**UNIVERSIDAD MAYOR DE SAN ANDRÉS**

**FACULTAD DE TECNOLOGÍA**

**CARRERA DE ELECTRONICA Y TELECOMUNICACIONES**

**EXAMEN DE GRADO**

**NIVEL LICENCIATURA**

**TRABAJO DE APLICACIÓN**

# "SISTEMA DE MONITOREO Y CONTROL VEHICULAR"

**POSTULANTE:** HENRY COARITE MAMANI

**LA PAZ – BOLIVIA**

**2013**

## Resumen

El sistema de monitoreo y control vehicular es un sistema que permite observar la posición del vehículo en una página web, gracias a que el sistema será capaz de informar la temperatura interna del vehículo, la posición, tendrá acceso a las luces del vehículo y al encendido/apagado del motor. Todo esto gracias al micro-controlador PIC16F876A y android.

Un teléfono con S.O. android posee acceso a sus módulos GPS, bluetooh y wifi; es por eso que se aprovecha esta característica para la comunicación con el micro-controlador y el servidor; la comunicación con el micro-controlador se hace vía bluetooh, a fin de recibir o enviar datos al micro-controlador; los módulos wifi, bluetooh y GPS son relativamente costosos, es por eso que no se hace uso de de estos módulos por separado; si no que se aprovecha los módulos incorporados del teléfono móvil con android.

Se hace uso del modulo bluetooh HC-06 para la comunicación del micro-controlador con android, desde el micro-controlador se tiene acceso a las luces del vehículo, temperatura y al encendido o apagado del motor, desde android se obtiene la posición y acceso a la red de datos.

Android se encarga de enviar los datos al servidor, que posteriormente el usuario podrá consultar en cualquier momento la ubicación de su vehículo, temperatura, acceso a las luces y al encendido/ apagado del motor desde una página web.

**Índice** pág.

**Índice de figuras** **Pág.**

## Introducción

Debido al gran incremento de vehículos que se venden y utilizan a nivel mundial, se vuelve una necesidad conocer la ubicación exacta en el planeta y a cualquier momento, ya sea esta por seguridad, por caso de robo o para el control y monitoreo del vehículo. Es por eso que es este proyecto de aplicación describe el desarrollo de un prototipo de un sistema de monitoreo y control vehicular, el cual permite monitorear el vehículo de un usuario; gracias al micro-controlador PIC16F876A se tiene acceso a la alimentación del motor y al acceso de las luces del vehículo como ejemplo, pudiendo estas ser modificadas a gusto del cliente, además se hace el uso del modulo bluetooh HC-06(BT BOARD v1.3) para la comunicación entre el micro-controlador 16F876A y android; y este último es la encarga enviar datos al servidor.

## Antecedentes

En Bolivia existen varias empresas que ofrecen sistemas de localización vehicular, pero son relativamente caras; algunas de estas empresas son:

- **UBICAR** es una red internacional de seguridad vehicular, actualmente está presente casi en toda Latinoamérica; provee soluciones y servicios de localización y protección electrónica de vehículos, personas y bienes.
- **Visión Satelital Tracking SRL** Rastreo Satelital de vehículo y Vigilancia electrónica**.**
- **AVL.** Es un Sistema Electrónico integral que combina tecnología GPS con comunicación inalámbrica a través de la Red GSM. Ésta combinación de tecnologías permite a sus clientes reportar a una o varias bases predeterminadas su posición geográfica y estado, según las necesidades pre-programadas para satisfacer al usuario.

## CAPITULO I

### 1 Planteamiento del problema.

Según los antecedentes presentados, se ha logrado detectar los siguientes problemas que surgen de la necesidad y la oferta en nuestro mercado; más allá de una comparación de ser sistemas de localización extranjeros.

- No existen sistemas de localización económicos, mínimamente se paga 100 Bs por mes, y se debe tener un contrato mayor o igual a 2 años.
- Los sistemas son completamente extranjeros y no se apoya a los productos bolivianos.
- La mayoría de los servicios ofrecidos en Bolivia solo ofrecen localización y no así actuación al momento de robo (ejemplo: detener el motor).

### 1.2 Justificación del trabajo

#### 1.2.1 Justificación Económica

Un sistema de localización que cuente con la característica de que pueda detener el motor es mayor en cuanto al precio, es por eso que la combinación de android y el micro-controlador hacen posible un menor precio de un producto similar.

#### 1.2.2. Justificación Social

En la actualidad uno de los problemas mayores con que la sociedad se enfrenta, es la creciente falta de seguridad de bienes; lo que nos lleva a buscar soluciones al alcance de la sociedad para impedir tales efectos provenientes de: robos y asaltos a los vehículos. Es de esta manera que surge la necesidad de contar con un sistema de localización al alcance de la sociedad

#### 1.2.3. Justificación Tecnológica

Los vertiginosos avances de la electrónica, la informática y las telecomunicaciones, han presionado a la industria para generar sistemas que provean aplicaciones y soluciones de mayor utilidad. En consecuencia nace este

trabajo de aplicación como una solución alternativa a las exigencias del usuario, las cuales combina la tecnología de android con los micro-controladores; además de aprovechar sus módulos incorporados en el mismo teléfono móvil con android, como ser GPS, bluetooh y wifi; que en caso de querer aplicar estos módulos por separado al micro-controlador nos costaría más que el mismo teléfono móvil con android.

## 1.3 Objetivos

### 1.3.1 Objetivo General

Implementar un prototipo de un sistema de monitoreo y control vehicular, que permita ver en un mapa (pagina web) la ubicación del vehículo, basado en el micro-controlador PIC16f876A, modulo bluetooh HC-06(BT BOARD v1.3) y un teléfono móvil con S.O. android v2.3.5 o superior.

### 1.3.2 Objetivos Específicos

a) Realizar la programación del micro-controlador para la comunicación con android a través de su puerto USART.

b) Realizar la programación del micro-controlador para la obtención de temperatura usando el sensor de temperatura LM35.

c) Realizar la programación del micro-controlador para encender y/o apagar la luces del vehículo, y apagado del motor.

d) Realizar la programación de comunicación con el servidor en android.

e) Realizar la programación de la página web para el cliente.

f) Realizar la programación del servidor web para la recepción de datos.

d) Diseñar e implementar el hardware del dispositivo de control del vehicular.

b) Implementar el trabajo de aplicación sobre un vehículo a control remoto de juguete.

## CAPITULO II

## 2 Fundamentación teórica.

### 2.1. Introducción.

El gran progreso tecnológico que han sufrido los sistemas de telecomunicación, desarrolló y la proliferación de Internet, ha incrementado exponencialmente la capacidad para crear información, almacenarla, transmitirla, recibirla, y procesarla; así por ello, en busca de cubrir los objetivos que se plantea el presente proyecto se hace un estudio sobre los sistemas de localización vehicular y elementos necesarios para el desarrollo del trabajo de aplicación.

### 2.2 Sistemas de localización

En 1957, la Unión Soviética lanzó al espacio el satélite Sputnik I, que era monitorizado mediante la observación del efecto Doppler de la señal que transmitía. Debido a este hecho, se comenzó a pensar que, de igual modo, la posición de un observador podría ser establecida mediante el estudio de la frecuencia Doppler de una señal transmitida por un satélite cuya órbita estuviera determinada con precisión.

La armada estadounidense rápidamente aplicó esta tecnología, para proveer a los sistemas de navegación de sus flotas de observaciones de posiciones actualizadas y precisas. Así surgió el sistema TRANSIT, que quedó operativo en 1964, y hacia 1967 estuvo disponible, además, para uso comercial.

Las actualizaciones de posición, en ese entonces, se encontraban disponibles cada 40 minutos y el observador debía permanecer casi estático para poder obtener información adecuada.

Posteriormente, en esa misma década y gracias al desarrollo de los relojes atómicos, se diseñó una constelación de satélites, portando cada uno de ellos uno de estos relojes y estando todos sincronizados con base en una referencia de tiempo determinado.

En 1973 se combinaron los programas de la Armada y el de la Fuerza Aérea de los Estados Unidos (este último consistente en una técnica de transmisión codificada que proveía datos precisos usando una señal modulada con un código de PRN (Pseudo-Random Noise: ruido pseudo-aleatorio), en lo que se conoció como Navigation Technology Program (programa de tecnología de navegación), posteriormente renombrado como NAVSTAR GPS.

Entre 1978 y 1985 se desarrollaron y lanzaron once satélites prototipo experimentales NAVSTAR, a los que siguieron otras generaciones de satélites, hasta completar la constelación actual, a la que se declaró con capacidad operacional inicial en diciembre de 1993 y con capacidad operacional total en abril de 1995. En 2009, este país ofreció el servicio normalizado de determinación de la posición para apoyar las necesidades de la OACI, y ésta aceptó el ofrecimiento.

### 2.3. Evolución del sistema GPS

El GPS está evolucionando hacia un sistema más sólido (GPS III), con una mayor disponibilidad y que reduzca la complejidad de las aumentaciones GPS. Algunas de las mejoras previstas comprenden:

- Incorporación de una nueva señal en L2 para uso civil.
- Adición de una tercera señal civil (L5): 1176,45 MHz
- Protección y disponibilidad de una de las dos nuevas señales para servicios de Seguridad Para la Vida (SOL).
- Mejora en la estructura de señales.
- Incremento en la potencia de señal (L5 tendrá un nivel de potencia de –154 dB).
- Mejora en la precisión (1 – 5 m).
- Aumento en el número de estaciones de monitorización: 12 (el doble)
- Permitir mejor interoperabilidad con la frecuencia L1 de Galileo.

El programa GPS III persigue el objetivo de garantizar que el GPS satisfaga requisitos militares y civiles previstos para los próximos 30 años. Este programa se está desarrollando para utilizar un enfoque en 3 etapas (una de las etapas de transición es el GPS II); muy flexible, permite cambios futuros y reduce riesgos. El

desarrollo de satélites GPS II comenzó en 2005, y el primero de ellos estará disponible para su lanzamiento en 2012, con el objetivo de lograr la transición completa de GPS III en 2017. Los desafíos son los siguientes:

- Representar los requisitos de usuarios, tanto civiles como militares, en cuanto a GPS.
- Limitar los requisitos GPS III dentro de los objetivos operacionales.
- Proporcionar flexibilidad que permita cambios futuros para satisfacer requisitos de los usuarios hasta 2030.
- Proporcionar solidez para la creciente dependencia en la determinación de posición y de hora precisa como servicio internacional.

El sistema ha evolucionado y de él han derivado nuevos sistemas de posicionamiento IPS-2 se refiere a Inertial Positioning System, sistema de posicionamiento inercial, un sistema de captura de datos, que permite al usuario realizar mediciones a tiempo real y en movimiento, el llamado Mobile Mapping. Este sistema obtiene cartografía móvil 3D basándose en un aparato que recoge un escáner láser, un sensor inercial, sistema GNSS y un odómetro a bordo de un vehículo. Se consiguen grandes precisiones, gracias a las tres tecnologías de posicionamiento: IMU + GNSS + odómetro, que trabajando a la vez dan la opción de medir incluso en zonas donde la señal de satélite no es buena.

### 2.4 Características del micro-controlador

Los **PIC16F87X** forman una subfamilia de micro-controladores PIC (*Peripheral Interface Controller*) de gama media de 8 bits, fabricados por Microchip Technology Inc.

Cuentan con memoria de programa de tipo EEPROM Flash mejorada, lo que permite programarlos fácilmente usando un dispositivo programador de PIC. Esta característica facilita sustancialmente el diseño de proyectos, minimizando el tiempo empleado en programar los micro-controladores.

Esta subfamilia consta de los siguientes modelos que varían de acuerdo a prestaciones, cantidad de terminales y encapsulados:

- PIC16F870
- PIC16F871
- PIC16F872
- PIC16F873A
- PIC16F874A
- PIC16F876A
- PIC16F877A

**Figura 2,1**
El micro-controlador PIC16F876A
**Fuente:** http:// preyproject.com/

La "A" al final de los modelos PIC16F873A, PIC16F874A, PIC16F876A y PIC16F877A indica que estos modelos cuentan con módulos de comparación analógicos. El hecho de que se clasifiquen como micro-controladores (MCU) de 8 bits hace referencia a la longitud de los datos que manejan las instrucciones, y que se corresponde con el tamaño del bus de datos y el de los registros de la CPU.

