

**Reporte Final de Residencia Profesional**

Diseño de un sistema de control electrónico para el movimiento  
de una silla de ruedas eléctrica.

Ing. Electrónica

Julio César Torres Peña  
No. Control: 05270186

Empresa donde se realizó la Residencia

Secretaría de la Función Pública  
Dirección de Innovación y Desarrollo Tecnológico

Asesor

M.C. Rigoberto Jiménez Jonapá

Tuxtla Gutiérrez; Chiapas, a Junio de 2009

# INDICE

<b>Introducción</b> .....	<b>1</b>
<b>Planteamiento del Problema</b> .....	<b>1</b>
<b>Justificación</b> .....	<b>2</b>
<b>Objetivos</b> .....	<b>2</b>
General.....	2
Específicos.....	2
<b>Delimitación del Problema.</b> .....	<b>2</b>
<b>Fundamentos Teóricos</b> .....	<b>3</b>
Microcontrolador.....	3
Firmware .....	3
Bluetooth .....	4
Serial Port Profile (SPP).....	4
RFCOMM.....	5
Windows Mobile .....	5
<b>Desarrollo</b> .....	<b>6</b>
<i>Diseño de Software BlueControl</i>	
Características del Software.....	14
Operación y Funciones.....	14
Código Fuente.....	14
Instalación y Uso del Software.....	21
Operación del Software BlueControl .....	26
<i>Módulo de Control Electrónico para motores</i>	
Operación y Funciones.....	6
Firmware.....	7
Diagrama Esquemático .....	13
<b>Resultados</b> .....	<b>27</b>
<b>Conclusiones</b> .....	<b>28</b>
<b>Bibliografía</b> .....	<b>29</b>

## **Introducción**

El Presente documento tiene como finalidad describir el diseño de un sistema electrónico para el control de motores en una silla de ruedas eléctrica e incorporación de conectividad Bluetooth, para su operación a distancia mediante el desarrollo de un Software programado en plataforma Windows Mobile y su implementación en dispositivos móviles como PDA y Smartphone que dispongan de conectividad Bluetooth.

## **Planteamiento del Problema**

En nuestra sociedad actual hay un número cada vez mayor de personas con capacidades motrices limitadas debido, por ejemplo, a enfermedades degenerativas u otras patologías, accidentes o una edad avanzada. En algunos casos, estas discapacidades motoras también vienen asociadas de problemas cognitivos. Las sillas de ruedas eléctricas son un tipo especial de vehículos de ayuda a la movilidad cuyo objetivo es mejorar la calidad de la vida de este sector de la sociedad aumentando su movilidad y autonomía.

Dado al avance tecnológico las sillas de ruedas han mejorado sus características y funcionalidad, incorporando sistemas de movimiento y control, eléctrico, electrónico y mecánico.

Lamentablemente las compañías que diseñan y fabrican este tipo de silla de ruedas se han centrado en un sector de la población que a pesar de tener problemas motrices aun mantienen capacidad para operar una silla de ruedas mediante el control de un joystick o palanca de dirección para realizar su desplazamiento.

Sin embargo existen también personas que tienen problemas motrices o físicos severos que le impiden el uso de ese tipo de métodos de control y que requieren ser desplazadas mediante el uso de sillas de ruedas eléctricas con el apoyo de familiares o personal médico en hospitales.

Es por ello que se plantea el diseño de un sistema de control electrónico que tenga la cualidad de poder ser controlado a distancia utilizando tecnologías de comunicación inalámbricas y a su vez incorporando las mismas características de movilidad en una silla de ruedas eléctrica tradicional, para de esta forma tratar de cubrir alguna de las necesidades de este sector de la población.

## **Justificación**

El diseño de este sistema de control electrónico, tiene como finalidad disponer al usuario y al personal de apoyo la facilidad de poder controlar la movilidad de la silla de ruedas sin estar restringidos al uso del joystick que tradicionalmente se incorpora.

Así también el desarrollo de un software de control gráfico y sencillo contribuirá a que el usuario o el personal de apoyo pueda controlar la silla de ruedas desde un dispositivo móvil como una PDA o Smartphone.

Con ello la utilización de tecnologías inalámbricas como es el caso de Bluetooth facilitará la integración e interacción de todo el sistema de control con el usuario.

## **Objetivos**

### *General*

Diseñar una tarjeta de control electrónica para motores que pueda ser operada desde una PDA o Smartphone, utilizando un Software Gráfico desarrollado para Sistema Operativos Windows Mobile 5.0 o 6.0 y una conexión Bluetooth.

### *Específicos*

1. Diseño del Hardware de control para la regulación de velocidad y direccionamiento de los motores.
2. Desarrollo del Firmware para la comunicación y lógica de control.
3. Desarrollo del software de control gráfico para plataformas Windows Mobile 5.0 o 6.0.

## **Delimitación del Problema.**

Se diseñará el sistema de control electrónico para 2 motores de CD, que dispondrá de un módulo bluetooth para ser operado desde una PDA o Smartphone con ayuda de un software gráfico desarrollado para plataforma Windows Mobile 5.0 o 6.0, finalizando con la implementación del sistema en el prototipo a escala de una silla de ruedas eléctrica.

