

**ESCUELA POLITÉCNICA DEL EJÉRCITO**

**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA  
Y TELECOMUNICACIONES**

**PROYECTO DE GRADO PARA LA OBTENCIÓN DEL  
TÍTULO EN INGENIERÍA ELECTRÓNICA**

**“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA AUTOMÁTICO  
PARA ASISTENCIA DE CONDUCCIÓN VEHICULAR”**

**Rodrigo Daniel Tipán Casamen**

**SANGOLQUÍ - ECUADOR**

**2010**

## CERTIFICACIÓN

Certificamos que el presente proyecto de grado titulado “Diseño e Implementación de un Sistema Automático Para Asistencia de Conducción Vehicular” ha sido desarrollado en su totalidad por el Sr. Rodrigo Daniel Tipán Casamen con C.I. 171472888-6 bajo nuestra dirección.

Ing. Alejandro Castro  
DIRECTOR

Ing. Evelio Granizo  
CODIRECTOR

## RESUMEN

El presente proyecto tiene por objetivo implementar un sistema de asistencia vehicular que ayudaría a los usuarios en las vías del Ecuador. Este sistema diseñado proporciona información de forma auditiva de puntos previamente almacenados de carreteras específicas, los cuales pueden ser: señales viales, poblados cercanos, etc. Para el proyecto específicamente se toma los puntos correspondientes a cada puente de la autopista general Rumiñahui.

Para realizar este proyecto se utilizaron conceptos como: Comunicación Serial Asíncrona, Comunicación Serial Síncrona, Protocolo de Comunicación NMEA-0183, microcontroladores, Sistema de Posicionamiento Global (GPS).

El sistema diseñado tiene un módulo GPS que, luego de ser configurado, envía datos de forma asíncrona, estos datos son recibidos por el microcontrolador y son almacenados, convertidos y comparados con los datos almacenados en el microcontrolador (correspondientes a cada puente de la Autopista General Rumiñahui). Cuando una comparación resulta afirmativa, el microcontrolador envía mediante el puerto serial síncrono una secuencia de datos correspondiente a direcciones de memoria a un grabador de voz, para que el grabador de voz reproduzca la secuencia de audio correspondiente a dichas direcciones de memoria.

Si bien, existen sistemas parecidos en otros países, la principal ventaja es que este sistema está diseñado para puntos o lugares específicos del Ecuador. Por el contrario, un inconveniente es la disponibilidad de elementos en nuestro medio, ya que esto hace que se produzca más margen de error o utilizar dos fuentes en lugar de una.

## DEDICATORIA

“A todas las personas que hicieron posible culminar mi carrera. A mi mami por apoyarme en todo durante toda mi vida, a las personas que me cuidan desde arriba y a todos mis amigos que he hecho durante toda mi vida, y que me han apoyado en las buenas y en las malas.”

Daniel Tipán C.

## **AGRADECIMIENTO**

“A todas las personas que hicieron posible culminar mi carrera. A mi mami por apoyarme en todo durante toda mi vida, a las personas que me cuidan desde arriba y a todos mis amigos que he hecho durante toda mi vida, y que me han apoyado en las buenas y en las malas.”

Daniel Tipán C.

## PRÓLOGO

La idea de implementar un sistema de asistencia vehicular ayudaría a los usuarios de las vías, sobretodo en las vías en el país, las cuales en muchos tramos tienen poca o nula señalización, también ayudaría a conductores que utilizan las vías en horas con poca visibilidad. Este sistema no pretende reemplazar la señalización visual, más bien, se busca complementarla, ya que por diversos factores en nuestro medio, existen muchas señalizaciones que se encuentran en mal estado o han sido robadas o manipuladas.

Este sistema utiliza una combinación de protocolos de comunicación, los cuales son controlados e interpretados por un microcontrolador permitiendo realizar advertencias de forma auditiva a una distancia prudente antes de llegar a un punto determinado.

Este proyecto se describe en los siguientes capítulos:

En el capítulo I se realiza una recopilación de información, para conocer información general acerca de los dispositivos a utilizarse, además se busca sistemas similares para analizar los algoritmos utilizados y dar una idea para realizar el proyecto.

En el capítulo II se presenta información específica de los elementos a utilizar, así como, del protocolo de comunicación NMEA-0183 y de los puertos de comunicación necesarios para el proyecto.

En el capítulo III se muestra el diseño realizado para acoplar los diferentes puertos de comunicación con el microcontrolador, también se presenta el diseño de cada etapa del proyecto, y al final se presenta el diseño de todo el proyecto.

En el capítulo IV se explica mediante diagramas de flujo el programa realizado para interpretar los datos enviados por el módulo GPS al microcontrolador, y a la vez, mediante diagrama de flujos se muestra el programa que permite enviar los datos del microcontrolador al grabador de voz y así presentar los avisos.

El capítulo V corresponde a las pruebas realizadas en la autopista general Rumiñahui con el sistema diseñado y el análisis de costos de implementación.

En los anexos se encuentran información específica de cada elemento (Módulo GPS, Microcontrolador, Grabador de voz) así como de su respectiva programación.

## ÍNDICE DE CONTENIDO

CERTIFICACIÓN .....	2
RESUMEN .....	3
DEDICATORIA .....	4
AGRADECIMIENTO .....	5
PRÓLOGO .....	6
ÍNDICE DE CONTENIDO.....	8
ÍNDICE DE TABLAS .....	11
ÍNDICE DE FIGURAS.....	12
CAPÍTULO I.....	14
INTRODUCCIÓN .....	14
1.1. RECOPIACIÓN DE INFORMACIÓN .....	14
1.1.1. GPS (Sistema de Posicionamiento Global) .....	14
1.1.1.1. SIRF .....	15
1.1.1.2. SIRF STAR III.....	16
1.1.2. MICROCONTROLADORES .....	16
1.1.2.1. Arquitectura de los Microcontroladores.....	17
1.1.2.2. Recursos Especiales de un Microcontrolador .....	19
1.2. BÚSQUEDA Y ESTUDIO DE SISTEMAS SIMILARES EXISTENTES.....	22
1.2.1. EASYTRACK – Localización Pasiva.....	22
1.2.2. STARFINDER – Localización en Tiempo Real .....	23
1.2.3. Localizador de Vehículos GPS – GSM – GPRS .....	23
1.3. ANÁLISIS DE LAS ESTRUCTURAS DE LOS SISTEMAS EXISTENTES.....	23
1.4. ANÁLISIS DE ALGORITMOS UTILIZADOS EN LOS SISTEMAS EXISTENTES .....	25
CAPÍTULO II.....	27
MARCO TEÓRICO .....	27



<b>2.1. TEORÍA DE FUNCIONAMIENTO DEL SISTEMA DE POSICIONAMIENTO GLOBAL (GPS)</b>	<b>27</b>
2.1.1. Descripción General del Sistema .....	27
2.1.2. Protocolo de Comunicación NMEA – 0183 .....	29
2.1.2.1. NMEA .....	29
2.1.2.2. Protocolo NMEA.....	29
2.1.2.3. Formato de Sentencias NMEA .....	30
<b>2.2. TRANSMISIÓN SERIAL .....</b>	<b>31</b>
2.2.1. Transmisor / Receptor Asíncrono - Síncrono Universal (USART).....	32
2.2.2. Puerto Serie Síncrono Principal (MSSP).....	33
2.2.2.1. Interfaz Serie para Periféricos (Modo SPI).....	33
2.2.2.2. Interfaz Inter – Circuitos (I <sup>2</sup> C).....	34
<b>CAPÍTULO III.....</b>	<b>35</b>
<b>DISEÑO DEL SISTEMA AUTOMÁTICO AUDITIVO DE ASISTENCIA VEHICULAR .....</b>	<b>35</b>
3.1. FUENTE DE VOLTAJE.....	36
3.1.1. Fuente de Voltaje de 5 Voltios .....	37
3.1.2. Fuente de Voltaje de 3 Voltios .....	37
3.2. COMUNICACIÓN GPS – MICROCONTROLADOR .....	40
3.3. COMUNICACIÓN MICROCONTROLADOR – GRABADOR DE VOZ .....	42
3.4. ESQUEMA GENERAL DE SISTEMA AUTOMÁTICO AUDITIVO DE ASISTENCIA VEHICULAR .....	83
<b>CAPÍTULO IV .....</b>	<b>85</b>
<b>DESARROLLO DEL FIRMWARE .....</b>	<b>85</b>
4.1. PROGRAMACIÓN DEL GPS .....	85
4.2. SOFTWARE PARA PROGRAMAR E INTERPRETAR EL GPS EN UN MICROCONTROLADOR. ....	87
4.2.1. Inicialización del GPS .....	88
4.2.2. Recepción de Datos del Módulo GPS .....	90
4.2.3. Conversión de Datos .....	93
4.2.4. Comparación de Datos .....	95
4.3. SOFTWARE QUE PERMITA ACTIVAR EL GRABADOR DE VOZ DESDE EL MICROCONTROLADOR. ....	97
4.3.1. Grabador de Voz ISD 4004-XMP.....	97
4.3.2. Software para Utilizar el Grabador de Voz .....	100
4.4. PROGRAMA FINAL .....	108
<b>CAPÍTULO V .....</b>	<b>110</b>
<b>PRUEBAS Y ANÁLISIS ECONÓMICO .....</b>	<b>110</b>
5.1 PRUEBAS EN LA AUTOPISTA GENERAL RUMIÑAHUI.....	110

5.1.1. Espacio Teórico de Muestreo del GPS.....	110
5.1.2. Velocidad Máxima del Vehículo.....	112
5.1.3. Error del Módulo GPS.....	113
5.2. ANÁLISIS DE COSTOS.....	116
<b>CAPÍTULO VI.....</b>	<b>118</b>
<b>CONCLUSIONES Y RECOMENDACIONES.....</b>	<b>118</b>
6.1. CONCLUSIONES.....	118
6.2. RECOMENDACIONES.....	120
<b>ANEXOS.....</b>	<b>121</b>

## ÍNDICE DE TABLAS

Tabla. 2.1. Formato de Mensajes de Salida NMEA	29
Tabla. 2.2. Formato de Mensajes de Entrada NMEA	30
Tabla. 4.1. Configuración de Puerto Serial	46
Tabla. 4.2. Formato de Control de Consulta y Velocidad de Transmisión	46
Tabla. 4.3. Formato de Dato RMC	51
Tabla. 4.4. Características de Frecuencia del Grabador de Voz	58
Tabla. 4.5. Registros del Grabador de Voz	59
Tabla. 4.6. Instrucciones del Grabador de Voz	60
Tabla. 4.7. Comandos de Voz	62
Tabla. 4.8. Datos Generales de Comparación e Interpretación	63
Tabla. 5.1. Primera Prueba Realizada (Menor a 60 Km/h)	73
Tabla. 5.2. Segunda Prueba Realizada (Veloc. Mayor a 60 Km/h)	74
Tabla. 5.3. Análisis de Costos del Dispositivo	76

## ÍNDICE DE FIGURAS

Figura. 1.1. Arquitectura Von Neumann	17
Figura. 1.2. Arquitectura Harvard	18
Figura. 1.3. Sistema EASYTRACK	21
Figura. 1.4. Sistema STARFINDER	22
Figura. 1.5. Dispositivo LAIPAC TECH GPRS/CDMA	22
Figura. 1.6. Algoritmo de Localización en Tiempo Real	24
Figura. 1.7. Algoritmo de Localización Pasiva	25
Figura. 3.1. Diagrama de Bloques General del Hardware	34
Figura. 3.2. Diagrama de Bloques	34
Figura. 3.3. Fuente de Voltaje de 5 V	36
Figura. 3.4. Esquema Básico de Fuente con LM317	37
Figura. 3.5. Fuente de Voltaje de 3V	39
Figura. 3.6. Comunicación GPS – Microcontrolador	40
Figura. 3.7. Comunicación Microcontrolador – Grabador de Voz	42
Figura. 3.8. Esquema Final del Hardware	44
Figura. 4.1. Esquema General del Programa	47
Figura. 4.2. Inicialización de GPS (Envío de la Primera Trama de Datos)	48
Figura. 4.3. Inicialización de GPS (Envío de la Segunda Trama de Datos)	49
Figura. 4.4. Transmisión Asíncrona	50
Figura. 4.5. Recepción de los Datos del GPS	52
Figura. 4.6. Recepción Asíncrona	53
Figura. 4.7. Conversión de Datos	54
Figura. 4.8. Esquema del Arreglo de Datos “gps”	55
Figura. 4.9. Comparación de Datos	56
Figura. 4.10. Esquema de una Instrucción para el Grabador de Voz	59
Figura. 4.11. Salida de Datos Mediante Grabador de Voz	61
Figura. 4.12. Inicialización del Grabador de Voz	64
Figura. 4.13. Reproducción en el Grabador de Voz	65

Figura. 4.14. Función para Detener el Grabador de Voz	66
Figura 4.15. Transmisión Sincrónica	67
Figura 4.16. Programa Final	68
Figura. 5.1. Entrada Autopista General Rumiñahui	70
Figura. 5.2. Autopista General Rumiñahui	71

## CAPÍTULO I

### INTRODUCCIÓN

#### 1.1. RECOPIACIÓN DE INFORMACIÓN

En el presente proyecto, se ha visto la necesidad de realizar primero una base de información acerca de los parámetros que serán usados en el transcurso del desarrollo del proyecto.

Para lo cual, se buscará información de los principales componentes del sistema (Interfaz Periférica Serial SPI, Comunicación Serial Asíncrona USART), además, se realizará la recopilación de los elementos con los que no se están familiarizados, tales como son: Módulo GPS EM-401 y el Grabador de Voz ISD4004-08M.

##### 1.1.1. GPS (Sistema de Posicionamiento Global)

El sistema GPS (Sistema de Posicionamiento Global) fue creado por el Departamento de Defensa de los Estados Unidos, para constituir un sistema de navegación preciso con fines militares, que sustituyera al antiguo sistema utilizado, que no era otro que las mediciones Doppler sobre la constelación Transit. Para ello, aprovecharon las condiciones de la propagación de las ondas de radio de la banda L en el espacio, así como la posibilidad de modular las ondas para que en ellas se pueda incluir la información necesaria, que permita posicionar un objeto en el sistema de referencia apropiado.

Este proyecto se hizo realidad entre los meses de febrero y diciembre de 1978, cuando se lanzaron los cuatro primeros satélites de la constelación NAVSTAR, que hacían posible el sistema que resolvería la incógnita de nuestra posición en la Tierra.

Actualmente, el GPS funciona mediante una red de 29 satélites que se encuentran orbitando alrededor de la Tierra. Cuando se desea determinar la posición específica de un punto, lugar u objeto, el dispositivo utilizado, localiza automáticamente cuatro satélites de la red como mínimo, de los que recibe señales indicando la posición del objeto o lugar y el reloj de cada uno de los satélites. En base a estas señales, el dispositivo sincroniza el reloj del GPS y calcula el retraso de las señales, es decir, la distancia del satélite. Por triangulación calcula la posición en el que objeto o lugar se encuentra.

La triangulación, en el caso del GPS, se basa en determinar la distancia de cada satélite respecto al punto de medición, por lo que, conocidas las distancias, se determina fácilmente la posición relativa del objeto respecto a los satélites que enviaron las señales. Además, conociendo las coordenadas o posición de cada satélite (por la señal que emite cada satélite), se obtiene la posición absoluta o las coordenadas reales del punto de medición. De igual forma, se obtiene una exactitud extrema en el reloj del GPS, similar a la de los relojes atómicos que desde la Tierra sincronizan a los satélites.

La antigua Unión Soviética tenía un sistema similar llamado GLONASS, ahora gestionado por la Federación Rusa.

Actualmente la Unión Europea intenta lanzar su propio sistema de posicionamiento por satélite denominado Galileo.

#### **1.1.1.1. SIRF**

Los fabricantes de GPS utilizan distintos fabricantes de chips (circuitos electrónicos) que controlan todo el sistema de recepción GPS, en donde la antena recibe las señales, pero es el chip el responsable de identificar las señales de los GPS de entre todas las emisiones de radiofrecuencias que están presentes en la Tierra, y es el responsable de procesar parte de estas señales.

Uno de estos chips, es fabricado por la empresa SIRF, y su principal característica y ventaja es que el grado de integración que se consigue en el chip, ha permitido miniaturizar el tamaño del chip y por ende de los dispositivos GPS.

Hay diversos tipos de circuitos diseñados por la empresa SIRF, pero se dará características básicas de SIRF STAR III, que es el chip utilizado en la implementación del proyecto.

#### **1.1.1.2. SIRF STAR III**

Como se mencionó anteriormente, los circuitos diseñados por la empresa SIRF permiten un mejor desempeño del dispositivo para trabajos en tiempo real, porque cuenta con algoritmos de corrección de errores, para suprimir especialmente los errores ocasionados por el efecto multitrayectoria.

La tecnología SIRF STAR III diseñada por esta empresa tiene las principales características:

- Dispositivos de tamaño reducido.
- Reconocimiento de señales de baja potencia (sensibilidad del receptor de -159 dBm).
- Diseñado para trabajar con redes asincrónicas de 2G, 2.5G o 3G.
- Rápido procesamiento de datos, gracias a la cantidad de correladores (alrededor de 200000) que tiene incorporado.

#### **1.1.2. MICROCONTROLADORES**

Un microcontrolador es un dispositivo electrónico capaz de llevar a cabo procesos lógicos, estos procesos o acciones son programados en lenguaje ensamblador por el usuario, que son introducidos en el microcontrolador a través de un programador.

Un microcontrolador dispone normalmente de los siguientes componentes:



- Procesador o CPU (Unidad Central de Proceso por sus siglas en inglés).
- Memoria RAM para almacenar los datos.
- Memoria de tipo ROM/PROM/EPROM para almacenar el programa.
- Líneas de E/S para comunicarse con el exterior.
- Diversos módulos para el control de periféricos (Temporizadores, Puertos Serie y Paralelo, Conversores Analógico/Digital, Conversores Digital/Analógico, etc).
- Generador de impulsos de reloj que sincronizan el funcionamiento de todo el sistema.

#### **1.1.2.1. Arquitectura de los Microcontroladores**

Inicialmente todos los microcontroladores adoptaron la arquitectura clásica Von Neumann, en la actualidad se impone la arquitectura Harvard. A continuación se detalla los dos tipos de arquitecturas:

**1. Arquitectura Von Neumann.** La arquitectura tradicional de computadoras y microprocesadores está basada en la arquitectura Von Neumann, en la cual, la unidad central de proceso (CPU), está conectada a una memoria única donde se almacenan las instrucciones del programa y los datos.

El tamaño de la unidad de datos o instrucciones está fijado por el tamaño del bus que comunica la memoria con la CPU, así un microprocesador de 8 bits con un bus de 8 bits, tendrá que manejar datos e instrucciones de una o más unidades de 8 bits de longitud. Si tiene que acceder a una instrucción o dato de más de un byte de longitud, tendrá que realizar más de un acceso a la memoria.

Además, al disponer de un único bus hace que el microprocesador sea más lento en su respuesta, porque no puede buscar en memoria una nueva instrucción mientras no finalice la transferencia de datos de la instrucción anterior.

En la figura 1.1 se presenta la arquitectura Von Neumann.

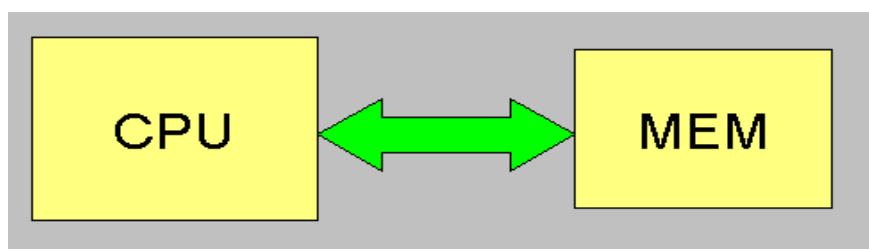


Figura. 1.1. Arquitectura Von Neumann

2. **Arquitectura Harvard.** Tiene la unidad central de proceso (CPU) conectada a dos memorias, una con las instrucciones y otra con los datos, por medio de dos buses diferentes. Una de las memorias contiene solamente las instrucciones del programa (Memoria de Programa), y la otra sólo almacena datos (Memoria de Datos), por lo tanto, ambos buses son totalmente independientes y pueden ser de distintos tamaños.

Además, al ser los buses independientes, la CPU puede acceder a los datos para completar la ejecución de una instrucción, y al mismo tiempo leer la siguiente instrucción a ejecutar.

Para un procesador de Set de Instrucciones Reducido o RISC (Reduced Instruction Set Computer), el set de instrucciones y el bus de memoria de programa pueden diseñarse de tal manera que todas las instrucciones tengan una sola posición de memoria de programa de longitud.

Las principales ventajas que presenta esta arquitectura son las siguientes:

- El tamaño de las instrucciones no está relacionado con el de los datos, y por lo tanto puede ser optimizado para que cualquier instrucción ocupe una sola posición de memoria de programa, logrando así mayor velocidad y menor longitud de programa.

- El tiempo de acceso a las instrucciones puede superponerse con el de los datos, logrando una mayor velocidad en cada operación.

En la figura 1.2 se presenta la Arquitectura Harvard.

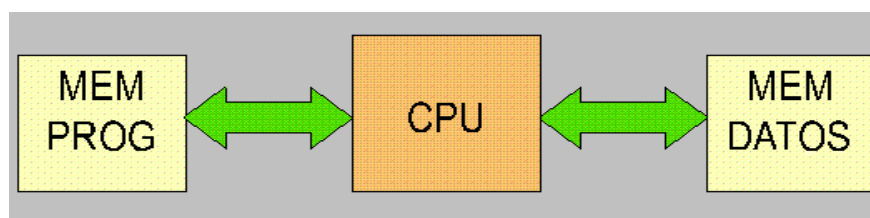


Figura. 1.2. Arquitectura Harvard

### 1.1.2.2. Recursos Especiales de un Microcontrolador

A continuación se detallan las funciones particulares de los microcontroladores, las cuales se encuentran en los microcontroladores dependiendo del modelo y fabricante que se utilice:

**1. Temporizadores (Timers).** Se emplean para controlar períodos de tiempo (temporizadores), y para llevar la cuenta de acontecimientos que suceden en el exterior (contadores).

Para la medida de tiempos se carga un registro con el valor adecuado y a continuación dicho valor se va incrementado o decrementado al ritmo de los impulsos de reloj o algún múltiplo hasta que se desborde y llegue a "0", momento en el que se produce un aviso.

Cuando se desea contar acontecimientos que se producen por cambios de nivel o flancos al exterior del microcontrolador, el mencionado registro se va incrementando o decrementando al ritmo de dichos impulsos.

**2. Watchdog (Perro Guardián).** Cuando el computador personal se bloquea por un fallo de software u otra causa, se pulsa el botón de reset y se

reinicializa el sistema. Un microcontrolador funciona sin el control de un supervisor y de forma continua las 24 horas del día, por lo que el watchdog consiste en un temporizador que, cuando se desborda y pasa por "0", provoca un reset automáticamente en el sistema.

**3. Protección Ante Fallo de Alimentación (Brownout).** Se trata de un circuito que resetea al microcontrolador cuando el voltaje de alimentación ( $V_{DD}$ ) es inferior a un voltaje mínimo (brownout). Mientras el voltaje de alimentación sea inferior al de brownout, el dispositivo se mantiene reseteado, comenzando a funcionar normalmente cuando se sobrepasa dicho valor.

**4. Estado de Reposo (Bajo Consumo).** Para ahorrar energía los microcontroladores disponen de una instrucción especial (SLEEP en los PIC), que pone a los circuitos del microcontrolador en estado de reposo o bajo consumo, en el cual los requerimientos de potencia son mínimos. En dicho estado se detiene el reloj principal y se paran sus circuitos asociados. Al activarse una interrupción ocasionada por el acontecimiento esperado, el microcontrolador reinicia y reanuda su trabajo.

**5. Conversor A/D (CAD).** Los microcontroladores que incorporan un conversor A/D (Analógico/Digital) pueden procesar señales analógicas, tan abundantes en las aplicaciones. Suelen disponer de un multiplexor que permite aplicar a la entrada del CAD diversas señales analógicas desde los pines del circuito integrado.

**6. Conversor D/A.** Transforma los datos digitales obtenidos del procesamiento del microcontrolador en su correspondiente señal analógica, que es enviada al exterior por uno de los pines.

**7. Comparador Analógico.** Algunos modelos de microcontroladores disponen inicialmente de un amplificador operacional, que actúa como

comparador entre una señal fija de referencia y otra variable que se aplica por uno de los pines del microcontrolador. La salida del comparador proporciona un nivel lógico de “1” ó “0” según sea mayor o menor que la otra señal.

**8. Modulador de Anchura de Pulsos (PWM).** Son circuitos que proporcionan en su salida, pulsos de anchura variable, que se ofrecen al exterior a través de los pines del microcontrolador.

**9. Puertos de E/S Digitales.** Todos los microcontroladores destinan algunos de sus pines para soportar líneas de E/S digitales.

Las líneas digitales de los puertos pueden configurarse como Entrada o Salida, cargando un “1” ó “0” en el bit correspondiente de un registro destinado a su configuración.

**10. Puertos de Comunicación.** Con el objeto de dotar al microcontrolador de la posibilidad de comunicarse con otros dispositivos externos, y con diferentes protocolos y normas de comunicación, algunos modelos disponen de recursos que permiten directamente esta tarea, entre los que se destacan los siguientes:

- **UART**, adaptador de comunicación seria asíncrona.
- **USART**, adaptador de comunicación serie síncrona y asíncrona.
- **Puerta Paralela Esclava**, para conseguir conectarse con los buses de otros microcontroladores.
- **USB (Universal Serial Bus)**, es un moderno bus serie para PC.
- **Bus I<sup>2</sup>C**, es un interfaz serie de dos hilos desarrollado por Philips.
- **CAN (Controller Area Network)**, permite la adaptación con redes de conexión multiplexado, desarrollado conjuntamente por Bosch e Intel para el cableado de dispositivos en automóviles.

## 1.2. BÚSQUEDA Y ESTUDIO DE SISTEMAS SIMILARES EXISTENTES

Después de realizar una búsqueda de sistemas existentes, no se encontraron sistemas que puedan presentar similitudes a lo que se desea diseñar, porque los sistemas existentes encontrados, están enfocados a la supervisión y control vehicular, teniendo una red GSM o CDMA para la transmisión de datos hacia un servidor de control y seguimiento de un vehículo.

El sistema que más se acerca a lo deseado, es el GPS a bordo de diferentes automóviles, los cuales permiten visualizar en pantalla (pantalla localizada en el panel del automóvil) la ubicación actual del automóvil, más no permite o no proporciona advertencias o características de las rutas.

Además, ningún sistema estudiado, proporciona la información de forma auditiva, solo se limita a informaciones mediante visualización en pantalla, ya sea, en el automóvil o en un servidor (computador).

Entre los sistemas encontrados y estudiados se pueden citar a los siguientes:

### 1.2.1. EASYTRACK – Localización Pasiva

Es un sistema para la administración vehicular que permite conocer a detalle los movimientos realizados por el vehículo, desde horas trabajadas hasta velocidades máximas entre otras. Todo esto sin pagar mensualidades. En la figura 1.3 se presenta este dispositivo.



Figura. 1.3. Sistema EASYTRACK

### 1.2.2. STARFINDER – Localización en Tiempo Real

La serie de localizadores STARFINDER AVL de Laipac Tech provee una poderosa solución a bajo costo, para la localización automática de vehículos u otras aplicaciones de rastreo donde se necesite GPS. En la figura 1.4 se presenta este tipo de dispositivo.



Figura. 1.4. Sistema STARFINDER

### 1.2.3. Localizador de Vehículos GPS – GSM – GPRS

Este dispositivo de localización automática de vehículos AVL es de bajo costo y altas prestaciones, que ha sido diseñado para conectarse con una amplia gama de módems GSM y módems Satélite (ORBCOMM), el cual trabaja con CDMA/1X y GSM/SMS & GPRS. La comunicación a dos bandas se basa en SMS, GPRS o paquete de datos 1X. En la figura 1.5 se presenta el dispositivo.



Figura. 1.5. Dispositivo LAIPAC TECH GPRS/CDMA

## 1.3. ANÁLISIS DE LAS ESTRUCTURAS DE LOS SISTEMAS EXISTENTES

Después de estudiar los sistemas encontrados, se los ha dividido en dos tipos:

1. Localización Pasiva.
2. Localización en Tiempo Real.

**1. Localización Pasiva.** Esta orientada básicamente a llevar un registro en el mismo controlador, el cual puede ser descargado al final de un día o al llegar al destino final dependiendo del caso. Este tipo de sistema no cuenta con un módem GPRS o CDMA que permita la comunicación con un servidor, de igual forma, no brinda ninguna advertencia, solo al momento de descargar la información se incluyen datos como: rutas recorridas, promedio de velocidad, excesos de velocidad, paradas no autorizadas; entre otros parámetros dependiendo del dispositivo utilizado.

**2. Localización en Tiempo Real.** Se basa en transmitir la información obtenida por el GPS a un servidor mediante el uso de módems GPRS o CDMA, ésta información es interpretada por un servidor y visualizada en tiempo real, la cual es utilizada para el control y supervisión de automóviles. De igual forma muchas empresas aseguradoras están implementando este tipo de sistemas en vehículos, y con la ayuda de un servidor y mapas digitales realizan la localización vehicular.

Además, este tipo de localización en Tiempo Real se utiliza en las computadoras a bordo de los automóviles de gama alta, para proporcionar información al usuario de la posición del automóvil.

Este tipo de localizadores, no manipulan los datos del GPS directamente, sino que los datos son transmitidos para ser procesados en servidores, con ayuda de mapas digitales y software diseñados para aplicaciones específicas. De la misma forma, en las computadoras a bordo de automóviles, los datos son interpretados y mostrados con ayuda de mapas.



#### 1.4. ANÁLISIS DE ALGORITMOS UTILIZADOS EN LOS SISTEMAS EXISTENTES

Después de estudiar los sistemas existentes, se revisó el tipo de adquisición de datos, por lo que la interpretación de los mismos, se realizan de distinta forma.

Los sistemas, dependiendo de su tipo, Localización Pasiva o Localización en Tiempo Real, proporcionan datos como localización, velocidad, dirección, entre otras.

La figura 1.6 representa el algoritmo básico utilizado para la Localización en Tiempo Real, como se observa, se dispone de un módem GPS/GPRS (o CDMA según sea el caso) que se instala en el vehículo o dispositivo a ser supervisado, este módem dependiendo de su programación y configuración; envía datos proporcionados por el GPS (el GPS se encuentra en el mismo módem GPRS) cada cierto intervalo de tiempo utilizando una red de telefonía móvil a un receptor específico. Este receptor se encuentra conectado a un servidor o PC, el cual mediante un software específico interpreta los datos recibidos y los visualiza en pantalla con la ayuda de mapas digitales.

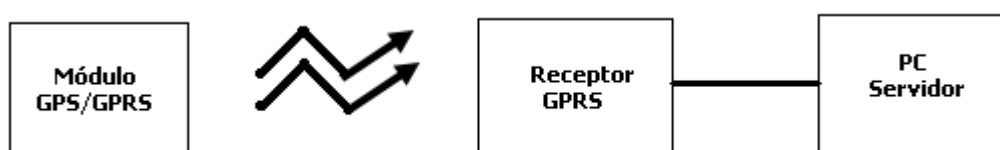
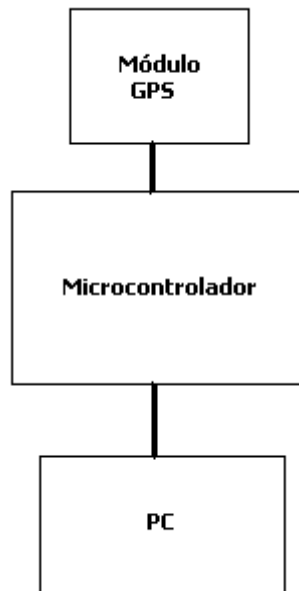


Figura. 1.6. Algoritmo de Localización en Tiempo Real

La figura 1.7 muestra el algoritmo utilizado para la Localización Activa, como se mencionó anteriormente, este tipo de localización no transmite ni interpreta datos en tiempo real; solo los almacena. El módulo GPS se configura para que envíe datos específicos (dependiendo de las necesidades de cada usuario) cada intervalo de tiempo, estos datos son almacenados en la memoria EPROM del microcontrolador, para que no se borren en caso de existir alguna interrupción de energía. Al momento de llegar al destino, el microcontrolador se conecta a un PC mediante el puerto serial o paralelo y descarga toda la

información almacenada en el microcontrolador, los datos son interpretados por la computadora y presentados en pantalla mediante mapas digitales o simplemente mediante coordenadas, velocidades y tiempo de viaje, entre otros.



**Figura. 1.7. Algoritmo de Localización Pasiva**

En los dos casos, la localización se realiza mediante comparación de puntos almacenados en las computadoras y se proporciona la posición exacta o una bitácora de rutas utilizadas. Para el diseño que se plantea, se sigue el mismo algoritmo de comparación de puntos, pero dependiendo de la dirección y sentido del automóvil se dará a conocer los próximos puntos a observar (pueden ser semáforos, zonas pobladas, entre otros).

## CAPÍTULO II

### MARCO TEÓRICO

#### 2.1. TEORÍA DE FUNCIONAMIENTO DEL SISTEMA DE POSICIONAMIENTO GLOBAL (GPS)

##### 2.1.1. Descripción General del Sistema

Para facilitar el estudio del Sistema de Posicionamiento Global se ha visto conveniente dividirlo en las siguientes tres etapas:

1. Sector Espacial.
2. Sector de Control.
3. Sector de Usuarios.

A continuación se describirá cada una de las etapas.

**1. Sector Espacial.** Este sector lo forman los satélites de la constelación NAVSTAR (Navegación por satélite en tiempo y distancia), que está conformada por seis planos orbitales, y en cada uno de ellos existe una órbita elíptica casi circular donde se alojan los satélites regularmente distribuidos, además, estos planos poseen una inclinación de  $55^\circ$  respecto al plano del ecuador, y se nombran como A, B, C, D, E y F. Cada órbita contiene al menos cuatro satélites, aunque pueden contener más, los cuales se sitúan a una distancia de 20200 Km. Respecto del geocentro, y completan una órbita en doce horas sidéreas<sup>1</sup>. Estos satélites son puestos en funcionamiento por el Comando de las Fuerzas Aéreas Espaciales de U.S.A. (AFSPC).

Con estos fundamentos, se garantiza la presencia de al menos cuatro satélites sobre el horizonte en todos los lugares de la superficie de la Tierra.

La señal de los satélites de la constelación NAVSTAR se forma a partir de un oscilador de Cesio o Rubidio; que generan una frecuencia fundamental  $\nu_0$  de 10.23 MHz. A partir de esta frecuencia fundamental se generan dos portadoras en la banda L (rango de frecuencias: 1 GHz a 2 GHz) de radiofrecuencia, denominadas L1 y L2.

**2. Sector de Control.** Es el encargado de realizar un seguimiento de todos los satélites de la red NAVSTAR. Este sector se encarga específicamente de:

- Establecer la órbita de cada satélite, así como determinar el estado de sus osciladores.
- Enviar los parámetros obtenidos anteriormente a los satélites, para que éstos puedan difundirlos a los usuarios.

De este modo, el usuario recibe la información de las efemérides de posición de los satélites y el error que se está produciendo en su reloj, todo ello incluido en el mensaje de navegación.

Las estaciones de control de dichos satélites, son principalmente las siguientes:

- Colorado Springs (USA). Central de cálculo y operaciones.
- Ascensión (Atlántico Sur).
- Hawai (Pacífico Oriental).
- Kwajalein (Pacífico Occidental).
- Diego García (Índico).

Además, existen otras estaciones de seguimiento (láser, radar y ópticas), cuyo fin es la obtención de efemérides que no estén afectadas por la disponibilidad selectiva, denominadas precisas, y que están al alcance del usuario a través de organismos científicos como el IGS (Internacional Geodynamic Service) o el NGS (Nacional Geodetic Survey). Con ellas, se tiene la seguridad de posicionamiento en el sistema WGS84 con los errores típicos del sistema.

**3. Sector de Usuarios.** En este sector se considera al instrumental que deben utilizar los usuarios para la recepción, lectura, tratamiento y configuración de las señales, con el fin de alcanzar los objetivos de su trabajo. Los elementos que conforman el sector de usuarios son: el equipo de observación y el software de cálculo; el software de cálculo puede ser objeto de uso tras la observación, o bien realizable en tiempo real.

