Wireless Communication between two robots using a piconet network

Comunicación inalambrica entre dos robots usando una red piconet

HERNÁNDEZ-MENDOZA, César Manuel†*, SERRANO-RUBIO, Juan Pablo and MANJARREZ-ESTRADA, Ernesto

Instituto Tecnológico Superior de Irapuato, Department of Computer Systems Engineering, Irapuato, Guanajuato, Mexico

ID 1st Author: *César Manuel, Hernández-Mendoza* / **ORC ID**: 0000-0003-4667-9816

ID 1st Coauthor: *Juan Pablo, Serrano-Rubio* / **ORC ID**: 0000-0003-3705-5112

ID 2nd Coauthor: *Ernesto, Manjarrez-Estrada* / **ORC ID**: 0000-0001-6866-6413

Received September 15, 2019; Accepted December 10, 2019

Abstract

The objective of this work is to develop an investigation related to the configuration of a piconet network. This piconet network consist of two mobile devices based on an architecture of communication master-slave. One of the devices sends orders via wireless and the other performs the instructions as well as it works with the parameters indicated by the master devices, such as: speed, time and direction. An Arduino nano, an engine controller, an aluminum chassis and caterpillar tires which visually resembles mobile devices to small tanks were employed. The physical structure of our approach can be adapted to various applications in the industry, production lines or in the automation of food services for the transfer of plates from one point to another. Its configuration is implemented based on AT commands and, as a result, we obtained a product that does not depend on the mechanical system, since the technology developed can be implemented in any device or mobile mechanism that can be directed by the motor controller. Finally, we procured a documented investigation related to the configuration and programming of the piconet network, as well as the measurements that indicate a speed-distance-voltage relation.

Piconet, Prototype, Arduino Nano, HC05

Resumen

El objetivo de este trabajo es desarrollar una investigación relacionada con la configuración de una red de piconet. Esta red de piconet consta de dos dispositivos móviles basados en una arquitectura de comunicación maestro-esclavo. Uno de los dispositivos envía pedidos por vía inalámbrica y el otro realiza las instrucciones y trabaja con los parámetros indicados por los dispositivos maestros, tales como: velocidad, tiempo y dirección. Se empleó un Arduino nano, un controlador de motor, un chasis de aluminio y neumáticos de oruga que visualmente se asemeja a dispositivos móviles a tanques pequeños. La estructura física de nuestro enfoque se puede adaptar a diversas aplicaciones en la industria, líneas de producción o en la automatización de servicios de alimentos para la transferencia de platos de un punto a otro. Su configuración se implementa en base a comandos AT y, como resultado, obtuvimos un producto que no depende del sistema mecánico, ya que la tecnología desarrollada puede implementarse en cualquier dispositivo o mecanismo móvil que pueda ser dirigido por el controlador del motor. Finalmente, obtuvimos una investigación documentada relacionada con la configuración y programación de la red de piconet, así como las mediciones que indican una relación de velocidad-distancia-voltaje.

Piconet, Prototipo, Arduino Nano, HC05

Citation: HERNÁNDEZ-MENDOZA, César Manuel, SERRANO-RUBIO, Juan Pablo and MANJARREZ-ESTRADA, Ernesto. Wireless Communication between two robots using a piconet network. Journal of Technological Prototypes. 2019. 5-16: 12-20.

* Correspondence to Author (email: cesar.hernandez@itesi.edu.mx)

† Researcher contributing first author

Introduction

Bluetooth is a short-range wireless communication technology, initially developed to replace cable connections. These connections frequently use point-to-point links between Bluetooth devices; however, the specifications of this protocol offer solutions for the most complex interconnection topologies, called Piconets (Bluetooth, 2004).

To analyze Bluetoth technology, some theoretical concepts are required, such as: analog signal, which is represented by various functions that take an infinite value of values in sinusoidal signals to transmit data; digital signal, that is represented by the coding of electrical pulses to represent ones and zeros according to the increase or decrease of voltage; wireless network, which consists of the connection of two or more nodes connected to each other by means of electromagnetic waves; carrier wave, which is an analog signal in sinusoidal form that contains data and information that travels through a wireless medium; modulation and demodulation, that are techniques in which a series of conversions are processed to generate a digital signal in analog or vice versa. Once synchronized, two devices in the wireless network must operate on the same channel and frequency to transmit the data with a certain amplitude and carrier wave period.