## Fundamentos Teóricos

### Microcontrolador

Un Microcontrolador es un circuito integrado (chip) que incluye en su interior las tres unidades funcionales de una computadora: CPU, memoria y unidades de E/S, es decir, se trata de una computadora completa en un solo circuito integrado. Pueden encontrarse en casi cualquier dispositivo eléctrico como automóviles, lavarropas, hornos microondas, teléfonos, equipos de audio, etc.. Típicamente, un Microcontrolador puede disponer de un generador de reloj integrado y una pequeña cantidad de memoria RAM y ROM/EPROM/EEPROM, significando que para hacerlo funcionar, todo lo que se necesita son unos pocos programas de control y un cristal de sincronización. Los microcontroladores disponen generalmente también de una gran variedad de dispositivos de entrada/salida, como convertidores de analógico a digital, temporizadores, UARTs y buses de interfaz serie especializados, como I2C y CAN, entre otros. Frecuentemente, estos dispositivos integrados pueden ser controlados por instrucciones de procesadores especializados. Los modernos microcontroladores a veces incluyen un lenguaje de programación integrado, como el BASIC que se utiliza bastante con este propósito. En caso de que no dispongan de un intérprete "on-board", se pueden programar desde el ordenador, usando también alguna versión de Basic, C, Pascal, assembler u otros.

### Firmware

Firmware o Programación en Firme, es un bloque de instrucciones de programa para propósitos específicos, grabado en una memoria de tipo no volátil (ROM, EEPROM, flash), que establece la lógica de más bajo nivel que controla los circuitos electrónicos de un dispositivo de cualquier tipo. Al estar integrado en la electrónica del dispositivo es en parte hardware, pero también es software, ya que proporciona lógica y se dispone en algún tipo de lenguaje de programación. Funcionalmente, el firmware es el intermediario (interfaz) entre las órdenes externas que recibe el dispositivo y su electrónica, ya que es el encargado de controlar a ésta última para ejecutar correctamente dichas órdenes externas.

Encontramos Firmware en memorias ROM de los sistemas de diversos dispositivos periféricos, como en monitores de video, unidades de disco, impresoras, etc., pero también en los propios microprocesadores, chips de memoria principal y en general en cualquier circuito integrado.

Muchos de los Firmwares almacenados en ROM están protegidos por Derechos de Autor.

El programa BIOS de una computadora es un firmware cuyo propósito es activar una máquina desde su encendido y preparar el entorno para la instalación de un Sistema Operativo complejo, así como responder a otros eventos externos (botones de pulsación humana) y al intercambio de órdenes entre distintos componentes de la computadora.

En un microprocesador el firmware es el que recibe las instrucciones de los programas y las ejecuta en la compleja circuitería del mismo, emitiendo órdenes a otros dispositivos del sistema.

### Bluetooth

La tecnología inalámbrica Bluetooth es una tecnología de ondas de radio de corto alcance (2.4 gigahertzios de frecuencia) cuyo objetivo es el simplificar las comunicaciones entre dispositivos informáticos, como ordenadores móviles, teléfonos móviles, otros dispositivos de mano y entre estos dispositivos e Internet. También pretende simplificar la sincronización de datos entre los dispositivos y otros ordenadores, permitiendo comunicaciones, incluso a través de obstáculos, a distancias de hasta unos 10 metros.

### Serial Port Profile (SPP)

Un perfil Bluetooth es la especificación de un interfaz de alto nivel para su uso entre dispositivos Bluetooth. Para utilizar una cierta tecnología Bluetooth un dispositivo deberá soportar ciertos perfiles.

Los perfiles son descripciones de comportamientos generales que los dispositivos pueden utilizar para comunicarse, formalizados para favorecer un uso unificado. La forma de utilizar las capacidades de Bluetooth se basa, por tanto, en los perfiles que soporta cada dispositivo. Los perfiles permiten la manufactura de dispositivos que se adapten a sus necesidades.

El Serial Port Profile está Basado en la especificación 07.10 de ETSI por medio del protocolo RFCOMM. Emula una línea serie y provee un interfaz de reemplazo de comunicaciones basadas en RS-232, con las señales de control típicas. Base de DUN, FAX, HSP y AVRCP.

## RFCOMM

El protocolo RFCOMM emula los parámetros de un cable de serie y el estado de un puerto RS-232 para transmitir datos en serie. El RFCOMM se conecta a las capas inferiores de la pila de protocolos *Bluetooth* a través de la capa L2CAP.

Al emular los puertos en serie, es compatible con aplicaciones heredadas que utilizan este tipo de conexión y admite, al mismo tiempo, el protocolo OBEX entre otros. Se trata de un subconjunto del estándar ETSI TS 07.10 con algunas adaptaciones específicas para la tecnología *Bluetooth*.

El protocolo RFCOMM es compatible con un máximo de 60 conexiones simultáneas entre dos dispositivos *Bluetooth*. El número de conexiones que pueden usarse simultáneamente en un dispositivo *Bluetooth* depende de la aplicación en cuestión.

En el contexto de RFCOMM, una ruta de comunicación completa necesita dos aplicaciones funcionando en dispositivos diferentes (los extremos de la comunicación) con un segmento de comunicación entre ellos.

RFCOMM se destina a aplicaciones que utilizan los puertos de serie de los dispositivos en los que residen. En configuración sencilla, el segmento de comunicación es un enlace *Bluetooth* de un dispositivo a otro (conexión directa). Cuando el segmento de comunicación es otra red, la tecnología inalámbrica *Bluetooth* se usa como la ruta entre el dispositivo y un dispositivo de conexión a la red, como un módem.

RFCOMM solo se ocupa de la conexión entre los dispositivos en el caso de una conexión directa, o entre el dispositivo y un módem en el caso de una conexión en red.

Básicamente, hay dos tipos de dispositivo con los que RFCOMM debe ser compatible. Los dispositivos de tipo 1 son terminales de comunicación, como ordenadores e impresoras. Los dispositivos de tipo 2 son los que forman parte del segmento de comunicación, como el módem, por ejemplo. Si bien RFCOMM no distingue entre estos dos tipos de dispositivo en el protocolo, la compatibilidad con ambos tipos de dispositivos afecta al protocolo RFCOMM.

