ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

“DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA QUE PERMITA MEDIR Y ALMACENAR PARÁMETROS DE VELOCIDAD, TIEMPO Y DISTANCIA RECORRIDA DE UN AUTOMOTOR EN UNA MEMORIA FLASH O EN UNA MEMORIA SD.”

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

JUAN FRANCISCO CABRERA ALVEAR
(jottafcabrera@hotmail.com)

DIRECTOR: Dr. ING. LUIS CORRALES
(luis_corrales@yahoo.com.es)

Quito, Enero 2009

DECLARACIÓN

Yo, Juan Francisco Cabrera Alvear, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo los derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normatividad institucional vigente.

_____

Juan Francisco Cabrera Alvear

CERTIFICACIÓN

Certifico que el presente trabajo, “DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA QUE PERMITA MEDIR Y ALMACENAR PARÁMETROS DE VELOCIDAD, TIEMPO Y DISTANCIA RECORRIDA DE UN AUTOMOTOR EN UNA MEMORIA FLASH O EN UNA MEMORIA SD”, fue desarrollado por el señor: Juan Francisco Cabrera Alvear, bajo mi supervisión.

_____

Dr. Luis Corrales

DIRECTOR DEL PROYECTO

AGRADECIMIENTOS

Al Dr. Luis Corrales por su valioso tiempo y soporte en el avance de este proyecto.

A todos mis compañeros que siempre me brindaron una palabra de apoyo para que siga adelante día a día.

A todos mis amigos que me exigían día tras día más fuerza, más concentración, más dedicación para culminar este proyecto.

A Dios por darme fuerza y la sabiduría suficiente para enfrentar cada meta que se me ponga adelante.

DEDICATORIA

Dedico el presente trabajo a mis amados Padres Juan y Patty por todo su amor, ejemplo y sacrificio. A mis amados hermanos Sebas y Andrés, de quienes siempre puedo confiar y día a día se van convirtiendo en un ejemplo para mí.

Dedico a mi linda abuelita mamá Elvira que siempre me supo cuidar con amor y paciencia. A muchos seres queridos que ya no están con migo, como papá Juan, papá Fausto, mamá Eufemia y a toda mi familia que siempre me brinda todo su cariño y comprensión.

Dedico también este trabajo a May de quien siempre tuve una palabra de aliento.

JottaFCA – Juan Francisco Cabrera Alvear

CONTENIDO

CAPÍTULO 5:
CONCLUSIONES Y RECOMENDACIONES

ÍNDICE DE FIGURAS

CAPÍTULO 1

CAPÍTULO 2

CAPÍTULO 3

CAPÍTULO 4

ÍNDICE DE TABLAS

CAPÍTULO 1

CAPÍTULO 2

CAPÍTULO 4

ANEXOS

RESUMEN

En el presente trabajo se diseña y construye un módulo electrónico experimental para la medición de los parámetros de velocidad, tiempo y distancia recorrida de un automotor y el almacenamiento de estos parámetros en una memoria flash o en una tarjeta SD.

El módulo electrónico está construido a base de un microcontrolador AVR ATmega16, el cual realiza la medición de la velocidad del automotor que se desea analizar, la medida de la velocidad se obtiene a través de la señal generada por el velocímetro instalado en el automotor. Además este chip establece una interfaz de comunicación con el integrado USBwiz-OEM que le posibilita acceder a dispositivos de almacenamiento.

El software programado en el microcontrolador se encarga de procesar la información recibida del automóvil (velocidad) para obtener por medio de cálculos la distancia recorrida y el tiempo de cada trayecto, estos parámetros se visualizan en un LCD que actualiza su información de forma continua. Posteriormente la información procesada se envía al integrado USBwiz-OEM, desde el cual se establece comunicación entre los dispositivos de almacenamiento y el módulo electrónico con la finalidad de almacenar la información en una memoria flash USB o en una tarjeta SD.

La actividad del automóvil es almacenada en un archivo de extensión CVS (Valores separados por comas) este tipo de archivo puede ser abierto en varios programas computacionales como Microsoft Word, WordPad, Microsoft Excel, Open Office entre los principales. La creación del archivo y el almacenamiento de la actividad del automóvil se realizan de forma automática.

Luego de haber concluido este proyecto se obtuvo resultados con un error del 4.19% en el valor experimental de la distancia recorrida por el automóvil con respecto al esperado; en cambio el valor obtenido experimentalmente de la velocidad tiene errores menores al 1%, dichos resultados se discuten en el respectivo capítulo.

PRESENTACIÓN

Las empresas dedicadas al transporte se ven gravemente afectadas por la falta de control en sus unidades, llevando a un deteriorando en su servicio donde el principal perjudicando es el usuario, esta particularidad podría ser evitada llevando un mejor registro de las actividades del automotor. De esta forma las empresas que brindan servicio de transporte podrían controlar de mejor forma a sus empleados optimizando el gasto en repuestos, en combustibles, y tiempo de uso de cada unidad de transporte. El llevar un registro detallado y amplio de las actividades del automotor llevaría a la utilización de unidades de almacenamiento de gran capacidad como son las memorias flash o tarjetas SD.

Hoy en día los dispositivos de almacenamiento como las memorias flash USB y las tarjetas SD, son muy utilizadas en la implementación de sistemas de control y adquisición de datos. La popularización de estos dispositivos se debe a que ahora prácticamente todo computador tanto personal como portátil poseen el conector necesario para estos dispositivos. Sin embargo, el hecho de necesitar un computador para acceder a la información de estos dispositivos almacenamiento seria una limitante para utilizarlos en cualquier sitio, resultaría de gran utilidad disponer de un medio que permita utilizar cualquier memoria flash o tarjeta SD sin la necesidad de la utilización de un computador.

Con estos objetivos en mente, este proyecto se desarrolla para satisfacer dichas necesidades, pues se comienza con la implementación de un sistema que permita adquirir parámetros de velocidad, tiempo y distancia recorrida de un automotor y se complementa con el diseño de un módulo electrónico que posibilite almacenar dichos parámetros en una memoria flash o tarjeta SD sin la necesidad de ocupar un computador.

Para cumplir con dicho objetivo en el Capítulo 1 se hace un análisis completo sobre los sensores de velocidad que son comúnmente instalados en los automotores, un estudio de la utilización del USBwiz-OEM para utilizarlo como interfaz entre el módulo electrónico y los distintos medios de almacenamiento además del análisis de las características de las memorias flash USB y tarjetas SD.

En el Capítulo 2 trata el diseño del hardware y software a implementar para realizar las mediciones de velocidad, tiempo y distancia recorrida por el automotor.

El Capítulo 3 se presenta la instalación realizada del módulo electrónico en el automotor escogido (camioneta Chevrolet LUV 2005) para realizar las pruebas.

En el Capítulo 4 se presentan las pruebas realizadas sobre el módulo electrónico instalado en su totalidad en el automotor y se discute los resultados obtenidos.

El Capítulo 5 presenta las conclusiones y recomendaciones sobre el proyecto.

CAPÍTULO 1

DESCRIPCIÓN DE LOS COMPONENTES PRINCIPALES

CAPITULO 1

DESCRIPCIÓN DE LOS COMPONENTES PRINCIPALES

5.3 OBJETIVO GENERAL

En el presente trabajo se pretende diseñar y construir un módulo el cual permita medir y almacenar parámetros de velocidad, tiempo y distancia recorrida de un automotor. Como medio de almacenamiento se utilizara memorias flash comerciales o tarjetas SD (Secure Digital).

El sistema propuesto se realiza con el fin de controlar la actividad de los automotores de manera que se lleve un registro de actividades en una memoria no volátil. El registro en la memoria tiene como objetivo forzar al conductor a no cometer infracciones tales como exceso de velocidad o conducir en horas no permitidas por la ley o el empleador.

Lo novedoso de este proyecto es que la adquisición de datos se realizará directamente a la memoria flash o a la memoria SD sin necesidad de utilizar un computador u otro tipo de adaptador. El archivo que se guardará en la memoria tendrá las diferentes actividades del automotor.

5.4 MEDICIÓN DE LA VELOCIDAD EN UN AUTOMOTOR

En la industria automotriz los desarrolladores de automóviles implementan en sus diseños varios dispositivos visuales, con el propósito de alertar al conductor sobre el funcionamiento del automotor o el estado en que se encuentra, uno de los principales medios visuales que se implemento en los automotores hace varias décadas es el velocímetro, este artefacto posibilita medir la rapidez instantánea en el automotor.

5.4.1 EL VELOCIMETRO [1]

El velocímetro es un instrumento que se ha popularizado en la mayoría de automotores, este instrumento mide la rapidez instantánea del automotor, los velocímetros tradicionales están controlados por un cable recubierto que es torsionado por un conjunto de pequeñas ruedas dentadas en el sistema de transmisión[1]; sin embargo, en los últimos años se han desarrollado sensores de velocidad digitales que se ubican internamente en la transmisión del automotor.

En la actualidad para medir la velocidad del automotor se utilizan varios métodos, entre los que más sobresalen son: medir la velocidad de rotación de los neumáticos, o medir la velocidad de rotación de los piñones en la transmisión, sea cual sea el método utilizado el velocímetro en un automotor tiene la capacidad de interpretar la señal generada por el sensor y traducirla a un equivalente de kilómetros por hora (Km/h) o millas por hora (mph). En la Figura 1.1 se muestra algunos modelos de velocímetros utilizados en los automotores.

Figura 1.1 "Velocímetros utilizados en el mercado automotriz"

[1] http://es.wikipedia.org/wiki/Veloc%C3%ADmetro

5.4.1.1 Tipos de Velocímetros

Al igual que existen varios modelos de automotores en el mercado, también existe una amplia variedad de velocímetros, esto se debe a sus diferente métodos de medición de velocidad, sus diversos sensores usados, y su diferente colocación en el automotor. Para reducir los diferentes grupos que se pudiesen obtener de los velocímetros se va a clasificarlos en dos grupos; los velocímetros de acción mecánica y los velocímetros de acción digital.

5.4.1.1.1 Velocímetro de acción Mecánica

Los automotores que tienen velocímetros de acción mecánica para medir su velocidad utilizan un cable flexible (cable de velocidad) unido al árbol de la transmisión que hace girar un imán permanente en su orilla, esto induce un campo magnético que tiende a arrastrar un tambor que rodea al imán. Cuanto mayor es la velocidad del vehículo, más fuerza se ejerce sobre el tambor y más alta es la desviación de la aguja del velocímetro, la aguja determinará la velocidad que posee el automotor.

Para medir la velocidad con dispositivos que no están instalados en el automotor (automotor con velocímetro de acción mecánica), se debe instalar un sensor en el cable que hace girar el tambor del velocímetro. El sensor produce una señal de frecuencia variable que es proporcional a la velocidad del automotor. La señal producida por el sensor es acoplada por medio de transductores que posibilitan la manipulación de la señal con el dispositivo externo que realizará las medidas de la velocidad del automotor.

El cable del velocímetro se conecta a la transmisión del automotor por medio de un rotor incorporado en la transmisión, el rotor hace girar el filamento interno del cable del velocímetro cuando el automotor se encuentra en movimiento. En la Figura 1.2 se muestra la conexión de la transmisión y el cable del velocímetro.

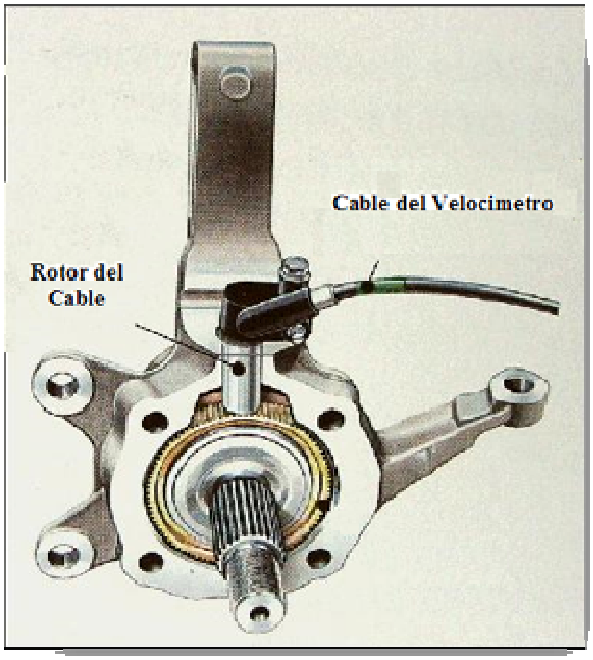


Figura 1.2 “Velocímetro de Acción Mecánica”

En la Figura 1.3 se muestra dos tipos de cables de velocidad, estos cables poseen un filamento interno que gira proporcionalmente a la velocidad de los piñones de la transmisión del automotor donde son utilizados.

Figura 1.3 “Cable del Velocímetro” [2]

5.4.1.1.2 Velocímetros de acción digital [3]

En los automotores modernos el velocímetro ya no está conectado mecánicamente a la transmisión. Un dispositivo (sensor de velocidad) situado en la transmisión produce una serie de pulsos electrónicos cuya frecuencia varía de acuerdo con la velocidad del vehículo; estos pulsos electrónicos se transmiten a un dispositivo calibrado que determina la velocidad y envía esta información al velocímetro. La velocidad del automotor puede visualizarse a partir de la desviación de la aguja en el velocímetro o directamente en una pantalla digital.

Para medir la velocidad en automotores que no utilizan sistemas mecánicos instalados en su transmisión (automotor con velocímetro de acción digital), se debe realizar una derivación en el conductor que transmite la señal generada del sensor de velocidad hacia el velocímetro. La derivación realizada proveerá una señal de pulsos cuya frecuencia es proporcional a la velocidad del automotor.

En la Figura 1.4 se muestra la conexión del sensor de velocidad con el velocímetro del automotor, cabe notar que el sensor está instalado internamente en el sistema de transmisión y que el cable que comunica el dispositivo de calibración con el velocímetro es un conductor de cobre común y corriente.

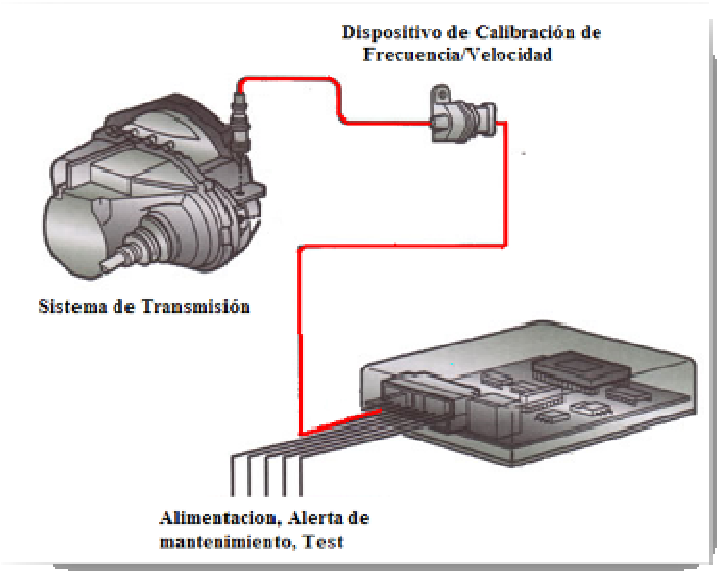


Figura 1.4 “Velocímetro de acción Digital”

En la Figura 1.5 se distingue las partes de un sensor de velocidad instalado internamente en la transmisión, el sensor genera pulsos a ritmo que la rueda dentada gira y estos pulsos son enviados a un dispositivo de calibración para después enviar la información al velocímetro instalado en el automotor.

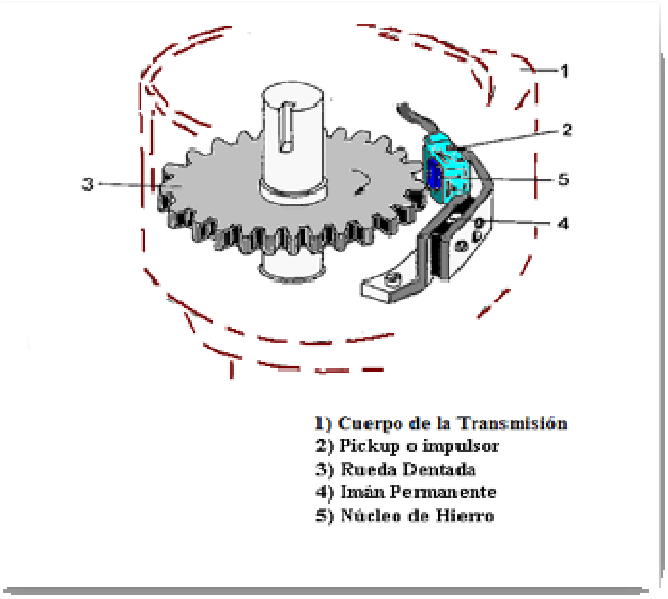


Figura 1.5 "Esquema de Sensor interno de la Transmisión"

5.4.1.2 Análisis de los Sensores de Velocidad

Los sensores son dispositivos que, a partir de la energía del medio donde mide, suministran una señal de salida que es función de la variable medida. Dicho en otras palabras el sensor es un dispositivo capaz de transformar magnitudes físicas, en magnitudes eléctricas (voltaje, corriente).

Si bien el automotor posee muchos sensores con los cuales se puede prevenir y supervisar su funcionamiento, se pondrá énfasis en los sensores de velocidad buscando el más apropiado para incluirlo en el proyecto que se está desarrollando o analizar el funcionamiento del sensor de velocidad que este instalado en el automotor. Los sensores de velocidad que se presentaran a continuación son los más comerciales en el mercado ecuatoriano y los más utilizados en los automotores existentes.

5.4.1.2.1 Sensores de Velocidad "PICK UP COIL"

Los sensores pick up coil, son generadores de voltaje que se utilizan para medir la velocidad de rotación del componente donde se los instala (por lo general su ubicación es en la transmisión del automotor o en la cadena del automotor[2]). Este sensor consiste en un imán permanente, una pieza ferromagnética, un pick up coil[3], y un rotor de ruedas dentadas.

A medida que la rueda dentada gira, la pieza ferromagnética cambia el flujo de corriente que circula a través de la bobina, esto se traduce en un cambio de voltaje (voltaje AC) que se mide por un circuito externo. La frecuencia de la señal generada en el sensor es proporcional a la velocidad angular del rotor y es utilizada para medir la velocidad del automotor. En la Figura 1.6 se muestra el esquema básico de un sensor de velocidad pick up coil.

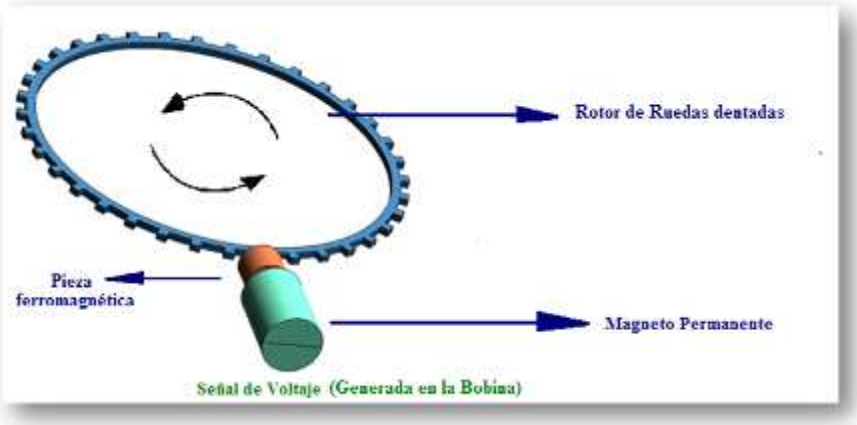


Figura 1.6 "Esquema de un sensor PICK UP COIL" [4]

Este tipo de sensor generalmente es utilizado en la cadena del automotor pero en ciertos modelos de automóviles se incorpora sensores pick up coil en su transmisión para medir la velocidad. En la Figura 1.7 se presenta un sensor Pick Up Coil comercial utilizado en la cadena del automotor, este tipo de sensor en su armazón es muy resistente y presenta una extensión por la cual se puede medir la señal que genera el sensor, para utilizar este tipo de sensor hay que adaptar la

[2] **Cadena del Automotor**: Denominación que se suele utilizar al cable del velocímetro en el automotor.

[3] **Pick up coil**: Bobinado utilizado para generar una señal de Voltaje AC

señal que salida ya que la amplitud de la señal no es constante y podría provocar errores en sus lecturas o daños por exceso de voltaje en los equipos al que se lo conectara.

Figura 1.7 "Sensor Pick Up Coil Comercial"

En la Figura 1.8 se muestra un sensor pick up coil montado en la transmisión del automotor. Este tipo de sensores actualmente no se utilizan, debido a que las lecturas producidas cuando el automotor va a baja velocidad son erróneas. [5]

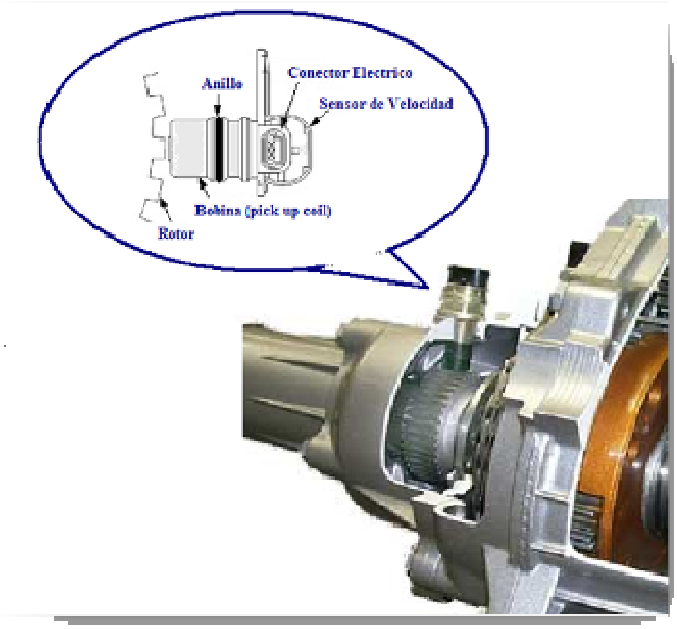


Figura 1.8 "Sensor de Velocidad pick up coil instalado en el sistema de transmisión del automotor"

Un problema que se presenta en este tipo de sensor es la ruptura en los dientes del rotor provocando alteraciones en la frecuencia de la salida, esto se debe por la gran velocidad a la que gira. Al transcurrir el tiempo y su uso este sensor presenta anomalías en sus lecturas, el error es producido por el efecto de histéresis[4] en los dientes Rotor.

5.4.1.2.2 Sensores de Velocidad "Fotoeléctricos"

Este tipo de sensores de velocidad son capaces de emitir una señal de voltaje cuya frecuencia es proporcional a la velocidad del automotor, se usa normalmente como contadores de revoluciones y debe estar montado directamente en el cable del velocímetro.

El funcionamiento de este tipo de sensores se basa en la interrupción óptica de un fototransistor, el fototransistor utilizado por lo general es el integrado H21A3 el mismo que consta de un diodo emisor de luz infrarroja y un fototransistor de silicio recubiertos en un encapsulamiento plástico. Su encapsulamiento proporciona un espacio que se utiliza para interrumpir la señal del LED emisor y el fototransistor por lo general se utiliza un material opaco (color negro), al interponerse el material opaco en el haz infrarrojo genera un señal digital (On –Off) de frecuencia variable según las veces que se corte el haz infrarrojo. En la Figura 1.9 se presenta el conmutador óptico H21A3.

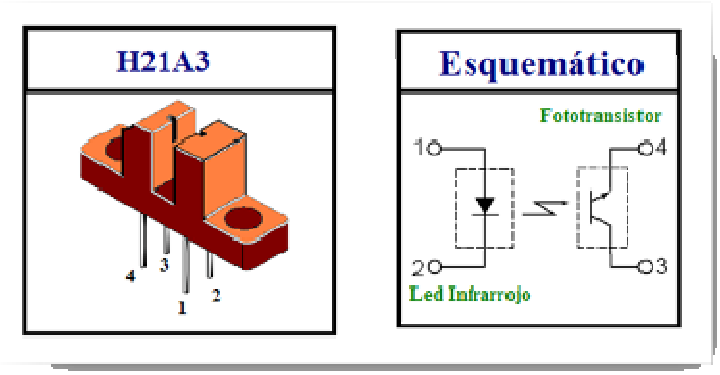


Figura 1.9 "Conmutador Óptico H21A3 y Esquemático"

[4] **Histéresis:** Es la tendencia de un material a conservar cierta propiedad debido a un estímulo externo, a pesar de que el estímulo es retirado del material.

Al existir el corte del haz lumínico se genera una señal digital (On – Off) cuya frecuencia es proporcional a la velocidad del automotor, el corte del haz es producido por el material opaco que posee el rotor interno del sensor, una vez anclado el rotor del sensor al cable del velocímetro este gira provocando un corte en el haz del conmutador óptico, el corte y la frecuencia generada dependen de la cantidad de agujeros que posea el rotor del sensor. En la Figura 1.10 se muestra un ejemplo del rotor y la señal resultante cuando este gira.