### **2.1.2. Protocolo de Comunicación NMEA – 0183**

#### **2.1.2.1. NMEA**

NMEA (Nacional Marine Electronics Association) es una asociación sin fines de lucro de fabricantes, distribuidores, instituciones educativas y otros, interesados en equipos periféricos. La definición estándar de un NMEA – 0183 es una interfaz eléctrica y un protocolo de datos para la comunicación entre instrumentos marinos. Este protocolo se lanza por primera vez en marzo de 1983 y su última versión fue publicada en el año 2001.

#### **2.1.2.2. Protocolo NMEA**

Es utilizado para la comunicación entre dispositivos de uso marino para transmitir datos. La salida NMEA es EIA-422A, aunque para la mayoría de los propósitos se puede considerar RS-232 compatible.

Todos los datos son transmitidos a través de sentencias con caracteres ASCII, cada sentencia comienza con "\$" y termina con <CR><LF> (CF: carriage

return y LF: line feed). Los primeros dos caracteres después de “\$” son los que identifican al equipo y los siguientes tres caracteres es el identificador de tipo de sentencia que se está enviando. Los tres tipos de sentencias NMEA que existen son:

- Envío (Talker Sentences).
- Origen de Equipo (Proprietary Sentences); y
- Consulta (Query Sentences).

Los datos están delimitados por “,” (coma), por lo que deben incluirse todas las comas porque actúan como marcas.

### 2.1.2.3. Formato de Sentencias NMEA

Existen sentencias NMEA tanto para mensajes de entrada y salida; del dispositivo que se esté utilizando. A continuación se muestra en la tabla 2.1 los principales formatos de sentencias de salida de mensajes.

**Tabla. 2.1. Formato de Mensajes de Salida NMEA**

<b>Formato</b>	<b>Nombre</b>	<b>Descripción</b>
GGA	Datos del Fijo del Sistema de Posicionamiento Global	Tiempo, Posición, Tipo de datos fijo.
GLL	Posición Geográfica (Latitud, Longitud)	Latitud, Longitud, Tiempo UTC.
GSA	GNSS (Sistema Global de Navegación por Satélite) DOP y Satélites Activos	Modo de recepción GPS utilizada, satélites utilizados y valores DOP.
GSV	GNSS Satélites en Vista	Número de satélites en vista, Identificador de cada Satélite, elevación, Azimut, Valores SNR.
MSS	Receptor de Señal MSK	Señal de Ruido, Señal de frecuencia, tasa de bit.
RMC	Mínimo de Datos GNSS recomendados	Tiempo, Fecha, Posición, Curso, Velocidad
VTG	Curso Sobre Tierra y Velocidad de Tierra	Información de curso y velocidad relativa respecto a tierra.
ZDA	Mensaje SIRF	Mensaje PPS
150	Listo para Enviar	Listo para Enviar.

En la tabla 2.2 se presenta de manera rápida los formatos NMEA de Mensajes de Entrada. Estos mensajes son utilizados para configurar los dispositivos, es decir, para especificar los parámetros que se desean adquirir, la velocidad de transmisión de datos, frecuencia de adquisición, entre otros.

**Tabla. 2.2. Formato de Mensajes de Entrada NMEA**

<b>Mensaje</b>	<b>Identificador de Mensaje</b>	<b>Descripción</b>
Configurar el Puerto Serial	100	Configura los parámetros del puerto A y el protocolo a usar.
Inicialización de Navegación	101	Parámetros requeridos para comenzar a utilizar las coordenadas X/Y/Z.
Configurar el Puerto DGPS	102	Configura los parámetros DGPS en el puerto B.
Control de Consulta y Velocidad de Transmisión	103	Estándar NMEA de consulta de datos y/o configuración de velocidad de transmisión de datos de salida.
Inicialización de Navegación LLAN	104	Parámetros requeridos para comenzar a utilizar coordenadas Longitud/Latitud/Altitud
Datos de desarrollo ON/OFF	105	Mensaje de Datos ON/OFF
Selección de Dato	106	Selecciona el tipo de dato que se lee para su respectiva transformación
Interfaz de recepción MSK	MSK	Comando de mensajes para un radio faro receptor MSK

La información detallada de cada formato, tanto de entrada como de salida, se encuentra en el Anexo 1, además se muestra un ejemplo de cada formato.

## **2.2. TRANSMISIÓN SERIAL**

La transmisión serial es un protocolo muy utilizado para la comunicación entre dispositivos, por lo que se incluyen de manera estándar en cualquier computador. También, la transmisión serial es utilizada por varios dispositivos de instrumentación y adquisición de datos.

El concepto de transmisión serial es básicamente el siguiente: el puerto serial envía y recibe bytes de información un bit a la vez, aunque este método es

más lento que la comunicación en paralelo, la comunicación serial es más utilizada porque permite alcanzar mayores distancia y es más sencillo de utilizar.

### **2.2.1. Transmisor / Receptor Asíncrono - Síncrono Universal (USART)**

El módulo USART o SCI (Interfaz de Comunicaciones Serie) se utiliza en dispositivos periféricos dependiendo de su configuración, así si se utiliza como sistema asíncrono Full Duplex se puede conectar a terminales CRT: computadoras personales, microcontroladores, etc; en cambio, si se utiliza como un sistema síncrono Half Duplex sirve para conversores A/D, conversores D/A, memorias EEPROM seriales, entre otras.

Específicamente, en el presente proyecto se utiliza el módulo USART del microcontrolador configurado como Asíncrono Full-Duplex para conectarse con el módulo GPS. La comunicación en esta configuración es bidireccional, el pin RC6/Tx/CK del microcontrolador actúa como línea de transmisión y el pin RC7/Rx/DT como línea de recepción. Teniendo en cuenta que cada dato lleva un bit de inicio y uno de parada.

Las características principales del módulo USART configurado de forma Asíncrona Full – Duplex se presentan a continuación:

- La transmisión y recepción de datos funcionan a la misma tasa de baudios configurados previamente.
- Se puede enviar datos de 7 u 8 bits con paridad par/impar o sin paridad.
- Los registros utilizados para la transmisión y recepción de datos son independientes.
- El bit menos significativo (LSB) es el primero en transmitirse y recibirse.
- Posee interrupciones para transmisión y recepción.



Los modos de configuración específicos para el microcontrolador PIC16F877A, se presenta detalladamente en el Anexo 2.

### **2.2.2. Puerto Serie Síncrono Principal (MSSP)**

Es un periférico diseñado para soportar un interfaz serie síncrono, que resulta muy eficiente para la comunicación del microcontrolador con dispositivos, tales como: displays, memorias EEPROM, ADC, y en el caso específico del proyecto con Grabadores de Voz. El Puerto Serie Síncrono Principal (MSSP) dispone de los dos siguientes modos de trabajo:

1. Interfaz Serie para Periféricos (SPI).
2. Interfaz Inter – Circuitos (I<sup>2</sup>C).

En el siguiente subtema se describe el modo SPI, que es el modo de trabajo utilizado en el proyecto para comunicar el microcontrolador con el grabador de voz, y a continuación se indica brevemente el modo I<sup>2</sup>C.

#### **2.2.2.1. Interfaz Serie para Periféricos (Modo SPI)**

Es utilizado para conectar varios microcontroladores o dispositivos periféricos bajo el formato maestro – esclavo, siempre que se disponga de un interfaz compatible.

En este modo se puede utilizar 3 ó 4 señales de control: salida de datos (SDO), entrada de datos (SDI), reloj (SCK) y selección de esclavo (SS). En donde cada señal debe programarse como entrada o salida según su condición; utilizando los registros TRIS.

Cuando se recibe un dato se introduce en serie en el registro SSPSR y luego se convierte a paralelo en el registro SSPBUF para que pueda ser leído. En la transmisión se realiza la operación contraria, es decir, el dato se almacena de forma paralela en el registro SSPBUF y se pasa al registro SSPSR para ser

enviado de forma serie. De lo que puede determinarse que el registro SSPSR es un registro de desplazamiento que funciona serie/paralelo/serie.

Se utiliza el registro de control SSPCON para seleccionar el modo SPI de trabajo y para configurarlo. En el Anexo 2 se detalla la función de cada bit de dicho registro, así como las diferentes formas de configuración.

#### **2.2.2.2. Interfaz Inter – Circuitos (I<sup>2</sup>C)**

Para simplificar la interconexión de dispositivos al microprocesador, Philips desarrolló un sencillo bus bidireccional basado en dos hilos por el que se transmiten los datos vía serie. Las líneas SDA y SCL son bidireccionales y están polarizadas a positivo mediante resistencia de "pull-up" de forma que en reposo están a nivel alto.

En el bus existen maestros (que generan la señal de SCL y controlan la comunicación) y esclavos que responden a peticiones del maestro. El dato en SDA debe estar estable durante el periodo ALTO de reloj. SDA sólo puede cambiar mientras SCL se encuentre a nivel BAJO.

## CAPÍTULO III

### DISEÑO DEL SISTEMA AUTOMÁTICO AUDITIVO DE ASISTENCIA VEHICULAR

El diseño del hardware del proyecto, se realiza de acuerdo a los diagramas de bloque presentados en las figuras 3.1 y 3.2.

En la figura 3.1 se muestra el diagrama de bloques que representa el hardware de forma básica, en este diagrama se presentan los tres elementos fundamentales que forman el hardware: el grabador de voz ISD4004-8M requiere de una fuente de alimentación de 3 voltios, el módulo GPS EM-401 utiliza una fuente de alimentación de 5 voltios y el microcontrolador puede trabajar con cualquiera de las dos fuentes.



Figura. 3.1. Diagrama de Bloques General del Hardware

En la figura 3.2 se presenta el tipo de fuente que utiliza cada elemento.

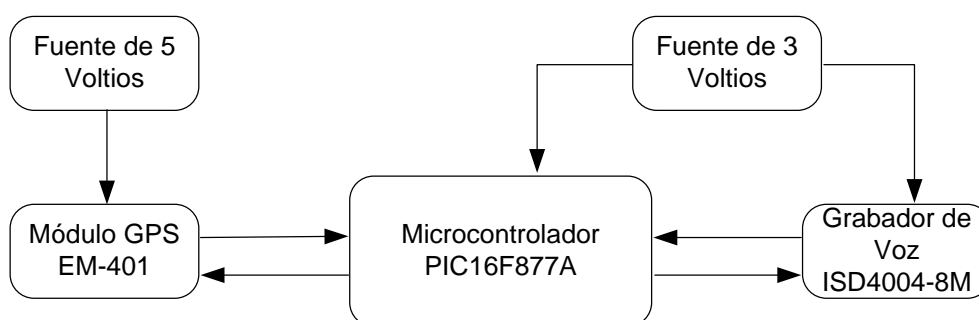


Figura. 3.2. Diagrama de Bloques

En la figura 3.2 se puede observar que el microcontrolador utiliza la fuente de alimentación de 3 voltios, esto sirve para que se pueda establecer la comunicación de datos entre los diferentes elementos, porque uno de los problemas que se presenta en el diseño del hardware es el rango de voltaje del grabador de voz.

Tomando como base a las figuras 3.1 y 3.2 para el diseño del hardware, el sistema consta de tres partes fundamentales, las cuales se indican a continuación:

1. Fuente de Voltaje de 3 y 5 Voltios.
2. Comunicación GPS – Microcontrolador.
3. Comunicación Microcontrolador – Grabador de Voz.

En los siguientes subtemas se diseñan cada parte del sistema.

### **3.1. FUENTE DE VOLTAJE**

Para diseñar las fuentes de voltaje, se toma en cuenta el voltaje de funcionamiento de los diferentes dispositivos: EM-401; ISD4004-8MP y PIC16F877A, así como también la corriente requerida. A continuación se detalla el rango de voltaje y la corriente máxima que necesita cada elemento:

- Módulo GPS EM-401 funciona en un rango de voltaje de 3.3 V – 5.5 V. y una corriente máxima de 80 mA.
- Grabador de Voz ISD4004-8MP funciona en un rango de voltaje de 2.7 V – 3.3 V y una corriente máxima de 25 mA.
- Microcontrolador PIC16F877A funciona en un rango de voltaje de 2 V – 6 V y funciona con una corriente máxima de 300 mA.

El Módulo GPS va a ser alimentado con 5 Voltios porque la transmisión de datos se va a realizar con un voltaje máximo de 2.85 V, el microcontrolador PIC16F877A y el Grabador de Voz ISD4004-8MP serán alimentados con 3 Voltios.

### 3.1.1. Fuente de Voltaje de 5 Voltios

Para el diseño de la fuente de 5 Voltios, se utiliza un Regulador de Voltaje Positivo LM7805, el cual proporciona una salida de 5 Voltios y 1 Amperio.

En la figura 3.3 se muestra el esquema básico de una fuente de alimentación de 5 voltios, este esquema, es el proporcionado por el fabricante del regulador de voltaje en el datasheet del elemento LM7805. Se añadió un capacitor de 0.1 $\mu$ F para estabilizar el voltaje de salida, pero no se tomó muchas precauciones en la estabilidad de esta fuente, porque el módulo GPS dispone de un amplio rango de voltaje de operación y utiliza una baja corriente de 80 mA máximo.

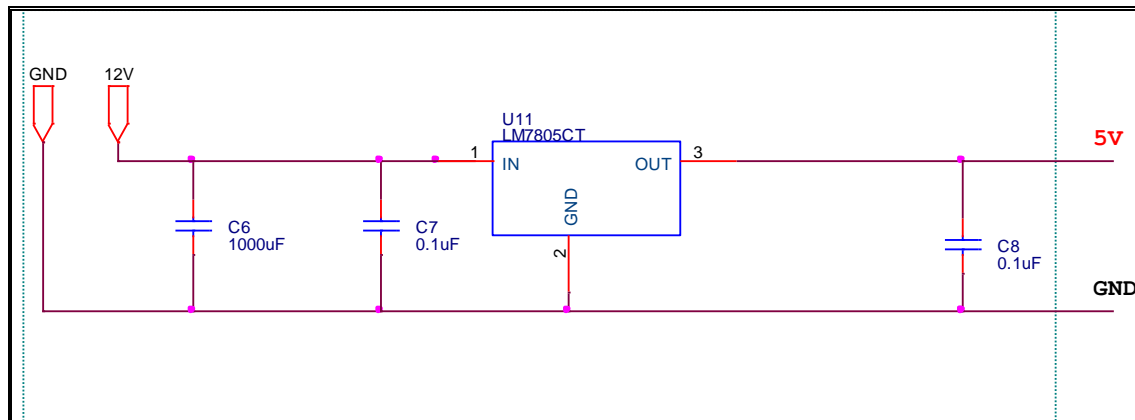


Figura. 3.3. Fuente de Voltaje de 5 V.

### 3.1.2. Fuente de Voltaje de 3 Voltios

Para realizar el diseño de la fuente de voltaje de 3 Voltios, se debe tomar en cuenta que el voltaje generado por la fuente debe ser muy estable por el rango de trabajo del grabador de voz. Además, la mayor parte del hardware va a trabajar con 3 voltios, por lo que se decidió utilizar un regulador de voltaje regulable, el LM317T, el cual permite trabajar con rangos de voltaje de 1.2 V – 30 V y obtener

una corriente máxima de 1.5 Amperios. En este proyecto el regulador LM317 va a entregar una corriente máxima de 500 mA al circuito.

En la figura 3.4 se muestra el esquema básico de una fuente regulable utilizando el regulador LM317, como se observa, el voltaje de salida depende del valor dado a las resistencias  $R_1$  y  $R_2$ .

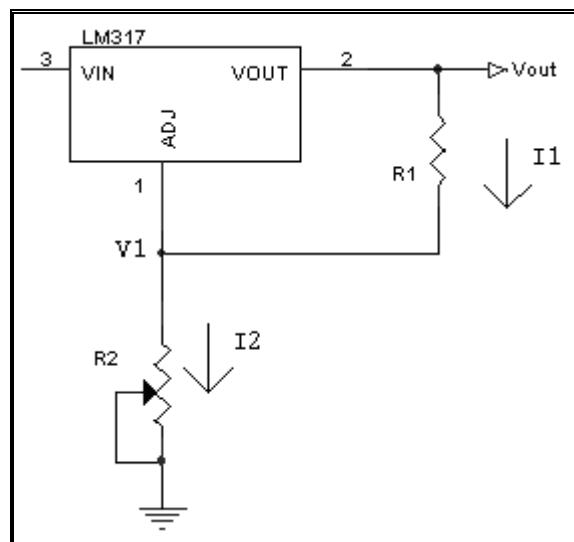


Figura. 3.4. Esquema Básico de Fuente con LM317

En la figura 3.4, la tensión entre los terminales Vout y ADJ es de 1,25 voltios, por lo que se puede calcular la corriente  $I_1$  que pasa por  $R_1$ , mediante la siguiente relación:

$$I_1 = \frac{1.25}{R_1}$$

Además, se puede calcular  $I_2$  como:

$$I_2 = \frac{(Vout - 1.25)}{R_2}$$

La corriente que entra por el terminal ADJ se puede considerar para la mayoría de aplicaciones despreciable, entonces toda la corriente  $I_1$  pasará por el potenciómetro  $R_2$ , obteniéndose:

$$\begin{aligned}
 I_1 &= I_2 \\
 \frac{1.25}{R_1} &= \frac{(V_{out} - 1.25)}{R_2} \\
 V_{out} &= 1.25 \times \left( 1 + \frac{R_2}{R_1} \right) \qquad \text{Ecuac. 3.1}
 \end{aligned}$$

Como se observa en la ecuación 3.1, los valores de  $R_1$  y  $R_2$  dependerán de la tensión de salida máxima que se desee obtener, como solo se dispone una ecuación para calcular las 2 resistencias, se tiene que dar un valor a una de ellas y a partir de esto, calcular la otra. De acuerdo al datasheet, recomienda dar un valor de entre  $100\Omega$  a  $330\Omega$  para  $R_1$ .

El valor de  $R_1$  para el diseño de la fuente se da de  $220\Omega$ , y el cálculo de  $R_2$  se determina a partir del voltaje de salida (ecuación 3.1) y el valor de  $R_1$ , como se muestra a continuación:

$$\begin{aligned}
 R_2 &= (V_{out} - 1.25) \frac{R_1}{1.25} \\
 R_2 &= (3 - 1.25) \times \frac{220}{1.25} \\
 R_2 &= 308\Omega \approx 330\Omega
 \end{aligned}$$

El valor de  $R_2$  se aproxima a un valor de resistencia comercial de  $330\Omega$ , con lo cual se obtiene un Voltaje de Salida de:

$$V_{out} = 1.25 \times \left( 1 + \frac{R_2}{R_1} \right)$$

$$V_{out} = 1.25 \times \left( 1 + \frac{330}{220} \right)$$

$$V_{out} = 3.125 V$$

Este valor de  $V_{out}$  está en el rango permitido para trabajo del grabador de voz, y, como se dijo anteriormente esta fuente debe ser estable, para lo cual se coloca un capacitor a la salida estabilizando el voltaje de salida. Además, el capacitor de 10  $\mu F$  mejora el rechazo de rizado y los diodos colocados protegen que dicho capacitor se descargue, protegiendo al integrado LM317T.

El circuito resultante de la fuente es el que se muestra en la figura 3.5, en el cual, al diseño sugerido por el fabricante, se añade los capacitares para estabilidad y diodos para protección; anteriormente explicados.

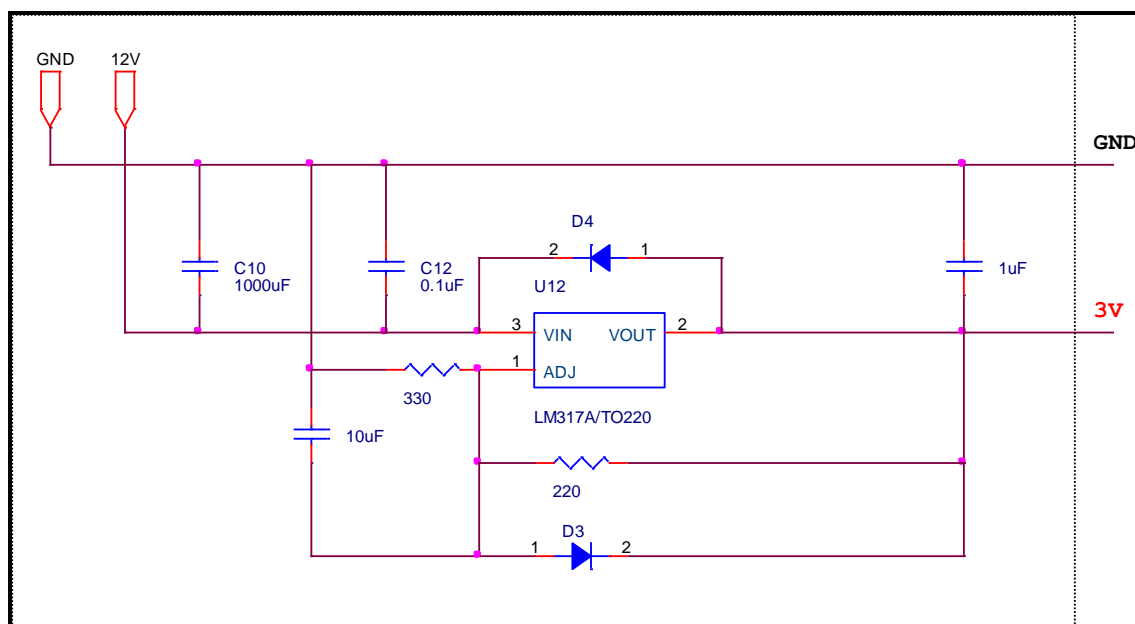


Figura. 3.5. Fuente de Voltaje de 3V.

### 3.2. COMUNICACIÓN GPS – MICROCONTROLADOR

Para la comunicación entre el Módulo GPS y el Microcontrolador se utilizan los pines de transmisión y recepción sincrónico/ asincrónico (USART por sus siglas en inglés) del microcontrolador. No hay problema con los voltajes de



alimentación porque el módulo GPS envía datos de hasta 2.85 voltios y no necesita señal de reloj, por lo que la conexión es sencilla como se muestra en la figura 3.6.

En el capítulo II, se explica que el módulo GPS utiliza un protocolo de comunicaciones NMEA-0183, por lo que la salida de este protocolo es la EIA-411A, la cual para la mayoría de aplicaciones se puede considerar como RS-232 compatible. En el proyecto se utiliza el puerto USART del microcontrolador para la comunicación con el módulo GPS, con lo cual se establece o se cumple la compatibilidad con RS-232.

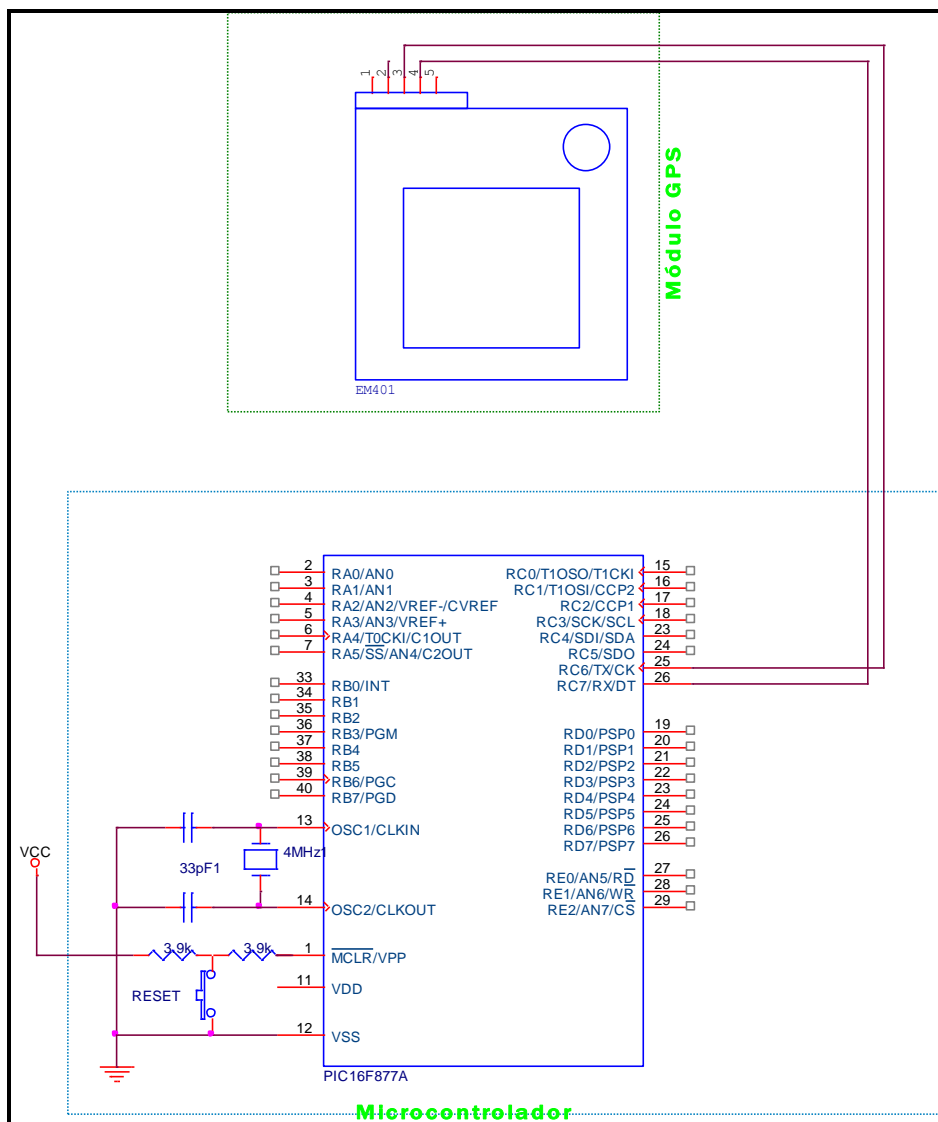


Figura. 3.6. Comunicación GPS – Microcontrolador

### 3.3. COMUNICACIÓN MICROCONTROLADOR – GRABADOR DE VOZ

Para la comunicación entre estos dispositivos, se utiliza el puerto serial sincrónico del microcontrolador (MSSP por sus siglas en inglés) en modo de Interfaz Serial (SPI por sus siglas en inglés).

Para esta comunicación, además de los pines de transmisión y recepción, se debe tomar en cuenta otros aspectos, tal como el reloj y hacerle trabajar al grabador de voz en modo secundario o de esclavo, para que el microcontrolador sea puesto como primario.

Igualmente se añade un pin de interrupción, el cual indica cuando el grabador de voz termina una secuencia de audio, para poder enviar la o las próximas secuencias de audio según sea el caso.

Para la comunicación entre microcontrolador y el grabador de voz, se utiliza el puerto MSSP, ya que el grabador de voz está diseñado para utilizar este tipo de comunicación (a diferencia de sus antecesores que utilizan un puerto paralelo), lo cual permite que el hardware sea más sencillo de implementar.

La comunicación entre el Microcontrolador y el Grabador de Voz se presenta en la figura 3.7.



### **3.4. ESQUEMA GENERAL DE SISTEMA AUTOMÁTICO AUDITIVO DE ASISTENCIA VEHICULAR**

Antes de presentar el diseño final del hardware, se puede destacar, que se utiliza un cristal de 4 MHz en el microcontrolador, porque permite utilizar menor corriente (la misma corriente que utiliza el microcontrolador con un  $V_i = 5V$  y un cristal de 20MHz.) y para la comunicación con el Grabador de Voz se especifica un valor de frecuencia de trabajo igual a 1Mhz.

El esquema final del hardware es el que se presenta en la figura 3.8.

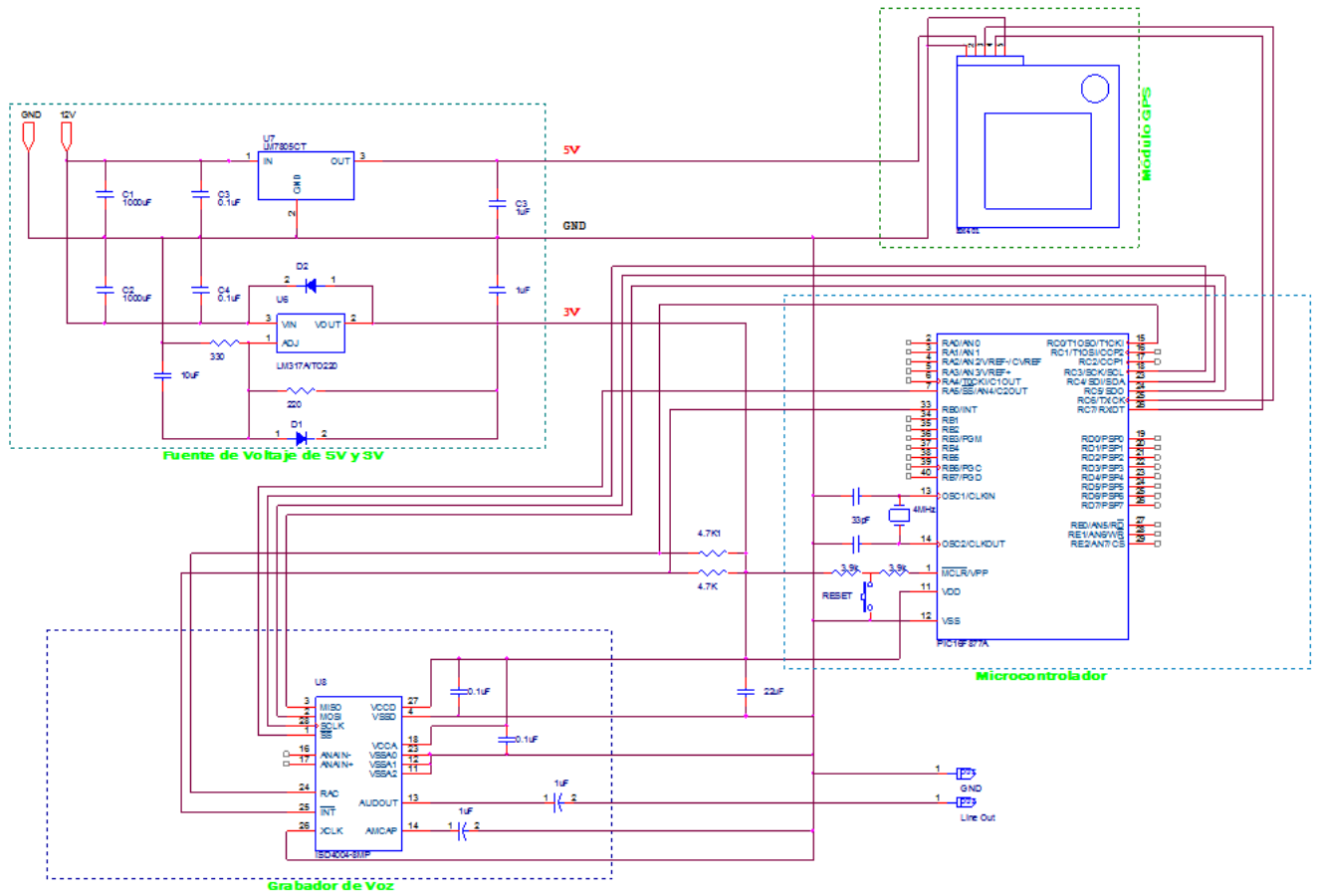


Figura 3.8. Esquema Final del Hardware

## CAPÍTULO IV

### DESARROLLO DEL FIRMWARE

#### 4.1. PROGRAMACIÓN DEL GPS

El módulo GPS EM – 401 puede configurarse dependiendo de las necesidades del usuario, a partir de los datos que se desea interpretar, así como los parámetros de transmisión y recepción de datos. En el capítulo II se mencionó que este dispositivo es compatible con el protocolo de comunicaciones NMEA – 0183, el cual indica los parámetros estandarizados para entrada y salida de datos de dispositivos compatibles con el protocolo.

Teniendo en cuenta las necesidades del proyecto se escogió los parámetros de transmisión y el tipo de datos como se muestran en las tablas 4.1 y 4.2.

La tabla 4.1 representa la configuración del puerto serial para el GPS, escogiéndose: un tipo de datos NMEA (es una trama de datos ASCII, cada dato separado por “comas”), la tasa de transmisión de 9600, y el tipo de dato que recibe el microcontrolador es de 8 bits sin paridad. Cabe mencionar que el microcontrolador debe tener configurado el puerto USART con los dos últimos parámetros del módulo GPS.

Por lo tanto, la trama de datos enviados por el microcontrolador al módulo GPS es la siguiente:

\$PSRF100,1,9600,8,0,0,\*0C

Tabla. 4.1. Configuración de Puerto Serial

NOMBRE	EJEMPLO	DESCRIPCIÓN
MSG ID	&PSRF100	Cabecera de Protocolo PSRF100
Protocolo	1	0 = SIRF Binario 1 = NMEA
Tasa	9600	4800, 9600, 19200, 38400
Bits de Dato	8	
Bit de Parada	0	0, 1
Paridad	0	0 = Sin Paridad
Checksum	*0C	
<CR><LF>		Finalización de Mensaje

La tabla 4.2 representa el Formato de Control de Consulta y Velocidad de Transmisión. El primer dato enviado (\$PSRF103) representa el Formato NMEA de Mensaje de Entrada que se explicó brevemente en el capítulo II (en la tabla 2.2 se presentan los diferentes formatos NMEA de Mensajes de Entrada). El segundo dato enviado (04) representa el tipo de dato que va a ser transmitido por el GPS, en este caso representa a Mínimo de datos GNSS recomendados (revisar tabla 2.1). Además se configura la tasa de transmisión, la cual es de envío de datos de cada segundo para el proyecto.

Este formato de datos al GPS se lo envía mediante una trama de datos, cada dato separado por “comas”. La trama de datos correspondiente a la tabla 4.2 es la siguiente:

\$PSRF103,04,00,01,00,\*25

Tabla. 4.2. Formato de Control de Consulta y Velocidad de Transmisión

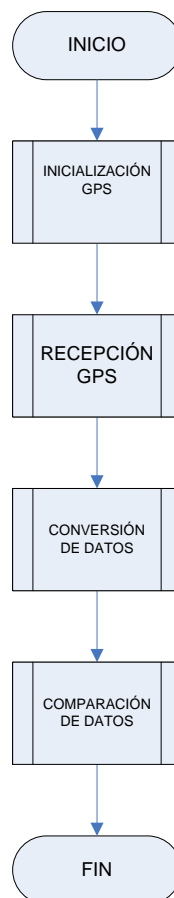
NOMBRE	EJEMPLO	UNIDADES	DESCRIPCIÓN
MSG ID	\$PSRF103		Cabecera de protocolo PSRF103
MSG	04		Envía el tipo de dato que se va a adquirir (RMC)
Modo	00		00 = Configurar el tasa 01 = Control
Tasa	01	segundos	
Habilitar Checksum	00		00 = Deshabilitado 01 = Habilitado
Checksum	*25		
<CR><LF>			Finalización de Mensaje

Por lo tanto, la programación del GPS se la realiza mediante el envío de las 2 tramas de datos mencionadas anteriormente, con lo cual el módulo GPS comienza a enviar datos con un intervalo de 1 segundo.

#### 4.2. SOFTWARE PARA PROGRAMAR E INTERPRETAR EL GPS EN UN MICROCONTROLADOR.

El programa para el microcontrolador fue creado mediante la utilización de las herramientas MPLAB IDE v.7.52 y PICC v.9.50, que permiten programar el microcontrolador mediante lenguaje C, lo cual facilita el empleo de sentencias y condiciones.

A continuación, en la figura 4.1 se presenta un esquema general de los procesos del programa creado para manipular al módulo GPS.





### Figura 4.1. Esquema General del Programa

En los siguientes subtemas se explica detalladamente cada proceso de la figura 4.1.

#### 4.2.1. Inicialización del GPS

El primer proceso que realiza el programa es la inicialización o configuración del GPS, correspondientes al envío de las tramas como se explicó en el tema anterior. En la figuras 4.2 y 4.3 se presentan los diagramas de flujo de la transmisión de las tramas de datos.

En la figura 4.2 se puede observar el arreglo de datos SSPDF, que contiene la trama de datos equivalente a la tabla 4.1. El envío de datos se lo realiza mediante el puerto asincrónico del microcontrolador.