## Windows Mobile

Es un sistema operativo compacto, con una suite de aplicaciones básicas para dispositivos móviles basados en la API Win32 de Microsoft. Los dispositivos que llevan Windows Mobile son Pocket PC, Smartphones y Media Center portátil. Ha sido diseñado para ser similar a las versiones de escritorio de Windows.

## Desarrollo

### *Módulo de Control Electrónico para motores.*

- *Operación y Funciones.*

El sistema de control electrónico, está desarrollado en base a un microcontrolador PIC16F876A en el cual se ejecuta el firmware que realiza las tareas de lógica y control del módulo, para la operación del sistema se reciben comandos mediante el protocolo USART provenientes de un módulo bluetooth **BlueSMiRF®**, que proporcionan información al sistema sobre la velocidad y dirección que debe seguir la silla de ruedas.

Los movimientos y funciones que ejecuta la silla de ruedas están clasificados en 10 tipos (Fig. 1.0).

1. Avance.
2. Retroceso.
3. Paro.
4. Giro cerrado a la Izquierda.
5. Giro cerrado a la Derecha.
6. Giro amplio Avance – Izquierda.
7. Giro amplio Avance – Derecha.
8. Giro amplio Retroceso – Izquierda.
9. Giro amplio Retroceso – Derecha.
10. Regulación de Velocidad.

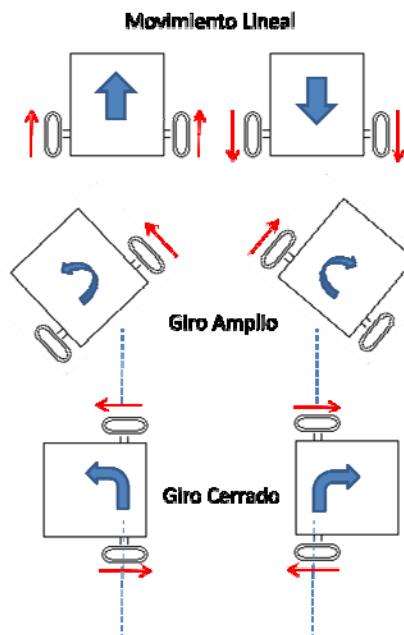


Fig. 1.0



Para la realización de los giros es necesario disponer de un par de motores acoplados a cada rueda de la silla, con la finalidad de tener un sistema de tracción diferencial.

La regulación de la velocidad en los motores es realizada mediante Modulación por Ancho de Pulso (PWM), consiguiendo así una disminución o aumento gradual en la velocidad de desplazamiento en la silla, con la finalidad de tener giros más precisos y seguros.

- Firmware

```
//-----  
//-----  
// Autor: Julio César Torres Peña.  
// Proyecto: Control Bluetooth para Silla de Ruedas Eléctrica.  
// Fecha: Lunes 26 Enero 2008.  
// Xtal: 4 Mhz  
// uC: PIC16F876A  
//-----  
//-----  
int receive = 0; // byte de recepción  
unsigned short Dutycycle; // Porcentaje PWM  
int IntCount; // Contador de Interrupción  
//-----  
//-----  
void interrupt(void){  
TMR0 = 99; // Recarga valor de tmr0  
IntCount--;  
if (IntCount == 0) // Inicializa las Salidas PWM con un estado alto  
{  
PORTC.F0 = 1;  
PORTC.F3 = 1;  
IntCount = 32;  

```

```

INTCON.F2 = 0;
return;
}
if (Dutycycle == IntCount) // Genera el cambio de estado en salida PWM
{
    //al igualar el valor de pwm y contador de interrupciones
    PORTC.F0 = 0;
    PORTC.F3 = 0;
}
INTCON.F2 = 0;
}
//-----
//-----
void ledTX(void) // Función para generar destello en led indicador de Trasmisión
PORTA.F4 = 1;
Delay_ms(5);
PORTA.F4 = 0;
Delay_ms(30);
PORTA.F4 = 1;
Delay_ms(5);
PORTA.F4 = 0;
}
//-----
//-----
void ledRX(void) // Función para generar destello en led indicador de Recepción
PORTA.F5 = 1;
Delay_ms(5);
PORTA.F5 = 0;
Delay_ms(30);
PORTA.F5 = 1;
Delay_ms(5);
PORTA.F5 = 0;
}
//-----
//-----

```

```

void ledReady(void){ // Función para generar destello en led de Estado
PORTA.F3 = 1;
Delay_ms(10);
PORTA.F3 = 0;
Delay_ms(50);
PORTA.F3 = 1;
Delay_ms(10);
PORTA.F3 = 0;
}
//-----
//-----

void Init (void){
PORTB = 0x00;    // Puerto B = Entradas, C = Salidas
TRISB = 0xFF;
PORTC = 0x00;
TRISC = 0x80;
PORTA = 0x00;
TRISA = 0x01;
CMCON = 0x07;
OPTION_REG = 0x01; // Habilitación de Interrupción por tmr0
TMR0 = 99;
INTCON = 0xA0;
Usart_Init(9600); // Inicialización del Modulo Usart a 9600bps
Dutycycle = 10;
}

//-----
//-----

void main (void){
Init();
do{
do{
if (Usart_Data_Ready()) //Lectura de un datos recibidos en puerto usart
{

