Design and implementation of applications to control a Lego NXT robot via Bluetooth from a Pocket PC

Erik Coral, Michel García, Cinhtia González, Sergio González, Carlos Miranda, Teresita Montañez

Universidad Autónoma de Yucatán, Facultad de Matemáticas, Unidad Multidisciplinaria Tizimín

Summary
This work presents the design and implementation of two applications to control a Lego NXT robot, one of them running on the robot itself and the other on a pocket PC. The application for the robot was developed in the NXC language, while the Pocket PC application was developed in Microsoft Visual Basic. The final part of this paper shows the results of the tests, which were implemented in both applications, demonstrating their speed and efficiency.
Key words:
Bluetooth, Lego NXT, pocket PC, mobile robot.

1. Introduction

Lego Mindstorm NXT robots are built from small pieces of Lego building blocks and controlled by a small brain called Intelligent Brick [10]. These robots can be used to teach concepts of robotics with the basic educational set and educational software, sold separately.
Lego NXT Robots have servo motors to simulate a wide range of motion of machines, animals and even humans; from a displacement of the robot by rotating the wheels that support it, to a system of gears and levers that can mimic the movements of a human arm. They also have sensors to perceive the surrounding environment, including the basic sensors of touch, light, ultrasonic and sound, although there is also color sensor, compass, temperature sensor, among others.
One of the most important characteristics of Lego NXT robots is the ability to connect to other devices supported by Bluetooth. This last feature allows the robot to communicate with other robots in their environment by sending or receiving orders. It is even possible to use their Bluetooth connection to send commands from other kind of devices and thus control the robot remotely.
Among the projects built using Lego NXT robots are: a robot able to maintain balance with only two wheels, described by Sanchez [1], an algorithm for exploration in controlled environments using a Lego NXT robot, described by Gomez and Garcia [2], a robot guided by vision with basic operations of localization and mapping in a controlled environment using Matlab, as described by Gómez [3], a mining robot programmed with Matlab and Simulink, described by Cabrera [4], to mention some of the many projects carried out with these robots.
This article describes how to use the Bluetooth connection to control an NXT robot remotely, using a pocket PC. Section II deals with some of the different languages to develop applications on mobile devices and explains why the language used in this project was chosen. Section III focuses on the brief description of the programming languages used in the project and the tests performed to the applications. Section IV explains how the Pocket PC gives orders to the robot remotely through direct and indirect commands. Also, Section V and Section VI describe in more detail the applications of the pocket PC and the Lego NXT robot respectively. And finally, in Section VII the result of the tests are presented, and in Section VIII the conclusions drawn from the results, as well as future work for the project.

2. Scheduling in mobile devices

With the emergence of mobile devices also reached new development platforms for such applications, of which there is now a great variety and in this section some of the most used are described.
Among the most popular development language is Java, which is used in the Android operating system as the main application development language [5].
On the other hand, Java ME is a collection of Java APIs used for developing applications and games in limited specifications devices [6]. There is also Python for Symbian, an adaptation of Nokia of the Python language for developing applications and games for their Symbian devices [7]. C++ is also used to develop applications for Symbian and Windows Mobile devices. Similarly, C# is used for development on Windows Mobile as well as Microsoft's Visual Basic language, which is the language used to implement the project described in this paper.
The Visual Basic language was chosen mainly because Python for Symbian and Java ME were previously discarded as development languages for this project, because the devices that supposedly supported those platforms proved to be inconsistent with the Bluetooth

Manuscript received June 5, 2011
Manuscript revised June 20, 2011

specification of the NXT brick, making communication between them impossible.
The chosen device was a Pocket PC running Microsoft Windows Mobile 2003 and with its Bluetooth specification fully compatible with the NXT brick, making it the perfect testing device for this project..