Bluetooth communication uses the Simplex transmission type, which consists of one-way transmission such as the modules available for Arduino technology; a simple example is to be able to control a land drone through the cell phone, in this case, the user through the device orders the drone to move only, also in the synchronization of two phones, or in the pairing of the phone with the car. When talking about a piconet network that uses several bluetooth slaves and a master, the type of Half-Duplex transmission is used, which, unlike Simplex, allows communication in two directions in one direction at a time, that is, it uses an alternate transmission.

Bluetooth is one of the most common wireless transmission means that we can find today, it is efficient, fast and economical; thus, it is an analog medium capable of transmitting data and voice at high frequencies in which the IEEE802.15.1 standard operates.

Like any transmission medium, it requires a transmitter and receiver in most equipment and devices such as phones, tablets, cameras or laptops. There are 3 classes of Bluetooth, which can be differentiated due to the scope they have: Class 1: 100 meters approximately, Class 2: 10 meters and Class 3: 1 meter. The hardware it requires consists of the radio it modulates to transmit the signal and a digital controller that processes the signals.

To begin with the development of a piconet network it is necessary to know some basic aspects of electronics, telecommunications and programming, which are the scientific areas involved in the project. The topology of a piconet network is a star configuration: slave devices will connect to a master, who will regulate traffic on the channel. By definition, the unit that establishes the piconet assumes this role and all others will be slaves (Moraleda, Gonzalez, Almendra, Jimenez-Leube, & Sanz Maudes, 2014).

In the electronic part, it is necessary to know the terms that are used for the connection pins of the bluetooth modules, Arduino nano, and the motor controller, the latter being the most complex due to the various configurations available for motor control, since they must be adapted and synchronized based on programming.

Arduino Nano, as the name implies, is an abbreviated version of a common Arduino, for the creation of smaller electronic objects (Caicedo Pedrera, 2017). The voltage flow and resistance are other parameters that must be well mastered in order to adapt the voltage supply to the various modules with which they work, since we used 5-volt components, while the motor controller has an input from 9 to 32, and this in turn can have an output of five, as well as digital and analog inputs.

A particular case of digital signal is the binary signal, where the number of possible values is only 2. Knowing this type of signals is important because in electronics, it is very common to work with voltages (or intensities) with only two values (Torrente Artero, 2016). Likewise, an analog electrical signal the values of the voltage vary constantly and can take any value (Crespo, 2017).

To program and configure the piconet network, it is necessary to establish a digital and direct communication with the hardware based on the factory settings of the bluetooth modules using AT commands (Attention Command). The AT commands were developed by Dennis Hayes in 1977, and they are a set of universal open standard commands that were created in order to control and configure modems from a PC or terminal (Aranda, 2014). These are coded instructions that make up a communication language between man and a terminal, whose main purpose is communication with modems; however, GSM mobile telephony has also adapted this language as a standard to be able to communicate with its terminals (Bluehack the Spanish Bluetooth Security Group, 2005).

One of the best-known parameters for identifying devices within a network is the IP address of the device, but this IP address can change dynamically, so it is not an accurate value. Therefore, there is the MAC address (acronym for Media Access Control), a unique and irreplaceable identifier for any network device. In this way, it is much more precise to identify a specific computer, a printer or a smartphone by its MAC address (Barbero, 2018). Some of the parameters that are configured are: baud rate, password, master/slave and the MAC address of the slave device in the master, among others. This ensures efficient and secure communication between the two mobile prototypes as there is communication, if and only if, one has the MAC address and password set.

In terms of telecommunications, it is necessary to know the characteristics of bluetooth technology, such as modulation and multiplexing techniques, to understand the transmission by channels, along with characteristics of the wireless medium like amplitude, period and frequency of a carrier wave, since this allows to determine the proper transmission speed between one device and another. Bluetooth technology was designed from the outset to support data and multiple voice channels with a range of approximately ten to one hundred meters, without requiring a direct line of sight between the devices (Bellido O., De la Cruz F., Torres R., & Gistas P., 2004). As for the transmission speed, the 2.4 GHz band is the one that is physically responsible for sending and transmitting the information.

It emits and receives the electrical signals to the other bluetooth devices of the piconet (Varela & Domínguez, 2002). Finally, the main advantage of Bluetooth is the ability to handle voice and data transmission simultaneously, with the ability to support an asynchronous data channel and up to three synchronous voice channels (Torres Hurtado & Álvaro Bernal, 2006).