```

```

receive = Usart_Read(); // Se guarda el dato en la variable receive

Usart_Write(receive);

LedRX();

}

LedReady();

}while(receive != 0x23); // Se espera confirmación del Modulo Bluetooth

receive = 0;          // de conexión creada con otro dispositivo

PORTA.F3 = 1;

do{

if (Usart_Data_Ready()) //Los datos que lleguen posteriormente serán
{
//léidos y evaluados.

receive = Usart_Read();

LedRX();

PORTA.F3 = 1;

}

switch (receive) //Comando de control para el puente H
{

case 0x31: PORTC.F1 = 1; //Input 4

PORTC.F2 = 0; //Input 3 //Adelante

PORTC.F4 = 1; //Input 2

PORTC.F5 = 0; //Input 1

break;

case 0x32: PORTC.F1 = 0; //Input 4

PORTC.F2 = 0; //Input 3 //Giro amplio Adelante-Derecha

PORTC.F4 = 1; //Input 2

PORTC.F5 = 0; //Input 1

break;

case 0x33: PORTC.F1 = 0; //Input 4

PORTC.F2 = 0; //Input 3 //Giro amplio Atrás-Izquierda

PORTC.F4 = 0; //Input 2

PORTC.F5 = 1; //Input 1

break;

case 0x34: PORTC.F1 = 0; //Input 4

PORTC.F2 = 1; //Input 3 //Reversa

```

```

    PORTC.F4 = 0; //Input 2
    PORTC.F5 = 1; //Input 1
    break;
case 0x35: PORTC.F1 = 0; //Input 4
    PORTC.F2 = 1; //Input 3    //Giro amplio Atrás-Derecha
    PORTC.F4 = 0; //Input 2
    PORTC.F5 = 0; //Input 1
    break;
case 0x36: PORTC.F1 = 1; //Input 4
    PORTC.F2 = 0; //Input 3    //Giro amplio Adelante-Izquierda
    PORTC.F4 = 0; //Input 2
    PORTC.F5 = 0; //Input 1
    break;
case 0x37: PORTC.F1 = 0; //Input 4
    PORTC.F2 = 0; //Input 3    //Paro
    PORTC.F4 = 0; //Input 2
    PORTC.F5 = 0; //Input 1
    break;
case 0x38: PORTC.F1 = 1; //Input 4
    PORTC.F2 = 0; //Input 3    //Giro corto Derecha
    PORTC.F4 = 0; //Input 2
    PORTC.F5 = 1; //Input 1
    break;
case 0x39: PORTC.F1 = 0; //Input 4
    PORTC.F2 = 1; //Input 3    //Giro corto Izquierda
    PORTC.F4 = 1; //Input 2
    PORTC.F5 = 0; //Input 1
    break;
case 0x2B: if(Dutycycle > 0)
    {
        Dutycycle--;    // Decremento de Velocidad
    }
    break;
case 0x2D: if(Dutycycle < 25)

```

```

        {
            Dutycycle++; // Incremento de Velocidad
        }
        break;
    default: break;
}

receive = 0;
}while(receive != 0x23); //Sale del Do while hasta o cuando
receive = 0; //la conexión del modulo Bluetooth sea
Dutycycle = 10; //finalizada o la silla este fuera del
LedReady(); //rango de la señal, deteniendo los
PORTC.F1 = 0; //Input //motores de la silla de ruedas
PORTC.F2 = 0; //Input 3
PORTC.F4 = 0; //Input 2
PORTC.F5 = 0; //Input 1
}while(1);
} //fin del programa

```

- Diagrama Esquemático (Fig. 1.1)

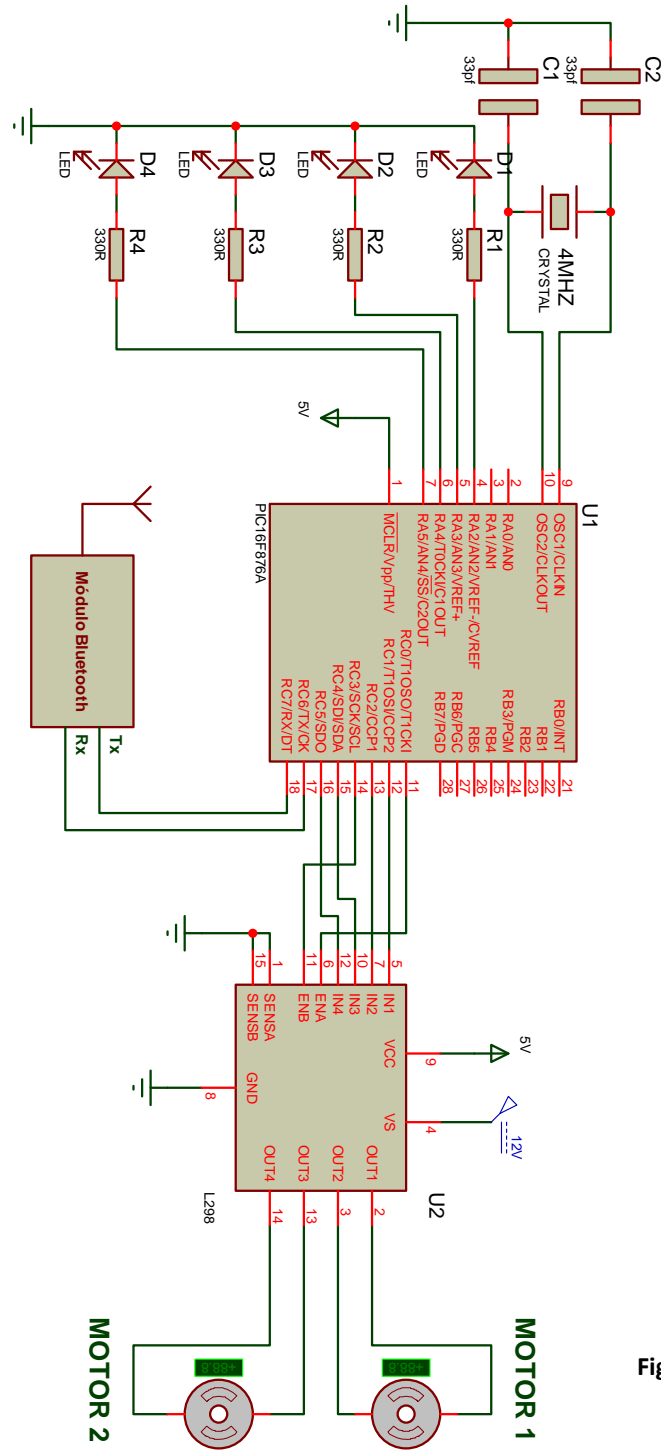


Fig. 1.1

## Diseño de Software BlueControl

- *Operación y Funciones.*

El diseño del software tiene como objetivo disponer al usuario de un panel de control gráfico y sencillo que mediante botones de direccionamiento controle la dirección y velocidad de desplazamiento en la silla de ruedas eléctrica.

