

DISEÑO PROTOTIPO ROBOT EXPLORADOR MOVIL CONTROLADO A DISTANCIA CON USO DE ANDROID, PIC16F627A Y BLUETOOTH RN-42.

Héctor Gerardo Lara Gutiérrez¹, Efraín Andrade Hernández¹, Víctor Tomás Tomás Mariano¹

¹Lic. En Sistemas Computacionales, Universidad Autónoma del Estado de Hidalgo - Escuela Superior Huejutla. Corredor Industrial S/N, Parque de Poblamiento, Huejutla de Reyes, Hidalgo. CP. 43000. Tel. 771-7172000, ext. 5880,5881, e-mail: rojiblanco9021@hotmail.com, efrain.andrade.h@gmail.com

RESUMEN

Se presenta el prototipo de un carro explorador controlado a distancia mediante dispositivo móvil que usa sistema operativo android 4.0 y uso de pic, y modulo bluetooth. Se detalla construcción, configuración y modo de ensamble. El prototipo propuesto tiene la capacidad de enviar imágenes y video mediante cámara Ip Foscam Fi8907w, usado para ambientes de difícil acceso. También se presentan los resultados obtenidos en pruebas realizadas en ambientes controlados y no controlados.

Palabras Clave: robot, dispositivo móvil, bluetooth, android.

ABSTRACT

Prototype of a remotely via mobile device that uses Android 4.0 operating system and use of PIC, and explorer bluetooth car module is presented. Construction, configuration and detailed assembly mode. The proposed prototype has the ability to send images and video using IP camera Foscam FI8907W, used for difficult to access environments. The results of tests conducted in controlled and uncontrolled environments are also presented.

Keywords: robot, mobile device, bluetooth, android.

INTRODUCCION.

La robótica móvil es una valiosa herramienta para explorar entornos inaccesibles al ser humano por su lejanía, coste o peligrosidad y para realizar tareas desagradables o laboriosas. Es un campo relativamente nuevo, hasta hace poco experimental, pero que ya se está aplicando a problemas reales con resultados satisfactorios [1].

Asociado a esto en el presente proyecto se documenta la metodología para desarrollar un prototipo robótico con capacidad de explorar y transmitir imágenes en tiempo real, maniobrado a una distancia lejana [1].

DESARROLLO.

El proceso de ensamble del prototipo robótico se maneja en cuatro etapas principales, en la quinta y sexta etapa se desarrollan las pruebas pertinentes para la valoración del vehículo explorador controlado a través de un dispositivo móvil con sistema operativo Android.

En el siguiente listado se muestra de manera general los componentes del prototipo:

1. Módulo bluetooth RN-42.
2. Microcontrolador Pic16F627A.
3. Puente H L293B.
4. Motorreductor Doble (dos motores giro independiente) 3v a 6v corriente continua.
5. Capacitor cerámico de 100nF.
6. Capacitor electrolítico (Radial) de aluminio, de 1000 uF.
7. Alimentación 5V.
8. Porta baterías AA(4 baterías).
9. Base de plástico.
10. Sistema de oruga, consiste en un juego de ruedas (dentadas, lisas) y bandas de plástico.

Etapas 1: Configuración bluetooth RN-4.

Herramientas utilizadas en la configuración del módulo bluetooth RN-42.

1. HyperTerminal emulador de terminal.
2. Ordenador con bluetooth integrado
3. Módulo de Bluetooth RN-42 para Microcontroladores.
4. Protoboard.
5. Fuente de alimentación 5v

Módulo bluetooth RN-42: Dispositivo que permite transmitir voz y datos entre diferentes dispositivos que se encuentran dentro de una Red Inalámbrica de Área Personal (WPAN), utiliza un enlace de radiofrecuencia entre 2.402 – 2.48 GHz. Permite la comunicación entre equipos móviles y fijos, se utiliza con mayor frecuencia

en los teléfonos móviles, computadoras portátiles, ordenadores personales, impresoras o cámaras digitales[6].

A continuación se describe el proceso para el desarrollo

- 1.1 Situar el modulo Bluetooth RN-42 sobre la protoboard, alimentar la corriente eléctrica mediante los pines VCC, GND, corriente y tierra física, respectivamente. Por default al conectar el modulo Bluetooth se encenderá automáticamente, ver figura 1.

