Design and Implementation of a System for Wireless Control of a Robot

Christian Hernández[1], Raciel Poot[1], Lizzie Narváez[2], Erika Llanes[2] and Victor Chi[2]

**[1] Department of Electronics, Universidad Autónoma de Yucatán
Tizimín, Yucatán 97700, México**

**[2] Department of Networks, Universidad Autónoma de Yucatán
Tizimín, Yucatán 97700, México**

Abstract
This article presents the design and implementation of a wireless control system of a robot with help of a computer using LPT interface in conjunction with Arduino + X-bee, which is an electronic device that uses the Zigbee protocol that allows a simple implementation, low power consumption, and allows the robot to be controlled wirelessly, with freedom of movement. In the implementation were used two Arduino with wireless communication using X-bee modules. The first Arduino + X-bee were connected to the computer, from which received signals that were sent by the wireless module to the Arduino X-bee that was in the robot. This last module received and processed signals to control the movement of the robot. The novelty of this work lies in the autonomy of the robot, designed to be incorporated into applications that use mini-robots, which require small size without compromising the freedom of their movement.
Keywords: *Integrated Circuit, Parallel Port, ATmega 168 Microcontroller, Arduino, X-bee, Zigbee.*

1. Introduction

It is currently being investigated and put on the market many wireless products because of the large development of this type of communication technologies.
Mobile robotics is no stranger to this fact; the benefits of joining the wireless communication technologies with the developments in mobile robots are clear as we can appreciate.
Nowadays, a large number of universities work with these devices, including many people without knowledge in electronics who use them, for the simplicity to artistic, textiles, musical and botanical projects, among others. This is because its range is very varied, ranging from robotics, sensors, audio and monitoring to navigation and action systems. Examples include: TiltMouse (an accelerometer that transforms Arduino into a pitch mouse), the Arduino Pong (which is a ping pong game programmed in Arduino) and the optical tachometer among others [1].
Other works consulted include important developments. In [2] we find the implementation of a mobile robot whose main skill lies in the possibility of being controlled by a remote user connected via the Internet through a Wi-Fi network running on the TCP/IP architecture. Thus, the user can control robot's movements, while through a camera also connected via Wi-Fi, watches the robot's movements using a web browser. Consulted in the investigation [3] a construction pathology detection system, based on a wireless sensor network using the Zigbee and Arduino technology is presented. This enables continuous monitoring of the parameters of interest, meeting the requirements of low consumption, ease of maintenance and installation flexibility. In [4], it is built a prototype of an embedded wireless sensor network based on easy-to-use Arduino microcontroller board and X-bee module. They consider a temperature monitoring application to demonstrate the proof of concept of their system.
Moreover, in [5] it is described a project that is a clear example of the technology and university combination. This project, called R4P, consisted on design, build and program a small quadruped robot so that it can be used in teaching to form youth in robotics. The robot was designed to be easy to set up so young people can build it for themselves, as well as it is simple to control and program. This is one reason why this type of technology has become so popular. Also, in [6] it is presented a study that designed an UAV quadrotor system; with this we will have an open platform which will allow future implementing and testing of new navigation and control systems. And finally, [7] describes a two-wheel balancing robot based on a popular open-architecture development Arduino board and the ATmega168 chip from Atmel corporation that runs on a clock speed of 16MHz with 1KB of SRAM and 16KB of flash memory.
The article structure is as follows; the second section describes the problem and the technical solution; the third section presents the logical design of the system, explaining how they operate; the fourth section describes

the implementation, including the construction of the prototype and finally, the fifth section presents the conclusions.

2. Problem Statement and Technology

The main objective to be achieved with this research is being able to control the movements of a robot wirelessly, keeping in mind that this development could be deployed later in an electronic wheelchair which is a study that is being developed parallel to this one and is being moved through a wired controller, but the ultimate goal is to be wireless. To achieve this, in this study we used the X-bee, which uses technology WPAN (Wireless Personal Area Network) and a Diecimila Arduino board which contains a ATmega 168 microcontroller, which is widely used for programming the data transmission .
At the beginning of the research it was designed a robot that was controlled by parallel port LPT (Line Printer Terminal); however, it did not allow a free movement, because the LPT connected to the computer was limited in distance, and it depended on the length of the parallel port cable (figure 1).