Las características principales de estos dispositivos son:

- CPU de arquitectura RISC (*Reduced Instruction Set Computer*).
- Set de 35 instrucciones.
- Frecuencia de reloj de hasta 20MHz (ciclo de instrucción de 200ns).
- Todas las instrucciones se ejecutan en un único ciclo de instrucción, excepto las de salto.
- Hasta 8K x 14 palabras de Memoria de Programa FLASH (ver tabla a continuación).

- Hasta 368 x 8 bytes de Memoria de Datos tipo RAM (ver tabla a continuación).
- Hasta 256 x 8 bytes de Memoria de Datos tipo EEPROM (ver tabla a continuación).
- Hasta 15 fuentes de Interrupción posibles.
- 8 niveles de profundidad en la Pila hardware.
- Modo de bajo consumo (Sleep).
- Tipo de oscilador seleccionable (RC, HS, XT, LP y externo).
- Rango de voltaje de operación desde 2,0V a 5,5V.
- Conversór Analógico/Digital de 10 bits multicanal.
- 3 Temporizadores.
- Watchdog Timer o Perro Guardián.
- 2 módulos de captura/comparación/PWM.
- Comunicaciones por interfaz USART (Universal Synchronous Asynchronous Receiver Transmitter).
- Puerto Paralelo Esclavo de 8 bits (PSP).
- Puerto Serie Síncrono (SSP) con SPI e I²C.

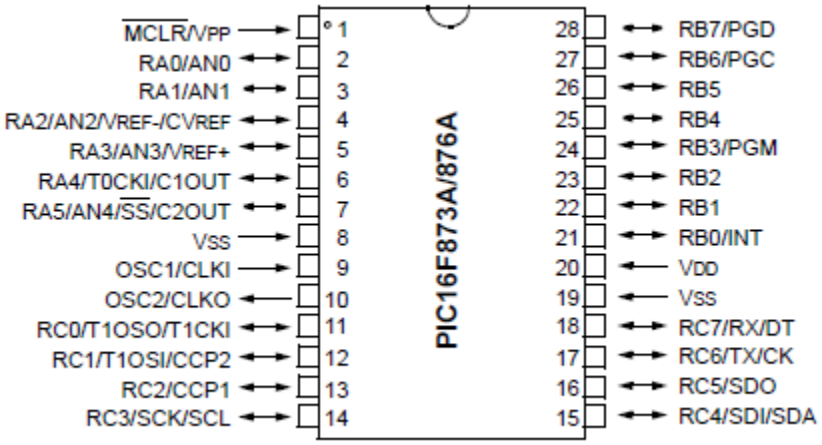



**Figura 2,2**
Distribución de del micro-controlador PIC16F876A
**Fuente:** http://search.datasheetcatalog.net/

### 2.4.1 Circuitería externa auxiliar

Para que el micro-controlador sea capaz de funcionar en cualquier proyecto, se necesita al menos la siguiente circuitería externa:

- La alimentación.
- El reloj (oscilador).

También, y de manera opcional:

- El circuito de reinicio.

### 2.4.2 La alimentación

Los dispositivos de la familia PIC16F87X admiten un amplio rango de tensiones de alimentación, que va de 2,0 V a 5,5 V. La tensión a la cual se alimenten determinará la frecuencia máxima de trabajo.

La potencia máxima disipada es de 1 W y se calcula mediante la siguiente fórmula:

$$P_{disipada} = V_{DD}(I_{DD} - \Sigma I_{OH}) + \Sigma[(V_{DD} - V_{OH})I_{OH}] + \Sigma(V_{OL}I_{OL})$$

Donde:

- $V_{DD}$ es la tensión suministrada por la fuente de alimentación.
- $I_{OH}$ es la corriente suministrada por las salidas del PIC en estado alto.
- $I_{OL}$ es la corriente absorbida por las salidas del PIC en estado bajo.
- $V_{OH}$ es la tensión entregada por los terminales en estado alto.
- $V_{OL}$ es la tensión presente en los terminales en estado bajo.

### 2.4.3 El reloj

El reloj u oscilador se utiliza para generar la base de tiempo del micro-controlador. Para la conexión del oscilador se emplean los terminales OSC1 y OSC2 del dispositivo.

Los micro-controladores PIC16F87X emplean por cada ciclo de instrucción cuatro ciclos de reloj. Esto significa que por ejemplo, si el micro-controlador debe ejecutar un programa de 1000 instrucciones con un reloj de 10 MHz (periodo de reloj de 100 ns), el tiempo total que empleará para ejecutar todo el programa (asumiendo que todas las instrucciones fueran de un ciclo de instrucción) será de:

$$T = \frac{1000 * 4}{10 * 10^6} = 400\mu s$$

La serie PIC16F87X puede trabajar a una frecuencia de reloj máxima de 20 MHz Esto quiere decir que, a esta frecuencia, el tiempo necesario para ejecutar las instrucciones de un ciclo de instrucción es de 200 ns, y de 400 ns para las de dos ciclos de instrucción (instrucciones de salto).

La señal de reloj puede generarse mediante una red resistencia-condensador, un cristal de cuarzo piezoeléctrico o un resonador cerámico, aunque empleando cristales de cuarzo se consiguen frecuencias de oscilación muy exactas, lo cual es útil para calcular tiempos de ejecución, temporizaciones precisas, etc.
Estos microprocesadores permiten escoger entre cinco tipos distintos de osciladores:

- LP (Low Power): reloj de bajo consumo, estable, con frecuencia de oscilación de hasta 200 kHz.
- XT (Xtal, Crystal): estable, frecuencia de oscilación de hasta 4 MHz.
- HS (High Speed): estable, frecuencia de oscilación de hasta 20 MHz.
- RC (Resistor/Condensador): frecuencia de oscilación dependiente de resistencia, condensador, voltaje de alimentación y temperatura de trabajo. Es el tipo más económico, pero también el más inestable.
- Externo: cuando la señal de reloj es externa, generada por otro circuito.

Los modos LP, XT y HS suponen la conexión de un cristal de cuarzo o resonador cerámico entre las patitas OSC1/CLKIN y OSC2/CLKOUT del dispositivo, mientras que el modo RC y Externo solo ocupan la patita OSC1/CLKIN.

### 2.4.4 El circuito de reinicio

El terminal MCLR (Master Clear) debe estar a valor lógico alto para que el dispositivo funcione normalmente, esto es, sin irse a reinicio. Con un valor lógico bajo el dispositivo se reinicia, comenzando la ejecución desde el principio del programa que tenga cargado en memoria.

Lo más práctico, para facilitar el hecho de poder realizar un reinicio manual, es utilizar un pulsador (pulsador de reinicio), similar al que se puede encontrar en la

mayoría de ordenadores. El fabricante recomienda que se intercale una resistencia de 50 a 100 ohmios entre el pulsador y el pin MCLR, para evitar posibles corrientes inducidas de más de 80 mA que podrían bloquear el dispositivo cuando este se lleva a masa (reinicio).

Debido a que el pulsador no produce una respuesta instantánea, producto de los rebotes de éste (transitorio), se generan una serie de pulsos hasta quedar estabilizado en su estado permanente. Para evitar esto se puede usar un condensador instalado en paralelo con la entrada MCLR (filtro pasa bajo).
También resulta muy efectivo el uso de un filtro pasa alto para hacer la señal de reinicio independiente del tiempo en que se presiona el pulsador. Inmediatamente después de pulsar el pulsador el micro-controlador se reinicia, sin tener en cuenta cuánto tiempo se mantiene presionado dicho pulsador.

### 2.4.5 Memoria interna

Existen tres bloques bien diferenciados de memoria. Estos son:

- Memoria de programa EEPROM Flash: es el lugar físico donde se guarda el programa de usuario. Es de tipo no volátil.
- Memoria de datos SRAM: es el lugar físico donde se guardan datos. Es de tipo volátil.
- Memoria de datos EEPROM: es el lugar físico donde se guardan datos. Es de tipo no volátil.

### 2.4.6 Memoria de datos SRAM

Esta memoria es de tipo volátil, lo que significa que no conserva su contenido después de un apagado de alimentación.
En esta memoria se encuentran los registros de funciones especiales (SFR) y los registros de propósito general (GPR), y está particionada en cuatro bancos (0, 1, 2 y 3), seleccionables independientemente. El banco 0 es el banco seleccionado por defecto cuando se alimenta al micro-controlador.

### 2.4.7 Registros de funciones especiales (SFR)

Todos los micro-controladores cuentan con registros internos que permiten controlar y supervisar las funciones y recursos disponibles del dispositivo.

Los registros en los micro-controladores PIC se encuentran en un espacio especial de la memoria de datos, el SFR (Special Function Registers). En los dispositivos PIC16F87X estos registros son de 8 bits, la mayoría de lectura y escritura. Se puede acceder a los bits de manera individual, o bien a todo el registro a la vez.

Determinados pares de registros tienen funciones especiales para las cuales se pueden considerar unidos en un único registro de 16 bits, aunque físicamente siguen estando separados.

### 2.4.8 El registro STATUS

El registro de estado (STATUS) es uno de los más importantes y empleados en el micro-controlador.

Proporciona información acerca del resultado de operaciones aritméticas, operaciones lógicas y causa de reinicios, además de permitir la selección del banco de memoria de datos.

### 2.4.9 El registro W

El registro de trabajo W (Working Register) es un registro relevante especial de 8 bits que participa en la mayoría de instrucciones. A diferencia de los SFR, se encuentra dentro de la misma CPU, y puede ser accedido tanto para lectura como para escritura.

## 2.5 Módulo Bluetooth HC-06

El **módulo Bluetooth HC-06** es un dispositivo que permite la comunicación vía bluetooh como si se tratase de un puerto serial; el modulo bluetooh utilizado para este proyecto (JY-MCU BT-BOARD v1.3) este es un circuito que incluye el modulo que no requiere alimentación de 3.3 voltios de corriente continua, ya que el circuito cuenta con reguladores de tensión internamente.

**Figura 2,3**
Modulo bluetooh HC-06
**Fuente:** http://www.43oh.com/

Una de las ventajas principales ventajas del **módulo HC-06**, además de su pequeño tamaño y sus buenas características de transmisión y recepción que le brindan un alcance muy amplio (por tratarse de un sistema local **Bluetooth**), es el **bajo consumo de corriente** que posee tanto en funcionamiento, como en modo de espera, es decir, alimentado con energía, pero sin conexión o enlace a otro dispositivo, por ejemplo, un móvil con SO **Android**. Otra característica interesante de este módulo es que una vez que ha realizado un enlace con otro dispositivo es capaz de recordarlo en su memoria y no solicita validación alguna ("*1234*" por defecto), pero si se activa el pin 26 (*KEY*) hacia la tensión de alimentación, esta información se elimina y el **módulo HC-06** solicitará nuevamente la validación del enlace. Otro detalle particular es que su tensión de alimentación de 3,3Volts y su bajo consumo (8mA en transmisión/recepción activa) lo transforman en un dispositivo ideal para trabajar con micro-controladores de la misma tensión de alimentación, logrando de este modo equipos portátiles que pueden ser alimentados durante muchas horas por baterías comunes, demostrando características excepcionales en aplicaciones médicas, o para actividades recreativas donde la fuente energética debe ser liviana y portátil.

Algunas de sus características son:

- Alta sensibilidad, puede alcanzar hasta -80dBm.

- La gama de cambio del poder de salida: -4 - +6dBm.
- Descripción de Función (solución de Bluetooth perfecta )
- Tiene un módulo de EDR; y la gama de cambio de profundidad de modulación: 2Mbps - 3Mbps.
- Tiene una complexión en 2.4GHz antena; el usuario no necesita probar antena.
- Tiene el 8Mbit FLASH externo
- Puede trabajar al voltaje bajo (3.1V~4.2V.)
- La corriente en comunicación tiene 8mA años.
- Puerto estándar USART o USB.
- Protocolo de USB : A toda velocidad USB1.1 Sumiso Con 2.0
- Se puede utilizar este módulo en el SMD.
- Tiene un transceptor de 2.4GHz sin hilos digital.
- Tiene la función de frecuencia adaptivo hopping.
- El circuito de periférico es sencillo.