Development

The configuration begins by identifying the conventional and easily accessible bluetooth modules, since it is common to find two types of them, the HC05 and the HC06, which have some characteristics and differences. The HC06 module, a component with only four pins, is quickly identified, this indicates that they can only act as a slave and have the pins: Vcc (voltage), Gnd (ground), Txd (transmission) and Rxd (reception), at the same time, it has a lower cost and for its implementation it is only required to have the login password, which is usually 1234.

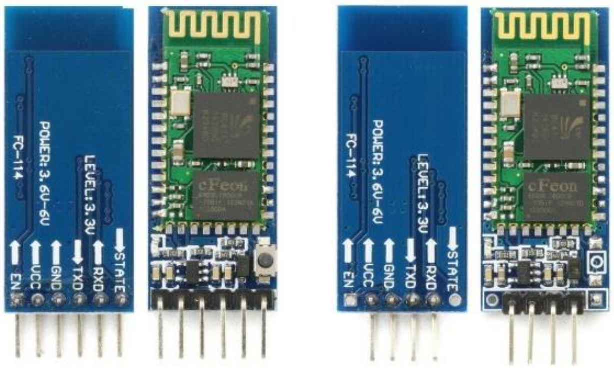


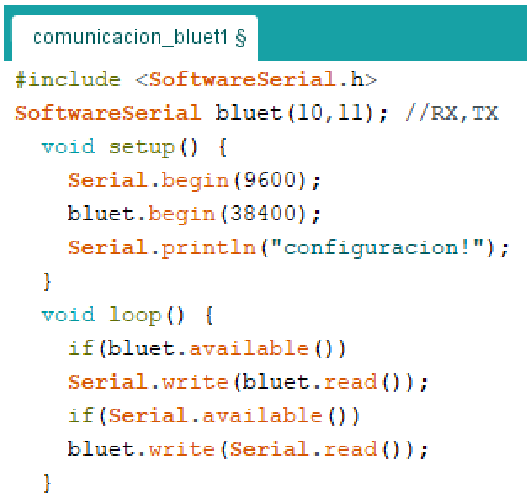
Figure 1 Bluetooth Modules
Fuente: https://aprendiendoarduino.wordpress.com/tag/hc-05/

In contrast, the HC05 module is relatively more expensive than the HC06 and can be visually identified by having six pins, Vcc (voltage), GND (ground), Rxd (reception), Txd (transmission), State (state), En (enable key) and, depending on the manufacturer, it can have or not a reset button, which is indispensable for the configuration of the Master mode. Both modules work specifically at 3.3v; however, the pins and connectors can support up to 5v. Initially, when the HC05 module is new, it is preconfigured in slave mode and ready to use. Figure 1 shows both modules, on the left the HC05, on the right the HC06.

The devices send and receive data through a channel and are continuously jumping in a pseudorandom way from one channel to another; for the communication to be correct, sender and receiver must know that sequence of jumps (Pascual, 2012). Then we proceed to configure two HC05 modules, one as a slave and one as a master.

Slave configuration

Starting the configuration, a communication code between the Arduino IDE and the Bluetooth must be uploaded, in Figure 2, the example code to enter the AT commands in the HC05 module is shown. Once the Arduino has the program, it is required to press and hold the "reset" button of the HC06 module for about 3 seconds while the voltage pin is connected. To verify that it was successful, the module LED stops blinking continuously and must make longer pauses between one ignition and another, in this way, the module enters configuration mode.



```
#include <SoftwareSerial.h>
SoftwareSerial bluet(10,11); //RX,TX
  void setup() {
    Serial.begin(9600);
    bluet.begin(38400);
    Serial.println("configuracion!");
  }
  void loop() {
    if(bluet.available())
    Serial.write(bluet.read());
    if(Serial.available())
    bluet.write(Serial.read());
  }
```

Figure 2 Configuration of AT commands
Source: *Prepared by the authors in Arduino-IDE*

A terminal is required to access the configuration of AT commands, in this case Arduino IDE, it has this option when choosing the menu: "Tools," submenu "Serial Monitor."

Subsequently, the AT commands must be entered, the most important ones are listed and exemplified in Table 1. It should be emphasized that it is important to know the MAC address of the slave module (last command in the table), since it will be required at the time of configuring the Master, to indicate the module to which it will be connected.