Fig. 1. Initial design.

Having in mind giving greater autonomy and scope to the robot, it was implemented in the system an Arduino device with an X-bee module (figure 2).

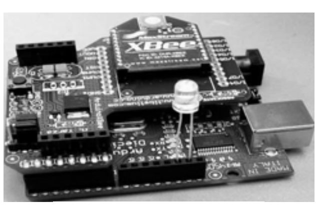


Fig. 2. Arduino + X-bee.

Arduino Diecimila was specifically the device used, first in the computer via parallel communication through the parallel port and by the other hand, implemented in the robot. The Arduino microcontroller used in the system is a board based on the ATmega 168 chip, which has 14 digital I/O, 6 of them can be used as PWM outputs (Pulse Width Modulation), 6 as analog inputs, a 16MHz crystal, USB connection and reset button. It contains everything needed to support the microcontroller, simply connecting to a computer with a USB cable and feeding it with an AC/DC (Alternating Current/Direct Current) adapter, or with battery, as in this project. The main components of Arduino Diecimila are listed below (figure 3):

- Power supply:
  - VIN: It is the input of voltage to the Arduino board when an external power supply is used (instead of 5v of the USB cable or other regulated power supply); i.e., voltage can be supplied through this pin.
  - 5v: Regulated power supply used to power the microcontroller and other components of the board. This can get from the VIN input via the integrated voltage regulator. It can be provided by a USB cable or any other regulated power supply [8].
- Memory: The ATmega168 has 16KB of memory to storage sketches (the name used for an Arduino program, it is the unit of code that loads and runs on an Arduino board, 2KB of the memory is reserved for the bootloader). It also has 1KB of SRAM (Static Random Access Memory) and 512 bytes of EEPROM (Electrically Erasable Programmable Read-Only Memory).
- Inputs and Outputs: Each one of the 14 Diecimila Arduino digital pins can be configured either as input or output using the functions pinMode(), digitalWrite() and digitalRead(), which operate at 5v. Each pin can provide or receive a maximum current of 40mA and has internal pull-up resistors (switched off by default) from 20 to 50 KOhms. In addition, some of the pins have special functions.
  - Serie, 0 (RX) and 1 (TX): Used to receive (RX) and transmit (TX) TTL serial data, which are connected with the corresponding pin of conversion from USB to Serial-TTL FTDI chip.
  - PWM 3, 5, 6, 9, 10 y 11: Generate an 8-bit output PWM signal with the function analogWrite().
  - LED 13: There is an on-board LED connected to pin 13, which if set to HIGH if LED lights and changes to LOW when the LED turns off.
- Reset: When this pin is set to LOW, it resets the microcontroller. Normally it is used when the reset button is inaccessible by the use of a shield that hides it.
- Communication: It can communicate from computer to computer with an Arduino or other microcontrollers. The ATmega168 has implemented UART (Universal Asynchronous Receiver Transmitter) TTL serial communication in the pins 0 (RX) and 1 (TX) and a FT232RL FTDI chip embedded into the motherboard that converts this serial communication into USB using the FTDI drivers to provide us a COM Virtual port to communicate with the computer. The Arduino

software includes a serial monitor, which allows simple data to be sent and received from the board through the USB connection. The TX and RX LEDs on the motherboard will flash while data is being transmitted by the FTDI chip and the USB connection to the computer [8].

- Features: The maximum length and width of the Diecimila PCB are 6.8cm and 5.3cm respectively, with USB connector and power connector protruding slightly from the edges. It contains three holes for crew to attach the board on a surface or a box. The distance between pins 7 and 8 is 4 mm, different to the other pins, which are separated by 2.5 mm [8].

The Arduino programming is very simple; however it depends on the complexity of the project. the Arduino platform is programmed using its own language based on the popular C high level programming language and supports all standard C and some C++ functions.