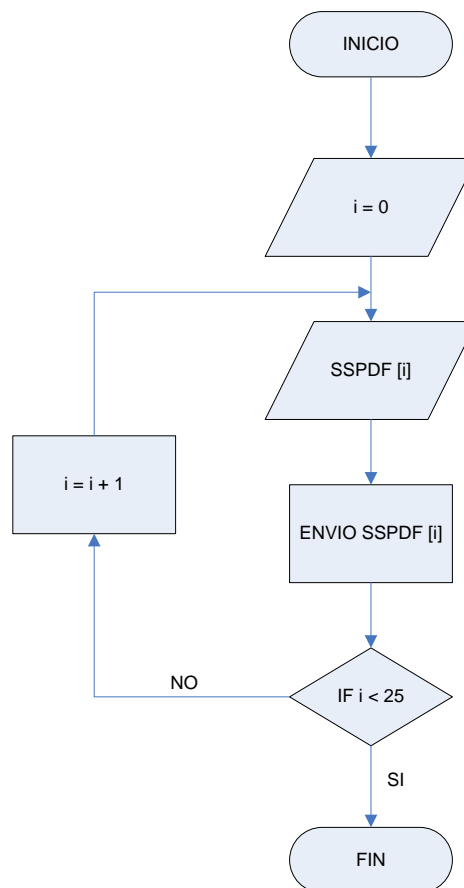
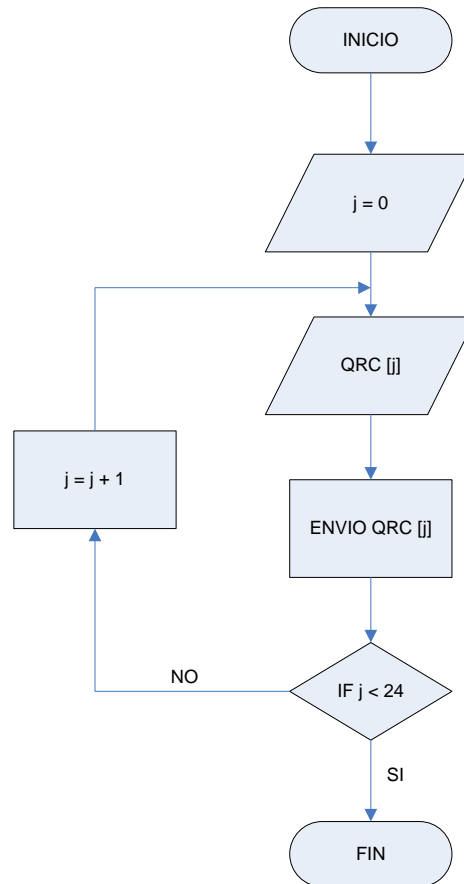


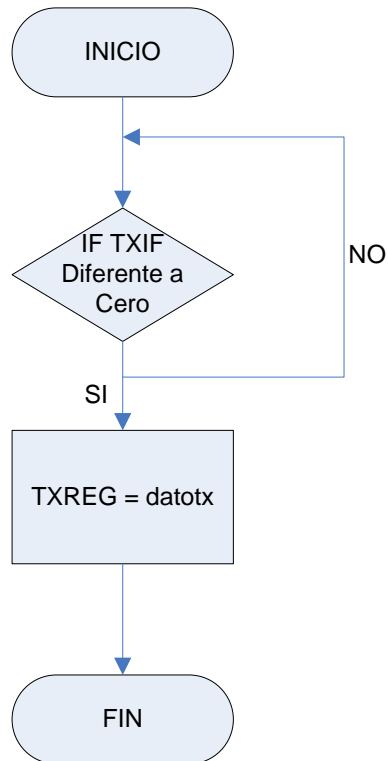
Figura 4.2. Inicialización de GPS (Envío de la Primera Trama de Datos)

La figura 4.3, al igual que la figura 4.2 representa el envío de una trama de datos al módulo GPS. Para esa trama de datos se ha asignado el arreglo de datos QRC.



**Figura 4.3. Inicialización de GPS (Envío de la Segunda Trama de Datos)**

El envío de datos utilizando el puerto serial asincrónico del microcontrolador se lo realiza mediante el registro TXREG, el cual es un convertidor paralelo – serial de datos. El registro TXIF controla cuando un dato ha terminado de enviarse. Así, la transmisión de datos por el puerto serial asincrónico se lo realiza como se presenta en la figura 4.4.



**Figura 4.4. Transmisión Asincrónica**

Finalizado la configuración del GPS, se procede a la recepción de datos enviados por el GPS. El módulo GPS envía una trama de datos dependiendo de la configuración escogida (en este caso RMC) y de igual forma cada dato es separado mediante “comas”.

#### **4.2.2. Recepción de Datos del Módulo GPS**

En la tabla 4.3 se muestra el formato RMC (Mínimo de datos GNSS recomendado).

Tabla. 4.3. Formato de Dato RMC

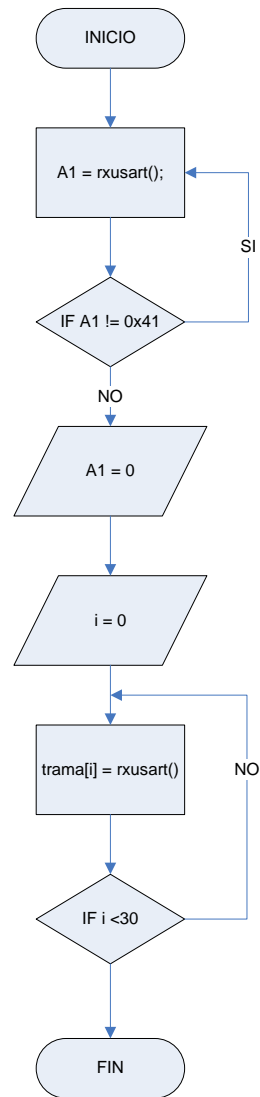
NOMBRE	EJEMPLO	UNIDADES	DESCRIPCIÓN
MSG ID	\$GPRMC		Cabecera de protocolo RMC
UTC Time	161229.487		hhmmss.sss
Status	A		A = Dato Válido V = Dato no Válido
Latitud	3723.2475		ddmm.mmmm
Indicador N/S	N		N = Norte S = Sur
Longitud	12158.3416		dddmm.mmmm
Indicador E/W	W		E = Este W = Oeste
Velocidad Sobre tierra	0.13	Nudos	
Curso sobre tierra	309.62	Grados	
Fecha	120598		ddmmyy
Variación Magnética		grados	E = Este W = Oeste
Modo	A		A = Autónomo
Checksum	*10		
<CR><LF>			Fin de Mensaje

La trama que representa a la tabla 4.3 es la siguiente:

\$GPRMS,161229.487,A,3723.2475,N,12158.3416,W,0.13,309.62,120598, , \*10

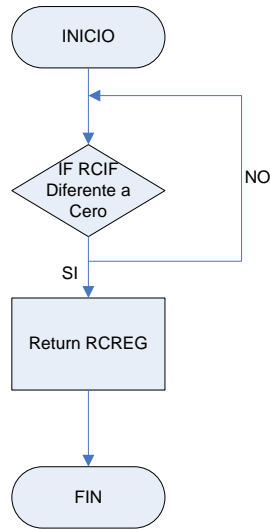
La recepción de esta trama se la realiza mediante el puerto serial asincrónico, y se almacena toda la trama en un arreglo de datos llamado “trama”.

En la figura 4.5 se presenta el diagrama de flujos correspondiente a la recepción de datos del GPS.



**Figura. 4.5. Recepción de los Datos del GPS**

La función para la recepción de datos asincrónica se explica mediante el diagrama de flujos de la figura 4.6.



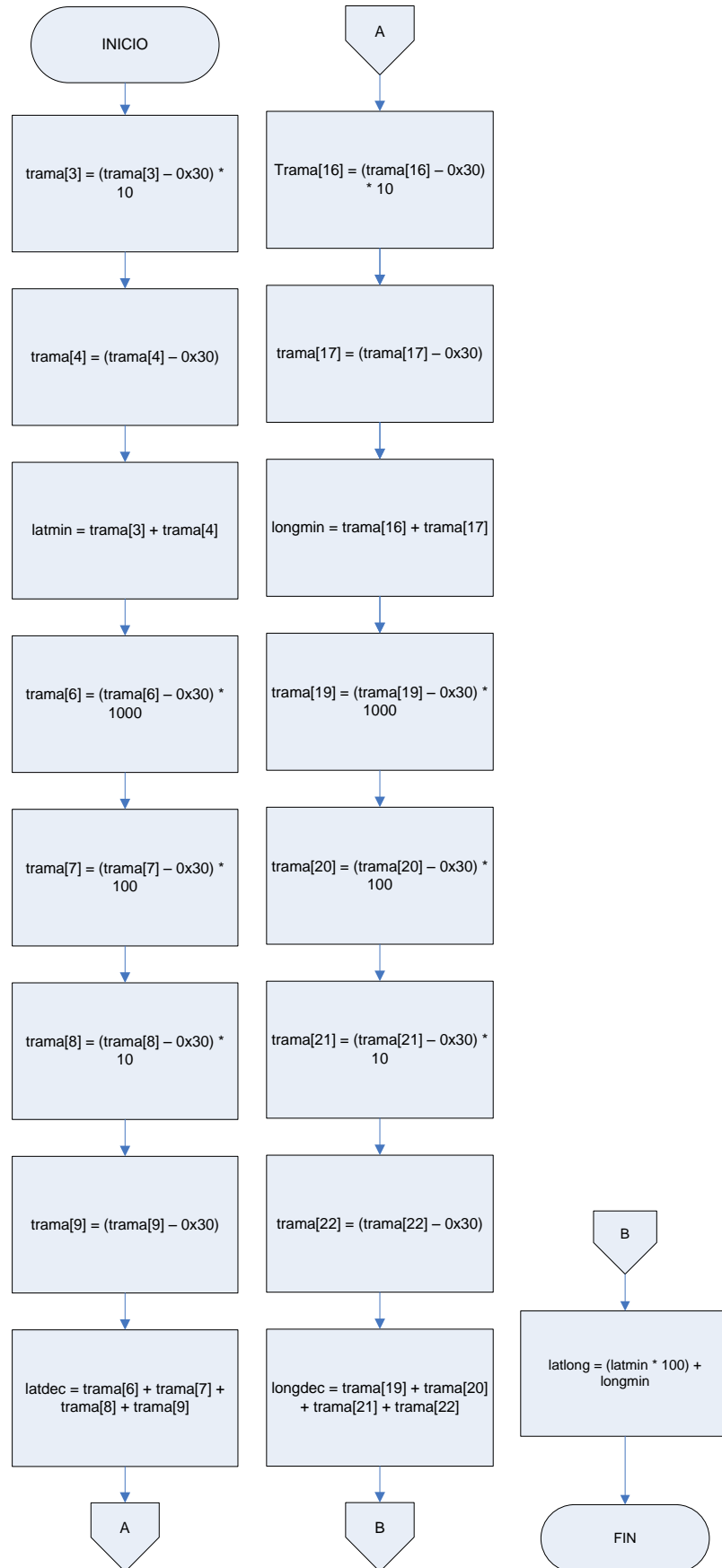
**Figura. 4.6. Recepción Asincrónica**

Como se observa en la figura 4.6, el registro RCIF permite conocer cuando un dato ha terminado de recibirse, luego este dato se almacena en el registro RCREG que es devuelto por la función de recepción asincrónica (en este caso rxusart), y el registro es almacenado en el arreglo de datos “trama”.

#### 4.2.3. Conversión de Datos

Terminado la transmisión y recepción del módulo GPS, se procede a trabajar con los datos recibidos, para posteriormente comparar e interpretar estos datos. El proceso llamado Conversión de Datos presentado en la figura 4.1 se lo realiza básicamente para convertir los datos recibidos del GPS, que son código ASCII a números decimales.

En la figura 4.7 se presenta el proceso de conversión de código ASCII a números decimales, para lo cual se realiza un resta a cada dato de 0x30 (hexadecimal), permitiendo extraer el número entero. Adicionalmente las cantidades que corresponden a los minutos (antes del punto en la trama de datos) tanto en coordenadas de latitud como en longitud son almacenadas en una sola variable denominada latlong. Los valores correspondientes a minutos restantes (después del punto en la trama de datos) se almacenan en las variables latdec (para latitud) y longdec (para longitud).



### Figura. 4.7. Conversión de Datos

#### 4.2.4. Comparación de Datos

Con los 3 datos obtenidos en el proceso de conversión de datos, se procede a comparar los datos que envía el módulo GPS con los datos almacenados en el microcontrolador.

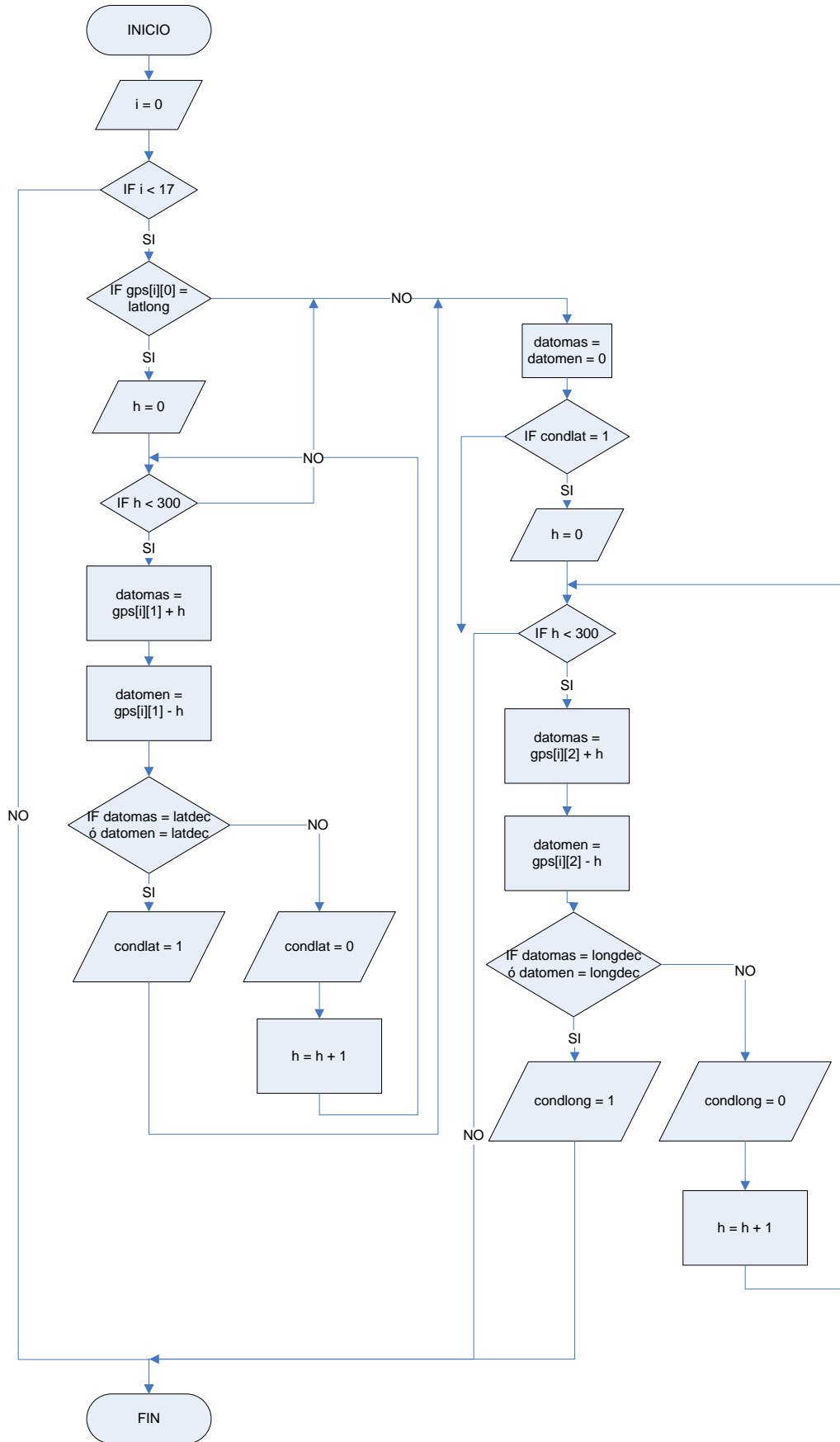
En la figura 4.9 se muestra el proceso de comparación de datos, en donde el arreglo bidimensional de datos llamado “gps” almacena las coordenadas: de cada puente peatonal de la Autopista General Rumiñahui, del peaje, de las paradas después del peaje, y del inicio de la Autopista General Rumiñahui tanto desde el valle como desde Quito. Este arreglo almacena los datos como se muestra en la figura 4.8:

gps[i][0]				gps[i][1]				gps[i][2]			
latmin	latmin	longmin	longmin	latmin	latmin	latmin	latmin	longmin	longmin	longmin	Longmin

**Figura 4.8. Esquema del Arreglo de Datos “gps”**

En la figura 4.9 se muestra el diagrama de flujo que representa a la comparación de datos.





### Figura. 4.9. Comparación de Datos

Con esos procesos realizados, queda terminado el software para la programación e interpretación de datos del GPS en un microcontrolador, todas las variables antes mencionadas son almacenadas de forma global en el microcontrolador para que puedan ser utilizadas por todas sus funciones.

## 4.3. SOFTWARE QUE PERMITA ACTIVAR EL GRABADOR DE VOZ DESDE EL MICROCONTROLADOR.

### 4.3.1. Grabador de Voz ISD 4004-XMP

El Grabador de Voz ISD4004 a diferencia de sus antecesores (como la serie ISD2560 e inferiores) presenta una mayor capacidad de grabación, pudiendo llegar hasta 16 minutos (ISD4004-16MP).

Además, a diferencia de los anteriores modelos, permite su programación mediante comunicación sincrónica, es decir, se utiliza el puerto Serial Sincrónico (MSSP = Master Synchronous Serial Port) del Microcontrolador configurado en modo de Interfaz Periférica Serial (SPI por sus siglas en ingles), lo cual simplifica y se facilita al momento de implementar el hardware (los modelos anteriores se programan mediante comunicación paralela).

Para programar y configurar el Grabador de Voz ISD4004-8MP se dispone de xx pines que se detallan a continuación:

- 1. SS (Selección de Esclavo).** Cuando esta en bajo (cero lógico) el grabador se selecciona como esclavo. Cuando se realiza la comunicación entre el microcontrolador y el grabador se configura en bajo este pin mediante software, y cuando termina la comunicación, es decir, cuando el Grabador va a realizar la instrucción enviada, este pin se lo configura en alto para que pueda realizar su tarea.

**2. MOSI (Master Out Slave In).** Este pin envía los datos desde el microcontrolador al grabador.

**3. MISO (Master In Slave Out).** Envía los datos desde el Grabador al Microcontrolador, cuando se requiera información del grabador. Cuando no se utiliza este pin, se configura como alta impedancia.

**4. AUD OUT (Audio Out).** Es el pin de salida de Audio.

**5. AM CAP (Auto Mute Feature).** Este pin permite, mediante un capacitor de  $1\mu\text{F}$  reducir el ruido hasta 6db en operaciones de reproducción.

**6. ANA IN- (Inverting Analog Input) ANA IN+ (Non Inverting Analog Input).** Estos pines permiten grabar señales de audio, mediante la ayuda de un micrófono. Cada entrada soporta hasta 16 mVpp, es decir, si se conecta el micrófono a las 2 entradas, soporta hasta 32 mVpp. Cabe aclarar, que se requiere un circuito sencillo adicional para conectar el micrófono al grabador (El circuito se incluye en el Anexo 3).

**7. INT (Interrupción).** Este pin indica cuando el grabador ha terminado una instrucción.

**8. XCLK (External Clock Input).** Este pin permite utilizar la señal de reloj generada por el microcontrolador para producir la frecuencia de muestreo necesaria para el Grabador. Esto se detalla mejor en la tabla 4.4:

**Tabla 4.4. Características de Frecuencia del Grabador de Voz**

Dispositivo	Frecuencia de Muestreo	Señal de reloj Requerida
ISD4004-8M	8.0 KHz	1024 KHz
ISD4004-10M	6.4 KHz	819.2 KHz
ISD4004-12M	5.3 KHz	682.7 KHz

ISD4004-16M	4.0 KHz	512 KHz
-------------	---------	---------

A continuación se muestran las instrucciones necesarias para programar el grabador de voz, es necesario enviar una instrucción formada por 3 bytes, distribuidos de la siguiente forma:

2 Bytes (A0 – A15) para direccionar el grabador.

1 Byte para activar los registros de control.

La figura 4.10 muestra la distribución de de los bytes (instrucción) enviados al grabador de voz.

Dirección																Bits de Control							
A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	x	x	x	C0	C1	C2	C3	C4

**Figura 4.10. Esquema de una Instrucción para el Grabador de Voz**

Hay que tomar en cuenta que el primer bit leído por el Grabador es A0, mientras q el primer bit enviado por el microcontrolador el es bit mas significativo MSB.

El significado de cada registro se muestra en la tabla 4.5:

**Tabla 4.5 Registros del Grabador de Voz**

Bit de Control	Registro de Control	Bit	Función
C0	MC	1	Función para adelantar la dirección. Activa la Función
		0	Deshabilita la Función
C1	IAB	1	Función para Ignorar la Dirección Ingresada Ignora la Dirección Ingresada (A0 – A15)
		0	Utiliza la dirección ingresada (A0 – A15)
C2	PU	1	Función Encendido Enciende
		0	Apaga

C3	P/R	1 0	Función Graba/Reproduce Reproduce Graba
C4	RUN	1 0	Función para Habilitar/Deshabilitar una Operación Inicia Para

Las Instrucciones se forman a partir de la combinación de los registros antes mencionados, las instrucciones mas importantes son presentadas en la tabla 4.6:

**Tabla 4.6 Instrucciones del Grabador de Voz**

Instrucción	Opcode		Descripción
	Dirección	Bits de Control XXXC0C1C3C4	
POWERUP	XXXX	XXX001000	Enciende el Grabador
SETPLAY	(A0-A15)	XXX00111	Inicia una operación de reproducción desde una dirección específica.
SETREC	(A0-A15)	XXX00101	Inicia una Operación de grabación desde una dirección específica.
STOP	XXXX	XXX011X0	Detiene una operación
STOPPWRDN	XXXX	XXXX10X0	Detiene una operación y apaga el grabador.

En el anexo 3 se encuentra la tabla completa de instrucciones.

Para realizar una operación de lectura o escritura en el grabador, no solo se debe enviar la instrucción, se debe realizar un procedimiento, el cual esta incluido en el anexo 3.

#### 4.3.2. Software para Utilizar el Grabador de Voz

Como se menciona anteriormente, las variables son almacenadas de forma global para tener acceso por cualquier función. Para el caso de las funciones del grabador de

voz se utilizan las variables “i” utilizadas para identificar la coordenada a la cual se refiere y la variable “condlong” la cual establece si la comparación es positiva.

La figura 4.11 muestra el diagrama de flujo que explica el proceso de salida de datos mediante el grabador de voz.

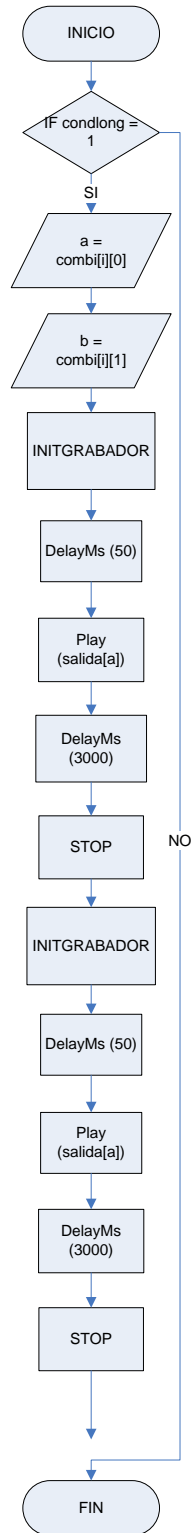


Figura. 4.11. Salida de Datos Mediante Grabador de Voz

En la tabla 4.7 se presenta los comandos almacenados en el grabador de voz y su correspondiente dirección de memoria.

**Tabla 4.7. Comandos de Voz**

<b>Índice "i"</b>	<b>Mensaje</b>	<b>Salida Hex</b>	<b>Salida</b>	<b>A7-A0</b>
0	Autopista General Rumiñahui	0x00	0B00000000	0B00000000
1	Se aproxima	0x08	0B00001000	0B00010000
2	Peaje	0x04	0B00000100	0B00100000
3	Puente Uno	0x0C	0B00001100	0B00110000
4	Puente Dos	0x02	0B00000010	0B01000000
5	Puente Tres	0x0A	0B00001010	0B01010000
6	Puente Cuatro	0x06	0B00000110	0B01100000
7	Puente Cinco	0x0E	0B00001110	0B01110000
8	Puente Seis	0x01	0B00000001	0B10000000
9	Puente Siete	0x09	0B00001001	0B10010000
10	Puente Ocho	0x05	0B00000101	0B10100000
11	Puente Nueve	0x0D	0B00001101	0B10110000
12	Nueva Oriental	0x03	0B00000011	0B11000000
13	Conocoto	0x0B	0B00001011	0B11010000
14	Parada de Buses	0x07	0B00000111	0B11100000
15	Reduzca la Velocidad	0x0F	0B00001111	0B11110000

La tabla 4.8 muestra el mensaje producido por el grabador de voz, así como sus coordenadas y su respectiva combinación de la tabla 4.7.



Tabla 4.8. Datos Generales de Comparación e Interpretación

Indice "i"	Salida GPS	Dato GPS		Rangos Mínimos		Rangos Máximos		Combi nación
	Aviso	Latitud	Longitud	(+1.8°)		(-1.8°)		
0	Autopista General Rumíñahui	0°17,6415'	78°27,7847'	0°17'40.29"	78°27'48.882"	0°17'36.69"	78°27'45.282"	0
1	Se aproxima Puente Nueve	0°17,4183'	78°27,9358'	0°17'26.898"	78°27'57.948"	0°17'23.298"	78°27'54.348"	(1-11)
2	Se aproxima Conocoto	0°17,0455'	78°28,1880'	0°17'4.53"	78°28'13.08"	0°17'0.93"	78°28'9.48"	(1-13)
3	Se aproxima Puente Ocho	0°16,9057'	78°28,2742'	0°16'56.142"	78°28'18.252"	0°16'52.542"	78°28'14.652"	(1-10)
4	Se aproxima Puente Siete	0°16,7792'	78°28,4785'	0°16'48.55"	78°28'30.51"	0°16'44.95"	78°28'26.91"	(1-9)
5	Se aproxima Puente Seis	0°16,7112'	78°28,8802'	0°16'44.47"	78°28'54.61"	0°16'40.87"	78°28'51.01"	(1-8)
6	Se aproxima Puente Cinco	0°16,5510'	78°29,3570'	0°16'34.86"	78°29'23.22"	0°16'31.26"	78°29'19.62"	(1-7)
7	Se aproxima Puente Cuatro	0°16,1742'	78°29,2605'	0°16'12.25"	78°29'17.43"	0°16'8.65"	78°29'13.83"	(1-6)
8	Se aproxima Puente Tres	0°15,8143'	78°29,2910'	0°15'50.66"	78°29'19.26"	0°15'47.06"	78°29'15.66"	(1-5)
9	Se aproxima Puente Dos	0°15,2448'	78°29,0300'	0°15'16.49"	78°29'3.6"	0°15'12.89"	78°29'0"	(1-4)
10	Se aproxima Puente Uno	0°14,7137'	78°29,1022'	0°14'44.62"	78°29'7.93"	0°14'41.02"	78°29'4.33"	(1-3)
11	Se aproxima Peaje	0°14,6140'	78°29,1205'	0°14'38.64"	78°29'9.03"	0°14'35.04"	0°14'5.43"	(1-2)
12	Se aproxima Parada de Buses	0°14,5027'	78°29,1398'	0°14'31.962"	78°29'10.188"	0°14'28.362"	78°29'6.588"	(1-14)
13	Se aproxima Nueva Oriental	0°14,2565'	78°29,0672'	0°14'17.19"	78°29'5.83"	0°14'13.59"	78°29'2.23"	(1-12)
14	Se aproxima Parada de Buses	0°13,6715'	78°29,1942'	0°13'42.09"	78°29'13.45"	0°13'38.49"	78°29'9.85"	(1-14)
15	Se aproxima Parada de Buses	0°13,6125'	78°29,4272'	0°13'38.55"	78°29'27.43"	0°13'34.95"	78°29'23.83"	(1-14)
16	Autopista General Rumíñahui	0°13,8645'	78°30,0835'	0°13'59.67"	78°30'6.81"	0°13'56.07"	78°30'3.21"	0

Además, en la figura 4.11 se observa las funciones para inicializar, reproducir y parar el grabador. Estas funciones se las explica mediante diagrama de flujos en las figuras 4.12, 4.13 y 4.14 respectivamente.

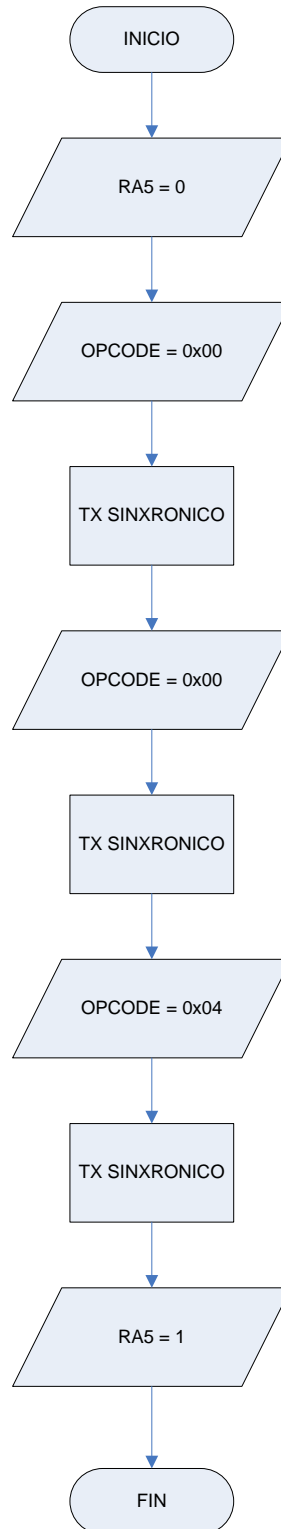


Figura 4.12. Inicialización del Grabador de Voz

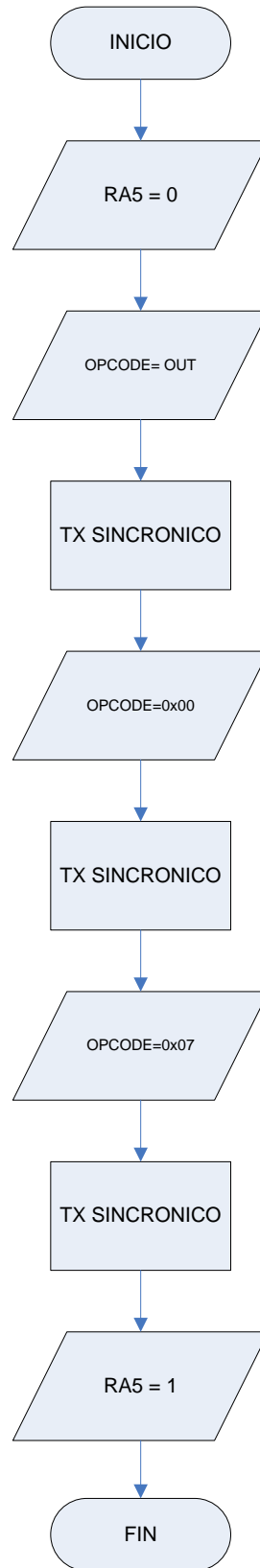
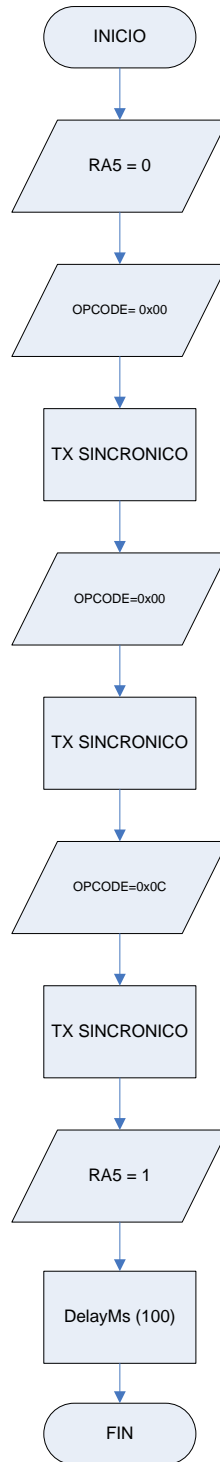
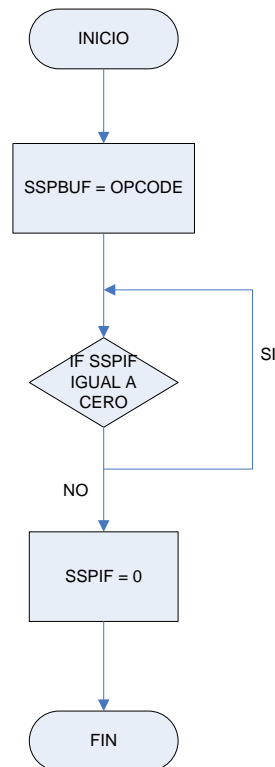


Figura 4.13. Reproducción en el Grabador de Voz



**Figura 4.14. Función para Detener el Grabador de Voz**

La figura 4.12, 4.13 y 4.14 correspondientes a las funciones del grabador necesitan de la transmisión de datos de forma sincrónica, la cual es explicada en la figura 4.15 mediante diagrama de flujo.



**Figura 4.15. Transmisión Sincrónica**

En las figuras 4.12, 4.13 y 4.14 se observa que el pin RA5 es desactivado y activado en cada proceso, esto porque este pin representa al selector de maestro o esclavo y cada que se envía un dato al grabador, éste debe cambiar su estado de esclavo al de maestro mediante la recepción de cualquier tipo de instrucción.

#### **4.4. PROGRAMA FINAL**

De esta forma el programa final del Microcontrolador se presenta en la figura 4.16.

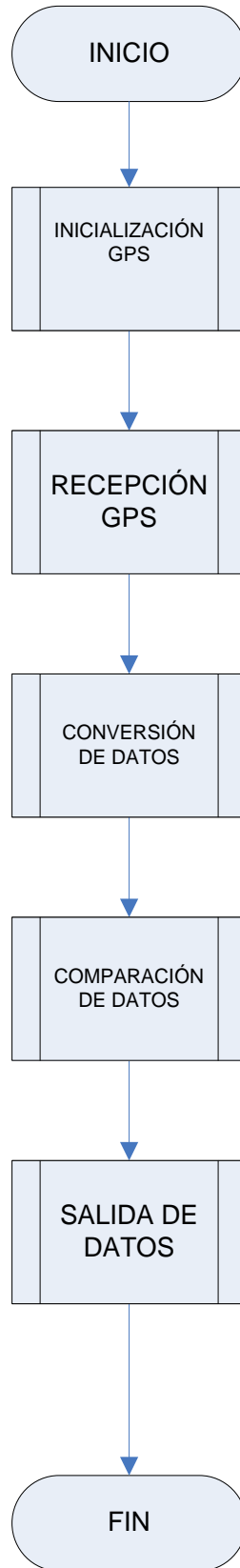


Figura 4.16. Programa Final

## CAPÍTULO V

### PRUEBAS Y ANÁLISIS ECONÓMICO

#### 5.1 PRUEBAS EN LA AUTOPISTA GENERAL RUMIÑAHUI

Para realizar las pruebas en la autopista general Rumiñahui se tomaron en cuenta ciertos aspectos, tales como:

- Espacio teórico de muestreo del GPS.
- Velocidad Máxima del Vehículo.
- Error del Módulo GPS.

##### 5.1.1. Espacio Teórico de Muestreo del GPS

Como se explicó en el capítulo anterior, en el programa se tomó un margen de muestreo de +/- 300 con respecto al valor obtenido por el módulo GPS, esto equivale a +/- 1.8" (segundos). En la tabla 4.8 se muestra el valor de cada punto con sus respectivos máximos y mínimos. Con la ayuda del software Google Earth se visualizó cada punto que se tomo de la autopista general Rumiñahui, así como los márgenes de muestreo del programa. En la figura 5.1 se presenta la imagen correspondiente a la entrada a la autopista General Rumiñahui desde Quito.