### 2.6 Sistema operativo Android

Este es un sistema operativo basado en Linux, diseñado principalmente para dispositivos móviles con pantalla táctil como teléfonos inteligentes o tabletas inicialmente desarrollados por Android, Inc., que Google respaldó económicamente y más tarde compró en 2005, Android fue presentado en 2007 junto la fundación del Open Handset Alliance: un consorcio de compañías de hardware, software y telecomunicaciones para avanzar en los estándares abiertos de los dispositivos móviles. El primer móvil con el sistema operativo Android se vendió en octubre de 2008.

### 2.6.1 Historia del S.O.

Fue desarrollado inicialmente por **Android Inc.**, una firma comprada por **Google** en **2005**. Es el principal producto de la **Open Handset Alliance**, un conglomerado de fabricantes y desarrolladores de hardware, software y operadores de servicio.[12] Las unidades vendidas de teléfonos inteligentes con Android se ubican en el primer puesto en los **Estados Unidos**, en el segundo y tercer trimestres de **2010**, con una cuota de mercado de 43,6% en el tercer trimestre. A nivel mundial alcanzó una cuota de mercado del 50,9% durante el cuarto trimestre de 2011, más del doble que el segundo sistema operativo (IOS de Apple, Inc.) con más cuota.

Tiene una gran comunidad de desarrolladores escribiendo aplicaciones para extender la funcionalidad de los dispositivos. A la fecha, se han sobrepasado las 700.000 aplicaciones (de las cuales, dos tercios son gratuitas) disponibles para la tienda de aplicaciones oficial de Android: Google Play, sin tener en cuenta aplicaciones de otras tiendas no oficiales para Android como la tienda de aplicaciones Samsung Apps de Samsung. Google Play es la tienda de aplicaciones en línea administrada por Google, aunque existe la posibilidad de obtener software externamente. Los programas están escritos en el lenguaje de programación Java. No obstante, no es un sistema operativo libre de malware, aunque la mayoría de ello es descargado de sitios de terceros.

El anuncio del sistema Android se realizó el 5 de noviembre de 2007 junto con la creación de la Open Handset Alliance, un consorcio de 78 compañías de hardware, software y telecomunicaciones dedicadas al desarrollo de estándares abiertos para dispositivos móviles. Google liberó la mayoría del código de Android bajo la licencia Apache, una licencia libre y de código abierto.

La estructura del sistema operativo Android se compone de aplicaciones que se ejecutan en un framework Java de aplicaciones orientadas a objetos sobre el núcleo de las bibliotecas de Java en una máquina virtual Dalvik con compilación en tiempo de ejecución. Las bibliotecas escritas en lenguaje C incluyen un administrador de interfaz gráfica (*surface manager*), un framework OpenCore, una base de datos relacional SQLite, una Interfaz de programación de API gráfica

OpenGL ES 2.0 3D, un motor de renderizado WebKit, un motor gráfico SGL, SSL y una biblioteca estándar de C Bionic. El sistema operativo está compuesto por 12 millones de líneas de código, incluyendo 3 millones de líneas de XML, 2,8 millones de líneas de lenguaje C, 2,1 millones de líneas de Java y 1,75 millones de líneas de C++.

**Figura 2,4**
Logotipo de android
**Fuente:** http:// android.com/

### 2.6.2 Etimología de android

Tanto el nombre *Android* (**androide** en español) como **Nexus One** hacen alusión a la novela de **Philip K. Dick *¿Sueñan los androides con ovejas eléctricas?***, que posteriormente fue adaptada al cine como ***Blade Runner***. Tanto el libro como la película se centran en un grupo de androides llamados *replicantes* del modelo Nexus-6.

### 2.6.3 Adquisición por parte de Google

En julio de 2005, Google adquirió Android Inc., una pequeña compañía de Palo Alto, California fundada en 2003. Entre los cofundadores de Android que se fueron a trabajar a Google están Andy Rubin (co-fundador de Danger), Rich Miner (co-fundador de Wildfire Communications, Inc.), Nick Sears (alguna vez VP en T-Mobile), y Chris White (quien encabezó el diseño y el desarrollo de la interfaz en WebTV). En aquel entonces, poco se sabía de las funciones de Android Inc. fuera

de que desarrollaban software para teléfonos móviles. Esto dio pie a rumores de que Google estaba planeando entrar en el mercado de los teléfonos móviles.

En Google, el equipo liderado por Rubin desarrolló una plataforma para dispositivos móviles basada en el núcleo Linux que fue promocionado a fabricantes de dispositivos y operadores con la promesa de proveer un sistema flexible y actualizable. Se informó que Google había alineado ya una serie de fabricantes de hardware y software y señaló a los operadores que estaba abierto a diversos grados de cooperación por su parte.

La especulación sobre que el sistema Android de Google entraría en el mercado de la telefonía móvil se incrementó en diciembre de 2006. Reportes de BBC y The Wall Street Journal señalaron que Google quería sus servicios de búsqueda y aplicaciones en teléfonos móviles y estaba muy empeñado en ello. Medios impresos y en línea pronto reportaron que Google estaba desarrollando un teléfono con su marca.

En septiembre de 2007, InformationWeek difundió un estudio de Evalueserve que reportaba que Google había solicitado diversas patentes en el área de la telefonía móvil.

### 2.6.4 Open Handset Alliance

El 5 de noviembre de 2007 la Open Handset Alliance, un consorcio de varias compañías entre las que están Texas Instruments, Broadcom Corporation, Nvidia, Qualcomm, Samsung Electronics, Sprint Nextel, Intel, LG, Marvell Technology Group, Motorola, y T-Mobile; se estrenó con el fin de desarrollar estándares abiertos para dispositivos móviles. Junto con la formación de la Open Handset Alliance, la OHA estrenó su primer producto, Android, una plataforma para dispositivos móviles construida sobre la versión 2.6 de Linux.

## 2.7 Leguaje de programación JAVA

El lenguaje de programación Java fue originalmente desarrollado por James Gosling de Sun Microsystems (la cual fue adquirida por la compañía Oracle) y publicado en el 1995 como un componente fundamental de la plataforma Java de Sun Microsystems. Su sintaxis deriva mucho de C y C++, pero tiene menos facilidades de bajo nivel que cualquiera de ellos. Las aplicaciones de Java son generalmente compiladas a bytecode (clase Java) que puede ejecutarse en cualquier máquina virtual Java (JVM) sin importar la arquitectura de la computadora subyacente. Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos y basado en clases que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como *WORA*, o "*write once, run anywhere*"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir del 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web, con unos 10 millones de usuarios reportados.

La compañía Sun desarrolló la implementación de referencia original para los compiladores de Java, máquinas virtuales, y librerías de clases en 1991 y las publicó por primera vez en el 1995. A partir de mayo del 2007, en cumplimiento con las especificaciones del Proceso de la Comunidad Java, Sun volvió a licenciar la mayoría de sus tecnologías de Java bajo la Licencia Pública General de GNU. Otros también han desarrollado implementaciones alternas a estas tecnologías de Sun, tales como el Compilador de Java de GNU y el GNU Classpath.

### 2.7.1 Leguaje orientado a objetos

La primera característica, orientado a objetos (“POO”), se refiere a un método de programación y al diseño del lenguaje. Aunque hay muchas interpretaciones para OO, una primera idea es diseñar el software de forma que los distintos tipos de datos que usen estén unidos a sus operaciones. Así, los datos y el código

(funciones o métodos) se combinan en entidades llamadas objetos. Un objeto puede verse como un paquete que contiene el “comportamiento” (el código) y el “estado” (datos). El principio es separar aquello que cambia de las cosas que permanecen inalterables. Frecuentemente, cambiar una estructura de datos implica un cambio en el código que opera sobre los mismos, o viceversa. Esta separación en objetos coherentes e independientes ofrece una base más estable para el diseño de un sistema software. El objetivo es hacer que grandes proyectos sean fáciles de gestionar y manejar, mejorando como consecuencia su calidad y reduciendo el número de proyectos fallidos. Otra de las grandes promesas de la programación orientada a objetos es la creación de entidades más genéricas (objetos) que permitan la reutilización del software entre proyectos, una de las premisas fundamentales de la Ingeniería del Software. Un objeto genérico “cliente”, por ejemplo, debería en teoría tener el mismo conjunto de comportamiento en diferentes proyectos, sobre todo cuando estos coinciden en cierta medida, algo que suele suceder en las grandes organizaciones. En este sentido, los objetos podrían verse como piezas reutilizables que pueden emplearse en múltiples proyectos distintos, posibilitando así a la industria del software a construir proyectos de envergadura empleando componentes ya existentes y de comprobada calidad; conduciendo esto finalmente a una reducción drástica del tiempo de desarrollo. Podemos usar como ejemplo de objeto el aluminio. Una vez definidos datos (peso, maleabilidad, etc.), y su “comportamiento” (soldar dos piezas, etc.), el objeto “aluminio” puede ser reutilizado en el campo de la construcción, del automóvil, de la aviación, etc.

La reutilización del software ha experimentado resultados dispares, encontrando dos dificultades principales: el diseño de objetos realmente genéricos es pobremente comprendido, y falta una metodología para la amplia comunicación de oportunidades de reutilización. Algunas comunidades de “código abierto” (open source) quieren ayudar en este problema dando medios a los desarrolladores para diseminar la información sobre el uso y versatilidad de objetos reutilizables y bibliotecas de objetos.

### 2.7.2 Independencia de la plataforma

La segunda característica, la independencia de la plataforma, significa que programas escritos en el lenguaje Java pueden ejecutarse igualmente en cualquier tipo de hardware. Este es el significado de ser capaz de escribir un programa una vez y que pueda ejecutarse en cualquier dispositivo, tal como reza el axioma de Java, “write once, run anywhere”.

Para ello, se compila el código fuente escrito en lenguaje Java, para generar un código conocido como “bytecode” (específicamente Java bytecode) instrucciones máquina simplificada específica de la plataforma Java. Esta pieza está “a medio camino” entre el código fuente y el código máquina que entiende el dispositivo destino. El bytecode es ejecutado entonces en la máquina virtual (JVM), un programa escrito en código nativo de la plataforma destino (que es el que entiende su hardware), que interpreta y ejecuta el código. Además, se suministran bibliotecas adicionales para acceder a las características de cada dispositivo (como los gráficos, ejecución mediante hebras o threads, la interfaz de red) de forma unificada. Se debe tener presente que, aunque hay una etapa explícita de compilación, el bytecode generado es interpretado o convertido a instrucciones máquina del código nativo por el compilador JIT (Just In Time).

Hay implementaciones del compilador de Java que convierten el código fuente directamente en código objeto nativo, como GCJ. Esto elimina la etapa intermedia donde se genera el bytecode, pero la salida de este tipo de compiladores sólo puede ejecutarse en un tipo de arquitectura.

La licencia sobre Java de Sun insiste que todas las implementaciones sean “compatibles”. Esto dio lugar a una disputa legal entre Microsoft y Sun, cuando éste último alegó que la implementación de Microsoft no daba soporte a las interfaces RMI y JNI además de haber añadido características ‘’dependientes’’ de su plataforma. Sun demandó a Microsoft y ganó por daños y perjuicios (unos 20 millones de dólares) así como una orden judicial forzando la acotación de la licencia de Sun. Como respuesta, Microsoft no ofrece Java con su versión de sistema operativo, y en recientes versiones de Windows, su navegador Internet

Explorer no admite la ejecución de applets sin un conector (o plugin) aparte. Sin embargo, Sun y otras fuentes ofrecen versiones gratuitas para distintas versiones de Windows.

Las primeras implementaciones del lenguaje usaban una máquina virtual interpretada para conseguir la portabilidad. Sin embargo, el resultado eran programas que se ejecutaban comparativamente más lentos que aquellos escritos en C o C++. Esto hizo que Java se ganase una reputación de lento en rendimiento. Las implementaciones recientes de la JVM dan lugar a programas que se ejecutan considerablemente más rápido que las versiones antiguas, empleando diversas técnicas, aunque sigue siendo mucho más lento que otros lenguajes.

La portabilidad es técnicamente difícil de lograr, y el éxito de Java en ese campo ha sido enorme. Aunque es de hecho posible escribir programas para la plataforma Java que actúen de forma correcta en múltiples plataformas de distinta arquitectura, el gran número de estas con pequeños errores o inconsistencias llevan a que a veces se parodie el eslogan de Sun, "Write once, run anywhere" como "Write once, debug everywhere" (o “Escríbelo una vez, ejecútalo en cualquier parte” por “Escríbelo una vez, depúralo en todas partes”).