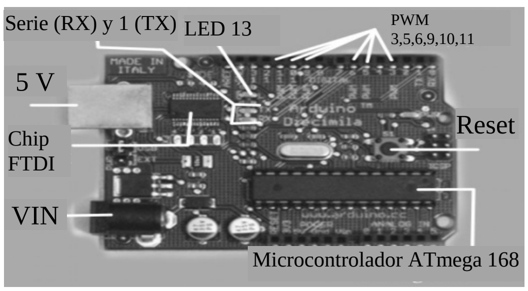


Fig. 3. Main components of Arduino.

Currently the modules X-bee from MaxStream allow serial links of TTL signals on distances of 30m indoors, 100m outdoors with line of sight and over to 1.5Km with Pro modules [9], which would be enough to control the electronic wheelchair, the final goal.
X-bee modules use the protocol known as Zigbee. Zigbee is a specification for a suite of high level communication protocols using small, low-power digital radios based on the IEEE 802.15.4 standard for WPAN's. In turn, IEEE 802.15.4 specifies the physical layer and media access control for low-rate wireless personal area networks (LR-WPANs). Zigbee operates in the band: ISM (Industrial Scientific and Medical), 868 MHz in Europe, 915 MHz in the U.S. and Australia, and 2.4 GHz in most jurisdictions around the world [10]. This technology is simpler, cheaper and has lower power consumption than other WPANs such as Bluetooth.

The Zigbee application areas are [11]:

- Home entertainment and control: Intelligent lighting, advanced temperature control, security, movies and music.
- Forewarned home: Water, power and access sensors, smart appliances.
- Mobile services: Mobile payments, control and monitoring, security, access control, health care and telecare.
- Commercial buildings: Energy monitoring, HVAC, lighting and access control.
- Industrial plants: Process control, management of benefits, environmental and energy and control of industrial devices.

In a Zigbee network there are three different types of nodes [12]:

- ZC (Zigbee Coordinator): There should be a network, its functions include the control of the network and the paths to be followed by devices to connect to each other.
- ZR (Zigbee Router): It interfaces separated devices in the topology of the network, besides providing a level of implementation for the execution of user code.
- ZED (Zigbee End Device): It has the functionality required to communicate with its parent node (ZC or ZR) but it cannot transmit information to other devices. Thus, this type of node can be asleep most of the time, increasing the average life of its batteries. A ZED has minimal memory requirements and is therefore significantly cheaper.

Based on the previous, an Arduino requires the implementation of the X-bee module to communicate wirelessly with other Arduino.
Figure 4 is a diagram representing the use of the Arduino and the X-bee. This study used two X-bee modules to create a wireless communication. It was firstly logged on a computer's parallel port to Arduino + X-bee which was the sender of data, and on the other side it had another Arduino + X-bee placed easily thanks to its small size and weight in a robot designed with lego, which received the data.

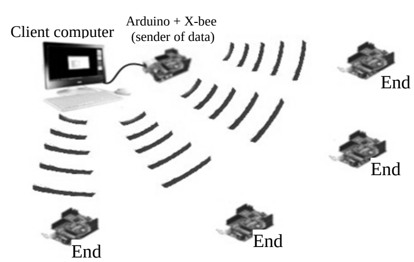


Fig. 4. Wireless communication, Arduino + X-bee.

3. Design

This section presents the architecture of the system and the description of the main components.

3.1 Architecture and System Description

Figure 5 shows the architecture of the designed system and then it is described each of its components.

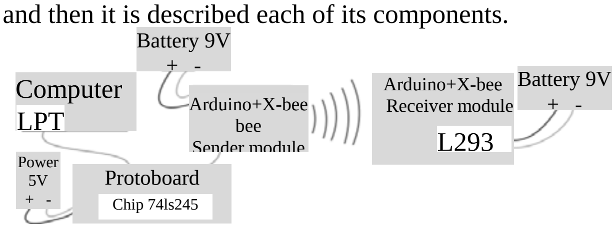


Fig. 5. System architecture.

Computer: It was used to program the robot; programming language Dev-C++ was used to program the movements, by means of the appropriate libraries to control the motors in order to initialize the LPT or the parallel port or parallel, among other functions. It was also used to make the data sender module and the module receiving data for the Arduinos, both in the Arduino programming language-0016, which is based on the functions of the standard C and some C++ functions.

