



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**  
**ESCUELA DE INGENIERÍA ELECTRÓNICA**  
**TELECOMUNICACIONES Y REDES**

“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE TRANSFERENCIA DE  
DATOS POR TELEMETRIA PARA UN VEHICULO DE COMPETENCIA CON  
EL USO TECNOLOGÍA INALÁMBRICA BLUETOOTH”

**TESIS DE GRADO**  
**Previa a la obtención del título de**  
**INGENIERO EN ELECTRÓNICA TELECOMUNICACIONES Y**  
**REDES**

**Presentado Por:**

VICTOR HUGO TAPIA BERRONES

**RIOBAMBA – ECUADOR**

**2013**

**A los Ingenieros:**

**Edwin Altamirano**

**Diego Álvarez**

**Neiser Ortiz**

Mi más sincero agradecimiento por brindarme su amistad y constante apoyo para la realización del presente trabajo, de igual manera aquellas personas que he conocido a lo largo de mi vida estudiantil y que aportaron con su valioso conocimiento.

Un agradecimiento de manera especial al Taller Mecánico "ALVAREZ". Y a todo su personal en especial a la área Técnica; por el asesoramiento a lo largo de la tesis, brindándonos su confianza al compartir su tiempo, conocimiento y amistad.

Por sobre todo primero a Dios y a la Virgen de Agua Santa, que me ha sabido entregar su amor y comprensión a través de mi Madre y Hermana, Ana Berrones y Sandra Tapia, por darme una hija tan hermosa como es Mishell Tapia, por la paciencia de su madre Vero y la familia Maila Toledo, a quienes agradezco su entero apoyo, su amor y comprensión a pesar de mis errores; gracias por que siempre se encuentran a mi lado de una o de otra forma guiándome en el camino correcto.

De igual manera a mis amigos incondicionales que con sus constantes palabras de apoyo, con su motivación, me guiaron a seguir adelante; a Lic. Marianita quien confió en mí, y sobre todo al Ing. Diego Álvarez por depositar su confianza, para realizar el presente proyecto de tesis.

*VICTOR*

## FIRMAS RESPONSABLES Y NOTA

NOMBRE	FIRMA	FECHA
Ing. Iván Ménes <b>DÉCANO FACULTAD INFORMÁTICA Y ELECTRÓNICA</b>	.....	.....
Ing. Wilson Baldeón <b>DIRECTOR DE ESCUELA ING. EN ELECTRONICA TELECOMUNICACIONES Y REDES</b>	.....	.....
Ing. Edwin Altamirano <b>DIRECTOR DE TESIS</b>	.....	.....
Ing. Neiser Ortiz <b>MIEMBRO DEL TRIBUNAL</b>	.....	.....
Tlgo. Carlos Rodríguez <b>DIRECTOR CENTRO DE DOCUMENTACIÓN</b>	.....	.....
<b>NOTA DE LA TESIS</b>	.....	

“Yo, Víctor Hugo Tapia Berrones soy responsable de las ideas, doctrinas y resultados expuestos en esta tesis; y, el patrimonio intelectual de la Tesis de Grado pertenece a la ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO”.

---

AUTOR

## ÍNDICE DE ABREVIATURAS

<b>ADC</b>	Convertidor Analógico Digital (Analog-to-Digital Converter)
<b>ARP</b>	Nombre del Punto de Acceso
<b>ASCII</b>	Código Estadounidense Estandarizado para el intercambio de Información
<b>AFR</b>	Relación Aire / Combustible
<b>CMOS</b>	Semiconductor de Metal Oxido Complementario
<b>CLT</b>	Sensor de temperatura del refrigerante del motor
<b>ECM</b>	Mando de control electrónico, sinónimo de ECU
<b>ECU</b>	Unidad Electrónica de Control
<b>EEPROM</b>	Memoria de solo lectura eléctricamente borrable
<b>EIA-232</b>	Alianza de Industrias Electrónicas – 232
<b>FUEL REQ</b>	Requerimiento de Combustible
<b>HTTP</b>	Protocolo de transferencia de Híper-texto
<b>LAN</b>	Red de Área Local (Local Area Network)
<b>MAT</b>	Sensor de temperatura de aire del MANIFOLD
<b>MAP</b>	Sensor de presión del aire en el MANIFOLD
<b>MSSP</b>	Módulo de Puerto Serial Máster Síncrono
<b>MOSFET</b>	Transistor de Efecto de CMOS
<b>NTC</b>	Coeficiente negativo de temperatura
<b>O2</b>	Sensor de Oxígeno o Sonda Lambda
<b>PLL</b>	En español: Fase de Lazo Cerrado
<b>PBC</b>	En español: Placa de circuito impreso
<b>PWM</b>	Modulación por Ancho de Pulso
<b>RPM</b>	Revoluciones por minuto
<b>RAM</b>	Memoria de acceso aleatorio
<b>ROM</b>	Memoria de solo lectura
<b>RF</b>	Radio Frecuencia
<b>RS-232</b>	Estándar recomendado 232
<b>RTC</b>	Contador de Tiempo Real
<b>Rx</b>	Recepción
<b>SCI</b>	Interfaz serial de comunicaciones
<b>SPI</b>	Interfaz Serial de Periféricos
<b>TTL</b>	Lógica de Transistor a Transistor
<b>Tx</b>	Transmisión
<b>TPS</b>	Sensor de posición del acelerador
<b>TELNET</b>	Red de Telecomunicaciones
<b>USART</b>	Módulo de Transmisión y Recepción Síncrona y Asíncrona Universal
<b>WPAN</b>	Red de Área personal Inalámbrica

# ÍNDICE GENERAL

INTRODUCCIÓN .....	14
CAPÍTULO I .....	16
MARCO REFERENCIAL .....	16
1.1.- ANTECEDENTES.....	16
1.2.- JUSTIFICACIÓN.....	19
1.3.- OBJETIVOS.....	20
1.3.1.- OBJETIVO GENERAL: .....	20
1.3.2.- OBJETIVOS ESPECÍFICOS: .....	20
1.4.-HIPÓTESIS.....	21
CAPÍTULO II .....	22
TELEMETRIA Y SOFTWARE LIBRE.....	22
2.1- TELEMETRIA.....	22
2.1.1.- DEFINICIÓN DE TELEMETRÍA .....	22
2.1.2.- APLICACIONES DE TELEMETRÍA .....	23
2.2.- SOFTWARE LIBRE .....	25
2.2.1.- DISTRIBUCIÓN DE LINUX PARA MEJOR DESEMPEÑO.....	29
2.2.2.1.- MEJOR DISTRIBUCIÓN PARA ESCRITORIO .....	30
2.2.2.2.- MEJOR DISTRIBUCIÓN PARA PORTÁTILES .....	31
2.2.2.3.- MEJOR DISTRIBUCIÓN PARA EMPRESAS .....	32
2.2.2.4.- MEJOR DISTRIBUCIÓN PARA SERVIDOR.....	33
2.2.2.5.- MEJOR DISTRIBUCIÓN EN LIVE-CD.....	34
2.2.2.6.- MEJOR DISTRIBUCIÓN ORIENTADA A LA SEGURIDAD .....	35
2.2.2.7.- MEJOR DISTRIBUCIÓN MULTIMEDIA .....	35
2.3.- COMPATIBILIDAD DEL SISTEMA OPERATIVO .....	37
2.4.- CARACTERÍSTICAS DE LAS REDES INALÁMBRICAS.....	37
2.4.1.- ONDAS DE RADIO .....	38
2.4.2.- MICROONDAS TERRESTRES .....	38
2.4.3.1.- MICROONDAS POR SATÉLITE .....	39
2.4.3.2.- INFRARROJOS .....	40

2.5.- CLASIFICACIÓN DE REDES.....	41
2.5.1.- POR SU ALCANCE: .....	41
2.5.3.- APLICACIONES DE LAS REDES INALÁMBRICAS. ....	42
CAPÍTULO III .....	44
MICROCONTROLADOR 68HC08.....	44
3.1.- INTRODUCCION .....	44
3.1.2.- MODOS DE BAJO CONSUMO.....	45
3.2.- ARQUITECTURA DE LA CPU 68HC08.....	47
3.2.1.- TIEMPOS DE LOS CICLOS INTERNOS DE LA CPU 68HC08.....	49
3.2.2.- ORGANIZACIÓN DE LA MEMORIA DEL 68HC08 .....	50
3.2.3.- JUEGO DE INSTRUCCIONES DEL 68HC08.....	51
3.3.- MODOS DE DIRECCIONAMIENTO DEL 68HC05.....	53
3.3.1.- INHERENTE: CLRA .....	53
3.3.2.- INMEDIATO: LDA #20.....	54
3.3.3.- DIRECTO: LDA \$40 .....	54
3.3.4.- EXTENDIDO: LDA \$4000.....	54
3.3.5.- RELATIVO ( $\pm 128$ ) BLT LOOP .....	55
3.3.6.- INDEXADO SIN 'OFFSET' LDA, X.....	55
3.3.7.- INDEXADO CON 'OFFSET' DE 8 BITS LDA \$40, X .....	55
3.3.8.- INDEXADO CON 'OFFSET' DE 16 BITS LDA \$4000, X.....	55
3.4.- MODOS DE BAJO CONSUMO DEL 68HC08 (WAIT Y STOP) .....	55
3.4.1.- EFECTOS: .....	56
3.4.2.- MODO STOP: .....	56
3.4.3.- EJERCICIO PARA ENCONTRAR EL VALOR MÁXIMO EN UNA LISTA DE DIEZ BYTES. ....	56
3.5.- TIEMPOS DEL RESET INTERNO .....	58
3.5.1.- MANIPULACIÓN DE UNA EXCEPCIÓN.....	59
3.5.2.- DIAGRAMA DE BLOQUES DEL LVI.....	60
3.5.3.- REGISTRO DE ESTADO DEL LVI (LVISR).....	62
3.6.- FUNCIONAMIENTO DEL SPI .....	63
3.6.1.- CONEXIONES MASTER-SLAVE DEL SPI .....	64



3.6.2.- CÁLCULO DEL BAUD RATE .....	65
3.6.3.- EJEMPLO DE TRANSMISIÓN DE DATOS MASTER-SLAVE.....	67
3.6.4.- FORMATOS DE TRANSMISIÓN .....	69
3.7.- MÓDULO ADC (ANALOG TO DIGITAL CONVERTER) .....	71
3.7.1.- CARACTERÍSTICAS DEL ADC .....	71
3.7.1.1.- SELECCIÓN DEL CANAL DE ENTRADA .....	73
3.7.2.- CÁLCULO DEL TIEMPO DE UNA CONVERSIÓN .....	73
3.8.- CARACTERÍSTICAS DEL OSCILADOR A CRISTAL .....	74
3.8.1.- OSCILADOR EXTERNO .....	75
3.9.- CARACTERÍSTICAS DEL PLL .....	76
3.9.1.- CIRCUITO PLL.....	77
3.9.1.1.- FRECUENCIAS PRECALCULADAS .....	79
3.9.2.- DIAGRAMA DE FLUJO DE CONFIGURACIÓN DEL PLL.....	80
3.10.- MÓDULO SCI (SERIAL COMMUNICATIONS INTERFACE).....	81
3.10.1.- CARACTERÍSTICAS DEL MÓDULO SCI .....	81
3.10.2.- DOBLE BUFFER.....	84
3.10.3.- RECUPERACIÓN DE DATOS DEL RECEPTOR .....	86
3.10.4.- CÁLCULO DE LA VELOCIDAD DE TRANSMISIÓN DEL SCI .....	87
3.10.5.- REGISTRO SCBR DE VELOCIDAD DE TRANSMISIÓN .....	89
3.10.6.- CÁLCULO DEL BAUD RATE A 19200 .....	90
CAPITULO IV .....	91
PERIFERICOS Y BLUETOOTH .....	91
4.1.- PERIFERICOS INTERNOS.....	91
4.1.1.- UN SENSOR .....	91
4.1.1.1.- SENSORES SEGÚN EL TIPO DE SEÑAL DE ENTRADA.....	92
4.1.1.2.- ATENDIENDO AL TIPO DE SEÑAL ENTREGADA POR EL SENSOR .....	93
4.1.1.3.- ATENDIENDO A LA NATURALEZA DE LA SEÑAL ELÉCTRICA GENERADA .....	94
4.2.- ESTÁNDAR RS-232C .....	95
4.2.1.- PUERTOS SERIE MODERNOS.....	98
4.2.2.- TIPOS DE COMUNICACIÓN EN SERIE .....	98

<b>4.3.- TECNOLOGIA INALAMBRICA BLUETOOTH .....</b>	<b>99</b>
4.3.1.- INTRODUCCIÓN .....	99
4.3.2.- ESTÁNDAR 802.15.1 (BLUETOOTH) .....	101
4.3.2.1.- CAPA DE TRANSPORTE .....	102
4.3.2.2.- RADIO CAPA.....	102
4.2.1.3 BASEBAND Y LINK LAYERS .....	103
4.2.1.4 MIDDLEWARE CAPA .....	105
4.3.3.- TOPOLOGÍA DE RED BLUETOOTH .....	106
4.3.4.- SEGURIDAD DE BLUETOOTH.....	108
<b>CAPITULO V .....</b>	<b>111</b>
<b>IMPLEMENTACION DE SISTEMA TELEMETRICO CON BLUETOOTH .....</b>	<b>111</b>
5.1.1.- CONFIGURACIÓN DEL MODULO BLUETOOTH HC-06 .....	113
5.1.2.- DESCRIPCIÓN TÉCNICA DE BLUETOOTH .....	117
5.1.3.- COMANDOS ALD .....	120
5.2.- ENLACE BLUETOOTH .....	121
5.2.1.- TTL A RS232.....	124
5.3.- PRUEBAS Y APLICACIONES DEL MODULO BLUETOOTH .....	125
5.3.1.- PRUEBAS EN SOFTWARE LIBRE .....	126
5.4.- COMPROBACION DE LA HIPÓTESIS.....	128
<b>CONCLUSIONES</b>	
<b>RECOMENDACIONES</b>	
<b>RESUMEN</b>	
<b>SUMMARY</b>	
<b>BIBLIOGRAFÍA</b>	
<b>ANEXOS</b>	

## ÍNDICE DE FIGURAS

FIG. II.1.- LINUX MINT.....	30
FIG. II.2.- ESCRITORIO DISTRIBUCIÓN UBUNTU10.10 .....	31
FIG. II.3.- ESCRITORIO DE RED HAD. ....	32
FIG. II.4.- ESCRITORIO SUSE LINUX ENTERPRISE SERVER .....	33
FIG. II.5.- ESCRITORIO DE LIVE-CD .....	34
FIG. II.6.- ESCRITORIO LINUX MULTIMEDIA .....	36
FIG. II.7.- COMPATIBILIDAD DE SISTEMAS OPERATIVOS .....	37
FIG. II.8.- ENLACE DE MICROONDAS DE LINEA DE VISTA .....	39
FIG. II.9.- MICROONDAS POR SATÉLITE .....	40
FIG. II.10.- INFRARROJO.....	40
FIG. II.11.- REFERENCIAS DE FRECUENCIAS .....	41
FIG. III.12.- MICROPROCESADOR. ....	46
FIG. III.13.- ARQUITECTURA DE LA CPU 68HC08. ....	48
FIG. III.14.- TIEMPOS DE CICLOS INTERNOS DE LA CPU 68HC08 .....	49
FIG. III.15.- ORGANIZACIÓN DE MEMORIA DE 68HC08 .....	58
FIG. III.16.- TIEMPOS DE RESET INTERNO . ....	59
FIG. III.17.- MANIPULACIÓN DE UNA EXCEPCIÓN. ....	60
FIG. III.18.- DIAGRAMA DE BLOQUES DEL LVI .....	61
FIG. III.19.- FUNCIONAMIENTO DEL PSI .....	64
FIG. III.20.- CONEXIÓN DE MASTER-SLAVE DEL PSI .....	65
FIG. III.21.- TRANSMISIÓN DE DATOS MASTER-SLAVE .....	67
FIG. III.22.- FORMATO DE TRANSMISIÓN CPH=0 .....	69
FIG. III.23.- FORMATO DE TRANSMISIÓN CPH=1 .....	70
FIG. III.24.- MODULO ADC . ....	72
FIG. III.25.- CIRCUITO OSCILADOR A CRISTAL.....	75
FIG. III.26.- CIRCUITO OSCILADOR EXTERNO . ....	76
FIG. III.27.- CIRCUITO PPL .....	78
FIG. III.28.- MODULO SCI . ....	83
FIG. III.29.- DIAGRAMA DE BLOQUES DE TRANSMISOR SCI .....	85

FIG. III.30.- RECUPERACIÓN DE DATOS DEL RECEPTOR .	86
FIG. III.31.- DIAGRAMA DE BLOQUES DEL CÁLCULO DE VELOCIDAD DE SCI	88
FIG. IV.32.- SENSOR PASIVO DEL TIPO RESISTIVO	94
FIG. IV.33.- SENSOR TERMOELÉCTRICO .	95
FIG. IV.34.- CONECTOR DE DB-25 Y DB-9.	97
FIG. IV.35.- PILA DE PROTOCOLOS BLUETOOTH	102
FIG. IV.36.- SCATTERNET O RED DISPERSA .	107
FIG. V.37.- SISTEMA DE TELEMETRÍA POR BLUETOOTH	112
FIG. V.38.- MODULO BLUETOOTH HC-06.	113
FIG. V.39.- MODULO CONVERSOR DE SERIAL A USB QUE FUNCIONA CON NIVELES TTL	115
FIG. V.40.- CONECTIVIDAD AL MODULO, CONFIGURACIÓN DE BAUNDRATE Y CAMBIO DE NOMBRE	120
FIG. V.41.- ENLACE BLUETOOTH CON VEHÍCULO DE COMPETENCIA MAZDA 323.	121
FIG. V.42.- COMPROBACIÓN DE SEÑAL TRANSMISIÓN Y RECEPCIÓN EN LA ECU	122
FIG. V.43.- CIRCUITO DE TTL A RS232	124
FIG. V.44.- COMUNICACIÓN CON LA APLICACIÓN MEDIANTE CONEXIÓN BLUETOOTH	125
FIG. V.45.- PRUEBA DE CONECTIVIDAD CON EL SISTEMA BLUETOOTH	126
FIG. V.46.- PRUEBA EN SOFTWARE LIBRE	127

## ÍNDICE DE TABLAS

TABLA III.I.- TABLA LVISR:.....	62
TABLA III.II.- CÁLCULO DE BAUD RATE .....	66
TABLA. III.III.- CANALES DEL ADC DEL 68HC908GP32.....	73
TABLA III.IV.- FRECUENCIAS CALCULADAS .....	79
TABLA III.V.- REGISTRO DE SCBR DE VELOCIDAD DE TRANSMISIÓN .....	89
TABLA III.VI.- REGISTRO DE SCP1-SCP2 Y SCI .....	89
TABLA III.VII.- REGISTRO DE SCR2 A SCRO Y DIVISOR DE BAUD RATE .....	90
TABLA IV.VIII.- FUNCIONES DE RS232C .....	96
TABLA IV.IX.- PINES DE DB-25 Y DB-9.....	97
TABLA V.X.- COMANDOS PARA LA CONFIGURACIÓN DEL MODULO .....	116
TABLA V.XI.- REFERENCIA DE VOLTAJES MEDIDOS EN DB-09 DE LA ECU .....	124

## INTRODUCCIÓN

Las telecomunicaciones se han basado en un sin número de métodos, técnicas y tecnologías, para la transmisión de datos ya sean estos para ser almacenados o para ser visualizados en tiempo real, estos provienen de la obtención de información mediante una aplicación (software).

Hoy en día cuando la tecnología sigue avanzando en la vida moderna es primordial ocupar esta información en el momento oportuno para tomar decisiones, ya sea personal o laboral, cuyo entorno puede ser importante para el desarrollo o crecimiento de una determinada empresa u organización basada en la utilización de la información de datos específicos.

La telemetría es la medición de parámetros a una cierta distancia, transmisión y recepción automática de datos por: cables, radio u otros medios desde una fuente remota. Un sistema de telemetría básico que puede ser de forma simple, pero eficaz al presentar datos mediante conexión inalámbrica o cableada.

El Objeto de Observación, puede ser cualquier parámetro químico o físico, que son censados para obtener un nivel de voltaje, una variación de frecuencia u otro aspecto que cumpla algún estándar de comunicación.

Cuando se trata con señales eléctricas analógicas es necesario acondicionarlas y digitalizarlas para convertirlas en la entrada de un procesador y colocar los datos en el medio de transmisión. El objeto a ser observado es el desempeño del vehículo de competencia Mazda 323.

El proceso de Transmisión y Recepción de datos, para poder enviar y recibir datos a través de un medio de transmisión debe cumplir ciertas características del canal de transmisión, un ejemplo es la cantidad de información que se envía y que no exceda la capacidad de transmisión como el ancho de banda, información del destinatario o el método de envío.

El medio de transmisión, se lo realiza mediante cable o inalámbricamente, la cantidad de protocolos de comunicación que existen son diversas, y cada una de ellas se ha adaptado al medio de transmisión en el aire, por ello se ha logrado un sin número de sistemas muy complejos de codificación y comprensión haciendo el uso de protocolos robustos como el: 802.11x y 802.15.1 (Bluetooth).

La presentación de Datos, es el despliegue de la información contenida en el microcontrolador; MegaSquirt siendo un programa diseñado para testear los controladores electrónicos de inyección programable del vehículo como: Temperatura de motor, Revoluciones del motor, Suministro de combustible a los inyectores, Temperatura del ambiente, entre otros. Estos controles son revisados de manera eficiente para transmitir y recibir datos de la ECU hacia una estación central o viceversa, por medio del módulo bluetooth; presentando los datos de una manera visual en una aplicación llamada MegaTune en Windows y MegaTunix en Linux.

## **CAPÍTULO I**

### **MARCO REFERENCIAL**

#### **1.1.- ANTECEDENTES**

Los sistemas embebidos a pesar de no ser muy nombrados están en muchas partes, en realidad; es difícil encontrar algún dispositivo cuyo funcionamiento no esté basado en algún sistema embebido, desde automotores más sencillos, hasta los más modernos.

Los sistemas embebidos suelen tener en una de sus partes una computadora con características especializadas conocidas como microcontroladores que viene a ser el cerebro del sistema. Este no es más que un microprocesador que incluye interfaces de entrada/salida en el mismo chip. Normalmente estos sistemas poseen una interfaz externa para efectuar un monitoreo del estado y hacer un diagnostico del sistema.



La telemetría y telemando permite el monitoreo y control de procesos por medio de un sistema de telecomunicaciones. Cuando existe un sistema que se encuentra localizada a cortas y largas distancias, es necesario un método que permita obtener información de variables, de manera mas rápida para poder llevar a cabo una acción que en determinados casos pueden ser de vital importancia para los proceso de generación de información en cualquier vehículo que este en movimiento o estático.

En el caso convencional en el que la persona tenga que realizar las acciones de obtención de información mediante variables y regular los procesos en lugares ubicados remotamente, se pueden encontrar varias dificultades como son: la ubicación topográfica del lugar, el costo que implica obtener estas variables, la rapidez con que se necesita procesar las variables y llevar a cabo una acción; es por ello que la telemetría y el telemando es una solución a estas dificultades, siendo importante para el correcto funcionamiento de estos sistemas.

La Telemetría consiste en la adquisición de datos de cualquier índole a distancias, mediante sensores o transductores ya sean estos analógicos o digitales y enviarlos a una estación de control a través de un sistema de telecomunicaciones donde es administrado, procesado y visualizado. Esta comunicación se realiza por medio de módems, routers, módulos, que acondicionan las señales informativas de acuerdo al medio en el que se realiza la comunicación.

El Telemando consiste en el uso remoto de dispositivos actuadores que responden ante las ordenes enviadas por la central de control con el fin de poder realizar procesos como encender/apagar sensores, aumentar/disminuir cantidad de fluidos como gasolina, aceite, entre otros.

Las aplicaciones de la telemetría y el telemando puede relacionarse a varios tipos de sistemas con fines científicos como por ejemplo en la industria aeroespacial, en el control de naves no tripuladas, también en las agencias espaciales como la Nasa para operar naves espaciales y satélites, en la perforación de pozos petrolíferos donde se obtienen datos telemáticos a través del lado de perforación.

Las aplicaciones también pueden tener fines no científicos como por ejemplo en la Formula 1, moto GT; donde se deben leer datos sobre el estado de los vehículos, así como otras aplicaciones importantes que se da en la empresa que brinda servicios básicos como: luz, agua y teléfono.

La eficiencia tanto de la telemetría, dependen en gran medida del sistema de comunicaciones que se utiliza, siendo este el responsable de la velocidad con que llegue la información enviado desde la central de control hasta el dispositivo ubicado a cierta distancia.

## 1.2.- JUSTIFICACIÓN

El deporte automovilístico de rally ecuatoriano presenta de una manera preliminar algunas dificultades como: deficiencia de conocimientos de sistemas programables, altos costos de implementación, necesidad de modificaciones y la ausencia de parámetros en tiempo real, para la puesta a punto teniendo que manejar dispositivos estáticos dentro del auto de una manera no muy eficiente. Lo que no sucede en países desarrollados donde los vehículos de competencia tienen instalados estos dispositivos lo que eleva su costo por la tecnología de punta desarrollada por las grandes corporaciones multinacionales dedicadas al diseño de estos sistemas.

El manejo e implementación de un sistema electrónico programable como un marco general involucra: tener conocimiento básico de sensores automotrices, actuadores eléctricos, disponer de herramienta adecuada para mecanizar y adaptar los elementos que requiera el sistema, mantener instrumentación que muestren lecturas en tiempo real lo que se considera un problema grave, que repercute en costos para las personas que preparan sus autos de competición generándose el rechazo de esta tecnología sin conocer el gran beneficio que ofrece los sistemas electrónicos programables.

Uno de los argumentos para mejorar el tiempo que demora en conocer e identificar el problema que tiene el vehículo es; verificar los datos que contiene los dispositivos internos instalados en el mismo, para mejorar este inconveniente es necesario tener a disposición los datos en tiempo real sin

conexión de cables para la rápida intervención del sistema interno, para esto se utilizará una tecnología inalámbrica capaz de interpretar de una manera más rápida los datos internos para una breve lectura y análisis de los mismos.

En vista de la necesidad de implementar un sistema en nuestro medio, que ayude a un vehículo de competencia (Mazda 323) a monitorear los dispositivos internos para un mejor desempeño, se propone la implementación de un sistema de obtención de datos por telemetría con la ayuda de la tecnología inalámbrica Bluetooth, donde se pretende obtener datos en tiempo real del desempeño del vehículo; para identificar los problemas del automotor o de algún dispositivo instalado en el mismo. Un valor agregado a este sistema será la utilización de software libre (Linux) para la monitorización de elementos básicos de rendimiento como los sensores eléctricos, y otros dispositivos internos.

### **1.3.- OBJETIVOS**

#### ***1.3.1.- Objetivo General:***

“Diseñar e implementar un sistema de transferencia de datos por telemetría para un vehículo de competencia con el uso tecnología inalámbrica Bluetooth”.

#### ***1.3.2.- Objetivos Específicos:***

- Desarrollar un sistema mediante software libre para transferir datos telemétricos desde el vehículo hasta la estación central.

- Obtener datos por medio de la tecnología inalámbrica Bluetooth como medio de comunicación entre el vehículo y la estación central.
- Realizar un escenario de pruebas que permita determinar de forma cualitativa el rendimiento y desempeño del vehículo con telemetría.
- Monitorear en tiempo real del desempeño del vehículo; para identificar los problemas del automotor o de algún dispositivo instalado en el mismo.

#### **1.4.-HIPÓTESIS**

*Mediante el diseño e implementación de un sistema de transferencia de datos por telemetría para el vehículo de competencia Mazda 323, con el uso tecnología inalámbrica Bluetooth, se pretende monitorear en tiempo real el desempeño y rendimiento cualitativo del mismo y corregir problemas en los dispositivos eléctrico/mecánico para un mejor performance del vehículo dentro de una competencia.*

## **CAPÍTULO II**

### **TELEMETRIA Y SOFTWARE LIBRE**

#### **2.1- TELEMETRIA**

##### ***2.1.1.- Definición de Telemetría***

Es el arte de medir distancias entre objetos lejanos a los que no se puede o no se quiere ir. Técnica de medir distancias entre objetos lejanos mediante un telémetro, los topógrafos se valen de la telemetría para sus mediciones.

Sistema de medida de magnitudes físicas en lugares difícilmente accesibles, que permite transmitir el resultado de la medición a un observador lejano: la telemetría permite controlar reactores nucleares. La telemetría es una

tecnología que permite la medición remota de magnitudes físicas y el posterior envío de la información hacia el operador del sistema. Se utilizó por primera vez en 1915, a mediados de la primera guerra mundial, por el alemán Khris Osterhein y el italiano Francisco Javier Sosao para medir a qué distancia se encontraban objetivos de artillería.

La palabra telemetría procede de las palabras griegas  $\tau\eta\lambda\epsilon$  (tele), que quiere decir a distancia, y la palabra  $\mu\epsilon\tau\rho\nu$  (metrón), que quiere decir medida.

