window, shown in Figure 20, the status of the virtual ports will appear, indicating that the ports selected in PROTEUS can be emulated.

ISSN 2414-4924

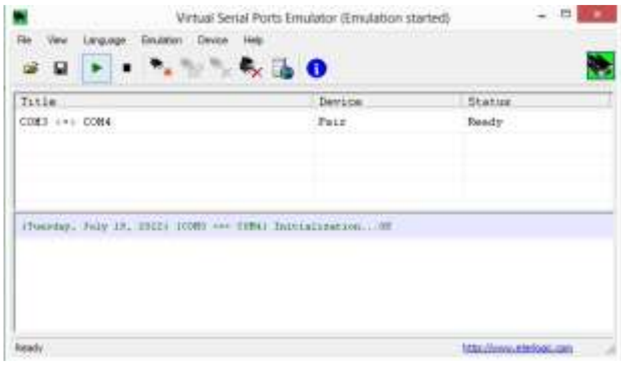

Figure 20 Status of the virtual ports in the emulation started window
Source: Own Elaboration, figure obtained from the VSPE emulator

The analogue and digital sensors were set to different temperature values to verify wireless point-to-point communication. Table 4 shows the temperature values at which the sensors were set to perform the communication test.

Analog Sensor	Temperature value
LM35	27 °C
TMP36	-40 °C
Digital Sensor DS1621	
A	35 °C
B	40 °C
C	27 °C

Table 4 Temperature values for testing the wireless communication in the simulation
Source: Own Elaboration

Figure 21 shows the temperature value of the LM35 analogue sensor during one-way wireless communication.

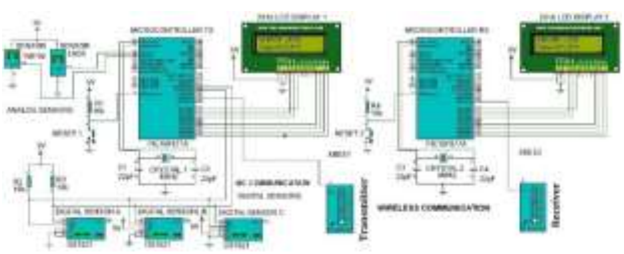

Figure 21 Wireless simulation: LM35 sensor data sent by the transmitter circuit and received by the receiver circuit
Source: Own Elaboration, figure obtained from PROTEUS ISIS environment.

On the LCD screen of the transmitter circuit, the temperature data were displayed after being sent. On the LCD screen of the receiver circuit, the temperature data were displayed after being received.

Figure 22 shows the temperature of the TMP36 sensor in the transmitter and receiver circuits.

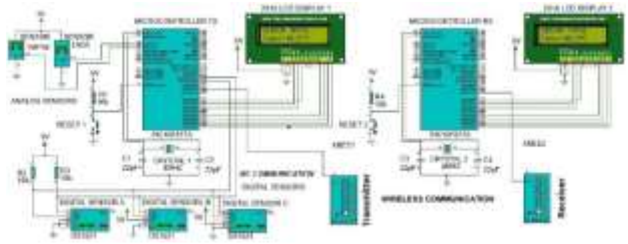

Figure 22 Wireless simulation: TMP36 sensor data sent by microcontroller TX and received by microcontroller RX
Source: Own Elaboration, figure obtained from PROTEUS ISIS environment.

Figure 23 shows the temperature of the digital sensor A DS1621, after being sent by the transmitter circuit and received by the receiver circuit.

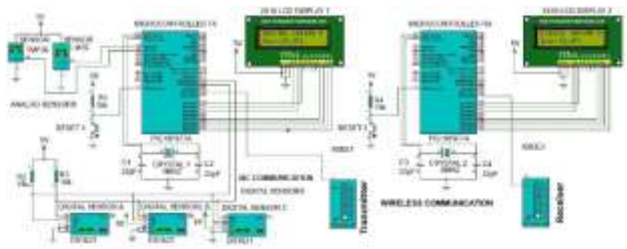

Figure 23 Wireless simulation: data from digital sensor A DS1621 sent by the transmitter circuit and received by the receiver circuit.
Source: Own Elaboration, figure obtained from PROTEUS ISIS environment.