3. Methodology

To develop the application of the pocket PC was used a programming language called "event driven Visual Basic", which facilitates the development of GUI applications to users [9]. On the other hand to develop the application of the robot it was used a structured programming language called NXC, very similar to the C language, because it provides functions that facilitate the control of motors and sensors of Lego NXT robots [8].
The tests performed to the applications were done in order to demonstrate the effectiveness and speed of the program, as well as their usability. The tests were:

1. To control the NXT Lego robot to steer through an environment with obstacles toward an object to be achieved, take it and return the robot with the object back to the starting position.
2. Park the robot between two objects with a space scarcely larger than the length of the robot.

As test subject it was chosen a Tribot robot, which consists of 3 engines, two for the wheels and one for the tweezers. It also has 4 sensors: a touch sensor, a sound sensor, a light sensor and an ultrasonic sensor.

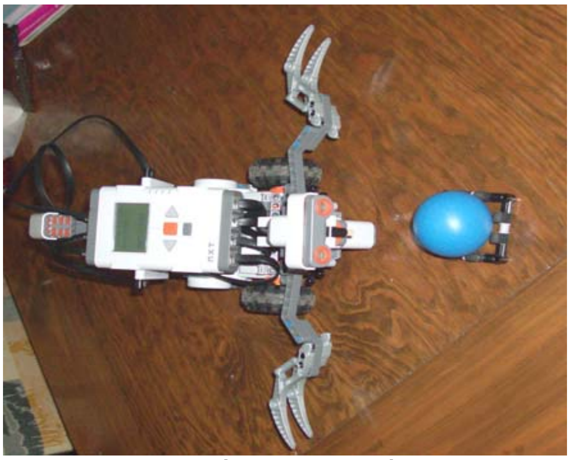

Fig 1 Robot LEGO NXT Tribot

The pocket PC used for testing is the O2 XDA IIs, whose specifications are:

- CPU Intel PXA263 32-bit
- Clock Speed: 400MHz
- RAM: 128MB

4. Development

Two applications are necessary for the proper control of the Lego NXT robot in this project, the first lies in the pocket PC and the other in the robot itself.
The pocket PC application uses a Bluetooth connection with the robot to send commands of what it must do. Each one of these orders is sent, depending on the action taken by the user on the interface.
Indirect commands are used, i.e. human language orders sent to the robot and translated by the resident application in the same robot, as well as direct commands, i.e. orders sent to the robot that are directly processed by the intelligent brick without necessity of a running application.
Indirect commands are used in this project to control the movements of the robot; an example of these commands is "IZQ_FORWARD3", which when the application in the robot receives, interprets as moving the motor connected to port C of brick with a power of 75 %.
Direct commands are used to obtain the values of sensor readings of the robot. These readings are displayed on the screen of Pocket PC for the user to know them whenever he wants. Direct commands are arrays of bytes whose length depends of the command itself. For example, the direct command to apply the battery level in millivolts Lego NXT robot has a length of 4 bytes while the command to configure a sensor consists of 7 bytes.

5. Application for the pocket pc

The application for the Pocket PC that was developed in Visual Basic language is called "NXT Move!" and shows a simple interface that contains controls for basic robot actions, as well as more complex actions. Among the basic controls are the directional buttons that allow the robot to move forward, backward or turn. And among the more complex controls is the balance beam or "steering", that as being located in the middle allows the robot to go straight, but when you drag the bar to one side the robot will rotate on a curve towards that side, and the more you drag the side bar more tighter is the turn, this control works like a steering wheel of a vehicle.
The application, "NXT Move!", also gives the user the option to activate a feature called "Sensory Assistance" which, using the ultrasonic sensor, avoids the robot to collision with an object that is too near as a wall, depending on where ultrasonic sensor is pointing.
Correspondingly, there is a button that allows the user to check the readings of some sensors on the robot, among which are the touch sensor, sound sensor and light sensor. These elements of the application interface and some others not mentioned can be seen in Figure 2.