El envío de información hacia el operador en un sistema de telemetría se realiza típicamente mediante comunicación inalámbrica, aunque también se puede realizar por otros medios (teléfono, redes de ordenadores, enlace de fibra óptica, etcétera). Los sistemas de telemetría reciben las instrucciones y los datos necesarios para operar mediante desde el Centro de Control.

### ***2.1.2.- Aplicaciones de telemetría***

La telemetría se utiliza en grandes sistemas, tales como naves espaciales, plantas químicas, redes de suministro eléctrico, redes de suministro de gas entre otras empresas de provisión de servicios públicos, debido a que facilita la monitorización automática y el registro de las mediciones, así como el envío de alertas o alarmas al centro de control, con el fin de que el funcionamiento sea seguro y eficiente. Por ejemplo, las agencias espaciales como la NASA, la Q.K, la ESA y otras, utilizan sistemas de telemetría y de telecontrol para operar con naves espaciales y satélites.

La Telemetría se utiliza en infinidad de campos, tales como la exploración científica con naves tripuladas o no (submarinos, aviones de reconocimiento y satélites), diversos tipos de competición (por ejemplo, Fórmula 1 y Moto GP), o la operación de modelos matemáticos destinados a dar sustento a la operación de embalses.

En las fábricas, oficinas y residencias, el monitoreo del uso de energía de cada sección o equipo y los fenómenos derivados (como la temperatura) en un punto de control por telemetría facilita la coordinación para un uso más eficiente de la energía.

Una aplicación muy importante de la telemetría es la perforación de pozos petrolíferos; ésta se utiliza para la medición con herramientas navegables MWD1y LWD2. Se utiliza básicamente la telemetría de pulso de lodo, que se transmite a través de la tubería de perforación por medio del lodo de perforación.

- Estaciones de bombeo de agua.
- Estaciones de bombeo de ductos y oleoductos
- Estaciones de bombeo y calentamiento de hidrocarburo.
- Tanques de almacenamiento en instalaciones petroleras.
- Monitoreo de pozos petroleros en tierra y mar.
- Estaciones de bombeo de gas.
- Monitoreo de consumo eléctrico.



- Monitoreo de vibración y temperatura en motores, compresores o cualquier equipo crítico de combustión interna o eléctrica.

## **2.2.- SOFTWARE LIBRE**

El software libre (en inglés free software, aunque esta denominación también se confunde a veces con "gratis" por la ambigüedad del término "free" en el idioma inglés, por lo que también se usa "libre software" y " logical libre") es la denominación del software que respeta la libertad de todos los usuarios que adquirieron el producto y, por tanto, una vez obtenido puede ser usado, copiado, estudiado, modificado, y redistribuido libremente de varias formas. Según la Free Software Foundation, el software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, modificar el software y distribuirlo modificado.

El software libre suele estar disponible gratuitamente, o al precio de costo de la distribución a través de otros medios; sin embargo no es obligatorio que sea así, por lo tanto no hay que asociar software libre a "software gratuito" (denominado usualmente freeware), ya que, conservando su carácter de libre, puede ser distribuido comercialmente ("software comercial"). Análogamente, el "software gratis" o "gratuito" incluye en ocasiones el código fuente; no obstante, este tipo de software no es libre en el mismo sentido que el software libre, a menos que se garanticen los derechos de modificación y redistribución de dichas versiones modificadas del programa.

El ``Software Libre`` es un asunto de libertad, no de precio. Para entender el concepto, debes pensar en ``libre`` como en ``libertad de expresión``, no como en ``aplicaciones gratis`` [En inglés una misma palabra (free) significa tanto libre como gratis, lo que ha dado lugar a cierta confusión].

``Software Libre`` se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso, se refiere a cuatro libertades de los usuarios del software:

- La libertad de usar el programa, con cualquier propósito (libertad 0).
- La libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades (libertad 1). El acceso al código fuente es una condición previa para esto.
- La libertad de distribuir copias, con lo que puedes ayudar a tu vecino (libertad 2).
- La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. (libertad 3). El acceso al código fuente es un requisito previo para esto.

Un programa es software libre si los usuarios tienen todas estas libertades. Así pues, deberías tener la libertad de distribuir copias, sea con o sin modificaciones, sea gratis o cobrando una cantidad por la distribución, a cualquiera y a cualquier lugar. El ser libre de hacer esto significa (entre otras cosas) que no tienes que pedir o pagar permisos.

También deberías tener la libertad de hacer modificaciones y utilizarlas de manera privada en tu trabajo u ocio, sin ni siquiera tener que anunciar que dichas modificaciones existen. Si publicas tus cambios, no tienes por qué avisar a nadie en particular, ni de ninguna manera en particular.

La libertad para usar un programa significa la libertad para cualquier persona u organización de usarlo en cualquier tipo de sistema informático, para cualquier clase de trabajo, y sin tener obligación de comunicárselo al desarrollador o a alguna otra entidad específica.

La libertad de distribuir copias debe incluir tanto las formas binarias o ejecutables del programa como su código fuente, sean versiones modificadas o sin modificar (distribuir programas de modo ejecutable es necesario para que los sistemas operativos libres sean fáciles de instalar). Está bien si no hay manera de producir un binario o ejecutable de un programa concreto (ya que algunos lenguajes no tienen esta capacidad), pero debes tener la libertad de distribuir estos formatos si encontraras o desarrollaras la manera de crearlos.

Para que las libertades de hacer modificaciones y de publicar versiones mejoradas tengan sentido, debes tener acceso al código fuente del programa. Por lo tanto, la posibilidad de acceder al código fuente es una condición necesaria para el software libre.

Para que estas libertades sean reales, deben ser irrevocables mientras no hagas nada incorrecto; si el desarrollador del software tiene el poder de revocar la licencia aunque no le hayas dado motivos, el software no es libre.

Son aceptables, sin embargo, ciertos tipos de reglas sobre la manera de distribuir software libre, mientras no entren en conflicto con las libertades centrales. Por ejemplo, copyleft [‘izquierdo de copia’] (expresado muy simplemente) es la regla que implica que, cuando se redistribuya el programa, no se pueden agregar restricciones para denegar a otras personas las libertades centrales. Esta regla no entra en conflicto con las libertades centrales, sino que más bien las protege.

Así pues, quizás hayas pagado para obtener copias de software GNU, o tal vez las hayas obtenido sin ningún coste. Pero independientemente de cómo hayas conseguido tus copias, siempre tienes la libertad de copiar y modificar el software, e incluso de vender copias.

‘Software libre’ no significa ‘no comercial’. Un programa libre debe estar disponible para uso comercial, desarrollo comercial y distribución comercial. El desarrollo comercial del software libre ha dejado de ser inusual; el software comercial libre es muy importante.

Es aceptable que haya reglas acerca de cómo empaquetar una versión modificada, siempre que no bloqueen a consecuencia de ello tu libertad de publicar versiones modificadas. Reglas como ‘Si haces disponible el programa de esta manera, debes hacerlo disponible también de esta otra’ pueden ser igualmente aceptables, bajo la misma condición. (Observa que una regla así todavía te deja decidir si publicar o no el programa). También es aceptable que la licencia requiera que, si has distribuido una versión modificada y el desarrollador anterior te pide una copia de ella, debas enviársela.

### **2.2.1.- Distribución de Linux para mejor desempeño**

Es el misterio de los misterios, la pregunta que se hace el usuario que quiere interactuar con el sistema operativo es, ¿cual es la mejor distribución de Linux de todas?

Es una pregunta constante en los usuarios nuevos de Linux; la clave, por supuesto, está en entender que la pregunta es en sí es errónea, no importa qué es lo que te digan, no hay una única distribución verdadera y perfecta. La realidad es más compleja: Depende de tus necesidades, y cual es la distribución de GNU/Linux que mejor se adapta a ellas.

Para ayudar a los neófitos a tratar de descubrir cual es la mejor distribución, recomendaremos una lista para los distintos tipos de usuarios, basada en experiencias y comprobaciones de otros usuarios. Las categorías que se subdividirán serán:

- Mejor distribución para escritorio
- Mejor distribución para portátil
- Mejor distribución para empresas
- Mejor distribución para servidor
- Mejor distribución en Live-CD
- Mejor distribución destinada a la seguridad
- Mejor distribución multimedia

### 2.2.2.1.- Mejor distribución para escritorio

Empecemos, con la categoría más “conflictiva”: la mejor distribución para uso general en un PC de escritorio. A lo largo de los años, este título ha recaído en Ubuntu, después en Fedora. En estos últimos años el título recae en un derivado de Ubuntu, aunque no directamente.

Para el 2012 el título al mejor GNU/Linux para escritorios recae en Linux Mint en concreto en Linux Mint 12 “Lisa.”



Figura II.1.- Linux Mint

Fuente: <http://www.linuxmint.com/>

Linux Mint es un derivado de Ubuntu, lo que significa, para aquellos que no están familiarizados con el lenguaje del software libre y abierto, es que el código de Linux Mint está basado en el de Ubuntu. Los desarrolladores de Linux Mint añadieron a ese sus propias contribuciones y crearon así esta nueva distribución.

Debido a la cercanía del proyecto Ubuntu, y por ende a Debían (ya que tiene una misma raíz, Debían) los usuarios de Linux Mint tienen acceso a una gran cantidad de paquetes, repositorios de software que incluyen aplicaciones, temas y widgets, que hacen del uso de Linux Mint una experiencia agradable.

### 2.2.2.2.- Mejor distribución para portátiles

Los entornos para portátiles no son equiparables a los entornos para escritorio. Aunque algunos portátiles todavía mantienen anticuados diseños con grandes pantallas que tienen sus teclados pegados a ellos, como el creador de Linux Linus Trovarles se lamentó en una entrevista ,algunos portátiles están gradualmente fijándose en los diseños de los Apple Macbook Air, y se está empezando a dar pasos, así Ubuntu Desktop sigue esa línea.

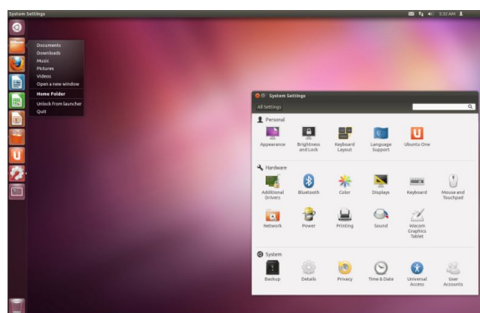


Figura II.2.- Escritorio distribución Ubuntu 10.10

Fuente: <http://www.ubuntu.com/>

En esta clasificación aconsejamos una distribución con una interfaz que se asemeja a estas nuevas líneas y entornos como es Ubuntu 12.04 LTS.

El último lanzamiento de Ubuntu tiene una excelente gestión del rendimiento de energía, y muchos drivers. Y, aunque es posible también encuadrarlo dentro de un escritorio tradicional, hay que reconocer el hecho de que la interfaz de Ubuntu es perfecta para un dispositivo móvil que necesita una capacidad de uso más racionalizado. Como una distribución portátil, Ubuntu sigue siendo la el líder.

### 2.2.2.3.- Mejor distribución para empresas

En los 2 años anteriores este título recayó en SUSE Linux Enterprise Desktop (SLED). Sin embargo, este año, el visto bueno tendrá que ir a Red Hat Enterprise Linux Desktop 6 (RHEL6).

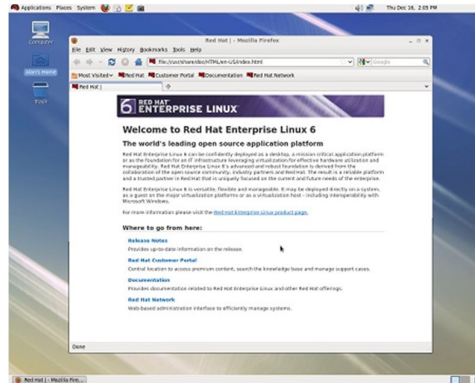


Fig. II.3.- Escritorio de Red Hat

Fuente: <http://www.redhat.com/>

¿Por qué el cambio? En el pasado, SLED mantiene una ligera ventaja sobre RHEL6 debido a las capacidades de SUSE Studio y de la fuerte comunidad de open SUSE. Pero con el paso del tiempo, la compañía SUSE Linux parece estar adoptando una postura mucho más tranquila debido a la ventaja de su mercado, mientras que Red Hat está haciendo grandes avances en nuevos territorios dentro de la virtualización y la nueva tecnología en la nube.

Quizás es un poco prematuro juzgar así, pero parece que SLED está quedándose un poco rezagada en cuanto a lo que puede ofrecer a las empresas que usan Linux, y en el mundo empresarial eso es todo lo que un proveedor de Linux puede hacer. Este año, el título va para Red Hat.



#### 2.2.2.4.- Mejor distribución para servidor

Durante los últimos dos años, esta categoría siempre se ha reducido a dos contendientes principales: Red Hat Enterprise Linux (RHEL) y SUSE Linux Enterprise Server (SLES).

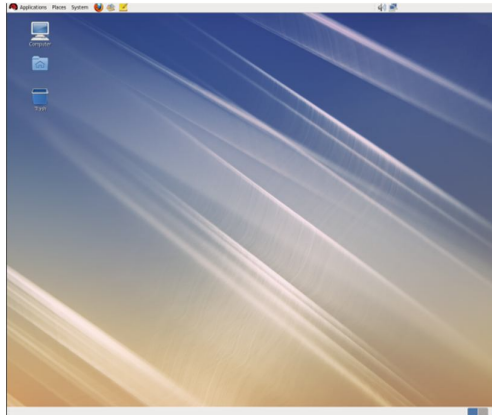


Fig. II.4.- Escritorio SUSE Linux Enterprise Server

Fuente: <https://www.suse.com/>

La lista es extensa; mientras que SUSE Linux se ha concentrado en las mejoras como el factor principal de lo que ofrece SLES, Red Hat ha estado haciendo eso y adentrándose a todos esos nuevos mercados mencionados anteriormente.

En el año 2012, tener una presencia activa y estable en la nube. Se puede pensar que esto es un tanto exagerado, pero la verdad es que la mayoría de las empresas se van a implementar en una especie de nube pública, privada o híbrida. Si no es ahora, pronto. Hasta que venga algo mejor, tiene sentido seguir ese camino. Red Hat Enterprise Linux 6 tiene este tipo de presencia en

la nube, además de todo el apoyo ISV y tecnología que han liderado durante años.

#### **2.2.2.5.- Mejor distribución en Live-CD**

Actualmente cada distribución de Linux es en realidad un live-CD. Se puede poner un CD o DVD con la ISO quemada de tu distribución favorita y arrancar desde la unidad lectora una completa distribución GNU/Linux sin necesidad de instalar nada en tu PC.

Pero existen un tipo de distribuciones específicamente hechas para correr únicamente como Live-CD. Estas distribuciones ofrecen herramientas, y utilidades de software de uso específica para reparar instalaciones ya existentes en el PC de otras distribuciones GNU/Linux e incluso sirven para Windows.



Fig. II.5.- Escritorio de Live-CD

Fuente: [www.somoslibres.org](http://www.somoslibres.org)

Estas distribuciones son muy prácticas independientemente de la distribución GNU/Linux que se use porque ante una crisis o problema, realizan una gran labor de rescate del sistema.

Una vez más el líder indiscutible de todas es KNOPPIX. Basada en Debian usa un escritorio ligero como es LXDE para manejar una robusta herramienta. Casi demasiado completa para ser simplemente una distribución en Live-CD, pero sigue manteniendo todo su potencial y velocidad.

#### **2.2.2.6.- Mejor distribución orientada a la seguridad**

La decisión de este año en la mejor distribución orientada a la seguridad ha sido un tanto difícil. En el pasado Back Track Linux era claramente líder, pero en Back Track Linux 5, fue descubierto recientemente un agujero de seguridad que la hacía vulnerable, y que conmocionó las expectativas de muchos profesionales en la materia de la seguridad informática. [Actualización: El agujero reside en la aplicación wicd que fue eliminado de BackTrack]

El problema se solucionó, naturalmente, pero el daño a BacTrack ya estaba hecho.

Es hora de resaltar lo bueno: todos pueden sufrir errores como este, y lo cierto es que a pesar de la vulnerabilidad, Back Track sigue ofreciendo la mejor serie de herramientas de seguridad de sombrero blanco para examinar su sistema o los problemas de la red.

#### **2.2.2.7.- Mejor distribución multimedia**

Cambiar entre distribuciones debido a la falta de compatibilidad con el formato para reproducir música o películas. Fue un verdadero problema, hace mucho tiempo.



Fig. II.6.- Escritorio Linux multimedia

Fuente: <http://victorhckinthefreeworld.wordpress.com/2012/05/08/las-7-mejores-distribuciones-de-gnulinix-del-2012/>

Ahora, sin embargo, esos días han quedado atrás, y los consumidores multimedia pueden utilizar cualquier distribución de Linux. Sin embargo, con la disponibilidad de cámaras de alta calidad y grabadoras de vídeo, las distribuciones de Linux tienen que incluir herramientas poderosas para la edición de audio y de vídeo, no sólo de consumo.

Por tercer año consecutivo, la mejor distribución en esta clase sigue siendo de Ubuntu Studio. Para la producción de audio, vídeo y gráfico, ya que contiene un conjunto muy completo de herramientas, así como compatibilidad con formatos y codecs para una amplia gama de formatos multimedia.

## 2.3.- COMPATIBILIDAD DEL SISTEMA OPERATIVO



Fig. 11.7.- Compatibilidad de Sistemas Operativos

Fuente: <http://www.fosforito.net/>

La compatibilidad es la condición que hace que un programa y un sistema, arquitectura o aplicación logren comprenderse correctamente tanto directamente o indirectamente (mediante un algoritmo). A este algoritmo que hace que un programa logre ser comprendido por un sistema, arquitectura o aplicación se lo denomina emulador por el hecho de que es un intérprete entre el programa y el sistema, arquitectura o aplicación.

No hay un Sistema Operativo mejor que otro. Depende del gusto y de los usos para el cual se quiere el Sistema Operativo.

## 2.4.- CARACTERÍSTICAS DE LAS REDES INALÁMBRICAS.

Las redes inalámbricas poseen ciertas características según el rango de frecuencias utilizado para transmitir, el medio de transmisión pueden ser ondas de radio, las microondas terrestres o satélite, y los infrarrojos. Dependiendo de estos medios, la red inalámbrica tendrá unas características u otras:

#### **2.4.1.- Ondas de Radio**

Las ondas electromagnéticas (Combinación de campos eléctricos y magnéticos oscilantes, que se propagan a través del espacio transportando energía de un lugar a otro) son omnidireccionales, así que no son necesarias antenas parabólicas. La transmisión no es sensible a las atenuaciones producidas por la lluvia, ya que se opera a frecuencias no demasiado elevadas. En rango se encuentran las bandas desde la ELF (Extremely Low Frequency) comprendida entre los 3 y los 30 Hz, hasta la UHF (Ultra High Frequency) que va de los 300 a los 3000 Hz, esto quiere decir que comprende el espectro radioeléctrico de 30 - 3000000000 Hz.

#### **2.4.2.- Microondas Terrestres**

Estas se utilizan en antenas parabólicas con un diámetro aproximado de unos tres metros. Tienen una cobertura de kilómetros, pero con el inconveniente de que el emisor y el receptor deben estar perfectamente alineados. Por eso se les denomina enlaces punto a punto en distancias cortas. En este caso, la atenuación producida por la lluvia es más importante ya que se opera a una frecuencia más elevada. Las microondas comprenden frecuencias desde 1 hasta 300 GHz.

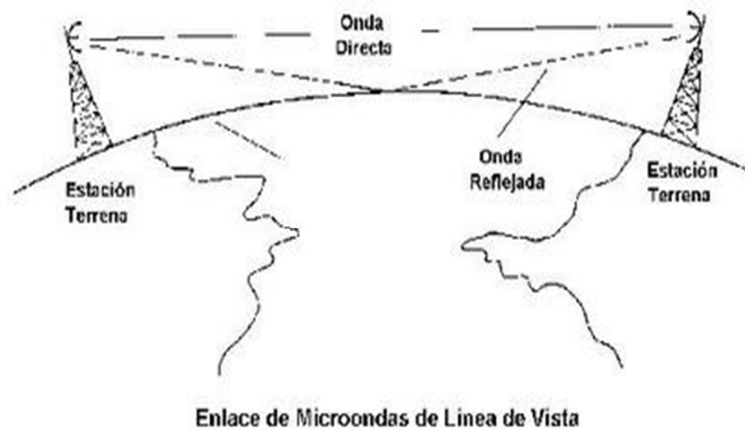


Fig. 11.8.-Enlace de Microondas de Línea de Vista

Fuente:<http://redesc.wikispaces.com/Caracter%C3%ADsticas+y+aplicaciones+de+las+redes+internacionales%C3%A1mblicas>.

#### 2.4.3.1.- Microondas por Satélite

Se hacen enlaces de dos o más estaciones terrestres que se les denominan estaciones base. El satélite recibe la señal (Denominada señal ascendente) en una banda de frecuencia, la amplifica y la retransmite en otra banda (Señal descendente). Cada satélite opera en unas bandas concretas. Las fronteras frecuencia les de las microondas, tanto terrestres como por satélite, con los infrarrojos y las ondas de radio de alta frecuencia se mezclan bastante, así que puede haber interferencias con las comunicaciones en determinadas frecuencias.

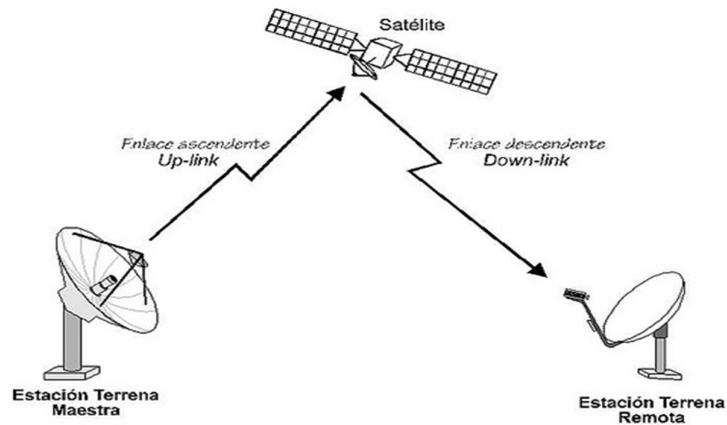


Fig. II.9.- Microondas por satélite

Fuente: <http://redesc.wikispaces.com/Caracter%C3%ADsticas+y+aplicaciones+de+las+redes+inal%C3%A1mbricas>

### 2.4.3.2.- Infrarrojos

Se enlazan transmisores y receptores que modulan la luz infrarroja no coherente. Deben estar alineados directamente con una reflexión en una superficie. No pueden atravesar las paredes. Los infrarrojos van desde 300 GHz hasta 384 THz.

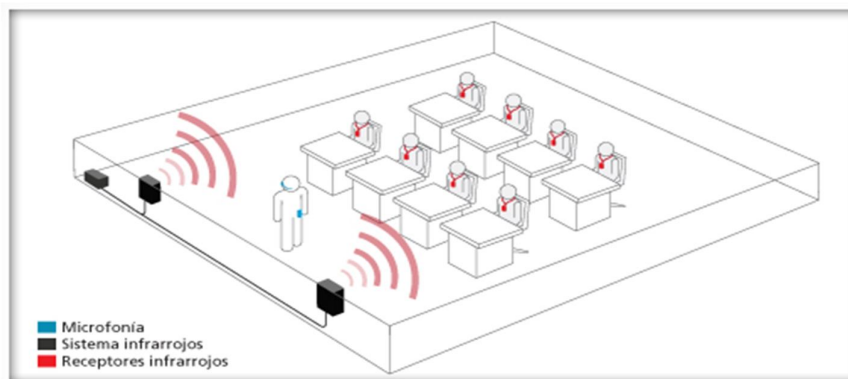


Fig. II.10.- Infrarrojo

Fuente: <http://redesc.wikispaces.com/Caracter%C3%ADsticas+y+aplicaciones+de+las+redes+inal%C3%A1mbricas> Infrarrojos



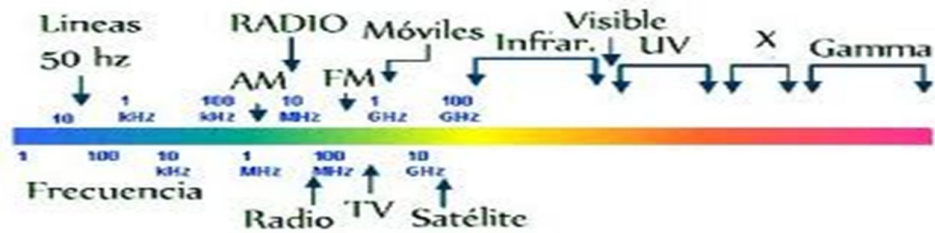


Fig. II.11.- Referencias de frecuencias

Fuente: <http://redesc.wikispaces.com/Caracter%C3%ADsticas+y+aplicaciones+de+las+redes+inal%C3%A1mbricas> Frecuencias de los distintos dispositivos

## 2.5.- CLASIFICACIÓN DE REDES

### 2.5.1.- Por su alcance:

- Red de área personal (PAN)
- Red de área local (LAN)
- Red de área de campus(CAN)
- Red de área metropolitana (MAN)
- Red de área amplia (WAN)
- Red de área simple (SPL)
- Red de área de almacenamiento (SAN)

### 2.5.2.- Por método de conexión:

- Medios guiados: cable coaxial, cable de par trenzado, fibra óptica y otros tipos de cables
- Medios no guiados: radio, infrarrojo, laser, microondas, y otras redes inalámbricas

### **2.5.3.- Aplicaciones de las redes inalámbricas.**

Existen diferentes tipos de aplicaciones para las redes inalámbricas, como por ejemplo;

- Las bandas más importantes con aplicaciones inalámbricas, del rango de frecuencias que abarcan las ondas de radio, son la VLF (Comunicación en submarinos), LF (Radio AM de onda larga), MF (Radio AM de onda media), HF (Radio AM de onda corta), VHF (Radio FM y TV), UHF (TV).
- Mediante las microondas terrestres, existen diferentes aplicaciones basadas en protocolos como Bluetooth o ZigBee para interconectar ordenadores portátiles, PDAs, teléfonos u otros aparatos. También se utilizan las microondas para comunicaciones con radares (Detección de velocidad) y para la televisión digital terrestre.
- Las microondas por satélite se usan para la difusión de televisión por satélite, transmisión telefónica a larga distancia y en redes privadas, por ejemplo.
- Los infrarrojos tienen aplicaciones como la comunicación a corta distancia de los ordenadores con sus periféricos. También se utilizan para mandos a distancia, ya que así no interfieren con otras señales electromagnéticas, por ejemplo la señal de televisión. Uno de los estándares más usados en estas comunicaciones es el IrDA (Infrared Data Association). Otros usos que tienen los infrarrojos son técnicas

como la termografía, la cual permite determinar la temperatura de objetos a distancia.

## **CAPÍTULO III**

### **MICROCONTROLADOR 68HC08**

#### **3.1.- INTRODUCCION**

Un microcontrolador (abreviado  $\mu\text{C}$ , UC o MCU) es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica. Un micro controlador incluye en su interior las tres principales unidades funcionales de una computadora: unidad central de procesamiento, memoria y periféricos de entrada/salida.

Algunos microcontroladores pueden utilizar palabras de cuatro bits y funcionan a velocidad de reloj con frecuencias tan bajas como 4 kHz, con un consumo de baja potencia (mW o microvatios). Por lo general, tendrá la capacidad para mantener la funcionalidad a la espera de un evento como pulsar un botón o de

otra interrupción, el consumo de energía durante el sueño (reloj de la CPU y los periférico de la mayoría), lo que hace que muchos de ellos muy adecuados para aplicaciones con batería de larga duración. Otros microcontroladores pueden servir para roles de rendimiento crítico, donde sea necesario actuar más como un procesador digital de señal (DSP), con velocidades de reloj y consumo de energía más altos.

Al ser fabricados, la memoria ROM del microcontrolador no posee datos. Para que pueda controlar algún proceso es necesario generar o crear y luego grabar en la EEPROM o equivalente del microcontrolador algún programa, el cual puede ser escrito en lenguaje ensamblador u otro lenguaje para microcontroladores; sin embargo, para que el programa pueda ser grabado en la memoria del microcontrolador.

Debe ser codificado en sistema numérico hexadecimal que es finalmente el sistema que hace trabajar al microcontrolador cuando éste es alimentado con el voltaje adecuado y asociado a dispositivos analógicos y discretos para su funcionamiento

### ***3.1.2.- Modos de bajo consumo***

Los microcontroladores avanzados ofrecen ventajas a las aplicaciones gracias a su bajo consumo de energía.

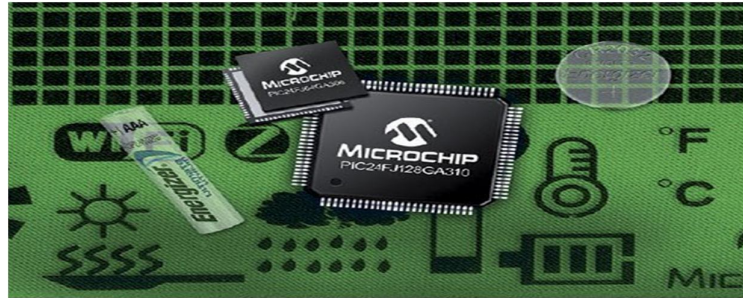


Figura III.12.- Microprocesador

Fuente: <http://entrnrnadorpic16f84.files.wordpress.com>

El bajo consumo se considera a menudo un aspecto fundamental en un producto ecológico, pero la naturaleza del bajo consumo rara vez se detalla o se cuantifica.