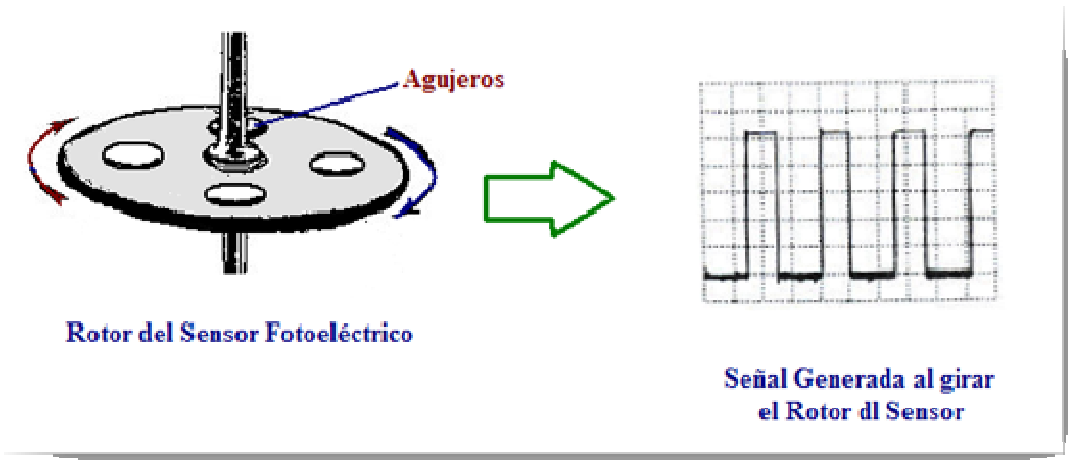


Figura 1.10 "Rotor del sensor Fotoeléctrico y Señal Generada"

La estructura del sensor a pesar de ser de un material plástico es muy resistente evitando así daños en el conmutador fotoeléctrico. Los desgastes y rupturas a causa de las grandes velocidades alcanzadas por el rotor son reducidos por recubrimientos metálicos que ayudan a preservar el correcto funcionamiento del sensor. Además la estructura posee dos piezas de ajuste para mantener el sensor fijo al cable del velocímetro.

Las limitantes para utilizar este tipo de sensor son causadas por el conmutador fotoeléctrico que posee, una de estas limitantes es la alimentación externa que necesita para su operación, si hubiese un corte de energía el sensor no podría dar lecturas correctas en el sistema que se lo emplee, de igual manera se producirían errores en las lecturas si algún desecho tapara el haz infrarrojo

producido en el sensor o si el conmutador fotoeléctrico se ensuciara con polvo o aceite propio del automotor.

En la Figura 1.11 se muestra un sensor de velocidad fotoeléctrico. Es tipo de sensores solo se utilizan en automotores que poseen velocímetros de acción mecánica.

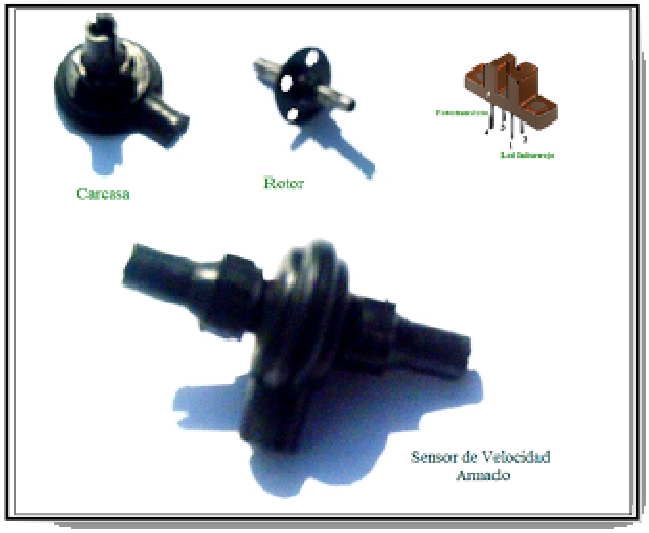


Figura 1.11 "Sensor de Velocidad Fotoeléctrico"

5.4.1.2.3 Sensores de Velocidad "Efecto Hall" [6]

Los sensores de efecto Hall se utilizan en los automóviles para medir velocidades de rotación en determinados elementos. Su principal ventaja es que pueden ofrecer datos fiables a cualquier velocidad de rotación. Y sus inconvenientes son la mayor complejidad y precio con respecto a los demás sensores que existen en el mercado.

El sensor de efecto hall se basa en un rotor que gira interrumpiendo el campo magnético de un imán enfrentado al sensor Hall. Si la pantalla del tambor permite que el campo magnético del imán incida en el generador Hall se producirá una tensión; pero cuando la pantalla interrumpe el campo magnético sobre el generador Hall la tensión resultante será igual a cero voltios, de esto se puede

concluir que el sensor de efecto Hall produce un tren de pulsos dependiendo de la incidencia de un campo magnético en el generador hall.

El sensor de efecto hall usualmente tiene tres cables, dos de los cuales son utilizados para alimentación del sensor y el restante utilizado como salida de la señal generada.

En la Figura 1.12 se muestra el esquema básico de un sensor Hall de velocidad y la conexión utilizada para su funcionamiento.

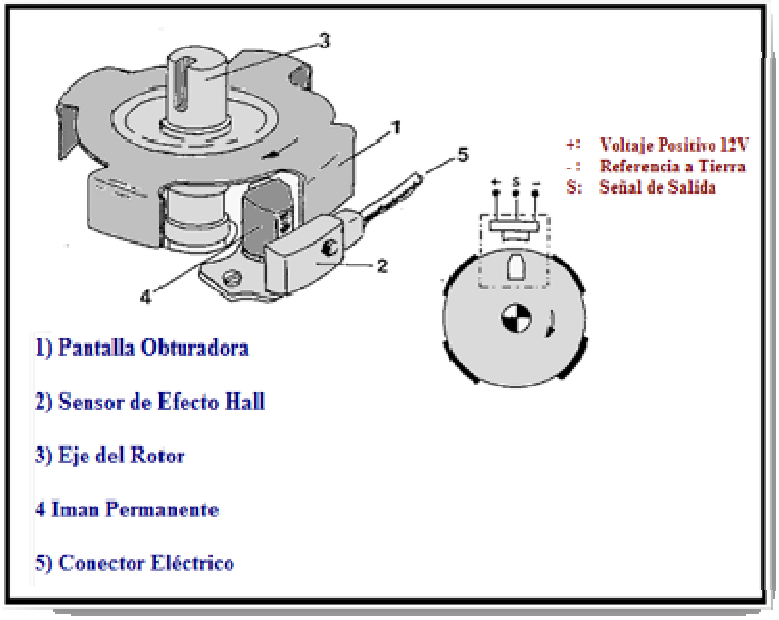


Figura 1.12 “Diagrama del sensor de Velocidad (Efecto Hall)”

Los sensores de velocidad de efecto hall proporcionan medidas muy exactas del movimiento incluso a la velocidad cero. Estos sensores generan una salida digital de amplitud constante sin importar la variación de la velocidad.

En la actualidad la mayoría de automotores emplean sensores de efecto Hall para medir su velocidad, generalmente el sensor se instala dentro de la transmisión del automotor y envía pulsos de frecuencia proporcional a la velocidad del automotor.

En la Figura 1.13 se muestra ejemplos de sensores de velocidad que se comercializan en el mercado, estos sensores vienen de fábrica instalados en el automotor.

Figura 1.13 "Sensores de Velocidad (Efecto Hall)" [7]

En la implementación del proyecto que se está elaborando se utilizará una camioneta Chevrolet LUV, esta camioneta utiliza un sensor de velocidad de tipo efecto hall, en base a las características del sensor se desarrollará el hardware necesario para obtener las medidas de velocidad del automotor.

5.5 MICROCONTROLADORES

Los microcontroladores son circuitos integrados de bajo costo y de pequeño tamaño. Estos pueden ofrecer servicios parecidos a los de un pequeño computador. En el interior de un microcontrolador se encuentra un microprocesador, una memoria RAM, una memoria ROM (generalmente de tipo FLASH), y líneas de entrada / salida.

Gracias a los avances tecnológicos algunos microcontroladores poseen otros bloques de apoyo que flexibilizan aún más su uso, tales como:

- Módulos para el control de periféricos: temporizadores, puertos serie y paralelo.
- CAD: Conversores Analógico/Digital.
- CDA: Conversores Digital/Analógico.
- Interfaz de comunicación serial (SPI, I2C, UART)
- Memorias EEPROM

Para que un microcontrolador pueda ser empleado en cualquier sistema, este tiene que ser programado. Los programas suelen ser realizados en programas compiladores (BASCOM AVR, ASAMBLER, etc.) y por medio de un programador "Quemados o Programados" en el microcontrolador.

Igualmente, para poder ser parte activa de varios sistemas no puede actuar solo sino que necesita periféricos, tanto en sus entradas como en sus salidas. A continuación, en la Figura 1.14 se presenta un diagrama general de un sistema donde su principal componente es un microcontrolador.

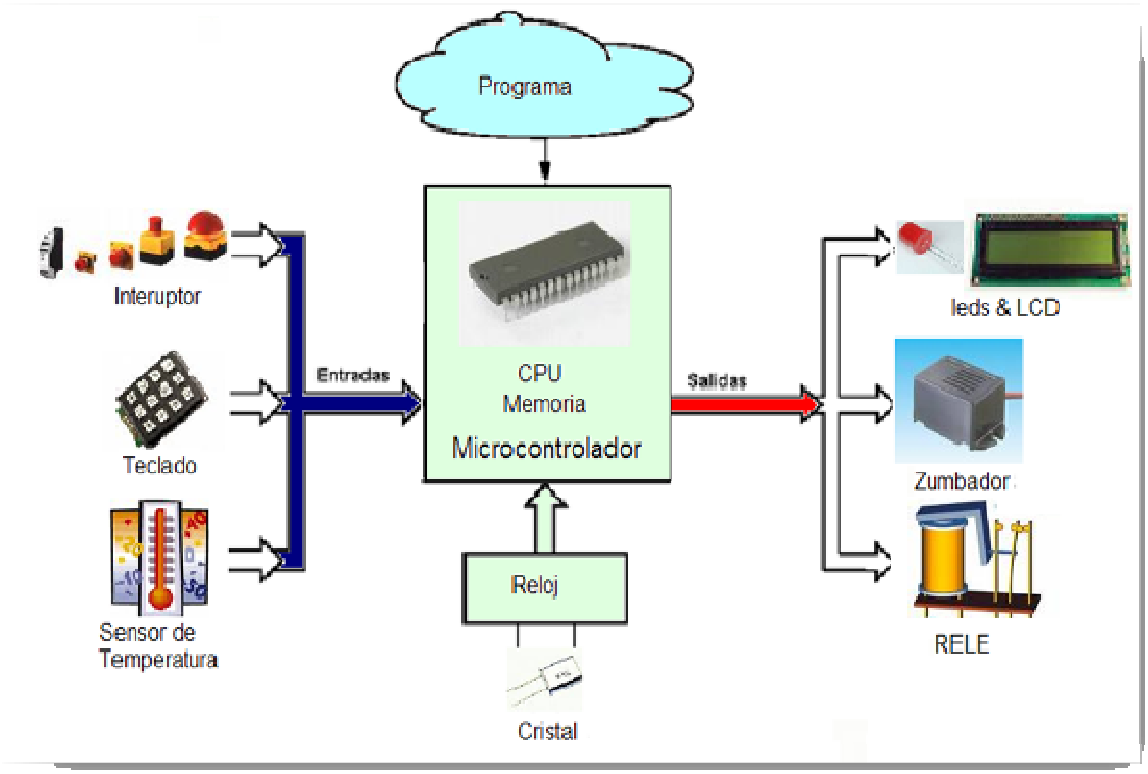


Figura 1.14 "Diagrama en bloques de un sistema en base a un Microcontrolador"

5.5.1 MICROCONTROLADORES ATMEL AVR [8]

Los microcontroladores AVR de la familia de ATMEL fueron desarrollados por la empresa SIGMA ELECTRONICA. Estos microcontroladores están basados en una nueva arquitectura RISC[5] que incorpora memoria Flash para el programa y memoria EEPROM para los datos. La cualidad insigne de estos microcontroladores es que fueron desarrollados para una total compatibilidad con lenguaje C permitiendo trabajar en alto nivel y optimizando así su algoritmo y facilitando la programación para el usuario. Además, todos los dispositivos de esta familia poseen un puerto serial que les permite reprogramar la memoria flash y así la aplicación.

Figura 1.15 "Logo de ATMEL con diversos microcontroladores"

Lo que distingue a cada microcontrolador son los periféricos y la cantidad de memoria RAM y memoria ROM que poseen. De esta forma se puede ir desde el *Tiny AVR ATtiny11* con 1Kb de memoria flash y sin RAM (sólo los 32 registros), y 8 pines, hasta el microcontrolador de la familia *MegaAVR ATmega2560* con 256KB de memoria flash, 8KB de memoria RAM, 4KB de memoria EEPROM,

---

[5] **RISC** (*del inglés Reduced Instruction Set Computer*), Computadora con Conjunto de Instrucciones Reducido. La arquitectura RISC proporciona a los microprocesadores las siguientes características fundamentales:

- Instrucciones de tamaño fijo y presentado en un reducido número de formatos.
- Posibilita la segmentación y el paralelismo en la ejecución de instrucciones y reduce los accesos a memoria.

conversor análogo digital de 10 bits y 16 canales, temporizadores, comparador analógico, JTAG, etc.

En un microcontrolador AVR la función principal la cumple el núcleo de la CPU este asegura una correcta ejecución del programa. La CPU, por lo tanto, debe acceder a memorias, realizar cálculos, controlar periféricos, y manejar interrupciones. En la Figura 1.16 se muestra la arquitectura utilizada por el microcontrolador AVR.

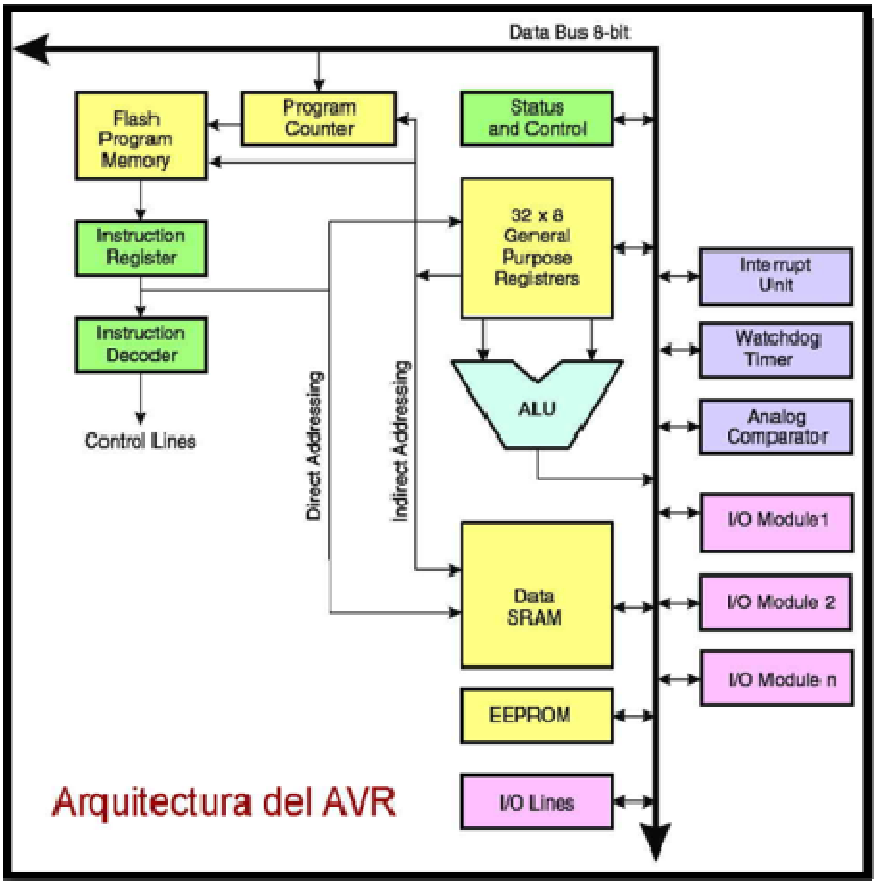


Figura 1.16 “Arquitectura del Microcontrolador AVR”

A fin de maximizar el desempeño y el paralelismo, el AVR usa una arquitectura Harvard, con memorias y buses separados para el programa y los datos. Las instrucciones que están en la memoria de programa se ejecutan en un solo ciclo de reloj este proceso es denominado “pipelining”. Mientras se ejecuta una instrucción, se extrae la siguiente instrucción de la memoria de Programa. Este

concepto permite que varias instrucciones se ejecuten en cada ciclo de reloj. La memoria de programa es la memoria flash reprogramable en el Sistema.

El Archivo de Registros que utilizan los microcontroladores AVR es de acceso rápido contiene 32 registros de trabajo de propósito general de 8 bits, con un tiempo de acceso de un solo ciclo de reloj. Esto permite la operación de la Unidad Aritmético Lógica (ALU) en un sólo ciclo.

Los microcontroladores AVR constituyen una herramienta de gran utilidad en la automatización y control de sistemas, este tipo de microcontroladores posibilitan su programación con hardware de bajo costo, y su precio es razonable en el mercado Ecuatoriano.

Debido a la cantidad de periféricos que se encuentran disponibles en el microcontrolador, se disminuye el espacio físico que requiere el sistema además de reducir el número de componentes adicionales a este. Al reducir el número de componentes, se disminuyen los costos y la posibilidad de fallo debido a estos.

5.6 MEMORIAS FLASH USB Y TARJETAS-FLASH

Figura 1.17 “Memorias Flash USB y Tarjetas Flash SD/MMC”

Las memorias se pueden clasificar siguiendo distintos criterios como: política de acceso, persistencia de información, tamaño, entre otros. Para la elaboración del sistema que nos incumbe, el tipo de memorias de interés son las memorias no volátiles, este tipo de memorias no necesitan tener una alimentación continua para conservar la información. En el grupo de memorias no volátiles se puede encontrar:

- ***ROMs (Read Only Memory)*:**
  Estas memorias son únicamente de lectura el contenido de estas no se puede alterar, ya que este se fija durante el proceso de manufacturación.

- ***PROMs (Programable Read Only Memory)*:**
  Este tipo de memoria es muy similar a la memoria ROM, estas memorias pueden ser programadas una única vez por el usuario mediante el programador adecuado.

- ***EPROMs (Erasable Programable Read Only Memory):***
  Estas memorias pueden ser programadas varias veces con el inconveniente que al momento de borrar la memoria se borra toda la información no se puede borrar sectores específicos, otro inconveniente es el demoroso proceso de borrado.

- ***EEPROM o E2PROM (Electrically Erasable Programable Read Only Memory):***
  Estas memorias son muy similares a las memorias EPROM pero con la ventaja de que el borrado es más rápido, ya que el borrado ocurre por medios eléctricos.

- ***Flash***:
  Este tipo de memoria se puede borrar eléctricamente al igual que las EEPROM. Este tipo de memorias permiten una mayor densidad de integración lo que se traduce en capacidades de almacenamiento mayores, aparte de que el coste de estas también es inferior al de las EEPROM. Otra ventaja de las memorias Flash respecto a las EEPROM es su mayor velocidad de acceso.

5.6.1 MEMORIAS FLASH

Las memorias flash son consideradas como un subgrupo de las memorias EEPROM, su estructura es muy similar. Estas contienen un arreglo de celdas con un transistor evolucionado con dos puertas en cada intersección.

Tradicionalmente sólo almacenan un bit de información. Las nuevas memorias flash, llamadas también dispositivos de celdas multi-nivel, pueden almacenar más de un bit por celda variando el número de electrones que almacenan.

Las memorias Flash se dividen en dos tipos: memorias flash tipo NOR y tipo NAND

5.6.1.1 Memorias flash de tipo NAND [9]

Las memorias flash basadas en compuertas lógicas NAND funcionan de forma ligeramente diferente: usan un túnel de inyección para la escritura y para el borrado un túnel de 'soltado'.

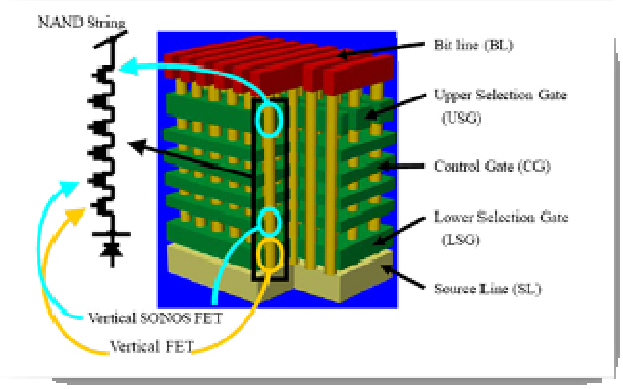


Figura 1.18 "Celda de memoria Flash del tipo NAND"

Las memorias basadas en NAND tienen, además de la evidente base en otro tipo de compuertas, un costo inferior, una mayor resistencia en las operaciones de escritura/lectura, frente a las memorias flash basadas en NOR.

La desventaja principal que tienen las memorias flash basadas en NAND es su acceso secuencial por otro lado las memorias flash basadas en NOR permiten lectura de acceso aleatorio. Sin embargo, han sido las NAND las que han permitido la expansión de este tipo de memoria, ya que el mecanismo de borrado es más sencillo lo que ha proporcionado una base más rentable para la creación de dispositivos de tipo tarjeta de memoria. Las populares memorias USB o también llamadas Pendrives, utilizan memorias flash de tipo NAND.

5.6.1.2 Memoria flash de tipo NOR [10]

Para programar o asignar un valor determinado a una celda de tipo NOR se permite el paso de la corriente desde la terminal fuente al terminal sumidero, entonces se coloca en CG un voltaje alto para absorber los electrones y retenerlos en el campo eléctrico que genera. Este proceso se denomina *hot-electron injection*. Para borrar (poner a “1L”, el estado natural del transistor) el contenido de una celda, se emplea la técnica de *Fowler-Nordheim tunnelling*, un proceso de tunelado mecánico – cuántico. Esto es, aplicar un voltaje inverso bastante alto al empleado para atraer a los electrones, convirtiendo al transistor en una pistola de electrones que permite, abriendo el terminal sumidero, que los electrones abandonen el mismo. Este proceso es el que provoca el deterioro de las celdas, al aplicar sobre un conductor tan delgado un voltaje tan alto.

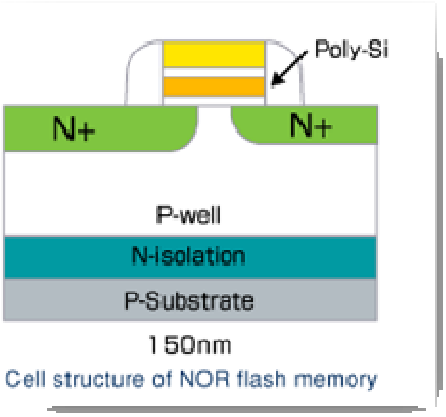


Figura 1.19 “Celda de memoria Flash del tipo NOR”

Es necesario destacar que las memorias flash están subdividas en bloques (en ocasiones denominados sectores) y por lo tanto, para el proceso de borrado, se limpian bloques enteros para agilizar este proceso. Por esta razón, las memorias flash son mucho más rápidas que las EEPROM convencionales, ya que borran o sobrescriben sectores enteros de su memoria. No obstante, para reescribir un dato es necesario limpiar el bloque primero para después reescribir su contenido.

5.6.1.3 Comparación de memorias flash basadas en NOR y NAND

Para comparar estos tipos de memoria se consideran los diferentes aspectos de las memorias tradicionalmente valorados:

- La densidad de almacenamiento de los chips mayoritariamente se basa en las memorias NAND. El coste de NOR es mucho mayor.
- El acceso NOR es aleatorio para lectura y orientado a bloques para su modificación. Sin embargo, NAND ofrece tan solo acceso directo para los bloques y lectura secuencial dentro de los mismos.
- En la escritura de NOR se puede llegar a modificar un solo bit. Esto destaca con la limitada reprogramación de las NAND que deben modificar bloques o palabras completas.
- La velocidad de lectura es muy superior en NOR (50-100 ns) frente a NAND (10 µs - 50 ns por byte).
- La velocidad de escritura para NOR es de 5 µs por byte frente a 200 µs por página en NAND.
- La velocidad de borrado para NOR es de 1s por bloque de 64 KB frente a los 2 ms por bloque de 16 KB en NAND.
- La fiabilidad de los dispositivos basados en NOR es realmente muy alta, es relativamente inmune a la corrupción de datos y tampoco tiene bloques erróneos frente a la escasa fiabilidad de los sistemas NAND que requieren corrección de datos y existe la posibilidad de que queden bloques marcados como erróneos e inservibles.

En resumen, los sistemas basados en NAND son más baratos y rápidos con una fiabilidad aceptable tomando como referencia a los sistemas basados en NOR. Se debe tomar en cuenta que los sistemas basados en NAND tienen procesos de escritura y borrado que son altamente superiores a los procesos de su contraparte, haciendo a este tipo de sistemas (sistema basado en NAND) muy atractivos para el mercado.

5.6.2 MEMORIAS FLASH USB O PENDRIVE

Las memorias Flash USB, denominadas también pendrive o USB flash drive son pequeños dispositivos de almacenamiento que utilizan una memoria flash para guardar la información. Estos dispositivos se han convertido en el sistema de almacenamiento y transporte personal de datos más utilizado, desplazando a otros dispositivos como los CD, disquete, zip. Una cualidad que ha popularizado a estos dispositivos es el de plug in play, esto significa que solo con enchufar a un conector USB de un equipo encendido (generalmente CPU del computador), este dispositivo está listo para su uso.

Una memoria USB es esencialmente una memoria flash del tipo NAND que cumplen la especificación USB 2.0, lo que les permite alcanzar velocidades de escritura/lectura de hasta 480 Mbit/s teóricos aunque en la práctica, como mucho, alcanzan unos 160 Mbit /s.

Sin embargo, puede existir dificultades para leer la información contenida en dispositivos de más 20 GB de capacidad cuando los drives no están preinstalados en el sistema operativo (Windows 95 - 2000).

Pudiendo almacenar muchísima más cantidad de información. Estos dispositivos utilizan el standard "USB mass storage" (Almacenamiento Masivo USB) para dispositivos de almacenamiento externo.

5.6.2.1 Componentes de una Memoria Flash USB [11]

A pesar de existir un sin número de modelos de memorias flash USB los principales componentes se describen a continuación:

1. **Conector USB:** Este conector provee la interfaz física entre el computador anfitrión y el dispositivo USB, este conector es Tipo A macho USB.