El concepto de independencia de la plataforma de Java cuenta, sin embargo, con un gran éxito en las aplicaciones en el entorno del servidor, como los Servicios Web, los Servlets, los Java Beans, así como en sistemas empotrados basados en OSGi, usando entornos Java empotrados.

### 2.8 Java script

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente equipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo. JavaScript se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).
Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. JavaScript se interpreta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML. Una cuarta edición está en desarrollo e incluirá nuevas características tales como paquetes, espacio de nombres y definición explícita de clases.

**2.9 Jquery.**

jQuery es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones (FLV) y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC. jQuery es la biblioteca de JavaScript más utilizada. jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privativos. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

Las empresas Microsoft y Nokia anunciaron que incluirán la biblioteca en sus plataformas. Microsoft la añadirá en su IDE Visual Studio y la usará junto con los frameworks ASP.NET AJAX y ASP.NET MVC, mientras que Nokia los integrará con su plataforma Web Run-Time.

## CAPITULO III

# 3 DESARROLLO DEL TRABAJO

## 3.1 Introducción

El diseño del modelo del sistema comienza con la descripción y arquitectura del sistema. El diseño se divide básicamente en dos partes el cliente y el servidor; en este caso el cliente viene a ser la persona que desea monitorizar su vehículo, y el servidor viene a ser aplicación android que es la que se comunica con el micro-controlador; este será instalado en el vehículo.

El micro-controlador será la encargada de enviar datos al teléfono móvil (servidor); esta estará recibirá los datos vía bluetooh y este finalmente enviara los datos al cliente cuando los solicite.
El cliente contara con una aplicación (pagina web) en la cual podrá monitorizar el vehículo.

El software del micro-controlador se hace con el lenguaje C, con un entorno de desarrollo que es CCS; el software de aplicación tanto cliente como servidor se realiza con el lenguaje de programación JAVA.

## 3.2 Descripción general del sistema

El sistema consiste en un sistema de localización remota de vehículos, para tales efectos se hace uso de un teléfono con S.O. android, un servidor, el micro-controlador 16F876A, LM35, 7805, TIP31, oscilador de 110592 Hz y resistencias eléctricas.

El cliente (usuario final) tendrá acceso al monitoreo y control de vehículo; podrá visualizar su vehículo en un mapa desde una página web con acceso a las luces, temperatura y al apagado del motor de dicho vehículo. Para el desarrollo de la página web se hace uso de los lenguajes de: html, java-script y JAVA estos lenguajes se combinan para tener la página web de cliente.


**Figura 3,1**
Pagina web del cliente
**Fuente:** localhost/ratser

Se hace las peticiones con los métodos get y post de jquery, estos métodos nos ayudan a establecer valores en los leds y a obtener el valor de temperatura del vehículo.

A continuación se tiene el código.

```
$(document).ready(function ()
{
    var modelView = new ModelView("canvasVisor");
    var mapa = new Mapa("divMapa");
    $(window).bind("load", function()
    {   modelView.loadModeloObj(
"rastercar/vehiculo/camioneta/camioneta.obj",
"rastercar/vehiculo/camioneta/camioneta.mtl",35);
    });
    $("#btnEnviarComando").click(function()
    {
        var comando =
        {
            ledIzqFrontal:1,
            ledDerFrontal:1,
            ledIzqTrasero:0,
            ledDerTrasero:1,
            hablitarMotor:0
        };
        $.postJSON("ser/ajson/comando",comando,
```

```
        function(jsonResponse)
        {    console.log(jsonResponse);
        });
});
function esperarTrack()
{
    $.ajax({
        dataType: "json",
        url: "ser/ajson/track",
        data: {},
        dataType: "json",
        cache:false,
        timeout:60*1000,
        success: function(track)
        {     procesarTrack(track);
        },    error:function()
        {     esperarTrack();
        }
    });
}
```

Por parte del servidor también está el dispositivo actuador los cuales está compuesto por el micro-controlador y sus componentes, y android.

En la grafica de abajo se puede observar el funcionamiento del sistema.

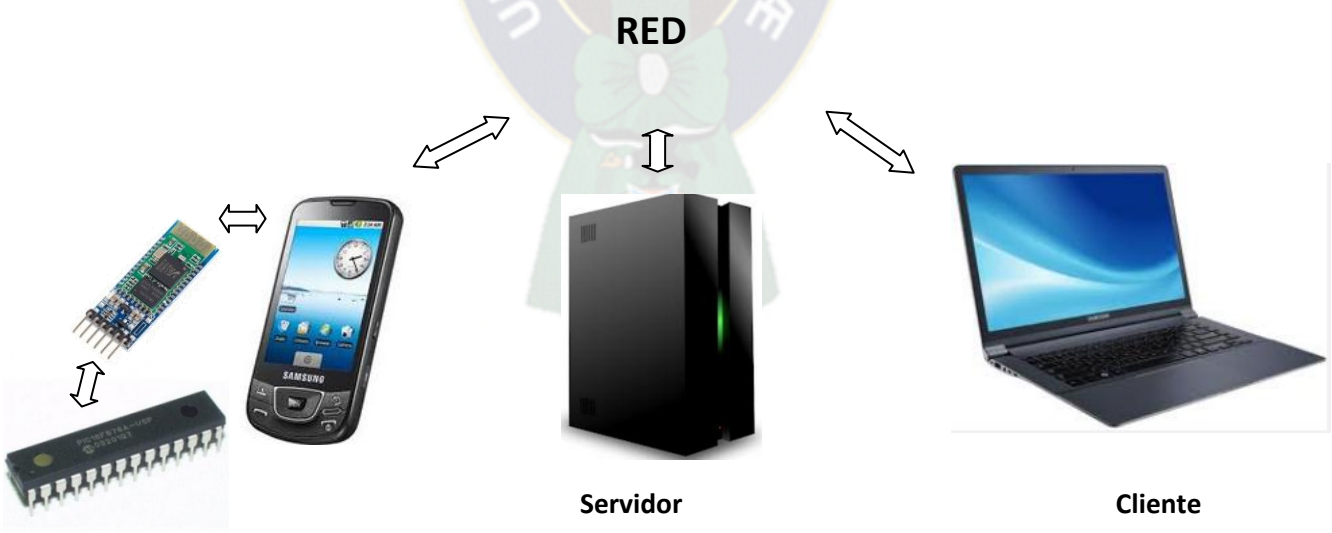



**Figura 3.2**
Descripción grafica del sistema
**Fuente:** Elaboración propia

Java-android; posee métodos de programación que facilitan la comunicación con dispositivos como ser el HC-06; la aplicación android cuenta de un hilo de conexión que escucha el puerto de comunicaciones vía bluetooh.
A continuación se muestra el diagrama de bloques del sistema.

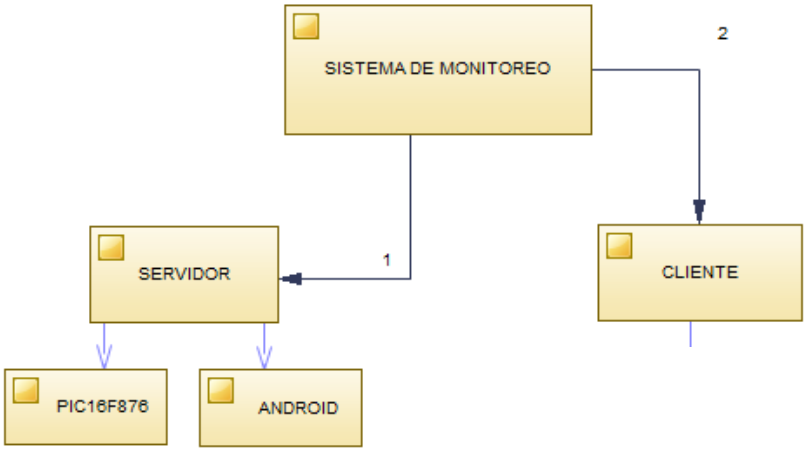



**Figura 3,3**
Diagrama de bloques
**Fuente:** Elaboración propia

## 3.2 Descripción de los bloques

### 3.3.1 El bloque PIC16F876A

El micro-controlador es el encargado de leer la temperatura del vehículo haciendo uso del modulo CAD del micro-controlador y este es enviada al teléfono celular vía puerto USART, como mencione anteriormente el modulo bluetooh consta de un puerto serie; los pines de TX y RX son conectados directamente a este modulo y posteriormente recibidos por el teléfono celular con S.O. android. Los voltajes manejados por el modulo HC-06(JY-MCU) son niveles TTL, por lo que se pueden conectar directamente al puerto serie del micro-controlador.

Para enviar los datos desde el micro-controlador al teléfono móvil, se deben emparejar el modulo bluetooh con el teléfono móvil que posteriormente se explica en la descripción del bloque android.

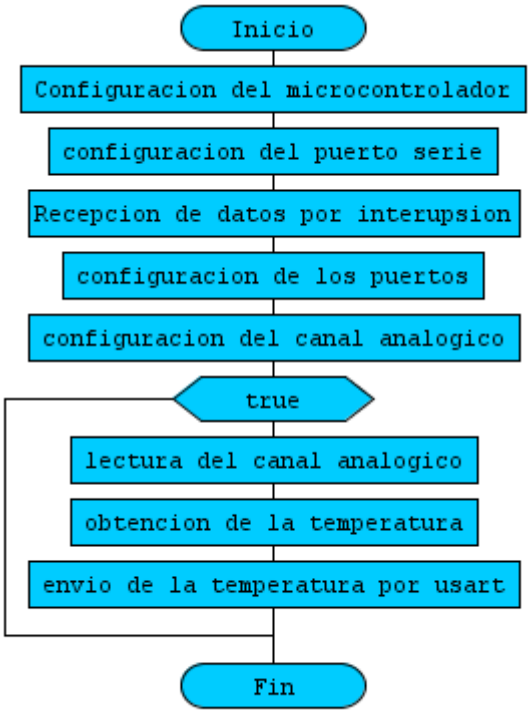



**Figura 3,4**
Diagrama de flujo para el micro-controlador
**Fuente:** Elaboración propia

El micro-controlador es programado con el lenguaje de programación C, bajo el entorno de desarrollo de PCWHD Compiler más conocido como pic C compiler. A continuación se tiene el código del micro-controlador.

```
#include <16F876A.h>
#fuses HS,NOWDT
#device adc=10
#use delay(clock=20000000)
#use rs232(baud=9600, xmit=pin_c6, rcv=pin_c7, bits=8, parity=N)
#include <stdlib.h>
int16 valor;char cad_dato[5];char inst;
#int_rda
void serial_isr()
{
   inst=getc();
```

```
    if(inst=='0')//encender led 0
      output_bit(PIN_B7,1);
    if(inst=='1')//apagar led 0
      output_bit(PIN_B7,0);
    if(inst=='2')//encender led 1
      output_bit(PIN_B6,1);
    if(inst=='3')//Apagar led 1
      output_bit(PIN_B6,0);
    if(inst=='4')//Encender led 2
     output_bit(PIN_B5,1);
    if(inst=='5')//Apagar led 2
     output_bit(PIN_B5,0);
    if(inst=='6')//Encender led 3
     output_bit(PIN_B4,1);
    if(inst=='7')//Apagar led 3
     output_bit(PIN_B4,0);
    if(inst=='8')//Inhabilitar carro
     output_bit(PIN_B3,1);
    if(inst=='9')//Habiliar carro
     output_bit(PIN_B3,0);
    putc(inst);
}
void main()
{
  delay_ms(1000);
  set_tris_b(0x00);
  setup_adc(ADC_CLOCK_INTERNAL);   //Reloj interno RC
  setup_adc_ports(AN0);
  set_adc_channel(0);
  enable_interrupts(INT_RDA);
  enable_interrupts(global);
  output_bit(PIN_B7,1);
```

```
  output_bit(PIN_B6,1);
  output_bit(PIN_B5,1);
  output_bit(PIN_B4,1);
  output_bit(PIN_B3,1);
  do {
       valor=read_adc();
       delay_ms(1000);
       itoa(valor,10, cad_dato);
       puts(cad_dato);
   }while(true);
}
```

Una vez compilada el código del micro-controlador, se hace la simulación en el simulador de circuitos ISIS de PROTEUS.