Los requisitos del microcontrolador de bajo consumo variarán dependiendo de la aplicación y de cómo se utilizará el microcontrolador en la aplicación. Su uso puede clasificarse en tres áreas principales:

- Modo de mínimo consumo – Se usará en aplicaciones como termostatos alimentados mediante batería. El modo de mínimo consumo define el nivel más bajo de energía disponible para controlar el visualizador LCD. Esta reducción en el consumo de energía permite prolongar la vida de la batería.
- Corriente activa consumida – Para aplicaciones como un contador de electricidad, el nivel y la naturaleza del bajo consumo se refiere a la corriente activa consumida por el sistema durante su funcionamiento.
- Aplicaciones determinadas por el tiempo – Hay sistemas que exigen mantener la fecha y la hora, con independencia de que haya una fuente

de alimentación principal del sistema, como un contador de electricidad si hay un fallo del suministro.

Al diversificarse los requisitos de las aplicaciones, los diseñadores buscan microcontroladores con modos de consumo más flexibles para adaptarse lo máximo posible al funcionamiento del sistema.

En el pasado, los microcontroladores tenían un modo activo para el funcionamiento del dispositivo; modos Idle (parado) o Doze (adormecido) para reducir o eliminar el consumo de la CPU en conmutación CPU mientras permite el funcionamiento de los periféricos; y el modo Sleep (dormido) que permite un funcionamiento limitado de los periféricos con un mínimo consumo de energía.

### **3.2.- ARQUITECTURA DE LA CPU 68HC08**

El modelo de programación de la CPU del 68HC08 es una evolución del modelo de CPU del 68HC05. Cada 68HC08 implementa este modelo de CPU sin tener en cuenta el tamaño o el conjunto de características propias. A continuación se verán las mejoras específicas que se incluyen en la arquitectura 68HC08:

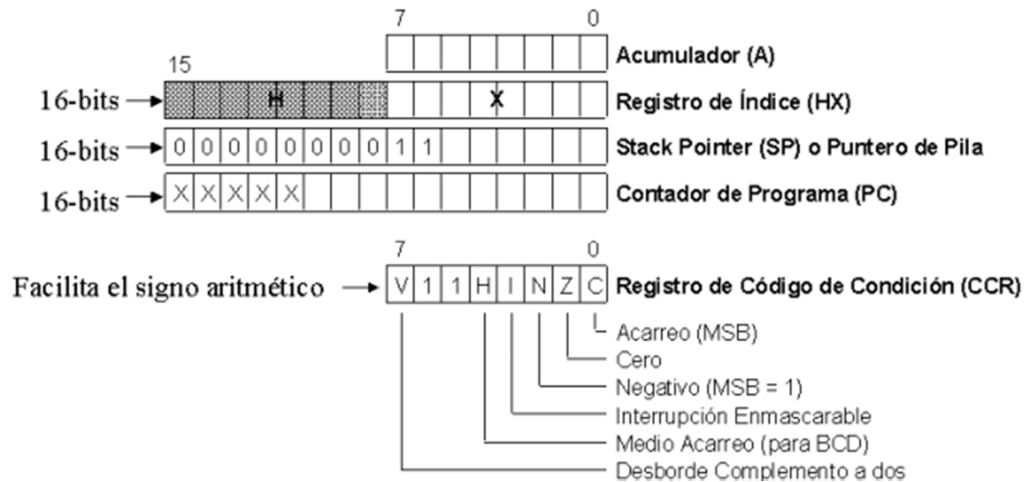


Figura III.13.- Arquitectura de la CPU 68HC08

Fuente: <http://arantxa.ii.uam.es/~gdrivera/robotica/hc08/guia68hc08.pdf>

- El registro de índice se ha ampliado a 16-bits, ayudando a manejar mejor las tablas o estructuras de datos que son mayores de 256 bytes.
- El puntero de pila y el contador de programa son registros de 16-bits, sin tener en cuenta la memoria interna disponible. Más adelante se verá la pila ('stack').
- En Power-On Reset, el 68HC08 se parece al 68HC05. Durante el Reset, los 8 bits más altos del registro de índice HX, de los 16 bits que tiene, se ponen a cero y el puntero de pila se inicializa a \$00FF como en el 68HC05. Considerando que la mayoría de 68HC08 tienen mayor cantidad de RAM disponible, es probable que el usuario relocalizará el puntero de pila ('stack pointer'). Sin embargo, esta característica ayuda a mantener la compatibilidad operacional con el software existente del 68HC05.
- En la CPU del 68HC08 el bit del registro de código de condición (CCR) facilita los cálculos aritméticos con signo. Esta mejora permite a los programadores de lenguaje ensamblador y a los compiladores, realizar cálculos de direccionamiento mucho mejor.



### 3.2.1.- Tiempos de los ciclos internos de La CPU 68HC08

El 68HC08 utiliza cuatro fases del reloj interno en cada ciclo de ejecución de la CPU. Si el 68HC08 está gobernado por un cristal, el ciclo de ejecución es un cuarto de la frecuencia del cristal. A este ciclo se le llama ciclo de bus o ciclo de instrucción.

En el 68HC08, todos los tiempos de cada instrucción se especifican en ciclos de bus. Por ejemplo, un reloj de entrada de 32 MHz producirá una frecuencia de bus de 8 MHz. Un ciclo de bus de una instrucción se ejecutará en 125ns o 1 dividido por 8 MHz.

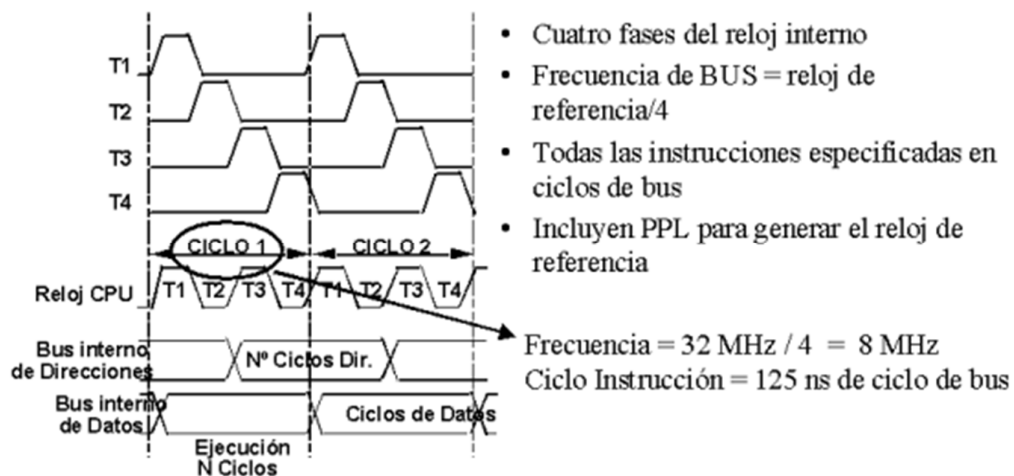


Figura III.14.- Tiempos de ciclos internos de la CPU 68HC08

Fuente: <http://arantxa.ii.uam.es/~gdrivera/robotica/hc08/guia68hc08.pdf>

La mayoría de 68HC08 incluye un circuito PLL interno que a partir de una frecuencia más baja de cristal, genera internamente una frecuencia de reloj mucho más alta que la del cristal. Para más información sobre la generación

del reloj y el circuito PLL, se puede ver en el Módulo de Generación de Reloj (CGM) de esta guía didáctica.

### **3.2.2.- Organización de la memoria Del 68HC08**

La figura siguiente muestra el mapa de memoria del 68HC08. Este mapa de memoria es idéntico al empleado para el 68HC05, sólo que se ha implementado hasta 64 kbytes, sin tener en cuenta los diferentes tamaños de memoria de cada 68HC08 disponible.

En el 68HC08, el mapa de memoria empieza con los registros de los periféricos con 64 bytes. A continuación le sigue el espacio para la RAM. La ROM o FLASH ocupa la parte superior de la memoria que precede a \$FE00, donde están los Vectores, el programa Monitor y los Registros de control. En el medio del mapa de memoria, entre la RAM y la ROM/FLASH, hay una zona de memoria no utilizable por el usuario que realiza las verificaciones de direcciones ilegales.

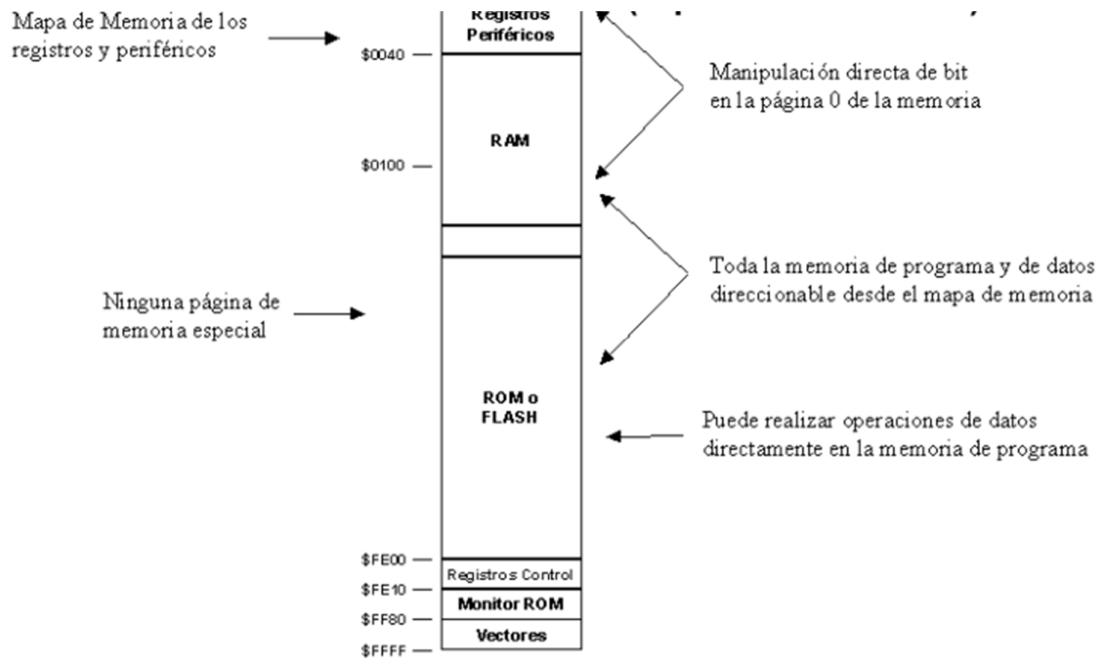


Figura III.15.- Organización de Memoria de 68HC08

Fuente: <http://arantxa.ii.uam.es/~gdrivera/robotica/hc08/guia68hc08.pdf>

Todas las MCU de la familia 68HC08 usan este modelo, aunque algunas de ellas tienen una dirección de inicio y final para las áreas específicas del monitor ROM o los registros de control.

### **3.2.3.- Juego de instrucciones Del 68HC08**

El 68HC05 tiene un total de 85 instrucciones que en la actualidad todavía forman un juego básico de instrucciones funcional y potente. El juego de instrucciones del 68HC08 se ha construido sobre la base de las instrucciones del 68HC05. El 68HC08 añade 28 instrucciones al juego de instrucciones del 68HC05, remarcadas en la tabla siguiente. Muchas de estas instrucciones soportan el registro del índice extendido a 16 bits y el puntero de pila ('stack pointer') es totalmente re localizable.

- Movimiento de Datos LDA, LDX, STA, STX, TAX, TXA, LDHX, MOV, PSHA, PSHH, PSHX, PULA, PULH, PULX, STHX
- Aritméticas ADD, ADC, SUB, SUBC, MUL, DAA, DIV
- Manipulación de Datos INCA, INCX, INC, DECA, DECX, DEC, CLR, NEGA, NEGX, NEG, (Inc/Dec/Neg/Clr) AIS, AIX, CLRH
- Manipulación de Datos ROLA, ROLX, ROL, RORA, RORX, ROR, LSLA, LSLX, LSL, LSRA, (Rotación/Desplazamiento) LSRX, LSR, ASRA, ASRX, ASR
- Manipulación de Datos BSET, BCLR
- Lógicas AND, ORA, EOR, COMA, COMX, COM, NSA
- Test de Datos CMP, CPX, BIT, TSTA, TSTX, TST, BRCLR, BRSET, CPHX
- Bifurcación BRA, BRN, BSR, BHI, BLO, BHS, BLS, BPL, BMI, BEQ, BNE, BCC, BCS, BHC, BHCC, BHCS, BMC, BMS, BIL, BIH, BGE, BGT, BLE, BLT, CBEQ, CBEQA, CBEQX, DBNZ
- Salto/Retorno JMP, JSR, RTS
- Control SEC, CLC, SEI, CLI, SWI, RTI, RSP, NOP, WAIT, STOP, TAP, TPA, TSX, TXS

Se han añadido varias instrucciones de bifurcación condicional, como: el ajuste decimal del acumulador DAA y la instrucción NSA (Nibble Swap Accumulator).

La instrucción de división DIV sin signo, dividiendo 16 por 8 ejecutándose en sólo 7 ciclos de bus. La instrucción multiplicación (MUL) se ha reforzado para que se ejecute en sólo 5 ciclos de bus, en lugar de los once ciclos necesarios para el 68HC05. La instrucción CLRA para Borrar el Acumulador es de un sólo ciclo para el 68HC08, mientras que el 68HC05 necesita 3 ciclos.

Esta mejora del juego de instrucciones del 68HC08, proporciona mayor eficacia de código y funcionamiento, recogiendo múltiples instrucciones y realizando la misma tarea con una sola instrucción. Más adelante se verán en detalle estas nuevas instrucciones.

### **3.3.- MODOS DE DIRECCIONAMIENTO DEL 68HC05**

Para comprender mejor los modos de direccionamiento del 68HC08, es mejor empezar viendo algunos ejemplos de los modos de direccionamiento del 68HC05, como el modo de direccionamiento inherente, inmediato, directo, extendido, relativo e indexado. Estos modos de direccionamiento tanto se pueden usar en el 68HC05 como en el 68HC08.

#### **3.3.1.- Inherente: CLRA**

Las instrucciones de direccionamiento inherente no tienen ningún operando, ya que el operando se define en el 'opcode' de 8-bits. Por ejemplo para borrar el acumulador, se usa la instrucción CLRA, que es una instrucción de un sólo ciclo para el HC08; el HC05 toma 3 ciclos.

### **3.3.2.- Inmediato: LDA #20**

Las instrucciones de direccionamiento inmediato tienen los operandos, que siguen inmediatamente al 'opcode', de 8-bits o de 16-bits. Cargar el acumulador con 20 es una instrucción de dos bytes, que se ejecuta en 2 ciclos de bus.

### **3.3.3.- Directo: LDA \$40**

Las instrucciones de direccionamiento directo tienen la dirección de 8-bits del operando que sigue inmediatamente al 'opcode'. Por consiguiente, las instrucciones acceden directamente a los primeros 256 bytes de la memoria. Anteriormente, se hizo referencia a este tipo de acceso como página directa o página cero. Cargar el acumulador con el operando a la posición de memoria 40, es una instrucción de dos bytes que se ejecuta en 3 ciclos de bus.

### **3.3.4.- Extendido: LDA \$4000**

Las instrucciones de direccionamiento extendido proporcionan direccionamiento absoluto a cualquier posición en los 64K del mapa de memoria, sin paginar. Requieren en total tres bytes para el 'opcode' más la dirección de 16-bits del operando. La mayoría de los ensambladores usan el modo directo más corto automáticamente, para cualquier acceso a los primeros 256 bytes del mapa de memoria.

### **3.3.5.- Relativo ( $\pm 128$ ) BLT LOOP**

Todas las instrucciones de bifurcación condicional usan el direccionamiento relativo. Si la condición de bifurcación es verdad, el contador de programa se agrega al byte con signo, que sigue inmediatamente al 'opcode' de bifurcación. Esto da un rango de -128 a +127 bytes, para la bifurcación. Las instrucciones de salto o de salto a subrutina, se pueden usar para mover el contador de programa en cualquier parte de los 64K del mapa de memoria.

### **3.3.6.- Indexado Sin 'Offset' LDA, X**

### **3.3.7.- Indexado Con 'Offset' de 8 bits LDA \$40, X**

### **3.3.8.- Indexado Con 'Offset' de 16 bits LDA \$4000, X**

Los modos de direccionamiento indexado son claves para direccionar tablas y otras estructuras de datos de una manera eficaz. El direccionamiento indexado sin desplazamiento ('offset'), se refiere a lo que en la mayoría de otras arquitecturas es el direccionamiento del puntero indirecto. El valor en el registro de índice es la dirección o el puntero del operando. Los modos de direccionamiento con 'offset' proporcionan al 68HC08 una gran eficacia de código comparado con otras arquitecturas con sólo direccionamiento indirecto o direccionamiento indirecto con otro registro de 'offset'.

## **3.4.- MODOS DE BAJO CONSUMO DEL 68HC08 (WAIT Y STOP)**

Modo Wait: Se entra ejecutando la instrucción WAIT (Consumo típico 50% de IDD en modo RUN)

### **3.4.1.- Efectos:**

- El reloj de la CPU se desactiva.
- El reloj del bus sigue activo.
- Los periféricos individuales se pueden desactivar para mejorar el consumo.

Se sale del modo 'Wait' por un Reset o por una interrupción externa o interna.

### **3.4.2.- Modo STOP:**

Se entra ejecutando la instrucción STOP (Consumo típico: IDD de 1µA a 3 µA)

Efectos:

- El reloj de la CPU se desactiva.
- El reloj del Bus opcionalmente se desactiva.

Se sale del modo 'Stop' por un Reset, por una interrupción externa o por una interrupción TBM (si el reloj del bus está activado).

### **3.4.3.- Ejercicio para encontrar el valor máximo en una lista de diez bytes.**

Se trata de escribir una rutina para encontrar el valor máximo en una lista de diez bytes de datos. En las instrucciones de inicialización, hay que poner el origen de los datos empezando en la dirección \$130. Usar una tabla para guardar los diez bytes de datos y poner el origen del programa en la dirección \$6E00. En el lazo del programa principal, hay que usar los punteros para buscar el valor máximo, incluyendo un chequeo para el valor 0 del byte.



ORG \$130; Origen de los datos en la dirección \$130

DATA FCB \$C0, \$40, \$8B; Tabla de bytes de datos

FCB \$75, \$A0, \$60

FCB \$DB, \$25, \$B0

FCB \$50

ORG \$6E00; Origen del programa en la dirección \$6E00

LDX #10 ; 1. Carga el número de bytes en el registro X.

GETBYTE LDA DATA-1,X ; 2. Carga el acumulador con el byte desde  
la dirección fuente.

DECCNT DECX ; 3. Decrementa el registro X.

BEQ DONE ; 4. Si el puntero = 0, entonces ir al paso 8 (Hecho)

CMP DATA-1,X ; 5. Compara el valor del acumulador del byte

; apuntado con el puntero fuente.

BHS DECCNT ; 6. Si el valor del acumulador es mayor del  
valor del byte apuntado

; con el puntero fuente ir al paso 3, el resto  
ir al paso 7.

BRA GETBYTE ; 7. Lazo hacia atrás con la instrucción

GETBYTE

DONE BRA DONE ; 8. Hecho.

### 3.5.- TIEMPOS DEL RESET INTERNO

Ahora que se han visto los diferentes tipos de reset interno, se pueden ver los tiempos internos del reset.

Todos los tipos de reset interno, excepto el POR, ponen el pin RST a un nivel bajo durante 32 ciclos de reloj CGMXCLK, para permitir hacer un reset a los dispositivos externos.

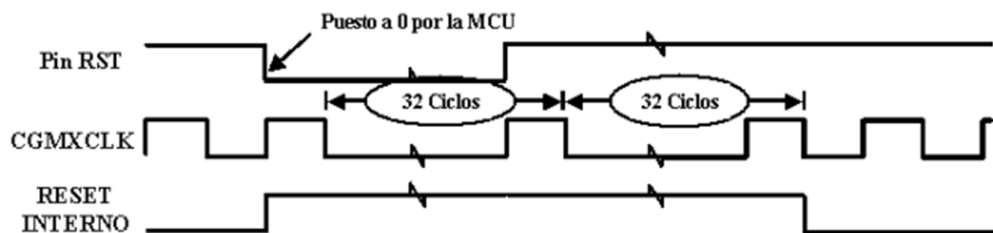


Figura III.16.- Tiempos de Reset Interno

Fuente: <http://arantxa.ii.uam.es/~gdrivera/robotica/hc08/guia68hc08.pdf>

La MCU se mantiene en reset durante 32 ciclos de reloj CGMXCLK después de que vuelva a un nivel alto el pin RST, para permitir estabilizar los dispositivos externos. El pin RST se prueba para verificar si el pin RST todavía está en estado bajo. En ese caso, indica que ha ocurrido un reset externo. Si no, indica que ha ocurrido un reset interno y el bit de reset interno se pone a 1, en el registro de estado del reset del SIM.

### 3.5.1.- Manipulación de una excepción

Véase el orden de apilado de los registros. Los registros se apilan de la dirección más alta a la más baja.

Antes se mencionó que el registro H no se guarda automáticamente por la CPU. Por consiguiente, debe guardarse al principio del manejo de la excepción, ejecutando una instrucción PSHH. Esto significa que se tiene que restaurar antes de terminar el manejo de la excepción.

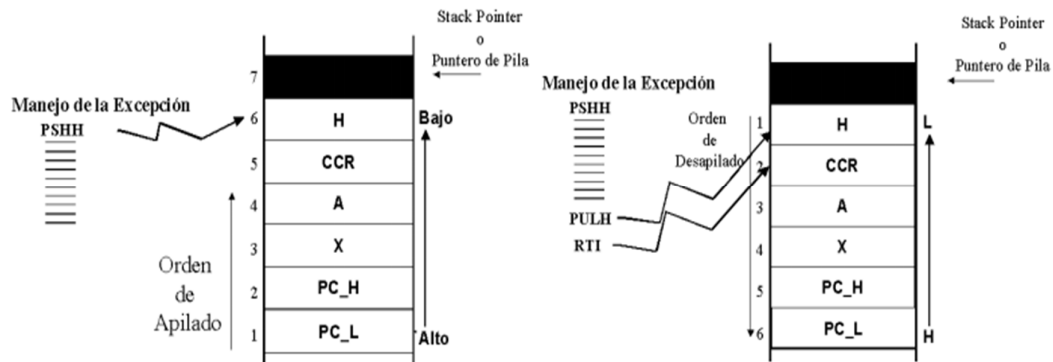


Figura III.17.- Manipulación de una excepción

Fuente: <http://arantxa.ii.uam.es/~gdrivera/robotica/hc08/guia68hc08.pdf>

Antes de que se restaure el contexto viejo, véase el orden para desapilado. Se empieza ejecutando la instrucción PULH para restaurar el registro H. Luego, se ejecuta la instrucción RTI para restaurar los registros restantes y se empieza con el registro CCR. Cuando el registro CCR se restaura, se borra a la condición original, el bit de máscara de interrupción global también se restaura y activa las interrupciones. Cuando el contexto viejo se restaura, continúa el

funcionamiento normal del programa de la aplicación. Si hay interrupciones adicionales pendientes, el proceso empezará de nuevo.

### 3.5.2.- Diagrama de bloques DEL LVI

Ahora, se puede ver cómo se configura este Módulo LVI. El Módulo LVI contiene un circuito de referencia 'bandgap' y un comparador que determinan cuando la MCU opera con un voltaje por debajo del punto de disparo especificado. El Módulo LVI se activa fuera del reset y se configura usando varios bits en el registro CONFIG de configuración del sistema. Este registro se inicializa típicamente con el 'Power-On Reset'. Una vez se ha escrito el registro no se puede escribir de nuevo hasta que ocurra el siguiente reset. Esta característica de 'escribir una sola vez' asegura que una aplicación no puede reconfigurar la MCU inadvertidamente.

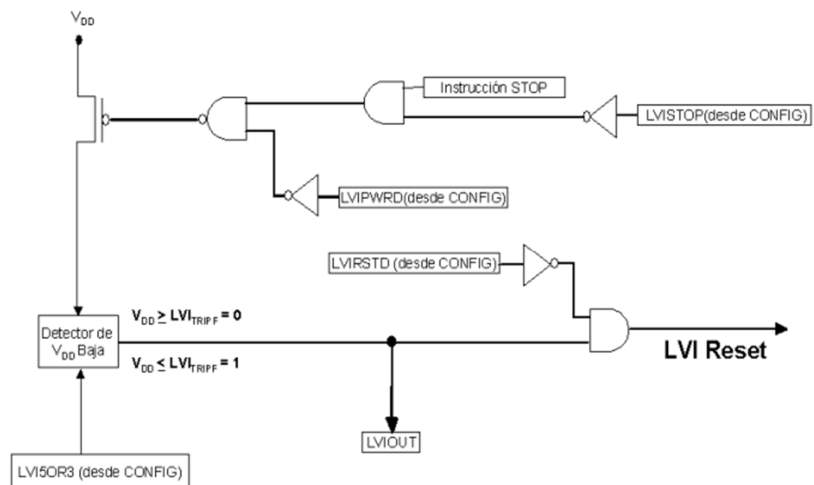


Figura III.18.- Diagrama de Bloques del LVI

- El bit LVIPWRD desactiva la alimentación del LVI, también permite al Módulo LVI supervisar el voltaje de alimentación (VDD). Poniendo este bit a 0, se aplica alimentación al circuito analógico LVI. Poniendo a 1 el bit LVIPWRD, se quita la alimentación del Módulo LVI.
- El bit LVISTOP activa el modo 'stop' del LVI, determina si el LVI opera en modo 'stop'. Cuando el bit LVIPWRD se pone a 0 y poniendo el bit LVISTOP a 1, permite al Módulo LVI operar durante el modo 'stop'. Un reset borra el bit LVISTOP. La habilidad de desactivar el LVI automáticamente en modo 'stop' permite reducir el consumo del circuito y alarga la vida de la batería, mientras protege a la CPU durante el funcionamiento normal.
- El bit LVI5OR3 selecciona el modo de voltaje en que trabaja el Módulo LVI, a 5 V o 3 V. El modo de voltaje en que trabaja el LVI, debe ser igual al voltaje de alimentación (VDD). Poniendo a 1 el bit LVI5OR3, configura el punto de disparo (VTRIPF) para un funcionamiento nominal de 5V. Borrando el bit LVI5OR3, configura el punto de disparo (VTRIPF) para un funcionamiento nominal de 3V. En reset, el LVI tiene un valor predefinido de 3V. Si alimenta un sistema a 5V, el bit LVI5OR3 típicamente hay que ponerlo a 1 para elevar el punto de disparo para trabajar a 5V, después de cada 'power-on reset'. Los valores del punto de disparo están típicamente en el rango de 4.2 V a 4.5 V en el modo de 5V o de 2.4 V a 2.7 V en el modo de 3V. Para la información sobre los valores del punto de disparo específico, véase las especificaciones eléctricas de cada MCU en el libro de datos técnicos.

- El bit LVIRSTD desactiva la señal de reset del Módulo LVI. Cuando se pone el bit LVIRSTD a 0, el Módulo LVI generará un reset cuando LVIOUT se pone a 1, lo que significa que VDD ha caído por debajo del voltaje de disparo VTRIPF. Si ocurre esta condición, la MCU permanecerá en la condición de reset hasta que VDD suba por encima del voltaje de disparo VTRIPR. VTRIPF es más bajo que VTRIPR, por histéresis, típicamente de 0.1 V, para evitar entrar y salir de 'reset' cuando el voltaje de alimentación tiene un ligero rizado cercano al voltaje de disparo.

### 3.5.3.- Registro de estado del LVI (LVISR)

El Módulo LVI usa el registro de estado LVISR, para indicar que el voltaje VDD ha caído por debajo del nivel de VTRIPF. El bit LVIOUT del LVI, es un indicador de estado de sólo lectura, que el Módulo LVI pone a 1 cuando VDD es menor de VTRIPF. El Módulo LVI pone a 0 este bit cuando VDD sube por encima de VTRIPR. Un reset pone a 0 el bit LVIOUT.

Tabla III.1.- Tabla LVISR:

SFEOC	Bit 7	6	5	4	3	2	1	Bit 0
Leer:	LVIOUT	0	0	0	0	0	0	0
Escribir:								
Reset:	0	0	0	0	0	0	0	0

Fuente: <http://arantxa.ii.uam.es/~gdrivera/robotica/hc08/guia68hc08.pdf>

Este indicador de estado es particularmente útil cuando el Módulo LVI se usa en modo de funcionamiento en modo consulta, llamado modo 'polled' (no es

operativo, solo monitoriza la alimentación y no se puede utilizar para generar un reset). Este modo normalmente se usa en aplicaciones donde se trabaja con niveles de VDD por debajo del nivel de VTRIPF, donde se desea aumentar la vida de la batería, por ejemplo. En este caso, el software puede supervisar VDD consultando ('polling') el indicador de estado LVIOUT.