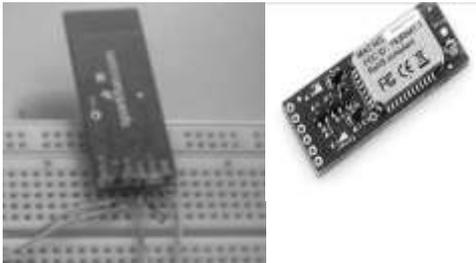


Figura 1 Conexión del módulo bluetooth RN-42.

- 1.2 Sincronizar el modulo con el computador agregando el código de seguridad del RN-42 que trae por default en la configuración de fábrica.
- 1.3 Instalar HyperTerminal generando una nueva conexión con su respectivo nombre, configurándose automáticamente.
- 1.4 Creada la conexión se enciende automáticamente el led de color verde y se apaga el rojo. Esto significa que la conexión es exitosa.
- 1.5 Reiniciar manualmente la conexión, para ello hacer clic en los iconos “desconectar” y “conectar” situados en la barra de herramientas en la parte superior izquierda. Una vez restablecida la conexión inmediatamente se introduce tres veces el siguiente carácter \$ el cual permitirá la entrada al modo CMD o modo comando.
- 1.6 Encendido del eco, para visualizar los comandos, se presiona la tecla + seguida de un *enter*, aparecerá la leyenda ECHO ON (eco encendido) se visualiza en pantalla el tipo de comandos.
- 1.7 Para la configuración de los baudios se presiona el carácter *D*, la tasa de baudios (Baudrt) se debe encontrar en 9600bps (baudios por segundo), de lo contrario se consulta en *SET COMMANDS*, el comando *SU (SU, <rate> -Baudrate)* que hará posible el cambio de baudios.

- 1.8 Se introduce el comando *SU, 9600* (*SU* coma 9600) seguido de un *enter*, inmediatamente aparece la leyenda *AOK* significa que la configuración es exitosa, para corroborarlo se consulta el ajuste presionando nuevamente el carácter *D*. El cual debe mostrar *Baudrt=9600* respectivamente.

En este punto culmina la configuración del módulo Bluetooth RN-42.

Etapa 2: Programación del PIC16F627A

Material utilizado para programar el PIC:

1. Ordenador
 2. PIC16F627A
 3. Programador USB para microcontroladores.
 4. Master Prog (software).
 5. MPLAB (Software)
- 2.1. MPLAB es una herramienta que se utiliza para la escritura y el desarrollo de código en lenguaje ensamblador para programar microcontroladores. En el se crea un nuevo proyecto para editar el código fuente, verificar posibles errores y compilar. Todo ello para obtener el archivo.hex (también llamado firmware)[9].
 - 2.2. Grabar el archivo.hex con la asistencia de Master Prog herramienta utilizada en conjunto con el “Programador USB para microcontroladores” software y hardware respectivamente.
 - 2.3. Se coloca de manera correcta el PIC dentro del “Programador USB” colocando el seguro respectivo.
 - 2.4. Cargar el archivo.hex hacia Master Prog. *Barra de menús> Archivo> Abrir HEX*. Se visualiza una leyenda indicando que el archivo se cargo exitosamente.
 - 2.5. Grabar el código al PIC para ello la barra de herramientas despliega una serie de opciones, utilizando solamente las mencionadas a continuación:
 1. Leer: lee el código cargado.
 2. Escribir: graba el código dentro del PIC.
 3. Verificar: comprueba la grabación del código dentro del PIC sea óptima.

De esta forma se concluye la etapa de la programación del PIC [8], [11].

Etapa 3: Construcción de la base y el circuito del prototipo.

Fase 1: Base para el prototipo

Se construye la estructura que conforma en mayor parte el prototipo robótico de exploración. Para ello es necesario los siguientes materiales:

1. Base universal de plástico
2. Motorreductor Doble (dos motores giro independiente) 3v a 6v corriente continua.
3. Sistema de oruga, consiste en un juego de ruedas (dentadas, lisas) y bandas de plástico.
4. Porta baterías AA (4 baterías).

Procedimiento:

- 3.1.1. Situar el Motorreductor sobre la base universal de plástico.
- 3.1.2. Incorporar el sistema de oruga, permite que el prototipo avance. Los motores son de giro totalmente independiente, esto contribuye a la rotación hacia ambos según sea el caso.
- 3.1.3. Se incorpora el porta baterías.