2. **Controlador USB de almacenamiento masivo**: Implementa el controlador USB y provee la interfaz homogénea y lineal para dispositivos USB seriales orientados a bloques, mientras oculta la complejidad de la orientación a bloques, eliminación de bloques y balance de desgaste.

3. **Pines de Prueba**: Estos se utilizan en pruebas durante la manufacturación, también se utilizan para la carga de código dentro del procesador.

4. **Memoria Flash:** Este es un chip de memoria Flash con tecnología NAND, es donde se almacenan los datos. Dependiendo de este chip se da la capacidad de las memorias flash USB.

5. **Oscilador**: Este dispositivo, es el que produce la señal de reloj para la memoria USB. Esta señal es de 12 MHz y controla la salida de datos a través de un bucle de fase cerrado.

6. **LED:** Son dispositivos visuales que emiten luz, estos se utilizan para indicar la transferencia de datos ya sean de lecturas o de escrituras.

7. **Interruptor de protección de escritura:** Utilizado para proteger los datos de operaciones de escritura o eliminación. Similar a las muescas que tenían los disquetes para evitar procesos de escritura.

8. **Espacio libre**: Este espacio es útil en sentido comercial ya que con este espacio las empresas pueden expandir la memoria de los dispositivos utilizando la misma placa.

En la Figura 1.20 se muestra las partes internas de una memoria flash USB identificando los componentes que se describieron anteriormente.

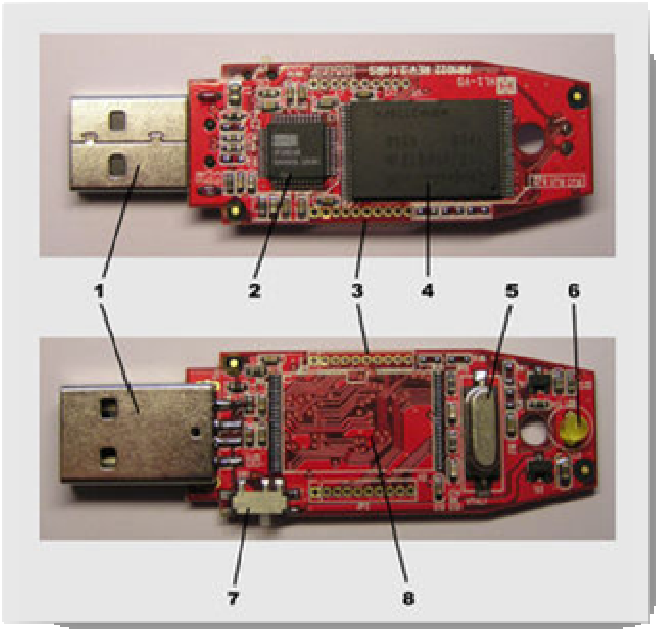


Figura 1.20 "Partes Internas de una Memoria Flash USB"

5.6.2.2 Utilidad de las Memorias Flash USB

Si bien inicialmente las memorias USB fueron concebidas para guardar y transportar datos, hoy en día es habitual encontrarlas en varios dispositivos entre los más conocidos los reproductores de MP3 hasta tarjetas wireless USB.

Las memorias USB ya no solo son dispositivos de gran almacenamiento y fácil transporte sino que han llegado a ser reales centros de entretenimiento pudiendo tener infinidad de música, videos, imágenes, documentos e inclusive programas que no necesitan ser instalados en un ordenador sino que corren desde la memoria flash USB.

En la actualidad, existen equipos de audio con entradas USB a los cuales se les puede conectar las memorias flash para poder reproducir la música almacenada en la memoria o inclusive ver las imágenes almacenadas allí.

5.6.2.3 Ventajas y desventajas

Si bien estos dispositivos son de bajo costo y una diversidad muy grande de usos, estos dispositivos pueden dejar de funcionar con facilidad por diversas circunstancias como variaciones de voltaje, caídas de lugares altos o inclusive derramamiento de líquidos sobre estas. Viendo esto las compañías que fabrican los pendrive desarrollan diferentes sistemas para su protección como, la carcasa que las protegen contra golpes, rayaduras o líquidos que puedan dañar la memoria flash.

Siendo las memorias USB unos dispositivos tan bien protegidos las hacen ideales para el transporte personal de datos, archivos de trabajo o datos personales, a los que se quiere acceder en múltiples lugares.

Si se pone en comparación el almacenamiento o el tamaño, las unidades de memoria flash son inmensamente superiores a las unidades magnéticas como los disquetes o los CD. Históricamente, el tamaño de estas unidades ha ido variando de varios megabytes hasta unos pocos gigabytes. En el año 2003 las unidades funcionaban a velocidades USB 1.0/1.1, unos 1.5 Mbit/s o 12 Mbit/s. En 2004 se lanzan los dispositivos con interfaces USB 2.0. Aunque USB 2.0 puede entregar hasta 480 Mbit/s, las unidades flash están limitadas por el ancho de banda del dispositivo de memoria interno. Por lo tanto se alcanzan velocidades de lectura de hasta 100 Mbit/s, realizando las operaciones de escritura un poco más lento.

Las memorias flash pueden soportar un número finito de ciclos de lectura/escritura antes de fallar, estas pueden soportar entre 100.000 y 1.000.000 dependiendo de la calidad de la tarjeta. Otros fabricantes como MSystem tiene módulos que aseguran hasta 5.000.000 millones de ciclos. El problema de estos últimos es su baja capacidad de almacenamiento “64MB”. Sin embargo, las

operaciones de escrituras serán cada vez más lentas a medida que la unidad envejezca. En condiciones óptimas, un dispositivo USB puede retener información durante unos 10 años.

5.6.2.4 Desarrollos futuros

Las diferentes empresas que fabrican estas memorias están haciendo un gran esfuerzo en reducir los costos de los componentes mediante la integración de varias funciones de estos dispositivos en un solo chip, esto produce una reducción de la cantidad de partes y, sobre todo, del costo total.

5.6.3 TARJETAS FLASH

5.6.3.1 Tarjetas SD (Secure Digital) [12]

Las Tarjetas SD o memorias SD son tarjetas de memoria flash. Estas tarjetas tienen unas dimensiones de 32 mm x 24 mm x 2.1 mm. Su delgado y compacto diseño promueve la facilidad de manejo y flexibilidad para moverse entre los diferentes dispositivos multimedia.

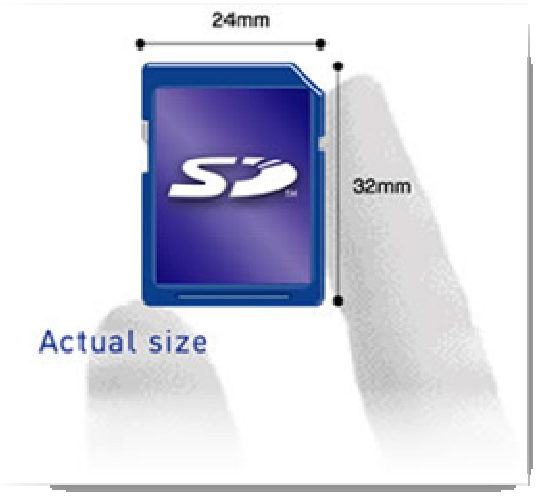


Figura 1.21 “Tarjeta de memoria SD”

Las memorias SD funcionan a diversas velocidades según la aplicación para las que son requeridas. Algunas cámaras fotográficas digitales requieren tarjetas de alta velocidad para poder grabar vídeo con fluidez o para capturar múltiples fotografías en una sucesión rápida.

Este tipo de memorias se han popularizado por los medios digitales, como son: cámaras, filmadoras, dispositivos multimedia, reproductores de MP3, PDA que poseen la ranura para memorias SD, inclusive existe variantes de memorias SD, como las memorias mini SD y la micro SD que son muy utilizadas en teléfonos móviles.

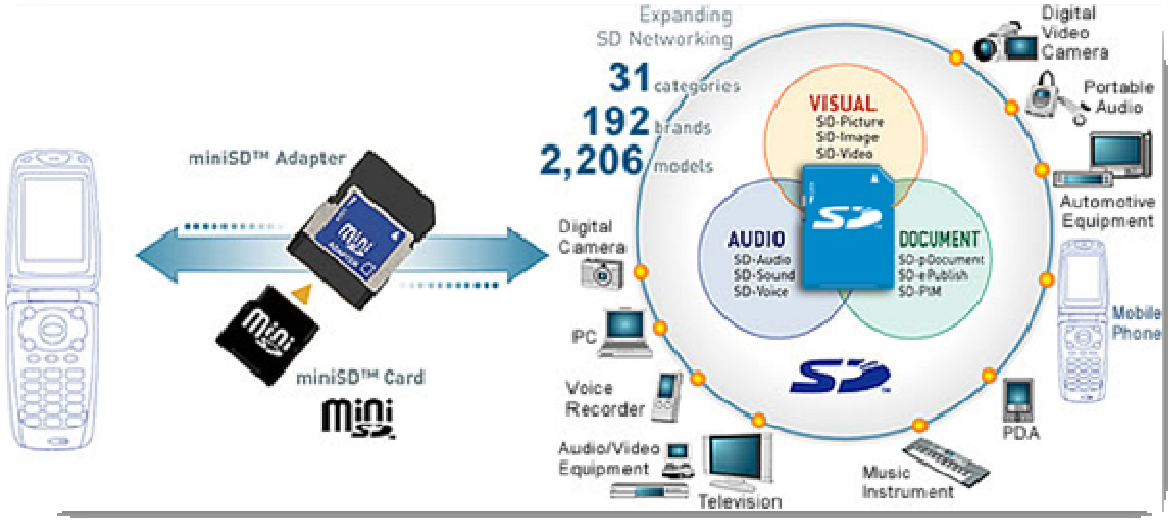


Figura 1.22 “Medios Multimedia que soportan tarjetas SD”

Un reciente desarrollo son las tarjetas SD con conectores USB integrados, para eliminar la necesidad de disponer de un adaptador SD/USD o una ranura SD en el PC, aunque a cambio de un precio inicial más alto. Un diseño pionero de SanDisk tenía una aleta que giraba y dejaba al descubierto el conector. Aunque SanDisk no pretendía en un primer momento comercializar una tarjeta SD con USB, este movimiento animó a otros fabricantes a seguir el modelo.

5.6.3.1.1 Tecnología de protección de contenido

Si bien numerosas empresas fabrican este tipo de memorias, deben estar licenciadas a través de la Secure Digital Card Association (Asociación de la Tarjeta Secure Digital). El acuerdo de licencia actual de esta organización no permite controladores de código abierto para lectores de tarjetas SD, un hecho que genera consternación en las comunidades de código abierto y software libre.

Esto significa que SD es menos abierto que CompactFlash o los llaveros USB que pueden ser implementados libremente (aunque requieren costes de licencia por las marcas registradas y logotipos asociados), pero aun así resulta mucho más abierto que XD o Memory Stick, donde no hay disponible ni documentación pública ni implementación documentada.

CPRM es una tecnología de protección de contenido utilizada por las tarjetas de memoria SD CPRM asegura un elevado nivel de protección contra la copia ilegal, esta fue desarrolladla por 4C (IBM, Intel, Matsushita y Toshiba) para llevar un historial de las copias hechas y limitar el número de copias. Las tarjetas SD poseen el sistema CPRM de control que permite que los datos sean leídos y escritos (en una zona de protección) sólo cuando un dispositivo externo apropiado es detectado. Un check-out (copia) desde un PC a la tarjeta de memoria SD se produce al ingresar la memoria SD a la ranura, esta copia se limita a 3 ejemplares en el cumplimiento de la especificación SDMI. Todos los productos SD-Audio cumplen con SDMI. En otras palabras, CPRM limita la cantidad de copias a distintos ordenadores, siendo una herramienta eficaz para evitar la piratería o copias ilegales.

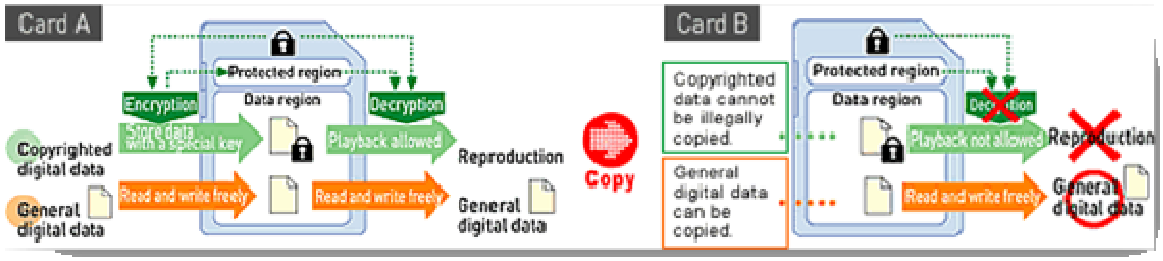


Figura 1.23 “Sistema CPRM”

La tarjeta de memoria SD funcionan como dispositivo de protección de derechos de autor por medio de las siguientes características:

- El acceso a la tarjeta de memoria SD deben tener la posibilidad de autenticación entre los dispositivos
- Un número aleatorio se genera cada vez que hay autenticación mutua y el intercambio de información de seguridad

5.6.3.2 Tarjetas MMC [13]

Las tarjetas MultiMedia Card son dispositivos de almacenamiento Flash muy versátiles pero paulatinamente descontinuadas debido al aparecimiento de las memorias SD. Sin embargo, en el mercado todavía se siguen fabricando dispositivos digitales (teléfonos, agendas electrónicas o videojuegos portátiles) con la ranura para tarjetas MultiMedia Card (MMC).

.

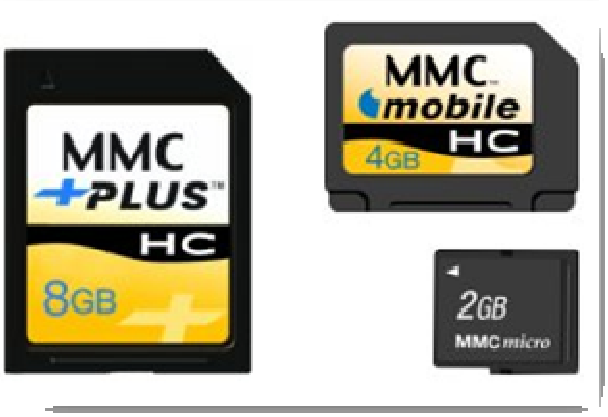


Figura 1.24 “Tarjetas MMC”

Prácticamente igual a la SD, carece de la pestaña de seguridad que evita sobrescribir la información grabada en ella. Su forma está inspirada en el aspecto de los antiguos disquetes de 3'5 pulgadas. Actualmente ofrece una capacidad máxima de 8GB.

Desde la introducción de la tarjeta Secure Digital y la ranura SDIO (Secure Digital Input/Output), pocas compañías fabrican ranuras MMC en sus dispositivos, pero las MMCs, ligeramente más delgadas y de pines compatibles, pueden usarse en casi cualquier dispositivo que soporte tarjetas SD.

5.6.3.2.1 Conexiones y señales [14]

El protocolo de acceso para una tarjeta MMC es el SPI este protocolo es muy común, se puede implementar mediante cualquier controlador (ordenador personal, microcontrolador) que disponga de un canal SPI.

Figura 1.25 "Tarjeta MultiMedia Card y la numeración de los pines"

En la Tabla 1.1 se resume la configuración de pines utilizados por las tarjetas MMC en la Interfaz SPI

# Pin:	Nombre:	Tipo:	Descripción SPI:
1	¡CS	Entrada	Chip Select (activo a 0)
2	DataIn	Entrada	Línea de datos y comandos hacia la tarjeta.
3	VSS1	Alimentación	Masa
4	VDD	Alimentación	Alimentación
5	CLK	Entrada	Clock
6	VSS2	Alimentación	Masa
7	DataOut	Salida	Línea de datos y status de la tarjeta al microcontrolador.

Tabla 1.1 "Líneas de la interfaz SPI"

Bus SPI

SPI es un bus de tres líneas sobre el cual se transmiten paquetes de información de 8 bits. Cada una de estas tres líneas porta la información entre los diferentes dispositivos conectados al bus. Cada dispositivo conectado al bus puede actuar como transmisor y receptor al mismo tiempo, por lo que este tipo de comunicación serial es full dúplex. Dos de estas líneas trasfieren los datos (una en cada dirección) y la tercer línea es la del reloj. Algunos dispositivos solo pueden ser transmisores y otros solo receptores, generalmente un dispositivo que tramite datos también puede recibir.

Todas las líneas del bus transmiten la información sobre una sola dirección.

- La señal sobre la línea de reloj (SCLK) es generada por el maestro y sincroniza la transferencia de datos.
- La línea MOSI (Master Out Slave In) transporta los datos del maestro hacia el esclavo.
- La línea MISO (Master In Slave Out) transporta los datos del esclavo hacia el maestro.

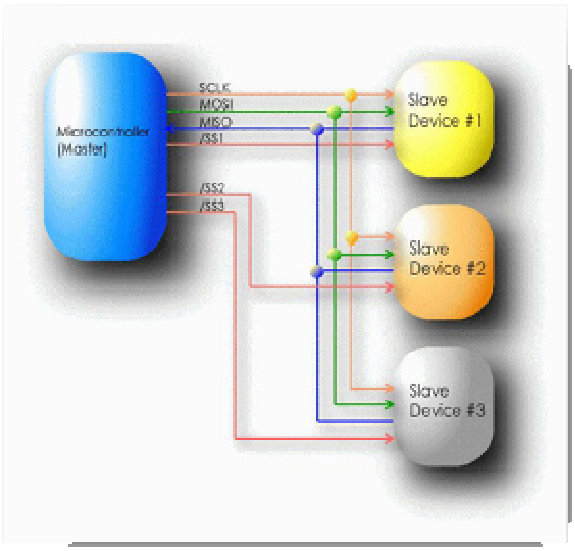


Figura 1.26 "Transmisión SPI, modelo Master-Slave"

Cada esclavo es seleccionado por un nivel lógico bajo ('0'L) a través de la línea (CS = Chip Select o SS Slave Select). Los datos sobre este bus pueden ser

transmitidos a una razón de casi cero bits /segundo hasta 1 Mbits/ segundo. Los datos son transferidos en bloques de 8 bits, en donde el bit más significativo (MSB) se transmite primero.

En la lectura, el controlador envía el comando de petición de lectura a la tarjeta y esta le envía la respuesta de confirmación seguida del bloque de datos con la información contenida a partir de la dirección solicitada.

En la escritura el proceso es parecido, el controlador indica a la tarjeta mediante el comando de escritura que quiere escribir información en una determinada dirección, esta le responde indicando que esta lista y a continuación el controlador envía el bloque de datos a escribir. Las operaciones que no requieren intercambio de datos funcionan de igual forma pero sin usar bloques de datos.

5.6.4 ESTANDARES UTILIZADOS EN LAS TARJETAS FLASH

Según el avance de la tecnología y las exigencias del mercado las tarjetas flash han ido evolucionando tanto en tamaño, capacidad, y velocidades. Esto ha generado un sin fin de memorias con diversas características.

Por ejemplo con las memorias SD se han ido desarrollando conjuntamente estándares como memorias mini SD y micro SD que agrandan la versatilidad del mercado.

5.6.4.1 Comparativa técnica

Debido al gran número de tarjetas flash y sus derivaciones se tomaran las principales tarjetas. Para esto nos ayudaremos de la Tabla 1.2 que se presenta a continuación:

Las tarjetas MMC pueden ser eléctricamente idénticas a las tarjetas SD pero en una carcasa más fina y con un fusible para deshabilitar las funciones de SD.

	MMC	MMC Plus	Secure MMC	SD	SDIO	MiniSD	MicroSD
Socket SD	Sí	Sí	Sí	Sí	Sí	Adaptador electromecánico	Adaptador electromecánico
Pines	7	13	7	9	9	11	8
Factor de forma	Fino	Fino	Fino	Grueso	Grueso	Estrecho/corto/fino	Estrecho/corto/extrafino
Ancho	24 mm	24 mm	24 mm	24 mm	24 mm	20mm	11 mm
Largo	32 mm	32 mm	32 mm	32 mm	32+ mm	21'5 mm	15 mm
Grosor	1'4 mm	1'4 mm	1'4 mm	2'1 mm	2'1 mm	1'4 mm	1 mm
Modo SPI	Opcional	Opcional	Necesario	Necesario	Necesario	Necesario	Opcional
Modo 1 bit	Sí	Sí	Sí	Sí	Sí	Sí	Sí
Modo 4 bits	No	Sí	Sí	Opcional	Opcional	Opcional	Opcional
Modo 8 bits	No	Sí	Sí	No	No	No	No
SPI XFR máximo	20 Mbps	54 Mbps	20 Mbps	25 Mbps	25 Mbps	25 Mbps	25 Mbps
DRM	No	No	Sí	Sí	N/D	Sí	Sí
Cifrado de usuario	No	No	Sí	No	No	No	No
Especificación simplificadas	Sí	No	Sí	Sí	Sí	No	No
Royalties de tarjeta de memoria	Sí	Sí	Sí	Sí	Sí	Sí	Sí
Compatible con desarrollo de código abierto	Sí	No	Sí	Sólo SPI	Sólo SPI	Sólo SPI	Sólo SPI

Tabla 1.2 "Comparativa Técnica de diversas Tarjetas Flash"

5.7 BUS SERIAL UNIVERSAL – USB [15]

El **USB** (*Bus de serie universal*), como su nombre lo sugiere, se basa en una arquitectura de tipo serial. Sin embargo, es una interfaz de entrada/salida mucho más rápida que los puertos seriales estándar. La arquitectura serial se utilizó para este tipo de puerto por dos razones principales:

- La arquitectura serial le brinda al usuario una velocidad de reloj mucho más alta que la interfaz paralela debido a que este tipo de interfaz no admite frecuencias demasiado altas (en la arquitectura de alta velocidad, los bits que circulan por cada hilo llegan con retraso y esto produce errores);

- Los cables seriales resultan mucho más económicos que los cables paralelos.

USB fue diseñado para permitir que muchos periféricos puedan estar conectados a través de una única interfaz y para mejorar el plug-and-play permitiendo a los dispositivos la capacidad de conectarse o desconectarse sin necesidad de reiniciar el equipo (hot swapping). Otra característica importante de USB es el consumo mínimo de corriente lo cual permite que muchos dispositivos estén anclados a la vez.

El USB se basa en el protocolo denominado paso de testigo. Un controlador USB distribuye un testigo por el bus. El dispositivo cuya dirección coincide con la que porta el testigo responde aceptando o enviando datos al controlador. Este también gestiona la distribución de energía a los periféricos que lo requieran. Un dispositivo USB puede usar cualquiera de cuatro tipos de transferencias existentes y una sus tres velocidades de transferencia de datos, esto de acuerdo a sus requerimientos de funcionalidad.

5.7.1 STÁNDARES USB

El estándar USB 1.0 ofrece dos modos de comunicación:

- 12 Mb/s en modo de alta velocidad,
- 1,5 Mb/s de baja velocidad.

El estándar USB 1.1 brinda varias aclaraciones para los fabricantes de dispositivos USB, pero no cambia los rasgos de velocidad. Los dispositivos certificados por el estándar USB 1.1 llevan el siguiente logotipo:

Figura 1.27 "Logotipo del estándar USB 1.1"

El estándar USB 2.0 permite alcanzar velocidades de hasta 480 Mbit/s. Los dispositivos certificados por el estándar USB 2.0 llevan el siguiente logotipo:

Figura 1.28 "Logotipo del estándar USB 2.0"

La compatibilidad entre USB 1.0, 1.1 y 2.0 está garantizada. Sin embargo, el uso de un dispositivo USB 2.0 en un puerto USB de baja velocidad (es decir 1.0 ó 1.1) limitará la velocidad a un máximo de 12 Mbit/s. Además, es probable que el

sistema operativo muestre un mensaje que indique que la velocidad será restringida.

5.7.2 COMPONENTES FÍSICOS DEL SISTEMA USB

Los componentes físicos del Bus Serie Universal consisten de circuitos, conectores y cables entre un host y uno o más dispositivos.

5.7.2.1 Dispositivos

Los dispositivos físicos USB proveen funcionalidad adicional al host. En bajo nivel un dispositivo puede referirse a un simple componente de hardware como una memoria. En alto nivel un dispositivo puede referirse a una colección de componentes de hardware que realizan una función particular, por ejemplo un Data/Fax MODEM, hub, etc.

Para asistir al host en identificación y configuración de dispositivos USB, cada dispositivo lleva y reporta información relacionada con la configuración. Otra información es específica con respecto a la funcionalidad provista por el dispositivo. El formato detallado de esta información varía y depende de la clase del dispositivo.

5.7.2.2 Cables

El cable USB consiste de 4 hilos o conductores, como muestra la Figura 1.29, blindado para transmisiones a 480 Mbps o 12 Mbps y no blindado para transmisiones a 1.5 Mbps.

De los cuatro conductores que lleva el cable, un par son trenzados y están destinados para la transmisión de datos: las líneas D_+ y D_-, en tanto que los otros dos no son trenzados y están destinados para alimentación V_{BUS} y GND.