Commands	Description	Example	Answer
AT	Check communication	AT	Ok
AT+NAME	Assign a name to the module	AT + NAME= ESCLAVO	Ok
AT + PSWD	Assign Password	AT + PSW= 5678	Ok
AT + BAUD	Change transmission speed	AT + BAUD = 9600	Ok
AT + ROLE	Assign Slave Mode	AT + ROLE = 0	Ok
AT + BIND	Check MAC Address	AT + BIND?	XXXX, XX, XXXXXX

Table 1 Commands to configure slave
Source: Prepared by the authors

Master Configuration

In the case of the Master, the code in Figure 1, the Arduino loaded with the same program and most of the commands in Table 1 can be used again, with some changes in their parameters. Likewise, it must be kept pressing the "reset" button of the module for a few seconds, while the voltage pin is connected. The main commands to configure the Master mode are shown in table 2.

Commands	Description	Example	Answer
AT	Check communication	AT	Ok
AT+NAME	Assign a name to the module	AT + NAME= MAESTRO	ok
AT + PSWD	Assign Password	AT + PSW= 5678	ok
AT + BAUD	Change transmission speed	AT + BAUD = 9600	ok
AT + ROLE	Assign Master Mode	AT + ROLE = 1	ok
AT + CMODE	Connect Master with a particular device (slave).	AT+CMODE=1	ok
AT + BIND	Check MAC Address	AT + BIND = XXXX, XX, XXXXXX	Ok

Table 2 Commands to configure the master
Source: Prepared by the authors

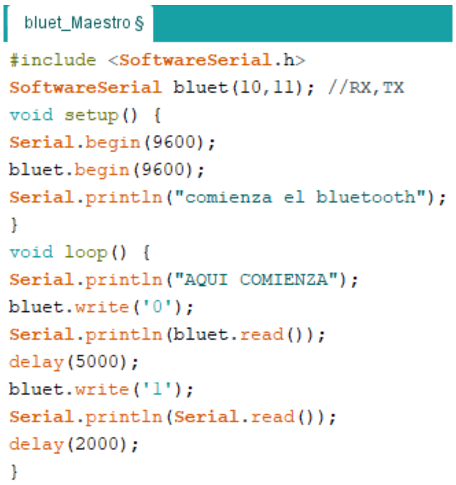
Once the Master and Slave devices have been configured, an examination is conducted. Simply supplying both devices with voltage, once they synchronize, they must stop blinking continuously. It is important to mention that during the automatic pairing process, in the first seconds both modules should flash every second, then the pauses will be longer (about 3 seconds), and their LED lights will be longer, if this action happens, the configuration is successful.

Programming and configuration of other components

Specifically, for the project developed, the configuration and programming of the following components was necessary:

- H bridge controller L298, capacity from 5 to 32v
- 12v motor, 21000rpm
- Arduino nano

The codes that are used as the basis of experimentation and tests to move forward or backward the mobile prototypes as indicated by the Master, are shown in Figure 3, in which the coding of the Master device is shown on the left side and on the right side is the coding of the slave device. It should be noted that this code is enough to make the prototypes move according to programmed instructions and based on the analysis or algorithm of the programmer, the necessary lines can be added so that the operations or actions to be performed are executed. Therefore, for demonstrative purposes this information is shared.



```
#include <SoftwareSerial.h>
SoftwareSerial bluet(10,11); //RX,TX
void setup() {
Serial.begin(9600);
bluet.begin(9600);
Serial.println("comienza el bluetooth");
}
void loop() {
Serial.println("AQUI COMIENZA");
bluet.write('0');
Serial.println(bluet.read());
delay(5000);
bluet.write('1');
Serial.println(Serial.read());
delay(2000);
}
```

Figure 3 Experimentation Code
Source: Prepared by the authors in Arduino-IDE

The electronic scheme was generated in the free software "Fritzing," since it allows to visualize the connections and pins used.