Fase 2: Armado del circuito.

Diagrama del circuito, Figura 2, tomado de blog Mexatronica, modificado a las necesidades requeridas, para el desarrollo del prototipo robótico [8].

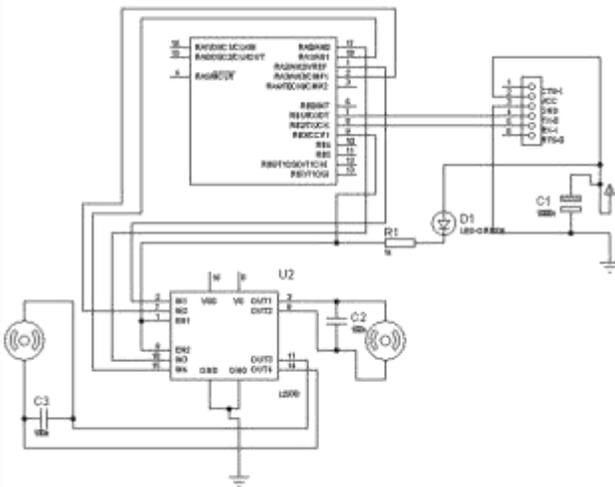


Figura 2 Diagrama pictórico vehículo explorador[8]

El circuito está conformado por los siguientes componentes:

1. Protoboard.
2. Capacitor electrolítico (Radial) de aluminio, de 1000 uF.
3. 2 capacitores Cerámicos 100nF.
4. PIC16F627A.
5. Puente H.

6. Módulo bluetooth RN-42.

Microcontrolador: circuito integrado programable, diseñado para ser capaz de ejecutar las instrucciones grabadas en su memoria ROM, incluye tres principales unidades de una computadora: unidad central de procesamiento, memoria y periféricos de entrada y salida Figura 3 [10].

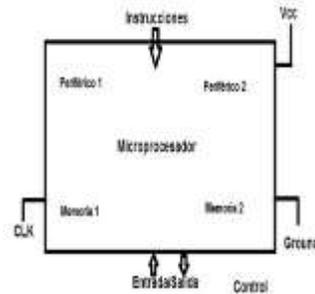


Figura 3 Esquema de microcontrolador

PIC16F627A del fabricante MICROCHIP Inc, es un CMOS de 8 bits, tiene un empaquetado DIP de 18 pines, una memoria programa de 2KWords, una memoria RAM de 224 registros, su memoria EEPROM de 128 bytes, cuenta con un puerto serial USART, cuenta con un oscilador interno de 4MHz capaz de soportar un oscilador externo hasta de 20MHz, preparado para 32 instrucciones Figura 4 [3] [4].

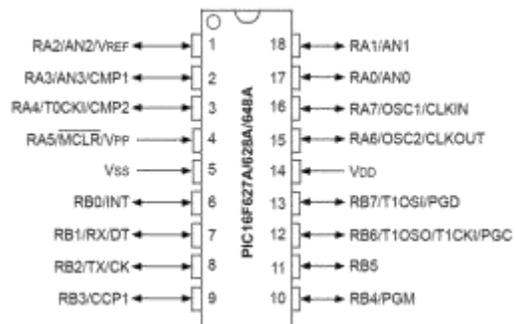


Figura 4 PIC16F627A

El proceso de ensamble del circuito consiste en situar los componentes (Modulo bluetooth, PIC16f627a, Puente H) sobre la protoboard para conectarlos mediante cables de cobre, como se muestra en la Figura 2.

La función del esquema presentado en la figura 2 se define como un circuito eléctrico, constituido por un módulo bluetooth RN-42 componente receptor de instrucciones, transmitidas por un dispositivo móvil. Las instrucciones son codificadas en señales analógicas para su envío a través de radio frecuencia, durante la recepción de las ondas de radio el módulo bluetooth RN-42 genera

un proceso de decodificación, transformando las señales análogas a digitales las cuales son canalizadas a través de los pines TX-0 Y RX-1 (pines de salida del módulo) para ingresar por los pines RB-1 y RB-2 (pines de entrada del pic). Este tipo de transmisiones se realiza de manera serial, significa que los bits que componen cada carácter se transmiten en “n” ciclos de 1 bit cada uno, hasta completar cada carácter.