LPT: The parallel port of the computer was used to connect it with the peripheral device Arduino. The main feature of LPT is that the data bits travel together, sending an 8-bits package at the time; i.e. a cable or another physic mean is implemented for each bit of data to form a bus. We can exemplify that controlling peripheral devices such as lights, motors and other devices suitable for automation by using the parallel port. The port consists of 4 control lines, 5 status lines and 8 data lines (8-bit bus) [12], (figure 6).

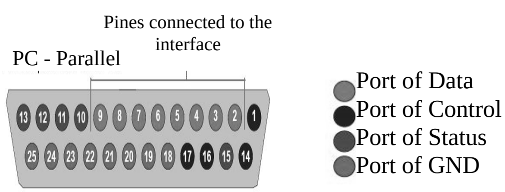


Fig. 6. Parallel port.

74ls245: This small chip attached to a protoboard, was very useful to access directly to the parallel port of the computer; three pin of data were connected to the parallel port (Fig. 6) with the 74ls245 chip pins (A1, A2, A). This same chip helped us to connect the 3, 5 and 6 PWM pins (figure 3) of the Arduino to the 74ls245 pins B1, B2 and B3 (figure 7). This chip used a power source to fuel of 5 volts.

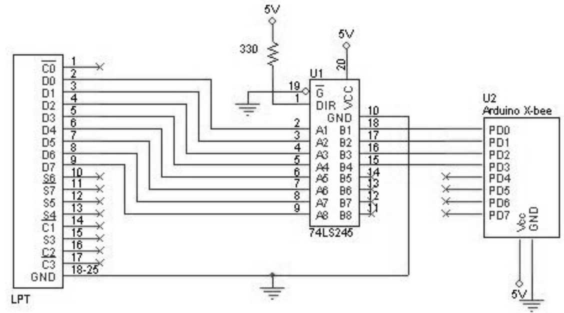


Fig. 7. Schematic diagram sender module.

In figure 8 we can see positive and negative pins for energy.

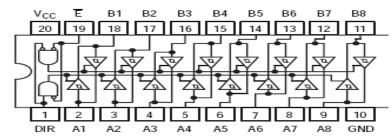


Fig. 8. Schematic diagram of 74ls245.

Controller board with I/O: There were used two Diecimila Arduino boards that have digital I/O to send to the servomotors of the robot PWM control signals. This sending of signals was achieved by using the connections previously performed (PC-LPT, LPT-74ls245, 74ls245-Arduino + X-bee); this board also has a Zigbee module that allows wireless communication with the computer [13]. See figure 9.

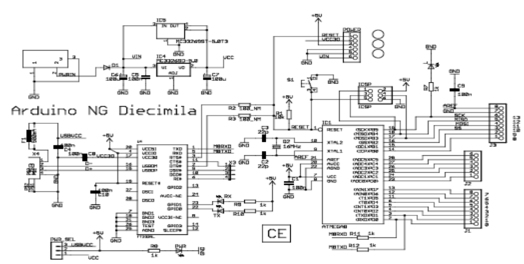


Fig. 9. Schematic Arduino Diecimila ATmega 168.

Wireless Modem X-bee or Zigbee: This modem is inserted into the two Arduino boards, which control the robot's movements and are responsible for sending commands to the servomotors [10].

To send motion commands to the servomotor, the following steps where done:

1. The user presses a key movement in the computer which is connected through a parallel port to the controller board (Arduino Diecimila + X-bee) who receives a series of bits.
2. The Diecimila + X-bee Arduino receives this series of bits, which is interpreted by the microcontroller that has a program loaded for this function [8].

3. Once the information is interpreted by the Arduino microcontroller, this is sent wirelessly to the other controller board located on the robot (Zigbee) [10,15].
4. The robot controller board interprets this information, similar to step number 2, as this module is programmed to receive data sent from the computer.
5. Now that information has been interpreted, the movement of the servo is carried out according to the key- action or movement, which the user has selected. Then, the controller board sends the bits through a communication cable connected to the servomotors and they perform the action needed.