En el siguiente circuito se puede observar la distribución de pines.

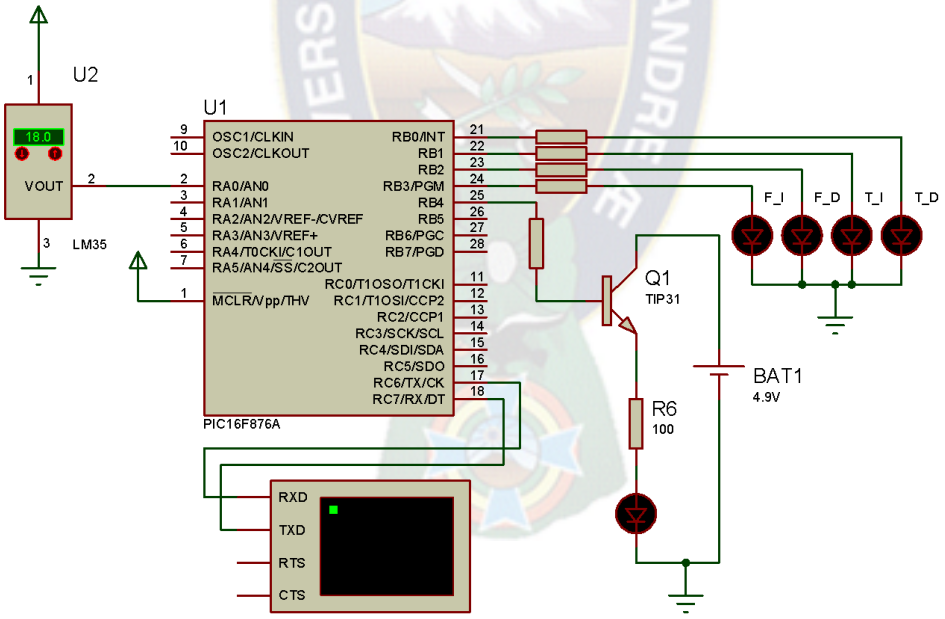



**Figura 3,5**
Circuito del sistema
**Fuente:** Simulador Isis

Para la alimentación correcta del micro-controlador se ha tomado de una fuente externa, en este caso una batería de 9 voltios, que con el uso de circuito integrado

7805 se tiene una fuente de alimentación estable de 5 voltios de corriente continua para el micro-controlador.

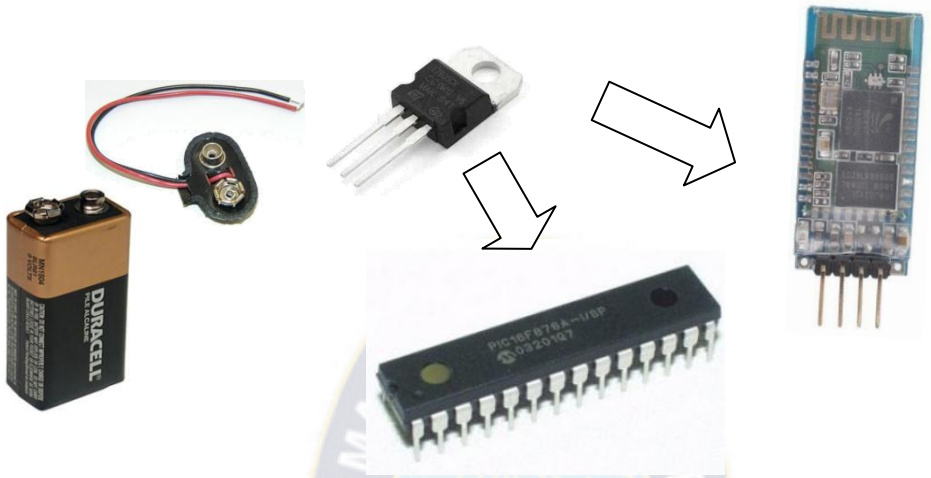



**Figura 3.6**
Grafica de alimentación
**Fuente:** Elaboración propia

### 3.3.2 El bloque Android

El bloque android es la encargada de recibir datos desde el micro-controlador, enviar datos al servidor y la obtención de posición con el GPS incorporado en el teléfono móvil.

Para recibir y/o enviar datos vía bluetooh al modulo HC-06, antes que nada se deben emparejar los dispositivos, primeramente proveer de una alimentación correcta al modulo bluetooh en sus pines VCC y GND tal como se detalla en la hoja técnica de los anexos, consta de un led indicador de conectividad, este led estará parpadeando en caso de que no esté conectado con ningún dispositivo y si estuviere conectado a otro dispositivo no podríamos emparejarlo.

Led rojo de estado debe estar parpadeando.

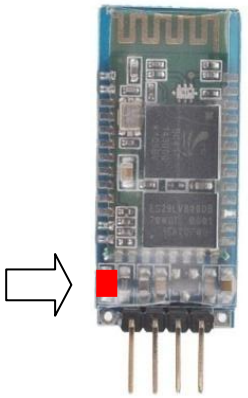


**Figura 3,7**
Indicación de no conectado del modulo bluetooh
**Fuente:** Elaboración propia

El modulo bluetooh HC-06, está en modo visible por defecto; por lo que debemos explorar con el teléfono móvil y emparejarlo.

Teléfono emparejado con el modulo bluetooh

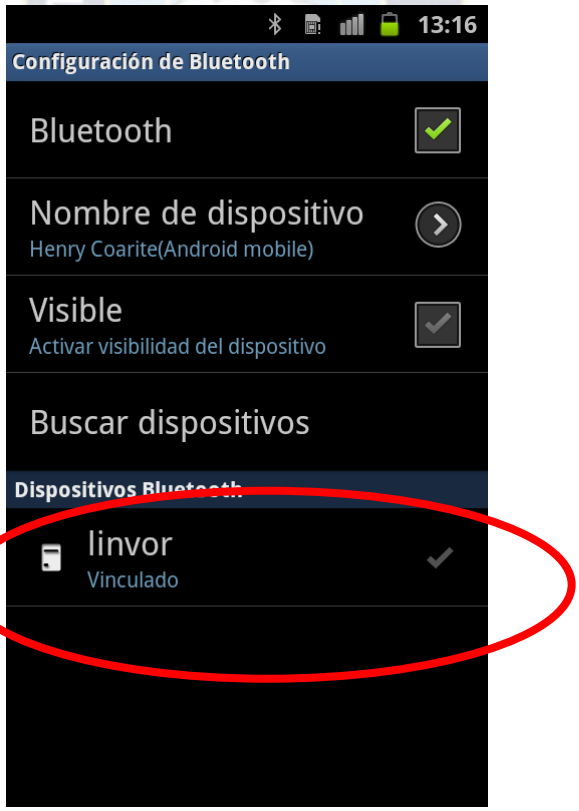



**Figura 3.8**
Exploración de dispositivos bluetooh
**Fuente:** Elaboración propia

Una vez encontrado el dispositivo emparejamos con el modulo; la contraseña de este modulo por defecto es: 1234.

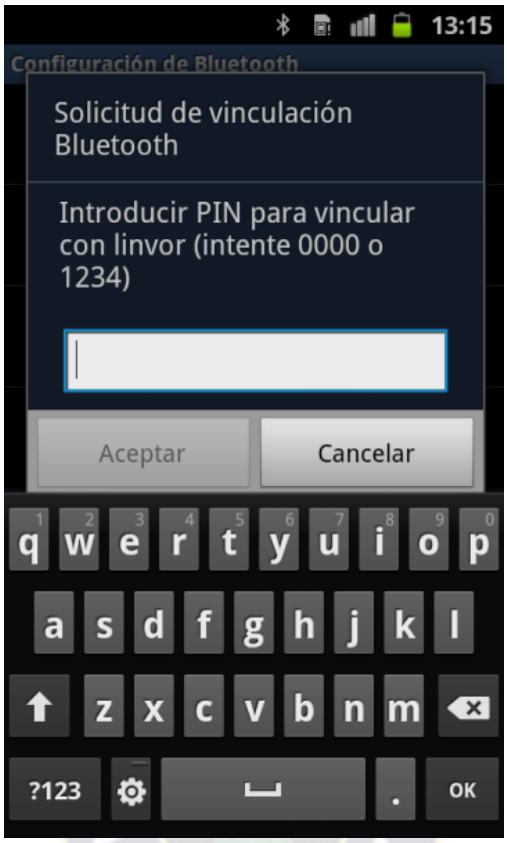



**Figura 3.9**
Emparejamiento de dispositivos bluetooh
**Fuente:** Eclipse

Una vez emparejada se tiene la aplicación android, esta es encargada de recibir datos desde el micro-controlador y enviarlos al servidor. La aplicación se ha desarrollado bajo el entorno de desarrollo eclipse, este es una herramienta de desarrollo para java.

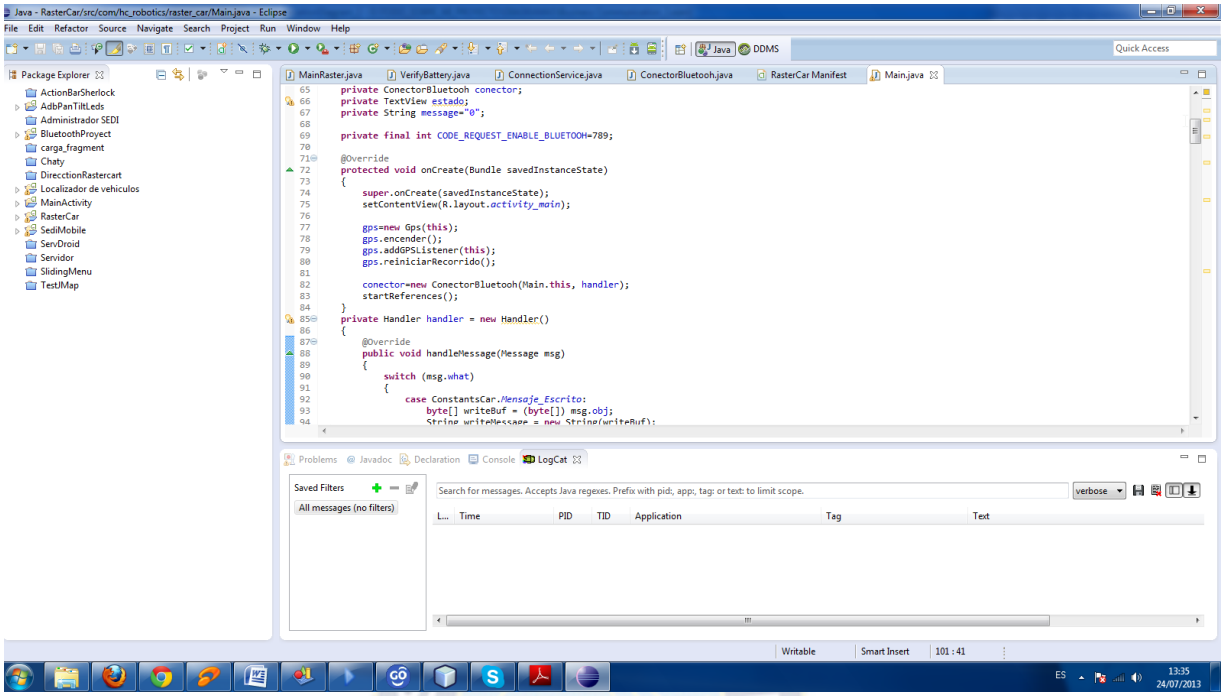


**Figura 3.10**
Pantalla de desarrollo para android
**Fuente:** Eclipse

Se ha hecho el uso del lenguaje de programación java y a continuación se tiene parte del código.

```
public class Main extends Activity  implements GPSListener
{
        private ToggleButton led1,led2, led3,led4;
        private ToggleButton habilitacionMotor;
        private Button connectBluetooh;
        private Gps gps;
        private Location location;
        private Dialog dialog;
        private ConectorBluetooh conector;
        private TextView estado;
        private String message="0";
```

```
    private final int CODE_REQUEST_ENABLE_BLUETOOH=789;
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_main);
            gps=new Gps(this);
            gps.encender();
            gps.addGPSListener(this);
            gps.reiniciarRecorrido();
            conector=new ConectorBluetooh(Main.this, handler);
            startReferences();
    }
    private Handler handler = new Handler()
    {
  @Override
  public void handleMessage(Message msg)
  {
     switch (msg.what)
     {
            case ConstantsCar.Mensaje_Escrito:
            byte[] writeBuf = (byte[]) msg.obj;
            String writeMessage = new String(writeBuf);
            break;
            case ConstantsCar.Mensaje_Leido:
            location=gps.getUltimaPosicion();
        break;
     }
  }
};
```

Java posee métodos que nos facilitan la programación de aplicaciones de comunicación con módulos bluetooh como ser el HC-06, dicho documento extenso y completo puede ser obtenido desde la página de google developers, que también se menciona en la bibliografía.