Dentro del PIC se procesa la señal emitida, decodificándose para trasladarse al Puente H, proceso en el cual se involucran los puertos RA0, RA1, RA3, RB3 (pines de salida, Pic16f627a) y los puertos IN1, IN2, IN3, IN4 y EN12 (pines de entrada, Puente H L293B).

La función del driver de manejo o Puente H dentro del circuito, es controlar la polaridad de corriente que deba salir por sus respectivos pines OUT1, OUT2 para el motor 1 y OUT3, OUT4 para el motor 2[10].

Para finalizar el ensamble se colocan los capacitores conforme es señalado en el circuito Figura 2 se sitúan de tal manera que establezcan las tensiones eléctricas, provenientes de la fuente de alimentación de 5v[5], [7], [8].

Fase 3: Implementación de la cámara ip

El concepto de cámara ip hace referencia a un dispositivo encargado de la captura de imágenes en tiempo real con la capacidad de transmitir directamente a la red sin necesidad de un ordenador [2].

El modelo de la cámara Ip Foscam Fi8907w Wifi Audio Night Vision Nocturna compatible con navegador indistinto [2].

La implantación del sistema de cámara ip se alimenta eléctricamente mediante una pila de 9v, para ello se adecua un adaptador especial para dicha conexión. Utilizando soportes de metal liviano para el montaje sobre el chasis del prototipo.

Para la visualización del video se asigna una dirección IP para la cámara, la cual se coloca en el navegador, inmediatamente arroja en pantalla el menú con dos opciones:

- Visualizar video
- Configuraciones

Etapa 4: Creación de la interfaz grafica

El funcionamiento del carro explorador se lleva a cabo a través de la aplicación BlueTerm para un dispositivo móvil con sistema operativo Android, para transferir

datos por medio del Bluetooth, la aplicación no cuenta con una interfaz amigable para el usuario, por lo que fue necesario desarrollar una nueva aplicación específica para el carrito explorador.

Requerimientos para el desarrollo de la aplicación.

1. - Paquete de Android SDK.
2. - Java Runtime Environment (JRE).
3. - Eclipse.
4. - Eclipse Plug-in (Android Development Tools - ADT).

Fase 1. Instalar cada uno de los componentes para integrar la plataforma de trabajo. Todos los paquetes son de distribución libre, que pueden ser descargados desde su página oficial [12].



Figura 5. Ventana para nuevo proyecto

Fase 2. Se utiliza Eclipse como entorno de desarrollo, en donde se genera un nuevo proyecto con Android Application Project, como se muestra en la figura 5.

Fase 3. Configuración de los requerimientos mínimos que debe tener el proyecto se recomienda utilizar las especificaciones indicadas en la Figura 6 con un Android 2.2 (Froyo), para mantener la compatibilidad con versiones posteriores al Android 2.2



Figura 6. Configuración del proyecto

Fase 4. La construcción de la interfaz se realiza a través de código xml (eXtensible Markup Language), tiene 9 botones, con las siguientes funciones:

1. Conectar con el Bluetooth del carrito.
2. Desconectar con Bluetooth del carrito.
3. Desplazamiento a la Izquierda
4. Desplazamiento a la Derecha
5. Desplazamiento hacia enfrente.
6. Desplazamiento hacia atrás.
7. Aumentar la velocidad
8. Disminuir la velocidad
9. Detener



Figura 7. Interfaz de la aplicación

Fase 5. Establecer la conexión vía Bluetooth, la plataforma android ofrece el soporte para conexión de redes mediante bluetooth, que permiten el intercambio de información de forma inalámbrica con otros dispositivos. La funcionalidad del bluetooth se lleva a cabo a través de las API's.

Con el uso de la API de Bluetooth, una aplicación para Android puede hacer lo siguiente:

1. Buscar otros dispositivos
2. Verificar los dispositivos conectados
3. Transferencia de datos con otros dispositivos
4. Administración de múltiples conexiones
5. A continuación enumeran las fases más importantes en el desarrollo de la conexión de los dos dispositivos.