**Figura. 5.1. Entrada Autopista General Rumiñahui**

Las figuras correspondientes a los puntos restantes tomados de la Autopista General Rumiñahui se encuentran en el Anexo 4. La figura 5.2 muestra todos los puntos tomados en la Autopista General Rumiñahui.



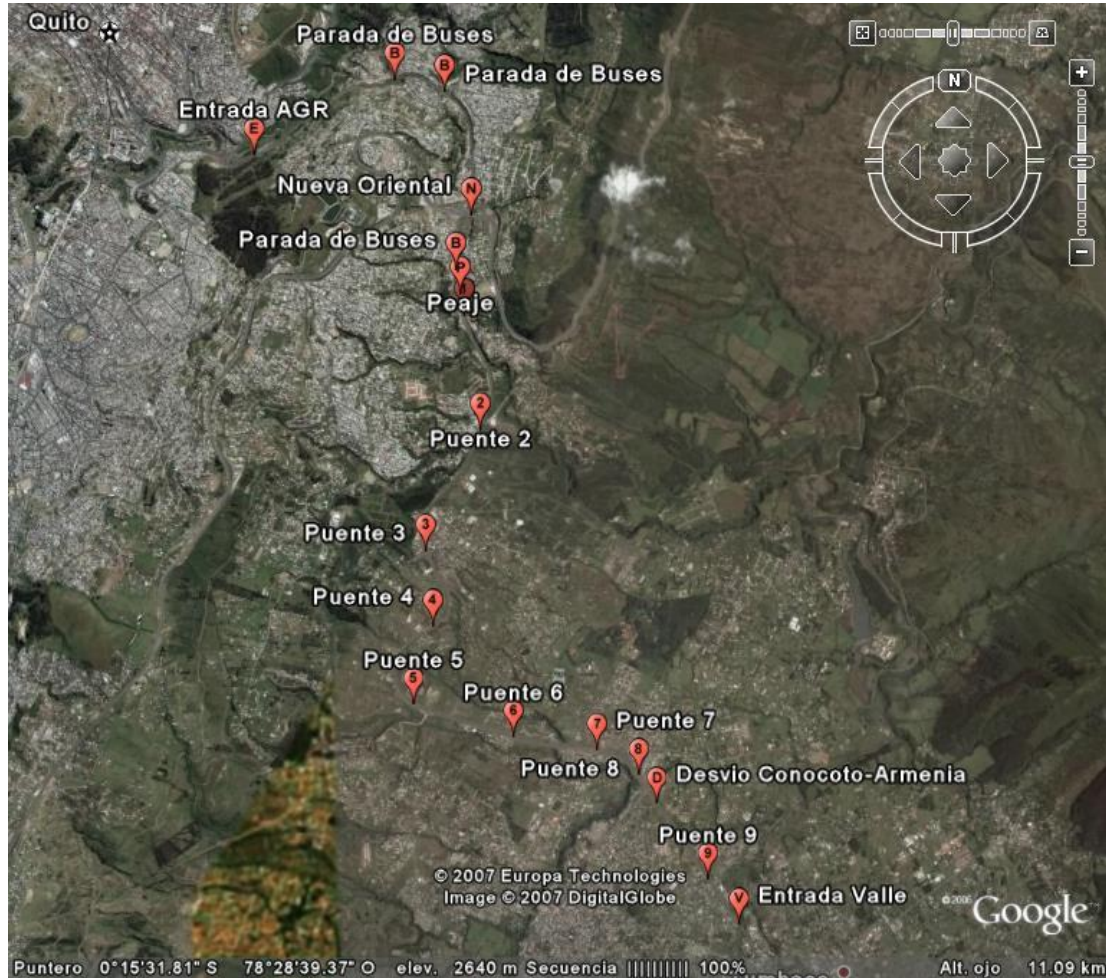


Figura. 5.2. Autopista General Rumiñahui

**5.1.2. Velocidad Máxima del Vehículo**

Con los puntos y márgenes de muestreo graficados, se puede obtener la distancia máxima (78.115 m) y mínima (44.52 m) que se muestrea con respecto del punto, teniendo una distancia promedio de 57.75 metros. A continuación se establece una velocidad máxima del vehículo de 60km/h, con lo cual se obtiene un anuncio del punto 3.46 segundos antes del punto. Esto se muestra en la ecuación 5.1:

$$60 \frac{km}{h} * \frac{1000 m}{1 km} * \frac{1 h}{3600 seg} = 16.6667 \frac{m}{s}$$

$$tiempo = \frac{Dist. promedio}{Veloc. \left( \frac{m}{s} \right)} = \frac{57.75 m}{16.6667 \frac{m}{s}} = 3.46 segundos \quad Ecuac. 5.1$$

### 5.1.3. Error del Módulo GPS

El datasheet del módulo GPS proporciona un valor de error máximo de +/- 10 metros, dependiendo del número de satélites con los que se realice la ubicación del punto. Con este valor, el tiempo en el que se produce un anuncio puede aumentar o disminuir 0.5 segundos.

Teniendo en cuenta los datos mencionados anteriormente, se procedió a realizar las pruebas en la Autopista General Rumiñahui en recorrido Valle – Quito y viceversa. La primera prueba se realizó con velocidad promedio de 50 km/h (menor a la velocidad máxima establecida) y la segunda prueba con una velocidad mayor a 60 km/h (Velocidad promedio 70 km/h). Las tablas 5.1 y 5.2 representan los resultados obtenidos en la primera y segunda prueba respectivamente.

Tabla. 5.1. Primera Prueba Realizada (Menor a 60 Km/h)

Sentido	Punto	Aviso	Distancia Antes del Punto (m)	Tiempo Antes de Punto (seg)
Valle - Quito	Entrada Valle	Autopista General Rumiñahui	61	3.66
	Puente 9	Se Aproxima Puente 9	64	3.83
	Conocoto – Armenia	Se Aproxima Conocoto	68	4.08
	Puente 8	Se Aproxima Puente 8	65	3.9
	Puente 7	Se Aproxima Puente 7	50	3
	Puente 6	Se Aproxima Puente 6	55	3.3
	Puente 5	Se Aproxima Puente 5	63	3.78
	Puente 4	Se Aproxima Puente 4	75	4.5
	Puente 3	Se Aproxima Puente 3	55	3.3
	Puente 2	Se Aproxima Puente 2	57	3.42
	Puente 1	Se Aproxima Puente 1	56	3.36
	Peaje	Se Aproxima Peaje	45	2.7
	Parada de Buses	Se Aproxima Parada de Buses	59	3.54
	Nueva Oriental	Se Aproxima Nueva Oriental	56	3.36
	Parada de Buses	Se Aproxima Parada de Buses	59	3.54
Parada de Buses	Se Aproxima Parada de Buses	60	3.6	
Entrada Trébol	Autopista General Rumiñahui	37	2.22	
Quito - Valle	Entrada Trébol	Autopista General Rumiñahui	60	3.6
	Parada de Buses	Se Aproxima Parada de Buses	68	4.08
	Parada de Buses	Se Aproxima Parada de Buses	58	3.48
	Nueva Oriental	Se Aproxima Nueva Oriental	52	3.12
	Parada de Buses	Se Aproxima Parada de Buses	49	2.94
	Peaje	Se Aproxima Peaje	51	3.06
	Puente 1	Se Aproxima Puente 1	63	3.78
	Puente 2	Se Aproxima Puente 2	55	3.23
	Puente 3	Se Aproxima Puente 3	60	3.6
	Puente 4	Se Aproxima Puente 4	70	4.2
	Puente 5	Se Aproxima Puente 5	62	3.72
	Puente 6	Se Aproxima Puente 6	55	3.3
	Puente 7	Se Aproxima Puente 7	56	3.36
	Puente 8	Se Aproxima Puente 8	72	4.32
	Conocoto – Armenia	Se Aproxima Conocoto	66	3.96
Puente 9	Se Aproxima Puente 9	65	3.9	
Entrada Valle	Autopista General Rumiñahui	60	3.6	

Tabla 5.2 Segunda Prueba Realizada ( Veloc. Mayor a 60 Km/h)

Sentido	Punto	Aviso	Distancia Antes del Punto (m)	Tiempo Antes de Punto (seg)
Valle - Quito	Entrada Valle	Autopista General Rumiñahui	40	2.05
	Puente 9	Se Aproxima Puente 9	52	2.67
	Conocoto – Armenia	Se Aproxima Conocoto	58	2.98
	Puente 8	Se Aproxima Puente 8	55	2.83
	Puente 7	Se Aproxima Puente 7	43	2.21
	Puente 6	Se Aproxima Puente 6	48	2.47
	Puente 5	Se Aproxima Puente 5	50	2.57
	Puente 4	Se Aproxima Puente 4	59	3.03
	Puente 3	Se Aproxima Puente 3	47	2.41
	Puente 2	Se Aproxima Puente 2	43	2.21
	Puente 1	Se Aproxima Puente 1	49	2.52
	Peaje	Se Aproxima Peaje	40	2.05
	Parada de Buses	Se Aproxima Parada de Buses	60	3.08
	Nueva Oriental	Se Aproxima Nueva Oriental	47	2.41
	Parada de Buses	Se Aproxima Parada de Buses	48	2.47
Parada de Buses	Se Aproxima Parada de Buses	53	2.73	
Entrada Trébol	Autopista General Rumiñahui	33	1.7	
Quito - Valle	Entrada Trébol	Autopista General Rumiñahui	53	2.73
	Parada de Buses	Se Aproxima Parada de Buses	56	2.88
	Parada de Buses	Se Aproxima Parada de Buses	49	2.52
	Nueva Oriental	Se Aproxima Nueva Oriental	46	2.37
	Parada de Buses	Se Aproxima Parada de Buses	42	2.16
	Peaje	Se Aproxima Peaje	45	2.31
	Puente 1	Se Aproxima Puente 1	51	2.62
	Puente 2	Se Aproxima Puente 2	49	2.52
	Puente 3	Se Aproxima Puente 3	52	2.67
	Puente 4	Se Aproxima Puente 4	56	2.88
	Puente 5	Se Aproxima Puente 5	53	2.73
	Puente 6	Se Aproxima Puente 6	50	2.57
	Puente 7	Se Aproxima Puente 7	45	2.31
	Puente 8	Se Aproxima Puente 8	59	3.03
	Conocoto – Armenia	Se Aproxima Conocoto	52	2.67
Puente 9	Se Aproxima Puente 9	50	2.57	
Entrada Valle	Autopista General Rumiñahui	48	2.47	

Los valores de distancia antes del punto que presentan las tablas 5.1 y 5.2 fueron obtenidos a partir de los datos proporcionados por el automóvil y a partir de estos datos se obtuvieron los valores del tiempo que presentan las tablas 5.1 y 5.2 con la ayuda de la

ecuación 5.1. El valor de velocidad máxima de la ecuación 5.1 es de 16.667 m/s para la primera prueba y para la segunda prueba es de 19.44 m/s (equivalente a 70 km/h).

En 3 puntos, Conocoto – Armenia, Puente 3 y Nueva Oriental, el aviso auditivo se presentó como se muestra en las tablas 5.1 con un tiempo y distancia prudente, sin embargo, en los 3 puntos, la advertencia auditiva se presentó luego de pasar sus respectivos desvíos. Esto quiere decir que el sistema sirve para anunciar el respectivo punto, más no, para anunciar el respectivo desvío.

En la tabla 5.1 se observa que los datos obtenidos proporcionan un correcto aviso preventivo del punto.

En tabla 5.2 por el contrario, se presenta las advertencias muy cercanas al punto y con un tiempo menor al punto, esto se debe al aumento de velocidad, ya que el modulo GPS fue configurado para adquirir datos cada 1 segundo, y en la segunda prueba en un segundo se recorre aproximadamente 20 metros.

## **5.2. ANÁLISIS DE COSTOS**

La tabla 5.3 muestra el costo de todos los elementos utilizados en la elaboración del presente proyecto.

Tabla 5.3. Análisis de Costos del Dispositivo

CANTIDAD	DESCRIPCIÓN	V. UNITARIO	V. TOTAL
1	Proto Board	20	20
1	Módulo GPS EM-401	83	99.6
1	Grabador de Voz ISD4004-8MP	24	28.8
1	Microcontrolador PIC16F877A	11.50	11.50
1	Programador PIC'S	17	17
2	Cristal 4 MHz	1	2
2	LM7805	0.50	1
2	LM317	0.85	1.70
4	Capacitor 1000uF 16 V	0.30	1.20
8	Capacitor 0.1 uF	0.10	0.80
10	Capacitor 1 uF 50 V	0.10	1
2	Capacitor 22 uF 16 V	0.10	0.20
4	Capacitor 33 pF	0.10	0.40
2	Resistencia 330 $\Omega$	0.05	0.10
2	Resistencia 220 $\Omega$	0.05	0.10
4	Resistencia 4.7 K $\Omega$	0.05	0.20
4	Resistencia 4.9 K $\Omega$	0.05	0.20
4	Diodo 1N4004	0.10	0.40
2	Pulsador	0.20	0.40
1	Zócalo 40 pines	0.39	0.39
1	Zócalo 28 pines	0.70	0.70
1	Placa un lado impresa	45	45
		<b>Valor Total</b>	232.69

La tabla 5.3 muestra la cantidad, el valor unitario de cada elemento y el valor total del sistema diseñado, sin contar con los elementos dañados o mal diseñados en el proceso. Además, no se toma en cuenta las horas trabajadas en el proceso. En el proceso se realizó el dispositivo tanto en proto board como en un placa final, por lo que, la cantidad de elementos que se presentan en la tabla 5.3 están tomados a partir de los 2 sistemas elaborados.

## CAPÍTULO VI

### CONCLUSIONES Y RECOMENDACIONES

#### 6.1. CONCLUSIONES

- El sistema de asistencia vehicular, puede tener diferentes aplicaciones a las que ha sido creada, por ejemplo, en la ayuda a personas no videntes a orientarse en lugares determinados.
- En el sistema diseñado se puede añadir avisos auditivos, para ayudar al conductor a mejorar su seguridad en el vehículo, teniendo alertas para: avisos de puertas abiertas, nivel de gasolina, etc, con la ayuda de los diferentes puertos del microcontrolador.
- Para un mejor tiempo de respuesta del sistema diseñado, es necesario aumentar el rango de muestreo de los puntos almacenados en el microcontrolador (tabla 4.8) para que los avisos generados por el sistema diseñado se presenten con un buen margen de tiempo.
- Se debe colocar el sistema diseñado de modo que el módulo GPS tenga vista al cielo para mejorar la recepción del sistema.
- El módulo GPS y el grabador de Voz pueden ser reemplazados por otros de mejores características (en este proyecto se emplearon los elementos con las menores características de cada familia por motivos de disponibilidad), porque

sus puertos están estandarizados dentro de sus respectivas familias de elementos y físicamente son de igual forma y tamaño.

- Para obtener una mayor exactitud al comparar los puntos almacenados en el microcontrolador, es necesario aumentar la frecuencia a la cual el módulo GPS envía los datos al microcontrolador.



## 6.2. RECOMENDACIONES

- Se recomienda añadir una etapa de amplificación de audio al sistema diseñado, porque al momento de acoplarlo a diferentes sistemas de audio vehicular, produce diferentes niveles de sonido, en algunos casos obteniendo muy bajo volumen.
- Se recomienda añadir, al diseño y placa final, una etapa de puertos (utilizando los puertos libres del microcontrolador) para poder aumentar funciones al sistema diseñado.
- Se recomienda cambiar el módulo GPS por uno de diferente voltaje para eliminar una de las dos fuentes diseñadas para el sistema y conseguir una disminución en el tamaño de la placa. En el proyecto no se pudo conseguir el módulo GPS que opera a 3 voltios por motivos de disponibilidad.
- Se recomienda cambiar el módulo GPS por un módulo de mejores características para disminuir el porcentaje de error.
- Si se desea disminuir el tamaño de la placa, se recomienda reemplazar el tipo de encapsulado (SOIC/PDIP) del microcontrolador y del grabador de voz por uno de menor tamaño (QFN).

## ANEXOS

## ANEXO I

## Output Messages



Table 1-1 lists each of the NMEA output messages specifically developed and defined by SiRF for use within SiRF products.

Table 1-1 NMEA Output Messages

Option	Description
GGA	Time, position and fix type data.
GLL	Latitude, longitude, UTC time of position fix and status.
GSA	GPS receiver operating mode, satellites used in the position solution, and DOP values.
GSV	The number of GPS satellites in view satellite ID numbers, elevation, azimuth, and SNR values.
MSS	Signal-to-noise ratio, signal strength, frequency, and bit rate from a radio-beacon receiver.
RMC	Time, date, position, course and speed data.
VTG	Course and speed information relative to the ground.
ZDA	PPS timing message (synchronized to PPS).
150	OK to send message.

A full description of the listed NMEA messages are provided in the following sections.

Table 1-2 provides a summary of SiRF NMEA output messages supported by the specific SiRF platforms.

Table 1-2 Supported NMEA Output Messages

Message	SiRF Software			
	GSW2	SiRFXTrac	SiRFLoc	GSW3
GGA	Ye	Ye	Ye	Ye
GLL	Ye	Ye	Ye	Ye
GSA	Ye	Ye	Ye	Ye
GSV	Ye	Ye	Ye	Ye
MS	Ye	No	No	No
RMC	Ye	Ye	Ye	Ye
VTG	Ye	Ye	Ye	Ye
ZDA	2.3.2 and above	No	No	No
15	2.3.2 and above	No	No	No

**Note** – GSW2 software only outputs NMEA version 2.20 (and earlier). XTrac and GSW3 software have conditional defines (UI\_NMEA\_VERSION\_XXX) to allow a choice between NMEA 2.20 and 3.00. The file `NMEA_SIF.H` contains the NMEA version defines.

## GGA—Global Positioning System Fixed Data

**Note** – Fields marked in italic *red* apply only to NMEA version 2.3 (and later) in this NMEA message description.

Table 1-3 contains the values for the following example:

```
$GPGGA,161229.487,3723.2475,N,12158.3416,W,1,07,1.0,9.0,M, , , ,0000*18
```

Table 1-3 GGA Data Format

Nam	Example	Units	Descriptio
Message ID	\$GPGGA		GGA protocol header
UTC Time	161229.487		hhmmss.sss
Latitude	3723.2475		ddmm.mmmm
N/S Indicator	N		N=north or S=south
Longitude	12158.3416		dddmm.mmmm
E/W Indicator	W		E=east or W=west
Position Fix Indicator	1		See Table 1-4
Satellites Used	07		Range 0 to 12
HDOP	1.0		Horizontal Dilution of Precision
MSL Altitude	9.0	meters	
Units	M	meters	
Geoid Separation		meters	
Units	M	meters	
Age of Diff. Corr.		second	Null fields when DGPS is not used
Diff. Ref. Station ID	0000		
Checksum	*18		
<CR> <LF>			End of message termination

Table 1-4 Position Fix Indicator

Value	Descriptio
0	Fix not available or invalid
1	GPS SPS Mode, fix valid
2	Differential GPS, SPS Mode, fix valid
3-5	Not supported
<i>6</i>	<i>Dead Reckoning Mode, fix valid</i>

**Note** – A valid position fix indicator is derived from the SiRF Binary M.I.D. 2 position mode 1. See the *SiRF Binary Protocol Reference Manual*.

## GLL—Geographic Position - Latitude/Longitude

**Note** – Fields marked in italic *red* apply only to NMEA version 2.3 (and later) in this NMEA message description.

Table 1-5 contains the values for the following example:

\$GPGLL, 3723.2475,N,12158.3416,W,161229.487,A,A\*41

Table 1-5 GLL Data Format

Nam	Example	Units	Descriptio
Message ID	\$GPGLL		GLL protocol header
Latitude	3723.2475		ddmm.mmmm
N/S Indicator	N		N=north or S=south
Longitude	12158.3416		dddmm.mmmm
E/W Indicator	W		E=east or W=west
UTC Time	161229.487		hhmmss.sss
Status	A		A=data valid or V=data not valid
<i>Mode</i>	<i>A</i>		<i>A=Autonomous, D=DGPS, E=DR (Only present in NMEA version</i>
Checksum	*41		
<CR> <LF>			End of message termination

## GSA—GNSS DOP and Active Satellites

**Note** – Fields marked in italic *red* apply only to NMEA version 2.3 (and later) in this NMEA message description.

Table 1-6 contains the values for the following example:

\$GPGSA,A,3,07,02,26,27,09,04,15,,,,,1.8,1.0,1.5\*33

Table 1-6 GSA Data Format

Nam	Example	Units	Descriptio
Message ID	\$GPGSA		GSA protocol header
Mode 1	A		See Table 1-7
Mode 2	3		See Table 1-8
Satellite Used <sup>1</sup>	07		Sv on Channel 1
Satellite Used <sup>1</sup>	02		Sv on Channel 2
...			...
Satellite Used <sup>1</sup>			Sv on Channel 12
PDOP	1.8		Position Dilution of Precision
HDOP	1.0		Horizontal Dilution of Precision
VDOP	1.5		Vertical Dilution of Precision
Checksum	*33		
<CR> <LF>			End of message termination

1. Satellite used in solution.

Table 1-7 Mode 1

Value	Descriptio
M	Manual—forced to operate in 2D or 3D mode
A	2D Automatic—allowed to automatically switch

Table 1-8 Mode 2

Value	Description
1	Fix not available
2	2D (<4 SVs used)
3	3D (>3 SVs used)

## GSV—GNSS Satellites in View

Table 1-9 contains the values for the following example:

```
$GPGSV,2,1,07,07,79,048,42,02,51,062,43,26,36,256,42,27,27,138,42*71
$GPGSV,2,2,07,09,23,313,42,04,19,159,41,15,12,041,42*41
```

Table 1-9 GSV Data Format

Nam	Example	Units	Descriptio
Message ID	\$GPGSV		GSV protocol header
Number of	2		Range 1 to 3
Message Number <sup>1</sup>	1		Range 1 to 3
Satellites in View	07		
Satellite ID	07		Channel 1 (Range 1 to 32)
Elevation	79	degrees	Channel 1 (Maximum 90)
Azimuth	048	degrees	Channel 1 (True, Range 0 to 359)
SNR (C/No)	42	dBHz	Range 0 to 99, null when not tracking
...			...
Satellite ID	27		Channel 4 (Range 1 to 32)
Elevation	27	degrees	Channel 4 (Maximum 90)
Azimuth	138	degrees	Channel 4 (True, Range 0 to 359)
SNR (C/No)	42	dBHz	Range 0 to 99, null when not tracking
Checksum	*71		
<CR> <LF>			End of message termination

1. Depending on the number of satellites tracked, multiple messages of GSV data may be required.

## MSS—MSK Receiver Signal

**Note** – Fields marked in italic *red* apply only to NMEA version 2.3 (and later) in this NMEA message description.

Table 1-10 contains the values for the following example:

```
$GPMSS,55,27,318.0,100,1,*
57
```

Table 1-10 MSS Data Format

Nam	Example	Units	Descriptio
Message ID	\$GPMSS		MSS protocol header
Signal Strength	55	dB	SS of tracked frequency
Signal-to-Noise Ratio	27	dB	SNR of tracked frequency
Beacon Frequency	318.0	kHz	Currently tracked frequency
Beacon Bit Rate	100		bits per second
<i>Channel Number</i>	<i>1</i>		<i>The channel of the beacon being used if a multi-channel beacon receiver is</i>
Checksum	*57		
<CR> <LF>			End of message termination



**Note** – The MSS NMEA message can only be polled or scheduled using the MSK NMEA input message. See “MSK—MSK Receiver Interface” on page 2-8.

## RMC—Recommended Minimum Specific GNSS Data

**Note** – Fields marked in italic *red* apply only to NMEA version 2.3 (and later) in this NMEA message description.

Table 1-11 contains the values for the following example:

```
$GPRMC,161229.487,A,3723.2475,N,12158.3416,W,0.13,309.62,120598,*,*10
```

Table 1-11 RMC Data Format

Nam	Example	Units	Descriptio
Message ID	\$GPRMC		RMC protocol header
UTC Time	161229.487		hhmmss.sss
Status <sup>1</sup>	A		A=data valid or V=data not valid
Latitude	3723.2475		ddmm.mmmm
N/S Indicator	N		N=north or S=south
Longitude	12158.3416		dddmm.mmmm
E/W Indicator	W		E=east or W=west
Speed Over Ground	0.13	knots	
Course Over Ground	309.62	degrees	True
Date	120598		ddmmyy
Magnetic Variation <sup>2</sup>		degrees	E=east or W=west
<i>Mode</i>	<i>A</i>		<i>A=Autonomous, D=DGPS, E=DR</i>
Checksum	*10		
<CR> <LF>			End of message termination

1. A valid status is derived from the SiRF Binary M.I.D 2 position mode 1. See the *SiRF Binary Protocol Reference Manual*.

2. SiRF Technology Inc. does not support magnetic declination. All “course over ground” data are geodetic WGS84 directions.

## VTG—Course Over Ground and Ground Speed

**Note** – Fields marked in italic *red* apply only to NMEA version 2.3 (and later) in this NMEA message description.

Table 1-12 contains the values for the following example:

```
$GPVTG,309.62,T,
,M,0.13,N,0.2,K,A*23
```

Table 1-12 VTG Data Format

Nam	Example	Units	Descriptio
Message ID	\$GPVTG		VTG protocol header
Course	309.62	degrees	Measured heading
Reference	T		True
Course		degrees	Measured heading
Reference	M		Magnetic <sup>1</sup>
Speed	0.13	knots	Measured horizontal speed
Units	N		Knots
Speed	0.2	km/hr	Measured horizontal speed
Units	K		Kilometers per hour
<i>Mode</i>	<i>A</i>		<i>A=Autonomous, D=DGPS, E=DR</i>
Checksum	*23		
<CR> <LF>			End of message termination

1. SiRF Technology Inc. does not support magnetic declination. All "course over ground" data are geodetic WGS84 directions.

## ZDA—SiRF Timing Message

Outputs the time associated with the current 1 PPS pulse. Each message is output within a few hundred ms after the 1 PPS pulse is output and tells the time of the pulse that just occurred.

Table 1-13 contains the values for the following example:

```
$GPZDA,181813,14,10,2003,00,00*
4F
```

Table 1-13 ZDA Data Format

Nam	Example	Units	Descriptio
Message ID	\$GPZDA		ZDA protocol header
UTC time	181813		Either using valid IONO/UTC or estimated from default leap seconds
Day	14		01 TO 31
Month	10		01 TO 12
Year	2003		1980 to 2079
Local zone hour	00	knots	Offset from UTC (set to 00)
Local zone minutes	00		Offset from UTC (set to 00)
Checksum			
<CR> <LF>			End of message termination

## 150—OkToSend

This message is being sent out during the trickle power mode to communicate with an outside program such as SiRFDemo to indicate whether the receiver is awake or not. Table 1-14 contains the values for the following examples:

1. OkToSend

\$PSRF150,1\*3F

2. not OkToSend

\$PSRF150,0\*3E

Table 1-14 OkToSend Message Data Format

Nam	Example	Units	Descriptio
Message ID	\$PSRF150		PSRF150 protocol header
OkToSend	1		1=OK to send, 0=not OK to send
Checksum	*3F		
<CR> <LF>			End of message termination

## *Input Messages*

---



NMEA input messages enable you to control the Evaluation Receiver while in NMEA protocol mode. The Evaluation Receiver may be put into NMEA mode by sending the SiRF binary protocol message “Switch to NMEA Protocol - Message I.D. 129” (see the *SiRF Binary Protocol Reference Manual*). This can be done by using a user program or by using the SiRFSDemo software and selecting Switch to NMEA Protocol from the Action menu (see the *SiRF Evaluation Kit User Guide* or the *SiRFDemo User Guide*). If the receiver is in SiRF binary mode, all NMEA input messages are ignored. Once the receiver is put into NMEA mode, the following messages may be used to command the module.

### *Transport Message*

Table 2-1 describes the transport message parameters.

*Table 2-* Transport Message parameters  
1



Start Sequence	Payload	Checksum	End Sequence
\$PSRF<MID> <sup>1</sup>	Data <sub>2</sub>	*CKSUM <sup>3</sup>	<CR> <LF> <sup>4</sup>

1. Message Identifier consisting of three numeric characters. Input messages begin at MID 100.
2. Message specific data. Refer to a specific message section for <data>...<data> definition.
3. CKSUM is a two-hex character checksum as defined in the NMEA specification, *NMEA-0183 Standard For Interfacing Marine Electronic Devices*. Use of checksums is required on all input messages.
4. Each message is terminated using Carriage Return (CR) Line Feed (LF) which is \r\n which is hex 0D 0A. Because \r\n are not printable ASCII characters, they are omitted from the example strings, but must be sent to terminate the message and cause the receiver to process that input message.

**Note –** All fields in all proprietary NMEA messages are required, none are optional.

All NMEA messages are comma delimited.

## NMEA Input Messages

Table 2-2 describes the NMEA input messages.

Table 2-2 NMEA Input Messages

Message	MID <sup>1</sup>	Description
SetSerialPort	100	Set PORT A parameters and protocol
NavigationInitialization	101	Parameters required for start using X/Y/Z <sup>2</sup>
SetDGPSPort	102	Set PORT B parameters for DGPS input
Query/Rate Control	103	Query standard NMEA message and/or set output
LLANavigationInitialization	104	Parameters required for start using Lat/Lon/Alt <sup>3</sup>
Development Data On/Off	105	Development Data messages On/Off
Select Datum	106	Selection of datum to be used for coordinate transformations.
MSK Receiver Interface	MSK	Command message to a MSK radio-beacon

1. Message Identification (MID).
2. Input coordinates must be WGS84.
3. Input coordinates must be WGS84.

**Note –** NMEA input messages 100 to 106 are SiRF proprietary NMEA messages. The MSK NMEA string is as defined by the NMEA 0183 standard.

Table 2-3 provides a summary of supported SiRF NMEA input messages by the specific SiRF platforms.

Table 2-3 Supported NMEA Input Messages

Message ID	SiRF Software Options		
	GSW2	SiRFXTrac	SiRFLoc
10	Ye	Ye	Ye
10	Ye	No	Ye
10	Ye	No	No
10	Ye	Ye	Ye
10	Ye	No	Ye
10	Ye	Ye	Ye
10	Ye	Ye	Ye
MSK	Ye	No	No

## 100—SetSerialPort

This command message is used to set the protocol (SiRF binary or NMEA) and/or the communication parameters (Baud, data bits, stop bits, and parity). Generally, this command is used to switch the module back to SiRF binary protocol mode where a more extensive command message set is available. When a valid message is received, the parameters are stored in battery-backed SRAM and the Evaluation Receiver restarts using the saved parameters.

Table 2-4 contains the input values for the following example:

Switch to SiRF binary protocol at 9600,8,N,1

```
$PSRF100,0,9600,8,1,0*0
C
```

Table 2-4 Set Serial Port Data Format

Nam	Example	Units	Descriptio
Message ID	\$PSRF100		PSRF100 protocol header
Protocol	0		0=SiRF binary, 1=NMEA
Baud	9600		4800, 9600, 19200, 38400
DataBits	8		8,7 <sup>1</sup>
StopBits	1		0,1
Parity	0		0=None, 1=Odd, 2=Even
Checksum	*0C		
<CR> <LF>			End of message termination

1. SiRF protocol is only valid for 8 data bits, 1stop bit, and no parity.

## 101—Navigation Initialization

This command is used to initialize the Evaluation Receiver by providing current position (in X, Y, Z coordinates), clock offset, and time. This enables the Evaluation Receiver to search for the correct satellite signals at the correct signal parameters. Correct initialization parameters enable the Evaluation Receiver to acquire signals quickly.

Table 2-5 contains the input values for the following

example: Start using known position and time.

```
$PSRF101,-2686700,-4304200,3851624,96000,497260,921,12,3*1C
```

Table 2-5 Navigation Initialization Data Format

Nam	Example	Units	Descriptio
Message ID	\$PSRF101		PSRF101 protocol header
ECEF X	-2686700	meters	X coordinate position
ECEF Y	-4304200	meters	Y coordinate position
ECEF Z	3851624	meters	Z coordinate position
ClkOffset	96000	Hz	Clock Offset of the Evaluation
TimeOfWeek	497260	seconds	GPS Time Of Week
WeekNo	921		GPS Week Number
ChannelCount	12		Range 1 to 12
ResetCfg	3		See Table 2-6 and Table 2-7
Checksum	*1C		
<CR> <LF>			End of message termination

1. Use 0 for last saved value if available. If this is unavailable, a default value of 96,000 is used.



Table 2-6 Reset Configuration - Non SiRFLoc Platforms

Hex	Descriptio
0x01	Hot Start— All data valid
0x02	Warm Start—Ephemeris cleared
0x03	Warm Start (with Init)—Ephemeris cleared, initialization data loaded
0x04	Cold Start—Clears all data in memory
0x08	Clear Memory—Clears all data in memory and resets the receiver back to

Table 2-7 Reset Configuration - SiRFLoc Specific

Hex	Descriptio
0x00	Perform a hot start using internal RAM data. No initialization data is used.
0x01	Use initialization data and begin in start mode. Uncertainties are 5 seconds time accuracy and 300 km position accuracy. Ephemeris data in
0x02	No initialization data is used, ephemeris data is cleared, and warm start
0x03	Initialization data is used, ephemeris data is cleared, and warm start performed using remaining data in RAM.
0x04	No initialization data is used. Position, time and ephemeris are cleared and a cold start is performed.
0x08	No initialization data is used. Internal RAM is cleared and a factory reset is performed.

## 102—SetDGPSPort

This command is used to control the serial port used to receive RTCM differential corrections. Differential receivers may output corrections using different communication parameters. If a DGPS receiver is used that has different communication parameters, use this command to allow the receiver to correctly decode the data. When a valid message is received, the parameters are stored in battery-backed SRAM and the receiver restarts using the saved parameters.

Table 2-8 contains the input values for the following

example: Set DGPS Port to be 9600,8,N,1.

\$PSRF102,9600,8,1,0\*  
12

Table 2-8 Set DGPS Port Data Format

Nam	Example	Units	Descriptio
Message ID	\$PSRF102		PSRF102 protocol header
Baud	9600		4800, 9600, 19200, 38400
DataBits	8		8,7
StopBits	1		0,1
Parity	0		0=None, 1=Odd, 2=Even
Checksum	*12		
<CR> <LF>			End of message termination

## 103—Query/Rate Control

This command is used to control the output of standard NMEA messages GGA, GLL, GSA, GSV, RMC, and VTG. Using this command message, standard NMEA messages may be polled once, or setup for periodic output. Checksums may also be enabled or disabled depending on the needs of the receiving program. NMEA message settings are saved in battery-backed memory for each entry when the message is accepted.

Table 2-9 contains the input values for the following examples:

1. Query the GGA message with checksum enabled

```
$PSRF103,00,01,00,01*25
```

2. Enable VTG message for a 1 Hz constant output with checksum enabled

```
$PSRF103,05,00,01,01*20
```

3. Disable VTG message

```
$PSRF103,05,00,00,01*21
```

Table 2-9 Query/Rate Control Data Format (See example 1)

Name	Example	Units	Descriptio
Message ID	\$PSRF103		PSRF103 protocol header
Msg	00		See Table 2-10
Mode	01		0=SetRate, 1=Query
Rate	00	seconds	Output—off=0, max=255
CksumEnable	01		0=Disable Checksum, 1=Enable Checksum
Checksum	*25		
<CR> <LF>			End of message termination

Table 2-10 Messages

Value	Descriptio
0	GGA
1	GLL
2	GSA
3	GSV
4	RMC
5	VTG
6	MSS (If internal beacon is supported)
7	Not defined
8	ZDA (if 1PPS output is supported)
9	Not defined

**Note** – In TricklePower mode, update rate is specified by the user. When switching to NMEA protocol, the message update rate is also required. The resulting update rate is the product of the TricklePower Update rate and the NMEA update rate (i.e., TricklePower update rate = 2 seconds, NMEA update rate = 5 seconds, resulting update rate is every 10 seconds, (2 X 5 = 10)).

## 104—LLA Navigation Initialization

This command is used to initialize the Evaluation Receiver by providing current position (in latitude, longitude, and altitude coordinates), clock offset, and time. This enables the receiver to search for the correct satellite signals at the correct signal parameters. Correct initialization parameters enable the receiver to acquire signals quickly.