Figure 24 shows the temperature of the digital sensor B DS1621, when it is sent by the transmitter circuit and received by the receiver circuit.

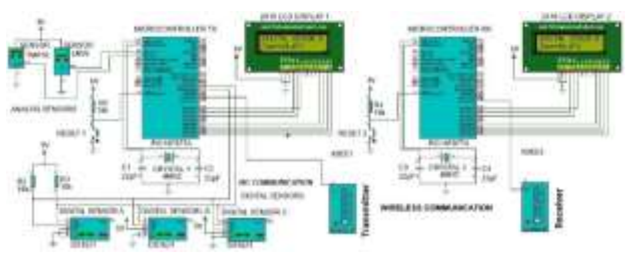

Figure 24 Wireless simulation: data from digital sensor B DS1621 sent by the transmitter circuit and received by the receiver circuit.
Source: Own Elaboration, figure obtained from PROTEUS ISIS environment

Figure 25 shows the temperature of the digital sensor C DS1621 during wireless point-to-point communication.

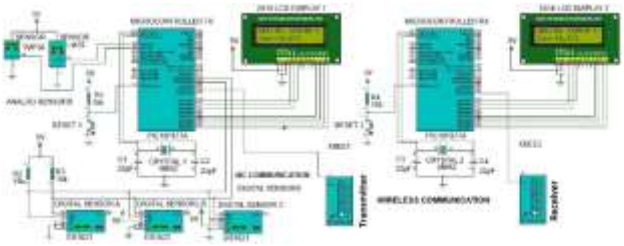

Figure 25 Wireless simulation: data from digital sensor C DS1621 sent by the transmitter circuit and received by the receiver circuit.
Source: Own Elaboration, figure obtained from PROTEUS ISIS environment

Finally, the RXD terminal of the VIRTUAL TERMINAL instrument was connected between the RX terminal of the XBEE module and the RC7/RX terminal of the PIC microcontroller of the receiver circuit to display the characters sent asynchronously from the PIC microcontroller of the transmitter circuit to the PIC microcontroller of the receiver circuit. The reception of the characters is shown in figure 26.

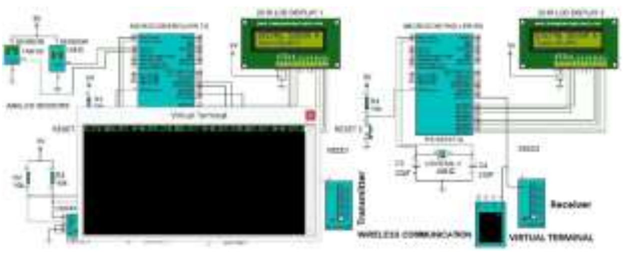

Figure 26 Display of the temperature data in character format with the VIRTUAL TERMINAL instrument
Source: Own Elaboration, figure obtained from PROTEUS ISIS environment

Acknowledgement

We are grateful for the support granted by the Polytechnic University of Juventino Rosas, which through the department of Network and Telecommunications Engineering provided the facilities for students to participate and contribute to this technological development work. We are also grateful for the collaboration of the authors of this article who participated in the work.

Conclusions and future work

In the simulation, asynchronous serial communication was implemented to send only the temperature data from five sensors of the *transmitter circuit* to the *receiver circuit*, however, the data sent can be of other variables such as: relative humidity, illumination, and CO_2.

The program code of the receiver circuit's PIC microcontroller can be modified to receive and identify data from more than five sensors.

In this work, a programming algorithm was designed to receive and identify data from different sensors. No matter if the data is from a temperature, relative humidity, illumination or CO_2. sensor, all data will be received with the *receiver circuit* in character packets. The importance to be given in this work, is that any data received by the receiver circuit is very well identified, since in future simulations it is intended to use up to 20 sensors approximately.

The PIC microcontroller in the *transmitter circuit* measured the temperature of two analogue and three digital sensors, and then the temperature data were displayed on an LCD screen and sent to the receiver circuit. In future work, it is recommended that analogue and digital relative humidity and illumination sensors be connected to the *transmitter circuit's* PIC microcontroller to simulate the measurement, display and sending of data. The sensor libraries available in the PROTEUS software should be used.