**Figura 4,11**
Programa de aplicación
**Fuente:** Eclipse

### 3.3.3 Bloque Servidor

El servidor es el encargado de recepciónar los datos del teléfono móvil, estos son mantenidos en la memoria del servidor para informar al cliente.
El servidor es un servlet que escucha las peticiones del teléfono y las guarda en su memoria, el código del servidor es el siguiente.

```
@WebServlet(name = "Rastreo", urlPatterns = {"/rastreo_vehiculo"})
public class Rastreo extends HttpServlet {
    /**
     * Processes requests for both HTTP
     * <code>GET</code> and
     * <code>POST</code> methods.
     *
     * @param request servlet request
```

```
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
      Rastreo i;
          protected void processRequest(HttpServletRequest request,
          HttpServletResponse response)
    {
        PrintWriter out = null;
        i=new Rastreo();
        try
        {
            response.setContentType("text/html;charset=UTF-8");
            out = response.getWriter();
            final String temp = request.getParameter("temp");
        } catch (NoSuchAttributeException ex) { }
        catch (IOException ex)
       {
           Logger.getLogger(Raster.class.getName()).log(Level.SEVERE, null, ex);
      }
      finally
      {    out.close();     }
    }
    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the
+ sign on the left to edit the code.">
    /**
     * Handles the HTTP
     * <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
```

```
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException
    {
        processRequest(request, response);
        //response.setContentType("text/xml");
    }

    /**
     * Handles the HTTP
     * <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse
…response)
   throws ServletException, IOException {
    processRequest(request,response);
    }
    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
```

```
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>
}
```

### 3.4 Implementación del prototipo (hardware)

Para la implementación del prototipo se ha montado el circuito en un pequeño vehículo a control remoto de juguete.

**Figura 4,12**
Vehículo en el cual será implementado el prototipo
**Fuente:** Elaboración propia

El circuito está montado en una placa prefabricada, Se han soldado los componentes a fin de evitar cualquier falla que se pueda dar en el proto-board; además se comparte el cable de GND que debe ser común para todos.

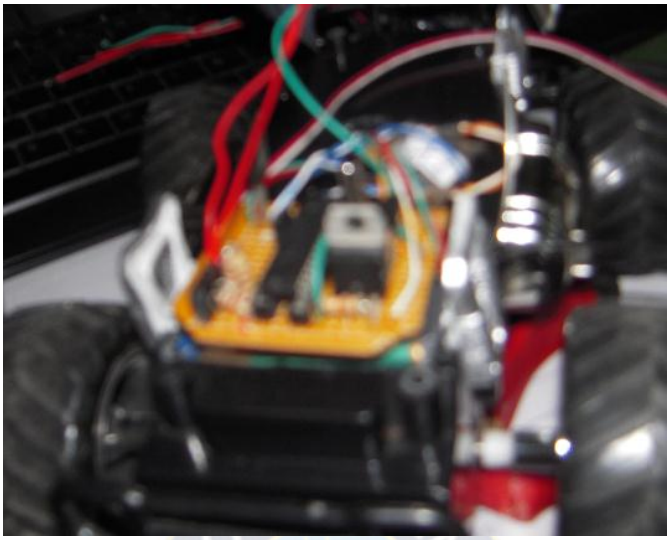


**Figura 4,13**
Circuito montado en el prototipo
**Fuente:** Elaboración propia

Finalmente el circuito y la aplicación web han funcionado sin ningún problema.

## CAPITULO IV

## CONCLUSIONES

El prototipo de monitoreo la localización del vehículo.

- El prototipo de monitoreo permite la localización de un vehículo.
- La combinación de android y el micro-controlador facilitan el desarrollo de proyectos.
- Se pueden usar módulos de wifi, bluetooh y GPS de android, evitando la compra por separado de estos módulos y reduciendo los costos.
- La implementación del prototipo fue exitosa, gracias a las tecnologías de android y los micro-controladores.

**BIBLIOGRAFÍA.**


http://tecbolivia.com/index.php/venta-de-componentes-electronicos-11/comunicaciones/modulo-serial-bluetooth-hc-06-maestro-detail
Consultado 05/07/2013

http://www.neoteo.com/modulo-bluetooth-hc-06-android
Consultado 05/07/2013

http://www.neoteo.com/comandos-at-modulo-bluetooth-hc-06
Consultado 05/07/2013

http://www.bricogeek.com/shop/203-pic-16f876a-20mhz-8k.html
Consultado 06/07/2013

http://www.futurlec.com/Microchip/PIC16F876A.shtml
Consultado 06/07/2013

http://developer.android.com/guide/topics/connectivity/bluetooth.html
Consultado 06/07/2013

http://www.oracle.com/technetwork/java/index.html
Consultado 06/07/2013

# ANEXOS

# Guangzhou HC Information Technology Co., Ltd.

## Product Data Sheet

Module Data Sheet

**Rev 1**

| 1. 0 | 2.0 | 2.1 | 2.2 | | | | |
|---|---|---|---|---|---|---|---|
| 2006/6/18 | 2006/9/6 | 2010/4/22 | 2011/4/6 | | | | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | |
| | | | | | | | | | |

| | | | | |
|---|---|---|---|---|
| **DRAWN BY:** | Ling Xin | | **MODEL:** HC-06 | |
| **CHECKED BY:** | Eric Huang | | **Description:** BC04 has external 8M Flash and EDR module HC-06 is industrial, and compatible with civil HC-04 | |
| **APPD. BY:** | Simon Mok | | **REV: 2.0** | **Page:** |
| **Former version introduction** | HC-06 is the higher version of LV_BC_2.0. Linvor is the former of wavesen. | | | |

www.wavesen.com Phone: 020-84083341 Fax: 020-84332079 QQ:1043073574
Address: Room 527, No.13, Jiangong Road, Tianhe software park, Tianhe district, Guangzhou Post: 510660
Technology consultant: support@wavesen.com Business consultant:sales@wavesen.com
Complaint and suggestion: sunbirdbt@hotmail.com

# 1. Product's picture

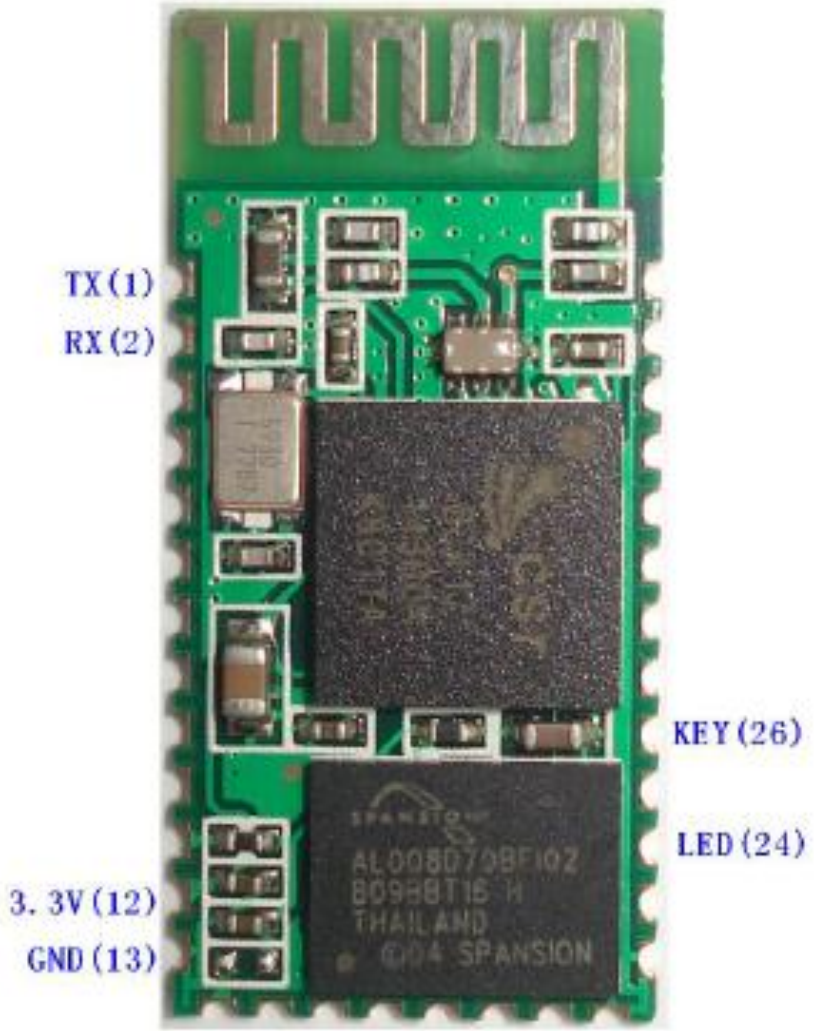



Figure 1 A Bluetooth module

## 2. Feature

- Wireless transceiver
  - Sensitivity (Bit error rate) can reach -80dBm.
  - The change range of output's power: -4 - +6dBm.
- Function description (perfect Bluetooth solution)
  - Has an EDR module; and the change range of modulation depth: 2Mbps - 3Mbps.
  - Has a build-in 2.4GHz antenna; user needn't test antenna.
  - Has the external 8Mbit FLASH
  - Can work at the low voltage (3.1V~4.2V). The current in pairing is in the range of 30~40mA. The current in communication is 8mA.
  - Standard HCI Port (UART or USB)
  - USB Protocol: Full Speed USB1.1, Compliant With 2.0
  - This module can be used in the SMD.
  - It's made through RoHS process.
  - The board PIN is half hole size.
  - Has a 2.4GHz digital wireless transceiver.
  - Bases at CSR BC04 Bluetooth technology.
  - Has the function of adaptive frequency hopping.
  - Small (27mm×13mm×2mm)
  - Peripherals circuit is simple.
  - It's at the Bluetooth class 2 power level.
  - Storage temperature range: -40 ℃ - 85℃, work temperature range: -25 ℃ - +75℃
  - Any wave inter Interference: 2.4MHz, the power of emitting: 3 dBm.
  - Bit error rate: 0. Only the signal decays at the transmission link, bit error may be produced. For example, when RS232 or TTL is being processed, some signals may decay.

