DEVELOPING AND IMPLEMENTING A SYSTEM TO USE A MOBILE DEVICE AS VIDEOGAME CONTROLLER USING BLUETOOTH LOW ENERGY

Estefanía González[1], Pablo Corral[1], Saulo Frões[2], A.C.D.C. Lima[2], Eliton Pires[3]
[1] Universidad Miguel Hernández de Elche / Departamento de Ingeniería de Comunicaciones / pcorral@umh.es
[2] Universidade Federal de Bahia / Departamento de Engenharia Elétrica / acdcl@ufba.br
[3] Instituto Federal Catarinense / Santa Rosa do Sul (Santa Catarina) / eliton.pires@santarosa.ifc.edu.br

Resumo: *The aim of this project is to provide a mobile device with the necessary software and hardware tools in order to be able to be used as a wireless controller for PC gaming. In order to achieve this, we will take advantage of the Bluetooth Low Energy technology, which allows us to share information between an Android Smartphone and an Arduino Leonardo with an HM-11 BLE module connected, and send the commands the user desires to a computer.*

Palavras-Chave: *Bluetooth Low Energy, Arduino, Android, Wireless Communications*

1 INTRODUCTION

The videogames industry currently represents one strong economic sectors. The market value is worth in billions euros and growing exponentially, especially in the latest years. It Is a very changing, demanding and profit market, which needs new products to continue evolving.

Many companies have been enhancing the videogames players experience through improving the peripheral devices used to play. If we compare the first consoles commands with the new models, we can appreciate a significant change produced by advances in consumer electronics.

New advances in telecommunications permit higher transmission bandwidth, new technologies like wireless communications make possible information transmission using more sophisticated devices.

In this project, we are going to use bluetooth technology to interconnect various devices using wireless communication. Control data can be sent from control devices to computers using radiofrequency links.

Bluetooth features like low latency and power consumption as well as coverage range make this technology suitable for applications like developing devices commands to control computer videogames.

2 PROJECT DESCRIPTION

The project aim is to build a system which will be connected to a computer and will control videogames through commands received from a mobile device using Bluetooth.To do it, we are going to use an Arduino Leonard board with a HM-II bluetooth

low energy module which will allow bluetooth connectivity between one mobile device and our implemented system.

The Arduino board will act as a bridge between a smartphone and a computer, it will receive the control data commands sent from the mobile device and will interpret them.

3 COMPONENTS

3.1 Arduino Leonardo board

The Arduino Leonard board is the main project element, its function is receiving information and translate it to commands to the attached computer.

The main board element is an Atmega32u4 microcontroller, unlike other board as Arduino UNO, Arduino Leonardo does not need a second microprocessor because it has an USB controller integrated on board.

The Arduino Leonardo will be connected by the USB output port to the computer on which we want to work. The microcontroller of the board has a circuit inside called an analog-to-digital converter or ADC that reads this changing voltage and converts it to a number between 0 and 1023. So the resolution is:

$$\text{Resolution} = \frac{5\,V}{1024} = 4.88\,mV \qquad (1)$$

3.2 HM-11

The HM-11 is a bluetooth low energy module which can be connected to Arduino board to allow Wireless communication.

Figura 01 – HM -11 device

Source: Seeedstudio

This Bluetooth module has a pre-built firmware that accept AT commands configuration. The next table shows the HM-11 BLE module pinout:

Table 01 – HM-11 pinout

Pin	Name	Description
1	UART-RTS	UART
2	UART-TX	UART data sending
3	UART-CTS	UART
4	UART-RX	UART data reception
5,6,7,8,10	NC/NV	
9	VCC	Power 3.3V
11	RESETB	RESET
12	GND	GROUND
13	PIO3	IO Port
14	PIO2	Digital input/output
15	PIO1	LED indicator
16	PIO0	Pin button

Source: Seeedstudio

In this project we are only using pin 2,4,9 and 12 from HM-11 module to interconnect it to Arduino Leonardo.