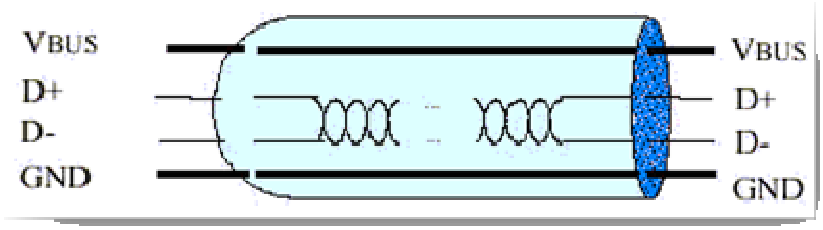
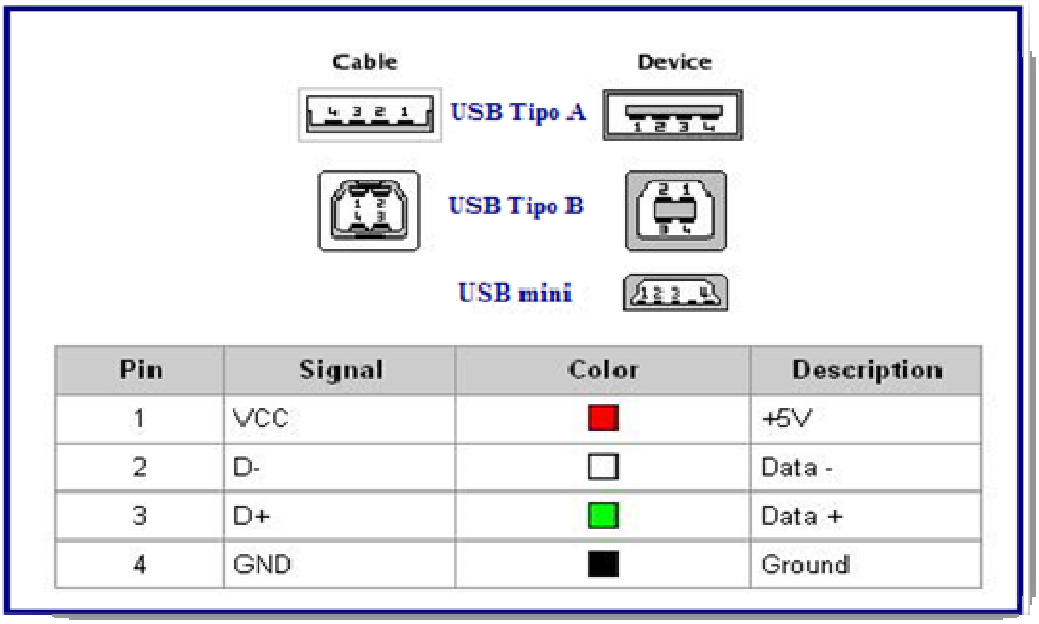


Figura 1.29 "Composición de cable USB"

El calibre de los conductores destinados a alimentación de los periféricos varía desde 20 a 26 AWG, mientras que los conductores de señal son de 28 AWG.

5.7.2.3 Conectores

En cuanto a conectores, la especificación los define básicamente como conector (macho) y receptáculo (hembra) de dos tipos: Tipo A y Tipo B. El conector Tipo A es bastante común dentro de los dispositivos listos para ser empleados con el host, y lo más probable es que tengan su propio cable con su conector Tipo A, como por ejemplo ratones, teclados, etc. La Figura 1.30 muestra los conectores Tipo A y Tipo B.



Pin	Signal	Color	Description
1	VCC		+5V
2	D-		Data -
3	D+		Data +
4	GND		Ground

Figura 1.30 "Estándares de Cable USB"

Los conectores de la Tipo B se emplean en los dispositivos USB que no poseen cable incorporado, para los cuales el conector Tipo B será una característica, como por ejemplo impresoras, scanners, y módems. Debido a que ambos conectores son físicamente diferentes una inserción equivocada es imposible incluso por la forma de las ranuras

5.7.3 ESPECIFICACIÓN ELÉCTRICA DEL USB

Una característica muy conveniente del USB es la habilidad que tiene de proveer corriente a los dispositivos que estén conectados al bus. Esto tiene varias ventajas; por ejemplo, elimina la necesidad de una fuente externa y hace al dispositivo pequeño y ligero.

5.7.3.1 Voltajes

Los voltajes nominales entre los puntos de conexión: VBUS y GND en un cable USB es 5 voltios, aunque dependiendo del tipo de dispositivo se permite un rango de variación, como se aprecia en la Tabla 1.3.

Dispositivo	Voltaje mínimo	Voltaje máximo
Alto Consumo	4.75	5.25
Bajo Consumo	4.4	5.25

Tabla 1.3 "Rango de Variación de Voltajes en el puerto USB"

Debido a pérdidas en los cables y otras más, un dispositivo debería ser capaz de funcionar con fuentes de voltaje de unas pocas décimas de voltios menos que el mínimo disponible en el puerto. El USB ha hecho una clasificación de los dispositivos de acuerdo al origen de su fuente de alimentación.

5.7.3.2 Dispositivos alimentados por el bus (Bus – Powered)

Son todos los dispositivos (incluyendo hubs) que drenan corriente desde el bus USB. La máxima corriente que se puede drenar del bus es de 500 mA.

5.7.3.3 Dispositivos Auto – Alimentados (Self – Powered)

Son dispositivos (incluyendo hubs) que no consumen corriente desde el bus USB. Un dispositivo de estas características tiene su propia fuente de alimentación.

Este tipo de clasificación no solo incluye a dispositivos que necesitan corrientes superiores a los 500 miliamperios, sino a dispositivos que necesitan operar aún cuando no estén conectados al bus USB. Sin embargo, un dispositivo auto – alimentado debe tener una conexión a V_{BUS}, para detectar la presencia de la corriente en el bus, incluso si éste no la necesita.

Durante la enumeración (proceso en el que el host configura al dispositivo), el host distingue si el dispositivo es auto – alimentado o alimentado desde el bus y la máxima corriente que el dispositivo drenará del bus. Existe un estado del USB denominado “Suspendido” que asegura que un dispositivo no consume corriente del bus cuando el host no tiene necesidad de comunicarse con éste. Un dispositivo entra en este estado cuando no hay actividad en el bus por al menos 10 milisegundos, o cuando el host envía un pedido de estado suspendido a los dispositivos del hub.

5.7.4 FUNCIONAMIENTO DEL USB

Una característica de la arquitectura USB es que puede proporcionar fuente de alimentación a los dispositivos con los que se conecta. Para poder hacerlo, utiliza un cable que consta de cuatro hilos (la conexión a tierra GND, la alimentación del BUS y dos hilos de datos llamados D- y D+).

La comunicación entre el host y los dispositivos se lleva a cabo según un protocolo basado en el principio de red en anillo. Esto significa que el ancho de banda se comparte temporalmente entre todos los dispositivos conectados. El host emite una señal para comenzar la secuencia cada milisegundo, el intervalo de tiempo durante el cual le ofrecerá simultáneamente a cada dispositivo la oportunidad de "hablar". Cuando el host desea comunicarse con un dispositivo, transmite una red (un paquete de datos que contiene la dirección del dispositivo cifrada en 7 bits) que designa un dispositivo, de manera tal que es el host el que decide "hablar" con los dispositivos. Si el dispositivo reconoce su dirección en la red, envía un paquete de datos (entre 8 y 255 bytes) como respuesta. De lo contrario, le pasa el paquete a los otros dispositivos conectados. Los datos que se intercambian de esta manera están cifrados conforme a la codificación NRZI.

Como la dirección está cifrada en 7 bits, 128 dispositivos (2^7) pueden estar conectados simultáneamente a un puerto de este tipo. En realidad, es recomendable reducir esta cantidad a 127 porque la dirección 0 es una dirección reservada.

Debido a la longitud máxima de 5 metros del cable entre los dos dispositivos y a la cantidad máxima de 5 concentradores (a los que se les suministra energía), es posible crear una cadena de 25 metros de longitud.

Los puertos USB admiten dispositivos Plug and play de conexión en caliente. Por lo tanto, los dispositivos pueden conectarse sin apagar el equipo. Cuando un dispositivo está conectado al host, detecta cuando se está agregando un nuevo elemento gracias a un cambio de tensión entre los hilos D+ y D-. En ese momento, el equipo envía una señal de inicialización al dispositivo durante 10 ms para después suministrarle la corriente eléctrica mediante los hilos *GND* y *VBUS* (hasta 100 mA). A continuación, se le suministra corriente eléctrica al dispositivo y temporalmente se apodera de la dirección predeterminada (dirección 0). La siguiente etapa consiste en brindarle la dirección definitiva (éste es el procedimiento de *lista*). Para hacerlo, el equipo interroga a los dispositivos ya

conectados para poder conocer sus direcciones y asigna una nueva, que lo identifica por retorno. Una vez que cuenta con todos los requisitos necesarios, el host puede cargar el driver adecuado.

5.8 COMO SE COMUNICA EL HOST CON EL DISPOSITIVO

Bajo Windows, cualquier comunicación con un dispositivo USB debe pasar a través de un driver de dispositivo que conozca como comunicar los drivers del sistema USB con la aplicación que accede al dispositivo.

5.8.1 CONCEPTOS BÁSICOS SOBRE LOS DRIVER

Un driver es un componente de software que habilita a las aplicaciones el acceso al hardware del dispositivo (comunicación de nivel aplicación y nivel físico), el mismo que debe conocer como comunicarse con los drives del bus de bajo nivel que controlan el hardware. Algunos drivers del dispositivo son drivers de clase que manejan comunicaciones con una variedad de dispositivos que tienen funciones similares. Es así, que los drivers cumplen con la misión de traducir entre el código a nivel de aplicación y el código a nivel de hardware.

El código a nivel de aplicación usa funciones soportadas por el sistema operativo para comunicarse con los drivers del dispositivo. El código específico de hardware maneja el protocolo necesario para acceder a los circuitos del periférico, detección de las señales de estado y cambiar las señales de control al tiempo apropiado. Windows incluye funciones API[16] que habilitan a las aplicaciones escritas por ejemplo en: Visual Basic, C/C++, Delphi, etc, poder comunicarse con los drivers del dispositivo.

[16] **API**: funciones de interfaz para programadores de aplicación, que son parte del subsistema Win32 de Windows.

5.8.2 TIPOS DE DRIVER

Existen varias formas de obtener un driver para un dispositivo: algunos dispositivos pueden usar un driver que esté incluido en Windows, o se puede usar un driver provisto por un vendedor o para otros dispositivos puede ser necesario escribir un driver personalizado.

5.8.2.1 Tipos de dispositivos estándar

Muchos periféricos caben en una clase estándar tales como impresoras, módems, teclados, ratones. Windows incluye drivers de clase para muchos tipos de dispositivos estándar.

5.8.3 PROCESO DE SELECCIÓN DE UN DRIVER BAJO WINDOWS

Cuando Windows detecta un nuevo dispositivo USB, una de las cosas que debe hacer es decidir cual driver del dispositivo debería usar la aplicación para comunicarse con el dispositivo y si es necesario, cargar el driver seleccionado. Este es el trabajo del Administrador de Dispositivos (Device Manager) de Windows.

El archivo INF[17] le dice a Windows que driver o drivers usar y que información almacenar en el registro de Windows. Cuando Windows enumera un nuevo dispositivo USB, el Administrador de Dispositivos compara los datos en todos los archivos INF del sistema con la información recuperada de los descriptores desde el dispositivo. Si este dispositivo es agregado y no tiene su propio archivo INF el Administrador de Dispositivos de Windows buscará el archivo INF más apropiado.

[17] **INF:** Archivo de texto que identifica el driver adecuado para uso del dispositivo.

5.8.4 CLASES DE DISPOSITIVOS

Muchos periféricos no son totalmente únicos, sino que comparten muchas cualidades con otros dispositivos; así, cuando un grupo de dispositivos o interfaces comparten muchos atributos o cuando proveen o piden servicios similares, toma sentido definir los atributos y servicios en una especificación de clase; la misma que sirve como una guía para quienes desarrollan dispositivos y escriben drivers del dispositivo.

Las clases ofrecen varias ventajas, y una de ellas es el hecho de que Windows y otros sistemas operativos incluyen drivers para clases comunes; y si la clase de un dispositivo es soportada por el sistema operativo no sería necesario proveer o escribir un driver para tal dispositivo, por ejemplo dispositivos de audio, dispositivos de comunicaciones, dispositivos de almacenamiento masivo de datos, impresoras, imagen, etc.

5.8.5 DISPOSITIVO DE INTERFAZ HUMANA (HID[18])

Pueden ser periféricos tales como teclados, ratones, joysticks, o cualquier dispositivo que transfiera bloques de información a velocidad moderada, usando transferencias de control o transferencias IN u OUT de interrupción. Un HID no tiene que ser un tipo de periférico estándar, y aun no necesita una interfaz humana. El único requerimiento es que el descriptor[19] almacenado en el dispositivo debe cumplir los requerimientos para los descriptores de clase HID, y el dispositivo debe enviar y recibir datos usando transferencias de interrupción o control como se define en la especificación HID, misma que se la puede encontrar en la página web www.usb.org en la sección *desarrolladores > HID tools*.

---

[18] **HID** (Dispositivo de interfaz humana).- Clase especial de dispositivos que son de uso genérico como por ejemplo mouse, teclados, scanners de códigos de barras, unidades de adquisición de datos.

[19] **Descriptores**: Estructuras de datos o bloques de información, que habilitan al host a aprender acerca del dispositivo

La principal limitación para las comunicaciones con un HID es la disponibilidad de tipos de transferencias, las mismas que como ya se mencionó se reducen al uso de las transferencias de control o transferencias IN u OUT de interrupción a partir de la segunda edición de Windows 98.

El intercambio de datos bajo la clase HID reside en estructuras llamadas reportes, así el host envía y recibe datos enviando y pidiendo reportes en la transferencias ya mencionadas; además, el formato del reporte es flexible, y puede manejar cualquier tipo de datos.

5.9 MANEJO DE FLASH USB CON UN MICROCONTROLADOR – USBWIZ [17]

Las memorias flash USB en los últimos años se han ido popularizando ya que estas son una manera práctica de almacenar información en un limitado paquete. Debido a esta gran ventaja, parece ideal para su uso en un registro de datos (data logger). Sin embargo, para poder manipular archivos de la memoria flash un microcontrolador debe tener un USB HOST y un apoyo a un sistema de ficheros propios de cada memoria flash. Ya que la mayoría de microcontroladores carecen de estas cualidades esto se convierte en un gran inconveniente.

Para simplificar el acceso a memorias flash la compañía GHI Electronics elaboro un chip denominado USBwiz. El microcontrolador puede acceder al USBwiz mediante transmisión serial (UART) o por puerto SPI o por el puerto I2C ya que el microcontrolador ya posee estos modos de transmisión, además el USBwiz también puede tener acceso a los discos duros USB y tarjetas-flash de memoria como MultimediaCards (MMCs) y Secure Digital (SD)

5.9.1 USBWIZ-OEM [18]

El USBwiz-OEM es un completo sistema, que permite con sencillos comandos establecer comunicación con archivos FAT en una memoria USB o en una tarjeta SD, estos comandos son enviados mediante puertos UART, SPI o I2C.

Figura 1.31 "Modulo USBwiz – OEM"

La USBwiz-OEM es una placa que facilita la comunicación con diferentes dispositivos USB, esta placa posee dos integrados, el primero es un microcontrolador Philips LPC2141 basado en un procesador ARM 16-32 bit en el cual ya se tienen pre programadas instrucciones para manejar los dispositivos USB y una base de datos de los diferentes drives de los dispositivos. Un segundo chip es el Philips ISP1160 que sirve como controlador Host USB, este chip realiza la verdadera comunicación con los dispositivos USB.

Esta tarjeta puede acceder simultáneamente a una tarjeta SD/MMC y dos unidades USB independientes ya que el USBwiz tiene 3 núcleos FAT[20] independientes y una manipulación de archivos muy rica de comandos. Pero eso

[20] **FAT**: File Allocation Table. Un archivo situado en el disco duro que actúa como índice del mismo. Contiene la referencia de cada uno de los de archivos almacenados allí. Si se daña puede perderse parte (generalmente todo) del contenido del disco

no es todo, este chip puede soportar varios dispositivos USB como son; Teclados, mouse, joysticks, impresoras, teléfonos celulares, FTDI, entre otros.

5.9.1.1 Configuración de pines

La interfaz de usuario es una línea de 18 agujeros que acepta cualquier tipo de conector de preferencia espadines machos. Esta interfaz es utilizada por el usuario realizar varias tareas como:

- Establecer el modo de comunicación con el USBwiz-OEM dígase de otra forma escoger el modo de comunicación ya sea en UART, I2C, SPI.
- Brindar una fuente de alimentación fija o alterna.
- Establecer el medio por donde se va a desarrollar la comunicación
- Reinicio del

En la Tabla 1.4 se resume la configuración de pines utilizada por el USBwiz-OEM.

PIN	USBwiz
1	UART_TX/DTRDY
2	UART_RS/BUSY
3	I2C_SCL
4	I2C_SDA
5	SPI_SCK
6	SPI_MISO/UART_RTS
7	SPI_MOSI/CTS
8	SPI_SSEL#
9	MISC
10	N/C
11	D6/BL# (Solo para fabricante)
12	VBAT

PIN	USBwiz
13	VCC (3.3V)
14	RESET#
15	GND
16	MODE0
17	MODE1
18	VCC (5V)

Tabla 1.4 “Pines de la placa USBwiz-OEM”

5.9.1.2 Clases de cliente USB soportadas

La organización USB define muchas clases de dispositivos USB. Esto significa que todos los dispositivos USB son de cierta clase. En el sistema operativo están pre cargados varias clases de clientes USB “USB drivers” que sin los cuales ningún dispositivo USB podría funcionar, el USBwiz viene también con driver cargados. Si una clase no es sustentada por USBwiz, se puede accesar al dispositivo por medio del comando raw USB.

5.9.1.2.1 Dispositivos USB que soporta

El USBwiz consta con una significativa base de driver con la que se pueden acoger varios dispositivos USB entre estos podemos nombrar los siguientes:

- La mayor parte dispositivos de interfaz humana (HID) como son mouse, teclados y joystick
- Impresoras con soporte ASCII
- Dispositivos de almacenamiento masivo (Mass Storage)
- Comunicación (módems y teléfonos celulares).

Figura 1.32 “Dispositivos USB que soporta el USBwiz-OEM”

5.9.1.3 Características USBwiz [19]

Esta herramienta es atractiva para diseñadores, por las características que ofrece entre otras tenemos:

- Soporta FAT32, FAT16 y FAT12.
- Acceso simultáneo a 3 dispositivos de FAT.
- Soporta tarjetas de memoria Secure Digital (SD).
- Soporta clases de HID USB.
- Fácilmente usado con cualquier microcontrolador incluyendo PIC, AVR, Zilog…etc.
- Posee varias comunicaciones como UART, I2C o SPI.
- Soporta comunicaciones UART tan altas como 921.6 K-baudios, I2C hasta 400kbps, y SPI reloj hasta arriba de 7 MHz.
- Distribución y actualización de firmware gratuita.
- Trabaja en el RTC (reloj de tiempo real)
- Consumo de corriente aproximada de 40 a 50 mA,
- Fuente de Alimentación de 3.3V – 5V.
- Rango de temperatura de funcionamiento -40ºC a +85ºC.

5.9.1.4 Aplicación

La apertura que brinda el USBwiz posibilita varios campos tecnológicos e infinidad de aplicaciones, entre las cuales tenemos:

- Cámaras Digitales
- Data Logger
- Visualización de Imágenes
- Interface con mouse, teclado
- USB-impresora a RS232
- Enviar automáticamente SMS a través de un teléfono celular

5.9.1.5 Gestor de Arranque del USBwiz

El gestor de arranque es usado para actualizar el firmware del USBwiz. El firmware es el código que se asienta en el chip USBwiz para poder hacer todo el trabajo. Este código puede ser descargado desde el sitio WEB y por medio de comandos simples ser actualizado.

Si el USBwiz está en estado virgen ose es nuevo, el firmware se carga copiando el código en una tarjeta SD he ingresándolo en la ranura correspondiente.

Comando	Uso	Notas
R	Carga y corre un firmware del USBwiz	si el gestor de arranque retorna "BL" este necesita reprogramar
LQUx	Carga un firmware desde dispositivo especifico	La x representa la ranura del dispositivo. Para LQUA corresponde a tarjetas SD/MMC, LQUB corresponde al Puerto USB0, LQUC corresponde al puerto USB1
W	Escribe un Sector de la Flash interna	La transacción debe terminarse por bye de checksum
V	Retorna la versión del firmware	Retorna un valor en ASCII

Tabla 1.5 “Comandos para el Gestor de Arranque”

5.9.1.6 Comandos USBwiz

Todas las órdenes deben ser enviadas en código ASCII. Este código al ser común para los usuarios facilita la comprensión de los comandos enviados. El set de comandos es muy amplio y facilita la manipulación de dispositivos externos.

A continuación se presentara una lista de comandos que pueden ser transmitidos por puerto serial (UART) para crear un archivo dentro de una carpeta.

1) FM 0
2) MD CARPETA1
3) CD CARPETA1
4) OF 0W<ARCHIVO.TXT
5) CF 0

El comando FM nos ayuda para montar una unidad o para ponerla activa en este caso se escogió la unidad “0 – USB0”, una vez montada la unidad se crea una carpeta de nombre CARPETA1 con el comando “MD CARPETA1”, el comando “CD CARPETA1” sirve para ingresar a la carpeta antes creada, el proceso de crear un archivo es más complejo este necesita especificar la extensión del archivo, la unidad montada, y si es un archivo nuevo o ya existente, esto obtenemos con el comando “OF 0W<ARCHIVO.TXT“, para finalizar el proceso y el archivo antes abierto quede guardado se usa el comando “CF 0”.

5.9.1.7 Puertos de Comunicación USBwiz

El USBwiz soporta tres tipos de comunicaciones. Estas comunicaciones se establecen fijando el modo de transmisión con los pines MODE0 (pin 16) y MODE1 (pin 17). Las comunicaciones que soporta el USBwiz son:

Comunicación UART

UART son las siglas de "Universal Asynchronous Receiver-Transmitter" (en español, "Transmisor-Receptor Asíncrono Universal"). Los chip que poseen comunicación UART tienen por objetivo convertir los datos recibidos en forma paralela, a forma serial, con el fin de comunicarse con otro sistema externo.

Comunicación SPI

SPI (del inglés Serial Peripheral Interface) es un estándar de comunicaciones, usado principalmente para la transferencia de información entre circuitos integrados en equipos electrónicos. El bus de interface de periféricos serie o bus SPI es un estándar para controlar casi cualquier electrónica digital que acepte un flujo de bits serie regulado por un reloj

Comunicación I^2C

I^2C es una comunicación que utiliza dos líneas para transmitir la información, una para los datos y por otra la señal de reloj. También es necesaria una tercera línea, pero esta sólo es la referencia (masa). Como suelen comunicarse circuitos en una misma placa que comparten una misma masa esta tercera línea no suele ser necesaria.

5.10 PROPUESTA PARA LA ADQUISICIÓN DE PARÁMETROS DEL MOVIMIENTO DE UN AUTOMOTOR

Luego de revisar la teoría sobre la que se sustenta este trabajo, se procede al diseño conceptual del mismo.

El esquema propuesto consta de cuatro etapas que se las puede ver en el diagrama de bloques de la Figura 1.33 “Diagrama de Bloques”a continuación.

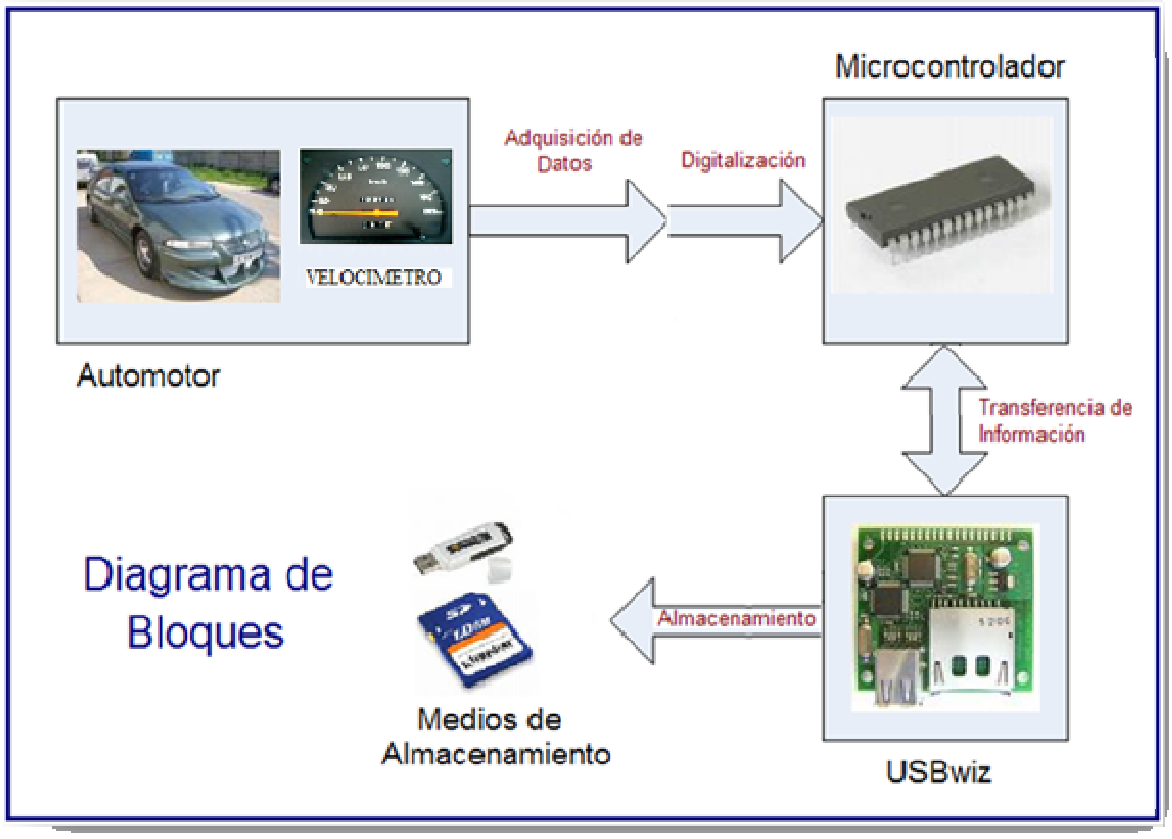


Figura 1.33 "Diagrama de Bloques"

1. **Adquisición de Datos**:

   Esta primera etapa se refiere a la obtención de datos a partir del automotor tomando la señal generada por el sensor de velocidad instalado en el mismo (Obtener la señal generada por el sensor de velocidad).