- Low power consumption
- Has high-performance wireless transceiver system
- Low Cost

- Application fields:
  - Bluetooth Car Handsfree Device
  - Bluetooth GPS
  - Bluetooth PCMCIA , USB Dongle
  - Bluetooth Data Transfer
- Software
  - CSR

## 3. PINs description

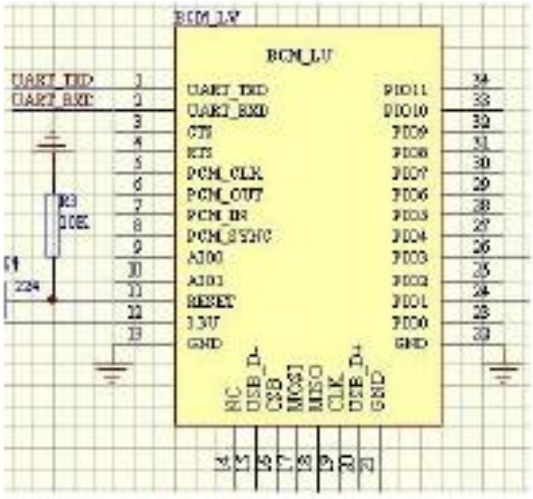



Figure 3 PIN configuration

The PINs at this block diagram is as same as the physical one.

| PIN Name | PIN # | Pad type | Description | Note |
|---|---|---|---|---|
| GND | 13 21 22 | VSS | Ground pot | |
| 1V8 | 14 | VDD | Integrated 1.8V (+) supply with On-chip linear regulator output within 1.7-1.9V | |
| VCC | 12 | 3.3V | | |
| AIO0 | 9 | Bi-Directional | Programmable input/output line | |
| AIO1 | 10 | Bi-Directional | Programmable input/output line | |

| | | | | |
|---|---|---|---|---|
| PIO0 | 23 | Bi-Directional<br>RX EN | Programmable input/output line, control output for LNA(if fitted) | |
| PIO1 | 24 | Bi-Directional<br>TX EN | Programmable input/output line, control output for PA(if fitted) | |
| PIO2 | 25 | Bi-Directional | Programmable input/output line | |
| PIO3 | 26 | Bi-Directional | Programmable input/output line | |
| PIO4 | 27 | Bi-Directional | Programmable input/output line | |
| PIO5 | 28 | Bi-Directional | Programmable input/output line | |
| PIO6 | 29 | Bi-Directional | Programmable input/output line | CLK_REQ |
| PIO7 | 30 | Bi-Directional | Programmable input/output line | CLK_OUT |
| PIO8 | 31 | Bi-Directional | Programmable input/output line | |
| PIO9 | 32 | Bi-Directional | Programmable input/output line | |
| PIO10 | 33 | Bi-Directional | Programmable input/output line | |
| PIO11 | 34 | Bi-Directional | Programmable input/output line | |
| RESETB | 11 | CMOS Input with weak internal pull-down | | |
| UART_RTS | 4 | CMOS output, tri-stable with weak internal pull-up | UART request to send, active low | |
| UART_CTS | 3 | CMOS input with weak internal pull-down | UART clear to send, active low | |
| UART_RX | 2 | CMOS input with weak internal pull-down | UART Data input | |
| UART_TX | 1 | CMOS output, Tri-stable with weak internal pull-up | UART Data output | |
| SPI_MOSI | 17 | CMOS input with weak internal pull-down | Serial peripheral interface data input | |
| SPI_CSB | 16 | CMOS input with weak internal | Chip select for serial peripheral interface, active low | |

| | | | | |
|---|---|---|---|---|
| | | pull-up | | |
| SPI_CLK | 19 | CMOS input with weak internal pull-down | Serial peripheral interface clock | |
| SPI_MISO | 18 | CMOS input with weak internal pull-down | Serial peripheral interface data Output | |
| USB_- | 15 | Bi-Directional | | |
| USB_+ | 20 | Bi-Directional | | |
| 1.8V | 14 | | 1.8V external power supply input | Default : 1.8V internal powe r supply. |
| PCM_CLK | 5 | Bi-Directional | | |
| PCM_OUT | 6 | CMOS output | | |
| PCM_IN | 7 | CMOS Input | | |
| PCM_SYNC | 8 | Bi-Directional | | |

# 5. Block diagram

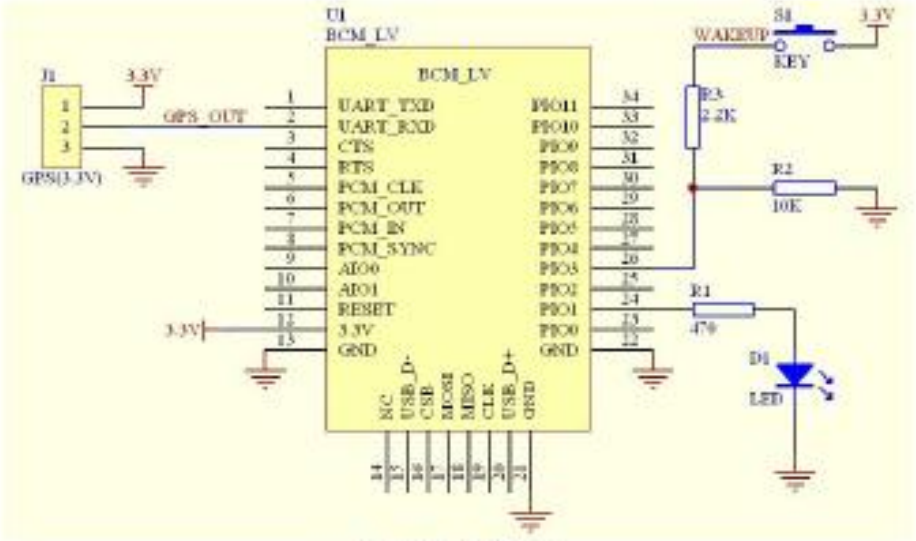



Figure 5 Block diagram 1

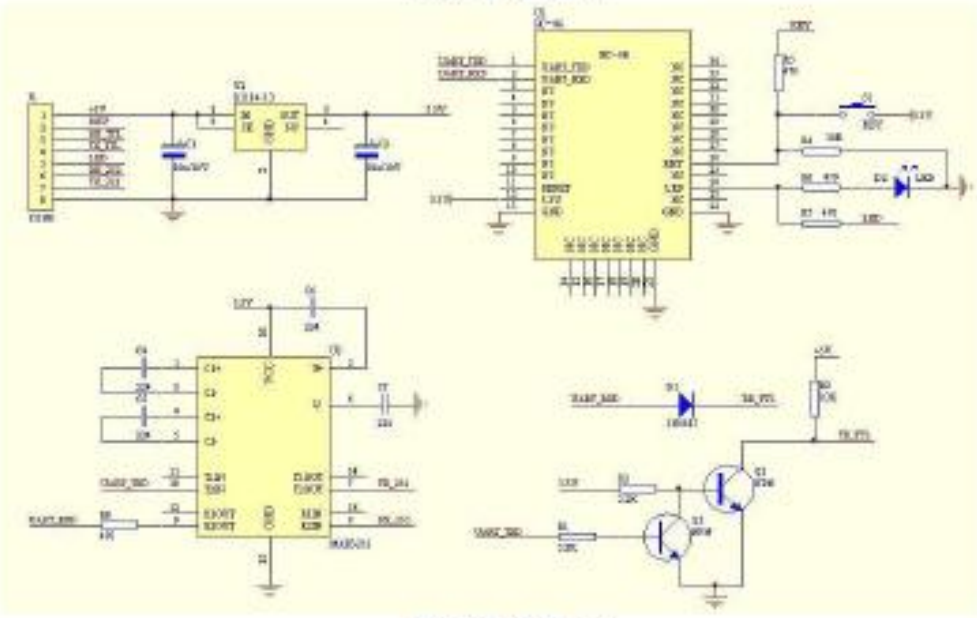


Figure 5 Block diagram 2

www.wavesen.com Phone: 020-84083341 Fax: 020-84332079 QQ:1043073574
Address: Room 527, No.13, Jiangong Road, Tianhe software park, Tianhe district, Guangzhou Post: 510660
Technology consultant: support@wavesen.com Business consultant:sales@wavesen.com
Complaint and suggestion: sunbirdt@hotmail.com

# PIC16F87XA

## 28/40/44-Pin Enhanced Flash Microcontrollers

### Devices Included in this Data Sheet:

- PIC16F873A
- PIC16F874A
- PIC16F876A
- PIC16F877A

### High-Performance RISC CPU:

- Only 35 single-word instructions to learn
- All single-cycle instructions except for program branches, which are two-cycle
- Operating speed: DC – 20 MHz clock input
  DC – 200 ns instruction cycle
- Up to 8K x 14 words of Flash Program Memory,
  Up to 368 x 8 bytes of Data Memory (RAM),
  Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to other 28-pin or 40/44-pin PIC16CXXX and PIC16FXXX microcontrollers

### Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during Sleep via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10-bit
- Synchronous Serial Port (SSP) with SPI™ (Master mode) and $I^2C$™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) – 8 bits wide with external $\overline{RD}$, $\overline{WR}$ and $\overline{CS}$ controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

### Analog Features:

- 10-bit, up to 8-channel Analog-to-Digital Converter (A/D)
- Brown-out Reset (BOR)
- Analog Comparator module with:
  - Two analog comparators
  - Programmable on-chip voltage reference ($V_{REF}$) module
  - Programmable input multiplexing from device inputs and internal voltage reference
  - Comparator outputs are externally accessible

### Special Microcontroller Features:

- 100,000 erase/write cycle Enhanced Flash program memory typical
- 1,000,000 erase/write cycle Data EEPROM memory typical
- Data EEPROM Retention > 40 years
- Self-reprogrammable under software control
- In-Circuit Serial Programming™ (ICSP™) via two pins
- Single-supply 5V In-Circuit Serial Programming
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving Sleep mode
- Selectable oscillator options
- In-Circuit Debug (ICD) via two pins

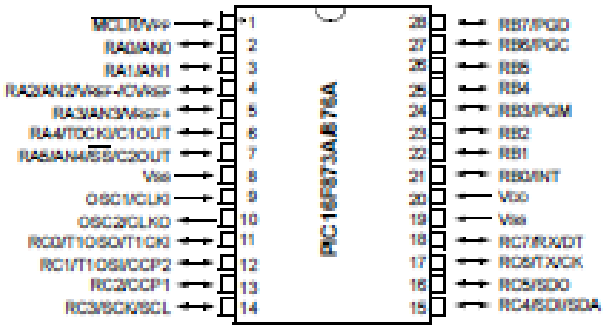
### CMOS Technology:

- Low-power, high-speed Flash/EEPROM technology
- Fully static design
- Wide operating voltage range (2.0V to 5.5V)
- Commercial and Industrial temperature ranges
- Low-power consumption

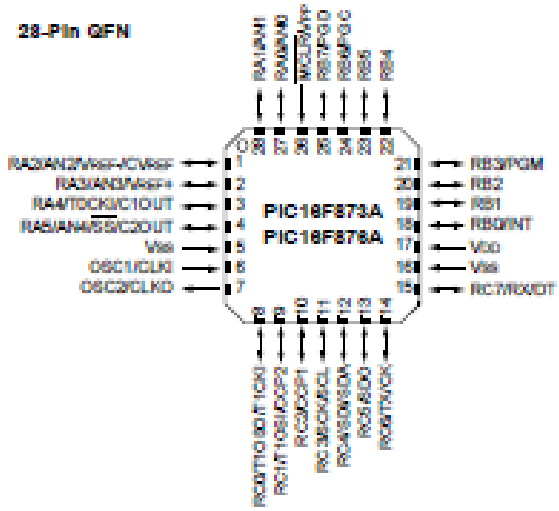
| Device | Program Memory | | Data SRAM (Bytes) | EEPROM (Bytes) | I/O | 10-bit A/D (ch) | CCP (PWM) | MSSP | | USART | Timers 8/16-bit | Comparators |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Bytes | # Single Word Instructions | | | | | | SPI | Master $I^2C$ | | | |
| PIC16F873A | 7.2K | 4096 | 192 | 128 | 22 | 5 | 2 | Yes | Yes | Yes | 2/1 | 2 |
| PIC16F874A | 7.2K | 4096 | 192 | 128 | 33 | 8 | 2 | Yes | Yes | Yes | 2/1 | 2 |
| PIC16F876A | 14.3K | 8192 | 368 | 256 | 22 | 5 | 2 | Yes | Yes | Yes | 2/1 | 2 |
| PIC16F877A | 14.3K | 8192 | 368 | 256 | 33 | 8 | 2 | Yes | Yes | Yes | 2/1 | 2 |