El funcionamiento del software se basa en enviar comandos de control correspondientes al botón pulsado, mediante una conexión bluetooth creando una canal RFCOMM con dispositivos que soporten el Serial Port Profile.

Diseñado y desarrollado para trabajar en plataforma Windows Mobile 5.0, 6.0 este software es apto para ser implementado en dispositivos móviles como PDAs o Smartphones

- Características del Software

- ✓ Lenguaje de Programación C#.
- ✓ Compilador Visual Studio 2008, Microsoft .Net Framework 3.5.
- ✓ Plataforma de Operación Windows Mobile.
- ✓ Conectividad Bluetooth RFCOMM, Serial Port Profile.

- Código Fuente

```
//-----
```

```
//-----
```

```
// Autor: Julio César Torres Peña.
```

```
// Proyecto: Silla de Ruedas Eléctrica.
```

```
// Fecha: Lunes 26 Enero 2008.
```

```
//-----
```

```
//-----
```

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Text;
```



```

using System.Windows.Forms;

namespace BlueControl
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
        }

        private void Form1_Closed(object sender, EventArgs e)
        {
            notification1.Visible = false; //Elimina Notificaciones presentes.
            Close();
        }

        private void tabPage2_Click(object sender, EventArgs e)
        {
            notification2.Visible = false; //Elimina Notificaciones de Conexión y Desconexión
            notification3.Visible = false; //del Dispositivo al salir del tab de Conexiones.
        }

        private void menuItem1_Click(object sender, EventArgs e) //Salir de la Aplicacion.
        {
            notification1.Visible = false; //Elimina todas las notificaciones presentes.
            notification2.Visible = false;
            notification3.Visible = false;
            if (serialPort1.IsOpen == true) //Verifica si hay una conexión Bluetooth.
            {
                serialPort1.Close(); //Cierra la conexión activa.
            }
            Close(); //Cierra la Aplicacion.
        }

        private void pictureBox4_Click(object sender, EventArgs e) //Giro Adelante Derecha.
        {
            if (serialPort1.IsOpen == false) //Verifica conexión con el modulo Bluetooth
            {
                notification1.Visible = true; //notificación "conecte el dispositivo".
                notification3.Visible = false;
                notification2.Visible = false;
                return;
            }
            pictureBox11.Visible = false; // efecto de aumento de tamaño en el boton
            pictureBox12.Visible = true;
            pictureBox13.Visible = false;
            pictureBox14.Visible = false;
            pictureBox15.Visible = false;
            pictureBox16.Visible = false;
            pictureBox17.Visible = false;
            pictureBox18.Visible = false;
            pictureBox19.Visible = false;
            serialPort1.WriteLine("2"); //Envia el comando para generar la accion de control.
        }

        private void pictureBox5_Click(object sender, EventArgs e) // Giro Cerrado Derecha

```

```

{
if (serialPort1.IsOpen == false)//Verifica conexión con el modulo Bluetooth
{
notification1.Visible = true;//notificación "conecte el dispositivo".
notification3.Visible = false;
notification2.Visible = false;
return;
}
pictureBox11.Visible = false;//efecto de aumento de tamaño en el boton pulsado.
pictureBox12.Visible = false;
pictureBox13.Visible = true;
pictureBox14.Visible = false;
pictureBox15.Visible = false;
pictureBox16.Visible = false;
pictureBox17.Visible = false;
pictureBox18.Visible = false;
pictureBox19.Visible = false;
serialPort1.WriteLine("9");//Envía el comando para generar la accion de control }

private void pictureBox6_Click(object sender, EventArgs e) //Giro Atras Derecha
{
if (serialPort1.IsOpen == false) //Verifica conexión con el modulo Bluetooth
{
notification1.Visible = true; //notificación "conecte el dispositivo".
notification3.Visible = false;
notification2.Visible = false;
return;
}
pictureBox11.Visible = false;//efecto de aumento de tamaño en el boton pulsado.
pictureBox12.Visible = false;
pictureBox13.Visible = false;
pictureBox14.Visible = true;
pictureBox15.Visible = false;
pictureBox16.Visible = false;
pictureBox17.Visible = false;
pictureBox18.Visible = false;
pictureBox19.Visible = false;
serialPort1.WriteLine("3");//Envía el comando para generar la accion de control
}

private void pictureBox1_Click(object sender, EventArgs e)//Reversa
{
if (serialPort1.IsOpen == false)//Verifica conexión con el modulo Bluetooth
{
notification1.Visible = true;//notificación "conecte el dispositivo".
notification3.Visible = false;
notification2.Visible = false;
return;
}
pictureBox11.Visible = false; //efecto de aumento de tamaño en el boton pulsado.
pictureBox12.Visible = false;
pictureBox13.Visible = false;
pictureBox14.Visible = false;
pictureBox15.Visible = true;
pictureBox16.Visible = false;
pictureBox17.Visible = false;
pictureBox18.Visible = false;
pictureBox19.Visible = false;
serialPort1.WriteLine("4");//Envía el comando para generar la accion de control.
}

private void pictureBox9_Click(object sender, EventArgs e) //Giro Atras Izquierda

```