2. **Digitalización:**

   La señal generada por el sensor de velocidad es una onda cuadrada cuya frecuencia es proporcional a la velocidad del automotor, esta información por sí sola no es relevante, por lo tanto se tiene que digitalizar. El encargado de digitalizar la señal será el microcontrolador que medirá el ancho de pulso mediante software.

3. **Transferencia de Información**:

   Una vez establecido el medio y la forma de comunicación, el microcontrolador debe comunicarse con el USBwiz para este pueda almacenar los datos en cualquier medio de almacenamiento

4. **Almacenamiento**:

   Según las instrucciones del microcontrolador el USBwiz se encargara de almacenar los datos en las distintas unidades.

CAPÍTULO 2

DISEÑO Y CONSTRUCCIÓN DEL HARDWARE Y SOFTWARE DEL SISTEMA

CAPÍTULO 2

DISEÑO Y CONSTRUCCIÓN DEL HARDWARE Y SOFTWARE DEL SISTEMA

6.1 DISEÑO Y CONSTRUCCIÓN DEL HARDWARE DEL SISTEMA

6.1.1 INTRODUCCIÓN

Se mencionó ya que el objetivo principal de este proyecto es la implementación de un sistema que permita medir los parámetros de velocidad, tiempo, distancia recorridas de un automotor, y mediante el adecuado software de soporte almacenar estos parámetros a una memoria Flash USB o en una Tarjeta SD. Para realizar estas operaciones el sistema debe cumplir con varias exigencias como: adecuar la señal del velocímetro, un sistema de procesamiento de datos, un sistema que permita almacenar los parámetros del automotor.

Para tener una concepción global del sistema, a continuación se presenta un diagrama de bloques funcional del mismo:

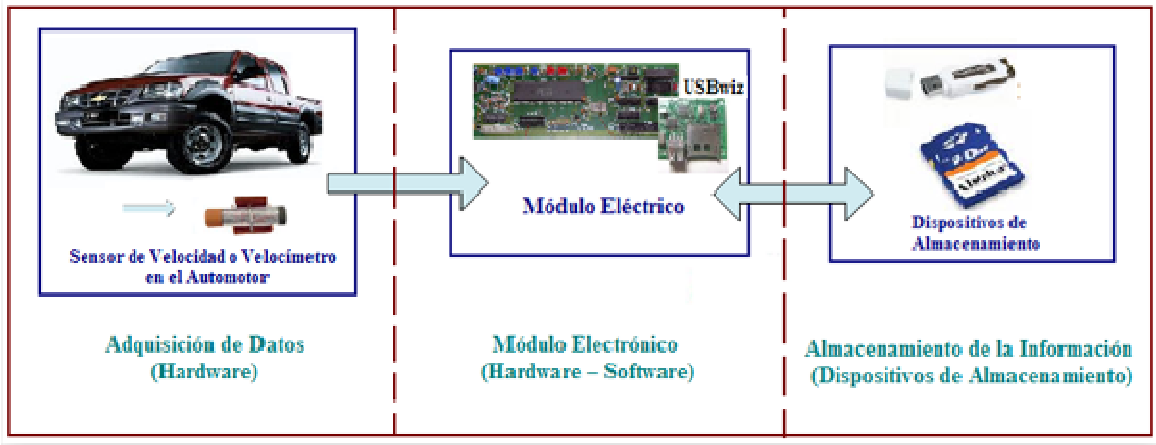


Figura 2.34 “Diagrama de Bloques funcional del Sistema”

La Figura 2.34 muestra un esquema simplificado del sistema que se pretende implementar para conseguir el objetivo planteado, el mismo que consta de tres bloques principales que se describen en detalle a continuación.

6.1.2 ADQUISICIÓN DE DATOS

El automotor a utilizar en el proyecto se muestra en la Figura 2.35. Este automotor posee un velocímetro digital del cual se obtendrá la velocidad instantánea. La velocidad obtenida servirá como base para calcular parámetros secundarios como distancia recorrida y aceleración del automotor.

El bloque adquisición de datos básicamente está conformado por el sensor de velocidad y el velocímetro que posee el automotor (Camioneta Chevrolet LUV). Dicho sensor está ubicado en la caja de cambios, midiendo internamente, con ayuda de contactos magnéticos, las revoluciones que dan los piñones dentro de la caja de cambios. El sensor posee tres conductores, dos de ellos utilizados para la alimentación y el restante utilizado para transportar la señal digital hacia el velocímetro. Cabe mencionar que no todos los automotores poseen sensores de velocidad de las mismas características; sin embargo, para adquirir el parámetro de velocidad en el proyecto se tomará como referencia la señal de salida del velocímetro ya que este procesa previamente la señal entregada por el sensor de velocidad y las revoluciones del automotor.

Figura 2.35 "Camioneta Chevrolet LUV 2005"

6.1.3 MÓDULO ELECTRÓNICO

El módulo electrónico a construirse debe permitir medir los parámetros de velocidad, tiempo y distancia recorrida de un automotor, los mismos que deberán ser almacenados en una Memoria Flash USB o en una Tarjeta SD. Para lograr esto, el módulo electrónico por medio de contadores adecúa la señal generada por el velocímetro y posibilita la medición de la velocidad instantánea del mismo; a partir de esta medida de velocidad se deducirán, usando ecuaciones incorporadas en el software del microcontrolador, las demás variables de interés como distancia y aceleración.

El tiempo será medido por medio de un microcontrolador AVR - ATmega16 de la ATMEL, el mismo que cumple con los parámetros técnicos para dar soporte al módulo electrónico

La Figura 2.36 muestra el diagrama de bloques del cual se partió para realizar la implementación del módulo electrónico.

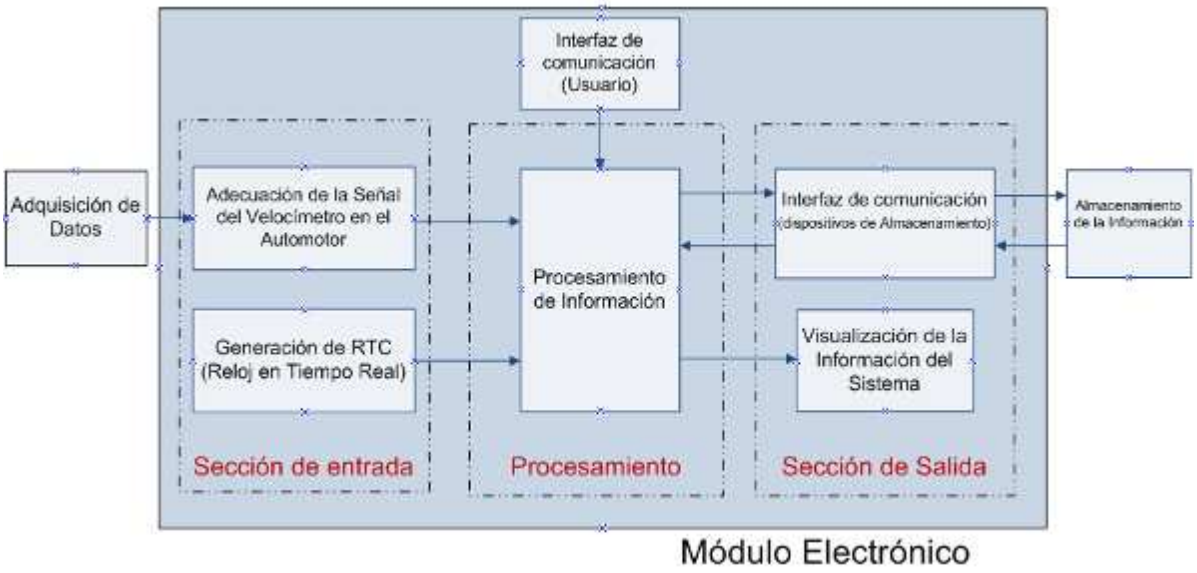


Figura 2.36 "Diagrama de Bloques del Módulo Electrónico"

- **Adecuación de la Señal**.- utilizando un divisor de voltaje reduce la amplitud de la señal generada por el velocímetro del automotor. Este proceso se realiza para manejar voltajes tolerables para el microcontrolador (0V – 5V)

- **Generación de RTC.-** circuito que se encarga de generar la hora en tiempo Real. Este circuito se basa en el integrado DS1307 que, por medio de una interfaz serial SPI con el microcontrolador, es capaz de proporcionar la hora en segundos, minutos, horas, días, meses, años.

- **Procesamiento de la Información.-** Circuito mediante el cual se toma los parámetros de velocidad y tiempo, para procesarlos mediante software y obtener la distancia del automotor. El procesamiento de información se realizan mediante software,

- **Interfaz de comunicación (Usuario).-** circuito que posibilitará el control de las actividades del sistema como: configurar el Reloj, Iniciar el almacenamiento de los parámetros del automotor, o desinstalar la unidad de almacenamiento en uso.

- **Interfaz de comunicación (Dispositivos de Almacenamiento).-** circuito mediante el cual se establece comunicación con los dispositivos de almacenamiento, cuyo objetivo es almacenar los datos adquiridos del automotor.

- **Visualización de la Información del Sistema**.- Muestra los diferentes parámetros obtenidos del automotor en un LCD y brinda información relevante de la actividad de los dispositivos de almacenamiento por medio de LEDs.

6.1.3.1 Sección de entrada

En esta sección se agrupan los diferentes dispositivos que generan o adquieren variables de entrada desde el automotor. Los bloques involucrados en esta sección son: Adecuación de la señal y generación de RTC (Reloj en tiempo Real) los mismos que se describen en detalle a continuación.

6.1.3.1.1 Adecuación de la señal (Velocímetro del Automotor)

Los niveles de voltaje que maneja el microcontrolador AVR ATmega16 oscilan de 0V a 5V; sin embargo, la señal entregada por el velocímetro del automotor tiene una amplitud de 12 V. Para que el microcontrolador admita la señal generada por el velocímetro esta primero tiene que ser adecuada (Reducir su amplitud de voltaje). En la Figura 2.37 se muestra la conexión establecida entre el sensor de velocidad con el velocímetro y los niveles de voltaje que manejan estos dispositivos.

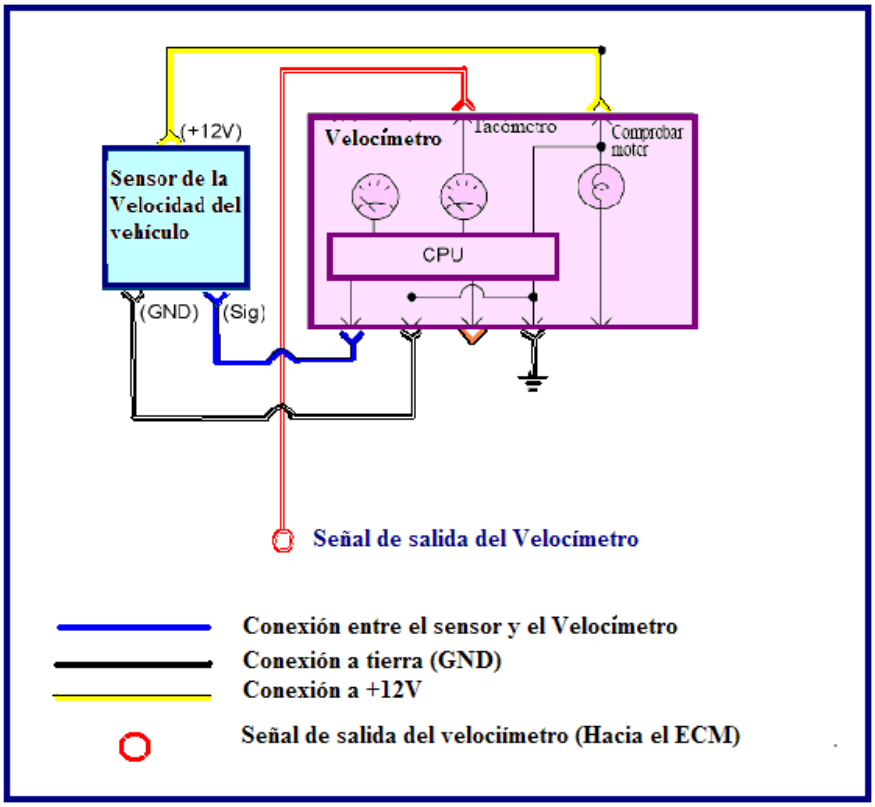


Figura 2.37 "Conexión entre el sensor de velocidad y el velocímetro"

Para reducir la complejidad en circuitería y el espacio físico en el módulo electrónico se optó por la elaboración de un divisor de voltaje. En la Figura 2.38 se muestra el circuito utilizado para adecuar la señal del velocímetro.

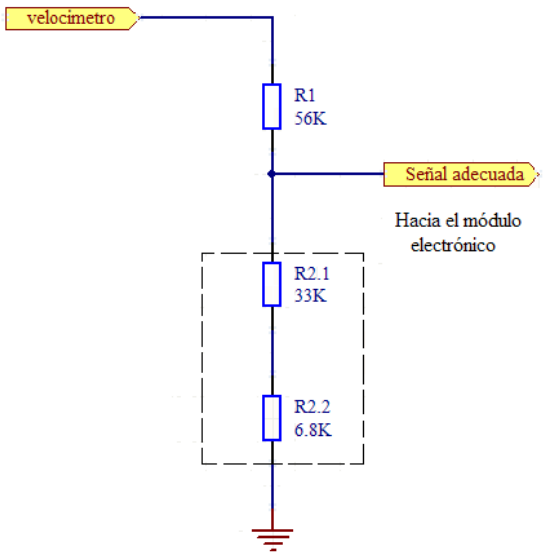


Figura 2.38 "Circuito para adecuar la señal del velocímetro"

Las resistencias utilizadas se determinaron por los siguientes cálculos:

$$V_0 = V_{IN} \times \frac{R_2}{R_1 + R_2}$$

$$R_2 = \frac{R_1}{\left(\frac{V_{IN}}{V_0} - 1\right)}$$

Se asume un valor referencial para R1

$$R_2 = \frac{56K\Omega}{\left(\frac{12}{5} - 1\right)} \approx 40K\Omega$$

$$R_2 = R_{2.1} + R_{2.2}$$

$$R_2 = 33K\Omega + 6.8K\Omega$$

$R_2 = 39.6K\Omega$

Resumen de Elementos

R1 = 56KΩ

R2.1 = 33KΩ

R2.2 = 6.8KΩ

6.1.3.1.2 Diseño del Circuito para Generación del Reloj en Tiempo Real "RTC"

Teniendo presente la necesidad de un reloj en tiempo real que pueda mantener su funcionamiento a pesar de pérdidas de energía o incoherencias en software, se desarrolló un circuito que genere un Reloj en Tiempo Real en el módulo electrónico, para que cumpla con el objetivo de proporcionar la fecha y la hora imprescindibles para el almacenamiento de datos. Otra alternativa que se consideró fue la utilización del USBwiz en la generación del RTC; sin embargo, se descartó esta alternativa por su baja resolución (2 segundos).

El circuito configurado para generar un Reloj en Tiempo Real tiene como principal elemento el integrado DS1307 manufacturado por la empresa MAXIM. El integrado DS1307 es un reloj de tiempo real (RTC) de baja potencia que opera en código binario (BCD). El código generado da la información de calendario (año, mes, día) y reloj (hora, minuto, segundo) además de utilizar una corrección automática para meses menores a 31 días, y para años bisiestos. Los datos y las direcciones se transfieren a través de una comunicación serie PC, mediante un bus bidireccional. El DS1307 opera en cualquiera formato de hora ya sea en formato de 24 horas o en formato de 12 horas AM/PM. El DS1307 tiene la posibilidad de conectar una fuente de alimentación secundaria (Batería de respaldo) para mantener sus datos a pesar de que la alimentación primaria se pierda. Para tener una referencia más profunda sobre el funcionamiento del integrado DS1307 se pone a consideración el ANEXO B-2 en el presente documento.

En la Figura 2.39 se muestra el esquema del circuito implementado para la generación del reloj en tiempo real (RTC).

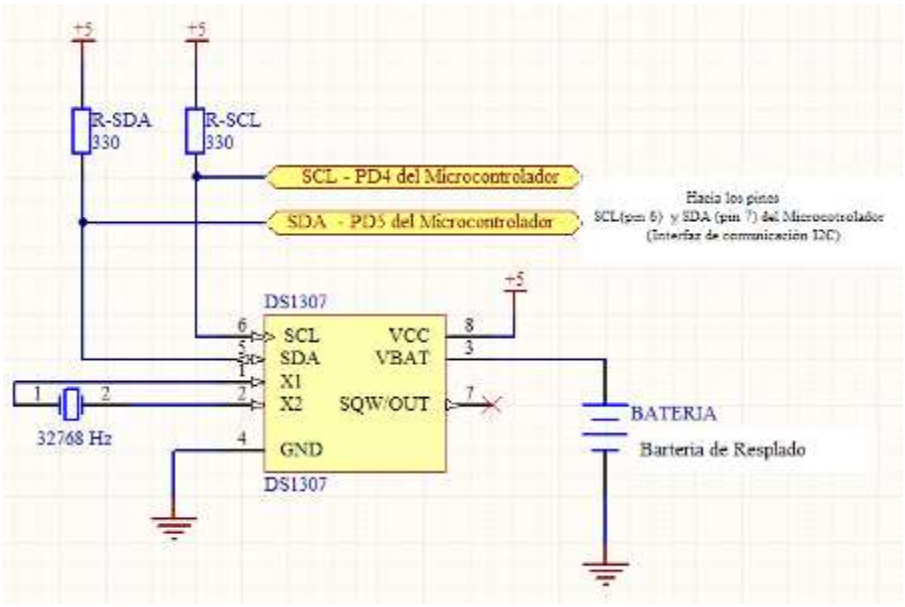


Figura 2.39 "Circuito para Generación del Reloj en tiempo Real "RTC"

Los pines SCL (pin 6) y SDA (pin 7) están conectados al microcontrolador para poder establecer la comunicación I2C por medio de la cual se puede controlar y obtener los valores del reloj en tiempo real. Debido a que los pines tienen una configuración de colector abierto, se deberá colocar una resistencia entre VCC y estos pines, la resistencia recomendada por el fabricante es de 330 Ω. Este circuito posee una batería de apoyo para evitar la interrupción de su funcionamiento y un cristal que genera el reloj interno del DS1307. A continuación se presenta en detalle la información de cada pin del integrado DS1307.

- **VCC, GND:** Estos pines son para la alimentación del integrado que normalmente es de 5V. Cuando aparte del voltaje VCC se aplica el voltaje de la batería, el voltaje VCC debe ser mayor a (1.25 x VBAT); sin embargo, la función del reloj en tiempo real no se ve alterada por caídas de voltaje sufridas en VCC, si se tiene un respaldo de Voltaje en VBAT.

- **VBAT:** Es el pin donde se ingresa el voltaje de respaldo (Voltaje de la batería). Se puede utilizar una batería estándar de litio de 3V u otra fuente de alimentación. El rango de batería que soporta el DS1307 es de 2V a 3.5v. Cabe notar que con una batería convencional de 48mAh, el dispositivo podrá estar respaldado por más de 10 años.

- **SCL (Serial Clock Input):** El pin SCL es usado para sincronizar la transmisión de los datos por la interfaz serial. El pin SCL funciona como colector abierto, por lo que necesita una resistencia de 330 Ω entre VCC y este pin.

- **SDA (Serial Data Input/Output):** SDA es el pin de entrada/salida de datos para la interfaz serial. El pin SDA funciona como colector abierto, por lo que necesita una resistencia de 330 Ω entre VCC y este pin.

- **SQW/OUT (Onda Cuadrada/Output Driver):** Cuando se habilita el SQWE (SQWE=1), la salida de este pin es una señal cuadrada de frecuencias (1Hz, 4kHz, 8kHz, 32kHz). Este pin funciona también en colector abierto por lo que necesita una resistencia de 330 Ω conectada entre VCC y este pin.

- **X1, X2**: El DS1307 necesita un cristal de 32.768 KHz para su correcto funcionamiento, en estos pines es donde se debe conectar dicho cristal.

Resumen de Elementos:

Batería CR2032 y porta Batería

Cristal 32.768 KHz

R_SDA = 330Ω

R_SCL = 330Ω

Figura 2.40 "Cristal 32.768 KHz"

Figura 2.41 "Batería CR2032 y porta Batería"

6.1.3.2 Sección de Salida

Una vez procesada la información proveniente del velocímetro del automotor esta información debe ser mostrada en dispositivos visuales y almacenada en las distintas unidades. En esta sección del módulo electrónico se agrupan los diferentes dispositivos para lograr el propósito de visualización y almacenamiento. Los bloques involucrados en esta sección son: Interfaz de comunicación (Dispositivos de Almacenamiento) y Visualización de la Información del Sistema.

6.1.3.2.1 Interfaz de comunicación con los Dispositivos de Almacenamiento

Para poder almacenar los parámetros de velocidad, tiempo y distancia del automotor se debe establecer una interfaz de comunicación entre el microcontrolador y los diferentes dispositivos de almacenamiento. Con este propósito se incluye en el módulo electrónico un dispositivo denominado USBwiz-OEM con el cual se puede almacenar los datos procesados por el microcontrolador AVR ATmega16 en una memoria Flash o tarjeta SD.

La interfaz de comunicación con los dispositivos de almacenamiento consta de una placa principal llamada USBwiz-OEM. Esta placa es manufacturada por la compañía GHI Electronics. El USBwiz-OEM facilita la comunicación con las memorias USB flash, y tarjetas SD, posibilitando utilizar estas memorias como el dispositivo de almacenamiento del sistema que se está elaborando.

En el proyecto, la placa del USBwiz será la interfaz para utilizar las diversas memorias flash y tarjetas SD. Si bien este dispositivo se puede comunicar de tres formas (SPI, I2C, UART), se utilizará la comunicación UART (transmisión recepción asíncrona universal) en el proyecto.

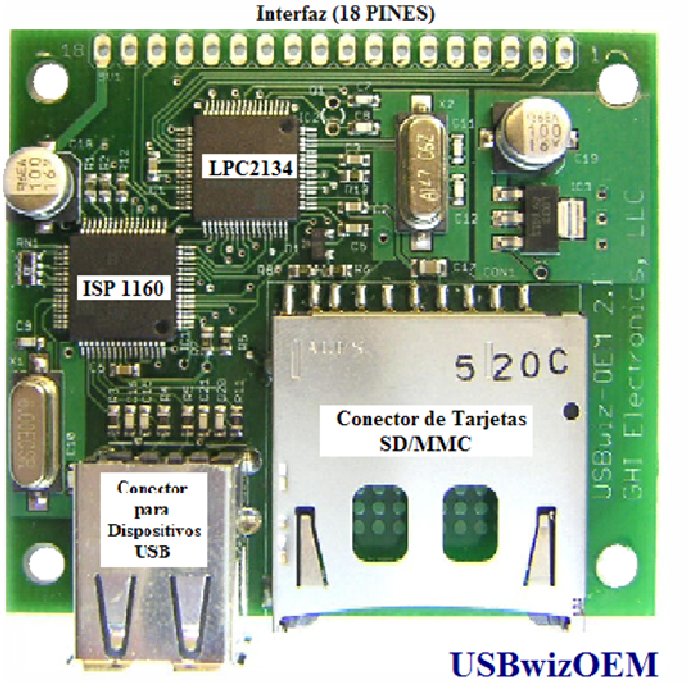


Figura 2.42 "Placa USBwíz-OEM, manufacturada por GUI Electronics"

Tomando en cuenta que el USBwiz-OEM es una placa externa que se instala en el módulo electrónico se deja un conector de 18 pines para la utilización de esta placa. A continuación se detallan las características físicas del USBwiz-OEM.

Características Físicas del USBwiz-OEM

- La placa tiene un tamaño de 56mm x 61mm, esta placa posee una interfaz de 18 conectores que se pueden ser utilizados mediante espadines machos.
- La placa requiere de una alimentación de 5V, (Admite variaciones de voltaje ya que posee un regulador interno).
- Posee dos conectores para dispositivos USB y un conector para tarjetas de memoria SD/MMC. El conector de la tarjeta SD/MMC posee un aditamento que asegura de mejor forma la tarjeta.
- Funciona con un cristal de 32Khz (Cristal incluido en la placa).
- Su consumo de corriente es de 40mA a 50mA, tiene la capacidad de apagarse mediante software.
- El rango de temperatura de operación es de -40℃ a 85℃.

Como se puede ver en la Figura 2.42 la placa USBwiz-OEM consta de dos integrados principales el LCP2134 y el ISP1160. Además posee dos conectores USB hembra TIPO A, un Conector SD/MMC hembra, y una interfaz de 18 pines con la cual el USBwiz-OEM puede ser controlado.

El Microcontrolador LCP2134 que se encuentra en la placa del USBwiz-OEM, cumple con la función de interpretar los comandos enviados por el usuario y controlar las funciones del ISP1160. La programación y el firmware en este microcontrolador son establecidas al momento de su manufacturación y no pueden ser modificadas.

El ISP1160 es un dispositivo de control de puerto USB (USB HOST), que cumple con las especificaciones de USB 2.0, soporta velocidades de transferencia altas de 12 Mbits y bajas velocidades de 1.5 Mbits. Este chip proporciona la posibilidad de albergar dos dispositivos USB, cada puerto tiene una detección automática del dispositivo albergado, además posibilita la creación y manipulación de ficheros que serán utilizados en el proyecto que nos concierne.

En la Figura 2.43 se presenta el diagrama para utilizar el USBwiz-OEM como interfaz de comunicación entre el módulo electrónico y los dispositivos de almacenamiento.