Fase 5.1. Buscar dispositivos Bluetooth remotos. La detección de dispositivos es un modo de exploración que busca en un área local los dispositivos habilitados para Bluetooth, posteriormente solicitar información acerca de cada uno de estos. Un dispositivo Bluetooth responderá a la solicitud solo si se encuentra activada de modo

visible. Si un dispositivo es detectado, este responderá a la petición de mediante el intercambio de información, como el nombre del dispositivo y su dirección MAC. Usando esta información, puede iniciar una conexión para una transferencia de datos con el dispositivo descubierto.

```

Void BuscarBT()
{
    Set<BluetoothDevice> pairedDevices
    =AdaptadorBluetooth.getBondedDevices();
    Device =
    AdaptadorBluetooth.getRemoteDevice(MACBlu
    e);
    if (pairedDevices.contains(Device))
    {
        estado.setText("El
        dispositivo "+Device.getName()+" ya
        esta vinculado.");
        Log.d("Carrito Buetooth", "BT is
        paired");
    }
}

```

Fase 5.2 Socket de comunicación para conectar el modulo Bluetooth.

Para crear una conexión entre los dos dispositivos, es necesario tener una conexión BluetoothSocket en el mismo canal RFCOMM (Radio Frequency Communication), como se ilustra en el siguiente código.

```

Void conectarBT() throwsIOException
{
    UUID uuid =
    UUID.fromString("00001101-0000-
    1000-8000-00805F9B34FB"); Socket =
    Device.createRfcommSocketToServiceR
    ecord(uuid);
    Socket.connect();
    OutputStream =
    Socket.getOutputStream();
    InputStream =
    Socket.getInputStream();
    enEsperaDeInstrucciones();
    estado.setText("Conectado a:
    "+Device.getName());
    BuscarBT();
    conectado=true;
}

```

Fase 5.3 La transferencia de datos por Bluetooth. Para realizar el envío de datos es necesario utilizar la subclase OutputStream, específicamente el método write, en el cual el envío se realiza en bytes.

```

void EnviarOrden(String Dato)
{
    try{
        if(conectado)

        OutputStream.write(Dato.getBytes());
        } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
        }
    }
}

```

Fase 6. Generación de paquete de instalación

El sistema operativo Android permite la instalación de aplicaciones de código abierto, para esto es necesario empaquetar la aplicación con un formato de archivo APK (Application Package File), El entorno de desarrollo de Eclipse ofrece una herramienta para la generación del paquete de instalación con el formato requerido para aplicaciones bajo la plataforma de Android. Generado el archivo APK, en necesario transferir el archivo al dispositivo móvil y proceder a su instalación.

Etapa 5: Modo de prueba.

El modo prueba es necesario para la evaluación de la funcionalidad del prototipo construido. Se aplican pruebas utilizando todos los mecanismos de hardware y software mencionados. Las siguientes tablas muestran los tipos de experimentos realizados.

Tabla 1. Pruebas de distancia entre el vehículo y el dispositivo de control

PRUEBAS VEHÍCULO	
Distancia	Tiempo de respuesta
50 metros, obstáculos 10 cm grosor	No hay reacción
50 metros, obstáculos no mayor a 5 cm de grosor.	10 segundos
50 metros, sin obstáculos	Entre 1 y 8 segundos
10 metros, obstáculos 10 cm grosor	5 segundos aprox.
10 metros, obstáculos no mayor a 5 cm de grosor.	2 y 3 segundos aprox.
10 metros, sin obstáculos	1 y 3 segundos aprox.
10 metros, sobre obstáculos grosor 10 cm	1 y 3 segundos aprox.
5 metros, obstáculos 10 cm grosor	1 segundo aprox.
5 metros, obstáculos no mayor a 5 cm de grosor.	1 segundo aprox.
5 metros, sin obstáculos	Milisegundos
5 metros, sobre obstáculos grosor 10 cm	1 y 2 segundos aprox.

El contenido de la tabla 1 arroja los tiempos estimados de reacción del prototipo, a cierta distancia y con obstáculos (paredes) de un grosor distinto.

Tabla 2. Prueba en tipos de suelo.

PRUEBAS VEHÍCULO EXPLORADOR		
Tipo de terreno	Distancia recorrida	Observaciones
Liso	Entre 40 y 45 metros	Sobrepasando los 45 metros, fallo en la comunicación entre dispositivos.
Rugoso	Entre 10 y 30 metros	Fallo en el sistema de bandas

Los resultados obtenidos en la tabla 2, muestran la distancia recorrida en dos tipos de suelo.

Tabla 3. Prueba cámara IP, calidad de transmisión y duración de la fuente de energía.