3.3 Mobile device

It is necessary to use a mobile device with bluetooth 4.0 connectivity to send commands to HM-11 module, such as:

Iphone (model 4s or higher)
Ipad (third generation or higher)
Smartphones or Android Tablet (4.3 or higher version)

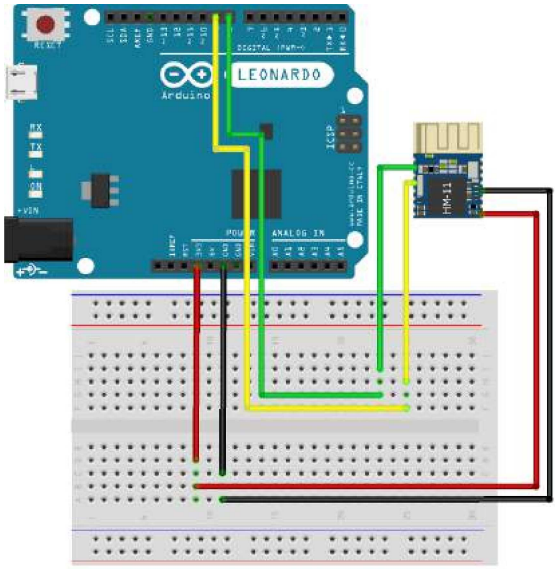
The main advantage of using BLE technology in mobile devices is consumption power saving compared with previous bluetooth versions, which is a very important feature to take into account because it will permit a longer duration time for end users.

Once selected the mobile device, we must install a graphical interface application used to send the different buttons commands to the HM-11 module using BLE. We have decide to develop an specific application to have a better control of commands to send using Bluetooth as well as making a customization according to requirements.

4 DEVELOPMENT

First of all, we need to interconnect the Arduino Leonard board and the HM-11 module, as it is shown in the figure below:

Figure 02 – Assembly schema



Source: Fritzing

The HM-11 TX pin must be connected to RX Arduino pin to receiv information sent by the Bluetooth module, similarly, the HM-11 RX pin should be connected to TX Arduino. The latest connection is not strictly necessary because the communication is unidirectional (from mobile device to Arduino). However, we have decided to connect them to be able to receive AT commands from Arduino to configure the HM-11 module. Finally, we connect the Arduino board to the computer through a micro USB cable and the assembly is done.

The wireless communications is the most important part of the Project. We must be able to transmit information from the mobile device to the HM-11 BLE module.

We should enable the Bluetooth network from the mobile device and run the developed application.

In this project we have used an Android 6.0.1 device, but we could also have used any IOS device with BLE connectivity.

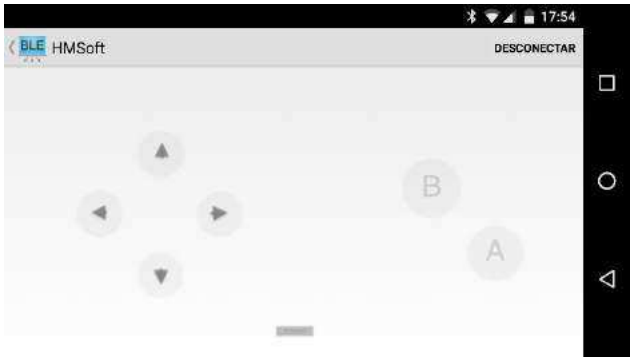
The application developed has been designed to search for any wireless device using BLE in the operational range and pair with one of them. Once paired, it permits sending commands to the HM-11 module and see the commands sent throught Arduino serial monitor.

Figure 03 – BLE devices found.

Source: BLE app

The application is able to send encoded keys from the mobile device to Arduino board, which will translate them to commands keyboard that will be sent to the computer. The application sends a code when pressing a button in the mobile device and another code when releasing it, this allows sending the command “button pressed” from Arduino board to the computer until the user release the key.