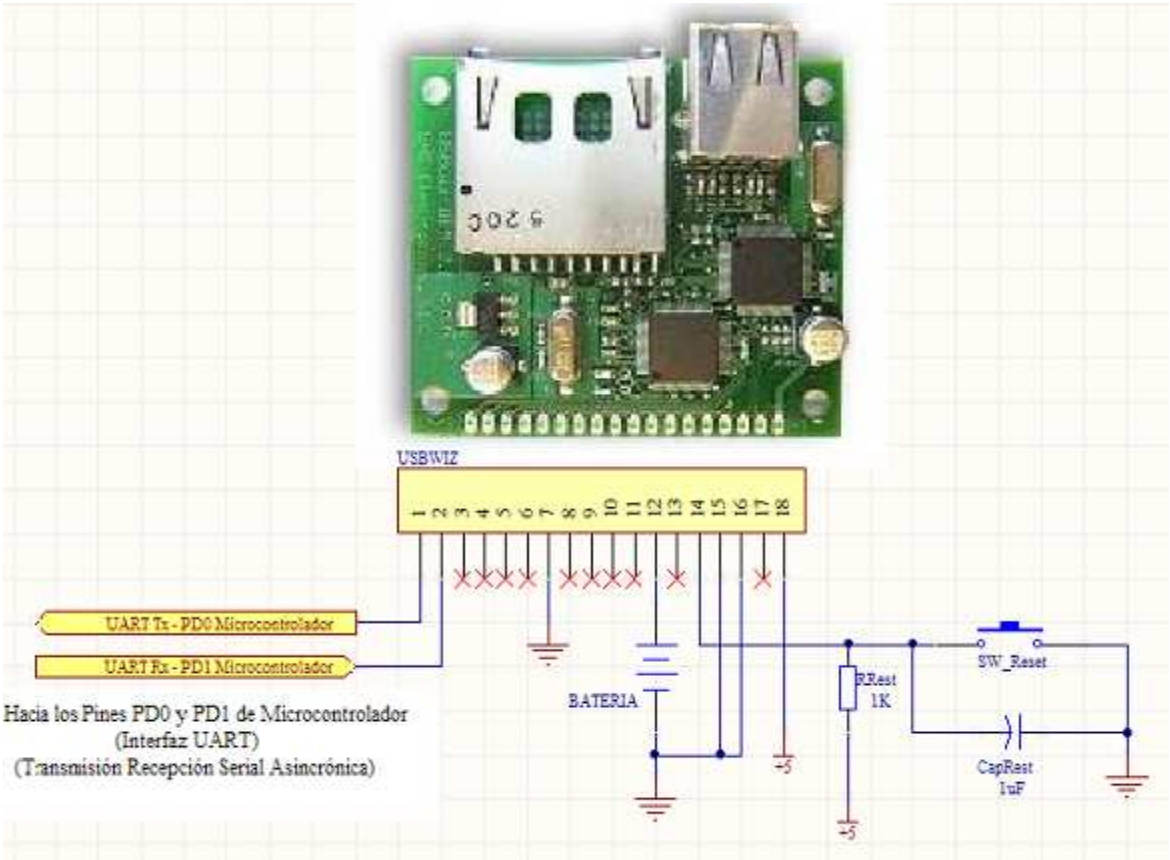


Figura 2.43 "Interfaz Física que comunican el módulo electrónico con los dispositivos de almacenamiento

Si bien la placa del USBwiz-OEM consta de un microcontrolador, el "LCP2134", que controla un Host USB "ISP1160", necesariamente la placa debe tener una interfaz externa ya sea para el usuario a través de un Hyper Terminal o por medio

de un microcontrolador. Para aplicaciones de almacenamiento continuo se opto por la implementación de un microcontrolador que controle el USBwiz-OEM por medio de una interfaz serial UART.

Para que el USBwiz-OEM sea controlado por medio de una interfaz UART debe ser configurada esta interfaz tanto en software como en hardware. En hardware debe establecer el modo de operación al que se desea transmitir y el bus físico por donde se va a comunicar el USBwiz-OEM con el exterior.

Establecimiento de la Interfaz de comunicación para el USBwiz-OEM

El USBwiz utiliza tres interfaces seriales (UART, SPI o I2C), las cuales permiten su control utilizando cualquier microcontrolador que admita las mismas. El USBwiz posee dos pines con los cuales se puede escoger el modo de interfaz que se utilizará. Estos pines son los denominados "MODE0" y "MODE1", y se rigen a la Tabla 2.6 siguiente para escoger la interfaz que se va utilizar.

MODE0	MODE1	Interfaz
0	0	Interfaz Serial UART (Baud Rate= 9600)
1	0	Interfaz Serial SPI
0	1	Interfaz Serial UART(Velocidad Variable)
1	1	Interfaz Serial SPI

Tabla 2.6 "Modos de transmisión del USBwiz-OEM"

Los pines de control determinan la interfaz de comunicación, se encuentran en un inicio en estado alto (1L), por lo que si se desea configurar un 1L no hay que conectar el pin a VCC si no dejarlo sin conexión. Para llevar el pin a una estado bajo "0L" hay que conectar el pin a GND. En el proyecto presente se utilizó como interfaz de comunicación la interfaz serial “UART” en velocidad variable (MODE0=OL: MODE1=1L).

MODE0 (pin16) conectar a GND (MODEO=OL)

MODE1 (pin17) mantener sin conexión (MODE1=1L)

Dado que las interfaces de control SPI y I2C no se utilizan en el proyecto que se está desarrollando, los pines de comunicación utilizados para estas interfaces quedan deshabilitados sin conexión.

Tx_USBwiz (pin 1) hacia UART_RX del Microcontrolador (pin PD0)
Rx_USBwiz (pin 2) hacia UART_TX del Microcontrolador (pin PD1)
Interfaz I2C "SCL" (pin 3) mantener sin conexión
Interfaz I2C "SDA" (pin 4) mantener sin conexión Interfaz
Interfaz SPI "SCK" (pin 5) mantener sin conexión
Interfaz SPI "MISO/RTS" (pin 6) mantener sin conexión
Interfaz SPI "MOSI/CTS" (pin 7) mantener sin conexión
Interfaz SPI "SSEL" (pin 8) mantener sin conexión
MISC (pin 9) mantener sin conexión

Conexión de los Pines de control de fábrica

Cuando existe algún error en software o en firmware internos del USBwiz-OEM debido a defectos de fábrica, existe la posibilidad de reparar dicha configuración, esta opción es exclusiva de la empresa que manufactura estos dispositivos y se realiza mediante el PIN 10 y el PIN 11.

D6/BL (pin11) mantener sin conexión, utilizado solo por la empresa que manufactura.
NC (pin 10) mantener sin conexión (NC)

Conexión de los pines de Alimentación del USBwiz y Conexión del RESET

El pin RESET (pin 14) debe estar en un estado alto para que el USBwiz funcione correctamente. Cuando se quiere reiniciar el USBwiz se debe pasar a un estado bajo, este pin será conectado al RESET principal del módulo electrónico que está conformado por un pulsador, una resistencia, y un capacitor. La resistencia es para limitar el voltaje de la fuente de alimentación y tiene un valor de 1KΩ, el capacitor es utilizado para reducir rebotes y posee un valor de 1uF.

R_Reset = 1KΩ

Cap_Rest =1uF

RESET (pin 14) conexión al principal del módulo electrónico.

El fabricante da la elección de dos formas de alimentación primaria, la primera una alimentación directa aplicando un voltaje de 3.3V al pin de alimentación (pin 13) y la segunda utilizando el regulador interno del USBwiz aplicando un voltaje de 5V al pin +5V (pin 18). Se utilizó para el diseño solo la alimentación primaria que utiliza 5V y se dejo sin conexión la alternativa de alimentación primaria de 3V. Para asegurar la integridad de los datos en el USBwiz se incorporo en el diseño una fuente de alimentación secundaria que consiste en una batería CR2032; esta fuente secundaria se conecta en el pin designado para este propósito (pin 12).

GND (pin 15) conexión a tierra.

VCC (pin 18) conexión a 5V.

VCC2 (pin 13) mantener sin conexión "Fuente primaría 3.3 V".

VBAT (pin 12) conexión a batería del módulo electrónico

"Conexión a batería CR2032".

6.1.3.2.2 Visualización de la Información del Sistema

La función principal de este bloque es presentar información de la actividad del sistema mediante dispositivos visuales. El sistema de Visualización consta de dos circuitos: uno que controlara el LCD y el segundo un grupo de LEDs que indican la actividad de las unidades de almacenamiento.

6.1.3.2.2.1 Diseño del Circuito de interfaz del LCD

El objetivo de diseñar un circuito de operación para el LCD[21] es visualizar de manera instantánea los parámetros de velocidad, distancia y tiempo del automotor. Para esto el software implementado tiene que procesar y actualizar las variables continuamente, de manera que se visualicen en el display del LCD en el menor tiempo posible.

[21] http://www.geocities.com/dinceraydin/lcd/index.html

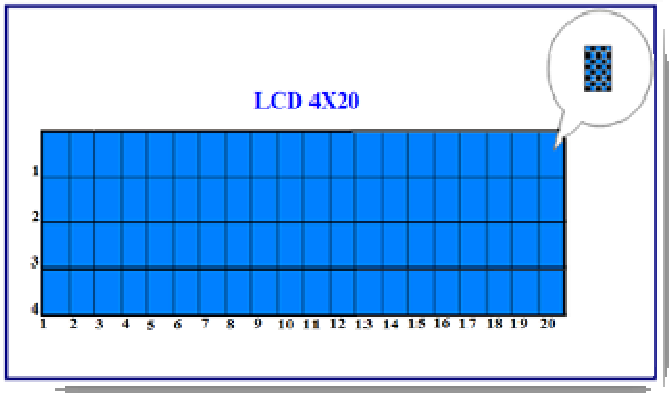


Figura 2.44 "Distribución del display de un LCD"

El LCD escogido para incorporarlo en el módulo electrónico, es un LCD de 4x20, esto significa que el LCD se distribuye en una matriz de 4 filas y 20 columnas, para visualizar los diferentes caracteres enviados por el microcontrolador. Cada celda de la matriz posee un arreglo de 5x8 pixeles como se puede ver en la Figura 2.44.

Para ahorrar espacio en la distribución de los elementos en el módulo electrónico, el LCD será un elemento externo controlado por un conector de 16 pines que proporcionara la interfaz entre el módulo electrónico y el LCD. A continuación, en la Figura 2.45, se presenta la interfaz utilizada por el módulo electrónico para controlar al LCD.

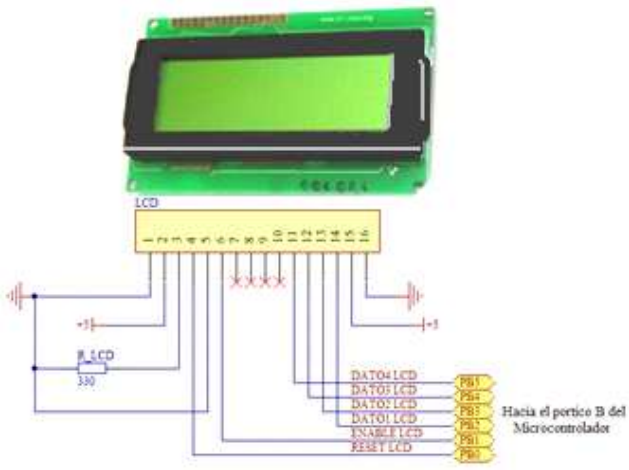


Figura 2.45 "Circuito que estable la interfaz de control del LCD"

Este dispositivo posee un conector de 16 pines, descrito a continuación:

- **GND (pin1) y +5V (pin2):** Por medio de estos pines se provee la alimentación de voltaje a los integrados que posee la placa del LCD (Buffer, Circuitos de Control). La alimentación especificada por el fabricante es de 5 Voltios.

GND (pin 1) Conexión a GND del módulo electrónico
VCC (pin 2) Conexión a 5V del módulo electrónico

- **Limitador de corriente (pin3):** por medio de este pin se limita la luminosidad del LCD. Generalmente se utiliza un potenciómetro pero, para el módulo electrónico que se está desarrollando se coloca una resistencia fija que limite la corriente en el LED del LCD.

R_LCD=330Ω

- **GND_T (pin5):** este pin funciona como referencia a tierra, para la transmisión de datos y debe ser conectado a GND del módulo electrónico.

GND_T (pin 5) Conexión a GND del módulo electrónico.

- **Reset (pin4):** cada vez que se acaba una transmisión de datos por parte del microcontrolador, activa este pin para indicar una nueva transmisión.

- **Enable (pin6):** este se utiliza para habilitar la escritura en el buffer del LCD.

- **Transmisión de datos (pin7 al pin14)**: Estos son los pórticos que sirven para transmitir los datos que se quieren visualizar en el LCD. Para este proyecto solo se utilizarán 4 pines para transmisión.

RESET (pin 4) Conexión al Microcontrolador – PB0.
ENABLE (pin 6) Conexión al Microcontrolador - PB1.
DATO1 (pin 7) Mantener sin conexión.
DATO1 (pin 8) Mantener sin conexión.
DATO1 (pin 9) Mantener sin conexión.
DATO1 (pin 10) Mantener sin conexión.
DATO1 (pin 11) Conexión al Microcontrolador - PB2.
DATO1 (pin 12) Conexión al Microcontrolador - PB3.
DATO1 (pin 13) Conexión al Microcontrolador - PB4.
DATO1 (pin 14) Conexión al Microcontrolador - PB5.

- **GND_Display (pin16) y +5V_Display (pin15):** Por medio de estos pines se alimenta de voltaje el LED que genera la luz del LCD.

GND_Display (pin 16) Conexión a GND del módulo electrónico.
+5V_Display (pin 15) Conexión a VCC=5V del módulo electrónico.

En el LCD se visualizaran los parámetros del automotor como son: Velocidad instantánea, Distancia recorrida, Tiempo de trayecto. El encargado de todo el proceso es el microcontrolador que envía estos datos al LCD.

6.1.3.2.2.2 Diseño del Circuito del estado de las Unidades de Almacenamiento (Visualización por medio LEDs Bicolores)

Este circuito fue diseñado para informar al usuario sobre el estado de las unidades de almacenamiento que se insertan en el módulo electrónico. Su funcionamiento consiste en identificar por medio de LEDs bicolores el estado de cada unidad. El estado de cada LED es controlado por el pórtico C del microcontrolador AVR ATmega16.

Al aplicar una corriente por cualquiera de los cátodos, el LED bicolor emitirá luz ya sea roja si la corriente es aplicada por el PIN1 o verde si la corriente es aplicada por el PIN3. En la Figura 2.46 se muestra la configuración de pines de un LED bicolor comercial.

Figura 2.46 “Distribución de pines de un LED Bicolor"

Este circuito tiene la finalidad de indicar que unidad de almacenamiento está activa, instalada o si ninguna unidad se encuentra insertada en el módulo electrónico. Para este propósito se utiliza tres LEDs bicolor que cambian de color según el estado de la unidad. Los colores utilizados para cada estado de la unidad son los siguientes:

- **LED de Color Rojo**.- Unidad de almacenamiento insertada en el módulo electrónico pero no utilizada para almacenar los datos.

- **LED de Color Verde**.- Unidad de almacenamiento insertada en el módulo electrónico y utilizada para el almacenamiento de los parámetros del automotor.

- **LED Apagado**.- Ninguna unidad de almacenamiento insertada en el slot de la unidad.

En la Figura 2.47 se muestra el diagrama del circuito utilizado para visualizar el estado de los dispositivos de almacenamiento insertados en el módulo electrónico y la interfaz necesaria para que los LEDs bicolores sean controlados por el microcontrolador.

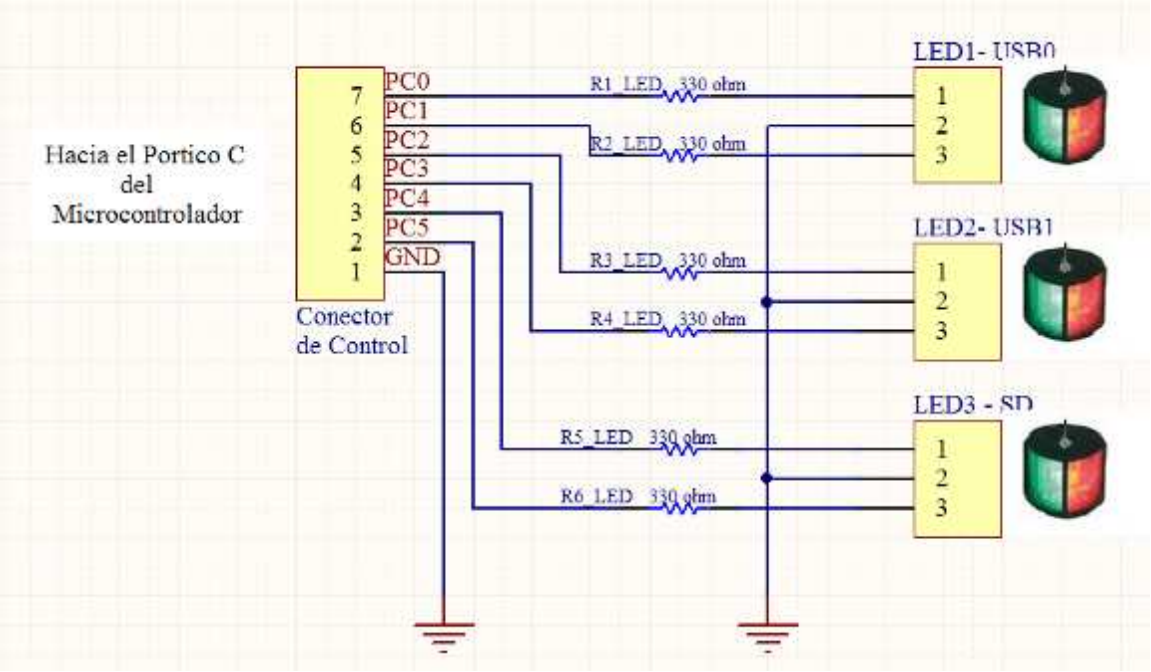


Figura 2.47 "Circuito de Visualización del Estado de los dispositivos de almacenamiento"

Este sistema de LEDs es implementado en una placa secundaria que se instala en el módulo electrónico por medio de un conector de control (Conector de 7 pines). Cada cátodo en los LEDs están conectados a una resistencia de 330 Ω para limitar la corriente de entrada en el LED y así evitar que los LEDs bicolores se quemen. Cada LED es controlado por dos pines del microcontrolador que accionan o cortan su emisión de luz.

Conexión de la Interfaz de control:

PIN1: Conexión a GND del módulo electrónico

PIN2: Conexión a PD5 del microcontrolador

PtN3: Conexión a PD4 del microcontrolador

PIN4: Conexión a PD3 del microcontrolador

PIN5: Conexión a PD2 del microcontrolador

PIN6: Conexión a PD1 del microcontrolador

PIN7: Conexión a PD0 del microcontrolador

Resumen de Elementos:

R1_LED = R2_LED = R3_LED = 330 Ω

R4_LED = R5_LED = R6_LED = 330 Ω

6.1.3.3 Sección de procesamiento de datos

En esta sección se agrupan los diferentes dispositivos que adquieren y procesan la información del módulo electrónico. El principal elemento en esta sección es el microcontrolador que procesa la información mediante el software con el cual fue configurado. Los bloques involucrados en esta sección son: Interfaz de configuración del Usuario, Circuito de Adquisición y Procesamiento de Información.

6.1.3.3.1 Diseño del circuito de configuración del Usuario

Este circuito establece una interfaz de comunicación entre el microcontrolador y el usuario, posibilitando que este último configure y habilite acciones que se desarrollaran en el microcontrolador. Las acciones que el usuario puede habilitar son: inicio de almacenamiento de datos, finalización de almacenamiento de datos, desinstalar unidad de almacenamiento actual, o para igualar el reloj del sistema.

En la Figura 2.48 se muestra el esquema del circuito de configuración del Usuario y su interfaz necesaria para comunicarse con el microcontrolador.

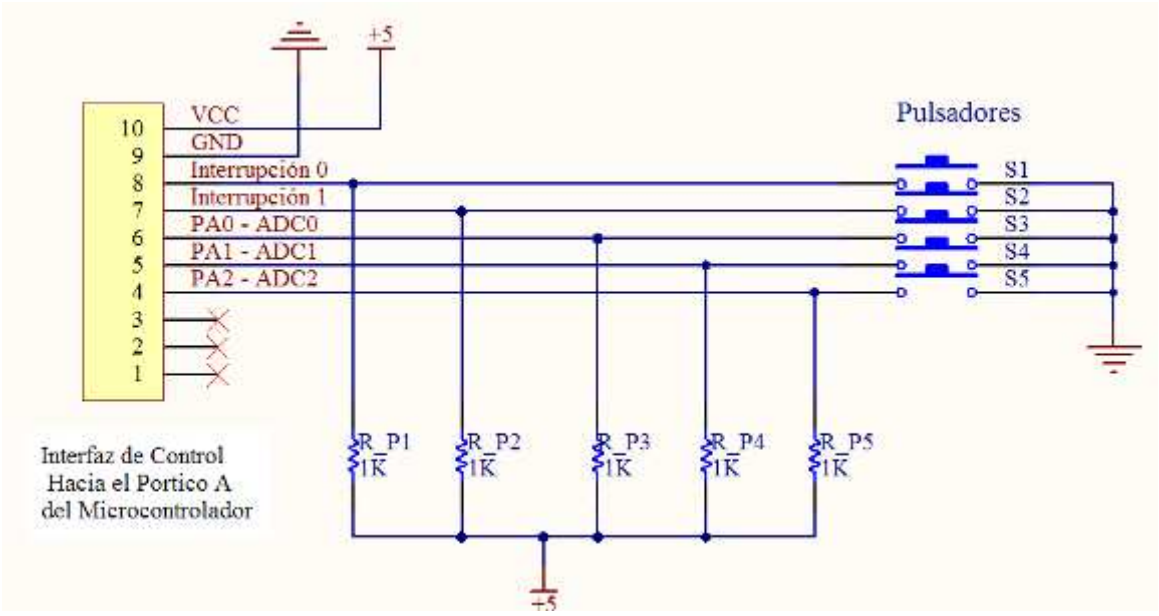


Figura 2.48"Circuito de configuración del Usuario"

Para poder entender el funcionamiento del panel de pulsadores incorporados en el circuito de configuración del Usuario se presenta la Figura 2.49 que ilustra la acción del pulsador cuando está en circuito abierto o en circuito cerrado. Cuando el pulsador se encuentra en circuito abierto, la conexión establecida se reduce a VCC y la resistencia incluida en el circuito, dando una salida de voltaje igual a VCC=5V. Por otra parte, cuando el pulsador se encuentra en circuito cerrado el terminal queda conectado a GND obteniéndose así una salida de voltaje de 0V.

Figura 2.49 "Funcionamiento de los pulsadores"

El Circuito de configuración del Usuario consta de un panel de cinco pulsadores que manejan rutinas individualmente incorporadas en el software del microcontrolador. Cada entrada de este circuito en un inicio posee un estado de alto (voltaje 5V), al presionar el pulsante las entradas de este circuito pasan a estado bajo (voltaje 0V). Para el correcto funcionamiento del circuito se utiliza resistencias de 1KΩ necesarias para que exista la transición de VCC a GND al aplastar el pulsante.

Puesto que este circuito va a ser elaborado en una placa secundaria, se estable una interfaz de comunicación con el microcontrolador, mediante un conector de 10 pines. El microcontrolador será accionado al distinguir un cambio de nivel lógico alto a nivel lógico bajo. Tres de las cinco entradas del microcontrolador usan ADC para su identificación, las otras dos entradas controlan dos interrupciones externas del microcontrolador.

Conexión de la Interfaz de control:

PIN1: Mantener sin conexión (pin de reserva)

PIN2: Mantener sin conexión (pin de reserva)

PIN3: Mantener sin conexión (pin de reserva)

PIN4: Conexión a PA2 del microcontrolador (Conversor Analógico/Digital ADC2)

PIN5: Conexión a PA1 del microcontrolador (Conversor Analógico/Digital ADC1)

PIN6: Conexión a PA0 del microcontrolador (Conversor Analógico/Digital ADC0)

PIN7: Conexión a PD3 del microcontrolador (Interrupción externa INT0)

PIN8: Conexión a PD2 del microcontrolador (Interrupción externa ÍNT0)

PIN10: Conexión a VCC del módulo Electrónico

PIN9: Conexión a GND del módulo Electrónico

Resumen de Elementos:

R_P1 = R_P2 = R_P3 = 1KΩ

R_P4 = R_P5 = 1KΩ

6.1.3.3.2 Circuito de Procesamiento de la Información

Si bien el proceso de procesamiento de la información se realiza mediante software se debe establecer un dispositivo físico que sea capaz de albergarlo. Con este fin se incorporó en el módulo electrónico un microcontrolador AVR ATmega16 de características apropiadas para su utilización.

Este microcontrolador es basado en la arquitectura RISC de alta eficiencia, que le permite mejorar el rendimiento de ejecución, poseer una eficiencia de 1 MIPS por Hertz, optimizar su consumo de energía y mejorar la velocidad de procesamiento en comparación de muchos microcontroladores existentes.

Utilización de los Puertos de Entrada y Salida

Todos los pórticos de un microcontrolador AVR tienen la capacidad de funcionar como puertos digitales I/O. Cada pórtico de I/O puede ser designado individualmente como entrada o como salida. Esta característica en particular, hace que se puedan controlar varios dispositivos periféricos individualmente.