Para configurar el LVI para este modo, hay que activar el Módulo LVI poniendo a 0 el bit LVIPWRD y a 1 el bit LVIRSTD para desactivar resets del LVI.

### **3.6.- FUNCIONAMIENTO DEL SPI**

El SPI se ha desarrollado alrededor de un doble registro de desplazamiento 'buffered' de 8 bits con ambos extremos del registro de desplazamiento, que van a los pines de la MCU. Un extremo del registro de desplazamiento se conecta al pin MISO (Master-In Slave-On). Este pin actúa como una entrada para el Módulo 'master' del SPI y como una salida para el Módulo 'slave' del SPI. El otro extremo del registro de desplazamiento se conecta al pin MOSI (Master-On Slave-In).

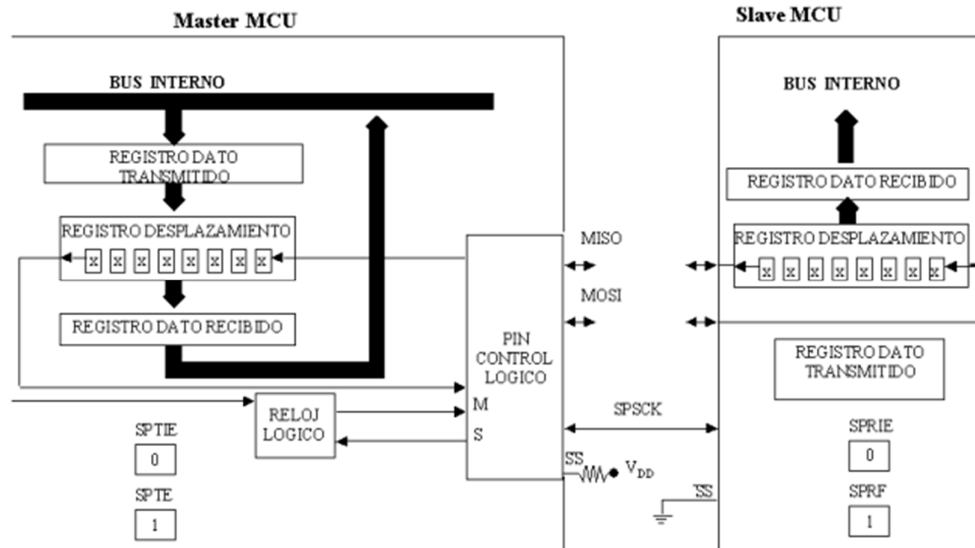


Figura III.19.- Funcionamiento del SPI

Fuente: <http://arantxa.ii.uam.es/~gdrivera/robotica/hc08/guia68hc08.pdf>

La CPU empieza una transferencia serie SPI, escribiendo un byte de datos en el registro de datos transmitidos. Se transferirán automáticamente vía serie todos los 8 bits de datos a través del pin MOSI del 'master', sincronizados con el reloj de salida del 'master' (SPSCK). Cada vez que se desplaza un bit a través del pin MOSI del 'master', se desplaza un bit a través del pin MISO del 'master', permitiendo una comunicación 'full-duplex'.

### 3.6.1.- Conexiones Master-Slave del SPI

En la figura siguiente se pueden ver las conexiones entre un SPI 'Master' y un SPI 'Slave'. Primero se debe configurar el SPI del circuito como 'Master' y el otro circuito como 'Slave'. Para activar el SPI como Master, se une el pin SSa VDD. Para activar el SPI como 'Slave', hay que conectar el pin SSa masa.



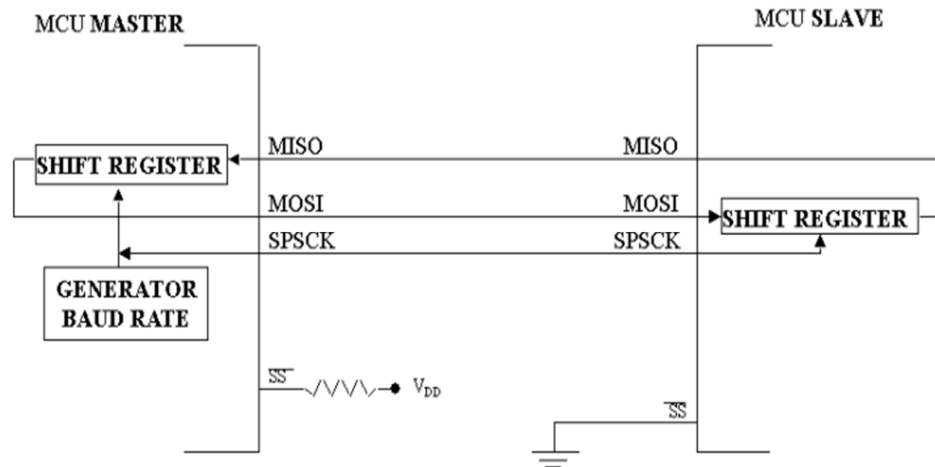


Figura III.20.- Conexión de Master-Slave del PSI

Fuente: <http://arantxa.ii.uam.es/~gdrivera/robotica/hc08/guia68hc08.pdf>

Esto permite a un circuito 'Master' seleccionar con qué circuito 'Slave' se va a comunicar, en el caso de que se necesiten varios circuitos periféricos 'Slave'. Típicamente, un 'Master' usa un pin del puerto de I/O de la MCU para controlar cada pin SS de cada 'Slave'. Si el pin SS de un 'Slave' se pone a 1, el pin MISO del 'Slave' se pone en un estado de alta impedancia. Cuando ocurre esto, el 'Slave' ignorará todos los relojes SPSCK entrantes, aunque esté en medio de una transmisión. Se puede configurar y supervisar el funcionamiento del SPI usando los tres registros del Módulo SPI.

### 3.6.2.- Cálculo del BAUD RATE

La tabla No siguiente muestra los códigos para seleccionar uno de los cuatro divisores de 'baud rate', usando los bits SPR1 y SPR0 de selección de la

velocidad de transmisión del SPI. La velocidad de transmisión se calcula dividiendo la frecuencia del bus por dos veces el divisor de 'baud rate'.

Tabla III.II.- Cálculo de Baud Rate

SPR1 y SPR0	Baud rate Divisor (BD)
00	2
01	8
10	32
11	128

Fuente: <http://arantxa.ii.uam.es/~gdrivera/robotica/hc08/guia68hc08.pdf>

Para calcular la velocidad de transmisión del SPI se usa esta formula:

$$Baud\ Rate = \frac{CGMOUT}{2 \times BD}$$

Donde,

CGMOUT= Reloj Base de salida del módulo generador de reloj (CGM)

BD= División de Baud rate

Por ejemplo, asumiendo una frecuencia de bus de 8 Mhz y poniendo SPR1 y SPR0 a 0, se selecciona el 'baud rate' divisor por 2. La frecuencia de bus es 8 Mhz, dividido por dos veces el divisor de 'baud rate', que es 4, da un baud rate de 2 Mhz.

$$Baud\ Rate = \frac{CGMOUT}{2 \times BD} = \frac{8}{2 \times 2} = \frac{8}{4} = 2MHz$$

### 3.6.3.- Ejemplo de transmisión de datos Master-Slave

El software empieza una transmisión SPI, escribiendo un byte en el registro de datos de transmisión del SPI. Si el registro de desplazamiento está vacío, el byte se transfiere inmediatamente a este registro de desplazamiento. Por otra parte, la transferencia empieza cuando el registro de desplazamiento termina de transferir el byte anterior. Una vez el byte es transferido desde el registro de transmisión del SPI al registro de desplazamiento, el bit SPTE se pone a 1 indicando que se puede escribir otro byte al registro de transmisión de datos del SPI. El bit SPTE genera una interrupción a la CPU, si el bit SPTIE se pone a 1. Esto permite manejar la interrupción de transmisión de transferencia multi-byte. En este ejemplo, el bit SPTIE se pone a 0, para desactivar las interrupciones.

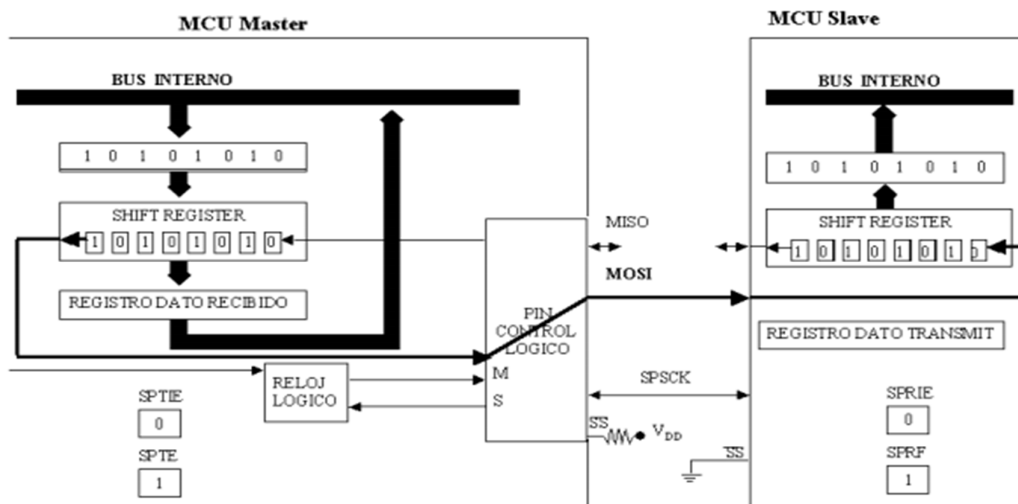


Figura III 21.- Transmisión de datos Master-Slave

Fuente: <http://arantxa.ii.uam.es/~gdrivera/robotica/hc08/guia68hc08.pdf>

El byte empieza desplazando un bit, en el pin MOSI sincronizado con la señal SPSCCK de reloj 'master'. La transferencia continuará durante 8 SPSCCK ciclos de reloj y transferirá todos los 8-bits. Como que el byte se ha desplazado fuera del SPI 'master', se desplaza otro byte en el pin MISO del 'slave'. En este ejemplo, se desatiende el byte de información que se desplaza del SPI 'slave' al SPI 'master'.

La transmisión acaba cuando el byte completo se desplaza fuera del registro de desplazamiento del SPI 'master' y en el registro de desplazamiento del SPI. El registro de desplazamiento 'slave' se transfiere automáticamente al registro de datos del SPI 'slave' si está vacío y permite transferir otro byte si hay alguno pendiente. El bit SPRF se pone a 1 indicando que el registro SPI receptor está lleno y esperando ser leído. Si el bit SPRIE se pone a 1, se genera también una petición de interrupción del SPI receptor. En este ejemplo, el bit SPRIE se pone a 0.

Para evitar un desbordamiento, el 'slave' debe leer el registro de datos del receptor anterior para intentar que el 'master' transfiera más de un byte adicional. Por ejemplo, si el 'master' ha transmitido con éxito un byte que está esperando ser leído en el registro de datos del receptor, el 'master' puede transferir un byte adicional.

Este byte adicional se mantendrá en el registro de desplazamiento y no se cargará en el registro del receptor hasta que el byte anterior no haya sido leído por la CPU. Si el 'master' intenta transmitir vía serie otro byte antes de leer el registro de datos del receptor, se sobrescribirán los datos en el registro de

desplazamiento 'slave' causando una condición de error por desbordamiento. El byte en el registro de datos del receptor no se sobrescribirá.

### 3.6.4.- Formatos de Transmisión

El software puede seleccionar cualquiera de las cuatro combinaciones de la fase del reloj y polaridad del SPI, usando los bits CPOL y CPHA. La polaridad del reloj selecciona un reloj alto o bajo activo y no tiene efecto significativo en el formato de la transmisión. La fase del reloj selecciona uno de dos formatos de la transmisión fundamentalmente diferentes.

Seguidamente se examina la transmisión SPI para cada fase del reloj, empezando con el bit de la fase del reloj puesto a 0 y después a 1.

#### CPH=0

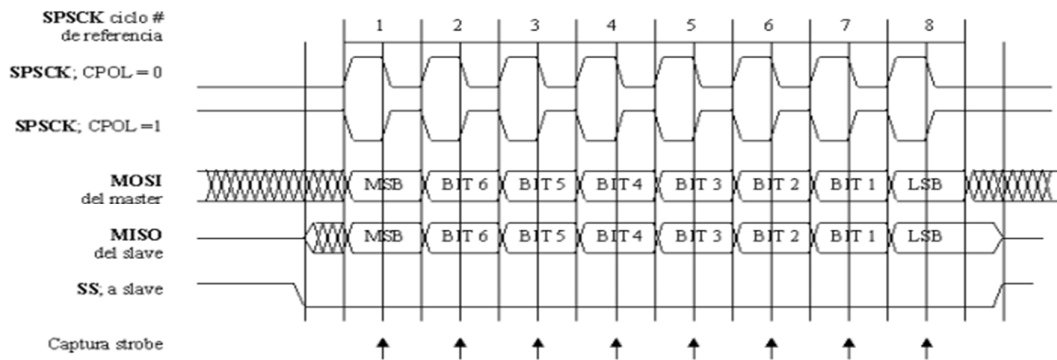


Figura III.22.- Formato de Transmisión CPH=0

Fuente: <http://arantxa.ii.uam.es/~gdrivera/robotica/hc08/guia68hc08.pdf>

El flanco de bajada del pin SS seleccionado como 'slave', activa al 'Slave' para empezar la transmisión manejando el pin MISO con el MSB del registro de desplazamiento 'Slave'. El 'Master' empieza la transmisión manejando el MSB del registro de desplazamiento del Master en el pin MOSI. Entonces el 'Master' manejará el reloj SPSCCK del SPI. Al primer flanco del reloj SPSCCK, ocurre la captura 'strobe'. Esto completa la transferencia de los bits MSB del Slave desde el Master y desplaza los datos en el registro de desplazamiento.

Después del siguiente flanco del reloj SPSCCK, el 'master' y el 'slave' empezarán manejando los seis bits hacia sus pins respectivos. El ciclo continúa hasta que todos los 8-bits se han transferido.

### CPH=1

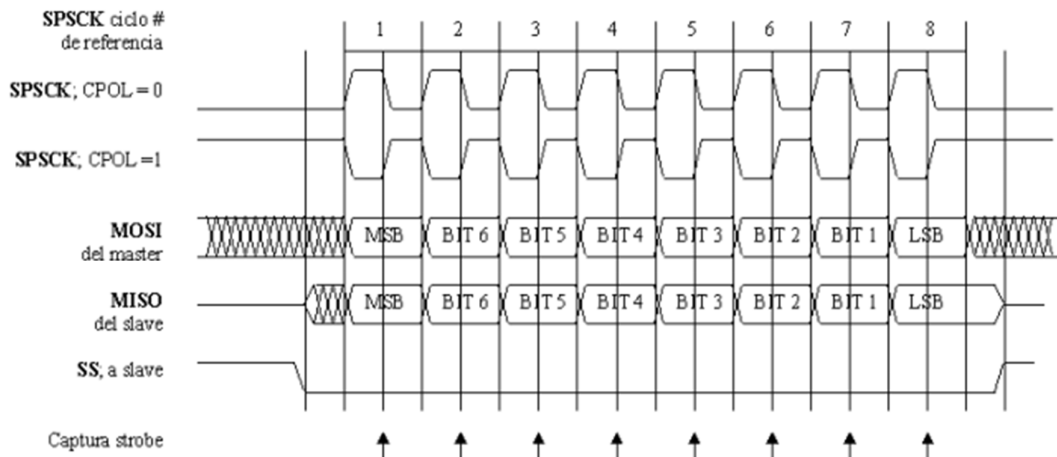


Figura III.23.- Formato de Transmisión CPH=1

Fuente: <http://arantxa.ii.uam.es/~gdrivera/robotica/hc08/guia68hc08.pdf>

Cuando el pin SS 'slave' pasa a nivel bajo, el 'Slave' espera hasta el primer flanco del reloj SPSCCK para manejar su pin MISO. Los bits MSB del 'master' y el 'slave', manejan a sus respectivos pins al primer flanco de reloj SPSCCK. Ocurre la captura 'strobe' en el segundo flanco de reloj SPSCCK con CPH puesto a 1, en lugar del primer flanco cuando CPH está puesto a 0. Finalmente, el ciclo continúa hasta que todos los 8-bits se han transferido.

### **3.7.- MÓDULO ADC (Analog To Digital Converter)**

En esta parte se describen las características y la configuración del Módulo ADC. Así como las técnicas para obtener la máxima precisión en las conversiones del ADC y se escribe un programa para hacer medidas con el convertidor.

#### **3.7.1.- Características del ADC**

Algunas MCU de la familia 68HC08 incluyen un convertidor A/D de 8-bits o 10-bits de resolución. Algunos ejemplos del 68HC08 con un ADC de 8-bits son el 68HC908GP, GR, JL, JK y KX. Algunos ejemplos del 68HC08 con un ADC de 10-bits son el 68HC908MR y 68HC908SR. Todos los Módulos ADC de los 68HC08 usan el principio de aproximaciones sucesivas. Una discusión de este principio se puede encontrar en la el Manual de Referencia del M68HC11 M68HC11RM/AD (véase en <http://www.mcu.motsps.com>)

Todos los Módulos ADC soportan dos modos de conversión, en modo de conversión continua y en modo de conversión única. En modo de conversión única, se completa una conversión entre escribir el registro de estado y el registro de control. En modo de conversión continua, la entrada analógica del ADC convierte continuamente y lo escribe en el registro de datos del ADC. En este modo, los datos de la conversión anterior se borran sin tener en cuenta si estos datos han sido leídos o no.

El ADC ofrece dos maneras diferentes de supervisar el estado de conversión completa. Dependiendo del modo de conversión, cuando se ha completado una conversión, se puede usar el software para consultar un indicador ('flag') o se puede configurar el ADC para generar una señal de interrupción. El ADC tiene un reloj de entrada seleccionable. Se puede usar este reloj de entrada para optimizar las conversiones del ADC para diferentes frecuencias de cristal y para acomodar el 68HC08 con el PLL interno.

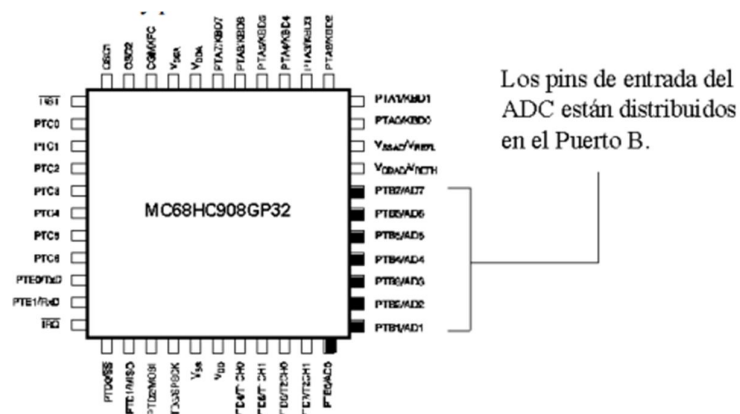


Figura III.24.- Modulo ADC

Fuente: <http://arantxa.ii.uam.es/~gdrivera/robotica/hc08/guia68hc08.pdf>



### 3.7.1.1.- SELECCIÓN DEL CANAL DE ENTRADA

La siguiente tabla muestra los códigos de selección de canales del ADC del 68HC908GP32. Con estos bits también se puede seleccionar VREFH y VSSAD, así como la señal de entrada para verificar el funcionamiento del ADC.

Tabla III.III.- Canales del ADC del 68HC908GP32

ADCH4	ADCH3	ADCH3	ADCH1	ADCH0	Selección de entrada
0	0	0	0	0	PTB0/AD0
0	0	0	0	1	PTB1/AD1
0	0	0	1	0	PTB2/AD2
0	0	0	1	1	PTB3/AD3
0	0	1	0	0	PTB4/AD4
0	0	1	0	1	PTB5/AD5
0	0	1	1	0	PTB6/AD6
0	0	1	1	1	PTB7/AD7
↓	↓	↓	↓	↓	Reservado
1	1	0	1	1	Reservado
1	1	1	0	0	Reservado
1	1	1	0	1	V <sub>REFH</sub>
1	1	1	1	0	V <sub>SSAD</sub>
1	1	1	1	1	Desconexión del ADC

Fuente: <http://arantxa.ii.uam.es/~gdrivera/robotica/hc08/guia68hc08.pdf>

Cuando no se necesita el ADC, se puede desconectar poniendo todos los canales seleccionados del ADC a 1. Esto ayudará a minimizar el consumo del sistema. Si se selecciona un canal de entrada sin usar o un código de bit reservado, el resultado de la conversión del ADC será desconocido.

### 3.7.2.- Cálculo del tiempo de una conversión

Una vez se ha seleccionado la fuente de reloj de entrada, se puede calcular la cantidad de tiempo que se usa para completar una sola conversión. Primero se determina el número de ciclos de reloj que se usa para completar la conversión y entonces dividir este valor por la frecuencia de reloj de entrada.

$$\text{Tiempo de una Conversión} = \frac{16 \text{ o } 17 \text{ (ciclos del ADC)}}{1\text{MHz (Frecuencia del ADC)}} = 16 \text{ o } 17 \mu\text{s}$$

El proceso de conversión empieza después de que se ha escrito el registro ADSCR. Para completar una conversión típica usa 16 ciclos de reloj del ADC. Cuando hay un retraso de sincronización del reloj, entre el reloj de la CPU y el reloj del ADC, para completar la conversión usará 17 ciclos de reloj. Es posible un retraso de sincronización del reloj si el reloj del A/D es diferente que el reloj de la CPU. Con la frecuencia de reloj de entrada puesta a 1MHz, una conversión típica usará aproximadamente de 16 a 17µs.

### **3.8.- CARACTERÍSTICAS DEL OSCILADOR A CRISTAL**

El CGMXCLK puede ser usado por módulos que requieren trabajar con unos tiempos muy precisos. El ciclo de servicio de CGMXCLK no se garantiza que sea del 50% y depende de factores externos, incluyendo el cristal y otros componentes externos. También se puede usar un reloj generado externamente entrando por el pin OSC1 del circuito oscilador a cristal. Para usar esta configuración, hay que conectar el reloj externo al pin OSC1 y dejar el pin OSC2 al aire.

El circuito oscilador a cristal puede tomar dos fuentes de frecuencia diferentes, la de un cristal o la de un oscilador externo. Si se elige usar un cristal, la frecuencia del cristal debe estar entre 30 kHz y 100 kHz, típicamente de 32.768 kHz. El oscilador no se ha diseñado para frecuencias de cristal fuera de este rango.

Si se elige usar un oscilador externo como fuente de reloj, se puede elegir tener el PLL activado o se puede desactivado. Con el PLL activado, la frecuencia del oscilador externo debe estar entre 30 kHz y 1.5 MHz. Con el PLL desactivado, la frecuencia del oscilador externo puede ir de cero a 32 MHz.

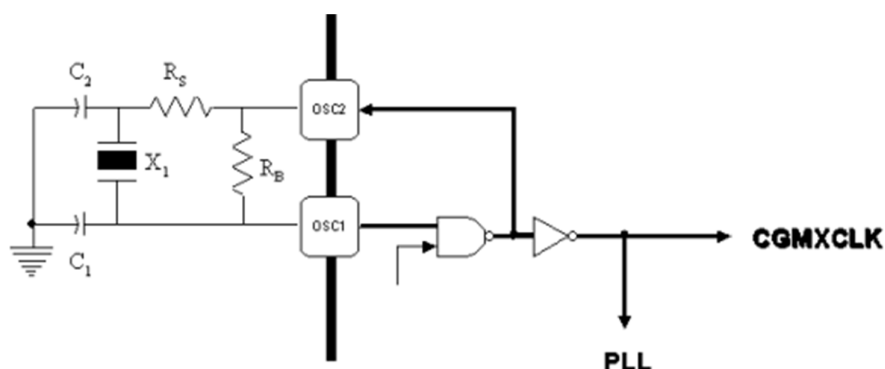


Figura III.25.- Circuito Oscilador a Cristal

Fuente: <http://arantxa.ii.uam.es/~gdrivera/robotica/hc08/guia68hc08.pdf>

### 3.8.1.- Oscilador Externo

Con la configuración de oscilador externo, se puede trabajar con el PLL activado o desactivado dependiendo de la frecuencia del oscilador.

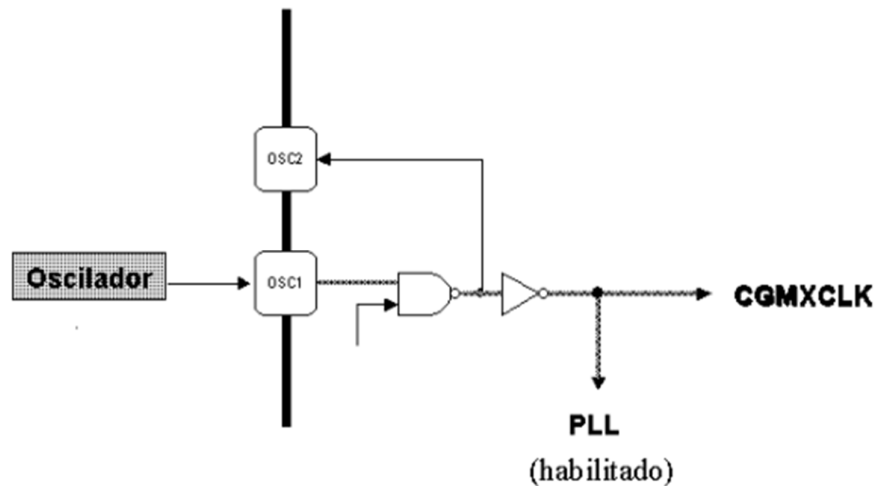


Figura III.26.- Circuito Oscilador Externo

Fuente: <http://arantxa.ii.uam.es/~gdrivera/robotica/hc08/guia68hc08.pdf>

### 3.9.- CARACTERÍSTICAS DEL PLL

Como se vio antes, el PLL genera la frecuencia del VCO programable. El PLL puede filtrar la frecuencia del VCO, usando dos modos: modo adquisición y modo rastreo ('tracking'). La selección del modo, depende la exactitud de la frecuencia de salida.

En **modo adquisición**, el filtro puede hacer correcciones de frecuencia grandes a la frecuencia del VCO. Este modo se usa al iniciar del PLL o cuando el PLL ha sufrido un ruido severo y la frecuencia del VCO está desplazada de la frecuencia deseada. Para seleccionar este modo, hay que seleccionar el bit ACQ en el registro de control del ancho de banda del PLL.

En **modo rastreo** ('tracking'), el filtro hace pequeñas correcciones a la frecuencia del VCO. Las fluctuaciones ('jitter') del PLL son muy bajas en modo

"tracking", pero la respuesta al ruido también es más lenta. El PLL entra en modo 'tracking' cuando la frecuencia del VCO es casi correcta, como cuando el PLL se selecciona como fuente de reloj base.

El PLL puede cambiar entre el modo adquisición y modo rastreo, automáticamente o manualmente. Se recomienda el modo automático para la mayoría de las aplicaciones. Hay muchas ventajas al incluir un PLL en la MCU. Se puede lograr una frecuencia de bus alta con un cristal económico, ya que los cristales de baja frecuencia son más baratos. Los cristales de baja frecuencia también consumen menos y proporcionan más alta inmunidad al ruido.

### **3.9.1.- Circuito PLL**

El PLL es básicamente un oscilador cuya frecuencia se bloquea en una componente de frecuencia de una señal de entrada. En este ejemplo, el PLL actúa como un multiplicador de frecuencia para lograr una frecuencia de alta resolución de salida, usando un cristal de baja frecuencia se minimiza el costo, consume menos y genera menos ruido.

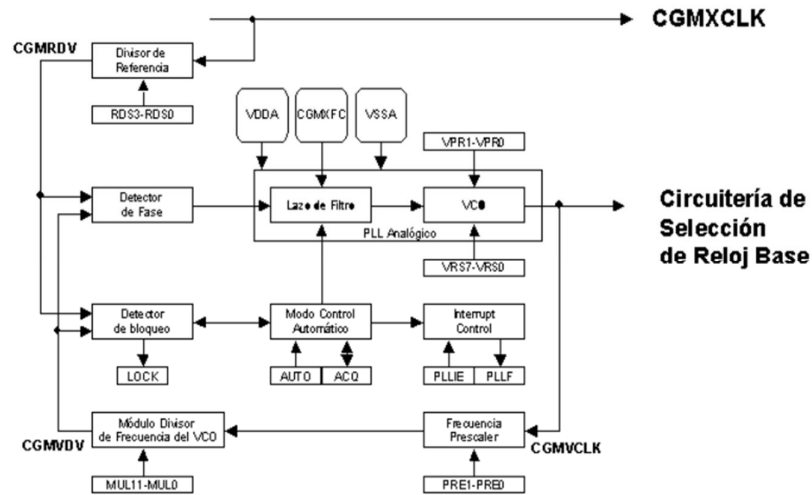


Figura III.27.- Circuito PPL

Fuente: <http://arantxa.ii.uam.es/~gdrivera/robotica/hc08/guia68hc08.pdf>

CGMRCLK es el reloj de referencia del PLL, que pasa a través de un "buffer", siendo entonces CGMXCLK. CGMRCLK corre a una frecuencia  $f_{RCLK}$  y alimenta al PLL a través de un módulo programable divisor de referencia, que divide la frecuencia por un factor R. La salida del divisor, es el reloj de referencia final, CGMRDV.