Figure 04 – Command Graphical Interface



Source: BLE app

The following figure is a description of connection phases between the smartphone application and the HM-11 module. The application graphic interface is simple, it has four direction arrow buttons, “A”, “B” and “Start” buttons which are intended to drive basic computer videogames. The application also has a disconnect button to stop the communication with Arduino any time.

Finally, we need to develop a program to translate all information received through HM-11 module to commands know by computers.

Arduino Leonard board can emulate a computer keyboard, using key pressing to control computer games.

4.1 Sketch code

Libraries referring to the software serial ports and keyboard control are included at the top of the sketch. There are necessary for the proper functioning of the program actions such as reading data from the Bluetooth module or pressing a keyboard key.

After the “include” section, we define the serial ports in which we have physically connected the HM-11 BLE module: 8 is the reception port and 9 is the transmission one. The string “tmp” is also defined and it will be used to associate the value the program reads from the serial port.

In the setup function, all the components used by the program are initialized.

```
#include <SoftwareSerial.h>
#include <Keyboard.h>

SoftwareSerial mySerial(8, 9); //RX,TX
String tmp;

void setup() {
  Serial.begin(9600);
  //Activar teclado y puertos serie
  Keyboard.begin();
  mySerial.begin(9600);
  mySerial.write("AT");
};
```

Finally, the loop will be programmed, which is the part of the code that will be repeated while the Arduino is on. While the defined serial ports are available, the device keeps reading and storing the data that receives into the String "tmp".

```
void loop() {
  //Leer puerto serie
  while (mySerial.available() > 0)  {
    char inChar = mySerial.read();
    tmp += inChar;
    delay(2);
  }
  tmp.trim();
  //
```

Using an “if” loop, when the “tmp” string contains something, it means that the Arduino has read information from the Bluetooth module, so the program must compare the numerical code that has received to the order or action that has related.

The relationship between the Android and the Arduino code is very important in this step, as the numerical value associated with each key should be the same in both codes so that the device works properly.

```
if(tmp.length() > 0) {
int numero = tmp.toInt();
  switch (numero) {
    case 1:
    Keyboard.press(KEY_UP_ARROW);
    break;
    case 2:
    Keyboard.release(KEY_UP_ARROW);
    break;
    case 3:
    Keyboard.press(KEY_DOWN_ARROW);
    break;
    case 4:
    Keyboard.release(KEY_DOWN_ARROW);
    break;
    case 5:
```

Cases 1 and 2 are linked to the gesture of pressing or releasing the up-arrow key. Analogously, cases 3 and 4 relate to the down-arrow key.

```
case 9: //Tecla START
Keyboard.press(KEY_RETURN);
delay(50);
Keyboard.release(KEY_RETURN);
break;
case 10: //Tecla B
Keyboard.press('k');
break;
case 11: //Tecla A
Keyboard.press('l');
case 12:
Keyboard.release('k');
break;
case 13:
Keyboard.release('l');
break;
```

When pressing the gamepad buttons, the program functions work exactly the same.

```
    case 20:
    Keyboard.press(KEY_RIGHT_ARROW);
    delay(10);
    Keyboard.press(KEY_DOWN_ARROW);
    break;
    case 21:
    Keyboard.release(KEY_RIGHT_ARROW);
    delay(10);
    Keyboard.release(KEY_DOWN_ARROW);
    break;
  }
  tmp = "";
}
```