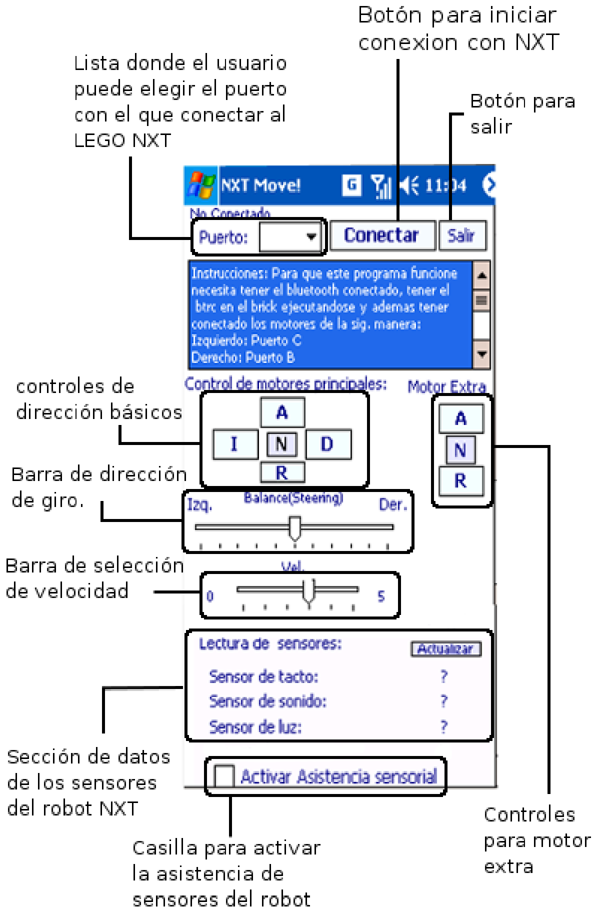

Fig. 2. Graphic environment of the NXT Move!.

Below is a block diagram illustrating the structure of the application, “NXT Move!”, and a brief explanation of each block.

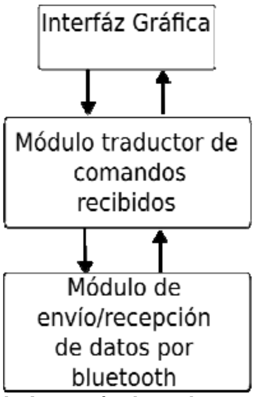

Fig. 3. Block diagram for the Pocket PC application.

The graphical interface is part of the application that the user sees and with which interacts. This contains the commands to start the connection as well as to control the robot.

The translator module of received commands is responsible for translating user commands received through the graphical interface, into orders sent to the robot that it can understand. It also translates the commands or information received from the robot via Bluetooth to information that the user can understand and then displays it on the interface’s screen.

The module sending / receiving data via Bluetooth is used, as its name implies, to send commands to the robot and to receive information that the robot sends, such as readings from its sensors.

And finally the Bluetooth module is responsible for managing the Bluetooth connection since the moment you start the connection, when sending or receiving data until the connection is closed.

6. The application for the LEGO NXT robot

The task of the application on the Lego NXT robot is to interpret commands received from the pocket PC and make the robot to perform the corresponding action. It also shows some data on the display of Lego NXT robot, this data is the status of each motor, for example "FWDx" means the engine is moving at speed x, "BCKx" means that the backward speed is x or "STOP" for the engine off, it also indicates if the sensory assistance is on or off and if an object is too close to the ultrasonic sensor.

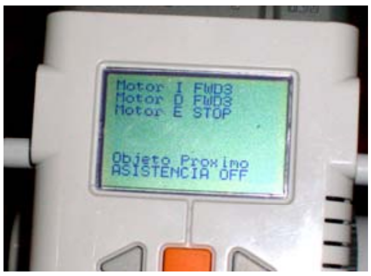

Fig. 4. Running NXT Program

Below is a diagram illustrating the operation of the application running on the NXT intelligent brick.

Fig. 5. NXT application running and displaying user data.