Table 2-11 contains the input values for the following

example: Start using known position and time.

```
$PSRF104,37.3875111,-121.97232,0,96000,237759,1946,12,1*07
```

Table 2-11 LLA Navigation Initialization Data Format

Name	Example	Units	Descriptio
Message ID	\$PSRF104		PSRF104 protocol header
Lat	37.3875111	degrees	Latitude position (Range 90 to -90)
Lon	-121.97232	degrees	Longitude position (Range 180 to -180)
Alt	0	meters	Altitude position
ClkOffset	96000	Hz	Clock Offset of the Evaluation Receiver <sup>1</sup>
TimeOfWeek	237759	seconds	GPS Time Of Week
WeekNo	1946		Extended GPS Week Number (1024
ChannelCount	12		Range 1 to 12
ResetCfg	1		See Table 2-12
Checksum	*07		
<CR> <LF>			End of message termination

1. Use 0 for last saved value if available. If this is unavailable, a default value of 96,000 is used.

Table 2-12 Reset Configuration

Hex	Descriptio
0x01	Hot Start— All data valid
0x02	Warm Start—Ephemeris cleared
0x03	Warm Start (with Init)—Ephemeris cleared, initialization data loaded
0x04	Cold Start—Clears all data in memory
0x08	Clear Memory—Clears all data in memory and

## 105—Development Data On/Off

Use this command to enable development data information if you are having trouble getting commands accepted. Invalid commands generate debug information that enables the you to determine the source of the command rejection. Common reasons for input command rejection are invalid checksum or parameter out of specified range.

Table 2-13 contains the input values for the following examples:

1. Debug On

\$PSRF105,1\*3E

2. Debug Off

\$PSRF105,0\*3F

Table 2-13 Development Data On/Off Data Format

Nam	Example	Units	Descriptio
Message ID	\$PSRF105		PSRF105 protocol header
Debug	1		0=Off, 1=On
Checksum	*3E		
<CR> <LF>			End of message termination

## 106—Select Datum

GPS receivers perform initial position and velocity calculations using an earth-centered earth-fixed (ECEF) coordinate system. Results may be converted to an earth model (geoid) defined by the selected datum. The default datum is WGS 84 (World Geodetic System 1984) which provides a worldwide common grid system that may be translated into local coordinate systems or map datums. (Local map datums are a best fit to the local shape of the earth and not valid worldwide.)

Table 2-14 contains the input values for the following examples:

1. Datum select TOKYO\_MEAN

\$PSRF106,178\*32

Table 2-14 Select Datum Data Format

Nam	Example	Units	Descriptio
Message ID	\$PSRF106		PSRF106 protocol header
Datum	178		21=WGS84 178=TOKYO_MEAN 179=TOKYO_JAPAN 180=TOKYO_KOREA 181=TOKYO_OKINAWA
Checksum	*32		
<CR> <LF>			End of message termination

## MSK—MSK Receiver Interface

Table 2-15 contains the values for the following example:

```
$GPMSK,318.0,A,10
0,M,2,*45
```

Table 2-15 RMC Data Format

Nam	Example	Units	Descriptio
Message ID	\$GPMSK		MSK protocol header
Beacon Frequency	318.0	kHz	Frequency to use
Auto/Manual Frequency <sup>1</sup>	A		A : Auto, M : Manual
Beacon Bit Rate	100		Bits per second
Auto/Manual Bit Rate <sup>1</sup>	M		A : Auto, M : Manual
Interval for Sending \$-- <sub>2</sub>	2	sec	Sending of MSS messages for

1. If Auto is specified the previous field value is ignored.

2. When status data is not to be transmitted this field is null.

---

**Note** – The NMEA messages supported by the Evaluation Receiver does not provide the ability to change the DGPS source. If you need to change the DGPS source to internal beacon, use the SiRF binary protocol and then switch to NMEA.

---

**ANEXO II**



### 9.3.1 REGISTERS

The MSSP module has four registers for SPI mode operation. These are:

- MSSP Control Register (SSPCON)
- MSSP Status Register (SSPSTAT)
- Serial Receive/Transmit Buffer Register (SSPBUF)
- MSSP Shift Register (SSPSR) – Not directly accessible

SSPCON and SSPSTAT are the control and status registers in SPI mode operation. The SSPCON register is readable and writable. The lower six bits of the SSPSTAT are read-only. The upper two bits of the SSPSTAT are read/write.

SSPSR is the shift register used for shifting data in or out. SSPBUF is the buffer register to which data bytes are written to or read from.

In receive operations, SSPSR and SSPBUF together create a double-buffered receiver. When SSPSR receives a complete byte, it is transferred to SSPBUF and the SSPIF interrupt is set.

During transmission, the SSPBUF is not double-buffered. A write to SSPBUF will write to both SSPBUF and SSPSR.

#### REGISTER 9-1: SSPSTAT: MSSP STATUS REGISTER (SPI MODE) (ADDRESS 94h)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/A	P	S	R/W	UA	BF
bit 7						bit 0	

- bit 7 **SMP:** Sample bit  
SPI Master mode:  
 1 = Input data sampled at end of data output time  
 0 = Input data sampled at middle of data output time  
SPI Slave mode:  
 SMP must be cleared when SPI is used in Slave mode.
- bit 6 **CKE:** SPI Clock Select bit  
 1 = Transmit occurs on transition from active to Idle clock state  
 0 = Transmit occurs on transition from Idle to active clock state  
**Note:** Polarity of clock state is set by the CKP bit (SSPCON1<4>).
- bit 5 **D/A:** Data/Address bit  
 Used in I<sup>2</sup>C mode only.
- bit 4 **P:** Stop bit  
 Used in I<sup>2</sup>C mode only. This bit is cleared when the MSSP module is disabled, SSPEN is cleared.
- bit 3 **S:** Start bit  
 Used in I<sup>2</sup>C mode only.
- bit 2 **R/W:** Read/Write bit information  
 Used in I<sup>2</sup>C mode only.
- bit 1 **UA:** Update Address bit  
 Used in I<sup>2</sup>C mode only.
- bit 0 **BF:** Buffer Full Status bit (Receive mode only)  
 1 = Receive complete, SSPBUF is full  
 0 = Receive not complete, SSPBUF is empty

#### Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared    x = Bit is unknown



**REGISTER 9-2: SSPCON1: MSSP CONTROL REGISTER 1 (SPI MODE) (ADDRESS 14h)**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0

bit 7

bit 0

bit 7 **WCOL:** Write Collision Detect bit (Transmit mode only)

1 = The SSPBUF register is written while it is still transmitting the previous word. (Must be cleared in software.)  
0 = No collision

bit 6 **SSPOV:** Receive Overflow Indicator bit

SPI Slave mode:

1 = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in Slave mode. The user must read the SSPBUF, even if only transmitting data, to avoid setting overflow. (Must be cleared in software.)  
0 = No overflow

**Note:** In Master mode, the overflow bit is not set, since each new reception (and transmission) is initiated by writing to the SSPBUF register.

bit 5 **SSPEN:** Synchronous Serial Port Enable bit

1 = Enables serial port and configures SCK, SDO, SDI, and  $\overline{SS}$  as serial port pins  
0 = Disables serial port and configures these pins as I/O port pins

**Note:** When enabled, these pins must be properly configured as input or output.

bit 4 **CKP:** Clock Polarity Select bit

1 = Idle state for clock is a high level  
0 = Idle state for clock is a low level

bit 3-0 **SSPM3:SSPM0:** Synchronous Serial Port Mode Select bits

0101 = SPI Slave mode, clock = SCK pin.  $\overline{SS}$  pin control disabled.  $\overline{SS}$  can be used as I/O pin.  
0100 = SPI Slave mode, clock = SCK pin.  $\overline{SS}$  pin control enabled.  
0011 = SPI Master mode, clock = TMR2 output/2  
0010 = SPI Master mode, clock = Fosc/64  
0001 = SPI Master mode, clock = Fosc/16  
0000 = SPI Master mode, clock = Fosc/4

**Note:** Bit combinations not specifically listed here are either reserved or implemented in I<sup>2</sup>C mode only.

**Legend:**

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared    x = Bit is unknown

### 9.3.2 OPERATION

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits (SSPCON<5:0> and SSPSTAT<7:6>). These control bits allow the following to be specified:

- Master mode (SCK is the clock output)
- Slave mode (SCK is the clock input)
- Clock Polarity (Idle state of SCK)
- Data Input Sample Phase (middle or end of data output time)
- Clock Edge (output data on rising/falling edge of SCK)
- Clock Rate (Master mode only)
- Slave Select mode (Slave mode only)

The MSSP consists of a transmit/receive shift register (SSPSR) and a buffer register (SSPBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPBUF holds the data that was written to the SSPSR until the received data is ready. Once the eight bits of data have been received, that byte is moved to the SSPBUF register. Then, the Buffer Full detect bit, BF (SSPSTAT<0>), and the interrupt flag bit, SSPIF, are set. This double-buffering of the received data (SSPBUF) allows the next byte to start reception before reading the data that was just received. Any write to the

SSPBUF register during transmission/reception of data will be ignored and the write collision detect bit, WCOL (SSPCON<7>), will be set. User software must clear the WCOL bit so that it can be determined if the following write(s) to the SSPBUF register completed successfully.

When the application software is expecting to receive valid data, the SSPBUF should be read before the next byte of data to transfer is written to the SSPBUF. Buffer Full bit, BF (SSPSTAT<0>), indicates when SSPBUF has been loaded with the received data (transmission is complete). When the SSPBUF is read, the BF bit is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally, the MSSP interrupt is used to determine when the transmission/reception has completed. The SSPBUF must be read and/or written. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur. Example 9-1 shows the loading of the SSPBUF (SSPSR) for data transmission.

The SSPSR is not directly readable or writable and can only be accessed by addressing the SSPBUF register. Additionally, the MSSP Status register (SSPSTAT) indicates the various status conditions.

#### EXAMPLE 9-1: LOADING THE SSPBUF (SSPSR) REGISTER

LOOP	BTFSS	SSPSTAT, BF	;Has data been received(transmit complete)?
	BRA	LOOP	;No
	MOVF	SSPBUF, W	;WREG reg = contents of SSPBUF
	MOVWF	RXDATA	;Save in user RAM, if data is meaningful
	MOVF	TXDATA, W	;W reg = contents of TXDATA
	MOVWF	SSPBUF	;New data to xmit

### 9.3.3 ENABLING SPI I/O

To enable the serial port, SSP Enable bit, SSPEN (SSPCON<5>), must be set. To reset or reconfigure SPI mode, clear the SSPEN bit, re-initialize the SSPCON registers and then set the SSPEN bit. This configures the SDI, SDO, SCK and  $\overline{SS}$  pins as serial port pins. For the pins to behave as the serial port function, some must have their data direction bits (in the TRIS register) appropriately programmed. That is:

- SDI is automatically controlled by the SPI module
- SDO must have TRISC<5> bit cleared
- SCK (Master mode) must have TRISC<3> bit cleared
- SCK (Slave mode) must have TRISC<3> bit set
- $\overline{SS}$  must have TRISC<4> bit set

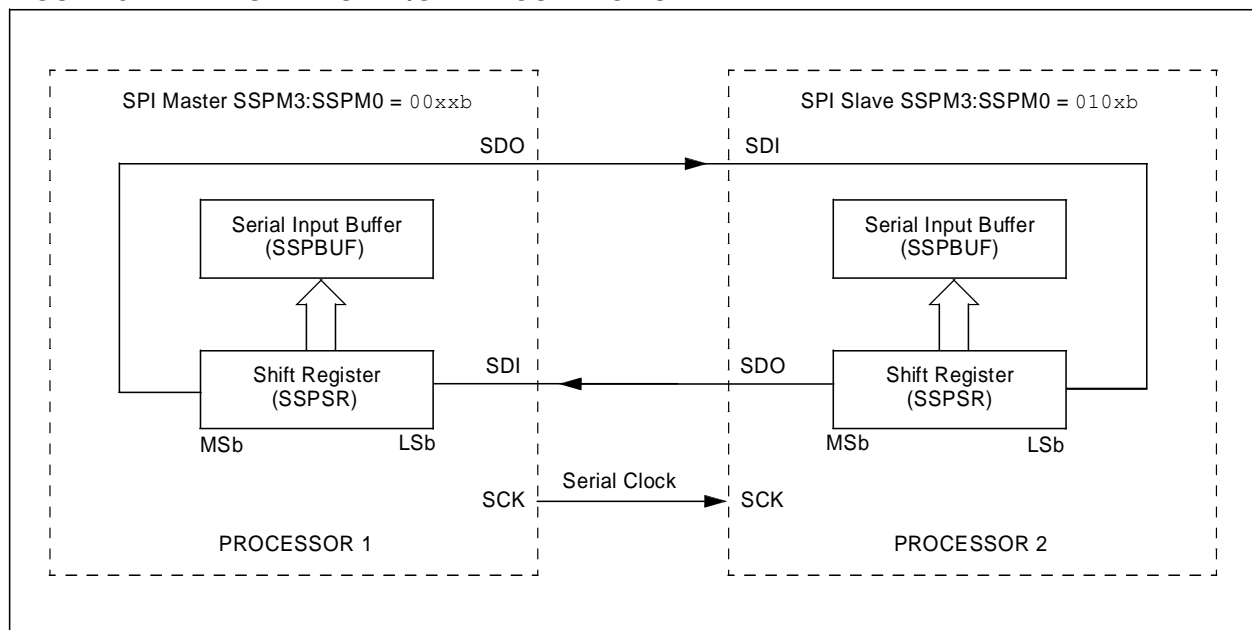
Any serial port function that is not desired may be overridden by programming the corresponding data direction (TRIS) register to the opposite value.

### 9.3.4 TYPICAL CONNECTION

Figure 9-2 shows a typical connection between two microcontrollers. The master controller (Processor 1) initiates the data transfer by sending the SCK signal. Data is shifted out of both shift registers on their programmed clock edge and latched on the opposite edge of the clock. Both processors should be programmed to the same Clock Polarity (CKP), then both controllers would send and receive data at the same time. Whether the data is meaningful (or dummy data) depends on the application software. This leads to three scenarios for data transmission:

- Master sends data – Slave sends dummy data
- Master sends data – Slave sends data
- Master sends dummy data – Slave sends data

**FIGURE 9-2: SPI MASTER/SLAVE CONNECTION**



9.3.5 MASTER MODE

The master can initiate the data transfer at any time because it controls the SCK. The master determines when the slave (Processor 2, Figure 9-2) is to broadcast data by the software protocol.

In Master mode, the data is transmitted/received as soon as the SSPBUF register is written to. If the SPI is only going to receive, the SDO output could be disabled (programmed as an input). The SSPSR register will continue to shift in the signal present on the SDI pin at the programmed clock rate. As each byte is received, it will be loaded into the SSPBUF register as if a normal received byte (interrupts and status bits appropriately set). This could be useful in receiver applications as a "Line Activity Monitor" mode.

The clock polarity is selected by appropriately programming the CKP bit (SSPCON<4>). This then, would give waveforms for SPI communication as shown in

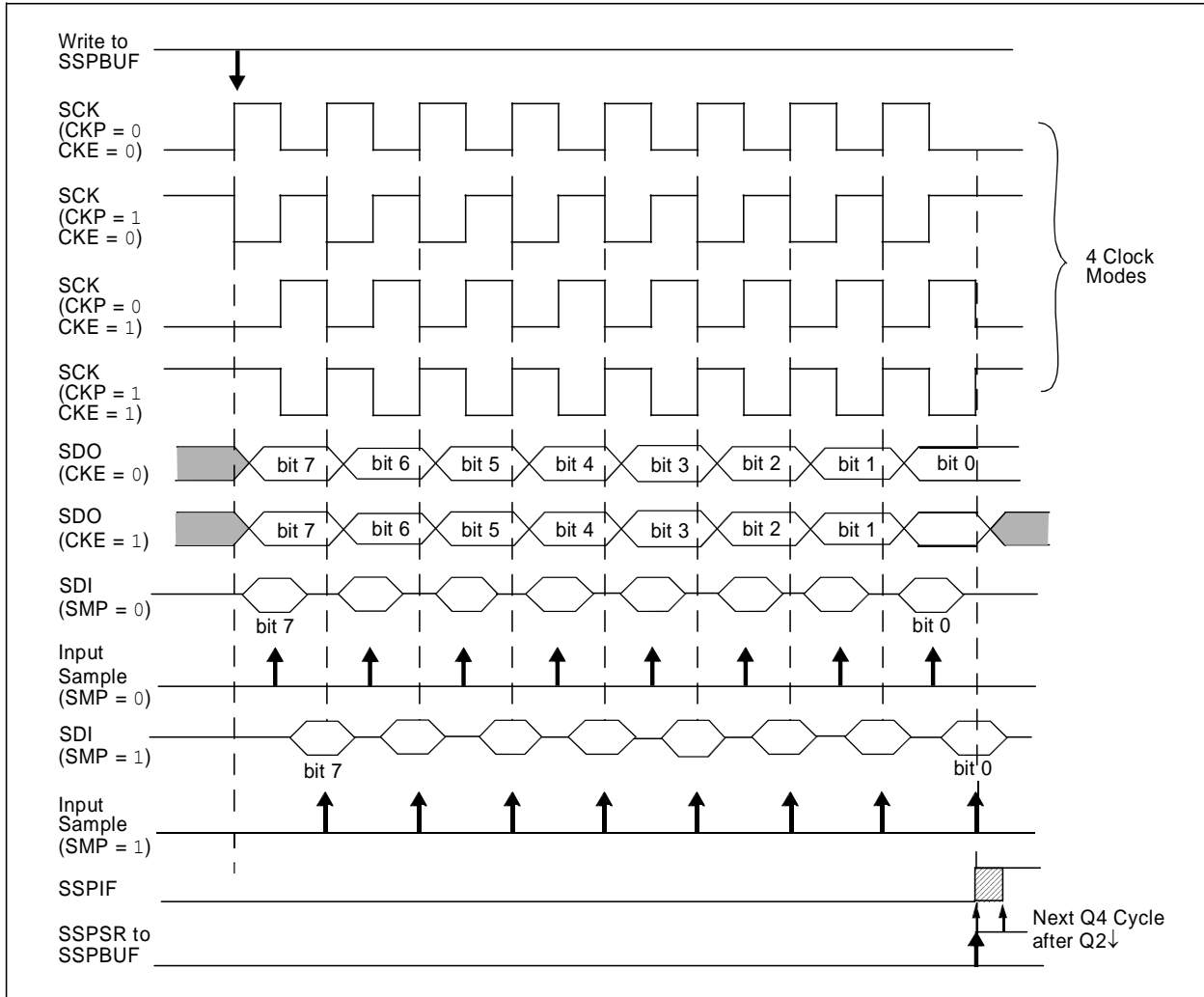
Figure 9-3, Figure 9-5 and Figure 9-6, where the MSB is transmitted first. In Master mode, the SPI clock rate (bit rate) is user programmable to be one of the following:

- Fosc/4 (or Tcy)
- Fosc/16 (or 4 • Tcy)
- Fosc/64 (or 16 • Tcy)
- Timer2 output/2

This allows a maximum data rate (at 40 MHz) of 10.00 Mbps.

Figure 9-3 shows the waveforms for Master mode. When the CKE bit is set, the SDO data is valid before there is a clock edge on SCK. The change of the input sample is shown based on the state of the SMP bit. The time when the SSPBUF is loaded with the received data is shown.

FIGURE 9-3: SPI MODE WAVEFORM (MASTER MODE)



### 9.3.6 SLAVE MODE

In Slave mode, the data is transmitted and received as the external clock pulses appear on SCK. When the last bit is latched, the SSPIF interrupt flag bit is set.

While in Slave mode, the external clock is supplied by the external clock source on the SCK pin. This external clock must meet the minimum high and low times as specified in the electrical specifications.

While in Sleep mode, the slave can transmit/receive data. When a byte is received, the device will wake-up from Sleep.

### 9.3.7 SLAVE SELECT SYNCHRONIZATION

The  $\overline{SS}$  pin allows a Synchronous Slave mode. The SPI must be in Slave mode with  $\overline{SS}$  pin control enabled ( $SSPCON<3:0> = 04h$ ). The pin must not be driven low for the  $\overline{SS}$  pin to function as an input. The data latch must be high. When the  $\overline{SS}$  pin is low, transmission and reception are enabled and the SDO pin is driven. When

the  $\overline{SS}$  pin goes high, the SDO pin is no longer driven even if in the middle of a transmitted byte and becomes a floating output. External pull-up/pull-down resistors may be desirable, depending on the application.

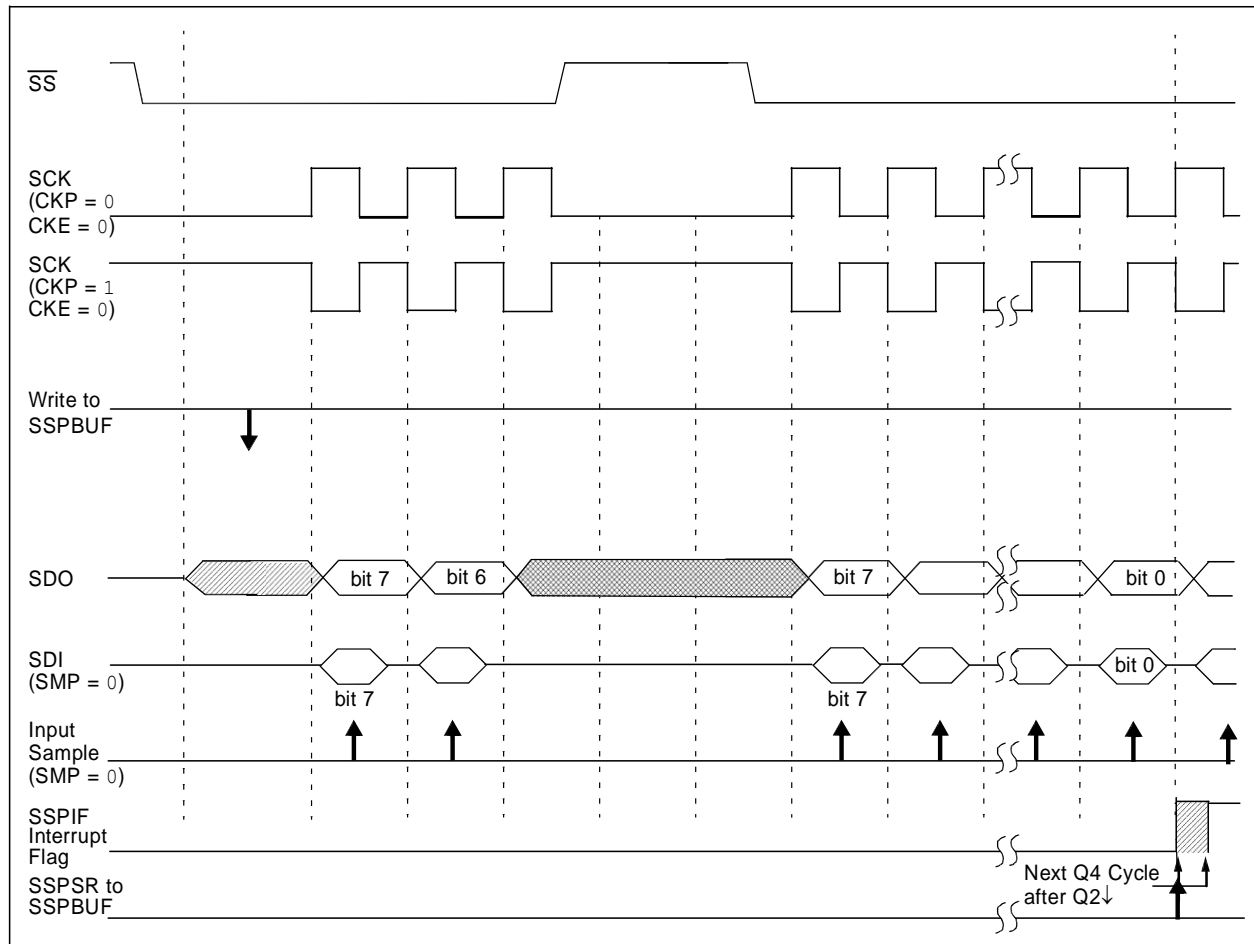
**Note 1:** When the SPI is in Slave mode with  $\overline{SS}$  pin control enabled ( $SSPCON<3:0> = 0100$ ), the SPI module will reset if the  $\overline{SS}$  pin is set to  $V_{DD}$ .

**Note 2:** If the SPI is used in Slave Mode with CKE set, then the  $\overline{SS}$  pin control must be enabled.

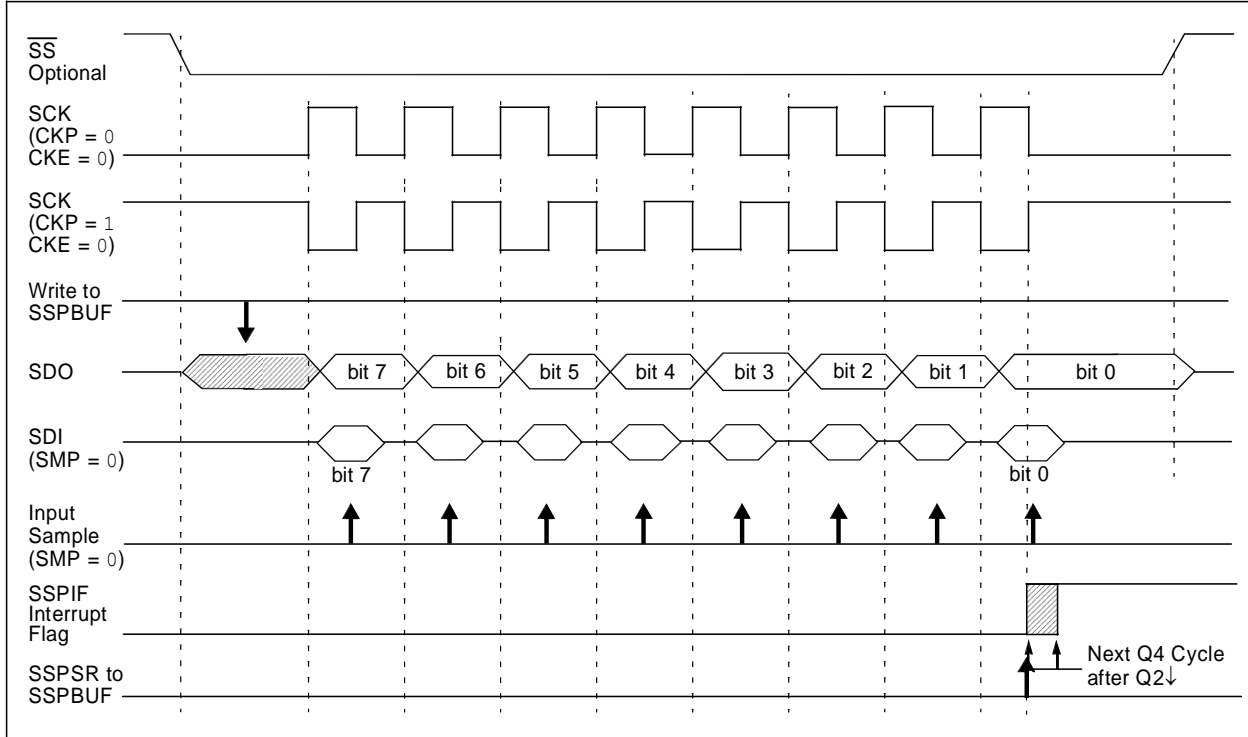
When the SPI module resets, the bit counter is forced to '0'. This can be done by either forcing the  $\overline{SS}$  pin to a high level or clearing the SSPEN bit.

To emulate two-wire communication, the SDO pin can be connected to the SDI pin. When the SPI needs to operate as a receiver, the SDO pin can be configured as an input. This disables transmissions from the SDO. The SDI can always be left as an input (SDI function) since it cannot create a bus conflict.

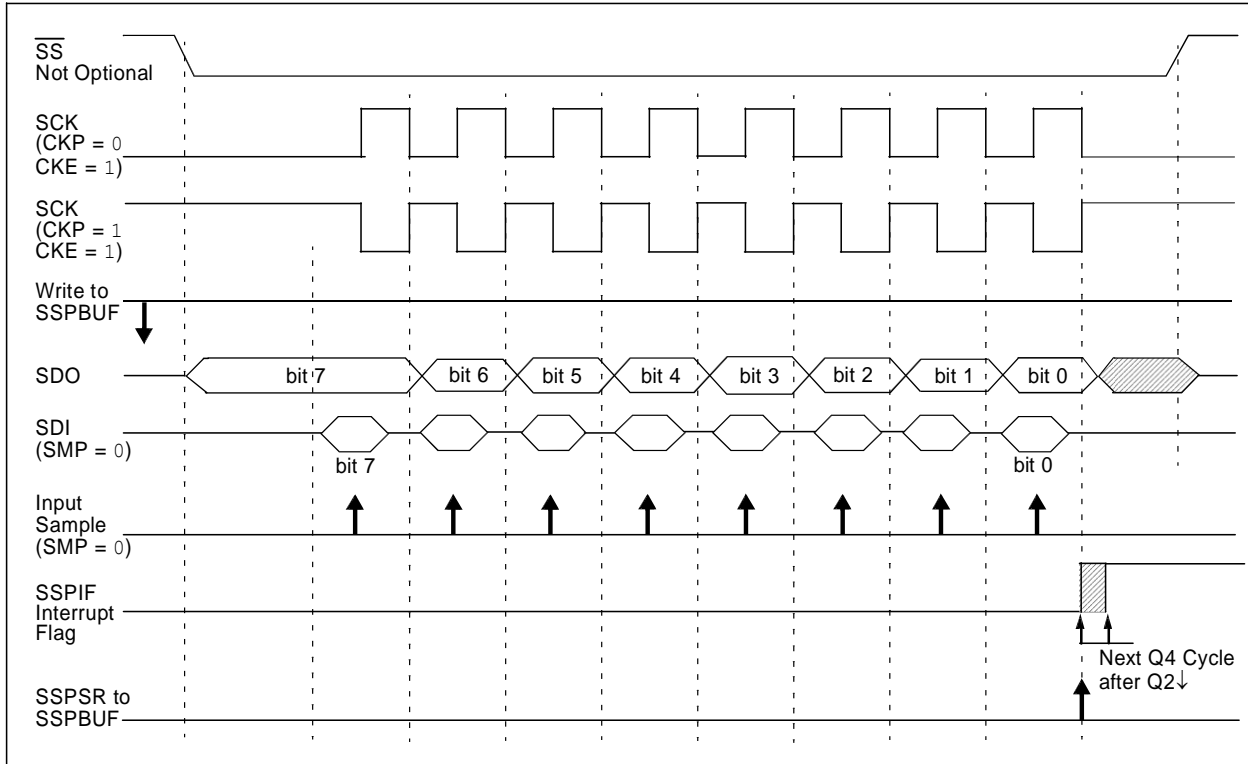
**FIGURE 9-4: SLAVE SYNCHRONIZATION WAVEFORM**



**FIGURE 9-5: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 0)**



**FIGURE 9-6: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 1)**



### 9.3.8 SLEEP OPERATION

In Master mode, all module clocks are halted and the transmission/reception will remain in that state until the device wakes from Sleep. After the device returns to normal mode, the module will continue to transmit/receive data.

In Slave mode, the SPI Transmit/Receive Shift register operates asynchronously to the device. This allows the device to be placed in Sleep mode and data to be shifted into the SPI Transmit/Receive Shift register. When all 8 bits have been received, the MSSP interrupt flag bit will be set and if enabled, will wake the device from Sleep.

### 9.3.9 EFFECTS OF A RESET

A Reset disables the MSSP module and terminates the current transfer.

### 9.3.10 BUS MODE COMPATIBILITY

Table 9-1 shows the compatibility between the standard SPI modes and the states of the CKP and CKE control bits.

**TABLE 9-1: SPI BUS MODES**

Standard SPI Mode Terminology	Control Bits State	
	CKP	CKE
0, 0	0	1
0, 1	0	0
1, 0	1	1
1, 1	1	0

There is also a SMP bit which controls when the data is sampled.

**TABLE 9-2: REGISTERS ASSOCIATED WITH SPI OPERATION**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets	
INTCON	GIE/ GIEH	PEIE/ GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u	
PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000	
PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000	
TRISC	PORTC Data Direction Register								1111 1111	1111 1111	
SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu	
SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000	
TRISA	—	PORTA Data Direction Register								--11 1111	--11 1111
SSPSTAT	SMP	CKE	D/A	P	S	R/W	UA	BF	0000 0000	0000 0000	

**Legend:** x = unknown, u = unchanged, - = unimplemented, read as '0'.  
Shaded cells are not used by the MSSP in SPI mode.

**Note 1:** The PSPIF, PSPIE and PSPIP bits are reserved on 28-pin devices; always maintain these bits clear.



## 10.0 ADDRESSABLE UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (USART)

The Universal Synchronous Asynchronous Receiver Transmitter (USART) module is one of the two serial I/O modules. (USART is also known as a Serial Communications Interface or SCI.) The USART can be configured as a full-duplex asynchronous system that can communicate with peripheral devices, such as CRT terminals and personal computers, or it can be configured as a half-duplex synchronous system that can communicate with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROMs, etc.

The USART can be configured in the following modes:

- Asynchronous (full-duplex)
- Synchronous – Master (half-duplex)
- Synchronous – Slave (half-duplex)

Bit SPEN (RCSTA<7>) and bits TRISC<7:6> have to be set in order to configure pins RC6/TX/CK and RC7/RX/DT as the Universal Synchronous Asynchronous Receiver Transmitter.

The USART module also has a multi-processor communication capability using 9-bit address detection.

### REGISTER 10-1: TXSTA: TRANSMIT STATUS AND CONTROL REGISTER (ADDRESS 98h)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D
						bit 7	bit 0

- bit 7 **CSRC:** Clock Source Select bit  
Asynchronous mode:  
 Don't care.  
Synchronous mode:  
 1 = Master mode (clock generated internally from BRG)  
 0 = Slave mode (clock from external source)
- bit 6 **TX9:** 9-bit Transmit Enable bit  
 1 = Selects 9-bit transmission  
 0 = Selects 8-bit transmission
- bit 5 **TXEN:** Transmit Enable bit  
 1 = Transmit enabled  
 0 = Transmit disabled  
**Note:** SREN/CREN overrides TXEN in Sync mode.
- bit 4 **SYNC:** USART Mode Select bit  
 1 = Synchronous mode  
 0 = Asynchronous mode
- bit 3 **Unimplemented:** Read as '0'
- bit 2 **BRGH:** High Baud Rate Select bit  
Asynchronous mode:  
 1 = High speed  
 0 = Low speed  
Synchronous mode:  
 Unused in this mode.
- bit 1 **TRMT:** Transmit Shift Register Status bit  
 1 = TSR empty  
 0 = TSR full
- bit 0 **TX9D:** 9th bit of Transmit Data, can be Parity bit

#### Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared    x = Bit is unknown

**REGISTER 10-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER (ADDRESS 18h)**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D

bit 7

bit 0

- bit 7     **SPEN:** Serial Port Enable bit  
1 = Serial port enabled (configures RC7/RX/DT and RC6/TX/CK pins as serial port pins)  
0 = Serial port disabled
- bit 6     **RX9:** 9-bit Receive Enable bit  
1 = Selects 9-bit reception  
0 = Selects 8-bit reception
- bit 5     **SREN:** Single Receive Enable bit  
Asynchronous mode:  
Don't care.  
Synchronous mode – Master:  
1 = Enables single receive  
0 = Disables single receive  
This bit is cleared after reception is complete.  
Synchronous mode – Slave:  
Don't care.
- bit 4     **CREN:** Continuous Receive Enable bit  
Asynchronous mode:  
1 = Enables continuous receive  
0 = Disables continuous receive  
Synchronous mode:  
1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)  
0 = Disables continuous receive
- bit 3     **ADDEN:** Address Detect Enable bit  
Asynchronous mode 9-bit (RX9 = 1):  
1 = Enables address detection, enables interrupt and load of the receive buffer when RSR<8> is set  
0 = Disables address detection, all bytes are received and ninth bit can be used as parity bit
- bit 2     **FERR:** Framing Error bit  
1 = Framing error (can be updated by reading RCREG register and receive next valid byte)  
0 = No framing error
- bit 1     **OERR:** Overrun Error bit  
1 = Overrun error (can be cleared by clearing bit CREN)  
0 = No overrun error
- bit 0     **RX9D:** 9th bit of Received Data (can be parity bit but must be calculated by user firmware)

**Legend:**

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared     x = Bit is unknown

## 10.1 USART Baud Rate Generator (BRG)

The BRG supports both the Asynchronous and Synchronous modes of the USART. It is a dedicated 8-bit baud rate generator. The SPBRG register controls the period of a free running 8-bit timer. In Asynchronous mode, bit BRGH (TXSTA<2>) also controls the baud rate. In Synchronous mode, bit BRGH is ignored. Table 10-1 shows the formula for computation of the baud rate for different USART modes which only apply in Master mode (internal clock).