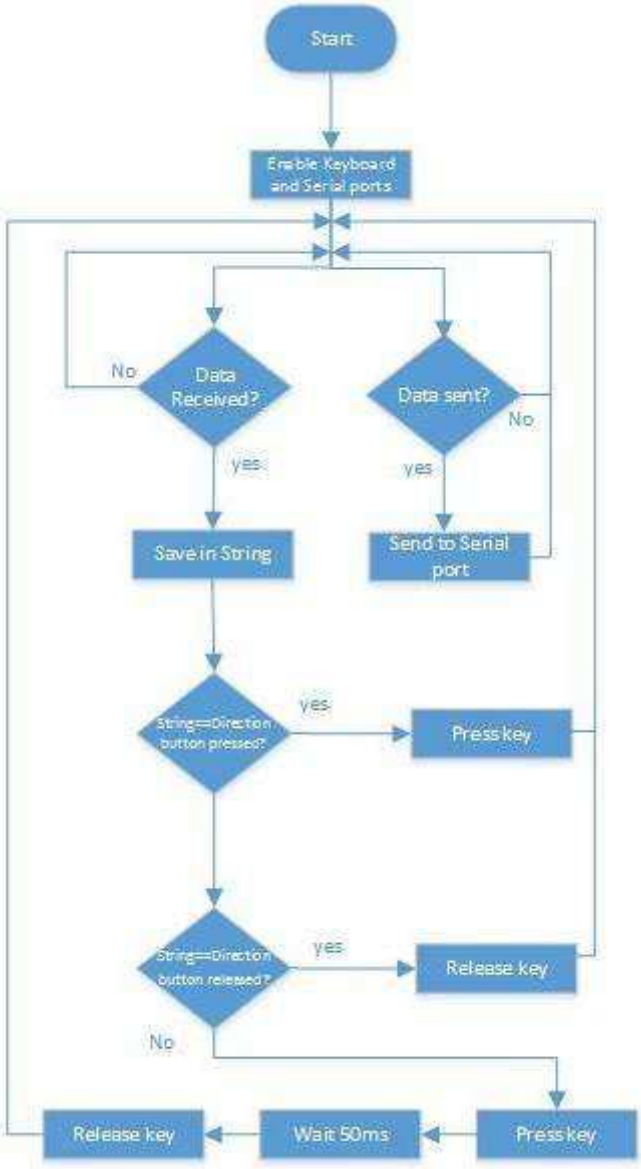
Cases 20 and 21 show how we have implemented the possibility of pressing two direction buttons at the same time, as many games require this feature.

Figure 05 –Running system in PC using app.

Source: BLE app

The application flow diagram is shown in the figure below.

Figure 06 – Flowchart



Source: BLE app

5 CONCLUSIONS

Once the system was completely operational, we tested it using a computer videogame verifying both control and behaviour correctness. There is no delay between pressing a key button in the smartphone and the corresponding action in the computer videogame due to low latency of Bluetooth low energy technology.

In the middle of Project development, there were some challenges that forced the next actions:

Arduino board can not receive data from some ports, therefore the HM-11 module was connected to ports 8 and 9 of Leonard.

Adding a delay between pressing and releasing actions was needed to permit differentiate both actions by the computer. Custom application development in the

smartphone was needed because there wasn't any application in Android Store that could use Bluetooth Low Energy technology.

REFERENCES

FLETCHER, L.; CORRAL, P; PIRES, E. DESIGN AND PROTOTYPE OF ARDUINO BASED SMARTWATCH FOR ALERGENIC PARTICLE DETECTION WITH BLUETOOTH TECHNOLOGY. In: 6º Simpósio de Integração Científica e Tecnológica do Sul Catarinense – SICT-Sul, oct. 2016.

Android Developers, "Bluetooth Low Energy", avalaible in:
https://developer.android.com/guide/topics/connectivity/bluetooth-le.html

"Bluetooth V4.0 HM-11 BLE Module", datasheet y configuraciones, avalaible in:
http://wiki.seeedstudio.com/wiki/Bluetooth_V4.0_HM-11_BLE_Module

"Getting started with Arduino Leonardo", disponible en:
https://www.arduino.cc/en/Guide/ArduinoLeonardoMicro

Library "Keyboard Arduino". [Online]. Avalaible in:
https://www.arduino.cc/en/Reference/MouseKeyboard

Library "Serial Arduino". [Online]. Avalaible in: https://www.arduino.cc/en/Reference/Serial