Para el módulo electrónico, el microcontrolador controlará dos dispositivos visuales: el LCD y los LEDs bicolor. Para el control de estos dispositivos periféricos en el microcontrolador se configuran los puertos PBO...PB7 y PCO...PC5 como salidas. Una representación gráfica se la ilustra en la Figura 2.50

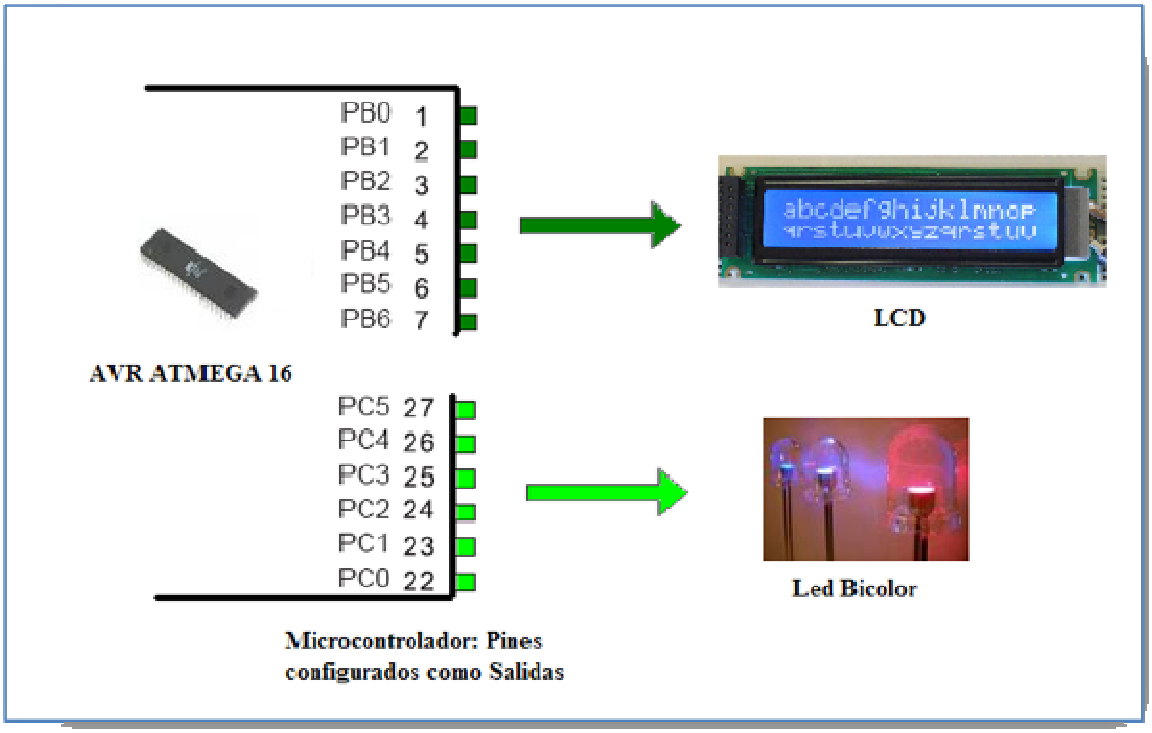


Figura 2.50 "Puertos configurados como salidas, para controlar dispositivos periféricos"

Utilización de la Interfaz de comunicación USART [19]

La interfaz asincrónica UART es utilizada por el microcontrolador para establecer una comunicación con el USBwiz-OEM. Ya que el termino asincrónico significa sin reloj, ambos dispositivos deberán tener bases de tiempo iguales. Típicamente, los datos envidos o recibidos son 7 u 8 bit pero el conjunto de bits transmitidos es superior, pues se incluye un bit de arranque (que indica al receptor que a continuación vendrá una serie de bits que corresponden al dato), un bit de paridad (opcional para el control de errores) y uno o dos bits de parada (que indican que ha finalizado la transmisión). El esquema de los datos utilizados en la transmisión serial asincrónica se muestra en la Figura 2.51.

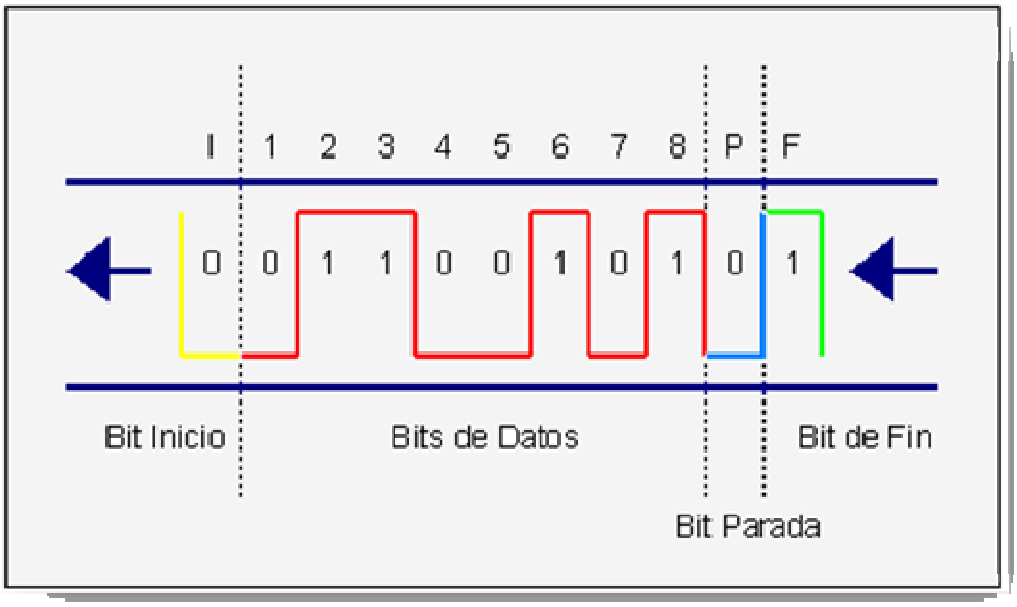


Figura 2.51 "Esquema de transmisión asincrónica"

La transmisión se puede realizar en dos sentidos, en la Figura 2.52 se pueden distinguir los pines involucrados en la interfaz de comunicación UART entre el USBwiz y el microcontrolador. El pin de transmisión UART (pin 15) del microcontrolador debe estar conectado al pin de recepción del USBwiz (pin 1). El pin de recepción del microcontrolador (pin 14) debe estar conectado pin de transmisión del USBwiz (pin 2).

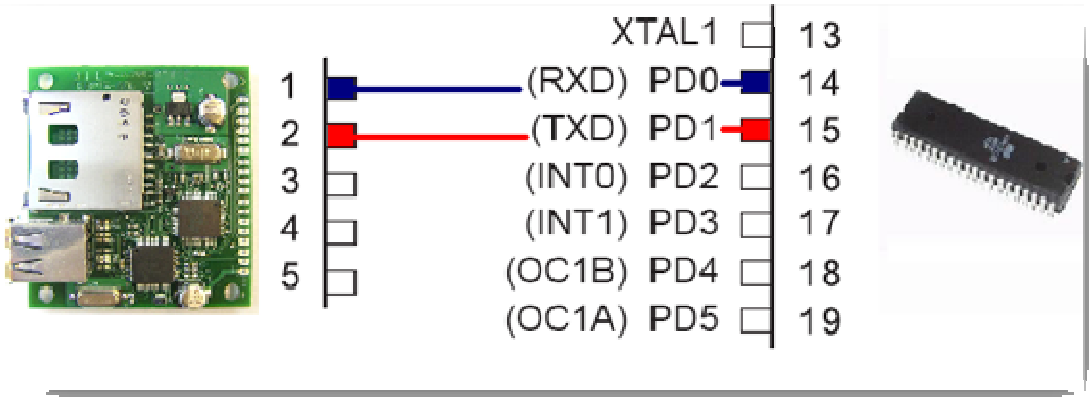


Figura 2.52 "Conexión Serial entre el Microcontrolador y el USBwiz-OEM"

Utilización de la Interfaz de comunicación I²C [20]

La principal característica de I²C[22] es que utiliza dos líneas para transmitir la información: una para los datos y por otra la señal de reloj. También es necesaria una tercera línea, pero esta sólo es la referencia (masa). Como varios circuitos están comunicados al módulo electrónico estos comparten una misma masa (conexión a tierra – GND) esta tercera línea no suele ser necesaria.

Los dispositivos conectados al bus I²C tienen una dirección única para cada uno. También pueden ser maestros o esclavos. El dispositivo maestro (microcontrolador AVR ATmega16) inicia la transferencia de datos y además genera la señal de reloj, pero no es necesario que el maestro sea siempre el mismo dispositivo, esta característica se la pueden ir pasando los dispositivos que tengan esa capacidad. Esta característica hace que al bus I²C se le denomine bus multimaestro.[23]

La interfaz I²C incluye una línea de reloj, línea datos entrante (SCL), línea de datos saliente (SDA) las cuales pueden albergar a varios periféricos. Para la conexión master/slave que se implementará en el sistema, solo se necesita los

[22] **I²C** es un bus de comunicaciones serie. Su nombre viene de *Inter-Integrated Circuit* (Circuitos Inter-Integrados).

[23] http://es.wikipedia.org/wiki/I%C2%B2C

buses SDA y SCL. La Figura 2.53 muestra la conexión que se emplea para que el microcontrolador se comunique con el reloj externo (DS1307).

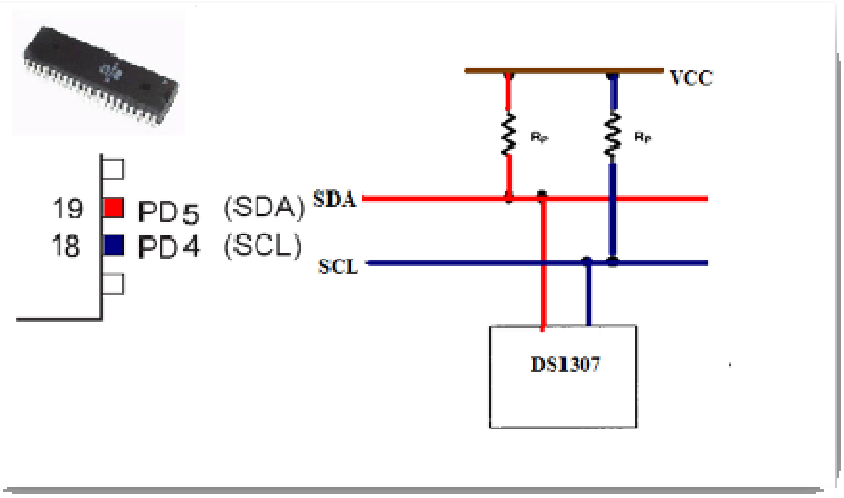


Figura 2.53 "Bus I²C entre el microcontrolador y el DS1307"

Utilización del Conversor Analógico-Digital

El ATmega16 cuenta con un conversón analógico/digital de 10 bits de resolución. El ADC tiene 8 canales para su uso; es decir, cualquier pin del pórtico A (PortA) puede ser utilizado como un ADC individual. Admite voltajes analógicos de 0V a VREF.

El ADC del microcontrolador ATmega16 tiene las siguientes características:

- Resolución de 10 bits.
- Tiempo de conversión de 13 µs a 260 µs.
- 8 canales para ser utilizados individualmente.
- 7 canales de entrada diferencial.
- Canales de entrada diferencial con opción de ganancia de 10X y 20X.
- Resolución ajustable del ADC.
- Voltaje de referencia interna del ADC (2.56V).
- Modo de conversión simple.
- Interrupción para realizar una conversión con el ADC.

El pórtico A del microcontrolador Atmega16 se utiliza en el control de los pulsadores de la interfaz de comunicación con el usuario, cuando un pulsador cambie su estado de alto a bajo, el ADC hará esa distinción de voltajes.

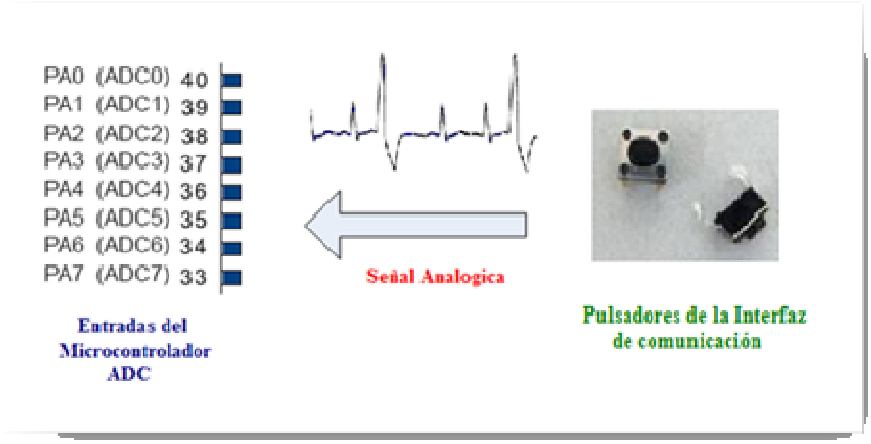


Figura 2.54 "Entradas ADC (Conversor Analógico/Digital) del Microcontrolador"

En la Figura 2.55 se muestra el diagrama de conexión del Circuito de Procesamiento de la Información.

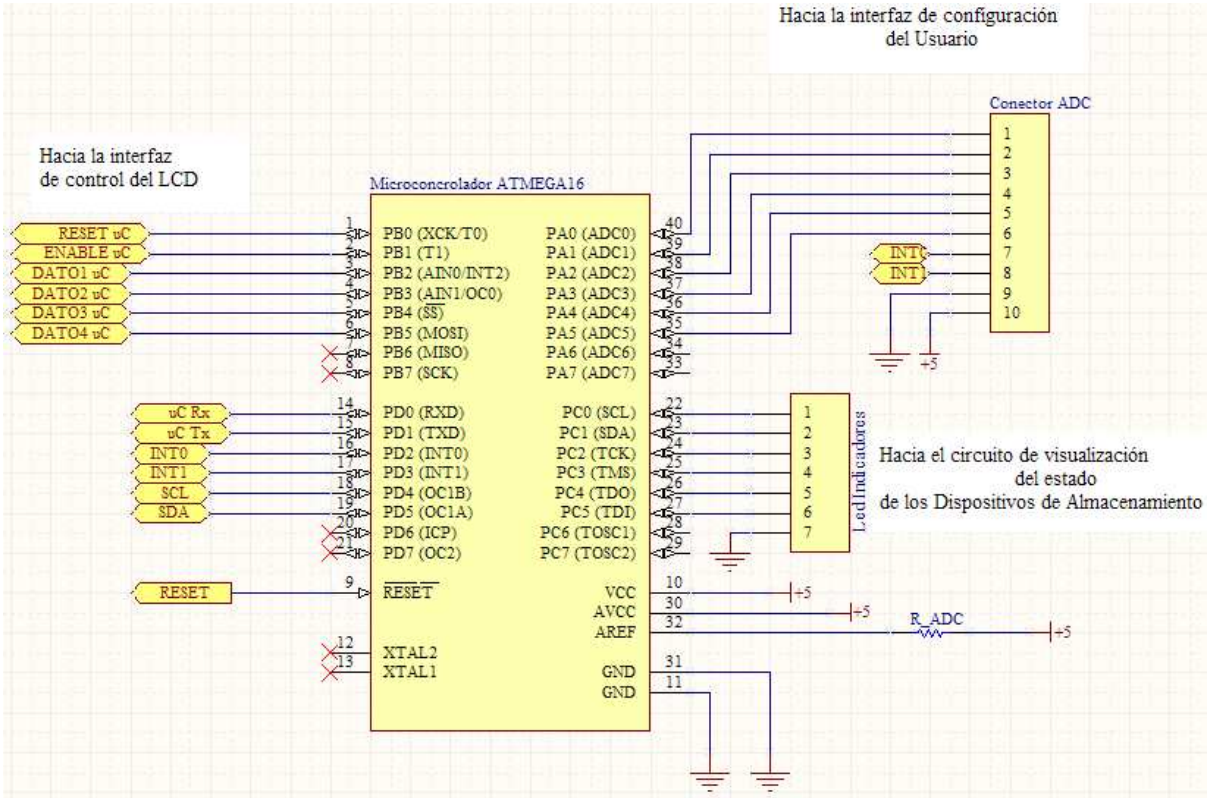


Figura 2.55 "Diagrama del Circuito de Procesamiento de la Información"

6.1.3.4 DISEÑO DEL CIRCUITO DE ALIMENTACIÓN DEL SISTEMA

Como fuente del sistema se aprovecha la batería que poseen todos los automotores. Esta fuente es generalmente de 12 Voltios y de 2 a 5 amperios hora de salida.

La mayoría de dispositivos que se utilizan (microcontrolador, LCD, USBwiz, entre otros) necesitan un fuente de alimentación de 5 Voltios. Para poder obtener los 5 voltios que necesitan estos dispositivos, se utiliza el integrado LM7805, que es un regulador de voltaje (+5V).

6.1.3.4.1 Regulador LM7805

El LM7805 es un regulador de tensión de tres terminales con una tensión de salida de 5 voltios. El encapsulamiento utilizado es usualmente el TO220, encapsulamiento que posee un reverso metálico y un agujero utilizado para instalar este tipo de dispositivos en un disipador.

Parámetros	Símbolo	Condiciones		LM7805 Min.	LM7805 Tip.	LM7805 Max.	Unidad
Voltaje de Salida	Vo	TJ =+25 ºC		4.8	5.0	5.2	V
		5.0mA ≤ Io ≤ 1.0A; Po ≤ 15W ;Vin = 7V a 20V		4.75	5.0	5.25	
Regulación de línea (Variación por Temperatura)	Regline	TJ=+25 ºC	Vo = 7V a 25V	-	4.0	100	mV
			Vin = 8V a 12V	-	1.6	50	
Regulación de Carga (Variación por Temperatura)	Regload	TJ=+25 ºC	Io = 5.0mA a 1.5A	-	9	100	mV
			Io =250mA a 750mA	-	4	50	
Corriente de Reposo	Iq	TJ =+25 ºC		-	5.0	8.0	mA
Drift del Voltaje del Salida	ΔVO/ΔT	Io=5mA		-	-0.8	-	mV/ ºC
Ruido del Voltaje de Salida	Vin	f = 10Hz to 100KHz, TA=+25 ºC		-	42	-	υV/Vo
Corriente de cortocircuito	Isc	VI = 35V, TA =+25 ºC		-	230	-	mA
Corriente máxima	Ipk	TJ =+25 ºC		-	2.2	-	A

Tabla 2.7 "Parámetros de Salida del LM7805"

Parámetros	Símbolo	Valor	Unidad
Entrada de Voltaje	Vin	35	V
	Vo	5 a 18	
	Vin	40	V
	Vo	24	
Rango de operación de Temperatura	TOPR	0 ~ +125	℃
Rango de Temperatura Almacenada	TSTG	-65 ~ +150	℃

Tabla 2.8 "Parámetros de entrada del LM7805"

De los parámetros en las Tabla 2.7 y Tabla 2.8 se puede concluir que el LM7805 es suficiente para la aplicación presente. En vez de utilizar alguna variante se decidió utilizar el circuito propuesto en el data sheet del LM7805. Más datos sobre el LM7805 constan en el ANEXO B-3.

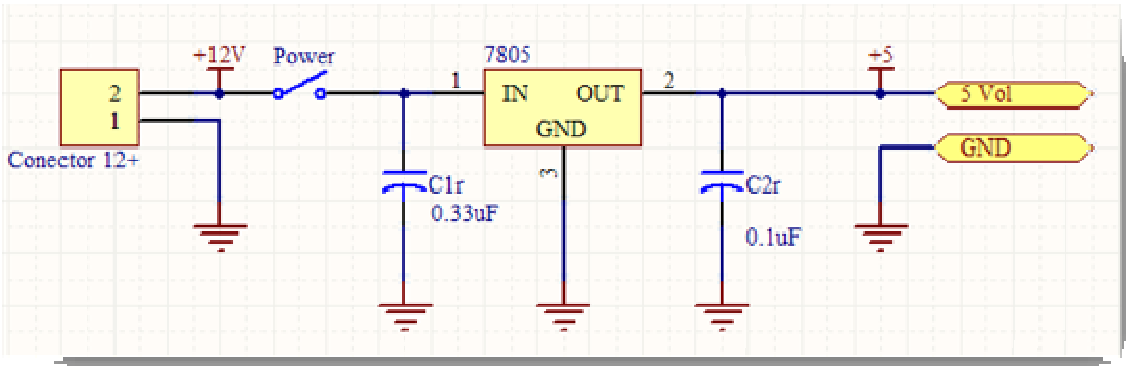


Figura 2.56 "Circuito de Regulación de voltaje +5V"

El circuito que se muestra en la Figura 2.56 consta de una conexión directa a la batería del automotor (Conector 12+), un interruptor para cortar la alimentación al sistema (Power) y el circuito que regula el voltaje a 5 voltios. Los capacitores añadidos al circuito de regulación fueron considerados para la eliminación del ruido.

Resumen de Elementos:

C1r = 0.33uF

C2r = 0,1uF

El diagrama del módulo electrónico concebido en su totalidad se muestra en la Figura 2.57. El diagrama del circuito para "establecer una interfaz con el usuario" y el circuito de "visualización del estado de las unidades de almacenamiento" se muestran en la Figura 2.58.

Módulo Electrónico

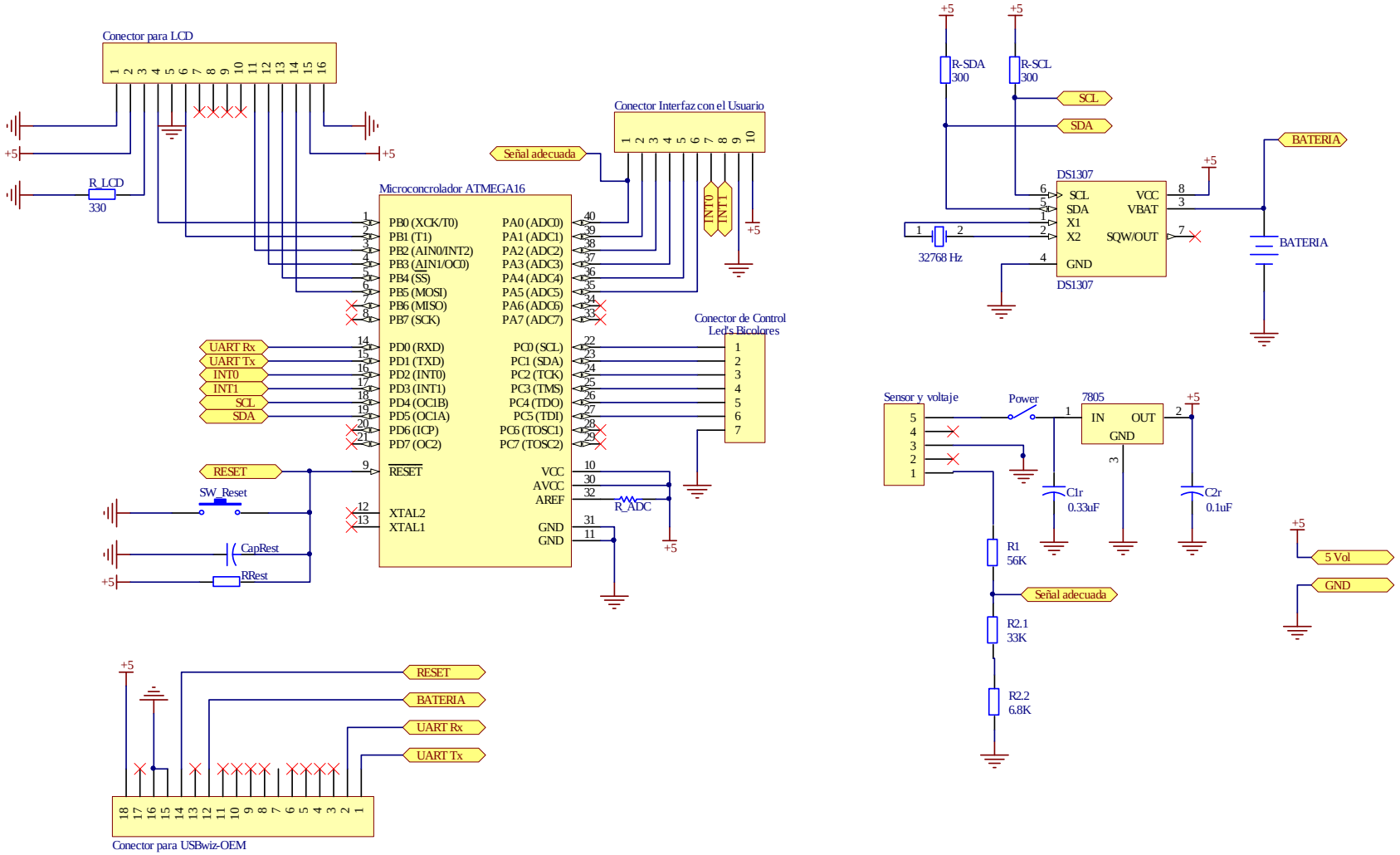
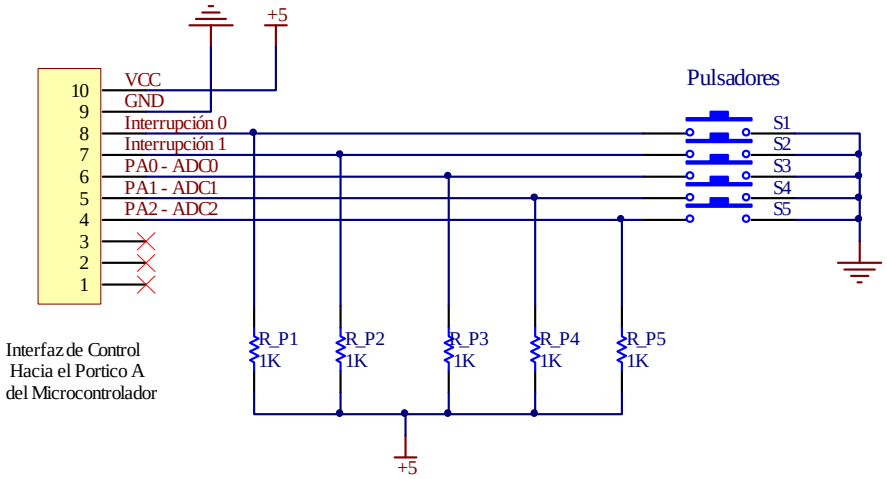


Figura 2.57 “Esquema del Módulo Electrónico”

Placas Montadas en el Módulo Electrócio

Esquema del circuito de configuración del Usuario



Esquema del circuito para visualizar el estado de las unidades

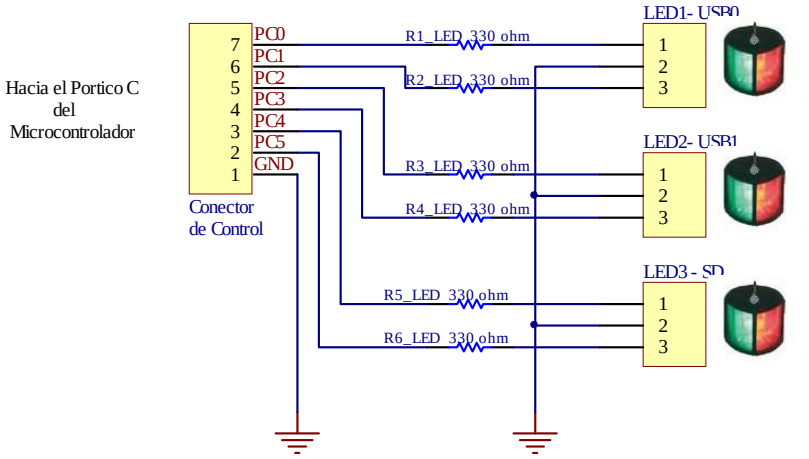


Figura 2.58 “Esquema de las placas externas montadas en el módulo electrónico”

6.1.4 REALIZACIÓN DEL CIRCUITO IMPRESO – PCB [21]

Actualmente los circuitos impresos son un medio seguro para reforzar cualquier sistema, sin el temor de que exista desconexiones por cables sueltos, perdida de elementos, cortocircuitos, como suele suceder en los proto board. Por los múltiples métodos que existen y los diferentes materiales puestos en el mercado, los circuitos impresos han ido bajando su precio, posibilitando su uso generalizado.