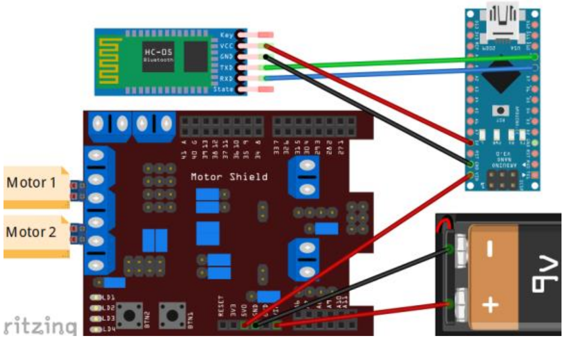


Figure 4 Schematic design of the prototype
Source: Prepared by the authors in Frtinzing

It is important to emphasize that Figure 4 is only a representative image and only similar to the real model, since the diversity and electronic components are in permanent development, the "H-bridge L298" module used is not available in Fritzing, so we used a Motor Shield for representation.

Algorithm used

The algorithm that was implemented considers the measurement and control of some important parameters, such as the response time of the slave, the established speed of the motors, the travel time in variance with the distance that it should travel, as well as the adaptation of the time of the slave according to the pattern to follow. One of the objectives of the mobile prototype was to go through different patterns, as rails in a production line, like a prototype capable of being implemented in the manufacturing industry.

Motor control pins:

- D1 advance motor 1
- D2 reverse engine 1
- D3 advance motor 2
- D4 reverse motor 2

Instruction sent from the Master to the Slave:

A = Advance
R = Go back
S = Stop
Terms:

Si Maestro avanza then Esclavo avanza

(Maestro= ON 5 segundos, D1=ON, D3=ON, velocidad=200, SEND= A,
, Esclavo= 5 segundos, D1=ON, D3=ON, velocidad= 200)

Si Maestro para then Esclavo avanza

(Maestro= OFF 5 segundos, D1=OFF, D3= OFF, velocidad=0, SEND= A
, Esclavo= 5 segundos, D1=ON, D3= ON, velocidad= 200)

Si Maestro para then Esclavo retrocede

(Maestro= OFF 5 segundos, D1=OFF, D3= OFF, velocidad=0, SEND= R,
, Esclavo= 5 segundos, D2= ON, D4= ON, velocidad= 200)

Si Maestro retrocede then Esclavo retrocede

(Maestro= ON 5 segundos, D2=ON, D4= ON, velocidad=200, SEND= R,
, Esclavo= 5 segundos, D2=ON, D4= ON, velocidad= 200)

Si Maestro retrocede Esclavo para

(Maestro= ON 5 segundos, D2=OFF, velocidad=200, SEND= S,
, Esclavo= 5 segundos, velocidad= 0)

Si Maestro retrocede Esclavo avanza

(Maestro= ON 5 segundos, D2= ON, D4= ON, velocidad=200, SEND= A,
, Esclavo= 5 segundos, D1= ON, D3= ON, velocidad= 200)

Some parameters always maintain the same value; however, they can be adapted according to the needs (routes or inclinations), in this case, the time and speed of the motors, since there is a distance-time relationship.

Results

To generate the results, some necessary tests were carried out to determine the efficiency of the prototypes by measuring the path of the Master in relation to the distance and time traveled in comparison to the Slave. One of the doubts to be resolved was the relatively short time it takes for the Master to send the instruction wirelessly and that the Slave can also process it. Specifically, this sending/receiving period was minimal, almost imperceptible, but it exists.

In Table 3, experimental results are shown with which a latency in the travel time that affects the Slave prototype is verified.

The values of the motors in Arduino are from 0 to 250, there is a voltage ratio, in which, if the motor speed is 0 then the assigned voltage is 0v, if the motor speed is 250, the assigned voltage is a voltage higher than it is working, in this case it is 9v. For which the route of both (Master and Slave) is programmed at a speed of 200 units. It should be noted that the minimum speed is 180 and maximum 250, since, if programmed to less than 180, the motors do not generate enough force to rotate, so the prototype does not move. Also, in Table 3, it is observed that the Time columns (sec.) are very even and imperceptible to the eye, but it is determined that the greater the distance traveled, the time needed by the Slave must be greater than that of the Master, in an approximate proportion of two hundredths of a second.