```

{
if (serialPort1.IsOpen == false)//Verifica conexión con el modulo Bluetooth
{
notification1.Visible = true; //notificación "conecte el dispositivo".
notification3.Visible = false;
notification2.Visible = false;
return;
}
pictureBox11.Visible = false; //efecto de aumento de tamaño en el boton pulsado.
pictureBox12.Visible = false;
pictureBox13.Visible = false;
pictureBox14.Visible = false;
pictureBox15.Visible = false;
pictureBox16.Visible = true;
pictureBox17.Visible = false;
pictureBox18.Visible = false;
pictureBox19.Visible = false;
serialPort1.WriteLine("5");//Envia el comando para generar la accion de control
}

private void pictureBox8_Click(object sender, EventArgs e)//Giro Cerrado Izquierda
{
if (serialPort1.IsOpen == false)//Verifica conexión con el modulo Bluetooth
{
notification1.Visible = true;//notificación "conecte el dispositivo".
notification3.Visible = false;
notification2.Visible = false;
return;
}
pictureBox11.Visible = false;//efecto de aumento de tamaño en el boton pulsado.
pictureBox12.Visible = false;
pictureBox13.Visible = false;
pictureBox14.Visible = false;
pictureBox15.Visible = false;
pictureBox16.Visible = false;
pictureBox17.Visible = true;
pictureBox18.Visible = false;
pictureBox19.Visible = false;
serialPort1.WriteLine("8");//Envia el comando para generar la accion de control
}

private void pictureBox3_Click(object sender, EventArgs e)//Stop
{
if (serialPort1.IsOpen == false)//Verifica conexión con el modulo Bluetooth
{
notification1.Visible = true;//notificación "conecte el dispositivo".
notification3.Visible = false;
notification2.Visible = false;
return;
}
pictureBox11.Visible = false;//efecto de aumento de tamaño en el boton pulsado.
pictureBox12.Visible = false;
pictureBox13.Visible = false;
pictureBox14.Visible = false;
pictureBox15.Visible = false;
pictureBox16.Visible = false;
pictureBox17.Visible = false;
pictureBox18.Visible = false;
pictureBox19.Visible = true;
serialPort1.WriteLine("7");//Envia el comando para generar la accion de control
}
}

```

```

private void pictureBox2_Click(object sender, EventArgs e)//Adelante
{
    if (serialPort1.IsOpen == false)//Verifica conexión con el modulo Bluetooth
    {
        notification1.Visible = true; //notificación "conecte el dispositivo".
        notification3.Visible = false;
        notification2.Visible = false;
        return;
    }
    pictureBox11.Visible = true;//efecto de aumento de tamaño en el boton pulsado.
    pictureBox12.Visible = false;
    pictureBox13.Visible = false;
    pictureBox14.Visible = false;
    pictureBox15.Visible = false;
    pictureBox16.Visible = false;
    pictureBox17.Visible = false;
    pictureBox18.Visible = false;
    pictureBox19.Visible = false;
    serialPort1.WriteLine("1");//Envia el comando para generar la accion de control
}

```

```

private void pictureBox7_Click(object sender, EventArgs e)//Adelante Izquierda.
{
    if (serialPort1.IsOpen == false)//Verifica conexión con el modulo Bluetooth
    {
        notification1.Visible = true;//notificación "conecte el dispositivo".
        notification3.Visible = false;
        notification2.Visible = false;
        return;
    }
    pictureBox11.Visible = false;//efecto de aumento de tamaño en el boton pulsado.
    pictureBox12.Visible = false;
    pictureBox13.Visible = false;
    pictureBox14.Visible = false;
    pictureBox15.Visible = false;
    pictureBox16.Visible = false;
    pictureBox17.Visible = false;
    pictureBox18.Visible = true;
    pictureBox19.Visible = false;
    serialPort1.WriteLine("6");//Envia el comando para generar la accion de control
}

```

```

private void button1_Click(object sender, EventArgs e)
{
    serialPort1.Open();//Conecta el dispositivo al módulo Bluetooth.
    if (serialPort1.IsOpen == true)
    {
        notification2.Visible = true; //Indica el estado de la conexion.
        notification3.Visible = false;
        notification1.Visible = false;
    }
}

```

```

private void button2_Click(object sender, EventArgs e)
{
    serialPort1.Close();//Desconecta el dispositivo al módulo Bluetooth.
    if (serialPort1.IsOpen == false)
    {
        notification2.Visible = false;//Indica el estado de la conexion.
        notification3.Visible = true;
        notification1.Visible = false;
    }
}

```

```

    }
}

private void tabControl1_SelectedIndexChanged(object sender, EventArgs e)
{
    notification2.Visible = false; //Oculta notificaciones
    notification3.Visible = false;
    notification1.Visible = false;
}

private void Form1_Closing(object sender, CancelEventArgs e)
{
    notification1.Visible = false; //Elimina Todas las notificaciones.
    notification2.Visible = false;
    notification3.Visible = false;
    Close();
}