La salida de reloj del VCO, corre a una frecuencia,  $f_{VCLK}$  y se realimenta a través de un divisor programable "prescaler" y a través de un divisor programable "VCO". El divisor "prescaler" divide el reloj del VCO por un factor potencia de dos (P) y el módulo divisor de frecuencia VCO reduce esta frecuencia por un factor N. La salida es la de realimentación del reloj del VCO, CGMVDV, que corre a una frecuencia  $f_{VDV}$ .

El detector de fase compara CGMVDV con CGMRDV y genera un pulso de corrección basado en la diferencia de la fase de las dos señales. El filtro del

lazo altera ligeramente el voltaje DC en el condensador externo conectado a CGM/XFC basado en la anchura y en la dirección del pulso de corrección.

El detector de bloqueo también compara CGMVDV con CGMRDV. La frecuencia se bloquea cuando la diferencia de la fase entre las dos señales está muy cerca de cero.

### 3.9.1.1.- FRECUENCIAS PRECALCULADAS

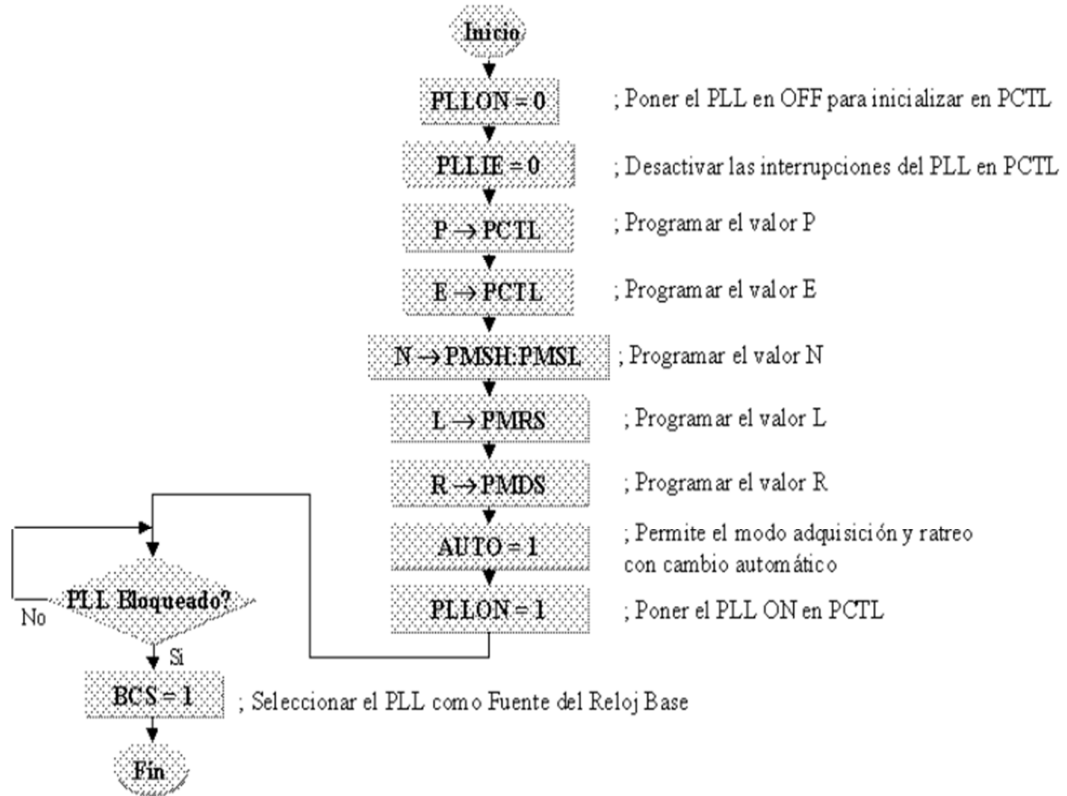
Motorola proporciona algunos ejemplos en una tabla para frecuencias de bus diferentes, usando un cristal típico de 32.768 kHz. Esto hace fácil de programar el PLL para frecuencias de bus típicas. Los valores ya están en anotación hexadecimal.

Tabla III.IV.- Frecuencias pre calculadas

$f_{BUS}$	$f_{RCLK}$	R	N	P	E	L
2.0 MHz	32.768 kHz	1	F5	0	0	01
2.4576 MHz	32.768 kHz	1	12C	0	1	80
2.5 MHz	32.768 kHz	1	132	0	1	83
4.0 MHz	32.768 kHz	1	1E9	0	1	D1
4.9152 MHz	32.768 kHz	1	258	0	2	80
5.0 MHz	32.768 kHz	1	263	0	2	82
7.3728 MHz	32.768 kHz	1	384	0	2	C0
8.0 MHz	32.768 kHz	1	3D1	0	2	D0

Fuente: <http://arantxa.ii.uam.es/~gdrivera/robotica/hc08/guia68hc08.pdf>

### 3.9.2.- Diagrama de flujo de configuración del PII



Primero se desactiva el PLL y las interrupciones, utilizando el modo consulta para verificar el bloqueo. Después se programan los valores P, E, N, L y R en los registros CGMC. Se pone el bit AUTO para permitir el filtro del lazo para cambiar automáticamente entre el modo adquisición y el modo rastreo. Se activa el PLL para alcanzar la frecuencia deseada. Se espera hasta que la frecuencia del VCO haya alcanzado el valor deseado, es decir, hasta la señal de bloqueo del PLL. Entonces se selecciona el PLL como la fuente de reloj base.



### **3.10. - MÓDULO SCI (Serial Communications Interface)**

El SCI es el módulo de interface de comunicación serie, ahora se describen las características y la configuración del Módulo SCI para la transmisión y la recepción de datos. También se preparan las comunicaciones serie del MC68HC08 a un PC.

#### ***3.10.1.- Características del módulo SCI***

El Módulo SCI, también llamada UART (Universal Asynchronous Receiver Transmitter), proporciona comunicaciones con dispositivos periféricos u otras MCUs, en modo full-duplex, asíncrono y hasta gran velocidad. El SCI usa un formato estándar de no retorno a cero (NRZ) e incluye un generador de velocidad de transmisión ('baud rate') interno. Este generador del SCI interno no requiere del 'Timer' del sistema del 68HC08.

En cambio, deriva una de las 32 frecuencias de 'baud rate' normales, directamente del oscilador de la MCU.

Se puede seleccionar una longitud del carácter de ocho o nueve bits. La configuración más común que se usa es un bit de inicio, 8-bits de datos del carácter y un bit de paro. La configuración de nueve bits usa un bit de inicio, 9-bits de datos del carácter y un bit de paro. El noveno bit de datos del carácter se puede usar como un bit de paro extra, como la función de 'wake-up' o de despertar el receptor del SCI o como un bit de paridad.

Se proporcionan dos métodos para despertar el receptor: 'wakeup' por dirección "marca" y el 'wakeup' por línea ocupada. En sistemas de múltiples receptores, este rasgo permite al Módulo SCI ignorar transmisiones predeterminadas a otros receptores entrando en un modo 'standby' o estado de espera.

Se activa el transmisor y receptor separadamente y son funcionalmente independientes, aunque usan los mismos formatos de datos y de velocidad de transmisión. El transmisor y el receptor son 'doble-buffered'. Esto significa que los caracteres "back to back" se manejan fácilmente, aun cuando la CPU tarda en responder al completar un carácter. Además, el receptor del SCI incluye varios rasgos avanzados para asegurar la recepción de datos muy fiable. Estos rasgos ayudan al desarrollo de redes de comunicación eficazmente.

Se puede programar la polaridad de salida del transmisor poniendo un bit en uno de los registros de control del SCI. Este rasgo le permite transmitir a nivel TTL. Algunos dispositivos transforman los niveles lógicos de TTL en niveles de datos estándar RS-232. Tales dispositivos requieren una polaridad de señal, en la que un nivel lógico 0 representa 0V y un nivel lógico 1 representa un voltaje por encima de 3V.

El SCI se puede interrumpir por medio de ocho indicadores ('flags') y peticiones de interrupción. El SCI mantiene las peticiones de interrupción y vectores, separados para el transmisor, el receptor y para las condiciones de error. Esto proporciona un proceso de interrupción muy eficaz, de función normal de transmisor/receptor sin ninguna exploración o verificación de

interrupción. Cualquier condición de error se puede manejar por una rutina de servicio de interrupción separada.

La figura siguiente muestra la asignación de los pins del SCI en el MC68HC908GP32 MCU en encapsulado 44 QFP.

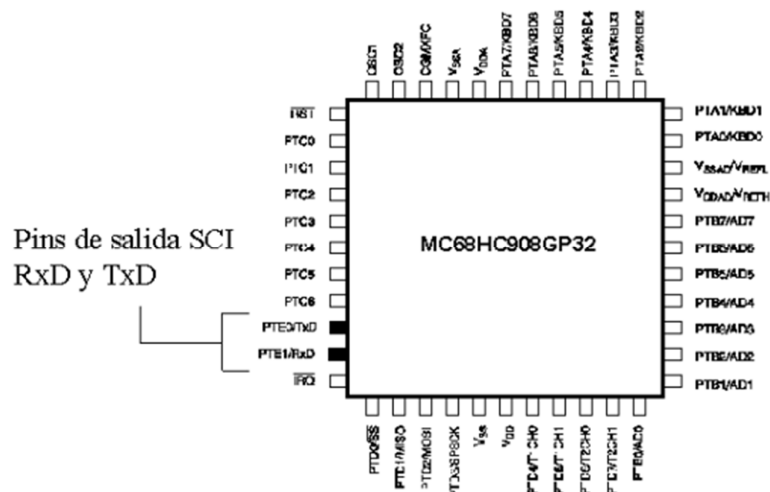


Figura III.28.- Modulo SCI

Fuente: <http://arantxa.ii.uam.es/~gdrivera/robotica/hc08/guia68hc08.pdf>

El Módulo SCI usa dos pines de entrada/salida. El pin RxD es un pin de entrada para recibir datos y el pin TxD es una salida para transmitir datos. Estos dos pins son compartidos con los pines del puerto de entrada/salida de la MCU. Cuando se desactiva el SCI, los dos pins se pueden usar como dos pines de entrada/salida de propósito general.

Si la aplicación requiere niveles RS-232 o RS-422, se necesita poner circuitos para cambiar a los niveles lógicos requeridos. Una transformación típica es de un nivel lógico 0 a 3 o 5V a niveles +/- 12V.

### **3.10.2.- Doble Buffer**

El Módulo SCI incluye un 'buffer' doble, el del transmisor y el del receptor. A continuación se verá cómo esta característica se lleva a cabo en el transmisor del SCI.

Para transmitir datos, la CPU escribe datos al registro de datos del SCI. Además del registro de datos, el transmisor del SCI incluye un registro de desplazamiento de 11-bits para guardar los datos durante la transmisión. Esto le permite a la CPU escribir un carácter de datos mientras el SCI está transmitiendo el carácter anterior de datos. Para transmitir datos, la CPU sólo necesita verificar que el registro de datos del SCI está vacío antes de escribirlo.

Viendo un ejemplo típico de una transmisión de datos. Se asume que el registro de datos está vacío y la CPU quiere transmitir un carácter a través del transmisor del SCI. La CPU escribe el carácter en el registro de datos del SCI.

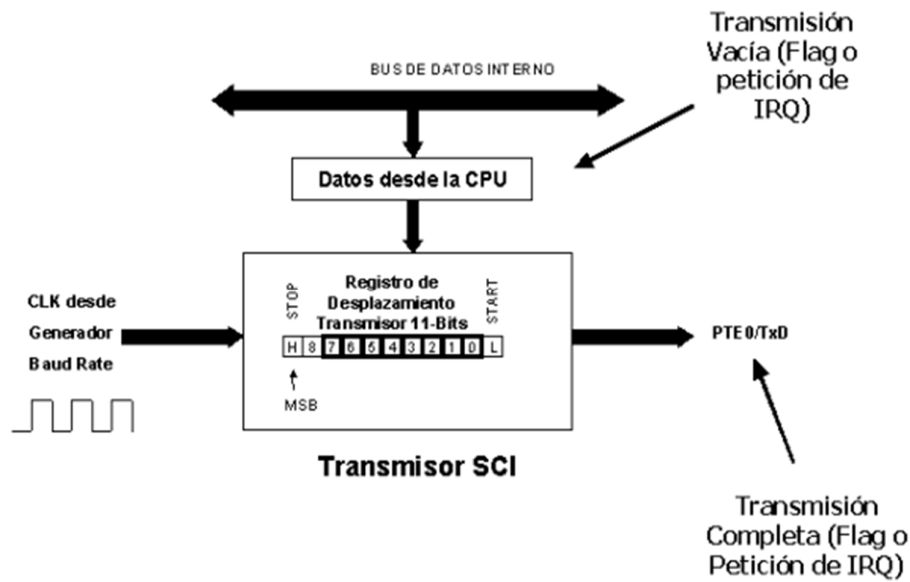


Figura III.29.- Diagrama de bloques de Transmisor SCI

Fuente: <http://arantxa.ii.uam.es/~gdrivera/robotica/hc08/guia68hc08.pdf>

Cuando el registro de desplazamiento está listo para aceptar nuevos datos, el hardware del SCI mueve automáticamente los datos desde el registro de datos del SCI al registro de desplazamiento. Una vez los datos se mueven al registro de desplazamiento, el SCI pone un indicador de transmisor vacío para indicar a la CPU que otro carácter se pueda enviar. Se puede usar el software para consultar el indicador o se puede configurar el SCI para generar una petición de interrupción optativa.

El registro de desplazamiento se bloquea por el auto generador de baudios y desplazan los 11 bits fuera del pin TxD. Una vez desplazado el último bit, el SCI pone el indicador de transmisión completa y opcionalmente genera una

petición de interrupción para el proceso de la CPU. El receptor del SCI también usa un 'buffer' doble para recibir datos.

### 3.10.3.- Recuperación de datos del Receptor

El receptor del SCI incluye un sistema de recuperación de datos que prueba y verifica los datos. Esto asegura la recepción de datos muy fiable. Se puede ver en detalle cómo el receptor maneja un flujo de bits asíncrono que entra en el pin RxD.

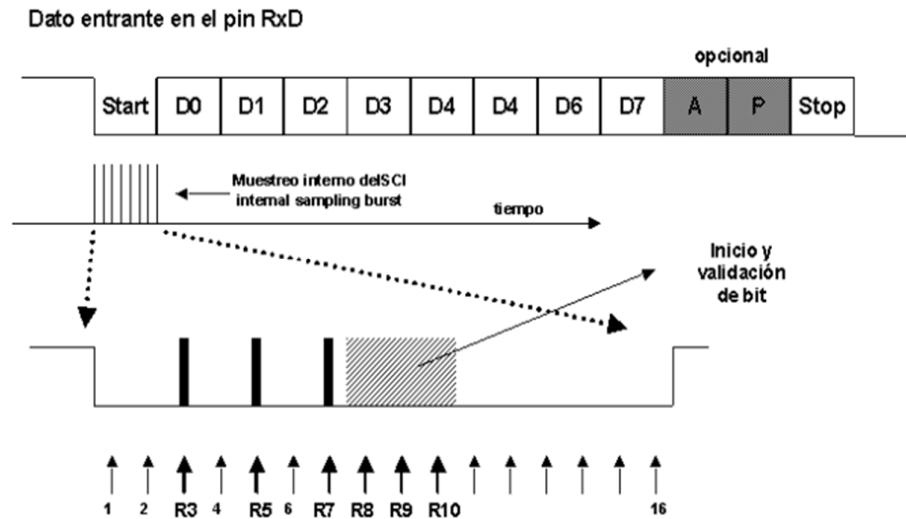


Figura III.30: Recuperación de datos del receptor

Fuente: <http://arantxa.ii.uam.es/~gdrivera/robotica/hc08/guia68hc08.pdf>

El sistema de recuperación de datos prueba el flujo de bits entrante a una velocidad fija de reloj, llamado RT, que es 16 veces la velocidad de transmisión. RT se sincroniza después de cada bit de inicio ('start') y después de cualquier bit de datos que cambia de nivel lógico 1 a 0. La sincronización

adicional hace menos sensible, al módulo SCI, a las diferencias en las velocidades de transmisión del dispositivo enviado y la MCU.

Para detectar un bit de 'start', el receptor hardware del SCI está ejecutando continuamente una búsqueda asíncrona para un nivel lógico 0 precedido por tres valores lógicos 1. Cuando ocurre un flanco de bajada de un posible bit de 'start', el reloj RT empieza a contar en R16.

La técnica de muestreo permite al SCI detectar posibles interferencias o ruidos que se solapan con el flujo de datos entrante. Para verificar la integridad del bit 'start' y para detectar el ruido, la lógica de recuperación de datos toma muestras en R3, R5 y R7 de RT. La verificación del bit 'start' falla si cualquiera de las dos de las tres muestras de la verificación, tiene un valor lógico 1. Cuando la verificación falla, el reloj RT se restablece y la búsqueda empieza de nuevo. Para determinar el valor de un bit de datos y para detectar el ruido, la lógica de recuperación toma muestras en R8, R9 y R10 de RT. Una situación similar ocurre cuando el receptor del SCI espera un bit de 'stop'. En este caso, la lógica de recuperación toma muestras de R8, R9 y R10 de RT, para verificar la integridad del bit y para detectar el ruido.

#### ***3.10.4.- Cálculo de la velocidad de Transmisión del SCI***

La figura siguiente muestra los bloques del generador de 'baud rate' del SCI. El generador de 'baud rate' incluye un prescaler y un divisor de 'baud rate' que se puede programar usando el registro de 'baud rate' del SCI.

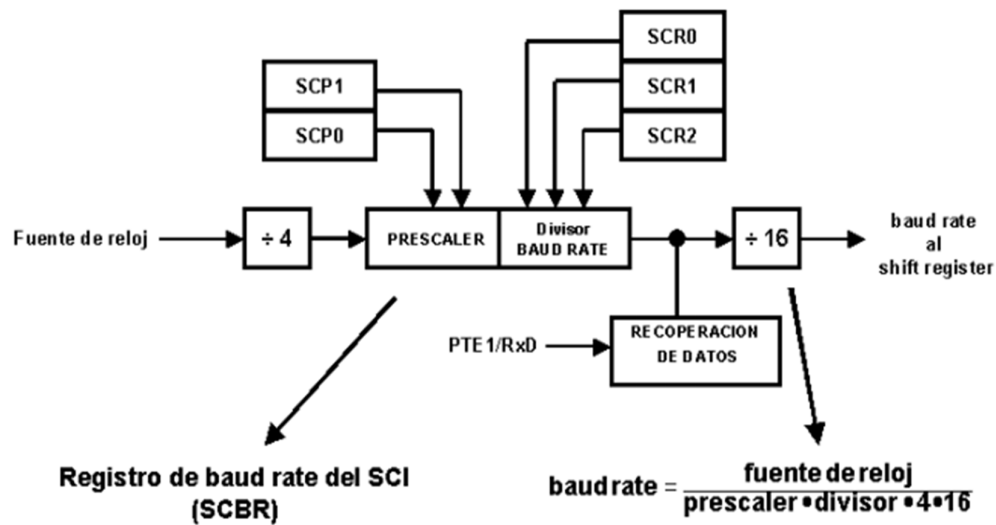


Figura III.31.-Diagrama de bloques del cálculo de velocidad de SCI

Fuente: <http://arantxa.ii.uam.es/~gdrivera/robotica/hc08/guia68hc08.pdf>

La fuente reloj del generador de 'baud rate' varía según el dispositivo. La fuente de reloj del SCI del 68HC08 puede ser la frecuencia del cristal CGMXCLK dividido por cuatro o puede ser la frecuencia del bus interno en dispositivos con PLL o generador de reloj interno. Hay que verificar en el libro de datos técnicos del dispositivo, la aplicación específica del SCI.

El divisor por 4 del reloj de entrada es para bajar la velocidad a propósito. El divisor por 16 que va al registro de desplazamiento ('shift register') permite al receptor realizar el muestreo de datos a 16 veces la velocidad de 'baud rate'. Para calcular el 'baud rate' del SCI, hay que dividir la fuente de reloj por el producto del prescaler, el divisor de 'baud rate', el divisor por 4 y el divisor por 16.



### 3.10.5.- Registro SCBR de velocidad de transmisión

Se puede programar el prescaler y el divisor 'baud rate' usando el registro de 'baud rate' del SCI, SCBR. Se deben seleccionar los valores apropiados para cada aplicación.

Tabla III.V.- Registro de SCBR de Velocidad de Transmisión

<b>SCBR</b>	Bit 7	6	5	4	3	2	1	Bit 0
Leer:			SCP1	SCP0	Reser- vado	SCR2	SCR1	SCR0
Escribir:								
Reset:	0	0	0	0	0	0	0	0

Fuente: <http://arantxa.ii.uam.es/~gdrivera/robotica/hc08/guia68hc08.pdf>

- Los bits SCP1 y SCP0 de 'baud rate' del prescaler del SCI, seleccionan uno de los cuatro valores del prescaler, como el que se muestra en la tabla.

Tabla III.VI.- Registro de SCP1-SCP2 y SCI

<b>SCP1 – SCP0</b>	<b>PRESCALER</b>
00	1
01	3
10	4
11	13

Fuente: <http://arantxa.ii.uam.es/~gdrivera/robotica/hc08/guia68hc08.pdf>

- Los bits SCR2 a SCR0 de selección de 'baud rate' del SCI, seleccionan uno de los ocho divisores, como el que se muestra en la tabla.

Tabla III.VII.- Registro de SCR2 a SCR0 y Divisor de Baud Rate

SCR2 a SCR0	DIVISOR DE BAUD RATE
000	1
001	2
010	4
011	8
100	16
101	32
110	64
111	128

Fuente: <http://arantxa.ii.uam.es/~gdrivera/robotica/hc08/guia68hc08.pdf>

### 3.10.6.- Cálculo del baud rate a 19200

Para generar un 'baud rate' de 19.200 baudios asumiendo una fuente de reloj con una frecuencia de 4.9152 MHz, se calcula de la siguiente manera:

$$\text{baud rate} = \frac{\text{Fuente de Reloj}}{\text{prescaler} \times \text{divisor} \times 4 \times 16} = \frac{4.9152 \text{ MHz}}{1 \times 4 \times 4 \times 16} = 19200 \text{ baudios}$$

La fuente de reloj dividido por el producto del prescaler por el divisor de 'baud rate' por el divisor por 4 y por 16. En este caso, se selecciona como fuente de reloj 4.9152, los bits SCP1-SPC0 con 00 para un valor del prescaler de 1 y los bits SCR2 a SCR0 con 010 para un divisor de 'baud rate' de 4.

## **CAPITULO IV**

### **PERIFERICOS Y BLUETOOTH**

#### **4.1.- PERIFERICOS INTERNOS**

##### **4.1.1.- Un sensor**

Es un dispositivo capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas. Las variables de instrumentación pueden ser por ejemplo: temperatura, intensidad lumínica, distancia, aceleración, inclinación, desplazamiento, presión, fuerza, torsión, humedad, movimiento, pH, etc. Una magnitud eléctrica puede ser una resistencia eléctrica (como en una RTD), una capacidad eléctrica (como en un sensor de humedad), una Tensión eléctrica (como en un termopar), una corriente eléctrica (como en un fototransistor), etc.

Un sensor se diferencia de un transductor en que el sensor está siempre en contacto con la variable de instrumentación con lo que puede decirse también que es un dispositivo que aprovecha una de sus propiedades con el fin de adaptar la señal que mide para que la pueda interpretar otro dispositivo. Como por ejemplo el termómetro de mercurio que aprovecha la propiedad que posee el mercurio de dilatarse o contraerse por la acción de la temperatura. Un sensor también puede decirse que es un dispositivo que convierte una forma de energía en otra.

Áreas de aplicación de los sensores: Industria automotriz, robótica, industria aeroespacial, medicina, industria de manufactura, etc.

Los sensores pueden estar conectados a un computador para obtener ventajas como son el acceso a una base de datos, la toma de valores desde el sensor y un sin número de aplicaciones, que cumplan este requerimiento.

#### ***4.1.1.1.- Sensores según el tipo de señal de entrada.***

Los sensores pueden ser clasificados dependiendo del tipo de señal al cual responden.

- **Mecánica:** Ejemplos: longitud, área, volumen, masa, flujo, fuerza, torque, presión, velocidad, aceleración, posición, acústica, longitud de onda, intensidad acústica.
- **Térmica:** Ejemplos: temperatura, calor, entropía, flujo de calor.

- **Eléctrica:** Ejemplos: voltaje, corriente, carga, resistencia, inductancia, capacitancia, constante dieléctrica, polarización, campo eléctrico, frecuencia, momento dipolar.
- **Magnética:** Ejemplos: intensidad de campo, densidad de flujo, momento magnético, permeabilidad.
- **Radiación:** Ejemplos: intensidad, longitud de onda, polarización, fase, reflectancia, transmitancia, índice de refractancia.
- **Química:** Ejemplos: composición, concentración, oxidación/potencial de reducción, porcentaje de reacción, PH.

#### **4.1.1.2.- Atendiendo al tipo de señal entregada por el sensor**

- **Sensores análogos.**

La gran mayoría de sensores entregan su señal de manera continua en el tiempo. Son ejemplo de ellos los sensores generadores de señal y los sensores de parámetros variables

- **Sensores digitales.**

Son dispositivos cuya salida es de carácter discreto. Son ejemplos de este tipo de sensores: codificadores de posición, codificadores incrementales, codificadores absolutos, los sensores auto resonantes (resonadores de cuarzo, galgas acústicas, cilindros vibrantes, de ondas superficiales (SAW), caudalímetros de vórtices digitales), entre otros.

#### 4.1.1.3.- Atendiendo a la naturaleza de la señal eléctrica generada.

Los sensores dependiendo de la naturaleza de la señal generada pueden ser clasificados en:

- **Sensores pasivos:**

Son aquellos que generan señales representativas de las magnitudes a medir por intermedio de una fuente auxiliar. Ejemplo: sensores de parámetros variables (de resistencia variable, de capacidad variable, de inductancia variable).

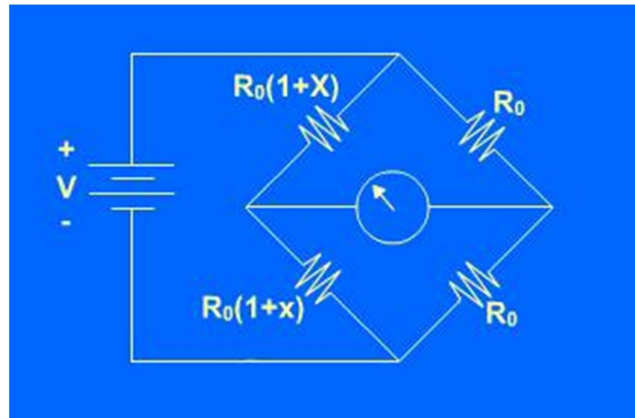


Figura IV.32.-Sensor pasivo del tipo resistivo

Fuente: [www.virtual.unal.edu.co/cursos/sedes/manizales/4040003/lecciones/cap3lecc4.htm](http://www.virtual.unal.edu.co/cursos/sedes/manizales/4040003/lecciones/cap3lecc4.htm)

- **Sensores activos o generadores de señal:**

Son aquellos que generan señales representativas de las magnitudes a medir en forma autónoma, sin requerir de fuente alguna de alimentación. Ejemplo: sensores piezoeléctricos, fotovoltaicos, termoeléctricos, electroquímicos, magnetoeléctricos.

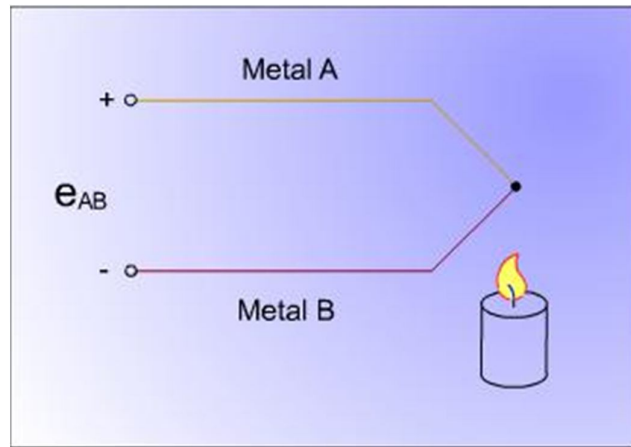


Figura IV.33.-Sensor termoeléctrico

Fuente: [www.virtual.unal.edu.co/cursos/sedes/manizales/4040003/lecciones/cap3lecc4.htm](http://www.virtual.unal.edu.co/cursos/sedes/manizales/4040003/lecciones/cap3lecc4.htm)

## 4.2.- ESTÁNDAR RS-232C

El puerto serie RS-232C, presente en todos los ordenadores actuales, es la forma más comúnmente usada para realizar transmisiones de datos entre ordenadores. El RS-232C es un estándar que constituye la tercera revisión de la antigua norma RS-232, propuesta por la EIA (Asociación de Industrias Electrónicas), realizándose posteriormente una versión internacional por el CCITT, conocida como V.24. Las diferencias entre ambas son mínimas, por lo que a veces se habla indistintamente de V.24 y de RS-232C (incluso sin el sufijo "C"), refiriéndose siempre al mismo estándar.