Master		
Speed	**Distance**	**Time(sec.)**
200	1.27 m	2.24
200	5 m	8.81
200	10 m	17.63
200	15m	26.4
Slave		
Speed	**Distance**	**Time(sec.)**
200	1.27 m	2.26
200	5 m	8.84
200	10 m	17.72
200	15 m	26.8

Table 3 Results of experimental tests
Source: Prepared by the authors

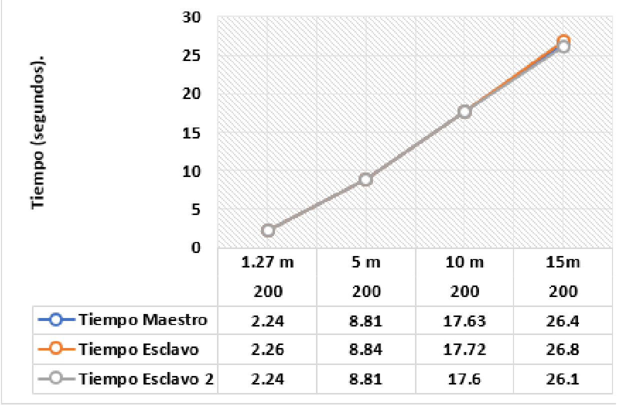
To improve these and obtain more equitable and precise comparative results, it was necessary, during the experimentation stage, to modify the parameter of the speed in the Slave, in order to match the values thrown by the Master. In Table 4, it is observed that the Slave speed is 220.

Master		
Speed	**Distance**	**Time(sec.)**
200	1.27 m	2.24
200	5 m	8.81
200	10 m	17.63
200	15 m	26.4
Slave		
Speed	**Distance**	**Time(sec.)**
225	1.27 m	2.24
225	5 m	8.81
225	10 m	17.60
225	15 m	26.1

Table 4 Results of experimental tests
Source: Prepared by the authors

Finally, Figure 6 shows a graph in which Tables 3 and 4 are presented, in which the path of:

- Master Time: 200 speed, blue line
- Slave Time: speed of 200, orange line
- Slave Time 2: 220 speed, gray line



	1.27 m	5 m	10 m	15m
	200	200	200	200
Tiempo Maestro	2.24	8.81	17.63	26.4
Tiempo Esclavo	2.26	8.84	17.72	26.8
Tiempo Esclavo 2	2.24	8.81	17.6	26.1

Figure 5 Graph with the obtained results
Source: Prepared by the authors

It is important to mention that the blue line can not be perceived because the times obtained once the speed in the Slave is modified to 220 equals the values obtained by the Master with a speed of 200, so both lines are overlapped, which is evidence of accurate results.

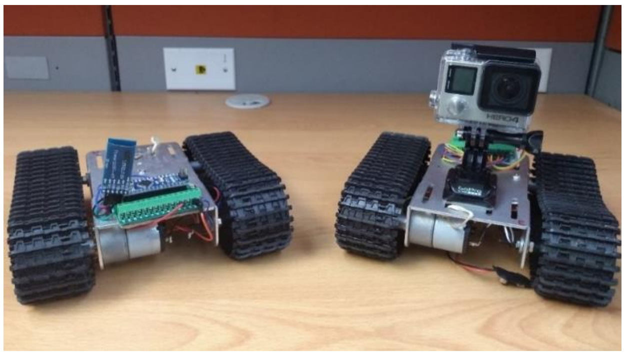


Figure 7 Piconet network prototypes
Source: Prepared by the authors

In figure 7, the two mobile prototypes are shown with the chassis, the tracks, engines, Arduino nano, controller, bluetooth modules, in addition to the finished project, a GoPro camera was added, capable of sending the live transmission of the route to the phone or laptop via Bluetooth added to the piconet network.

Discussion

The implementation of a Piconet network applied to mobile robots can improve production processes in manufacturing industries, so this project is feasible to apply with a chassis and professional mechanical system that allows the transfer of material between a workstation and another, since the brain of the project lies in the implementation of wireless technology and programming developed based on actuators, in addition to being able to add sensors that match the needs of the operator or the company.

In industrial environments, it is common to monitor parameters where Bluetooth can form a network of sensors and measuring instruments that removes the connections, which would allow the connection, monitoring and programming of PLCs and RTUs programmable logic controllers, and field points installed in production lines or plants (Linares Ruiz, Quijano Vásquez, & Holguin Londoño, 2004). This feature is one of the qualities of the proposed system.

Some of the benefits found include the following:

1. It is an automated but intelligent system to make decisions, according to the conditions or states of the master device.