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    switch (comboBox1.SelectedIndex)//Cambia el nombre del puerto RFCOMM
    {
        case 0: serialPort1.PortName = "COM0";//Asigna el puerto 0 a RFCOMM
            break;
        case 1: serialPort1.PortName = "COM1";
            break;
        case 2: serialPort1.PortName = "COM2";
            break;
        case 3: serialPort1.PortName = "COM3";
            break;
        case 4: serialPort1.PortName = "COM4";
            break;
        case 5: serialPort1.PortName = "COM5";
            break;
        case 6: serialPort1.PortName = "COM6";
            break;
        case 7: serialPort1.PortName = "COM7";
            break;
        case 8: serialPort1.PortName = "COM8";
            break;
        case 9: serialPort1.PortName = "COM9";
            break;
        case 10: serialPort1.PortName = "COM10";
            break;
        case 11: serialPort1.PortName = "COM11";
            break;
        case 12: serialPort1.PortName = "COM12";
            break;
        case 13: serialPort1.PortName = "COM13";
            break;
        case 14: serialPort1.PortName = "COM14";
            break;
        case 15: serialPort1.PortName = "COM15";
            break;
        default: break;
    }
}

private void button4_Click(object sender, EventArgs e)

```

```

{
    if (serialPort1.IsOpen == false) //Verifica que el puerto este Conectado.
    {
        notification1.Visible = true;
        notification3.Visible = false;
        notification2.Visible = false;
        return;
    }
    serialPort1.WriteLine("+");//Incrementa la Velocidad en los motores.
    if (progressBar1.Value < 96)
    {
        progressBar1.Value = progressBar1.Value + 4;
        //Modifica el estado de la barra indicando el incremento de la velocidad.
    }
}