El RS-232C consiste en un conector tipo DB-25 de 25 pines, aunque es normal encontrar la versión de 9 pines DB-9, más barato e incluso más extendido para cierto tipo de periféricos (como el ratón serie del PC). En cualquier caso, los PCs no suelen emplear más de 9 pines en el conector DB-25. Las señales con las que trabaja este puerto serie son digitales, de +12V (0 lógico) y -12V (1

lógico), para la entrada y salida de datos, y a la inversa en las señales de control. El estado de reposo en la entrada y salida de datos es -12V.

Cada pin puede ser de entrada o de salida, teniendo una función específica cada uno de ellos. Las más importantes son:

Tabla IV.VIII.- Funciones de RS232C

<b>Pin</b>	<b>Función</b>
TXD	(Transmitir Datos)
RXD	(Recibir Datos)
DTR	(Terminal de Datos Listo)
DSR	(Equipo de Datos Listo)
RTS	(Solicitud de Envío)
CTS	(Libre para Envío)
DCD	(Detección de Portadora)

Fuente: <http://www.euskalnet.net/shizuka/rs232.htm>

Las señales TXD, DTR y RTS son de salida, mientras que RXD, DSR, CTS y DCD son de entrada. La masa de referencia para todas las señales es SG (Tierra de Señal). Finalmente, existen otras señales como RI (Indicador de Llamada), y otras poco comunes que no se explican en este artículo por rebasar el alcance del mismo.



Tabla IV.IX.- Pines de DB-25 y DB-9

Numero	de Pin	Señal	Descripción	E/S
<b>En DB-25</b>	<b>En DB-9</b>			
1	1	-	<b>Masa chasis</b>	-
2	3	TxD	<b>Transmit Data</b>	S
3	2	RxD	<b>Receive Data</b>	E
4	7	RTS	<b>Request To Send</b>	S
5	8	CTS	<b>Clear To Send</b>	E
6	6	DSR	<b>Data Set Ready</b>	E
7	5	SG	<b>Signal Ground</b>	-
8	1	CD/DCD	<b>(Data) Carrier Detect</b>	E
15	-	TxC(*)	<b>Transmit Clock</b>	S
17	-	RxC(*)	<b>Receive Clock</b>	E
20	4	DTR	<b>Data Terminal Ready</b>	S
22	9	RI	<b>Ring Indicator</b>	E
24	-	RTxC(*)	<b>Transmit/Receive Clock</b>	S

Fuente: [www.euskalnet.net/shizuka/rs232.htm](http://www.euskalnet.net/shizuka/rs232.htm)

Conector DB 25	Conector DB 9
----------------	---------------

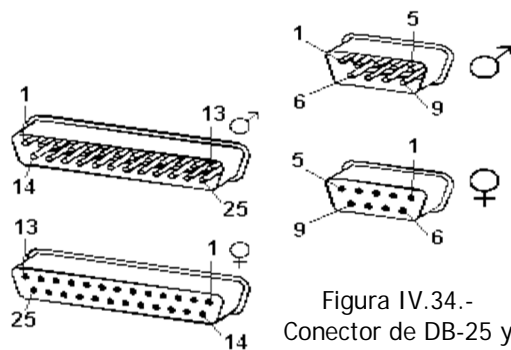


Figura IV.34.-  
Conector de DB-25 y  
DB-9

Figura IV.34.- Conector de DB-25 y DB-9

Fuente: [www.euskalnet.net/shizuka/rs232.htm](http://www.euskalnet.net/shizuka/rs232.htm)

#### **4.2.1.- Puertos serie modernos**

Uno de los defectos de los puertos serie iniciales era su lentitud en comparación con los puertos paralelos -hablamos de 19.2kbits por segundo- sin embargo, con el paso del tiempo, están apareciendo multitud de puertos serie de alta velocidad que los hacen muy interesantes ya que presentan las ventajas del menor cableado y solucionan el problema de la merma de velocidad usando un mayor apantallamiento, y más barato, usando la técnica del par trenzado. Por ello, el puerto RS-232, e incluso multitud de puertos paralelos, se están sustituyendo por los nuevos puertos serie como el USB, el FireWire o el Serial ATA.

#### **4.2.2.- Tipos de comunicación en serie**

- **Simplex**

En este caso el emisor y el receptor están perfectamente definidos y la comunicación es unidireccional. Este tipo de comunicaciones se emplean, usualmente, en redes de radiodifusión, donde los receptores no necesitan enviar ningún tipo de dato al transmisor.

- **Duplex, half duplex o semi-duplex**

En este caso ambos extremos del sistema de comunicación cumplen funciones de transmisor y receptor y los datos se desplazan en ambos sentidos pero no de manera simultánea. Este tipo de comunicación se utiliza habitualmente en la interacción entre terminales y una computadora central.

- **Full Duplex**

El sistema es similar al duplex, pero los datos se desplazan en ambos sentidos simultáneamente. Para que sea posible ambos emisores poseen diferentes frecuencias de transmisión o dos caminos de comunicación separados, mientras que la comunicación semi-duplex necesita normalmente uno solo. Para el intercambio de datos entre computadores este tipo de comunicaciones son más eficientes que las transmisiones semi-dúplex.letty

### **4.3.- TECNOLOGIA INALAMBRICA BLUETOOTH**

#### ***4.3.1.- Introducción***

Estándares de redes inalámbricas 802.11b y como Wi-Max suelen centrarse en la prestación de PC a PC o de conectividad de PC a ISP en el rango de un edificio o un área metropolitana. Sin embargo, muchas aplicaciones tienen requerimientos de rango mucho menos estrictas, tales como la conexión de periféricos de forma inalámbrica a un dispositivo móvil o la adición de componentes a un sistema de cine en casa. Redes de área personal (PAN) son una opción perfecta para estas aplicaciones: ofrecen rangos de señal en las proximidades de 1 m hasta 100 m, y una amplia variedad de tipos de datos. Además, dado que los tipos de dispositivos que queremos dotar de PAN suelen ser móvil y ligero, la energía es un bien escaso. Por lo tanto, el bajo consumo de energía de radios PAN es muy importante para su aceptación.

Bluetooth, ya ha sido ampliamente desplegado en cientos de millones de dispositivos. Ofrece velocidades de datos de hasta 3 Mbps y distancias de

hasta 100 m, con un consumo de energía mucho menor que 802.11b. Su capa de middleware se basa en la parte superior de las capas PHY y MAC para proporcionar un alto grado de interoperabilidad entre dispositivos equipados con Bluetooth. Este bajo consumo de energía y garantizar la interoperabilidad han impulsado la aceptación de Bluetooth en la comunidad móvil.

Esta tecnología, 802.15.4, va incluso más allá de Bluetooth en la velocidad de intercambio de poder. 802.15.4 ofrece velocidades de datos de hasta 250 kbps y puede soportar fácilmente los vínculos con un ciclo de trabajo muy bajo. Por lo tanto, es adecuado para el despliegue de los dispositivos alimentados por baterías que deben sobrevivir por hasta un año entre cargas. 802.15.4 ya ha encontrado una amplia aceptación en la comunidad de redes de sensores, pero aún es demasiado pronto para predecir su futuro en otros mercados (como los dispositivos de automatización del hogar).

Radios de banda ultra-ancha (UWB) emiten de baja potencia, los pulsos de alto ancho de banda que ofrecen velocidades de datos comparables de cable Ethernet. Sus altas tasas de datos y el consumo de energía relativamente bajo, lo hacen ideal para la sustitución de enlaces por cable corto, como las que se encuentran entre los periféricos de PC, y en cine en casa.

Desafortunadamente, la estandarización de UWB de IEEE ha fallado, lo que resulta en dos normas incompatibles: DS-UWB, abogó por el Foro UWB, y MB-OFDM, abogó por la Alianza WiMedia. Como 802.15.4, UWB está todavía en sus primeras etapas de implementación, por lo que es difícil predecir el éxito futuro.

Podemos destacar las diferencias en los principios de señalización y las diversas propiedades de los enlaces inalámbricos. En su caso, también se describen las capas de soporte que se sientan en la parte superior de la capas PHY y MAC. Las características de seguridad de cada tecnología, describimos algunas aplicaciones actuales y predecir sus perspectivas.

#### **4.3.2.- Estándar 802.15.1 (Bluetooth)**

802.15.1, más comúnmente conocida como Bluetooth, es un estándar de red inalámbrica de baja velocidad de datos, bajo consumo de energía destinado a sustituir los cables entre los dispositivos ligeros [IEEE802.15.1] . La pila de protocolos Bluetooth, se muestra en la Figura VI.36, es algo inusual en comparación con otras pilas de redes IEEE. La pila de Bluetooth define muchos de los componentes por encima de la PHY y MAC capas, algunos de los cuales son opcionales. Este diseño permite que el Grupo de Interés Especial de Bluetooth para componer estos componentes en los perfiles específicos de la aplicación, como se discute a continuación. En esta sección, le daremos una breve descripción de algunos de estos componentes.

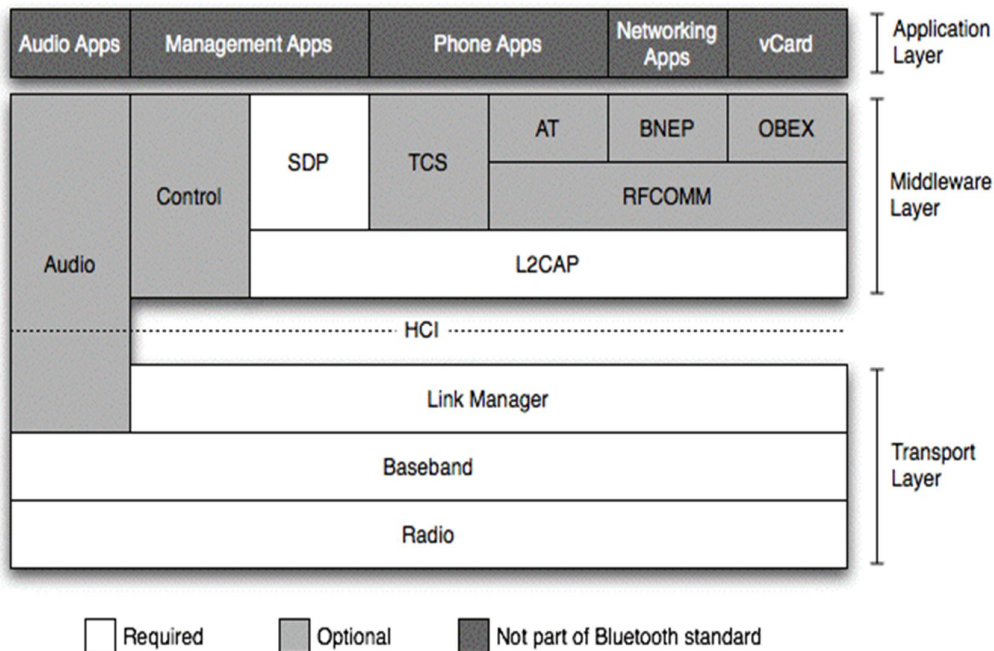


Figura IV.35.- Pila de protocolos Bluetooth

Fuente: [www.seguridadmobile.com/bluetooth/especificacion-bluetooth/estandar-bluetooth/](http://www.seguridadmobile.com/bluetooth/especificacion-bluetooth/estandar-bluetooth/)

#### 4.3.2.1.- Capa de Transporte

La capa de transporte Bluetooth es más o menos equivalente a la tradicional OSI capas PHY y MAC. Se requiere que todos los dispositivos Bluetooth para aplicar esta capa de hardware. La capa de transporte se compone de la radio, de banda base, y las capas de gestor de enlaces, que se describen a continuación.

#### 4.3.2.2.- Radio Capa

La capa de radio dicta la frecuencia, potencia, y la modulación utilizado por antenas Bluetooth. Bluetooth ocupa 79 canales de 1 MHz cada uno en el espectro de 2,4 GHz, de 2,402 GHz a 2,480 GHz. Los dispositivos utilizan sólo uno de estos canales a la vez, saltando entre ellos como se describe a

continuación. También hay que protegerse bandas reservadas en cada extremo del espectro, a 2,400 GHz-2,402 GHz y 2,480 GHz a 2,484 GHz-. En las revisiones iniciales de Bluetooth, todos los dispositivos utilizan modulación BPSK, lo que ofrece una velocidad de datos máxima de 1 Mbps, o alrededor de 723 kbps cuando se toma toda la sobrecarga de paquete en cuenta. (Como se discute a continuación, Bluetooth 2.0 añade modulaciones opcionales que apoyan a aumentar las tasas de datos.) Receptores Bluetooth están obligados a tener una tasa de error de bit (BER) de 0,1% o menos.

Dispositivos Bluetooth se dividen en una de las tres "clases", que especifican la potencia de salida de la antena. Los dispositivos de Clase 1 Mensaje con 1 mW a 100 mW de potencia, dispositivos de Clase 2 difunden utilizando 0,25 mW y 2,5 mW de potencia y dispositivos de Clase 3 emiten con un máximo de 1 mW de potencia. Los dispositivos de Clase 1 deben ser capaces de cambiar su potencia en incrementos de 2 dB a 8 dB, control de potencia es opcional para la Clase 2 y Clase 3 dispositivos. Estas clases tienen rangos de potencia de señal de unos 100 metros, 10 metros y 1 metro, respectivamente.

#### **4.2.1.3 Baseband y Link Layers**

En la capa de banda base, los dispositivos Bluetooth se forman en pico redes y/o redes dispersas. Piconets consisten en un dispositivo maestro que se comunica directamente con hasta 7 dispositivos esclavos activos. Piconets también pueden tener hasta 250 estacionado (es decir, inactivo) nodos esclavos en un momento dado. Múltiples piconets también se pueden combinar en una sola red dispersa de saltos múltiples.

La comunicación dentro de una piconet se realiza directamente a través del enlace de un salto entre un maestro y un esclavo, esclavos no pueden comunicarse directamente. Bluetooth utiliza un dúplex (TDD) esquema básico de división de tiempo, donde el tiempo se divide en ranuras de 625 microsiemens. El maestro puede comunicarse con un esclavo en las ranuras impares, y los esclavos responder durante los intervalos pares. Cada paquete puede consumir 1, 3 ó 5 slots. Después de cada paquete, la piconet salta a un canal de Bluetooth diferentes; frecuencia del siguiente canal se determina utilizando un generador de números pseudo-aleatorios.

Canales Piconet pueden subdividirse en enlaces que utilizan uno de los cinco tipos de transportes. Los slots de transporte síncrono orientado a conexión (SCO) reservas para la comunicación sincrónica y se utiliza con mayor frecuencia para reservar 64 kbps enlaces para datos de voz. Enlaces mejorados SCO (ESCO) también se reserva ranuras adicionales para la retransmisión de paquetes; enlaces eSCO también pueden robar las ranuras no utilizadas retransmisión otros enlaces. La mayoría de los enlaces utilizan la comunicación lógica orientada a la conexión asíncrona (ACL), donde se transmiten los paquetes durante las franjas horarias reservadas restantes. Por último, la emisión esclavo activo (ASB) y emisión esclavo aparcado (PSB) enlaces se utilizan para el maestro para enviar datos de control a los esclavos activos y aparcados, respectivamente. Todos los enlaces, además de ASB y el PSB son fiables: cada paquete debe ser reconocido individualmente por el receptor, y los paquetes se adjunta con su CRC para garantizar la coherencia.



Los enlaces pueden usar opcionalmente 2/3 de corrección de errores hacia adelante (FEC).

Bluetooth define tres modos de ahorro de energía. En el modo de espera, los dispositivos sólo manejan espacios reservados para los enlaces síncronos, y dormir el resto del tiempo. En el modo de aspiración, el dispositivo se queda dormido la mayor parte del tiempo, despertar periódicamente (de cada 1,25 ms para cada 40,9 s) para comunicarse. Por último, en el modo de estacionado, el dispositivo se apaga sus vínculos con el dispositivo maestro, excluyendo el enlace PSB. El dispositivo maestro puede despertar dispositivos aparcados por balizamiento ellos a través del enlace PSB.

#### **4.2.1.4 Middleware Capa**

La capa de middleware consta de varios componentes de software que están diseñados para fomentar la interoperabilidad entre los dispositivos Bluetooth. Muchos de los componentes en esta capa son opcionales, por lo general, sólo los dispositivos de alta potencia (como PCs) implementarán toda la pila. Los componentes de la capa de middleware se comunican con la capa de transporte con el Host Controller Interface estandarizada (HCI). Algunos de estos componentes incluyen:

- Control de enlace lógico y protocolo de adaptación (L2CAP): proporciona TCP y UDP-como características de ACL enlaces
- RFCOMM: emula enlaces infrarrojos IrDA en la parte superior de L2CAP
- Especificación del protocolo de control de telefonía (TCS): operaciones de telefonía controles

- AT: controla las operaciones del teléfono con el legado Hayes ("AT") conjunto de comandos
- Protocolo de encapsulamiento de red Bluetooth (bnep): Ethernet paquetes encapsulados en paquetes Bluetooth
- Objeto Protocolo Exchange (OBEX): compatible con funciones de sincronización de objetos de IrDA

En el interés de promover la interoperabilidad entre dispositivos Bluetooth, Bluetooth SIG también define perfiles, que proporcionan protocolos universales para las tareas específicas de la aplicación común. Cada perfil se crea mediante la combinación de componentes Bluetooth con el software específico del perfil. Por ejemplo, el perfil de transferencia de archivos (FTP), que permite que los dispositivos equipados con Bluetooth para intercambiar archivos, consiste en un protocolo cliente / servidor especificado construido en la cima de la Radio, banda base, Link Manager, L2CAP, RFCOMM, SDP y OBEX componentes. Al escribir estas líneas, el SIG de Bluetooth define 24 perfiles estándar.

#### ***4.3.3.- Topología de red bluetooth***

A diferencia de otras tecnologías LAN inalámbricas, como IEEE 802.11 (Wi-Fi), diseñadas para dispositivos que se hallen dentro o en los alrededores de un mismo edificio, los dispositivos que utilicen las redes PAN inalámbricas IEEE 802.15, incluyendo Bluetooth, podrán comunicarse en cualquier parte del mundo de forma stand-alone, incluso a bordo de un barco o avión y sin

necesidad de utilizar equipo hardware adicional, como puntos de acceso.

Cuando un dispositivo Bluetooth está dentro del radio de cobertura de otro, pueden establecer un enlace entre ellos. Hasta ocho unidades Bluetooth pueden comunicarse entre ellas y formar lo que se denomina una Piconet o Picorred. La unión de varias piconets se denomina Scatternet o Red Dispersa.

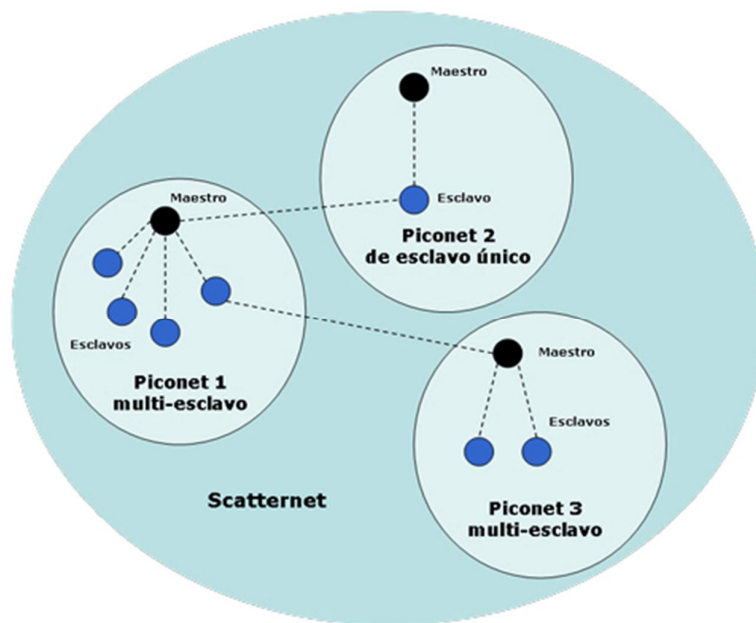


Figura IV.36.- Scatternet o Red Dispersa

Fuente: [www.seguridadmobile.com/bluetooth/especificacion-bluetooth/estandar-bluetooth/](http://www.seguridadmobile.com/bluetooth/especificacion-bluetooth/estandar-bluetooth/)

Los dispositivos dentro de una piconet juegan dos papeles: maestro o esclavo. En todas las piconets sólo puede haber una unidad maestro, que normalmente es quien inicia la conexión, el resto de unidades Bluetooth en la piconet se denominan esclavos.

Cualquier dispositivo puede realizar las funciones de maestro y esclavo, pero un mismo dispositivo únicamente puede ser maestro de una piconet.

El maestro es el dispositivo de una piconet cuyo reloj y patrón de saltos se utilizan para sincronizar a todos los demás dispositivos esclavos. Todas las unidades que participan en una piconet están sincronizadas desde el punto de vista del tiempo y de la secuencia de saltos entre canales. Cada unidad dispone de un reloj de sistema interno que determina la temporización y la secuencia de saltos que debe seguir el transceptor.

La topología Bluetooth se puede describir como una estructura de piconets múltiples. Dado que la especificación Bluetooth soporta tanto conexiones punto a punto como punto a multipunto, se pueden establecer y enlazar varias piconets en forma de scatternet. Las piconets pertenecientes a una misma scatternet no están coordinadas y los saltos de frecuencia suceden de forma independiente, es decir, todos los dispositivos que participan en la misma piconet se sincronizan con su correspondiente tiempo de reloj y patrón de saltos determinado. El resto de piconets utilizarán diferentes patrones de saltos y frecuencias de relojes distintas, lo que supone distintas velocidades de salto entre canales. Aunque no se permite la sincronización de diferentes piconets, los dispositivos pueden participar en diferentes piconets gracias a una multiplexación por división de tiempo (TDM). Esto permite a un dispositivo participar de forma secuencial en diferentes piconets, estando activo en sólo una piconet cada vez.

#### ***4.3.4.- Seguridad de bluetooth***

Bluetooth utiliza un proceso de emparejamiento para establecer el cifrado y la autenticación entre dos dispositivos. El proceso de sincronización se realiza

mediante una serie de teclas como una entrada a la SAFER bloque cifrado +. En aras de la brevedad, no vamos a discutir aquí el funcionamiento interno de SAFER + o la generación de claves en detalle, el lector interesado puede consultar [Shaked05] para más información.

En la primera etapa del proceso de emparejamiento, los dispositivos generan una clave de inicialización de 128 bits compartida utilizando la dirección de hardware de 48 bits del maestro; un número aleatorio de 128 bits compartido, y un PIN especificado por el usuario de hasta 128 bits. Los dos primeros valores se intercambian en texto plano, y el segundo se introducen manualmente por el usuario en ambos dispositivos. Los dispositivos se cifran su vínculo con esta clave, y negocian una clave de enlace de 128 bits. Los dos dispositivos a continuación, utilizan la clave de enlace para llevar a cabo un desafío / respuesta de protocolo. Si tiene éxito, entonces los dos dispositivos almacenan la clave de enlace y descartan la clave de inicialización. Cualquier futura comunicación entre estos dos dispositivos se cifra usando opcionalmente una clave de sesión de 128 bits basada en esta clave de enlace almacenado.

Desafortunadamente, no es posible explotar las debilidades en algunas implementaciones de Bluetooth con el fin de atacar a los enlaces encriptados [Shaked05]. Si un atacante olfatea los paquetes utilizados para emparejar dos dispositivos y se puede averiguar el PIN secreto, la clave de enlace puede ser fácilmente calculada. Aunque este tipo de ataque es poco práctico cuando se utiliza un PIN 128 bits, muchos dispositivos Bluetooth sólo permiten PIN construyen utilizando 1 a 4 dígitos. Estos pines son muy débiles y pueden ser fácilmente vulnerados, los investigadores fueron capaces de romper un PIN de

4 dígitos en 63 ms en una GHz Pentium 4 CPU 3. Afortunadamente, este ataque es algo difícil para un atacante ocasional de lograr, ya que él debe ser capaz de rastrear los paquetes enviados durante el proceso de sincronización. Sin embargo, un atacante determinado puede ser capaz de forzar a los dos dispositivos de asociar de nuevo bajo demanda utilizando hardware personalizado.

También ha habido muchos ataques específicos del proveedor de los dispositivos Bluetooth [Zetter04] . Muchos de los primeros teléfonos móviles con capacidad Bluetooth permite a los atacantes acceder a algunos componentes Bluetooth sin antes emparejamiento con el dispositivo. Estos ataques, conocidos como "bluebugging" y "Bluesnarfing", han permitido a los investigadores realizar llamadas en los teléfonos móviles de otras personas, leer sus libros de direcciones, y escuchar sus conversaciones telefónicas. (Por razones de seguridad, los investigadores involucrados no han revelado públicamente los métodos exactos utilizados para llevar a cabo estos ataques.) Dado que estos ataques aprovechan fallos específicos del proveedor en las implementaciones de la pila de Bluetooth, no son un ataque contra el protocolo Bluetooth per se. Sin embargo, estos ataques ponen de relieve la necesidad de mostrar cautela en la aplicación de las pilas de red inalámbrica, y los posibles problemas causados cuando una aplicación no está lista para el consumo público.

## **CAPITULO V**

### **IMPLEMENTACION DE SISTEMA TELEMETRICO CON BLUETOOTH**

La instalación y adecuación de la ECU y los diferentes sensores instalados en el automóvil de competencia Mazda 323 fue realizado por el personal que labora en el Taller Automotriz “Álvarez”. Bajo la dirección del Ingeniero Diego Álvarez, docente de la Escuela de Ingeniería Automotriz de la ESPOCH. El procedimiento de diseño, instalación y pruebas se explica en los siguientes sub-capítulos a excepción de las adaptaciones y tratamientos del motor que es parte del vehículo objeto de estudio.

### **5.1.- DISEÑO DEL SISTEMA DE TELEMETRIA CON TEGNOLOGIA BLUETOOTH**

El Sistema de Transmisión y Recepción de datos, es diseñado con el fin de obtener las mediciones de los diferentes tipos de sensores ubicados en el vehículo de competencia Mazda 323, mediante la utilización de un sistema microprocesado que se comunica por Bluetooth con una Estación Central, en donde se almacenan las mediciones realizadas.

Para el diseño del Sistema de adquisición de datos, se debe tomar en cuenta muchos factores tanto en calidad de los elementos electrónicos como de factibilidad de desarrollo e implementación del mismo dentro de un escenario real.

En la figura V.37 se indica un diagrama de bloques que constituyen el Sistema Transmisión y Recepción de datos del vehículo de competencia Mazda 323.

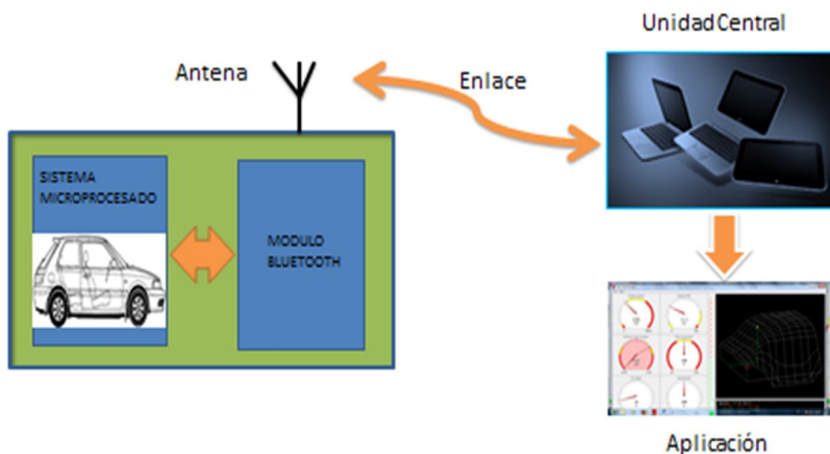


Figura V.37.- Sistema de Telemetría por Bluetooth

Fuente: El Autor



### 5.1.1.- Configuración del modulo bluetooth HC-06

Una vez conseguido el modulo Bluetooth HC-06 analicemos las características que se consideran importantes:

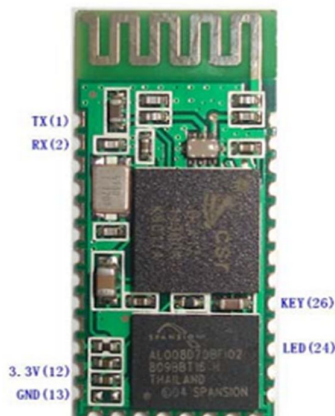


Figura V.38.- Modulo Bluetooth HC-06

Fuente: [aldprojects.blogspot.com/2012/10/configurando-el-modulo-bluetooth-hc-06.html](http://aldprojects.blogspot.com/2012/10/configurando-el-modulo-bluetooth-hc-06.html)

- Denominaremos a este modulo un dispositivo bluetooth ESCLAVO y a las computadoras o Estación Central a las que se conecte se llaman bluetooth HOST
- Este modulo que teniendo el firmware linvor solo funciona como ESCLAVO es decir solo puede conectarse con otros dispositivos Bluetooth que sean HOST (Laptops, Celulares, PC de escritorio con conectividad bluetooth y otros con capacidades HOST)
- Este modulo solo usa el servicio SPP ( Serial Port Protocol ), este modulo solo funciona con este tipo de servicio así que solo podríamos usar dispositivos HOST que usan este servicio, en cuanto a computadoras todas son compatibles con el SPP y si poseemos una

Tablet o celular con Android, deberíamos obtener la aplicación para este tipo de dispositivos.