2. It is a secure wireless system that pairs only with the selected equipment, avoiding interference.

3. It is adaptable to the needs and conditions of the company's production line.

4. It provides a work area free of operators or work to maintain its integrity.

5. Increased production within the production line.

6. It reduces the amount of time on production lines at the time of failures or stoppages.

7. Increased personnel safety in the area where this system will be placed.

8. It avoids deception or anomalies in production information.

9. It allows the transfer on flat and steep surfaces.

Conclusion

The contribution of the project with the Piconet network lies in the freedom that the mechanical system possesses for making programmed decisions, such as the route it takes in measuring time, distance, speed, as well as stopping or continue its way.

The project had a positive impact for the work group, since a tangible product was obtained that requires the skills of a Computer Systems Engineer, as well as the generation of new prototypes capable of being optimized and improved in order to participate in events and academic competitions. Likewise, the use of the Piconet network implemented in this prototype can be applied as pilot tests in production lines of the industries, which will allow to determine what are the specific requirements and benefits that we must have in order to use them professionally.

References

Aranda, D. (17 de julio de 2014). Electronica plataformas Arduino y Raspberry Pi. *RedUSERS*, 19-24. Retrieved from https://issuu.com/redusers/docs/electronica_-_plataformas_arduino_y

Barbero, R. (20 de octubre de 2018). *computerhoy.com*. Retrieved in June 12, 2019, from https://computerhoy.com/reportajes/tecnologia/que-es-direccion-mac-tu-ordenador-movil-que-sirve-317181

Bellido O., F., De la Cruz F., J., Torres R., M., & Gistas P., J. (Octubre de 2004). Comunicación inalámbrica con Bluetooth. *Técnica Industrial*, 18-22. Retrieved in June 12, 2019, from http://www.tecnicaindustrial.es/tiadmin/numeros/15/06/a06.pdf

Bluehack the Spanish Bluetooth Security Group. (junio de 2005). Retrieved in June 11, 2019, from http://bluehack.elhacker.net/proyectos/comandosat/comandosat.html

Bluetooth. (2004). *Specification of the Bluetooth System*. Bellevue, Washington: Bluetooth Special Interest Group. Recuperado el 20 de Junio de 2019

Caicedo Pedrera, A. (2017). *Arduino para principiantes* (Segunda ed.). Estados Unidos: Createspace Independent Publishing Platform. Retrieved in June 11, 2019

Crespo, E. (12 de Junio de 2017). *aprendiendoarduino.wordpress.com*. Retrieved in August 14, 2019, from aprendiendoarduino: https://aprendiendoarduino.wordpress.com/tag/entradas-analogicas/

Linares Ruiz , R., Quijano Vásquez, J. A., & Holguín Londoño, G. A. (2004). Implementación del protocolo bluetooth para la conexión inalámbrica de dispositivos electrónicos programables. *Scientia Et Technica [en linea]* , 7.

Linares Ruiz, R., Quijano Vásquez, J. A., & Holguin Londoño, G. A. (2004). Implementación del protocolo bluetooth para la conexión inalámbrica de dispositivos electrónicos programables. *Scientia Et Technica, 7,* 7. Retrieved in April 12, 2019

Moraleda, G., Gonzalez, C., Almendra, A., Jimenez-Leube, F. J., & Sanz Maudes, J. (6 de junio de 2014). Arquitectura para la integración en red de aparatos domésticos empleando tecnología bluetooth. *Researchgate*. Retrieved in June 11, 2019, from https://www.researchgate.net/profile/Carlos_Gonzalez23/

Pascual, C. M. (junio de 2012). Bluetooth: criterios de selección y comparativa con otras tecnologías inalámbricas. *Técnica Industrial*, 77-78. Retrieved in June 13, 2019, from http://www.tecnicaindustrial.es/tiadmin/numeros/83/1224/a1224.pdf

Torrente Artero, Ó. (2016). *El mundo Geniuino-Arduino. Curso práctico de formación*. Madrid, España: RC Libros.

Torres Hurtado, J. G., & Álvaro Bernal, N. (2006). Implementación de una topología de red de dispositivos Bluetooth capaz de acceder a Internet a través de una red LAN. *Revista Avances en Sistemas e Informática*, 37-41. Retrieved in August 02, 2019, from https://www.redalyc.org/pdf/1331/133114988007.pdf

Varela, C., & Domínguez, L. (2002). Redes Inalámbricas. *Escuela Técnica Supeior de Ingeniería Informática*, 7. June 12, 2019