private void button3_Click(object sender, EventArgs e)
{
    if (serialPort1.IsOpen == false)//Verifica que el puerto este Conectado.
    {
        notification1.Visible = true;
        notification3.Visible = false;
        notification2.Visible = false;
        return;
    }
    serialPort1.WriteLine("-");//Decrementa la Velocidad en los motores.
    if (progressBar1.Value > 4)
    {
        progressBar1.Value = progressBar1.Value - 4;
        //Modifica el estado de la barra indicando el decremento de la velocidad.
    }
}
}
}

```

## Instalación y Uso del Software

### *Procedimiento para la Instalación del Software BlueControl*

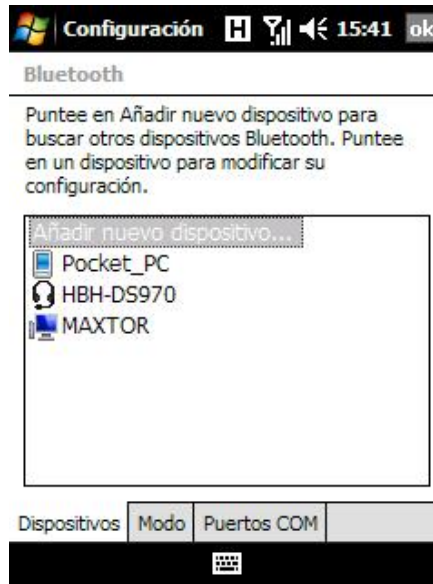
1. Habilitar el módulo bluetooth de la PDA o Smartphone.



2. Abrir las configuraciones del módulo Bluetooth en Inicio-Configuraciones-Conexiones-Bluetooth.



3. Seleccionar Añadir un nuevo dispositivo Bluetooth.

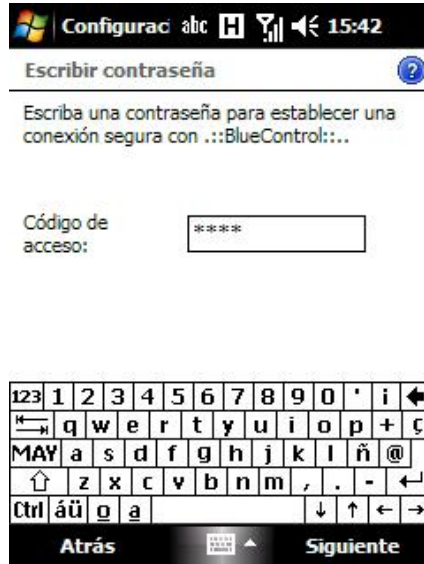


4. Al finalizar la búsqueda seleccionar el dispositivo `::BlueControl::`. Y hacer click en siguiente.

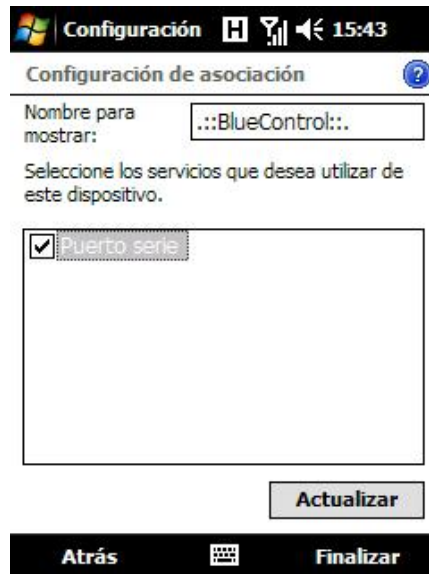




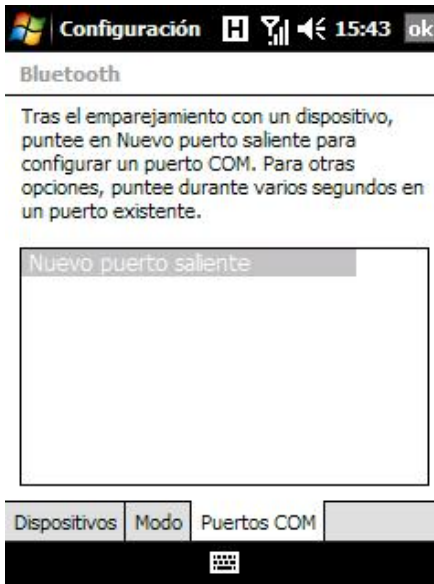
5. Introducir el PIN de identificación (1234) para realizar el acoplamiento del dispositivo y hacer click en siguiente.



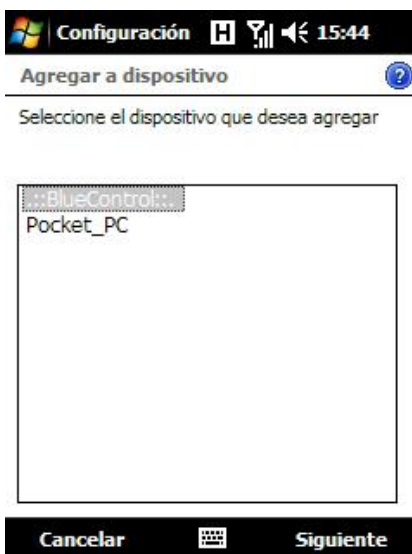
6. Una vez agregado el dispositivo seleccionar la propiedad Puerto Serie y hacer click en Finalizar.



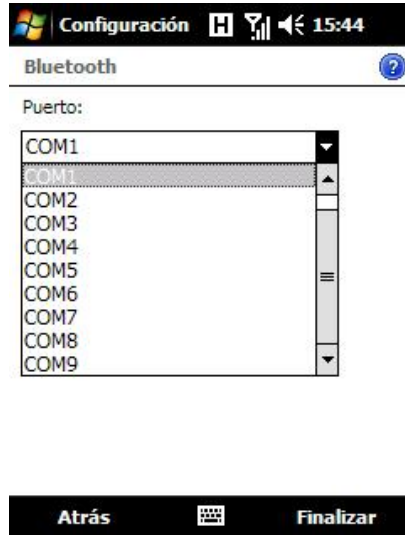
7. Regresar a la ventana para agregar nuevo dispositivo Bluetooth y seleccionar la pestaña "Puertos COM", posteriormente seleccione "Nuevo Puerto Saliente".



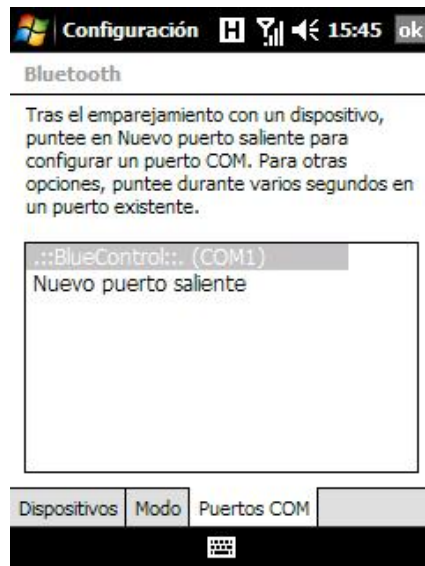
8. Seleccione el dispositivo `::BlueControl::`. Para agregar nuevo puerto saliente.



9. Seleccione el Puerto COM número 1 para la conexión saliente y haga click en Finalizar.



10. Finalmente el Puerto COM 1 será asociado al dispositivo `:::BlueControl:::`



## Operación del Software BlueControl

1. Una vez configurado el puerto de conexión para el dispositivo `::BlueControl::`, Ejecute la aplicación BlueControl.exe en su PDA o Smartphone.



2. Seleccione el Puerto COM 1 en la pestaña conexiones y de click en conectar.



3. Finalmente con el panel principal del software regule la velocidad y dirección de la silla de ruedas conectada al dispositivo.



## Resultados

Se verificó la correcta conexión del módulo bluetooth a la PDA, teniendo una buena conexión y recepción de datos, posteriormente se verificó la distancia efectiva de operación del sistema de control en interiores y exteriores, obteniendo los siguientes resultados:

Ambiente	Distancia
Interiores	7m.
Exteriores	13m.

Así también se verificó el correcto funcionamiento del software y la tarjeta de control, teniendo un resultado y desempeño favorable.

Finalmente se corrigió e incluyó en el firmware la detección de conexión fallida, para detectar el momento en el que ambos dispositivos pierdan la conexión bluetooth y así generar automáticamente el apagado de los motores de la silla de ruedas.

## Conclusiones

El desarrollo de este sistema de control, propició la incorporación y estudio de tecnologías inalámbricas actuales como es el caso del Bluetooth, así como el desarrollo de software aplicado a plataformas Windows Mobile, logrando el control de los motores y velocidad de desplazamiento en un silla de ruedas eléctrica, así también este sistema puede ser implementado en dispositivos o equipos electrónicos que requieran ser operados a distancia, como es el caso de plataformas para el traslado de materiales peligrosos o cargas pesadas.

Una mejora en características del sistema podría estar dada con la implementación de módulos Bluetooth de mayor potencia para así alcanzar mayores distancias de operación efectiva, así como también el desarrollo del software de control para plataformas como Java, Symbian, Palm OS y Windows.

Finalmente el desarrollo de este proyecto está orientado y fortalecido mediante los conocimientos adquiridos en el estudio profesional en el ámbito de Programación, Electrónica Digital y Sistemas de Comunicación; con la finalidad de desarrollar Sistemas Embebidos en la solución de problemas de la sociedad e incursionar en la investigación y desarrollo tecnológico, fortaleciendo nuestros conocimientos en ingeniería.

## **Bibliografía**

<http://es.wikipedia.org/wiki/Firmware>

<http://www.neoteo.com/microcontroladores.neo>

[http://spanish.bluetooth.com/Bluetooth/Technology/Works/Architecture\\_\\_Logical\\_Link\\_Control\\_and\\_Adaptation\\_Protocol\\_L2CAP.htm](http://spanish.bluetooth.com/Bluetooth/Technology/Works/Architecture__Logical_Link_Control_and_Adaptation_Protocol_L2CAP.htm)

[http://es.wikipedia.org/wiki/Perfil\\_Bluetooth](http://es.wikipedia.org/wiki/Perfil_Bluetooth)

<http://microchip.com>

<http://www.rovingnetworks.com/documents/RN-41.pdf>

<http://www.rovingnetworks.com/documents/BlueportII-ref-guide.pdf>

<http://msdn.microsoft.com/es-es/library/aa187916.aspx>