- Este modulo Bluetooth solo nos da una interface de puerto serial, es como tener un cable de puerto serial entre nuestro circuito y el dispositivo HOST.
- La única forma de configurar los parámetros es con comandos AT mientras el modulo no esta conectado a algún HOST.
- Su comunicación serial usa niveles TTL y no RS232 es decir NO NECESITAMOS UN MAX232 para convertir las tensiones.
- Se debe conectar de forma cruzada con el circuito o pc, es decir: TXD->RXD y RXD->TXD.

Para la configuración para esto es necesario que el modulo bluetooth no este conectado a algún otro dispositivo.

Cabe mencionar que si lo conectamos por bluetooth el dispositivo no nos va a dejar modificar sus parámetros debido a que la comunicación de baudrate es por defecto de fábrica de 9600bps.

Entonces nos conectamos al modulo con la computadora pero la pregunta es ¿como?, no todos tenemos puerto serial pues NO es recomendable usar el cable conversor de serial a usb o el puerto serial de la computadora porque: estos funcionan con RS232 es decir tensiones de -12V y 12V tendríamos que convertir los TTL del modulo 5V y 0V a estos otros usando el MAX232 y unos cuantos condensadores, es por esto recomiendo usar un conversor.



Figura V.39.- Modulo conversor de serial a usb que funciona con niveles TTL

Fuente: El Autor

Tendremos la siguiente configuración también es útil si usas el MAX232 u otras opciones.

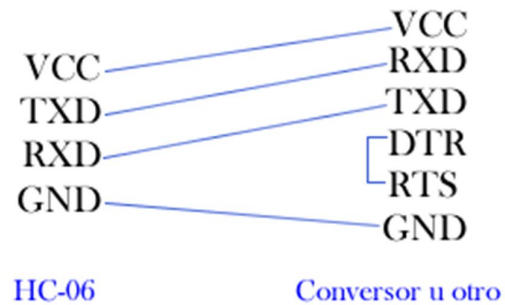


Figura V.40.- Modo de Conexión

Fuente: El Autor

Una vez conectado el modulo bluetooth, se debe ingresar los comandos AT a nuestro modulo, usando el hyperterminal con las instrucciones o comandos siguientes:

Tabla V.10.- Comandos para la configuración del modulo

<b>Comando</b>	<b>Respuesta</b>	<b>Descripción</b>
AT	OK	Comando de testear la comunicación
AT+VERSION	OKlinvorV1.5	Comando para obtener la versión del firmware
AT+BAUDx	OKbaudrate	Comando para cambiar el baudrate de la comunicación serial Donde x puede tomar los siguientes valores: 1200bps si x=1 2400bps si x=2 4800bps si x=3 9600bps si x=4 19200bps si x=5 38400bps si x=6 57600bps si x=7 115200bps si x=8 230400bps si x=9
AT+NAMEname	OKsetname	Comando para cambiar el nombre con el que tiene ante otros dispositivos el dispositivo bluetooth donde "name" es el nuevo nombre SOLO SE PERMITEN 20 LETRAS
AT+PINxyzw	OKsetpin	Comando para cambiar el pin o la contraseña para la conexión donde xyzw son 4 números que identifican el pin
AT+PN AT+PO AT+PE	OK None OK Odd OK Even	Estos tres comandos cambian la paridad de la comunicación serial donde: AT+PN equivale a paridad NONE(ninguna) AT+PO equivale a paridad ODD(impar) AT+PE equivale a paridad EVEN(par)

Fuente: <http://aldprojects.blogspot.com/2012/10/configurando-el-modulo-bluetooth-hc-06.htm>

Con la ayuda de esta tabla de equivalencias se tomaría en cuenta parámetros como cambiar el nombre, el tipo de paridad y otros pero el mas importante es el AT+BAUND, debido a que este comando es el que relaciona el modulo con la aplicación en la transmisión y recepción de datos.

### **5.1.2.- Descripción técnica de bluetooth**

Bluetooth incorpora las siguientes especificaciones técnicas:

- La frecuencia de radio con la que trabaja se sitúa en el rango de 2.4 a 2.48 GHz de la banda ISM (Industrial, Scientific and Medical) disponible a nivel mundial y que no requiere licencia de operador, lo que significa una compatibilidad universal entre dispositivos Bluetooth. Con el fin de evitar interferencias con otros protocolos que operen en la misma banda de frecuencias, Bluetooth emplea la técnica de salto de frecuencias (FHSS, Frequency Hopping Spread Spectrum), que consiste en dividir la banda en 79 canales (23 en España, Francia y Japón) de longitud 1 MHz y realizar 1600 saltos por segundo.
- La capacidad de transmisión varía según versiones del núcleo:
  - Versión 1.1: 723.1 Kbps
  - Versión 1.2: 1 Mbps
  - Versión 2.0 + EDR: 2.1 ~ 3 Mbps
- La potencia de transmisión se divide en 3 clases de productos:
  - Clase 1: 100 mW / 20 dBm, con un rango de ~100 m.
  - Clase 2: 2.5 mW / 4 dBm, con un rango de ~10 m.
  - Clase 3: 1 mW / 0 dBm, con un rango de ~1 m.
- La tecnología Bluetooth se implementa en transceptores de corto alcance.

- El protocolo de banda base es una combinación de conmutación de circuitos y paquetes que la hace apropiada para voz y datos.
- Se definen dos tipos de enlaces para soportar aplicaciones de voz y datos:
  - Enlace asíncrono sin conexión (ACL, Asynchronous Connectionless):
    - Conexiones simétricas o asimétricas punto-multipunto entre maestro y esclavo.
    - Conexión utilizada para tráfico de datos.
    - Sin garantía de entrega, se retransmiten paquetes.
    - La máxima velocidad de envío es de 721 Kbps en una dirección 57.6 Kbps en la otra.
  - Enlace síncrono orientado a conexión (SCO, Synchronous Connection-Oriented):
    - Conexiones simétricas punto a punto entre maestro y esclavo.
    - Conexión capaz de soportar voz en tiempo real y tráfico multimedia.
    - Velocidad de transmisión de 64 KB/s

A partir de la versión 1.0, que se ratificó en julio de 1999, se han publicado sucesivas versiones:

- Versión 1.1:
  - Soluciona erratas de la especificación 1.0.
  - Añade el Indicador de Calidad de Señal Recibida (RSSI)
- Versión 1.2:
  - Implementa la técnica de salto en frecuencia, Adaptive Frequency Hopping, para mejorar la resistencia a interferencias.
  - Introduce el tipo de enlace para aplicaciones de audio extended Synchronous Connections (eSCO) que mejora la calidad de voz.
  - Mejoras en el Host Controller Interface (HCI) para una sincronización más rápida de las comunicaciones.
- Versión 2.0:
  - Nueva versión compatible con la anterior 1.x.
  - Incorpora la tecnología Enhanced Data Rate (EDR), que incrementa las velocidades de transmisión hasta 3 Mbps.
  - Reducción del consumo de energía a pesar del incremento de velocidad.

### 5.1.3.- Comandos ALD

Esta aplicación nos permite manejar código de AT para la configuración del modulo bluetooth:

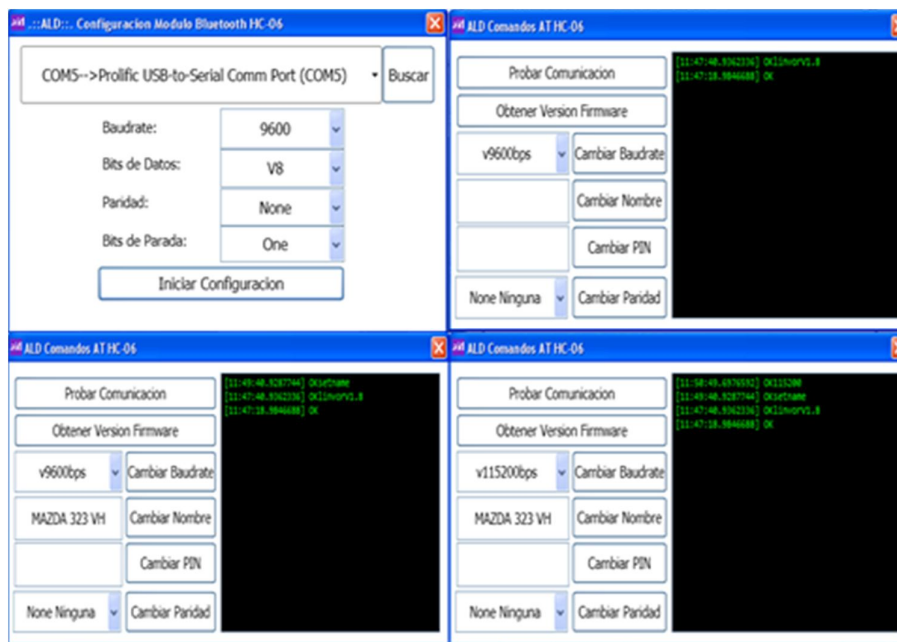


Figura V.40.- Conectividad al modulo, configuración de Baudrate y cambio de Nombre

Fuente: El Autor

Es necesario conocer la velocidad de transferencia de datos que maneja el puerto serial de nuestra ECU, para el caso de la comunicación nosotros nos basamos en la transmisión establecida del manual de megasquirt versión 2.0.

Considerando:

- El tiempo de envío de comandos debe ser mayor a 1 segundo, ya que el modulo junta todos los caracteres que le llegan en 1 segundo y luego los interpreta el mismo.
- El formato de cadena para el envío de comandos es ASCII.



## 5.2.- ENLACE BLUETOOTH

El enlace Bluetooth entre dispositivos, comprende la comunicación inalámbrica a corta distancia por medio de servicios o perfiles que este estándar provee.

Dicho enlace es realizado mediante una técnica de transmisión de datos, mediante saltos de frecuencia, para obtener un nivel de seguridad aceptable dentro de una banda libre.



Figura V.41.- Enlace bluetooth con vehículo de competencia Mazda 323

Fuente: El Autor

Para realizar el enlace es necesario verificar el puerto, enlazar el programa u aplicación realizada en Windows o en Linux, debido a que el bluetooth necesita saber el camino de comunicación para la transferencia de datos, una vez señalado el camino se necesita obtener información de la ECU como son Temperatura del motor, Temperatura del Ambiente, Las revoluciones por

minuto del motor entre otros parámetros que se envía de la estación central hacia la ECU y viceversa. Debido a que los datos son en tiempo real se debe de realizar un análisis del microprocesador para determinar las entradas y salidas de los datos hacia el microprocesador, por lo que dicho estudio esta realizado en el Capitulo III de este proyecto, en donde se detalla las mediciones y comparaciones de los pines de entrada y salida del procesador.

Una vez obtenida la información del procesador se procede a la medición de la señal de Transmisión y Recepción de la ECU para poder agregar nuestro modulo y realizar la transmisión Inalámbrica, para ello se obtiene datos del mismo procesador mediante la visualización y con la ayuda de un scanner se obtiene la información de transmisión y recepción, para luego conectarle al modulo bluetooth.

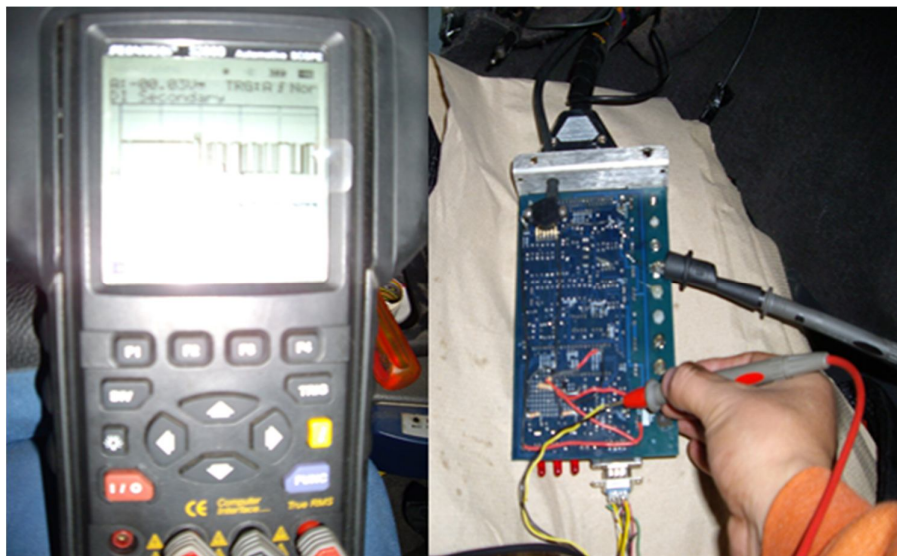


Figura V.42.- Comprobación de señal Transmisión y Recepción en la ECU

Fuente: El Autor

Una vez obtenido este tipo de señal se puede relacionar con las señales obtenidas de la conexión con el modulo bluetooth, debido a que la ECU maneja el puerto serial DB-9 por lo que se conecta directamente a este; se debe tomar en cuenta los voltajes que genera el conector DB-09 con los datos a la salida del puerto:

Tabla V.XI.- Referencia de Voltajes medidos en DB-09 de la ECU

PINES	VOLTAJE
Pin 1	5.02V
Pin 2	-9.69V
Pin 3	-0.03V
Pin 5	0V
Pin 9	5.02V

Fuente: El Autor

Tomando en cuenta los voltajes de referencia, se nota claramente que tenemos voltajes negativos, por lo que si se quiere conectarlo directamente se tiene que minimizar estos, por que el modulo bluetooth trabaja con voltajes positivos o negativos pero en el rango de 0V a 5V.

El siguiente diseño electrónico permite minimizar los voltajes negativos y toma voltajes positivos y negativos dentro del rango que acepta el modulo bluetooth:

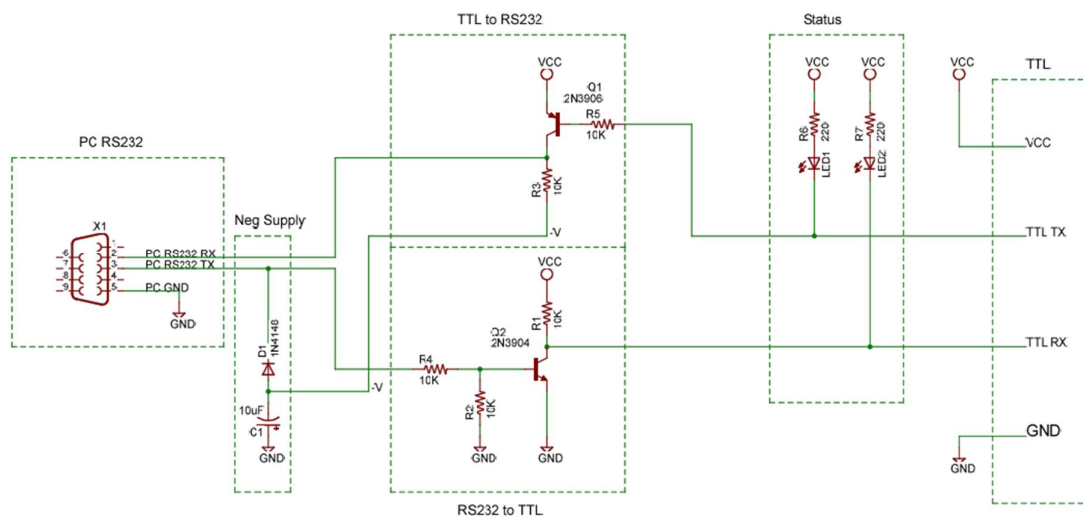


Figura V.43.- Circuito de TTL a RS232

Fuente: El Autor

### 5.2.1.-TTL a RS232

El segundo transistor es un tipo PNP y se encenderá cuando su base está por debajo de su emisor de tensión. Esto significa que cuando nuestra uC da salida a una lógica 1 o 5V el transistor se apagará debido a que de emisor está conectado a VCC. Puesto que la base no está por debajo del emisor el transistor se apaga.

Q1 de la figura V.44, se enciende cuando la uC envía un 0 lógico (0V) en su pin TX, porque su base es menor en tensión y su emisor. Cuando el transistor PNP está en ella detiene su colector, hacia el emisor que se ata a 5V. Esto invierte la salida de la uC y se alimenta de 5V a la entrada RS232 PC. Cuando la uC genera un 1 lógico (5V), el transistor se apaga y la resistencia de 10KΩ o R3 emite la señal hasta el voltaje negativo acumulado en la alimentación negativa

en C1. El voltaje negativo indicado en el PC RS232, satisface el requisito de voltaje negativo de la interfaz.

### 5.3.- PRUEBAS Y APLICACIONES DEL MODULO BLUETOOTH

Una vez realizado el enlace se procede a obtener la información de la ECU mediante la aplicación MegaTune, donde se toma en cuenta el modulo bluetooth y el puerto serial al que esta asociado, en este caso lo asociamos al puerto COM 5, para realizar la prueba.

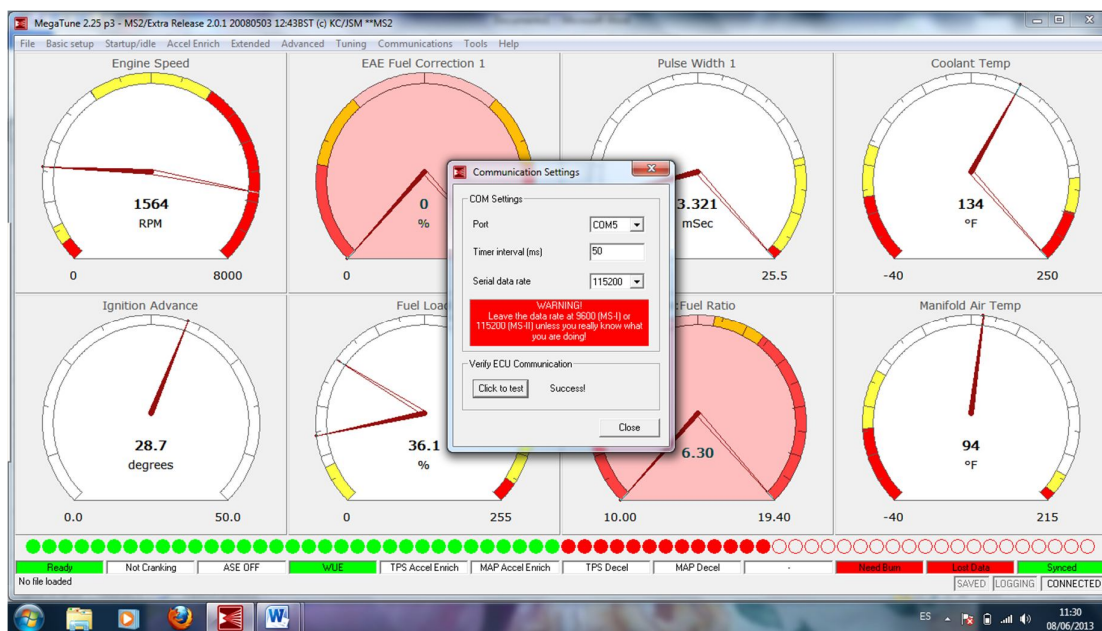


Figura V.44.- Comunicación con la Aplicación mediante conexión Bluetooth

Fuente: El Autor

Como podemos ver en la figura V.45 la comunicación con la ECU se lo realiza mediante la conexión con el puerto serial en este caso COM5, el tiempo de

intervalo es de 50ms por lo tanto la velocidad de la tasa de transferencia de transmisión y recepción del puerto serial es de 115200 bps.

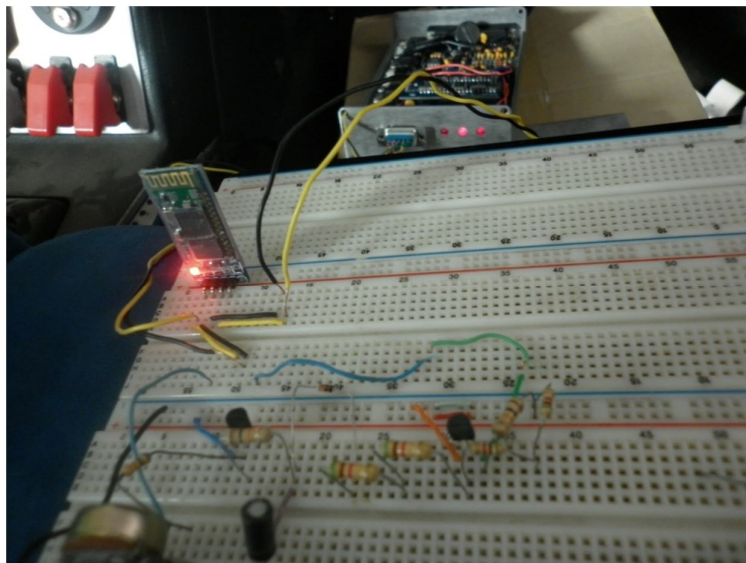


Figura V.45.- Prueba de conectividad con el sistema bluetooth

Fuente: El Autor

En la grafica V.45 permite visualizar la interactividad del modulo bluetooth trabajando con la ECU y adaptado a un circuito electrónico para eliminar los voltajes que alteran el funcionamiento del mismo, como se puede apreciar en la grafica V.45 la intermitencia de led (rojo) indica el trabajo de conectividad del sistema con la CPU del vehículo, por lo tanto permite la transmisión y recepción de datos.

### 5.3.1.- Pruebas en software libre

Esta prueba se la realizo en software libre con la ayuda de una maquina virtual, el sistema operativo empleado es Centos versión 5.6, por cumplir por todas las características necesarias para generar la aplicación llamada MegaTunix, la particularidad de este sistema operativo es factible obtener una actualización

del sistema operativo mediante la actualización en línea de los repositorios existentes.

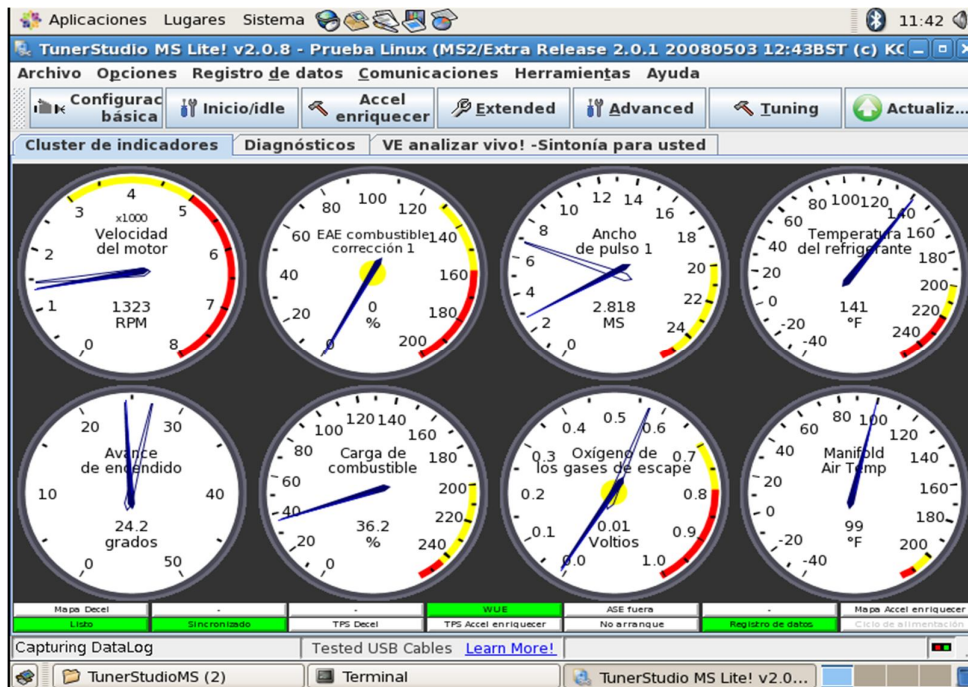


Figura V.46.- Prueba en Software Libre

Fuente: El Autor

La figura V.46, se demuestra la conectividad de la ECU, como podemos observar los distintos sensores del vehículo nos indican la información recibida mediante la visualización de los mismos.

Esta prueba demuestra que la aplicación de transferencia de datos se las puede obtener bajo un sistema operativo libre y por lo tanto las aplicaciones que corran en este sistema operativo están expuestas a revelar su código fuente o las carpetas contenedoras de algún repositorio del programa.

## 5.4.- COMPROBACION DE LA HIPÓTESIS

El método estadístico para comprobar las hipótesis fue chi-cuadrada ( $\chi^2$ ) por ser una prueba que permitió medir aspectos cualitativos y cuantitativos de las respuestas que se obtuvieron del instrumento instalado y medir la relación que existe entre las dos variables de las hipótesis en estudio.

El valor de chi-cuadrada se calculará a través de la fórmula siguiente:

$$\chi^2 = \frac{\sum(o_i - E_i)}{E_i}$$

$\chi^2$  = chi-cuadrada

$O_i$  = Frecuencia observada (respuestas obtenidas del instrumento)

$E_i$  = Frecuencia esperada (respuestas que se esperaban)

El criterio para la comprobación de las hipótesis se define así: si  $\chi^2_c$  (calculada) es menor que  $\chi^2_t$  (tabla) se acepta la hipótesis de trabajo y se rechaza la hipótesis nula; en caso contrario que  $\chi^2_t$  fuese mayor que  $\chi^2_c$  se acepta la hipótesis nula y se rechaza la de trabajo.



Datos de Frecuencia Observada

	Velocidad del Motor (RPM)	Ancho de Pulso (ms)	Temp. Refri. (°F)	Avance del encendido (grados)	Carga Combustible (%)	O2 gases de escape (Voltios)	Air Temp (°F)
Prueba 1	1646	3.977	122	29,9	41,6	20,3	76
Prueba 2	3249	3.005	125	-9	28	20	75
Prueba 3	6841	10.663	133	59,9	68	20,3	84
Total	11736	17.645	380	80,8	137,6	60,6	235

Datos de frecuencia Esperada

	Velocidad del Motor (RPM)	Ancho de Pulso (ms)	Temp. Refri. (°F)	Avance del encendido (grados)	Carga Combustible (%)	O2 gases de escape (Voltios)	Air Temp (°F)
Prueba 1	1700	4.000	120	30	45	24	78
Prueba 2	3300	3.030	125	10	28	24	77
Prueba 3	6900	11.000	135	60	68	24	88
Total	11900	18.030	380	100	141	72	243

FO	FE	FO-FE	(FO-FE) <sup>2</sup>	(FO-FE) <sup>2</sup> /FE
11736	11900	-164	26896	2,26017
17.645	18.030	-385	148225	8,22102
380	380	0	0	0
80,8	100	-19,2	368,64	3,6864
137,6	141	-3,4	11,56	0,08199
60,6	72	-11,4	129,96	1,805
235	243	-8	64	0,26337
			Xe=	16,3179

Grados de Libertad

$$V = (F-1)(C-1)$$

En donde: F= Filas; C= Columnas

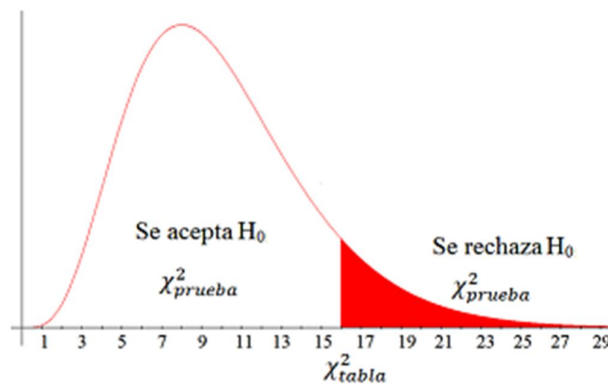
$$V = (7-1)(3-1)$$

$$V = 12$$

E= 5% (0.05)

X<sup>2</sup> C= MUESTRAL = 16,3179

X<sup>2</sup> T= Según tabla = 21.06



	0,995	0,990	0,975	0,950	0,900	0,750	0,500	0,250	0,100	<b>0,050</b>	0,025	0,010	0,005
1	0,000	0,000	0,001	0,004	0,016	0,102	0,455	1,323	2,706	3,841	5,024	6,635	7,879
2	0,010	0,020	0,051	0,103	0,211	0,575	1,386	2,773	4,605	5,991	7,378	9,210	10,597
3	0,072	0,115	0,216	0,352	0,584	1,213	2,366	4,108	6,251	7,815	9,348	11,345	12,838
4	0,207	0,297	0,484	0,711	1,064	1,923	3,357	5,385	7,779	9,488	11,143	13,277	14,860
5	0,412	0,554	0,831	1,145	1,610	2,675	4,351	6,626	9,236	11,070	12,833	15,086	16,750
6	0,676	0,872	1,237	1,635	2,204	3,455	5,348	7,841	10,645	12,592	14,449	16,812	18,548
7	0,989	1,239	1,690	2,167	2,833	4,255	6,346	9,037	12,017	14,067	16,013	18,475	20,278
8	1,344	1,646	2,180	2,733	3,490	5,071	7,344	10,219	13,362	15,507	17,535	20,090	21,955
9	1,735	2,088	2,700	3,325	4,168	5,899	8,343	11,389	14,684	16,919	19,023	21,666	23,589
10	2,156	2,558	3,247	3,940	4,865	6,737	9,342	12,549	15,987	18,307	20,483	23,209	25,188
11	2,603	3,053	3,816	4,575	5,578	7,584	10,341	13,701	17,275	19,675	21,920	24,725	26,757
<b>12</b>	<b>3,074</b>	<b>3,571</b>	<b>4,404</b>	<b>5,226</b>	<b>6,304</b>	<b>8,438</b>	<b>11,340</b>	<b>14,845</b>	<b>18,549</b>	<b>21,026</b>	<b>23,337</b>	<b>26,217</b>	<b>28,300</b>

Se rechaza la hipótesis  $H_0$  y se acepta la  $H_1$

Como  $X^2_C$  muestral (16,3179) es menor que el valor calculado en la tabla (21,026) se rechaza la hipótesis nula ( $H_0$ ) y se acepta la hipótesis de trabajo ( $H_1$ ) por lo tanto queda demostrado la hipótesis del presente trabajo de investigación.