L293B: L293B chip is a driver designed to provide bidirectional drives current up to 1A at voltages between 4.5 and 36V with a maximum dissipation power of 5W. It allows us to control the robot and motors, which can be done easily with multiple chips such as ULN2003A, L293B, L293C, ULN2803A, among others. This chip is installed on a printed circuit board, mounted on the robot where the data cables were connected (corresponding to the parallel port) in L293B chip pins. For the movement of the servomotors there were used communication cables connected to the servomotors on pins 3, 6 (servomotor1) and 11, 14 (servomotor 2) of the L293B (figure 10 and 11). [15].

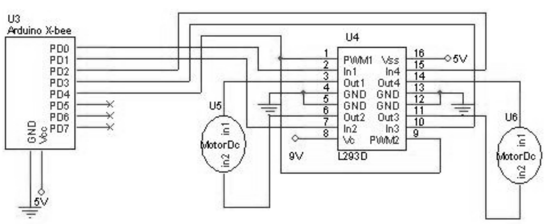
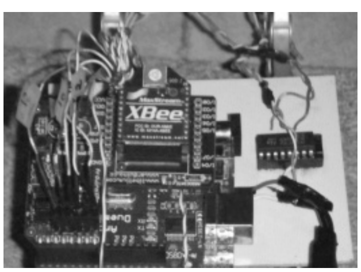


Fig. 10. Schematic diagram receiver module.



|Fig. 11. Connecting cables, parallel port and L293B.

Servomotor: An actuator device that has the ability to be positioned anywhere within its operating range, and remain stable in that position. The main component of a servomotor is a DC motor, which performs the function of actuator in the device. When a voltage is applied between its two terminals, the motor rotates in one direction at high speed, but producing an under par. To increase the torque of the device a gearbox is used, which transforms much of the speed of rotation in torsion (figure 12). Two servo motors were used in the model of the robot; they receive information for movement (forward, backward, left, right) through the controller board (Arduino Diecimila with X-bee module connected to the L293B), [16].

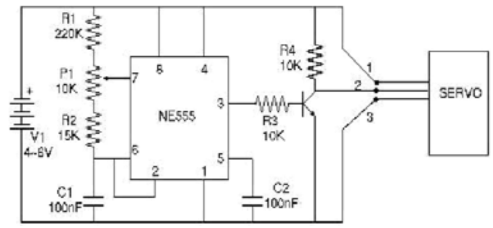


Fig. 12. Servomotor.

4. Implementation

Next, the hardware used and how it was implemented are described in detail.
There were located the data record cables [illegible] the parallel port with the help of a multimeter, and the plastic was removed, leaving a small tip of copper exposed in order to make the connection to the chip 74ls245 (figure 13).

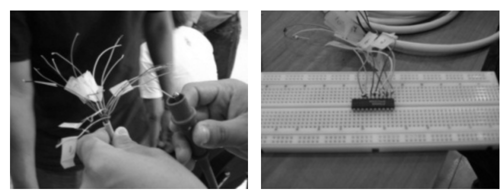


Fig. 13. LPT Data Cable and its connection to the chip 74ls245.

Once connected the data cables, the robot was programmed using programming language Dev-C ++. The movements performed were left, right, forward, backward, pause and exit, by making coincide the keys in Table 1.

Table 1. List of movements.

Keys	***Action***
↑	Forward.
↓	Back.
←	Left.
→	Right.
↵	Pause.
c	Exit.

In the following code, as mentioned previously was developed in Dev-C ++, we can see the programming of some of these movements.

```
do{
```

```
op2=getch();
 switch(op2)
 {
  case 72 :   (oup32)(0x378,1);
  Sleep(100);
  (oup32)(0x378,tmp_CToff);
  break;
  case 80 :   (oup32)(0x378,2);
  Sleep(100);
  (oup32)(0x378,tmp_CToff);
  break;
}
```

The execution of the previous code could be tested using the LPT monitor of the key pressed, which is used to view the status of the parallel port (figure 14).