The data reception module by Bluetooth handles, as its name indicates, of receiving data from the pocket PC through a Bluetooth connection; these data are the orders that the application must interpret.
The module that translates received messages and executes accord actions handles interpreting the orders received from the pocket PC and indicating the robot the appropriate action to take.
The module that displays information to the user from the screen is the module that prints messages with information about the status of motors and other features on the screen of the NXT intelligent brick.

7. Test

We applied two different experiments to test their applications quickly and effectively in reality. Each experiment was repeated a total of 30 times and recorded the results of each repetition. The tests with their respective results are:

Test 1

The applications are used to maneuver the robot through a path with obstacles until reach the objective, taking it with the robot gripper and maneuver the robot on the same path back to its original position. The road has a length of 3 meters and the travel time is measured.

Results:

In the first instance we found that the best time is 39 seconds, the worst time is 86 seconds (1 minute and 26 seconds) and the average time is 55.7 seconds.
By analyzing all the results is obtained:

- A total of 6 tests have a result within the time interval from 20 to 45 seconds. This represents 20% of tests.
- 19 tests have results of between 46 seconds and 1 minute 10 seconds, representing 63.33% of tests.
- Y 5 tests have a time longer than 1 minute 10 seconds, representing 16.67% of the tests.

Thus can be noted that most tests are achieved within the time interval between 46 seconds and 1 minute 10 seconds, and in the minority part of the evidence are those achieved in a time longer than 1 minute and 10 seconds.

Test 2

The applications are used to maneuver the robot and get to park in a space 5 cm longer than the robot (the clamps of the Tribot were removed because they were not needed in the experiment). The task must execute with the fewest possible moves.

Results:

Firstly in the results of the experiment can be seen that the best result was 3 movements for parking, while the worst was 6 movements. The average of the results of the experiment was 3.86 movements to park the robot.
By examining all the results is obtained:

- 12 of the 30 tests were achieved in 3 movements, which represents 40% of the tests.
- 16 of 30 tests were achieved with 4 or 5 movements, representing 53.33% of tests, i.e. slightly more than half.
- Only 2 of the 30 tests took more than 5 movements for parking the robot, representing only 6.67% of the tests.

Thus can be noticed that most of the tests were achieved with 4 or 5 movements, being very close to the number of tests that were achieved with only 3 movements, the difference is only 4 runs between them. And only 2 tests required more than 5 moves to complete.

8. Conclusions and future work

The program for the pocket PC works with minimal delay with the robot, causing the robot to respond to commands almost immediately.
The experimental results show that the maneuverability and speed of the robot controlled by applications is quite acceptable and generally serve their purpose applications with great efficiency and speed of response, in spite of the procedure for submission and processing of commands.
The project offers the user several options to control the robot such as basic management controls, a slider to adjust the speed in real time, a scroll bar to adjust the turn in real-time, extra engine controls, and even provides extra features such as a button to get the readings of some sensors and the option to prevent the robot colliding with an object that is too close to the ultrasonic sensor.
However, there are several future works, for example, to remove the application that resides on the Lego NXT robot, requiring only the application of the pocket PC to control the robot. It is also possible to expand the list of compatible sensors with others that are not yet supported by this project, such as the ultrasonic sensor, which is implemented in this project to prevent the robot shock but cannot get a reading yet, it is also possible to add the color sensor, among others.

References

[1] Sebastián Sánchez Prieto, Óscar Rodríguez Polo, Tomás Arribas Navarro. Utilización de Lego NXT en docencia universitaria. Universidad de Alcalá, España. Disponible en http://www.redusoi.org/docs/publicaciones/P6-Utilizacion%20-de%20LEGO%20NXT%20en%20docencia%20universitaria.pdf. Ultimo acceso el 12 de Diciembre de 2010.

[2] José Luis Gómez Ramos1, Federico García Bolaños2. Exploración de Ambientes Controlados con Un Robot Tipo Lego. División Académica de Informática y Sistemas Universidad Juárez Autónoma de Tabasco. Disponible en http://www.conais.com.mx/libroElectronico/2008/_4.pdf. Último acceso el 17/noviembre/2010.