## CONCLUSIONES

1. El desarrollo de la electrónica automotriz ha significado un paso importante en la fabricación de automóviles mucho más eficientes, autónomos, económicos, seguros y sobre todo el desarrollo de transmisión de datos almacenados dentro de un computador propio de los vehículos electrónicos
2. El sistema de telemetría es un factor importante en la comunicación con los sistemas de encendido electrónico donde se obtiene un control total del funcionamiento del automóvil.
3. Conforme con los resultados obtenidos la transferencia de datos con tecnología inalámbrica, desde una estación central (laptop) hacia el vehículo Mazda 323, permite mejor movilidad y portabilidad
4. Modificar el sistema convencional (cableado), de control hacia el vehículo Mazda 323 por un sistema con tecnología inalámbrica bluetooth.
5. Los microcontroladores son fundamentales en la implementación de sistemas de control dedicados, y más aún aquellos que pertenece a la familia de gama alta de microcontroladores como es el CPU 68HC08 de Microchip Motorola, quien posee todas las herramientas, módulos y componentes necesarios para diseñar aplicaciones automotrices y control de motores para mejorar el rendimiento de los vehículos y en especial aquellos de competencia.

6. El modo de funcionamiento de un motor respecto a las diferentes señales que emite un sensor, depende de la forma como estén calibrados cada uno de los componentes; sean electrónicos, eléctricos y mecánicos, hoy en día con la ayuda tecnológica se puede lograr con la ECU que calcula el tiempo de inyección de combustible, el oxígeno de gas de escape y otros parámetros.
7. La capacidad de procesamiento es importante en sistemas de control de motores de combustión interna y en todo tipo de aplicaciones donde las señales a ser tratadas alcanzan frecuencias altas de funcionamiento. En el caso del Mazda 323 la tasa de transferencia de datos es de 115200 bps. por lo que la implementación de este proyecto en cuanto a la velocidad de transmisión por parte del microcontrolador fue un factor muy importante.
8. Los sistemas operativos comunes como Windows manejan aplicaciones de fácil instalación mientras que para el sistema operativo Linux es más demorado debido a que los repositorios de la aplicación no están hechos como es el caso de Centos 5.6
9. La mecánica es un universo de conocimientos físicos, térmicos, dinámicos entre otros, aplicados a sistemas de conservación de energía. La electrónica por su parte, es otra área de aplicación de la física que mediante el estudio del comportamiento de efectos eléctricos y magnéticos hacen posible la implementación de dispositivos y equipos especializados en diferentes tareas. Unir estas dos áreas en la mecánica automotriz moderna, ha sido un cambio significativo para el desarrollo de

sistemas de transmisión de datos mucho más sofisticados y aplicados a controlar de manera electrónica los fenómenos físicos que suceden en los vehículos.

## RECOMENDACIONES

1. La demanda de velocidad de procesamiento por parte del microcontrolador, hacia una estación central, depende de la cantidad de información que se le envíe, por lo que se recomienda realizar el estudio, diseño e implementación de este tipo de proyecto en otras tecnologías inalámbricas para mejorar su cobertura.
2. El asesoramiento de técnico en el área de mecánica automotriz es importante. El comprender el funcionamiento de un motor y los sistemas que lo componen es mucho más sencillo si se cuenta con el personal adecuado que le guíe en el tema.
3. Para implementar este tipo de aplicaciones, se recomienda empezar desde lo básico como la detección de las conversiones Analógicas a Digitales hasta llegar a lo más complicado como lo es la transmisión de datos del microcontrolador hacia el modulo bluetooth.
4. Como es lógico, se necesita siempre estar realizando pruebas sobre el hardware que se desarrolla así como de la programación que se pueda utilizar con el objetivo de realizar cambios y ajustes oportunos durante la etapa de diseño, implementación y depuración del proyecto.
5. Para futuros estudios e implementaciones de proyectos en esta área, es recomendable aconsejar la utilización de otro tipo de dispositivos inalámbricos para abarcar mas cobertura en la transmisión y recepción de datos

6. En una opinión muy personal, de lo estudiado, aprendido e implementado, el sistema de transmisión inalámbrica es el más importante, primeramente porque permite la movilidad de la estación central y previene de sobrecalentamientos del motor y genera menor contaminación ambiental, por lo que se recomienda prestar especial atención a este sistema.
7. Recomendamos verificar los voltajes de los dispositivos electrónicos instalados en el vehículo (12VCD) para todo el sistema. Para la conexión del circuito electrónico con el módulo bluetooth la alimentación de voltaje debe estar entre 3.3VCD a 5VCD.
8. Se recomienda para próximas implementaciones de este tipo de sistemas, es que se debe contar con el apoyo de personal capacitado en el área de la mecánica automotriz, equipos y herramientas propias de una implementación para este tipo de tecnología.



## **RESUMEN**

La implementación de un sistema de transferencia de datos por telemetría para un vehículo de competencia con el uso de tecnología inalámbrica bluetooth, utiliza una aplicación interactiva llamada MegaTune2.25P3 mas un modulo bluetooth que va instalado junto al tablero de mandos que se conecta al CPU 68HC08 del vehículo de competencia Mazda 323, permite visualizar los parámetros técnicos, electrónicos, mecánicos, en tiempo real del rendimiento motor en una estación central de trabajo (laptop).

En el presente trabajo de investigación se aplicó el método experimental para medir cada uno de los parámetros técnicos, electrónicos, mecánicos, del rendimiento del motor en diferentes fases de prueba como arranque y condiciones extremas. De los resultados observados se puede ir calibrando las variaciones que existen en los parámetros antes mencionados.

Como resultado se demuestra que la tecnología empleada permite la conexión instantánea desde la estación central hacia el modulo bluetooth instalado en el vehículo con una tasa de transferencia de datos del 97% de eficiencia.

Conforme con los resultados obtenidos se concluye que la transferencia de datos con tecnología inalámbrica, desde una estación central (laptop) hacia el vehículo Mazda 323, permite mejor movilidad y portabilidad que un sistema convencional cableado.

Recomendamos verificar los voltajes de los dispositivos electrónicos instalados en el vehículo (12 Voltios de corriente directa) para todo el sistema. Para la

conexión del circuito electrónico con el modulo bluetooth la alimenta de voltaje debe estar entre 3.3 a 5 (Voltios de corriente directa).

## **SUMMARY**

The implementation of a data transferring system for telemetry for a competing vehicle by using bluetooth wireless technology, an interactive application called MegaTune2.25P3 is used plus a bluetooth module that is installed next to the board that is connected to the CPU 68HC08 of the vehicle of competition Mazda 323, it allows to display technical, electronic, mechanical, parameters in real time of the engine performance in a central workstation (laptop).

In the present research, the experimental method was applied in order to measure each one of the technical, electronic and mechanical parameters of the engine performance in the different test phases such as starting and external conditions. From the observed results, it can be calibrating the variations in parameters mentioned above.

The results show that the employed technology allows the instant connection from the central station up to the bluetooth module installed in the vehicle with a data transfer rate of 97% of efficiency.

According to the obtained results, it is concluded that the data efficiency with wireless technology, from a central station (laptop) up to the vehicle Mazda 323, it allows better portability and mobility than a conventional systems.

It is recommended checking the voltages of the electronic devices installed in the vehicle (12 Volts direct current) for the entire system. To connect the electronic circuit with the bluetooth module, the voltage feeding should be between 3.3 to 5 (Volts direct current).

## BIBLIOGRAFIA

### LIBROS

- 1.- **MÍNGUEZ, D.**, Monitoreo de parámetros a través de un sistema de telemetría., Instituto Politécnico Nacional., Centro de Investigación y Desarrollo de la Tecnología Digital., Tijuana-México., (**TESINA**) 2009., Pp. 23-50.  
<http://itzamna.bnct.ipn.mx:8080/dspace/bitstream/123456789/5720/1/MONITOREOPARAMET.pdf>  
2012-12-08
- 2.- **HERNANDEZ, J.**, Diseño e Implementación de una unidad electrónica para controlar el desempeño de un motor de combustión interna., Escuela Superior Politécnica de Chimborazo., Facultad de Informática y Electrónica., Escuela de Ingeniería Electrónica Telecomunicaciones y Redes., Riobamba-Ecuador., (**TESIS**) 2010., Pp. 64-96.  
2013-01-15
- 3.- **ÁLVAREZ, D. QUINTEROS, D.**, preparación y modificación de inyección electrónica programable con telemetría unidireccional de un motor para rally en un automóvil Mazda 323., Escuela Superior Politécnica de Chimborazo., Facultad de Mecánica., Escuela de Ingeniería Automotriz., Riobamba-Ecuador., (**TESIS**)., 2011., Pp. 318-353.  
2013-01-20
- 4.- **ULLOA, F. ACOSTA, A.**, Diseño de un sistema telemando y telemétrico experimental para una Aero estación de baja altura., Universidad Tecnológica Metropolitana., Facultad de Ingeniería., Departamento de Electricidad., Santiago de Chile., (**Documento**) 2011., Pp. 1-5.  
<http://www.haps.cl/documentos/01.pdf>

2013-02-08

- 5.- **EDGARDO, A. FRANCO, M.**, Clase 02 y 03: Introducción a la instrumentación electrónica 1a. ed., Ciudad México, Distrito Federal, México., 2007., Pp. 22-30
- [http://computacion.cs.cinvestav.mx/~efranco/docencia/instrumentacion/files/02\\_03\\_Introduccion\\_a\\_la\\_instrumentacion.pdf](http://computacion.cs.cinvestav.mx/~efranco/docencia/instrumentacion/files/02_03_Introduccion_a_la_instrumentacion.pdf)

2013-04-15

## **BIBLIOGRAFIA DE INTERNET**

### **6.- TELEMETRIA**

<http://dspace.ups.edu.ec/bitstream/123456789/41/8/Capitulo2.pdf>

<https://www.geocoding.es/index.php/servicios/servicio-de-telemetria>

2012-11-20

### **7.- MODÚLO BLUETOOTH HC-06 O GP-GC021 Y ARDUINO**

<http://tallerarduino.wordpress.com/2011/12/06/modulo-bluetooth-hc-06-o-gp-gc021-y-arduino/>

2012-12-04

### **8.- APLICACIONES DEL MODULO BLUETOOTH**

<http://www.puntofotante.net/BOLT-CONTROL-REMOTO-BLUETOOTH.htm>

<http://www.puntofotante.net/INSTALACION-DE-MODULO-BLUETOOTH-EN-PC.htm>

2012-12-07

### **9.- GUÍA DIDÁCTICA MOTOROLA**

<http://arantxa.ii.uam.es/~gdrivera/robotica/hc08/mc68hc908gp32.pdf>

2000-06-14

**10.- GUÍA DIDÁCTICA MODULO BLUETOOTH.**

<http://www.ii.uam.es/~gdrivera/robotica/hc08/hc908gp32.htm>

2012-11-17

**11.- CONECTIVIDAD ATRAVEZ DE BLUETHOO A INTERNET**

<http://www.youtube.com/watch?v=yMZI2V28gro>

2012-11-13

**11.- SOFTWARE LIBRE**

<http://www.fsfla.org/ikiwiki/about/what-is-free-software.es.html>

2012-12-10

**12.- ESTÁNDAR RS-232C**

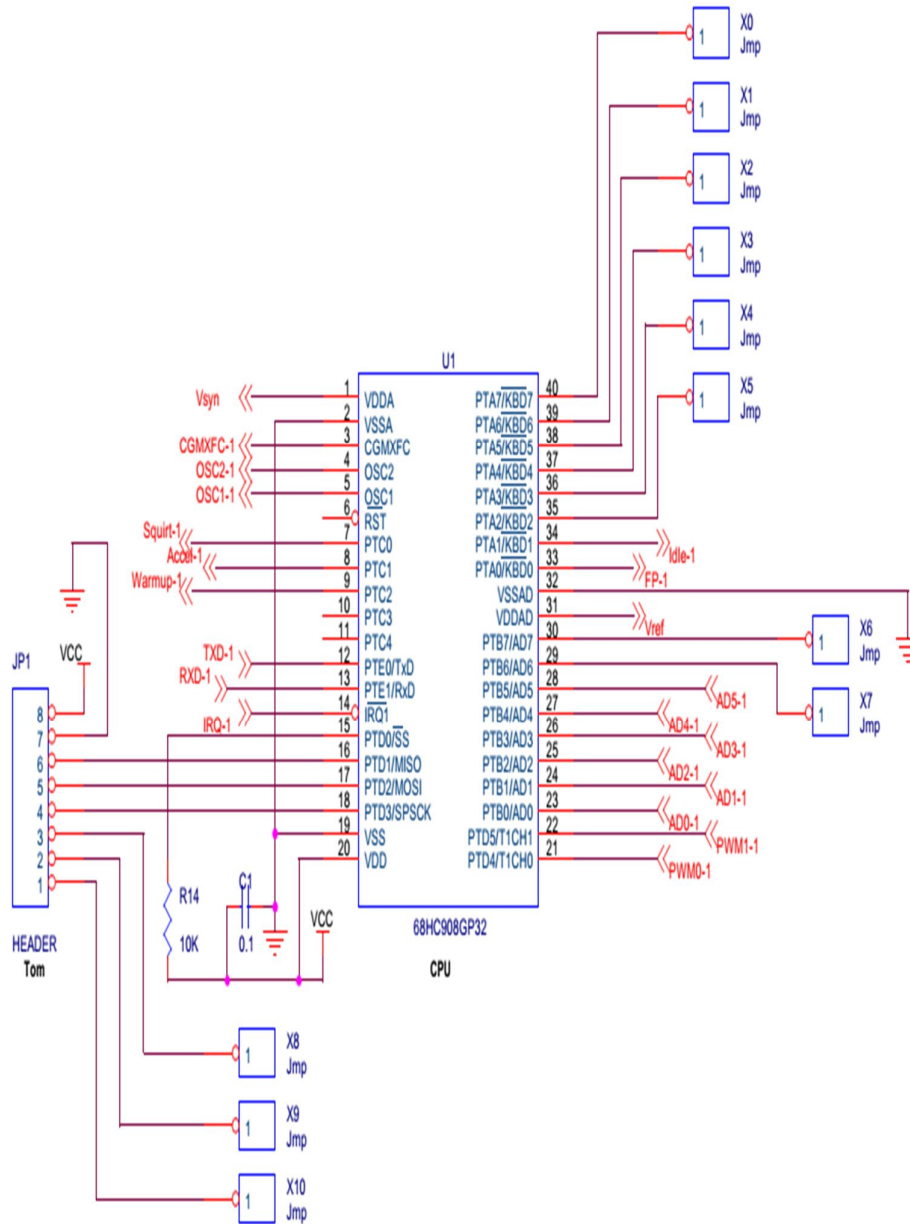
<http://www.euskalnet.net/shizuka/rs232.htm>

2013-03-14

# **ANEXOS**

# ANEXO 1

## Diagrama de microprocesador CPU 68HC08





## **ANEXO 2**

### **Scrip y Repositorio de Megatunix en CD**

**ANEXO 3**  
**BLUETOOTH 9600BPS MODULO**

---

## Chapter1. Bluetooth Serial Converter UART Interface

### 9600bps

---

#### Key Features

- Bluetooth Spec v2.0+EDR Compliant
- Enhanced Data Rate (EDR) compliant with V2.0.E.2 of specification for both 2Mbps and 3Mbps modulation modes
- Class 2 Type Output Power
- Full Speed Bluetooth Operation with Full Piconet Support
- Scatternet Support
- 3.3V Operation
- Minimum External Components
- UART Interface
- Support for 8Mbit External Flash Onboard
- Support for 802.11Co-Existence
- RoHS Compliant



#### Product Description

GL-6B is a Class 2 Bluetooth module using BlueCore4+AudioROM chipset from leading Bluetooth chipset supplier Cambridge Silicon Radio.

This converter is used to convert a UART with 9600bps, one start bit, 8 data bits, one stop bit, no parity bit format to Bluetooth UART protocol. It can not talk to a same device but only to PC Bluetooth dongle. That is, it will create communication between an embedded microcontroller and a PC, but not between 2 embedded devices.

For the password, normally it doesn't need a password, when PC could find the converter, just double click it and the connection will be created. If it requests you to input password all the same, it shall be 0000 or 1234.

#### Applications

- Bluetooth CarKit
- PCs
- Computer Accessories

## Bluetooth Serial Converter UART Interface 9600bps

— Access Points

### Specifications

Operating Frequency Band	2.4GHz~2.48GHz unlicensed ISM band
Bluetooth Specification	V2.0+EDR
Output Power Class	Class2
Operating Voltage	3.3V
Host Interface	UART
Dimension	28.9mm(L)×13mm(W)×2.2mm(H)

\* Specification are subject to change without prior notice

### Electronics Characteristics

Absolute Maximum Ratings		
Rating	Min	Max
Storage Temperature	-40°C	+150°C
Supply Voltage	+0.4V	5.6V
Other Terminal Voltage	VSS-0.4V	VDD+0.4V

Recommended Operating Conditions		
Operating Conditions	Min	Max
Operating Temperature Range	-40°C	+150°C
Guaranteed RF Performance Range*	-40°C	+150°C
Supply Voltage	2.2V	4.2V

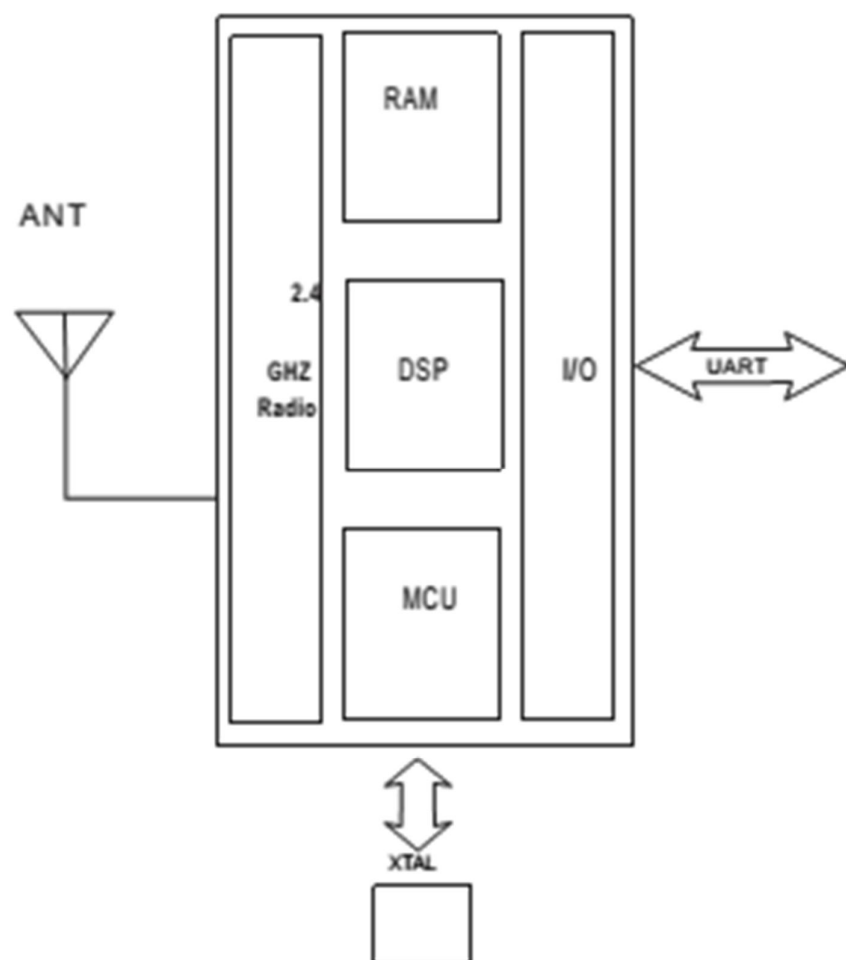
\* Typical figures are given for RF performance between -40°C and +105°C.

### Power Consumption

Operation Mode	Connection Type	UART Rate (Kbps)	Average	Unit
Page Scan	•	115.2	0.42	mA
ACL No Traffic	Master	115.2	4.60	mA
ACL with File Transfer	Master	115.2	10.3	mA
ACL 1.28s Sniff	Master	38.4	0.37	mA
ACL 1.28s Sniff	Slave	38.4	0.42	mA
Standby Host Connection	•	38.4	40	µA

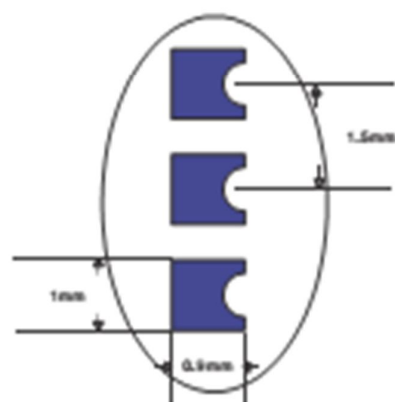
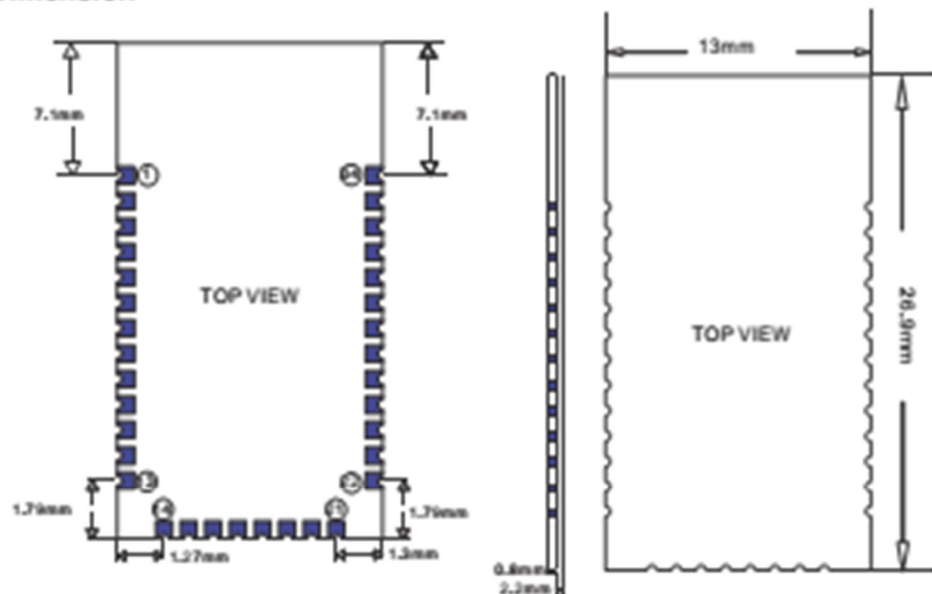
\* Low power mode on the linear regulator is entered and exited automatically when the chip enters/leaves Deep Sleep mode. For more information about the electrical characteristics of the linear regulator, see section 4 in this document.

Block Diagram



## Bluetooth Serial Converter UART Interface 9600bps

### Dimension



### Pin Configuration

PIN NO.	NAME	TYPE	FUNCTION	RE-MARK
1	UART-TX	CMOS Output	UART Data Output	
2	UART-RX	CMOS Input	UART Data Input	

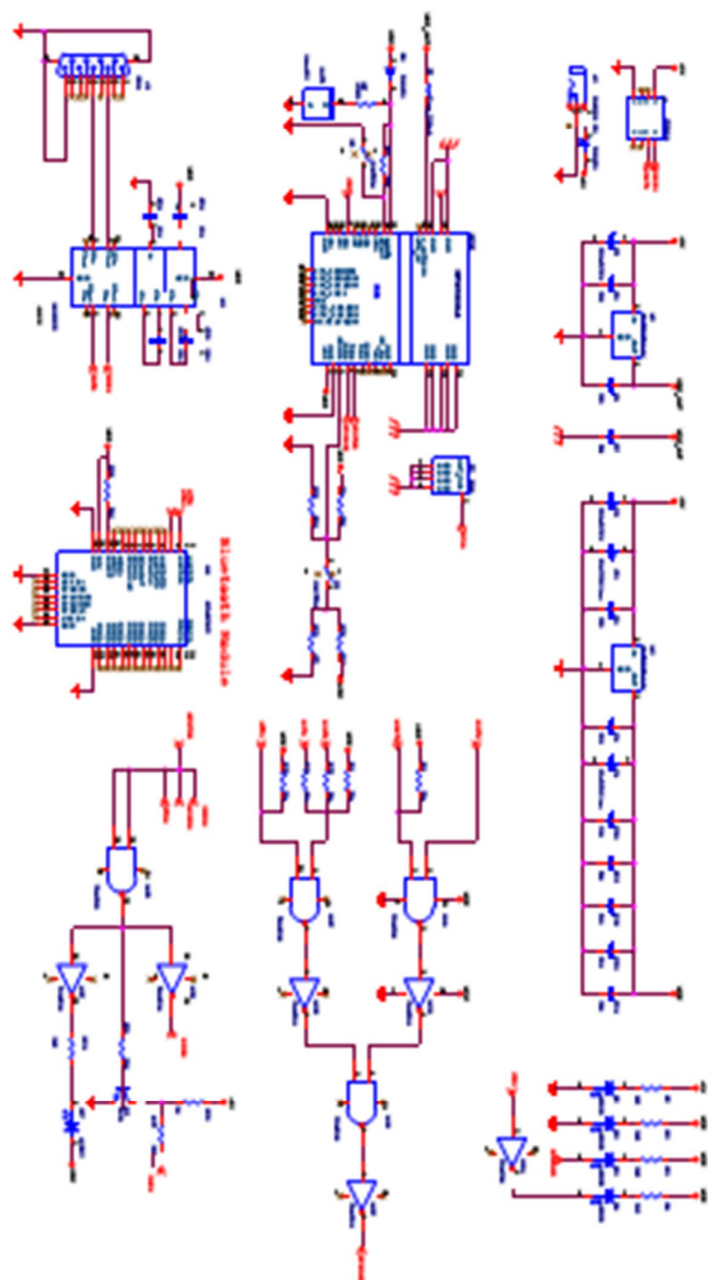
### Bluetooth Serial Converter UART Interface 9600bps

3	UART-CTS	CMOS Input	UART Clear to Send Active Low	
4	UART-RTS	CMOS Output	UART Request to Send Active Low	
5	NC	*	*	
6	NC	*	*	
7	NC	*	*	
8	NC	*	*	
9	NC	*	*	
10	NC	*	*	
11	RESETB	CMOS Input	Reset if low. Input debounced so must be for >5ms to cause a reset.	
12	3.3V	POWER	+3.3V Supply	For 3.3V Version
13	GND	GND	Ground	
14	GND	GND	Ground	
15	NC	*	*	
16	NC	*	*	
17	NC	*	*	
18	NC	*	*	
19	NC	*	*	
20	NC	*	*	
21	GND	GND	Ground	
22	GND	GND	Ground	
23	NC	*	*	
24	NC	*	*	
25	NC	*	*	
26	NC	*	*	
27	NC	*	*	
28	NC	*	*	
29	NC	*	*	
30	NC	*	*	
31	NC	*	*	
32	NC	*	*	
33	NC	*	*	
34	NC	*	*	

## Bluetooth Serial Converter UART Interface 9600bps

---

### Application Schematics





## **Bluetooth Serial Converter UART Interface 9600bps**

---

More information about the application schematics and hardware can be got on our website with product number GP•GS010.



## Chapter2. Contact Us

---

Sure Electronics Co., Ltd.

5F, Zone A,

Qinhuai Technology Innovation Center

105-2 DaMing Rd(Zip Code : 210022)

Nanjing

PRC

Tel: +86-25-66606340 (English Service) GMT1am-10am

Fax: +86-25-66606346

Website: [www.sure-electronics.net](http://www.sure-electronics.net)

[www.sureelectronics.net](http://www.sureelectronics.net)

[www.sure-electronics.com](http://www.sure-electronics.com)