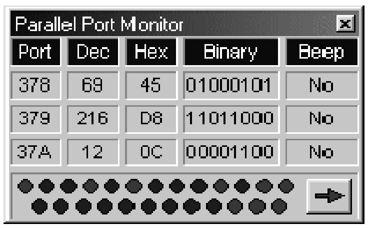


Fig.14. Parallel Port Monitor (LPT Monitor).

Testing that the programming of the Arduino was done, which consists of two modules: A module to connect the parallel port and the computer (Sender Module), and a module that will be set on the Arduino-L293B robot (Receiver Module). In figure 15 we can see the Arduino programming environment.

Fig. 15. Arduino-0016, programming environment.

In this section an extract of the code of the data transmission function is presented.

```
void setup()
{Serial.begin(9600);
  pinMode(13,INPUT);
  pinMode(12,INPUT);
  pinMode(11,INPUT);
  pinMode(6,INPUT);
  pinMode(5,INPUT);
  pinMode(4,INPUT);
  pinMode(3,INPUT);
  pinMode(2,INPUT);}
void loop(){delay(300);
bit1=digitalRead(2);
bit2=digitalRead(3);
bit3=digitalRead(4);
if(bit1==0)
{    if(bit2==0)
  {      if(bit3==0)
       {     Serial.write('/');   }
       else
       {     Serial.write('+');   } }
 else
  {  Serial.write(')');  } }
   else
  {    if(bit2==0)
    {   Serial.write('(');  }
     else
    {   Serial.write('*');  }}}
```

Once the programming of the transmission of data is done, the program was uploaded to the Arduino's microcontroller with a programmable device of microcontrollers. This was done without implementing the X-bee module. After the programs were uploaded to the microcontroller, the X-bee module was set in the Arduino; this fact was previously presented in figure 2. It was used the HyperTerminal program in Windows to configure the X-bee module and verify the data sending. Once the connections and the first tests were made, we got a design as the one shown in figure 16. The design considering the robot can be seen in figure 17.

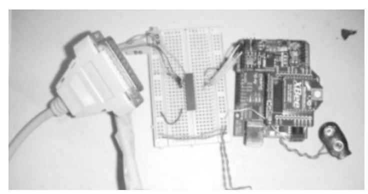


Fig. 16. LPT connection with Arduino + X-bee.

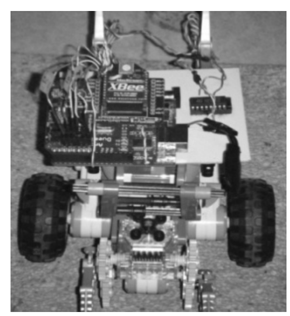


Fig. 17. L293B connection with Arduino + X-bee.

5. Conclusions

A robot was developed in this paper, which was controlled through the parallel port of the computer with an implementation of Arduino + X-bee, resulting that the robot could be guided with a certain freedom and autonomy without using wires; this robot can be used to enter to confined spaces when it is needed, for example.

Arduino Diecimila currently has many applications; it is an easy-to-use and program device, as can be seen in our implementation. The microcontroller is like a small computer which can be very useful for various applications; by mounting the Zigbee allowed to communicate with other Arduinos wirelessly.

In future work the improvements for this robot would be the implementation of a camera to be introduced to places where human beings could not get and at the same time without running any risk by not knowing the environment (for example in the case of an earthquake).

Finally, in the immediate future this development will be implemented in an electronic wheelchair, as mentioned in a parallel study to the present.

References

[1] http://electrolabo.com, http://www.ladyada.net, http://www.Arduino.cc, http://www.instructables.com. Ultima consulta Junio 30 de 2010.

[2] Alonso Álvarez, Jaime. Desarrollo de un robot móvil controlado desde Internet: Comunicaciones Wi-Fi. Universidad Pontificia Comillas – ICAI. Madrid. Junio 2005.

[3] Barro Torres, Santiago; Escudero Cascón, Carlos. Embedded Sensor System for Early Pathology Detection in Building Construction. Department of Electronics and Systems, University of A Coruña. España.