[3] Mariana N. Ibarra Bonilla, Juan M. Ramírez Cortés, Alejandro Díaz Méndez, Jorge Martínez Carballido, Rogerio Enríquez-Caldera, Irma J. García Enríquez. Navegación autónoma de un robot guiado por visión con operaciones básicas de localización y mapeo en un ambiente controlado. Coordinación de Electrónica; Instituto Nacional de Astrofísica, Óptica y Electrónica; Tonantzintla, Puebla, México. Disponible en http://www-elec.inaoep.mx/~jmram/pub/navaut.pdf. Último acceso el 22/noviembre/2010.

[4] Patricia Cabrera, Ronald Chicaiza, Jorge Herbozo, Carlos Valdivieso. Uso de Matlab y Simulink para el control de robots y la observación de sensores de sonido y luz. Facultad de Ingeniería en Electricidad y Computación. Escuela Superior Politécnica del Litoral (ESPOL). Disponible en

[5] http://dspace.espol.edu.ec/bitstream/123456789/7844/3/Uso%20de%20MATLAB%20y%20Simulink%20para%20el%20control%20de%20robots.ps. Último acceso el 22/noviembre/2010.

[6] ANDROID [En línea],

[7] http://developer.android.com/guide/basics/what-is-android.html. Último acceso: 12 de diciembre de 2010.

[8] JAVA MICRO EDITION [En línea], http://www.oracle.com/technetwork/java/javame/index.html. Último acceso: 12 de diciembre de 2010.

[9] Python for symbian S60 [En línea], http://www.devshed.com/c/a/Python/Mobile-Programming-in-Python-using-PyS60-Getting-Started/. Ultimo acceso: 12 de diciembre de 2010.

[10] NXC [En línea], http://www.cs.ru.nl/lab/nxt/content.html. Ultimo acceso: 12 de diciembre de 2010.

[11] VISUAL BASIC [En línea],

[12] http://visualbasic.about.com/od/applications/a/whatisvb.htm. Último acceso: 12 de diciembre de 2010.

[13] LEGO NXT [En línea], http://mindstorms.lego.com/en-us/whatisnxt/default.aspx. Último acceso: 12 de diciembre de 2010.

MC. Cinhtia Gonzalez Segura. Master in Computer Science for the Institute Technology of Monterrey in México, is professor of the Autonomous University of Yucatán. Responsible of the Intelligents Systems lab. His researcher lines: Optimization, Artificial Intelligence, Mobile Robots

Michel García. MC in Computer Science for the Institute Technology of Monterrey in México, is professor of the Autonomous University of Yucatán, Actually collaborate in the Intelligent Systems Lab. His researcher lines: Machine Learning, Artificial Intelligence, Mobile Robots.

Erik Alejandro Patrón Coral is currently the seventh semester student in the Bachelor of Computer Science, Faculty of Mathematics at the Multidisciplinary Unit Tizimín. Has been involved in software development, robotics and electronics.

Carlos Miranda Palma. Master in Computer Science for the Institute Technology of Monterrey in México, is professor of the Autonomous University of Yucatán, Is coordinator of the Degree in Computer Science, Faculty of Mathematics at the Multidisciplinary Unit Tizimín. His researcher lines: Voice recognition, Artificial Intelligence, Human Computer Interaction.

Sergio Gonzalez Segura. Master in Computer Science for the Centre for Research and Technological Development. Is professor of the Autonomous University of Yucatán and responsible of the electronic laboratory. His researches lines: Intelligent systems, stereo vision, robotics.

Teresita del Jesús Montañez May. Degree in mathematics for the Autonomous University of Yucatán. He was publicated several works related with Lego NXT robots applied in teaching mathematics, particularly calculus. His researcher lines: numerical algorithms, calculus, applied matemathics.