## Pin Diagrams

### 28-Pin PDIP, SOIC, SSOP




### 28-Pin QFN




### 44-Pin QFN

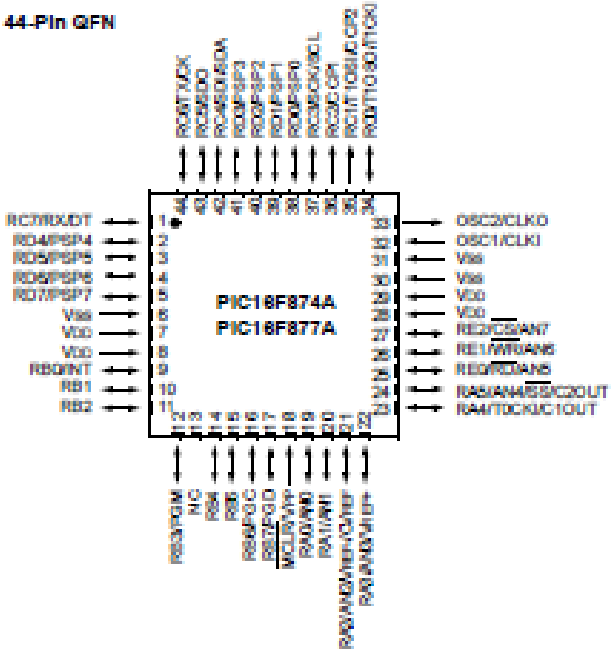
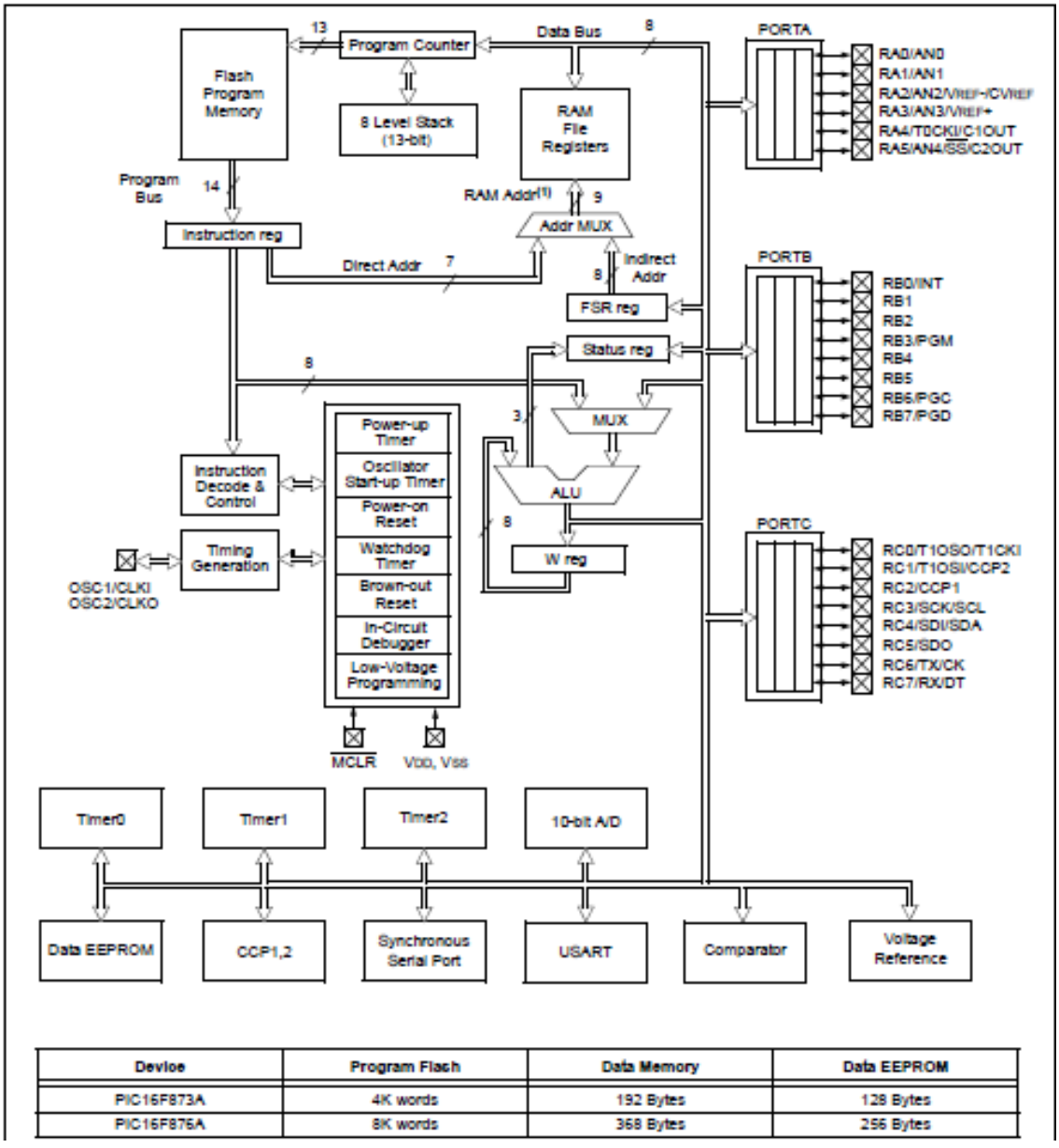


**FIGURE 1-1: PIC16F873A/876A BLOCK DIAGRAM**




| Device | Program Flash | Data Memory | Data EEPROM |
|---|---|---|---|
| PIC16F873A | 4K words | 192 Bytes | 128 Bytes |
| PIC16F876A | 8K words | 368 Bytes | 256 Bytes |

National Semiconductor

December 1994

# LM35/LM35A/LM35C/LM35CA/LM35D
# Precision Centigrade Temperature Sensors

## General Description

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¾°C over a full −55 to +150°C temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 µA from its supply, it has very low self-heating, less than 0.1°C in still air. The LM35 is rated to operate over a −55° to +150°C temperature range, while the LM35C is rated for a −40° to +110°C range (−10° with improved accuracy). The LM35 series is available packaged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-202 package.

## Features

- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/°C scale factor
- 0.5°C accuracy guaranteeable (at +25°C)
- Rated for full −55° to +150°C range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than 60 µA current drain
- Low self-heating, 0.08°C in still air
- Nonlinearity only ±¼°C typical
- Low impedance output, 0.1 Ω for 1 mA load

## Connection Diagrams

**TO-46**
**Metal Can Package***


TL/H/5516–1

*Case is connected to negative pin (GND)

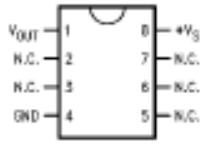
**Order Number LM35H, LM35AH, LM35CH, LM35CAH or LM35DH**
**See NS Package Number H03H**

**TO-92**
**Plastic Package**


TL/H/5516–2

**Order Number LM35CZ, LM35CAZ or LM35DZ**
**See NS Package Number Z03A**

**SO-8**
**Small Outline Molded Package**

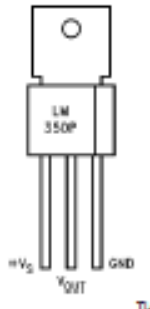



TL/H/5516–21

**Top View**
N.C. = No Connection

**Order Number LM35DM**
**See NS Package Number M08A**

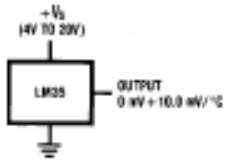
**TO-202**
**Plastic Package**




TL/H/5516–24

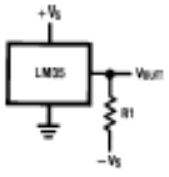
**Order Number LM35DP**
**See NS Package Number P03A**

## Typical Applications




TL/H/5516–3

**FIGURE 1. Basic Centigrade Temperature Sensor (+2°C to +150°C)**




TL/H/5516–4

Choose $R_1 = -V_S/50\ \mu A$

$V_{OUT} = +1,500$ mV at +150°C
$= +250$ mV at +25°C
$= -550$ mV at −55°C

**FIGURE 2. Full-Range Centigrade Temperature Sensor**

TRI-STAR® is a registered trademark of National Semiconductor Corporation.

RRD-B30M75/Printed in U.S.A.

## Absolute Maximum Ratings (Note 10)

**If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.**

| | |
|---|---|
| Supply Voltage | +35V to −0.2V |
| Output Voltage | +6V to −1.0V |
| Output Current | 10 mA |
| Storage Temp., TO-46 Package, | −60°C to +180°C |
| TO-92 Package, | −60°C to +150°C |
| SO-8 Package, | −65°C to +150°C |
| TO-202 Package, | −65°C to +150°C |
| Lead Temp.: | |
| TO-46 Package, (Soldering, 10 seconds) | 300°C |
| TO-92 Package, (Soldering, 10 seconds) | 260°C |
| TO-202 Package, (Soldering, 10 seconds) | +230°C |
| SO Package (Note 12): | |
| Vapor Phase (60 seconds) | 215°C |
| Infrared (15 seconds) | 220°C |
| ESD Susceptibility (Note 11) | 2500V |
| Specified Operating Temperature Range: $T_{MIN}$ to $T_{MAX}$ (Note 2) | |
| LM35, LM35A | −55°C to +150°C |
| LM35C, LM35CA | −40°C to +110°C |
| LM35D | 0°C to +100°C |

## Electrical Characteristics (Note 1) (Note 6)

| Parameter | Conditions | LM35A | | | LM35CA | | | Units (Max.) |
|---|---|---|---|---|---|---|---|---|
| | | Typical | Tested Limit (Note 4) | Design Limit (Note 5) | Typical | Tested Limit (Note 4) | Design Limit (Note 5) | |
| Accuracy (Note 7) | $T_A$ = +25°C | ±0.2 | ±0.5 | | ±0.2 | ±0.5 | | °C |
| | $T_A$ = −10°C | ±0.3 | | | ±0.3 | | ±1.0 | °C |
| | $T_A$ = $T_{MAX}$ | ±0.4 | ±1.0 | | ±0.4 | ±1.0 | | °C |
| | $T_A$ = $T_{MIN}$ | ±0.4 | ±1.0 | | ±0.4 | | ±1.5 | °C |
| Nonlinearity (Note 8) | $T_{MIN} \le T_A \le T_{MAX}$ | **±0.18** | | **±0.35** | **±0.15** | | **±0.3** | °C |
| Sensor Gain (Average Slope) | $T_{MIN} \le T_A \le T_{MAX}$ | **+10.0** | **+9.9, +10.1** | | **+10.0** | | **+9.9, +10.1** | mV/°C |
| Load Regulation (Note 3) $0 \le I_L \le 1$ mA | $T_A$ = +25°C | ±0.4 | ±1.0 | | ±0.4 | ±1.0 | | mV/mA |
| | $T_{MIN} \le T_A \le T_{MAX}$ | **±0.5** | | **±3.0** | **±0.5** | | **±3.0** | mV/mA |
| Line Regulation (Note 3) | $T_A$ = +25°C | ±0.01 | ±0.05 | | ±0.01 | ±0.05 | | mV/V |
| | $4V \le V_S \le 30V$ | **±0.02** | | **±0.1** | **±0.02** | | **±0.1** | mV/V |
| Quiescent Current (Note 9) | $V_S$ = +5V, +25°C | 56 | 67 | | 56 | 67 | | µA |
| | $V_S$ = +5V | **105** | | **131** | **91** | | **114** | µA |
| | $V_S$ = +30V, +25°C | 56.2 | 68 | | 56.2 | 68 | | µA |
| | $V_S$ = +30V | **105.5** | | **133** | **91.5** | | **116** | µA |
| Change of Quiescent Current (Note 3) | $4V \le V_S \le 30V$, +25°C | 0.2 | 1.0 | | 0.2 | 1.0 | | µA |
| | $4V \le V_S \le 30V$ | **0.5** | | **2.0** | **0.5** | | **2.0** | µA |
| Temperature Coefficient of Quiescent Current | | **+0.39** | | **+0.5** | **+0.39** | | **+0.5** | µA/°C |
| Minimum Temperature for Rated Accuracy | In circuit of *Figure 1*, $I_L$ = 0 | +1.5 | | +2.0 | +1.5 | | +2.0 | °C |
| Long Term Stability | $T_J$ = $T_{MAX}$, for 1000 hours | ±0.08 | | | ±0.08 | | | °C |

**Note 1:** Unless otherwise noted, these specifications apply: $-55°C \le T_J \le +150°C$ for the LM35 and LM35A; $-40° \le T_J \le +110°C$ for the LM35C and LM35CA; and $0° \le T_J \le +100°C$ for the LM35D. $V_S$ = +5Vdc and $I_{LOAD}$ = 50 µA, in the circuit of *Figure 2*. These specifications also apply from +2°C to $T_{MAX}$ in the circuit of *Figure 1*. Specifications in **boldface** apply over the full rated temperature range.

**Note 2:** Thermal resistance of the TO-46 package is 400°C/W, junction to ambient, and 24°C/W junction to case. Thermal resistance of the TO-92 package is 180°C/W junction to ambient. Thermal resistance of the small outline molded package is 220°C/W junction to ambient. Thermal resistance of the TO-202 package is 65°C/W junction to ambient. For additional thermal resistance information see table in the Applications section.