[4] Boonsawat, Vongsagon, et all. X-bee Wireless Sensor Networks for Temperature Monitoring. School of Information, Computer, and Communication Technology Sirindhorn International Institute of Technology, Thammasat University, Pathum-Thani. Thailand.

[5] Díaz del Dedo, Luis; Pérez García, Luis. Diseno, Construccion y Desarrollo de la Mecanica, Electronica y Software de Control del Robot Cuadrupedo R4P v3i. Universidad Europea de Madrid.Villaviciosa de Odon, Madrid. Septiembre de 2008.

[6] Nadales Real, Christian. Control de un Quadrotor mediante la plataforma Arduino. Escuela Politécnica Superior de Castelldefels. Julio de 2009.

[7] Baik, Kyungjae. BoRam: Balancing Robot using Arduino and Lego. September, 2008.

[8] Información general del Arduino. http://Arduino.cc/es/Main/ArduinoBoardDiecimila Última consulta Julio 2 de 2010.

[9] http://www.digi.com/products/wireless/point-multipoint/X-bee-series1-module.jsp#specs Última consulta Julio 2 de 2010.

[10]Zigbee standards Organization “Zigbee 2007” http://www.zigbee.org/Products/TechnicalDocumentsdownload/tabid/237/default.aspx Última consulta Junio 5 de 2010.

[11]X-bee primeros pasos http://www.scribd.com/doc/13069890/Primeros-Pasos-Con-Arduino-y-X-bee Última consulta Mayo 10 de 2010.

[12]http://mundobyte.files.wordpress.com/2007/12/lpt.jpg Última consulta Mayo 12 de 2010.

[13]http://www.atmel.com/dyn/resources/prod_documents/doc2545.pdf Última consulta Abril 20 de 2010.

[14]IEEE 802.15.4-2003 “IEEE standard for local and Metropolitan Area Networks: Specifications for Low-Rate Wirless Personal Area Networks”. 2003

[15]http://www.datasheetcatalog.net/es/datasheets_pdf/L/2/9/3/L293.shtml Última consulta Abril 15 de 2010.

[16]http://cfievalladolid2.net/tecno/cyr_01/robotica/sistema/motores_servo.htm Última consulta Abril 3 de 2010.

Christian Hernández Pool. Currently studying the seventh semester in computer science at the Autonomous University of Yucatan (UADY). He participated in the project fair held in the institution as well as software development projects.

Raciel Omar Poot Jiménez. Received a degree in Computer Engineer from the Autonomous University of Yucatan (UADY) in 2008. He has been a full time teacher at the Autonomous University of Yucatan since 2008 at the department of electronics in Tizimín México. He has participated in development projects of electronic engineering. Currently is giving courses on electronic interfaces and programming languages in the professional programs at the UADY.

Lizzie Edmea Narváez-Díaz. Received a degree in Computer Science from the the Autonomous University of Yucatán (UADY) in 1997. She received a Master of Computer Science degree from Monterrey Technological Institute (ITESM), Campus Cuernavaca, in 2007. She has been a full time teacher at the Autonomous University of Yucatán since 2000 in the Networking department in Tizimín México. She has participated in software engineering development projects. Currently she is giving courses on wireless networks in the professional programs in the UADY.

Erika Rossana Llanes-Castro. Received a degree in Computer Science from the Autonomous University of Yucatan (UADY) in 2002. She is about to receive a Computer Science degree from Monterrey Technological Institute (ITESM), Campus Estado de Mexico. Currently, she is a full time academic technician at the Autonomous University of Yucatan since 2002 in the department of computer in Tizimín México. She has participated in software engineering development projects. Currently is giving courses on programming mobile devices in the professional programs in the UADY.

Victor Manuel Chi-Pech. Obtained his degree in Computer Science from the Autonomous University of Yucatan (UADY) in 1996 and his M. Sc. degree in Wireless Network from Monterrey Technological Institute (ITESM), Campus Cuernavaca, in 2007. Victor Chi has worked since 2000 at the Autonomous University of Yucatan, as a full time professor. He has participated in software engineering development projects. He is currently giving courses on wireless networks and software engineering in the professional programs in the UADY.