PRUEBAS CAMARA IP			
Ambiente	Duración de la batería	Calidad de transmisión	Observaciones
Día	30 minutos aprox.	Buena	La calidad de video es satisfactoria.
Noche	Entre 20 y 25 minutos aprox.	Regular	El consumo de la batería es mayor debido al uso de LED'S IR (infrarrojo)

En la tabla 3 se muestra los datos obtenidos en base a la transmisión de imágenes en tiempo real de dos tipos de ambientes. Se registra principalmente la duración aproximada de la batería.

MEJORAS A IMPLEMENTAR EN EL PROTOTIPO

Se sugieren las siguientes opciones de mejora para el prototipo vehículo explorador:

- Mejorar el sistema de impulso, implementando servomotores 360°.
- Implementar mejor material en el sistema de orugas.
- Añadir una coraza, para proteger los circuitos que le dan funcionamiento.

- Añadir dispositivos de emergencia, como brazos robóticos evitando quedar atrapado en una misión.
- Mejorar la calidad de transmisión de video.
- Fuente de alimentación autosustentable.
- Ser completamente autónomo.
- Unificar el video con la interfaz
- Reconocimiento de objetos usando técnicas de procesamiento de imágenes.
- Reconstrucción de espacios de manera virtual.

La implementación de actualizaciones ofrece un mayor auge para el ámbito exploratorio.

CONCLUSIÓN

En base a las pruebas realizadas al carro explorador se observó una excelente comunicación con el dispositivo móvil, en la exploración de lugares abiertos y distancias cortas. Sin embargo cuando el carro transitó en un entorno con obstáculos (paredes de concreto) existe un retardo de aproximadamente 5 segundos en el tiempo de respuesta.

Durante un periodo de tiempo promedio de 20 minutos, la transmisión de video de la cámara es satisfactoria.

REFERENCIAS

- [1] Fernández Lancha, M. Á., Fernández Sanz, D., & Valmaseda Plasencia, C. (s.f.). *Universidad Complutense Madrid- Facultad de Informatica*. Recuperado el 11 de Septiembre de 2013, de Universidad Complutense Madrid- Facultad de Informatica.
- [2] Foscam. (2013). Foscam. *Manual Usuario*, 10.
- [3] M., M. (1 de Agosto de 2009). *Microcoro*. Recuperado el 3 de Abril de 2013, de Microcoro: <http://microcoro.blogspot.mx/2009/08/uso-de-mplab-escribir-y-compile-un.html>
- [4] Microchip. (10 de Octubre de 2009). *Microchip*. Recuperado el 15 de Febrero de 2013, de Microchip: <http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en010208>
- [5] Microelectrónicos. (03 de Diciembre de 2011). *Microelectrónicos*. Recuperado el 17 de Febrero de 2013, de Microelectrónicos: <http://www.microelectronicos.net/?p=1075>
- [6] O., P. (1 de Mayo de 2007). *Sparkfun Electronics*. Recuperado el 28 de Abril de 2013, de <https://www.sparkfun.com/tutorials/67>
- [7] Piedra, M. A. (16 de Noviembre de 2009). *Mexatronica*. Recuperado el 30 de Abril de 2013, de <http://mexatronica.blogspot.mx/2009/11/tutorial-para-uso-de-modulo-de.html>
- [8] Piedra, M. A. (03 de Agosto de 2010). *Mexatronica*. Recuperado el 27 de Marzo de 2013, de Mexatronica: <http://foros.mexatronica.com/viewtopic.php?f=3&t=23&sid=1b63cac3c3ce2ca277f439396b39fde9>
- [9] Ruiz, F. (4 de Julio de 2012). *IngenieroFernandoRuiz*. Recuperado el 14 de Febrero de 2013, de Proyectos Mecatronicos: <http://www.ingenierofernandoruiz.com/Puentes%20H/Puente%20H.html>
- [10] Universidad de Oviedo. (s.f.). *Pic, Microcontroladores*. Recuperado el 10 de Abril de 2013, de http://www.unioviedo.es/ate/alberto/Entorno%20MP LAB_v7xx.pdf
- [11] Penagos Plazas, J. R. (2010). *Cómo Programar en Lenguaje C los Microcontroladores PIC16F88, 16F628A y 16F877A*. Quito, Colombia: Empresa TecMikro.
- [12] Tomás Gironés, J. (2013). *El Gran Libro de Android*. España: MARCOMBO, S.A.