In this wireless communication, the *receiver circuit* performed the reception of the temperature variables, but did not implement the control of an actuator. The programming algorithm of the PIC microcontroller of the *receiver circuit* was designed only to receive by serial communication, identify and display on an LCD screen the data from the digital and analogue sensors. Future work can connect output peripherals to the PIC microcontroller, and modify the program code to control actuators based on a decision made by the received data.

The information exchange between the transmitter circuit and the *receiver circuit* was unidirectional. In the next phase of the simulation it is intended to implement a bidirectional wireless communication, where both electronic circuits have the role of transmitting and receiving data. In the bidirectional communication, the receiver circuit can respond to the *transmitter circuit* when it wants to receive data and when it does not, or the *receiver circuit* can tell the transmitter circuit that it only needs data from certain sensors.

In future work, it is intended to implement a wireless sensor network (WSN) with XBEE technology. The nodes of the wireless network will be sensors connected directly to the terminals of an XBEE module, to measure environmental variables in a timely manner. The coordinator in the wireless network will be a PIC microcontroller with an XBEE module as main devices. where it is available. The coordinator will strategically receive all the data sent by the nodes.

References

AGTECH AMERICA. (4 de noviembre de 2020). Obtenido de beneficios de la agricultura protegida en un invernadero: http://agtechamerica.com/beneficios-de-la-agricultura-protegida-en-un-invernadero/

Cama-Pinto, A., Gil-Montoya, F., Gómez-López, J., García-Cruz, A., & Manzano-Agugliaro, F. (2014). Sistema inalámbrico de monitorización para cultivos en invernadero. *Dyna*, *81*(184),164-170. [fecha de Consulta 24 de Julio de 2022]. ISSN: 0012-7353. Disponible en: https://www.redalyc.org/articulo.oa?id=49630405023

Castaño-Giraldo, S. A. (Junio de 2022). Comunicación serial con PIC. Control automático educación. https://controlautomaticoeducacion.com/microcontroladores-pic/16-comunicacion-serial-con-pic/

DIGI. (Junio de 2022). Obtenido de ¿qué es XBEE?: https://xbee.cl/que-es-xbee/

Eterlogic. (Junio de 2022). Obtenido de emulador de puertos serie virtuales, la solución de emulación: http://www.eterlogic.com/Products.VSPE.html

García-Breijo, E. (2008). Compilador C CCS y simulador PROTEUS para microcontroladores PIC. (Primera edición, página 23). Editorial MARCOMBO.

García-García, F. J. & Martínez-Tornero, J. F. (2016). Control climático en invernaderos y las nuevas aplicaciones. *NUTRICONTROL*, *Revista agrícola* (No. 7): https://nutricontrol.com/es/control-climatico-en-invernaderos-y-las-nuevas-aplicaciones/

INDIELEC. (30 de agosto de 2021). La simulación de circuitos electrónicos. *Blog de Ingeniería de diseño electrónico*. https://www.indielec.com/la-simulacion-de-circuitos-electronicos-blog-4-50-284/

INTAGRI. (s.f). Obtenido de principios básicos para el manejo climático de invernaderos: https://www.intagri.com/articulos/horticultura-protegida/%20%20principios-basicos-para-el-manejo-climatico-de-invernaderos

Labcenter. (Junio de 2022). Obtenido de simulación de circuitos: https://www.labcenter.com/simulation/

Morán-Inga, J.A. (2019). Diseño y simulación de un sistema de monitoreo usando módulos XBEE para disminuir el costo elevado del mantenimiento de jardines tipo vertical. Obtenido de https://repositorio.utp.edu.pe/handle/20.500.12867/3560

Villalobos, M., Mamani, M., & Herrera, R. (2017). Sistema web de bajo costo para monitorear y controlar un invernadero agrícola. Ingeniare. Revista Chilena de Ingeniería, 25(4),599-618. [fecha de Consulta 22 de Julio de 2022]. ISSN: 0718-3291. Disponible en: https://www.redalyc.org/articulo.oa?id=77254022005