Given the desired baud rate and  $F_{osc}$ , the nearest integer value for the SPBRG register can be calculated using the formula in Table 10-1. From this, the error in baud rate can be determined.

It may be advantageous to use the high baud rate (BRGH = 1) even for slower baud clocks. This is because the  $F_{osc}/(16(X+1))$  equation can reduce the baud rate error in some cases.

Writing a new value to the SPBRG register causes the BRG timer to be reset (or cleared). This ensures the BRG does not wait for a timer overflow before outputting the new baud rate.

### 10.1.1 SAMPLING

The data on the RC7/RX/DT pin is sampled three times by a majority detect circuit to determine if a high or a low level is present at the RX pin.

**TABLE 10-1: BAUD RATE FORMULA**

SYNC	BRGH = 0 (Low Speed)	BRGH = 1 (High Speed)
0	(Asynchronous) Baud Rate = $F_{osc}/(64(X+1))$	Baud Rate = $F_{osc}/(16(X+1))$
1	(Synchronous) Baud Rate = $F_{osc}/(4(X+1))$	N/A

**Legend:** X = value in SPBRG (0 to 255)

**TABLE 10-2: REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

**Legend:** x = unknown, - = unimplemented, read as '0'. Shaded cells are not used by the BRG.

**TABLE 10-3: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 0)**

BAUD RATE (K)	Fosc = 20 MHz			Fosc = 16 MHz			Fosc = 10 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-	-	-	-
1.2	1.221	1.75	255	1.202	0.17	207	1.202	0.17	129
2.4	2.404	0.17	129	2.404	0.17	103	2.404	0.17	64
9.6	9.766	1.73	31	9.615	0.16	25	9.766	1.73	15
19.2	19.531	1.72	15	19.231	0.16	12	19.531	1.72	7
28.8	31.250	8.51	9	27.778	3.55	8	31.250	8.51	4
33.6	34.722	3.34	8	35.714	6.29	6	31.250	6.99	4
57.6	62.500	8.51	4	62.500	8.51	3	52.083	9.58	2
HIGH	1.221	-	255	0.977	-	255	0.610	-	255
LOW	312.500	-	0	250.000	-	0	156.250	-	0

BAUD RATE (K)	Fosc = 4 MHz			Fosc = 3.6864 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	0.300	0	207	0.3	0	191
1.2	1.202	0.17	51	1.2	0	47
2.4	2.404	0.17	25	2.4	0	23
9.6	8.929	6.99	6	9.6	0	5
19.2	20.833	8.51	2	19.2	0	2
28.8	31.250	8.51	1	28.8	0	1
33.6	-	-	-	-	-	-
57.6	62.500	8.51	0	57.6	0	0
HIGH	0.244	-	255	0.225	-	255
LOW	62.500	-	0	57.6	-	0

**TABLE 10-4: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 1)**

BAUD RATE (K)	Fosc = 20 MHz			Fosc = 16 MHz			Fosc = 10 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-	-	-	-
1.2	-	-	-	-	-	-	-	-	-
2.4	-	-	-	-	-	-	2.441	1.71	255
9.6	9.615	0.16	129	9.615	0.16	103	9.615	0.16	64
19.2	19.231	0.16	64	19.231	0.16	51	19.531	1.72	31
28.8	29.070	0.94	42	29.412	2.13	33	28.409	1.36	21
33.6	33.784	0.55	36	33.333	0.79	29	32.895	2.10	18
57.6	59.524	3.34	20	58.824	2.13	16	56.818	1.36	10
HIGH	4.883	-	255	3.906	-	255	2.441	-	255
LOW	1250.000	-	0	1000.000	-	0	625.000	-	0

BAUD RATE (K)	Fosc = 4 MHz			Fosc = 3.6864 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-
1.2	1.202	0.17	207	1.2	0	191
2.4	2.404	0.17	103	2.4	0	95
9.6	9.615	0.16	25	9.6	0	23
19.2	19.231	0.16	12	19.2	0	11
28.8	27.798	3.55	8	28.8	0	7
33.6	35.714	6.29	6	32.9	2.04	6
57.6	62.500	8.51	3	57.6	0	3
HIGH	0.977	-	255	0.9	-	255
LOW	250.000	-	0	230.4	-	0

## 10.2 USART Asynchronous Mode

In this mode, the USART uses standard Non-Return-to-Zero (NRZ) format (one Start bit, eight or nine data bits and one Stop bit). The most common data format is 8 bits. An on-chip, dedicated, 8-bit Baud Rate Generator can be used to derive standard baud rate frequencies from the oscillator. The USART transmits and receives the LSb first. The transmitter and receiver are functionally independent but use the same data format and baud rate. The baud rate generator produces a clock, either x16 or x64 of the bit shift rate, depending on bit BRGH (TXSTA<2>). Parity is not supported by the hardware but can be implemented in software (and stored as the ninth data bit). Asynchronous mode is stopped during Sleep.

Asynchronous mode is selected by clearing bit SYNC (TXSTA<4>).

The USART Asynchronous module consists of the following important elements:

- Baud Rate Generator
- Sampling Circuit
- Asynchronous Transmitter
- Asynchronous Receiver

### 10.2.1 USART ASYNCHRONOUS TRANSMITTER

The USART transmitter block diagram is shown in Figure 10-1. The heart of the transmitter is the Transmit (Serial) Shift Register (TSR). The shift register obtains its data from the Read/Write Transmit Buffer, TXREG. The TXREG register is loaded with data in software. The TSR register is not loaded until the Stop bit has been transmitted from the previous load. As soon as the Stop bit is transmitted, the TSR is loaded with new data from the TXREG register (if available). Once the TXREG register transfers the data to the TSR register (occurs in one Tcy), the TXREG register is empty and flag bit, TXIF (PIR1<4>), is set. This interrupt can be

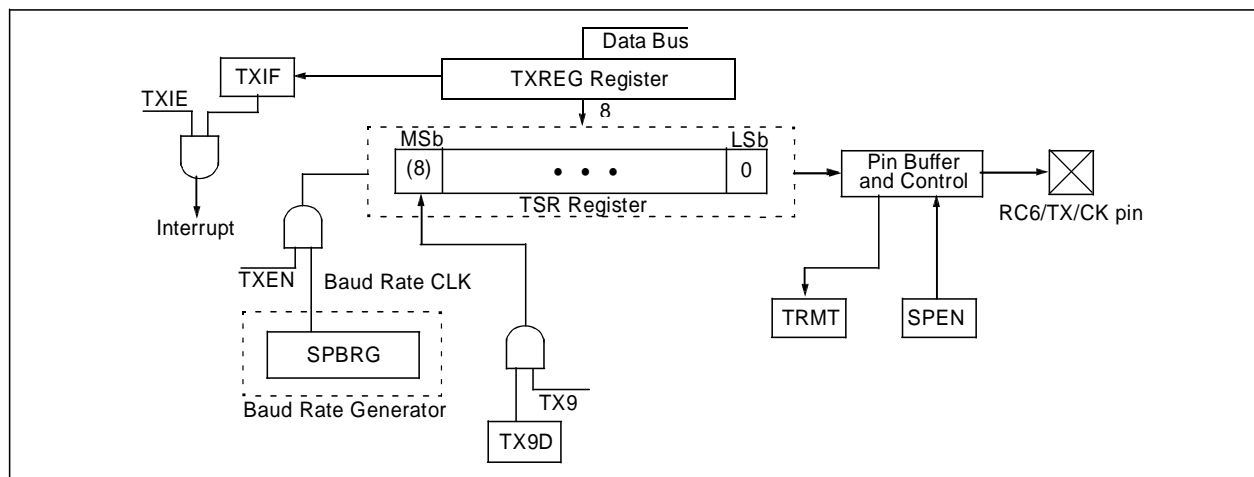
enabled/disabled by setting/clearing enable bit, TXIE (PIE1<4>). Flag bit TXIF will be set regardless of the state of enable bit TXIE and cannot be cleared in software. It will reset only when new data is loaded into the TXREG register. While flag bit TXIF indicates the status of the TXREG register, another bit, TRMT (TXSTA<1>), shows the status of the TSR register. Status bit TRMT is a read-only bit which is set when the TSR register is empty. No interrupt logic is tied to this bit so the user has to poll this bit in order to determine if the TSR register is empty.

**Note 1:** The TSR register is not mapped in data memory so it is not available to the user.  
**Note 2:** Flag bit TXIF is set when enable bit TXEN is set. TXIF is cleared by loading TXREG.

Transmission is enabled by setting enable bit, TXEN (TXSTA<5>). The actual transmission will not occur until the TXREG register has been loaded with data and the Baud Rate Generator (BRG) has produced a shift clock (Figure 10-2). The transmission can also be started by first loading the TXREG register and then setting enable bit TXEN. Normally, when transmission is first started, the TSR register is empty. At that point, transfer to the TXREG register will result in an immediate transfer to TSR, resulting in an empty TXREG. A back-to-back transfer is thus possible (Figure 10-3). Clearing enable bit TXEN during a transmission will cause the transmission to be aborted and will reset the transmitter. As a result, the RC6/TX/CK pin will revert to high-impedance.

In order to select 9-bit transmission, transmit bit TX9 (TXSTA<6>) should be set and the ninth bit should be written to TX9D (TXSTA<0>). The ninth bit must be written before writing the 8-bit data to the TXREG register. This is because a data write to the TXREG register can result in an immediate transfer of the data to the TSR register (if the TSR is empty). In such a case, an incorrect ninth data bit may be loaded in the TSR register.

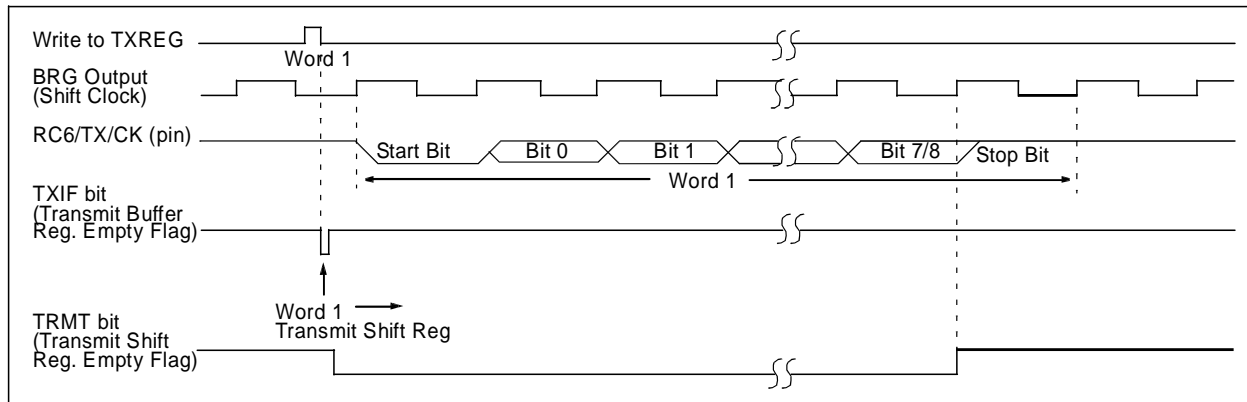
**FIGURE 10-1: USART TRANSMIT BLOCK DIAGRAM**



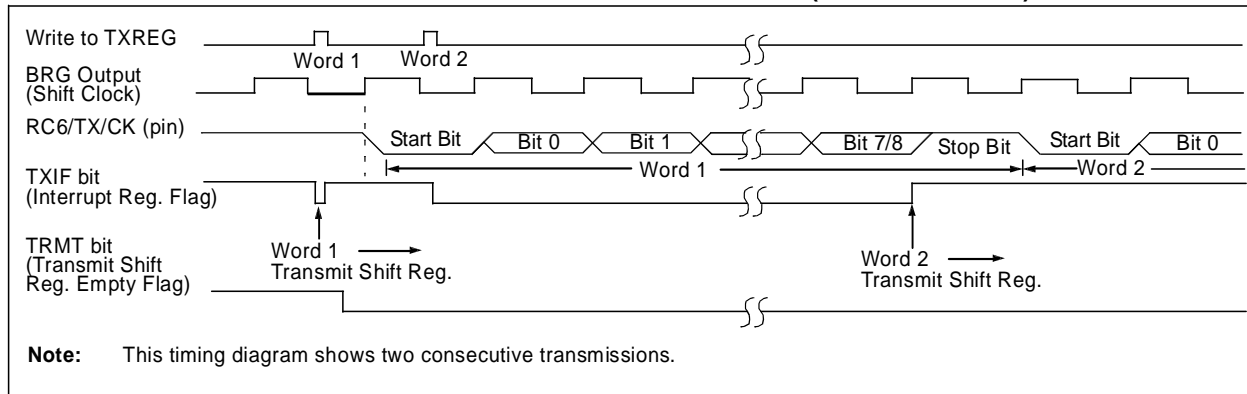
When setting up an Asynchronous Transmission, follow these steps:

1. Initialize the SPBRG register for the appropriate baud rate. If a high-speed baud rate is desired, set bit BRGH (Section 10.1 “USART Baud Rate Generator (BRG)”).
2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
3. If interrupts are desired, then set enable bit TXIE.
4. If 9-bit transmission is desired, then set transmit bit TX9.
5. Enable the transmission by setting bit TXEN, which will also set bit TXIF.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Load data to the TXREG register (starts transmission).
8. If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCON register are set.

**FIGURE 10-2: ASYNCHRONOUS MASTER TRANSMISSION**



**FIGURE 10-3: ASYNCHRONOUS MASTER TRANSMISSION (BACK TO BACK)**



**TABLE 10-5: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	R0IF	0000 000x	0000 000u
0Ch	PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
19h	TXREG	USART Transmit Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

**Legend:** x = unknown, - = unimplemented locations read as ‘0’. Shaded cells are not used for asynchronous transmission.

**Note 1:** Bits PSPIE and PSPIF are reserved on 28-pin devices; always maintain these bits clear.

### 10.2.2 USART ASYNCHRONOUS RECEIVER

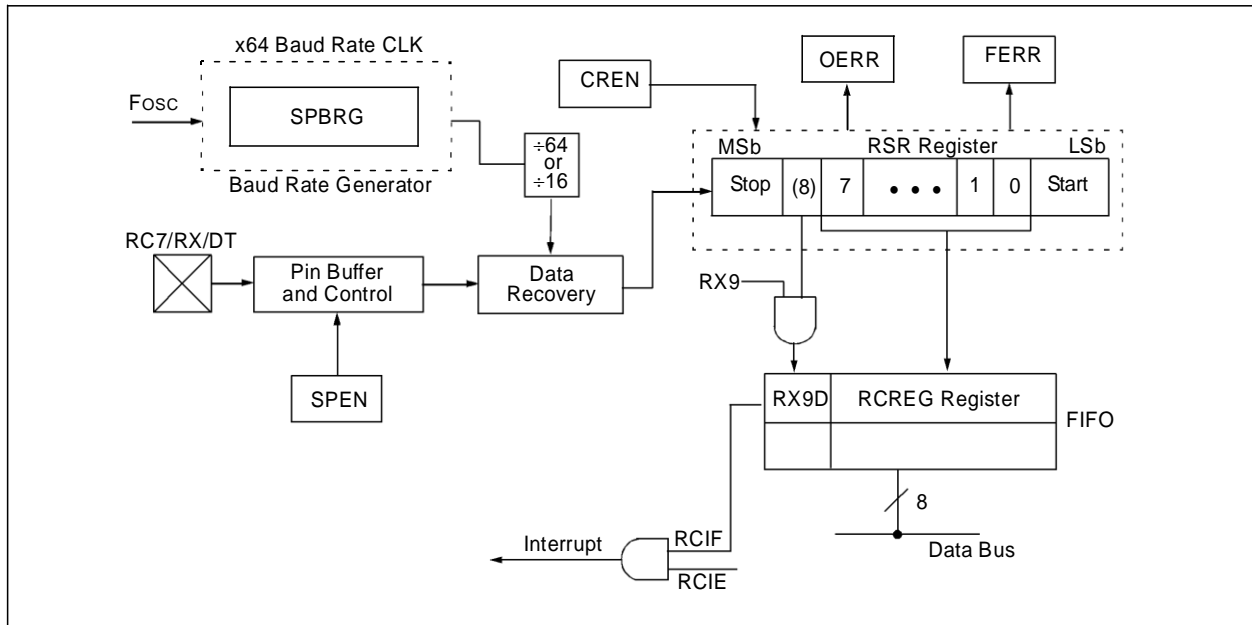
The receiver block diagram is shown in Figure 10-4. The data is received on the RC7/RX/DT pin and drives the data recovery block. The data recovery block is actually a high-speed shifter, operating at x16 times the baud rate; whereas the main receive serial shifter operates at the bit rate or at Fosc.

Once Asynchronous mode is selected, reception is enabled by setting bit CREN (RCSTA<4>).

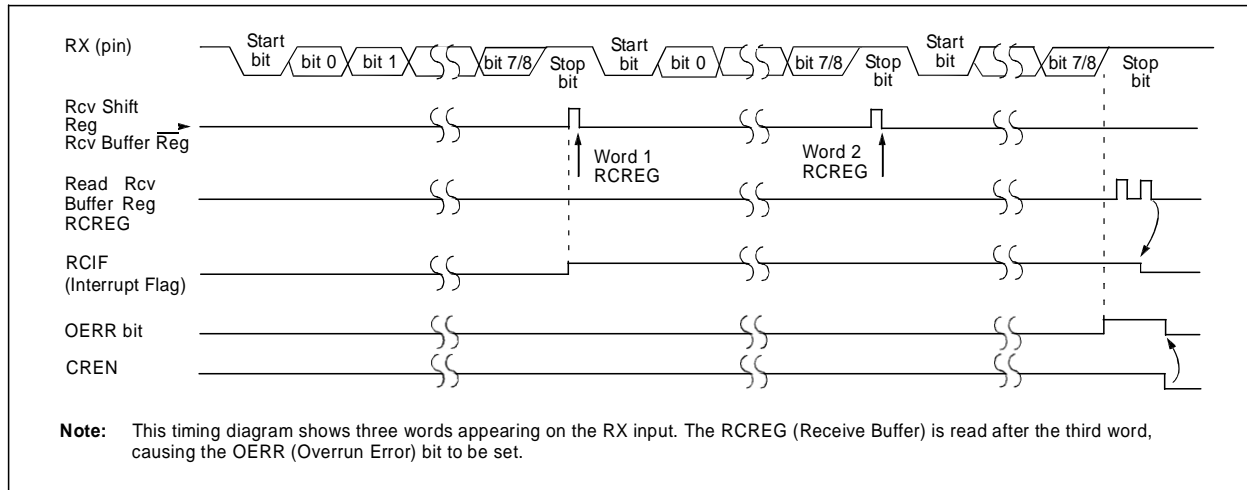
The heart of the receiver is the Receive (Serial) Shift Register (RSR). After sampling the Stop bit, the received data in the RSR is transferred to the RCREG register (if it is empty). If the transfer is complete, flag bit, RCIF (PIR1<5>), is set. The actual interrupt can be enabled/disabled by setting/clearing enable bit, RCIE (PIE1<5>). Flag bit RCIF is a read-only bit which is cleared by the hardware. It is cleared when the RCREG register has been read and is empty. The RCREG register is a double-buffered register (i.e., it is a two-deep FIFO). It

is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte to begin shifting to the RSR register. On the detection of the Stop bit of the third byte, if the RCREG register is still full, the Overrun Error bit, OERR (RCSTA<1>), will be set. The word in the RSR will be lost. The RCREG register can be read twice to retrieve the two bytes in the FIFO. Overrun bit OERR has to be cleared in software. This is done by resetting the receive logic (CREN is cleared and then set). If bit OERR is set, transfers from the RSR register to the RCREG register are inhibited and no further data will be received. It is, therefore, essential to clear error bit OERR if it is set. Framing error bit, FERR (RCSTA<2>), is set if a Stop bit is detected as clear. Bit FERR and the 9th receive bit are buffered the same way as the receive data. Reading the RCREG will load bits RX9D and FERR with new values, therefore, it is essential for the user to read the RCSTA register before reading the RCREG register in order not to lose the old FERR and RX9D information.

**FIGURE 10-4: USART RECEIVE BLOCK DIAGRAM**



**FIGURE 10-5: ASYNCHRONOUS RECEPTION**



When setting up an Asynchronous Reception, follow these steps:

1. Initialize the SPBRG register for the appropriate baud rate. If a high-speed baud rate is desired, set bit BRGH (Section 10.1 “USART Baud Rate Generator (BRG)”).
2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
3. If interrupts are desired, then set enable bit RCIE.
4. If 9-bit reception is desired, then set bit RX9.
5. Enable the reception by setting bit CREN.
6. Flag bit RCIF will be set when reception is complete and an interrupt will be generated if enable bit RCIE is set.
7. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
8. Read the 8-bit received data by reading the RCREG register.
9. If any error occurred, clear the error by clearing enable bit CREN.
10. If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCON register are set.

**TABLE 10-6: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	R0IF	0000 000x	0000 000u
0Ch	PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
1Ah	RCREG	USART Receive Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

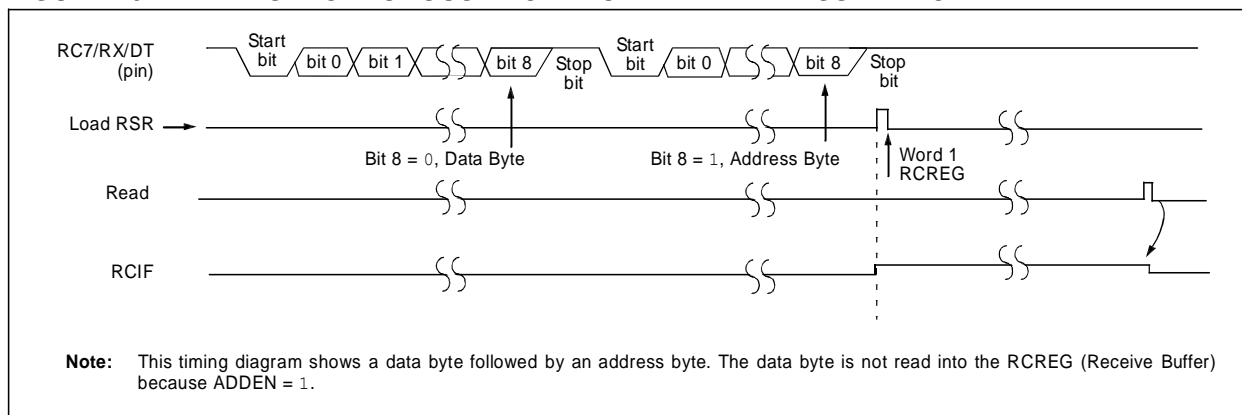
**Legend:** x = unknown, - = unimplemented locations read as '0'. Shaded cells are not used for asynchronous reception.

**Note 1:** Bits PSPIE and PSPIF are reserved on 28-pin devices; always maintain these bits clear.

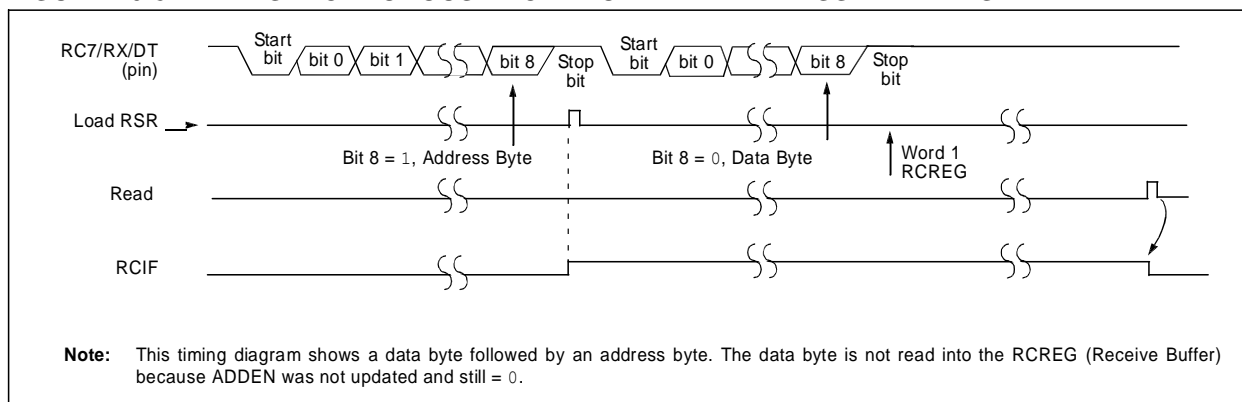




**FIGURE 10-7: ASYNCHRONOUS RECEPTION WITH ADDRESS DETECT**



**FIGURE 10-8: ASYNCHRONOUS RECEPTION WITH ADDRESS BYTE FIRST**



**TABLE 10-7: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	R0IF	0000 000x	0000 000u
0Ch	PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
1Ah	RCREG	USART Receive Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

**Legend:** x = unknown, - = unimplemented locations read as '0'. Shaded cells are not used for asynchronous reception.

**Note 1:** Bits PSPIE and PSPIF are reserved on 28-pin devices; always maintain these bits clear.

### 10.3 USART Synchronous Master Mode

In Synchronous Master mode, the data is transmitted in a half-duplex manner (i.e., transmission and reception do not occur at the same time). When transmitting data, the reception is inhibited and vice versa. Synchronous mode is entered by setting bit, SYNC (TXSTA<4>). In addition, enable bit, SPEN (RCSTA<7>), is set in order to configure the RC6/TX/CK and RC7/RX/DT I/O pins to CK (clock) and DT (data) lines, respectively. The Master mode indicates that the processor transmits the master clock on the CK line. The Master mode is entered by setting bit, CSRC (TXSTA<7>).

#### 10.3.1 USART SYNCHRONOUS MASTER TRANSMISSION

The USART transmitter block diagram is shown in Figure 10-6. The heart of the transmitter is the Transmit (Serial) Shift Register (TSR). The shift register obtains its data from the Read/Write Transmit Buffer register, TXREG. The TXREG register is loaded with data in software. The TSR register is not loaded until the last bit has been transmitted from the previous load. As soon as the last bit is transmitted, the TSR is loaded with new data from the TXREG (if available). Once the TXREG register transfers the data to the TSR register (occurs in one T<sub>CYCLE</sub>), the TXREG is empty and interrupt bit, TXIF (PIR1<4>), is set. The interrupt can be enabled/disabled by setting/clearing enable bit TXIE (PIE1<4>). Flag bit TXIF will be set regardless of the state of enable bit TXIE and cannot be cleared in software. It will reset only when new data is loaded into the TXREG register. While flag bit TXIF indicates the status of the TXREG register, another bit, TRMT (TXSTA<1>), shows the status of the TSR register. TRMT is a read-only bit which is set when the TSR is empty. No interrupt logic is tied to this bit so the user has to poll this bit in order to determine if the TSR register is empty. The TSR is not mapped in data memory so it is not available to the user.

Transmission is enabled by setting enable bit, TXEN (TXSTA<5>). The actual transmission will not occur until the TXREG register has been loaded with data. The first data bit will be shifted out on the next available rising edge of the clock on the CK line. Data out is stable around the falling edge of the synchronous clock (Figure 10-9). The transmission can also be started by first loading the TXREG register and then setting bit TXEN (Figure 10-10). This is advantageous when slow baud rates are selected since the BRG is kept in Reset when bits TXEN, CREN and SREN are clear. Setting enable bit TXEN will start the BRG, creating a shift clock immediately. Normally, when transmission is first started, the TSR register is empty so a transfer to the TXREG register will result in an immediate transfer to TSR, resulting in an empty TXREG. Back-to-back transfers are possible.

Clearing enable bit TXEN during a transmission will cause the transmission to be aborted and will reset the transmitter. The DT and CK pins will revert to high-impedance. If either bit CREN or bit SREN is set during a transmission, the transmission is aborted and the DT pin reverts to a high-impedance state (for a reception). The CK pin will remain an output if bit CSRC is set (internal clock). The transmitter logic, however, is not reset, although it is disconnected from the pins. In order to reset the transmitter, the user has to clear bit TXEN. If bit SREN is set (to interrupt an on-going transmission and receive a single word), then after the single word is received, bit SREN will be cleared and the serial port will revert back to transmitting since bit TXEN is still set. The DT line will immediately switch from High-Impedance Receive mode to transmit and start driving. To avoid this, bit TXEN should be cleared.

In order to select 9-bit transmission, the TX9 (TXSTA<6>) bit should be set and the ninth bit should be written to bit TX9D (TXSTA<0>). The ninth bit must be written before writing the 8-bit data to the TXREG register. This is because a data write to the TXREG can result in an immediate transfer of the data to the TSR register (if the TSR is empty). If the TSR was empty and the TXREG was written before writing the “new” TX9D, the “present” value of bit TX9D is loaded.

Steps to follow when setting up a Synchronous Master Transmission:

1. Initialize the SPBRG register for the appropriate baud rate (**Section 10.1 “USART Baud Rate Generator (BRG)”**).
2. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
3. If interrupts are desired, set enable bit TXIE.
4. If 9-bit transmission is desired, set bit TX9.
5. Enable the transmission by setting bit TXEN.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Start transmission by loading data to the TXREG register.
8. If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCON register are set.

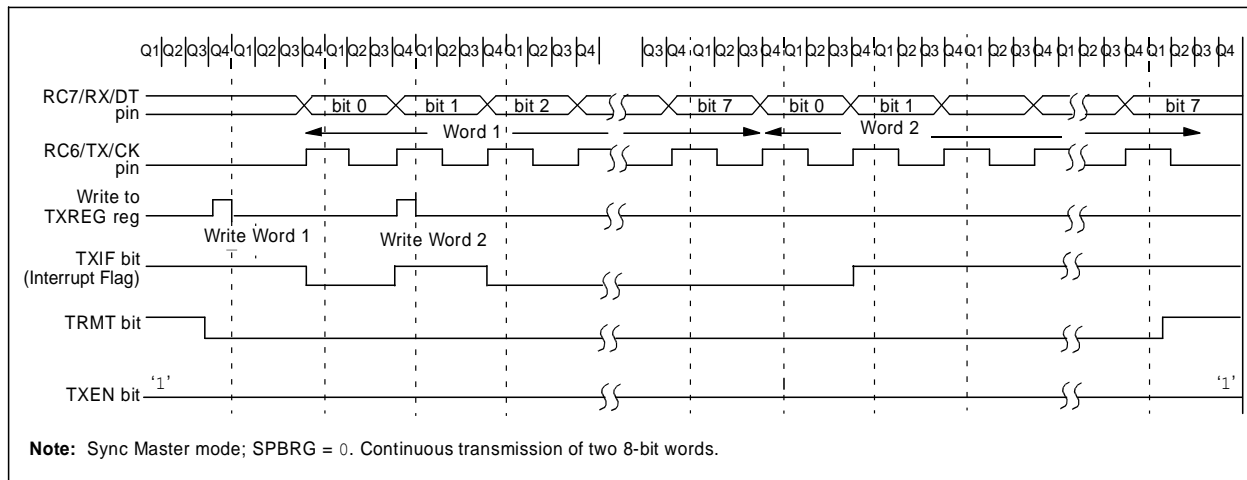
**TABLE 10-8: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	R0IF	0000 000x	0000 000u
0Ch	PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
19h	TXREG	USART Transmit Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

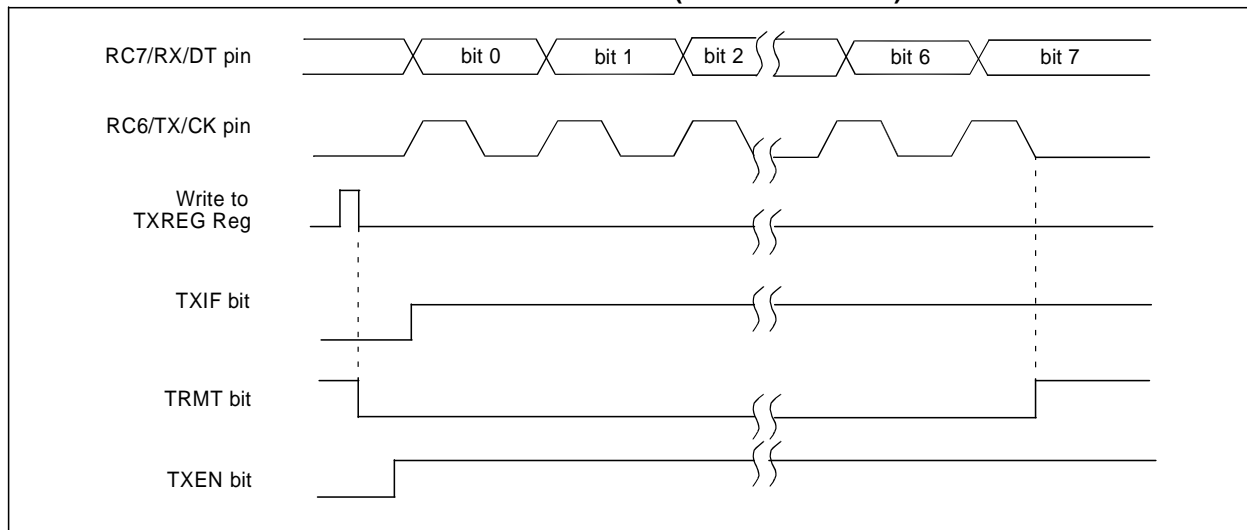
**Legend:** x = unknown, - = unimplemented, read as '0'. Shaded cells are not used for synchronous master transmission.

**Note 1:** Bits PSPIE and PSPIF are reserved on 28-pin devices; always maintain these bits clear.

**FIGURE 10-9: SYNCHRONOUS TRANSMISSION**



**FIGURE 10-10: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)**



### 10.3.2 USART SYNCHRONOUS MASTER RECEPTION

Once Synchronous mode is selected, reception is enabled by setting either enable bit, SREN (RCSTA<5>), or enable bit, CREN (RCSTA<4>). Data is sampled on the RC7/RX/DT pin on the falling edge of the clock. If enable bit SREN is set, then only a single word is received. If enable bit CREN is set, the reception is continuous until CREN is cleared. If both bits are set, CREN takes precedence. After clocking the last bit, the received data in the Receive Shift Register (RSR) is transferred to the RCREG register (if it is empty). When the transfer is complete, interrupt flag bit, RCIF (PIR1<5>), is set. The actual interrupt can be enabled/disabled by setting/clearing enable bit, RCIE (PIE1<5>). Flag bit RCIF is a read-only bit which is reset by the hardware. In this case, it is reset when the RCREG register has been read and is empty. The RCREG is a double-buffered register (i.e., it is a two-deep FIFO). It is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte to begin shifting into the RSR register. On the clocking of the last bit of the third byte, if the RCREG register is still full, then Overrun Error bit, OERR (RCSTA<1>), is set. The word in the RSR will be lost. The RCREG register can be read twice to retrieve the two bytes in the FIFO. Bit OERR has to be cleared in software (by clearing bit CREN). If bit OERR is set, transfers from the RSR to the RCREG are inhibited so it is essential to clear bit OERR if it is set. The ninth receive bit is buffered the same way as the receive

data. Reading the RCREG register will load bit RX9D with a new value, therefore, it is essential for the user to read the RCSTA register before reading RCREG in order not to lose the old RX9D information.

When setting up a Synchronous Master Reception:

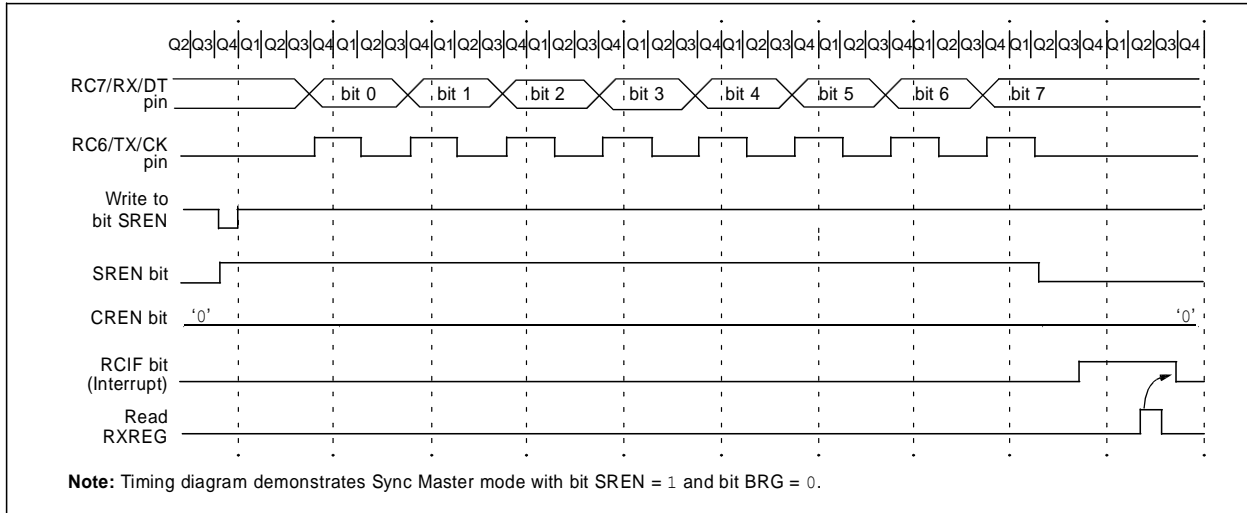
1. Initialize the SPBRG register for the appropriate baud rate (**Section 10.1 “USART Baud Rate Generator (BRG)”**).
2. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
3. Ensure bits CREN and SREN are clear.
4. If interrupts are desired, then set enable bit RCIE.
5. If 9-bit reception is desired, then set bit RX9.
6. If a single reception is required, set bit SREN. For continuous reception, set bit CREN.
7. Interrupt flag bit RCIF will be set when reception is complete and an interrupt will be generated if enable bit RCIE was set.
8. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
9. Read the 8-bit received data by reading the RCREG register.
10. If any error occurred, clear the error by clearing bit CREN.
11. If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCON register are set.

**TABLE 10-9: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	R0IF	0000 000x	0000 000u
0Ch	PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
1Ah	RCREG	USART Receive Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

**Legend:** x = unknown, - = unimplemented, read as '0'. Shaded cells are not used for synchronous master reception.

**Note 1:** Bits PSPIE and PSPIF are reserved on 28-pin devices; always maintain these bits clear.

**FIGURE 10-11: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)**

## 10.4 USART Synchronous Slave Mode

Synchronous Slave mode differs from the Master mode in the fact that the shift clock is supplied externally at the RC6/TX/CK pin (instead of being supplied internally in Master mode). This allows the device to transfer or receive data while in Sleep mode. Slave mode is entered by clearing bit, CSRC (TXSTA<7>).

### 10.4.1 USART SYNCHRONOUS SLAVE TRANSMIT

The operation of the Synchronous Master and Slave modes is identical, except in the case of the Sleep mode.

If two words are written to the TXREG and then the SLEEP instruction is executed, the following will occur:

- The first word will immediately transfer to the TSR register and transmit.
- The second word will remain in TXREG register.
- Flag bit TXIF will not be set.
- When the first word has been shifted out of TSR, the TXREG register will transfer the second word to the TSR and flag bit TXIF will now be set.
- If enable bit TXIE is set, the interrupt will wake the chip from Sleep and if the global interrupt is enabled, the program will branch to the interrupt vector (0004h).

When setting up a Synchronous Slave Transmission, follow these steps:

- Enable the synchronous slave serial port by setting bits SYNC and SPEN and clearing bit CSRC.
- Clear bits CREN and SREN.
- If interrupts are desired, then set enable bit TXIE.
- If 9-bit transmission is desired, then set bit TX9.
- Enable the transmission by setting enable bit TXEN.
- If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
- Start transmission by loading data to the TXREG register.
- If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCON register are set.

**TABLE 10-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE TRANSMISSION**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	R0IF	0000 000x	0000 000u
0Ch	PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
19h	TXREG	USART Transmit Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

**Legend:** x = unknown, - = unimplemented, read as '0'. Shaded cells are not used for synchronous slave transmission.

**Note 1:** Bits PSPIE and PSPIF are reserved on 28-pin devices; always maintain these bits clear.

#### 10.4.2 USART SYNCHRONOUS SLAVE RECEPTION

The operation of the Synchronous Master and Slave modes is identical, except in the case of the Sleep mode. Bit SREN is a “don't care” in Slave mode.

If receive is enabled by setting bit CREN prior to the SLEEP instruction, then a word may be received during Sleep. On completely receiving the word, the RSR register will transfer the data to the RCREG register and if enable bit RCIE bit is set, the interrupt generated will wake the chip from Sleep. If the global interrupt is enabled, the program will branch to the interrupt vector (0004h).

When setting up a Synchronous Slave Reception, follow these steps:

1. Enable the synchronous master serial port by setting bits SYNC and SPEN and clearing bit CSRC.
2. If interrupts are desired, set enable bit RCIE.
3. If 9-bit reception is desired, set bit RX9.
4. To enable reception, set enable bit CREN.
5. Flag bit RCIF will be set when reception is complete and an interrupt will be generated if enable bit RCIE was set.
6. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
7. Read the 8-bit received data by reading the RCREG register.
8. If any error occurred, clear the error by clearing bit CREN.
9. If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCON register are set.

# ISD4004 SERIES

**TABLE 10-11: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	R0IF	0000 000x	0000 000u
0Ch	PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
1Ah	RCREG	USART Receive Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

**Legend:** x = unknown, - = unimplemented, read as '0'. Shaded cells are not used for synchronous slave reception.

**Note 1:** Bits PSPIE and PSPIF are reserved on 28-pin devices, always maintain these bits clear.



# ISD4004 SERIES

## ANEXO III

# ISD4004 SERIES



## PIN DESCRIPTION

PIN NAME	PIN NO.		FUNCTION
	SOIC / PDIP	TSOP	
SS	1	9	<b>Slave Select:</b> This input, when LOW, will select the ISD4004 device.
MOSI	2	10	<b>Master Out Slave IN:</b> This is the serial input to the ISD4004 device when it is configured as slave. The master microcontroller places data on the MOSI line one half-cycle before the rising edge of SCLK for clocking into the device.
MISO	3	11	<b>Master In Slave Out:</b> This is the serial output (open drain) of the ISD4004 device. This output goes into a high-impedance state if the device is not selected.
V <sub>SSA</sub> / V <sub>SSD</sub>	11, 12, 23 / 4	1, 17, 18 / 12	<b>Ground:</b> The ISD4004 series utilizes separate analog and digital ground busses. The analog ground (V <sub>SSA</sub> ) pins should be tied together as close as possible and connected through a low-impedance path to power supply ground. The digital ground (V <sub>SSD</sub> ) pin should be connected through a separate low-impedance path to power supply ground. These ground paths should be large enough to ensure that the impedance between the V <sub>SSA</sub> pins and the V <sub>SSD</sub> pin is less than 3 Ω. The backside of the die is connected to V <sub>SS</sub> through the substrate. For chip-on-board design, the die attach area must be connected to V <sub>SS</sub> or left floating.
NC	5-10, 15, 19-22	3, 4, 13-16, 19, 21, 23, 27, 28	Not connected
AUD OUT <sup>[1]</sup>	13	20	<b>Audio Output:</b> This pin provides an audio output of the stored data and is recommended be AC coupled. It is capable of driving a 5 KΩ impedance R <sub>EXT</sub> .

<sup>[1]</sup> The AUD OUT pin is always at 1.2 volts when the device is powered up. When in playback, the output buffer connected to this pin can drive a load as small as 5 KΩ. When in record, a built-in resistor connects AUD OUT to the internal 1.2-volt analog ground supply. This resistor is approximately 850 KΩ, but will vary somewhat according to the sample rate of the device. This relatively high impedance allows this pin to be connected to an audio bus without loading it down.



## ISD4004 SERIES

PIN NAME	PIN NO.		FUNCTION
	SOIC / PDIP	TSOP	
AM CAP	14	22	<p><b>AutoMute™ Feature:</b> The AutoMute feature only applies for playback operation and helps to minimize noise (with 6 dB of attenuation) when there is no signal (i.e. during periods of silence). A 1 <math>\mu</math>F capacitor to ground is recommended to connect to the AM CAP pin.</p> <p>This capacitor becomes a part of an internal peak detector which senses the signal amplitude. This peak level is compared to an internally set threshold to determine the AutoMute trip point. For large signals, the AutoMute attenuation is set to 0 dB automatically but 6 dB of attenuation occurs for silence. The 1 <math>\mu</math>F capacitor also affects the rate at which the AutoMute feature changes with the signal amplitude (or the attack time).</p> <p>The AutoMute feature can be disabled by connecting the AM CAP pin directly to <math>V_{CCA}</math>.</p>
ANA IN-	16	24	<p><b>Inverting Analog Input:</b> This pin transfers the signal into the device during recording via differential-input mode.</p> <p>In this differential-input mode, a 16 mVp-p maximum input signal should be capacitively coupled to ANA IN- for optimal signal quality, as shown in Figure 1: ANA IN Modes. This capacitor value should be equal to that used on ANA IN+ pin. The input impedance at ANA IN- is normally 56 K<math>\Omega</math>.</p> <p>In the single-ended mode, ANA IN- should be capacitively coupled to <math>V_{SSA}</math> through a capacitor equal to that used on the ANA IN+ pin.</p>
ANA IN+	17	25	<p><b>Non-Inverting Analog Input:</b> This pin is the non-inverting analog input that transfers the signal to the device for recording. The analog input amplifier can be driven single ended or differentially.</p> <p>In the single-ended input mode, a 32 mVp-p (peak-to-peak) maximum signal should be capacitively connected to this pin for optimal signal quality. The external capacitor associated with ANA IN+ together with the 3 K<math>\Omega</math> input impedance are selected to give cutoff at the low frequency end of the voice passband.</p> <p>In the differential-input mode, the maximum input signal at ANA IN+ should be 16 mVp-p capacitively coupled for optimal signal quality. The circuit connections for the two modes are shown in Figure 1.</p>



## ISD4004 SERIES

PIN NAME	PIN NO.		FUNCTION
	SOIC / PDIP	TSOP	
$V_{CCA} / V_{CCD}$	18 / 27	26 / 7	<p><b>Supply Voltage:</b> To minimize noises, the analog and digital circuits in the ISD4004 devices use separate power busses. These +3V busses are brought out to separate pins and should be tied together as close to the supply as possible. In addition, these supplies should be decoupled as close to the package as possible.</p>
RAC	24	2	<p><b>Row Address Clock:</b> This is an open drain output that provides the signal of a ROW with a 200 ms period for 8 KHz sampling frequency. (This represents a single row of memory.) This signal stays HIGH for 175 ms and stays LOW for 25 ms when it reaches the end of a row.</p> <p>The RAC pin stays HIGH for 109.37 <math>\mu</math>sec and stays LOW for 15.63 <math>\mu</math>sec in Message Cueing mode (see Message Cueing section for detailed description). Refer to the AC Parameters table for RAC timing information at other sample rates.</p> <p>When a record command is first initiated, the RAC pin remains HIGH for an extra <math>T_{RACL}</math> period. This is due to the need of loading the internal sample and hold circuits in the device. This pin can be used for message management techniques.</p> <p>A pull-up resistor is required to connect this pin to other device.</p>
$\overline{\text{INT}}$	25	5	<p><b>Interrupt:</b> This is an open drain output pin. This pin goes LOW and stays LOW when an Overflow (OVF) or End of Message (EOM) marker is detected. Each operation that ends with an EOM or OVF will generate an interrupt. The interrupt will be cleared the next time an SPI cycle is initiated. The interrupt status can also be read by an <math>R_{INT}</math> instruction.</p> <p>A pull-up resistor is required to connect this pin to other device.</p> <p><i>Overflow Flag (OVF)</i> – The Overflow flag indicates that the end of memory has been reached during a record or playback operation.</p> <p><i>End of Message (EOM)</i> – The End of Message flag is set only during playback operation when an EOM is found. There are eight EOM flag position options per row.</p>

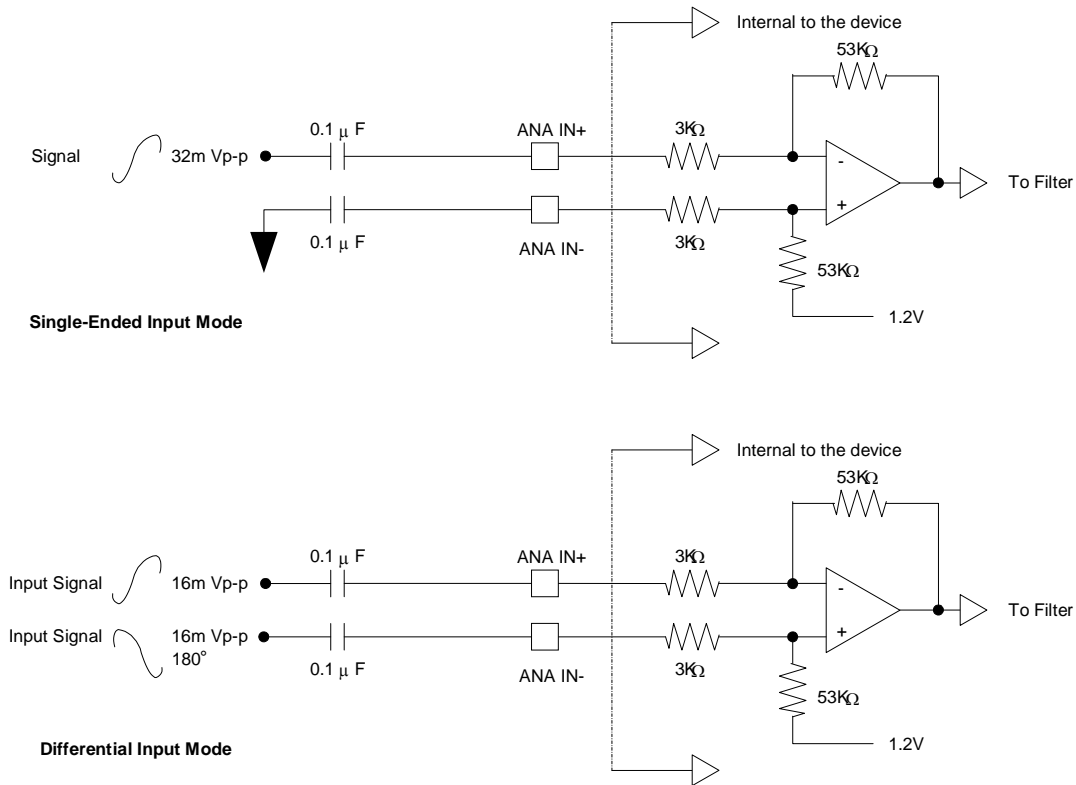


## ISD4004 SERIES

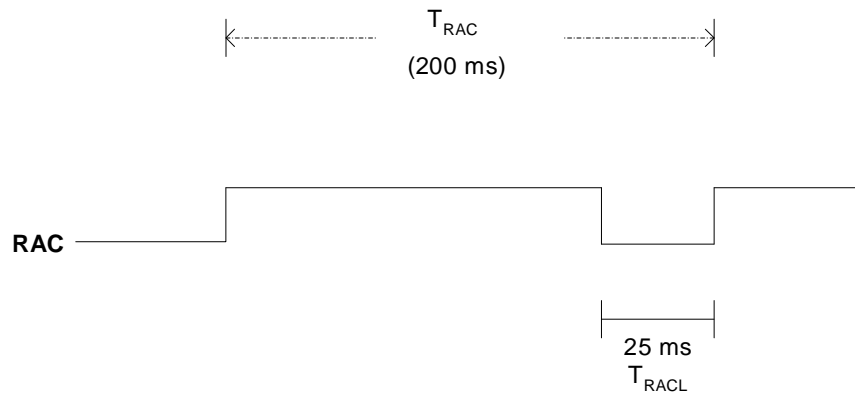
PIN NAME	PIN NO.		FUNCTION															
	SOIC / PDIP	TSOP																
XCLK	26	6	<p><b>External Clock Input:</b> The pin has an internal pull-down device. The ISD4004 series is configured at the factory with an internal sampling clock frequency centered to <math>\pm 1</math> percent of specification. The frequency is then maintained to a variation of <math>\pm 2.25</math> percent over the entire commercial temperature and operating voltage ranges. The internal clock has a <math>-6/+4</math> percent tolerance over the extended temperature, industrial temperature and voltage ranges. A regulated power supply is recommended for industrial temperature range parts. If greater precision is required, the device can be clocked through the XCLK pin as follows:</p> <table border="1"> <thead> <tr> <th>Part Number</th> <th>Sample Rate</th> <th>Required Clock</th> </tr> </thead> <tbody> <tr> <td>ISD4004-08M</td> <td>8.0 kHz</td> <td>1024 kHz</td> </tr> <tr> <td>ISD4004-10M</td> <td>6.4 kHz</td> <td>819.2 kHz</td> </tr> <tr> <td>ISD4004-12M</td> <td>5.3 kHz</td> <td>682.7 kHz</td> </tr> <tr> <td>ISD4004-16M</td> <td>4.0 kHz</td> <td>512 kHz</td> </tr> </tbody> </table> <p>These recommended clock rates should not be varied because the anti-aliasing and smoothing filters are fixed. Otherwise, aliasing problems can occur if the sample rate differs from the one recommended. The duty cycle on the input clock is not critical, as the clock is immediately divided by two. <b>If the XCLK is not used, this input must be connected to ground.</b></p>	Part Number	Sample Rate	Required Clock	ISD4004-08M	8.0 kHz	1024 kHz	ISD4004-10M	6.4 kHz	819.2 kHz	ISD4004-12M	5.3 kHz	682.7 kHz	ISD4004-16M	4.0 kHz	512 kHz
Part Number	Sample Rate	Required Clock																
ISD4004-08M	8.0 kHz	1024 kHz																
ISD4004-10M	6.4 kHz	819.2 kHz																
ISD4004-12M	5.3 kHz	682.7 kHz																
ISD4004-16M	4.0 kHz	512 kHz																
SCLK	28	8	<p><b>Serial Clock:</b> This is the input clock to the ISD4004 device. It is generated by the master device (typically microcontroller) and is used to synchronize the data transfer in and out of the device through the MOSI and MISO lines, respectively. Data is latched into the ISD4004 on the rising edge of SCLK and shifted out of the device on the falling edge of SCLK.</p>															



# ISD4004 SERIES



**FIGURE 1: ISD4004 SERIES ANA IN MODES**



**FIGURE 2: RAC TIMING WAVEFORM DURING NORMAL OPERATION**

(example of 8KHz sampling rate)



## ISD4004 SERIES

### FUNCTIONAL DESCRIPTION

#### DETAILED DESCRIPTION

##### Audio Quality

The Winbond's ISD4004 ChipCorder® series is offered at 8.0, 6.4, 5.3 and 4.0 kHz sampling frequencies, allowing the user a choice of speech quality options. Increasing the sampling frequency will produce better sound quality, but affects duration. Please refer to Table 1: Product Summary for details.

Analog speech samples are stored directly into on-chip non-volatile memory without the digitization and compression associated with other solutions. Direct analog storage provides higher quality reproduction of voice, music, tones, and sound effects than other solid-state solutions.

##### Duration

The ISD4004 Series is a single-chip solution with 8-, 10-, 12-, and 16-minute duration.

**TABLE 1: PRODUCT SUMMARY OF ISD4004 SERIES**

Part Number	Duration (Minutes)	Sample Rate (kHz)	Typical Filter Pass Band (kHz) *
ISD4004-08M	8	8.0	3.4
ISD4004-10M	10	6.4	2.7
ISD4004-12M	12	5.3	2.3
ISD4004-16M	16	4.0	1.7

\* This is the -3dB point. This parameter is not checked during production testing and may vary due to process variations and other factors. Therefore, the customer should not rely upon this value for testing purposes.

##### Flash Storage

The ISD4004 series utilizes on-chip Flash memory, providing zero-power message storage. The message is retained for up to 100 years typically without power. In addition, the device can be re-recorded typically over 100,000 times.

##### Memory Architecture

The ISD4004 series contains a total of 3,840K Flash memory cells, which is organized as 2,400 rows of 1,600 cells each. The address bits (A0-A15) are used to access various rows for multiple messages of different durations.



# ISD4004 SERIES

## Microcontroller Interface

A four-wire (SCLK, MOSI, MISO &  $\overline{SS}$ ) SPI interface is provided for controlling and addressing functions. The ISD4004 is configured to operate as a peripheral slave device, with a microcontroller-based SPI bus interface. Read and write operations are controlled through this SPI interface. An interrupt signal ( $\overline{INT}$ ) and internal read only Status Register are provided for handshake purposes.

## Programming

The ISD4004 series is also ideal for playback-only applications, where single- or multiple-messages playback is controlled through the SPI port. Once the desired message configuration is created, duplicates can easily be generated via a programmer.

### SERIAL PERIPHERAL INTERFACE (SPI) DESCRIPTION

The ISD4004 series operates via SPI serial interface with the following protocol.

First, the data transfer protocol assumes that the microcontroller's SPI shift registers are clocked on the falling edge of the SCLK. However, for the ISD4004, the protocols are as follows:

1. All serial data transfers begin with the falling edge of  $\overline{SS}$  pin.
2.  $\overline{SS}$  is held LOW during all serial communications and held HIGH between instructions.
3. Data is clocked in on the rising edge of the SCLK signal and clocked out on the falling edge of the SCLK signal, with LSB first.
4. Playback and record operations are initiated when the device is enabled by asserting the  $\overline{SS}$  pin LOW, shifting in an opcode and an address data to the ISD4004 device (refer to the Opcode Summary in the following page).
5. The opcodes contain <16 address bits> and <8 control bits>.
6. Each operation that ends with an EOM or Overflow will generate an interrupt. The Interrupt will be cleared the next time a SPI cycle is initiated.
7. As Interrupt data is shifted out of the MISO pin, while address and control data are simultaneously shifted into the MOSI pin. Care should be taken such that the data shifted in is compatible with current system operation. Because it is possible to read an interrupt data and start a new operation within the same SPI cycle.
8. An operation begins with the RUN bit set and ends with the RUN bit reset.
9. All operations begin after the rising edge of  $\overline{SS}$ .





# ISD4004 SERIES

## OPCODES

The available Opcodes are summarized as follows:

**TABLE 2: OPCODE SUMMARY**

Instructions	OpCodes		Descriptions
	Address (16 bits) <A0 – A15>	Control bits (8 bits) XXX C0 C1 C2 C3 C4	
POWERUP	<XXX.....XXX>	XXX 0 0 1 0 0	Power-Up: Device will be ready for an operation after $T_{PUD}$ .
SETPLAY	<A0 – A15>	XXX 0 0 1 1 1	Initiates playback from address <A0-A15>.
PLAY	<XXX.....XXX>	XXX 0 1 1 1 1	Playback from the current address (until EOM or OVF).
SETREC	<A0 – A15>	XXX 0 0 1 0 1	Initiates a record operation from address <A0-A15>.
REC	<XXX.....XXX>	XXX 0 1 1 0 1	Records from current address until OVF is reached or Stop command is sent.
SETMC	<A0 – A15>	XXX 1 0 1 1 1	Initiates Message Cueing (MC) from address <A0-A15>.
MC <sup>[1]</sup>	<XXX.....XXX>	XXX 1 1 1 1 1	Performs a Message Cueing from current location. Proceeds to the end of message (EOM) or enters OVF condition if no more messages are present.
STOP	<XXX.....XXX>	XXX 0 1 1 X 0	Stops the current operation.
STOPPWRDN	<XXX.....XXX>	XXX X 1 0 X 0	Stops the current operation and enters into standby (power-down) mode.
RINT <sup>[2]</sup>	<XXX.....XXX>	XXX 0 1 1 X 0	Read Interrupt status bits: Overflow and EOM.

Notes:

C0 = Message cueing

C1 = Ignore address bit

C2 = Master power control

C3 = Record or playback operation

C4 = Enable or disable an operation

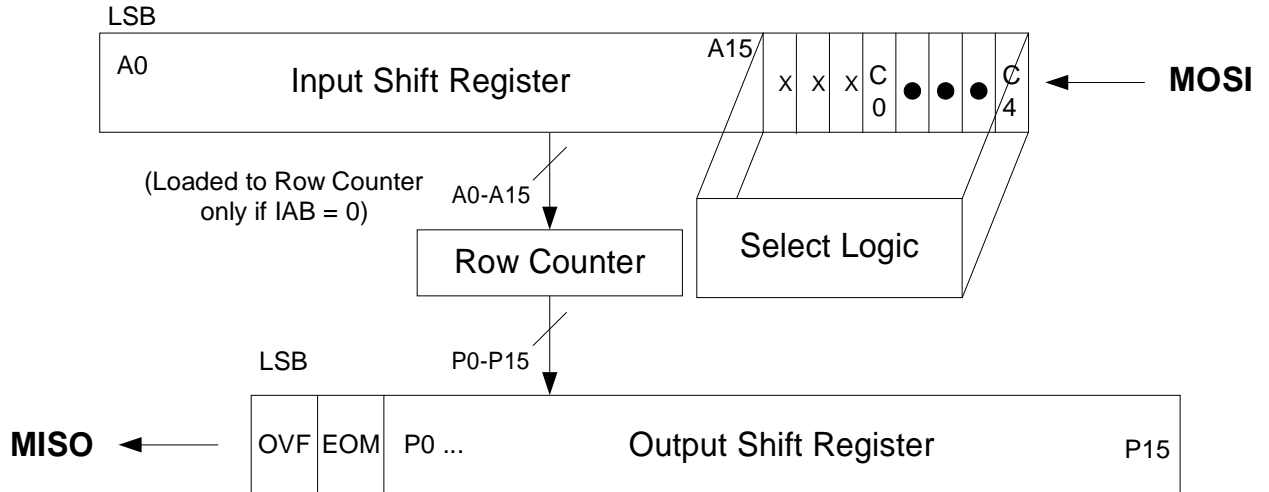
<sup>[1]</sup> Message Cueing can be selected only at the beginning of a playback operation.

<sup>[2]</sup> As the Interrupt data is shifted out of the ISD4004, control and address data are being shifted in. Care should be taken such that the data shifted in is compatible with current system operation. It is possible to read interrupt data and start a new operation at the same time. See Figures 5 - 8 for references.



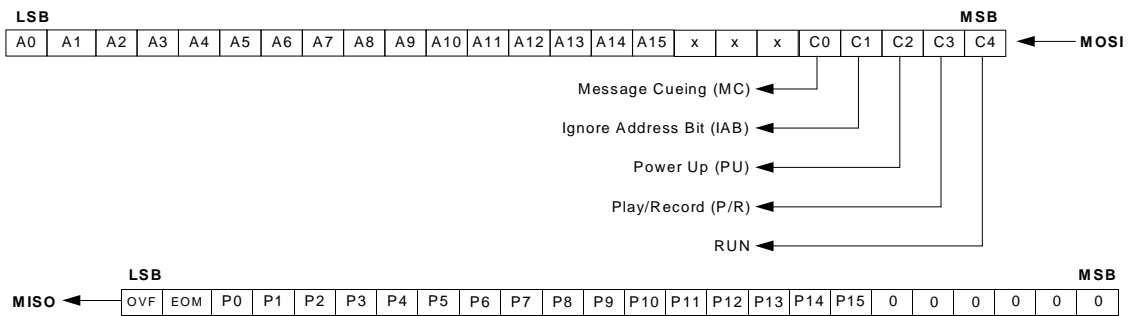
# ISD4004 SERIES

## SPI Diagrams



**FIGURE 3: SPI INTERFACE SIMPLIFIED BLOCK DIAGRAM**

The following diagram describes the SPI port and the control bits associated with it.



- Notes: 1. For MOSI, LSB is the 1<sup>st</sup> bit shifted into the ISD4004.
- 2. For MISO, LSB is the 1<sup>st</sup> bit shifted out from the ISD4004.

**FIGURE 4: SPI PORT**



## ISD4004 SERIES

### SPI Control and Output Registers

The SPI control register provides control of individual function such as play, record, message cueing, power-up, power-down, start, stop and ignore address pointer operations.

**TABLE 3: SPI CONTROL REGISTERS**

Control Bit	Control Register	Bit	Device Function
C0	MC		Message Cueing function
		= 1	Enable Message Cueing
		= 0	Disable Message Cueing
C1	IAB <sup>[1]</sup>		Ignore Address bit
		= 1	Ignore input address register (A0-A15)
		= 0	Use the input address register (A0-A15)
C2	PU		Power Up
		= 1	Power-Up
		= 0	Power-Down
C3	P/ $\bar{R}$		Playback or Record
		= 1	Play
		= 0	Record
C4	RUN		Enable or Disable an operation
		= 1	Start
		= 0	Stop
Address Bits	A0-A15		Input address register

**TABLE 4: SPI OUTPUT REGISTERS**

Output Bits	Description
OVF	Overflow
EOM	End-of-Message
P0-P15	Output of the row pointer register

<sup>[1]</sup> When IAB (Ignore Address Bit) is set to 0, a playback or record operation starts from address (A0-A15). For consecutive playback or record, IAB should be changed to a 1 before the end of that row (see RAC timing). Otherwise the ISD4004 will repeat the operation from the same row address. For memory management, the Row Address Clock (RAC) signal and IAB can be used to move around the memory segments.



## ISD4004 SERIES

### Message Cueing

Message cueing (MC) allows the user to skip through messages, without knowing the actual physical location of the messages. It will stop when an EOM marker is reached. Then, the internal address counter will point to the next message. Also, it will enter into OVF condition when it reaches the end of memory. In this mode, the messages are skipped 1,600 times faster than the normal playback mode.

### Power-Up Sequence

The ISD4004 will be ready for an operation after power-up command is sent and followed by the  $T_{PUD}$  timing (25 ms for 8 KHz sampling rate). Refer to the AC timing table for other  $T_{PUD}$  values with respect to different sampling rates.

The following sequences are recommended for optimized Record and Playback operations.

#### **Record Mode**

1. Send POWERUP command.
2. Wait  $T_{PUD}$  (power-up delay).
3. Send POWERUP command.
4. Wait  $2 \times T_{PUD}$  (power-up delay).
5. a). Send SETREC command with address xx, or  
b). Send REC command (recording from current location).
6. Send STOP command to stop recording.
7. Wait  $T_{STOP/PAUSE}$ .

For 3 & 4), please refer to Apps Brief 39A: recorded pop elimination in the ISD4000 series.

For 5.a), the device will start recording at address xx and will generate an interrupt when an overflow (end of memory array) is reached, if no STOP command is sent before that. Then, it will automatic stop recording operation.

#### **Playback Mode**

1. Send POWERUP command
2. Wait  $T_{PUD}$  (power-up delay)
3. a). Send SETPLAY command with address xx, or  
b). Send PLAY command (playback from current location).
4. a). Send STOP command to halt the playback operation, or  
b). Wait for playback operation to stop automatically, when an EOM or OVF is reached.
5. Wait  $T_{STOP/PAUSE}$ .

For 3.a), the device will start playback at address xx and it will generate an interrupt when an EOM or OVF is reached. It will then stop playback operation.

# ISD4004 SERIES



## TIMING DIAGRAMS

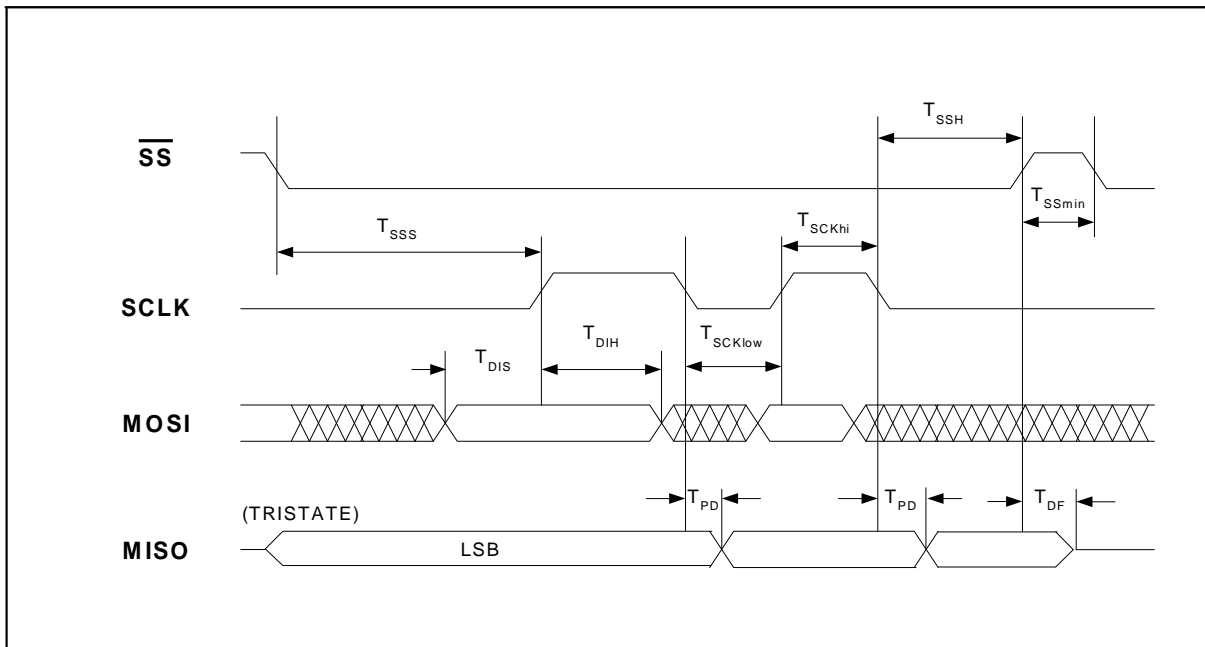


FIGURE 5: TIMING DIAGRAM

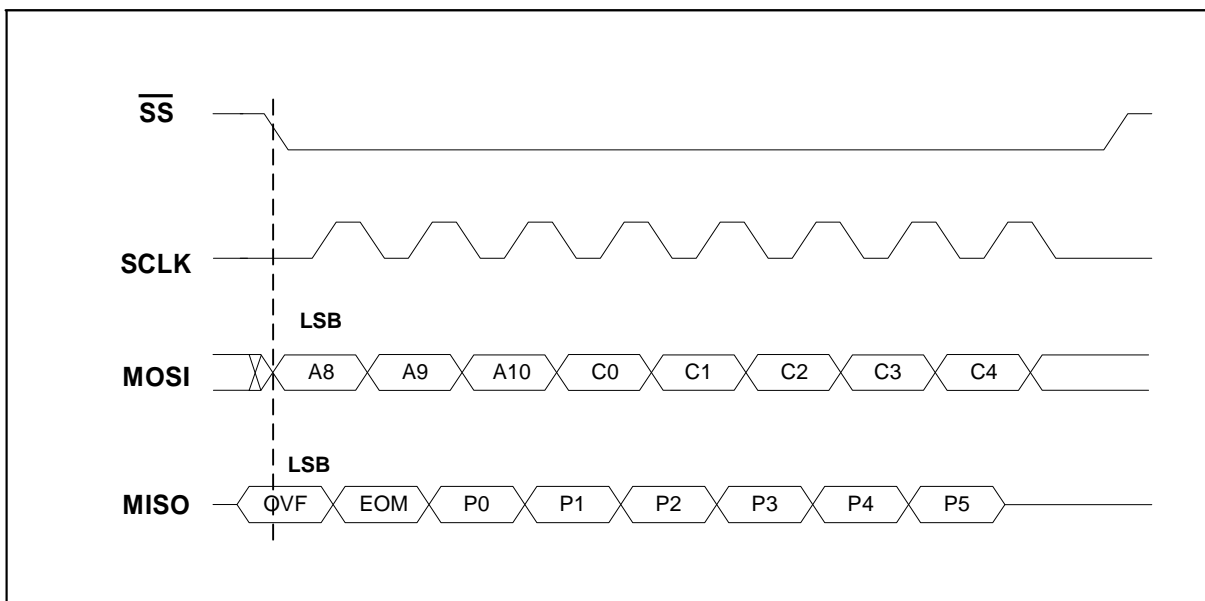


FIGURE 6: 8-BIT COMMAND FORMAT

# ISD4004 SERIES

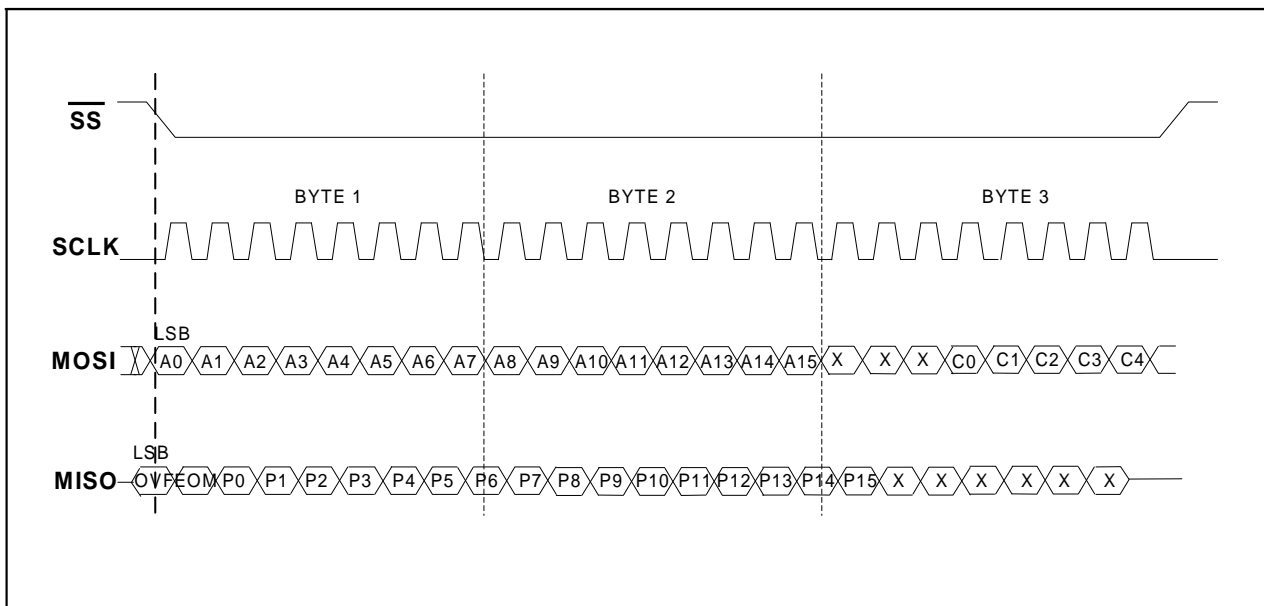


FIGURE 7: 16-BIT COMMAND FORMAT

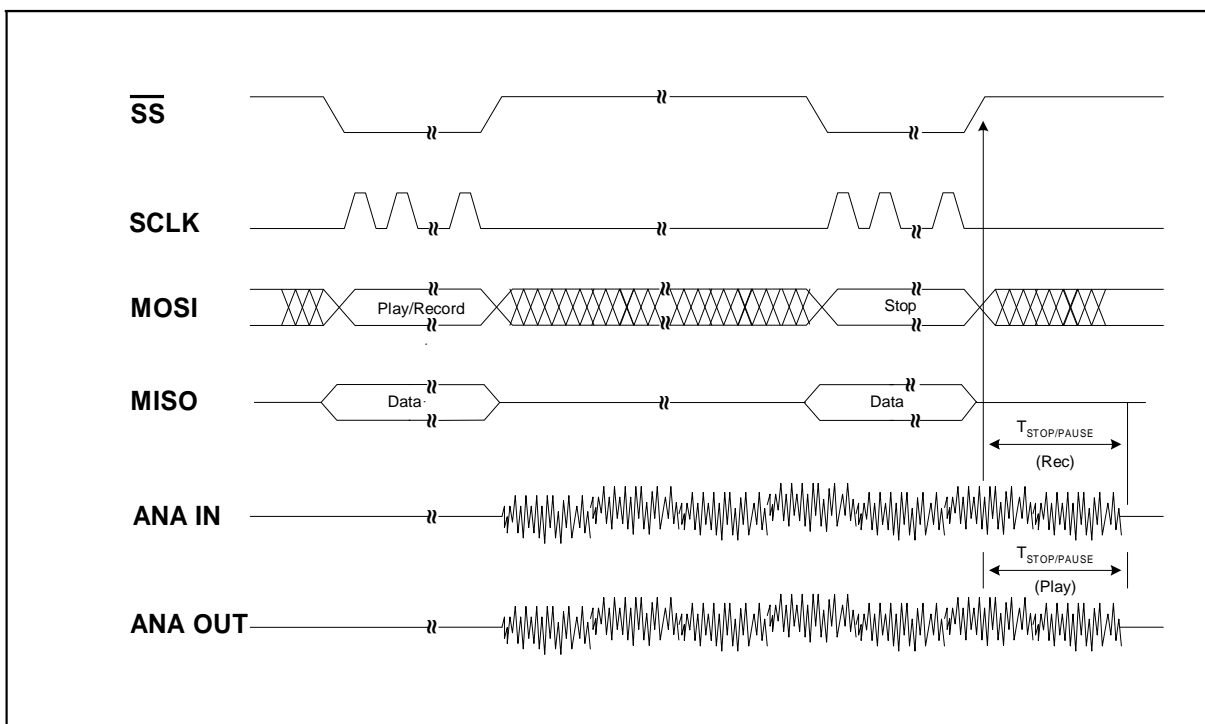


FIGURE 8: PLAYBACK/RECORD AND STOP CYCLE



# ISD4004 SERIES

## ABSOLUTE MAXIMUM RATINGS

### OPERATING CONDITIONS

**TABLE 7: OPERATING CONDITIONS (PACKAGED PARTS)**

CONDITIONS	VALUES
Commercial operating temperature range (Case temperature)	0°C to +70°C
Extended operating temperature (Case temperature)	-20°C to +70°C
Industrial operating temperature (Case temperature)	-40°C to +85°C
Supply voltage ( $V_{CC}$ ) <sup>[1]</sup>	+2.7V to +3.3V
Ground voltage ( $V_{SS}$ ) <sup>[2]</sup>	0V

**TABLE 8: OPERATING CONDITIONS (DIE)**

CONDITIONS	VALUES
Commercial operating temperature range	0°C to +50°C
Supply voltage ( $V_{CC}$ ) <sup>[1]</sup>	+2.7V to +3.3V
Ground voltage ( $V_{SS}$ ) <sup>[2]</sup>	0V

<sup>[1]</sup>  $V_{CC} = V_{CCA} = V_{CCD}$

<sup>[2]</sup>  $V_{SS} = V_{SSA} = V_{SSD}$



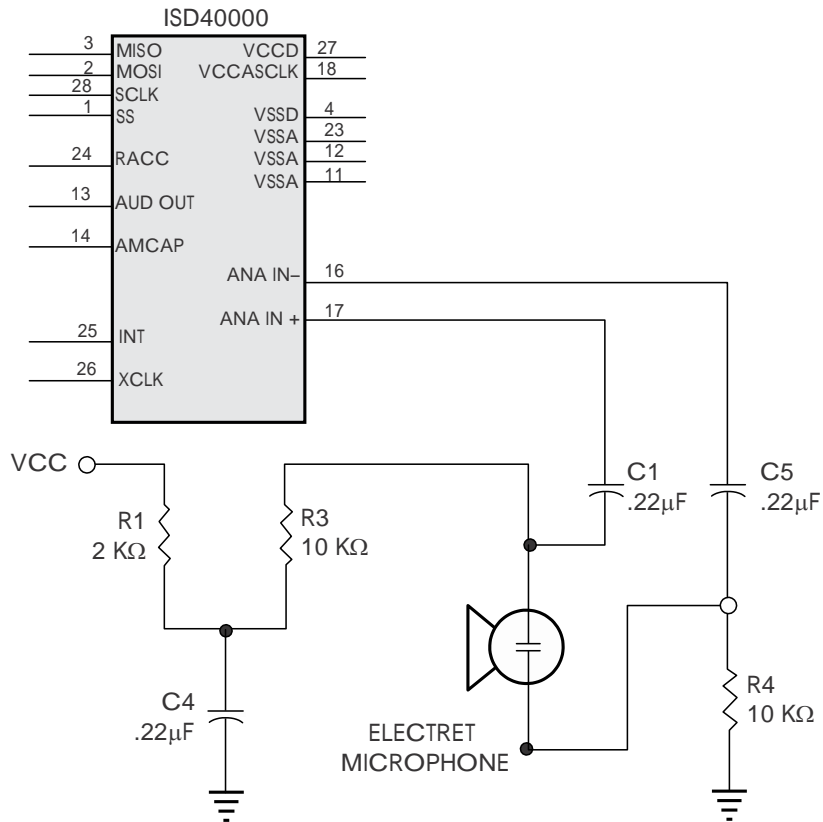


### APPLICATIONS BRIEF 22A — MICROPHONE CIRCUIT EXAMPLE FOR ISD4000 SERIES DEVICES

This microphone circuit can be used for ISD4000 series device applications where there is no need for AGC. Applications such as memo recorders or toys are some examples. In many applications, the performance of this circuit will be found adequate.

Please refer to Applications Note 4 for an AGC example.

**Figure 24: Circuit Example**



NOTE: Only microphone connections are shown.

**Table 11: Passive Component Functions**

Part	Function	Comments
R1, C4	Microphone power supply decoupling	Reduces power supply noise.
R3, R4	Microphone biasing resistors	Provide biasing for Microphone Operation.
C1, C5	Microphone DC-blocking capacitor low frequency cutoff	Decouples Microphone bias from the chip. Provides single-pole frequency cutoff and common mode noise rejection.

## ANEXO IV

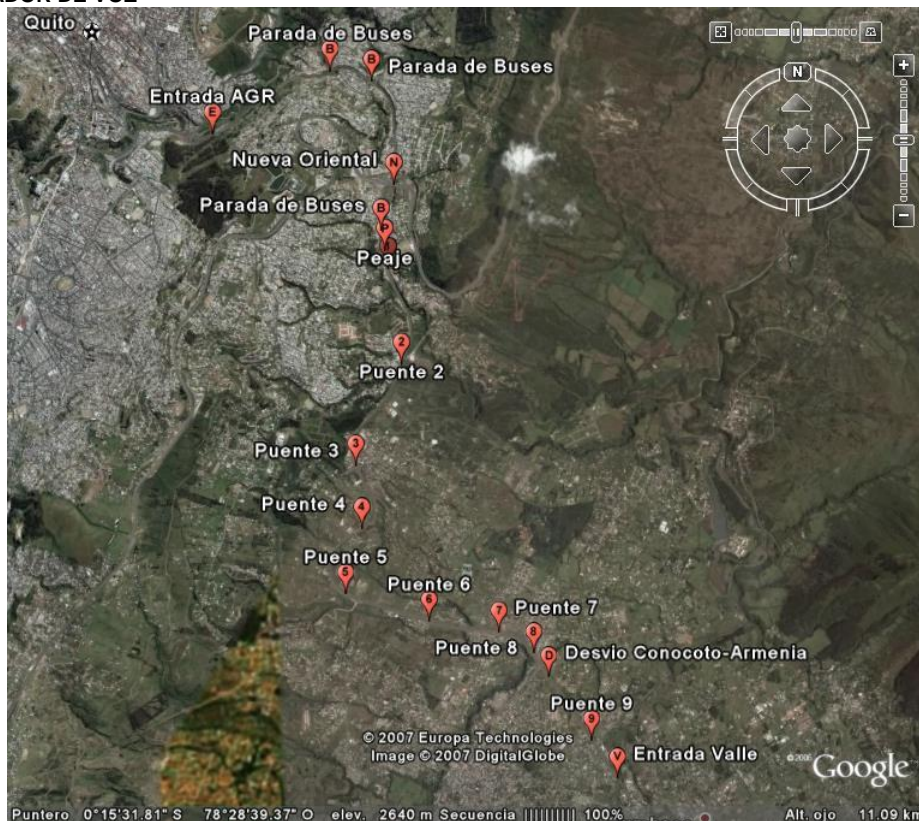


Figura 1. Autopista General Rumiñahui

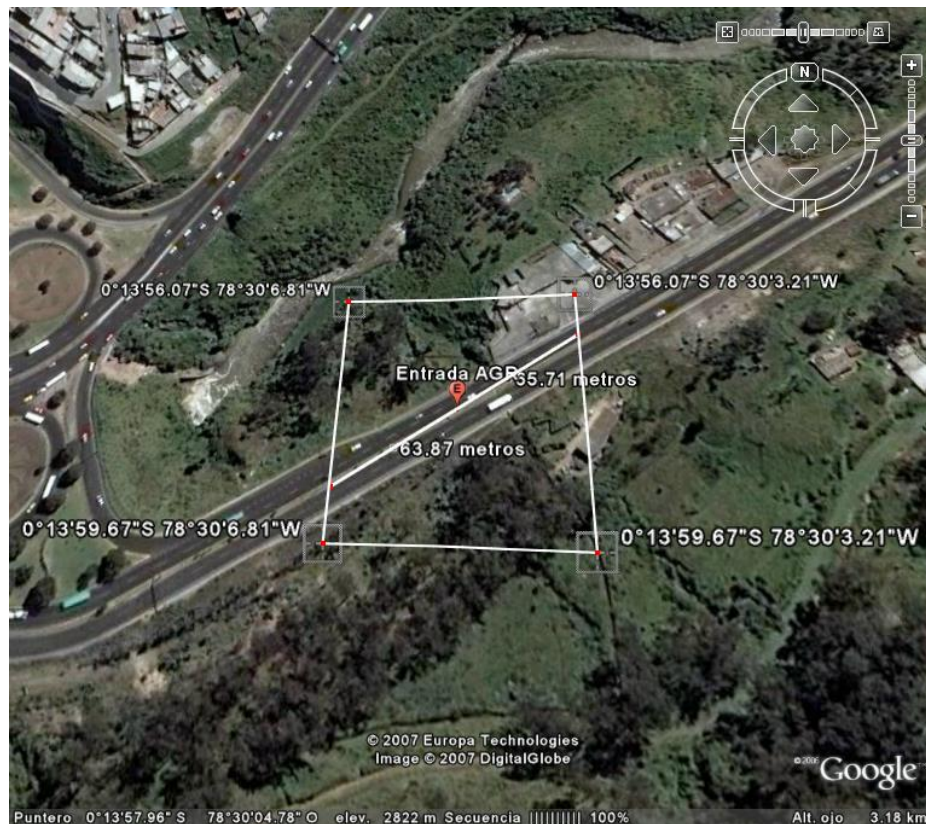


Figura 2. Entrada Trébol

Publication Release Date: October 26, 2005

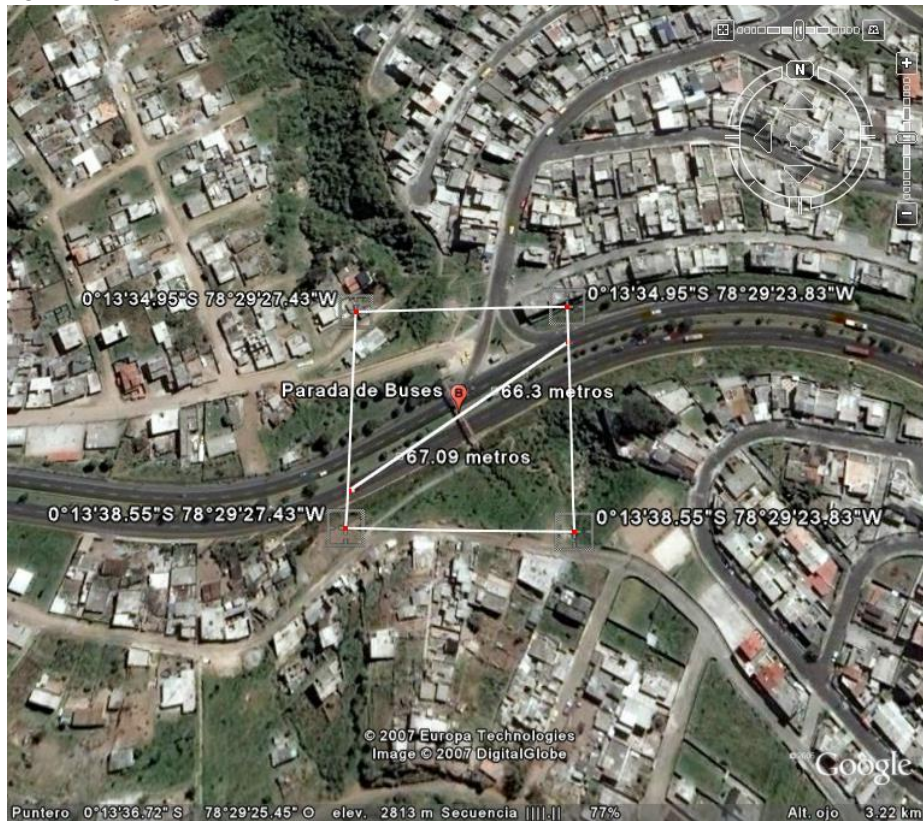


Figura 3. Parada de Buses Jardín del Valle



Figura 4. Parada de Buses Manjás

Publication Release Date: October 26, 2005

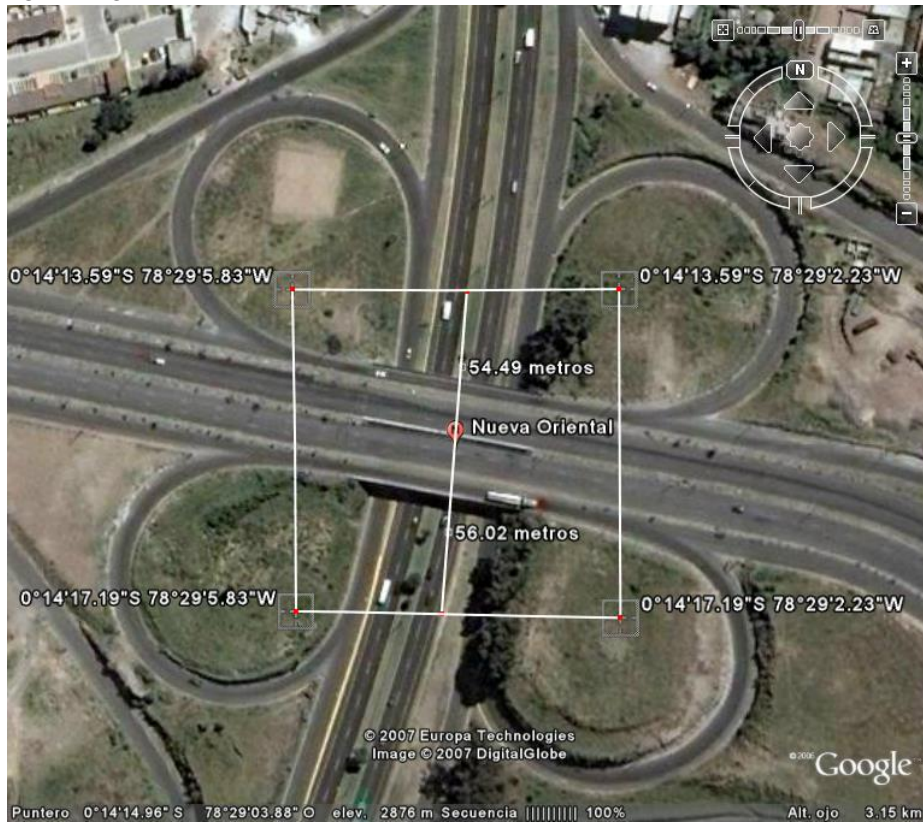


Figura 5. Desvío Nueva Oriental



Figura 6. Parada de Buses

Publication Release Date: October 26, 2005



Figura 7. Peaje



Figura 8. Puente 1  
Publication Release Date: October 26, 2005



Figura 9. Puente 2

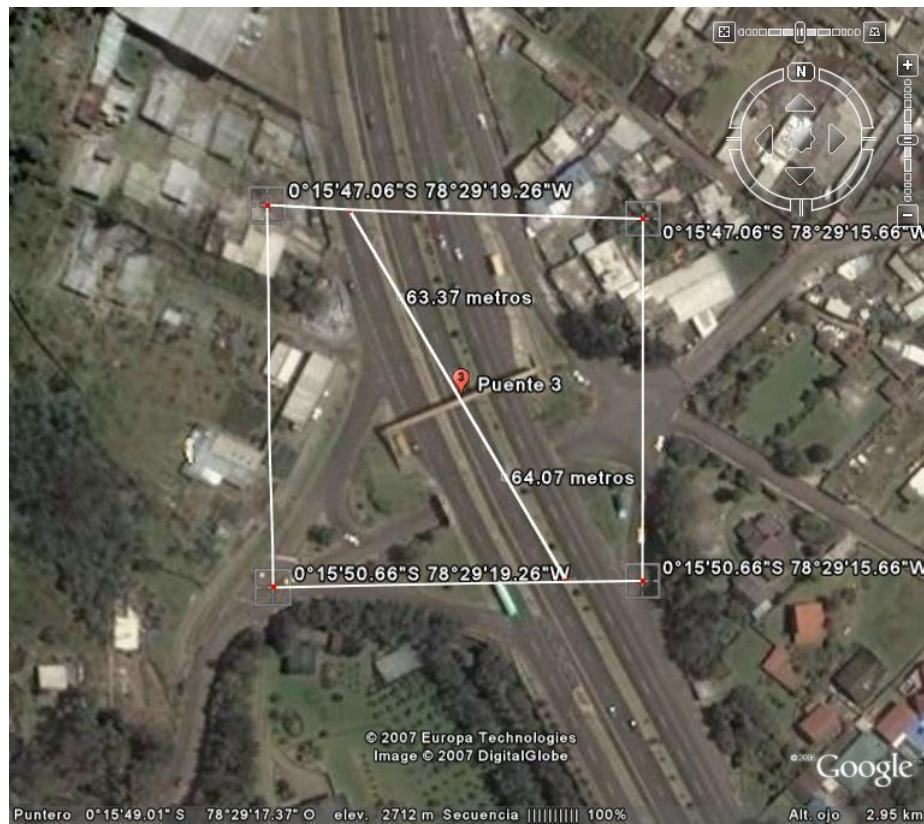


Figura 10. Puente 3  
Publication Release Date: October 26, 2005



Figura 11. Puente 4

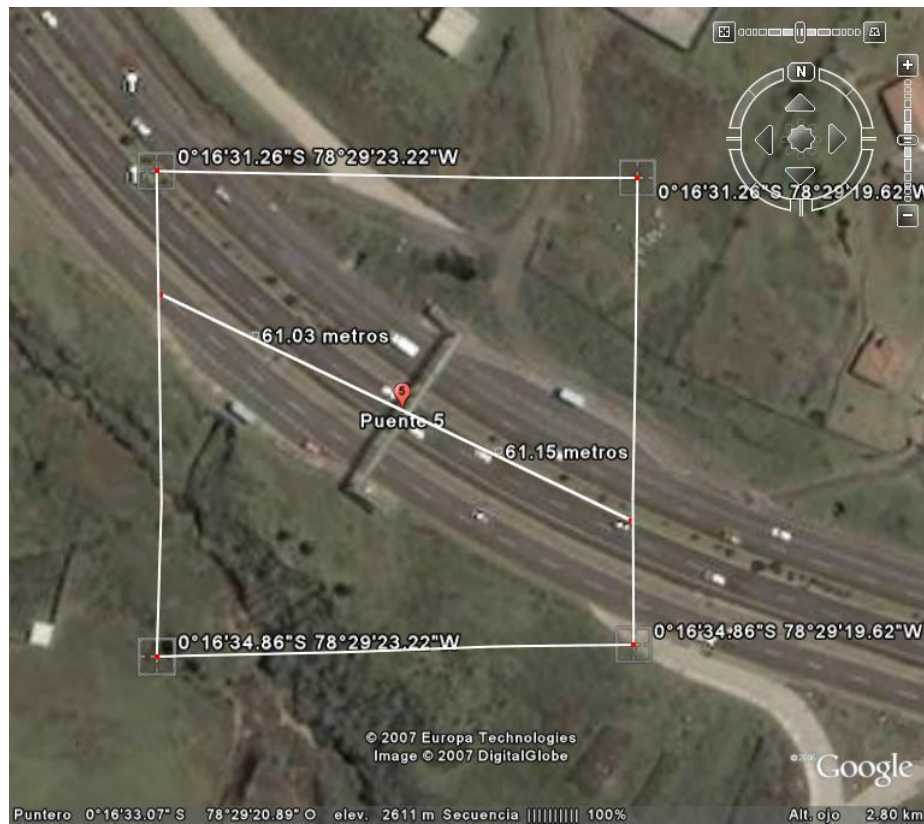


Figura 12. Puente 5

Publication Release Date: October 26, 2005





Figura 13. Puente 6

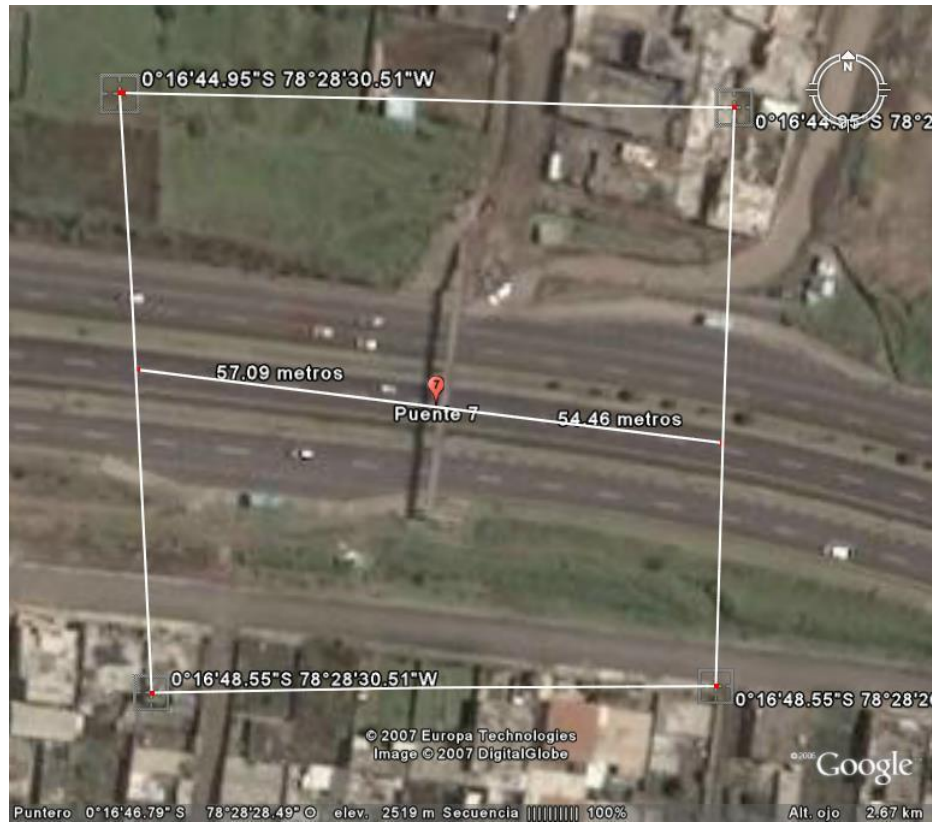


Figura 14. Puente 7

Publication Release Date: October 26, 2005



Figura 15. Puente 8

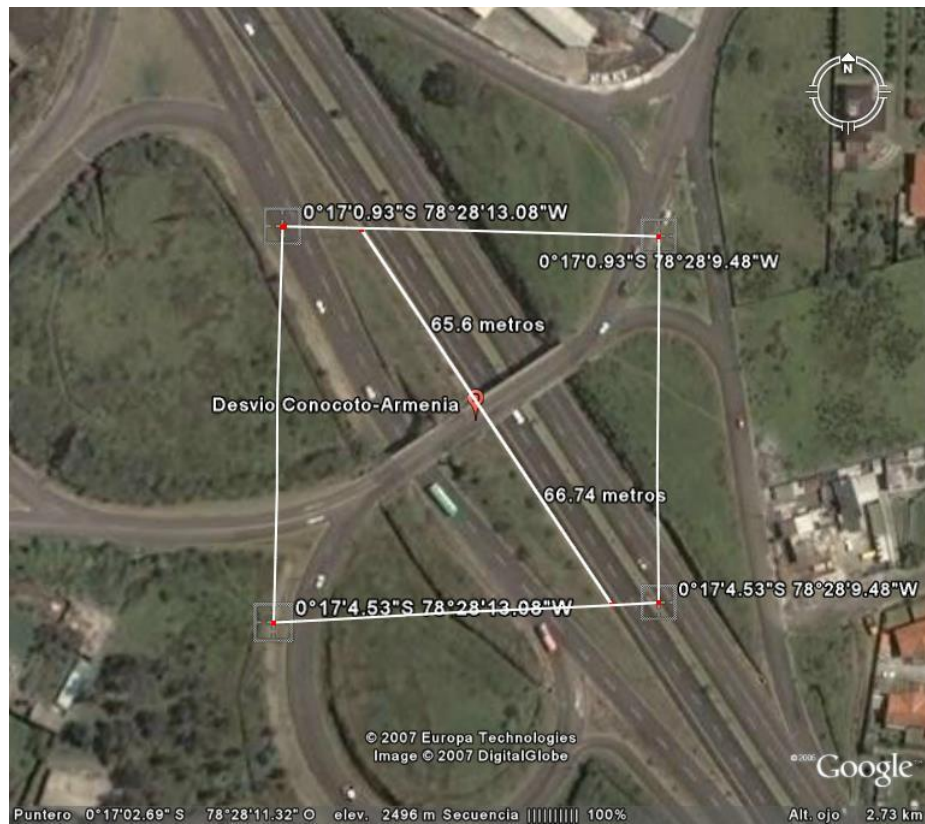


Figura 16. Desvío Conocoto-Armenia  
Publication Release Date: October 26, 2005

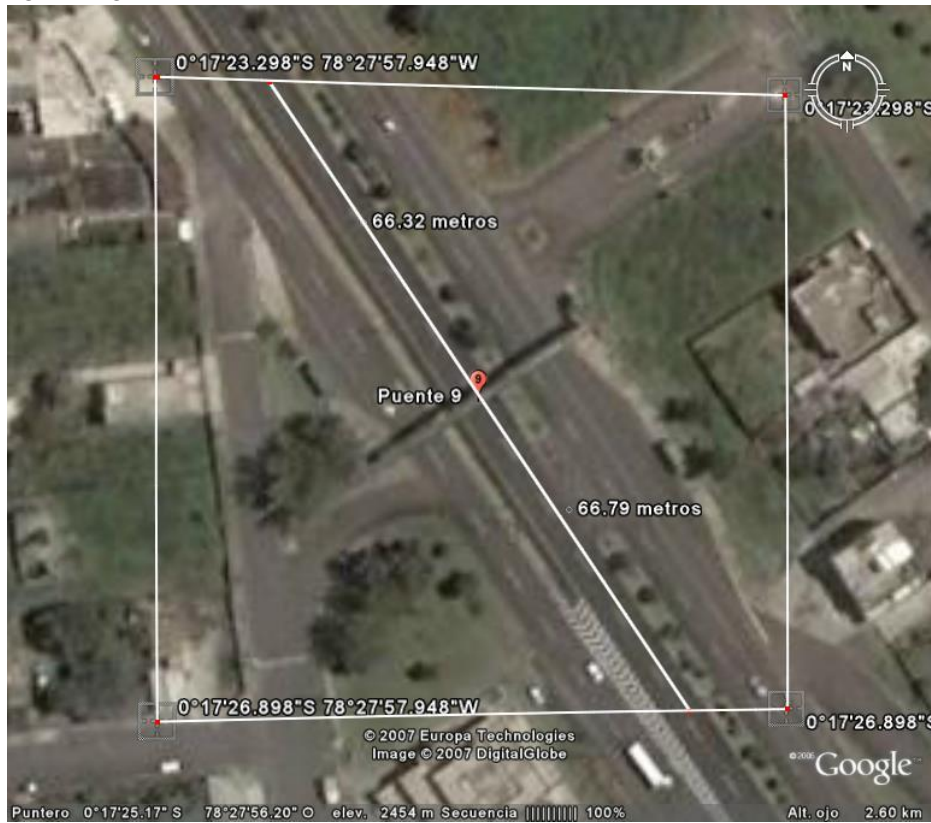


Figura 17. Puente 9

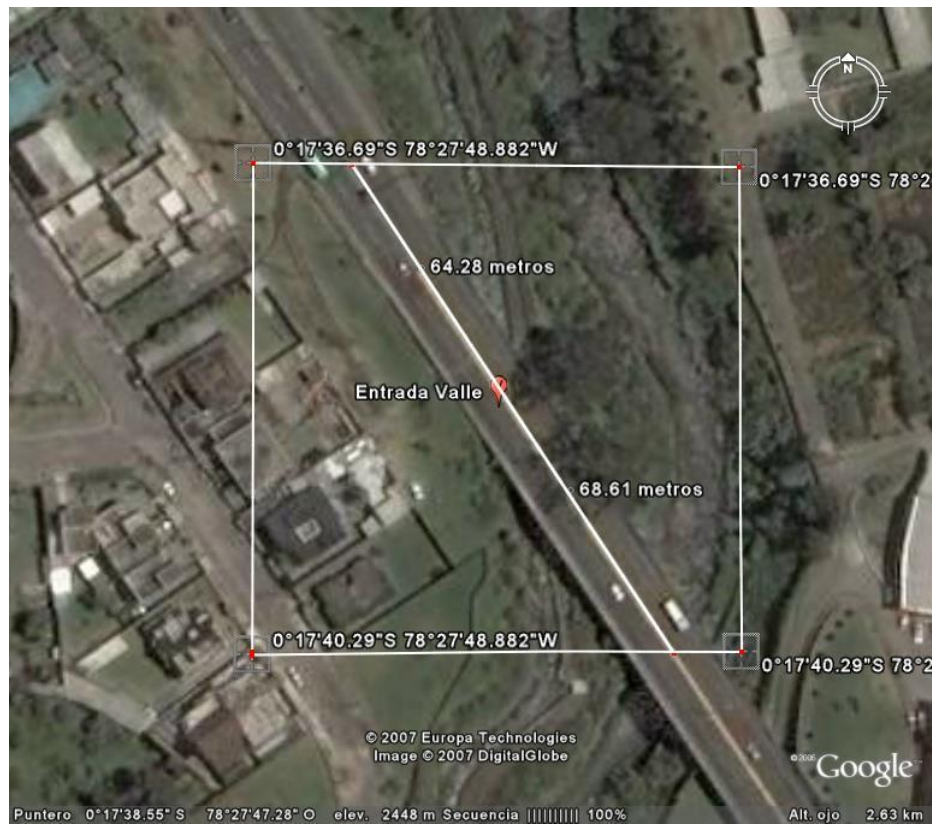


Figura 18. Entrada Valle

Publication Release Date: October 26, 2005

## REFERENCIAS BIBLIOGRÁFICAS

- Oscar Luis Vele, “Programación de PICs en Lenguaje C”, Cuenca, Ecuador.
- “PICC User Manual”, en <http://www.htsoft.com>, 2005.
- LÓPEZ, German, MARGNI, Santiago, “Proyecto de Grado: Programación de Microcontroladores: Rutinas, detalles y sintaxis”, Uruguay, 2003.
- “GPS – NMEA Reference”, en <http://home.mira.net/~gnb/gps/nmea.html#allgp>.
- “NMEA Command Reference Manual”, en [http://www.usglobalsat.com/downloads/NMEA\\_commands.pdf](http://www.usglobalsat.com/downloads/NMEA_commands.pdf)
- “SIRF Binary Protocol Reference Manual”, en [http://www.usglobalsat.com/downloads/SiRF\\_Binary\\_Protocol.pdf](http://www.usglobalsat.com/downloads/SiRF_Binary_Protocol.pdf)
- “ISD 4004 Series Data Sheet”, Winbond Electronics Corp., Taiwán, 2005.
- “EM-401 GPS Receiver Engine borrada Data Sheet”, GlobalSat Technology Corp., Taiwán, 2005.
- “Applications Note 5A”, Winbond Electronics Corp., Taiwán.
- “Basic Address Settings For ISD4000 Series”, Winbond Electronics Corp., Taiwán, 2006.
- “Example Circuit: Microphone Circuit” Winbond Electronics Corp., Taiwán.
- “PIC16F87XA Data Sheet”, Microchip Technology Inc., USA, 2003.
- “MC78XX/LM78XX/MC78XXA 3-Terminal 1A Positive Voltage Regulador”, en <http://www.fairchildsemi.com>
- “LM317 3-Terminal Positive Adjustable Regulator”, en <http://www.fairchildsemi.com>