A pesar de ser un método muy dificultoso para aplicarlo en cualquier sistema electrónico, es un método robusto y de una fiabilidad elevada. Este método es aconsejable cuando se requiere hacer placas en serie. Por todas estas cualidades el módulo electrónico propuesto será elaborado en una placa impresa.

Para la realización del módulo electrónico se trabajó con el programa computacional PROTEL DXP. Este programa proporciona una interfaz que posibilita realizar circuitos ruteados.

Una vez establecidos los diferentes sistemas electrónicos involucrados en el sistema, se procedió a interconectarlos entre si, como se muestra en la Figura 2.59.

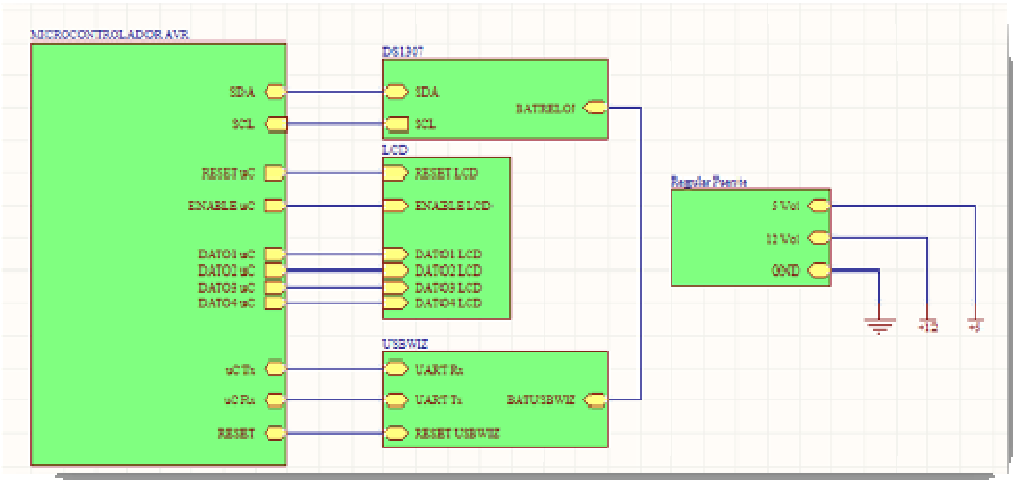


Figura 2.59 "Interconexión de Sistemas electrónicos en PROTEL DXP"

Los diferentes circuitos impresos realizados en PROTEL DXP se muestran en el ANEXO A-1.

Una utilidad del programa PROTEL DXP, es la visualización de las placas diseñadas en un formato 3D. Las siguientes ilustraciones muestran un diseño de las placas terminadas.

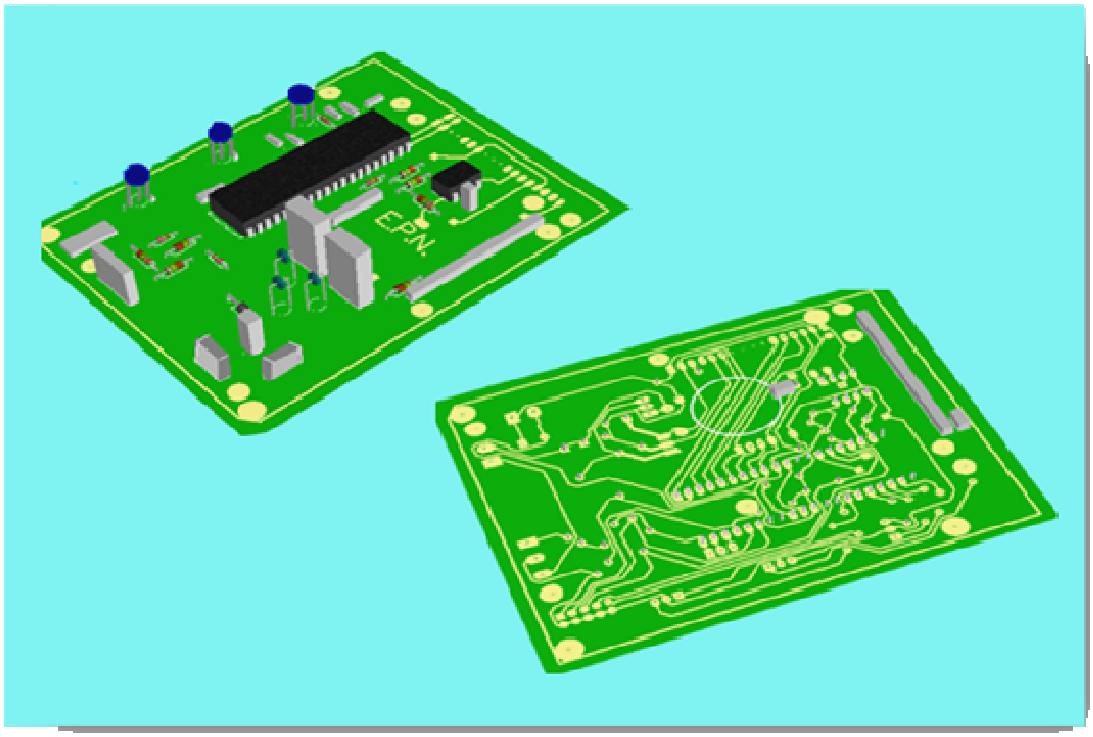


Figura 2.60 "Vista de la Placa Principal"

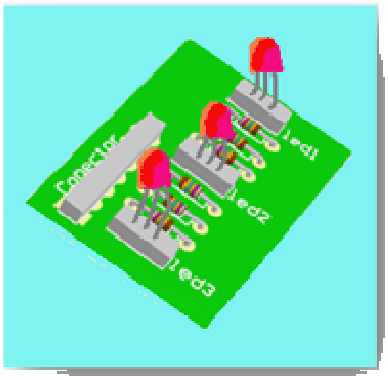


Figura 2.61 "Vista de la Placa de los LED Bicolores"

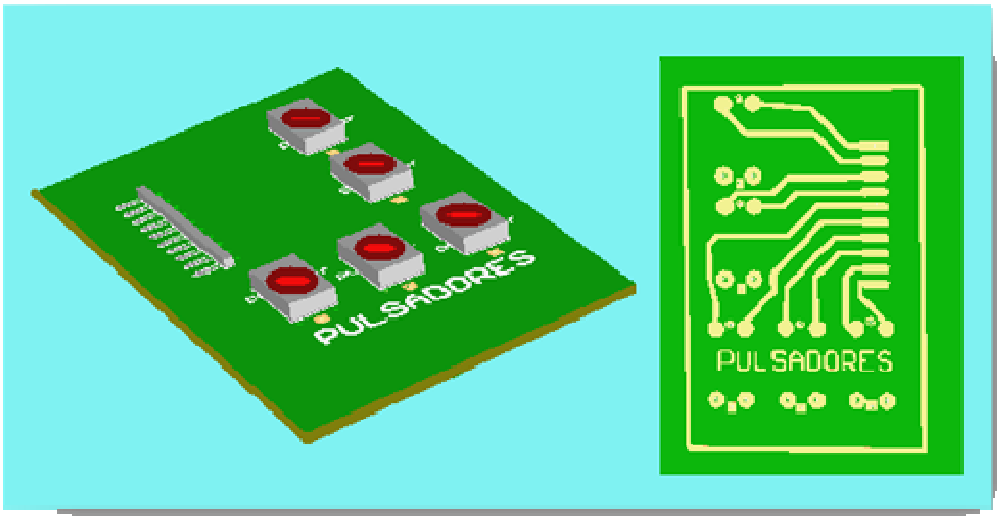


Figura 2.62 "Vista de la Placa de interfaz del Usuario"

Para tener una referencia del funcionamiento del programa PROTEL DXP se pone a consideración el ANEXO A-2, que recopila varias instrucciones con las cuales se puede elaborar un circuito ruteado.

A continuación se hablará sobre el desarrollo del software de control y soporte del presente sistema.

6.2 DISEÑO DEL SOFTWARE DEL SISTEMA

6.2.1 INTRODUCCIÓN

En el presente capitulo se describe el desarrollo del firmware del microcontrolador. El programa en el microcontrolador debe facilitar al usuario poder almacenar en la memoria flash los diferentes parámetros obtenidos del automotor. El software del microcontrolador también debe ser capaz de comunicarse con el USBwiz-OEM. Esta comunicación servirá para enviar

comandos al microcontrolador LCP2134 del USBwiz-OEM, para controlar los diferentes dispositivos de almacenamiento.

Otra función que debe realizar el software del microcontrolador es proveer dos interfaces con el usuario, la primera para que un grupo de pulsadores pueda habilitar la escritura en los dispositivos de almacenamiento e igualar la hora/fecha en el microcontrolador. La segunda una interfaz visual que presenta los datos instantáneos del automotor y la actividad de las distintas memorias insertadas en el sistema.

6.2.2 DESARROLLO DEL PROGRAMA PARA EL MICROCONTROLADOR

Para cumplir con el objetivo planteado el programa del microcontrolador debe realizar las siguientes tareas: primero el microcontrolador, por medio del uso de contares determinar la frecuencia de la señal que genera el sensor de velocidad del automóvil y posteriormente digitalizarla. Una vez digitalizada la señal los datos obtenidos deben ser procesados para presentarlos de manera comprensible para el usuario. Al llevar a cabo el proceso de adecuamiento de la señal, el microcontrolador debe establecer una interfaz de comunicación con el USBwiz. La interfaz de comunicación servirá para acceder a las memorias flash insertadas en el sistema.

Un proceso importante que debe realizar el programa del microcontrolador es proveer una interfaz amigable con el usuario, dicha interfaz debe ser capaz de visualizar parámetros instantáneos del automotor, así como el estado de las memorias insertadas y la posibilidad de controlar procesos internos del programa por medio de pulsadores.

El siguiente diagrama de flujo relaciona las funciones que el microcontrolador realizará.

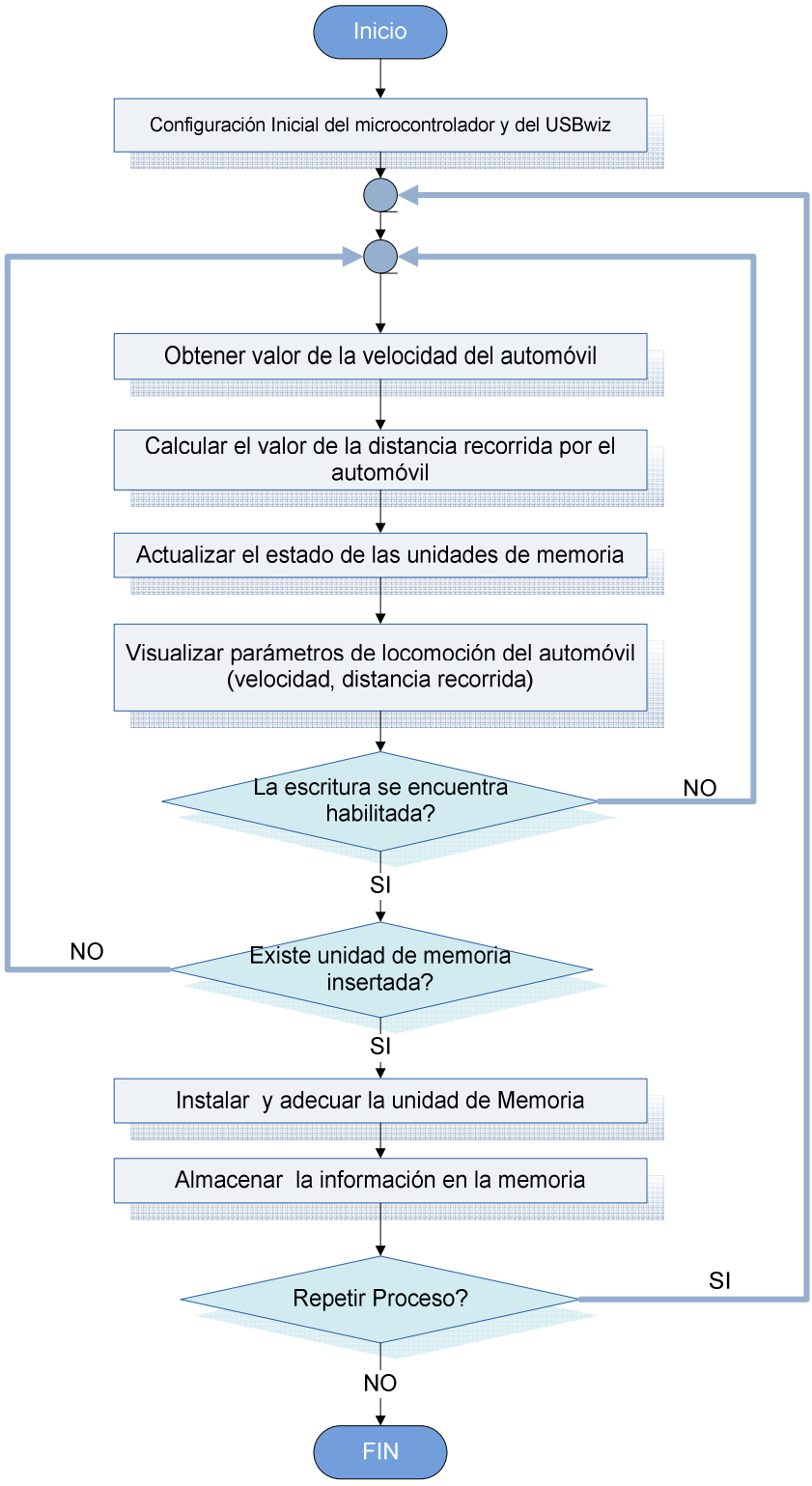


Figura 2.63 “Diagrama de Flujo del programa del microcontrolador”

A continuación se explicará la forma estructural de cada bloque que forma el algoritmo y se detalla cada tarea en lenguaje estructurado.

6.2.2.1 Configuración inicial

En este bloque se configura las variables necesarias para el funcionamiento del programa, además de establecer parámetros que permitan al microcontrolador comunicarse con los diferentes periféricos que se encuentran en el sistema.

La configuración inicial es realizada por dos rutinas, la configuración inicial del microcontrolador y la configuración inicial del USBwiz, descriptos a continuación:

6.2.2.1.1 Configuración inicial del microcontrolador

En esta tarea se configuraran todos los parámetros de inicialización del microcontrolador, esta sección es la que cimienta todas las operaciones del microcontrolador como las interfaces de comunicación, variables de almacenamiento, y declaración de subrutinas.

> ***Rutina de "Inicialización del Microcontrolador"***
> *Identificar el microcontrolador y el cristal que se utiliza.*
> *Configurar parámetros para la interfaz UART (Comunicación Serial).*
> *Configurar parámetros para controlar LCD*
> *Habilitar Conversor ADC, interrupción serial, interrupciones externas y timer.*
> *Establecer variables auxiliares de la Rutina*
> *Declaración de subrutinas.*
> *Establecer Valor inicial de las variables.*
> ***Fin tarea***

6.2.2.1.2 Configuración inicial del USBwiz

Para que el microcontrolador pueda almacenar los parámetros del automotor debe poseer una interfaz con la que controle el USBwiz. Esta tarea establece dicha interfaz, configurando un protocolo adecuado para su comunicación.

Rutina de "Inicialización del USBwiz"
Reiniciar USBwiz.
Configurar parámetros para la interfaz UART (Velocidad de Transmisión).
Configurar reloj/calendario del USBwiz.
Habilitar eventos.
Fin de Tarea

6.2.2.2 Obtener valor de la velocidad del automóvil

Esta rutina mide el valor de la frecuencia en la señal generada por el velocímetro y la relaciona con una frecuencia base para obtener los valores de velocidad en el automotor. El proceso establecido para medir la frecuencia de la señal del velocímetro, implementa un contador que mide la longitud del pulso en cada lectura. El contador utilizado tiene un tope de 65535 que equivale a 655.35mseg,

Para tener una mejor referencia de la utilización del contador en la obtención de la velocidad del automotor se presentara a continuación un ejemplo.

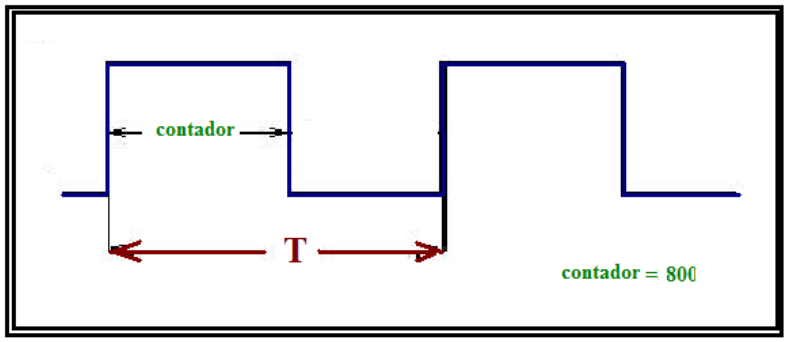


Figura 2.64 "Señal generada por el Velocímetro (medición de frecuencia)"

$$contador = 800$$

$$\Delta t_{contador} = 8\ mseg$$

$$T = 16\ mseg$$

$$f = \frac{1}{T} = \frac{1}{16\ mseg} = 62.5Hz$$

Se relaciona la frecuencia obtenida ($f = 62.5Hz$) con la frecuencia base.

$$120\ Hz\ equivalente\ a\ 180\ {}^{Km}/_{h}$$

$$V = \frac{180\ {}^{Km}/_{h} \times 62.5Hz}{120\ Hz} = 93.75\ {}^{Km}/_{h}$$

$$V = 93.75\ {}^{Km}/_{h}$$

Rutina "Obtener velocidad"

- *Iniciar el contador (Transición de bajo a alto).*
- *Parar el contador (Transición de alto a bajo), almacenar el contador en Val_Contador.*
- *Obtención de la frecuencia de la señal (frecuencia=0.5/Val_Contador)*
- *Obtención de la Velocidad del automotor (velocidad=1.5xfrecuencia*

Fin de la rutina

6.2.2.3 Calcular el valor de la distancia recorrida por el automóvil

Esta rutina procesa los valores de velocidad y tiempo para obtener la distancia recorrida por el vehículo. La distancia recorrida es calculada por medio de la siguiente ecuación.

$$r_F = r_o + (\Delta t \times V_m)$$

$$r_F = r_o + \left(\Delta t \times \frac{V_F + V_o}{2}\right)$$

Ecuación 2.1 "Calculo del desplazamiento realizado"

En primera instancia la rutina examina si es la primera lectura realizada, de ser así configura los parámetros iníciales (Vo = 0, Ro = 0), caso contrario sigue calculando la distancia recorrida según la ecuación planteada.

Rutina "Procesamiento de datos"

Examinar si se realiza la primera lectura?

SI:

- *Actualizar valores iniciálales de velocidad y distancia (Vo = 0, Ro = 0)*

N0:

- *Adecuar el valor de velocidad del automóvil, multiplicar por un factor dependiendo de la calibración realizada Vf = Vf * Cte*
- *Llamar a subrutina "tiempo" (obtención del tiempo mediante timer)*
- *Calcular distancia recorrida del móvil en cada lectura Rf = tiempo * (Vf + Vo) * 0.5 + Ro*
- *Actualizar valores de velocidad y distancia (Vo = Vf, Ro = Rf)*

Fin de la rutina

6.2.2.4 Actualizar el estado de las unidades de memoria

El programa del microcontrolador debe ser capaz de actualizar el estado de las unidades de almacenamiento de forma automático. Para esto el microcontrolador interpreta los comandos emitidos por el USBwiz, actualiza las memorias insertadas. Este bloque es desarrollado por dos rutinas descritas a continuación:

6.2.2.4.1 Chequear el estado de las unidades de memoria

Esta rutina es utilizada para chequear el estado de las memorias ya sea que fueron removidas, insertadas, instaladas o desinstaladas. La rutina está ligada intrínsecamente a la rutina "Control de inserción de las unidades de memoria", la misma que provee la información para actualizar el estado de cada slot que posee el USBwiz.

Cuando esta rutina actúa por primera vez, examina si existe alguna unidad insertada. De ser así, actualiza la variable "VarFM" con el valor de la unidad

insertada. En el caso de no existir ninguna unidad insertada actualiza el valor de la variable “VarFM” con un valor nulo (Sin ninguna memoria insertada). El programa del microcontrolador utiliza esta rutina para verificar si se realizó algún cambio en el estado de la unidad.

Rutina “Chequear el estado de las unidades de memoria”

Examinar si existe alguna unidad instalada?

SI:

Verificar si la Unidad Actual = Unidad Instalada?

SI:

- *Mantener Valor de Unidad Instalada*

N0:

- *Cerrar Archivo creado de la unidad instalada actual*
- *Actualizar la unidad instalada con la unidad actual (VarFM=Var2 o VarFM=Var3 o VarFM=Var4)*

N0:

Examinar si existe alguna unidad insertada?

SI:

- *Actualizar la variable VarFM con el valor de la unidad actual (VarFM=Var2 o VarFM=Var3 o VarFM=Var4)*

N0:

- *Actualizar el valor de la variable VarFM=” “ a valor nulo (Sin ninguna unidad insertada)*

Fin de Rutina

6.2.2.4.2 Control de inserción de las unidades de memoria

Cuando una memoria o tarjeta flash es insertada en el slot del USBwiz, este envía un comando por medio de la interfaz UART. El comando o evento es procesado por el microcontrolador para actualizar las variables Var2, Var3, Var4 que se usan para identificar si alguna memoria fue insertada o removida.

Rutina, "Control de inserción de las unidades de memoria"

Atender a la interrupción serial

Actualizar Memorias insertadas.

Registrar Valores de Actualización en variables: Var2 (USB0), Var3 (USB1), Var4 (SD/MMC).

Fin de Rutina

6.2.2.4.3 Control de LED bicolores

Esta rutina actúa de manera similar a la rutina "Chequear el estado de las unidades de memoria", pero es más orientada a la interfaz visual. La rutina define los puertos del pórtico C como salidas, para que el microcontrolador pueda accionar los LEDs bicolor y así el usuario llegue a distinguir el estado de cada slot del USBwiz.

Rutina "Control de LED Bicolores"

Examinar si existe alguna unidad insertada en el sistema?

SI:

Examinar cual es el estado de la memoria?

- *Memoria Instalada:*
  Activar los LEDs en color verde
- *Memoria Insertada:*
  Activar los LEDs en color rojo

NO:

- *Desactivar los LEDs (LED Apagados)*
- *Actualizar la variable VarFM=" " (Sin ninguna unidad de almacenamiento)*

Fin de Rutina

6.2.2.5 Visualización de los parámetros

Una vez obtenida la información de la velocidad y calculada la distancia recorrida del automóvil. Esta información debe ser adecuada para su visualización en un LCD y posterior almacenamiento en cualquier unidad instalada en el sistema. El procesamiento de los datos se efectúa para mostrar los diferentes parámetros del automóvil en una manera legible para el usuario.

6.2.2.5.1 Visualización en LCD

Esta rutina controla los diferentes caracteres que se visualizan en el LCD, los parámetros mostrados en el LCD son: la velocidad instantánea, la distancia recorrida, el tiempo del trayecto recorrido, la fecha y la hora. Si la escritura en el dispositivo de almacenamiento está deshabilitada, los valores de distancia recorrida y tiempo del trayecto recorrido son equivalentes a cero unidades

Los parámetros visualizados en el LCD son constantemente actualizados mediante la rutina “Procesamiento de datos”. Para no tener dificultades en el barrido del LCD, se implemento una subrutina que incorpora una pausa.

Rutina “Visualización del LCD”

- *Borrar buffer del LCD*
- *Establecer posición en el LCD de los parámetros de Fecha/Hora*
- *Enviar información de Fecha/Hora al LCD*
- *Establecer posición en el LCD de los parámetros de velocidad*
- *Enviar información de velocidad al LCD*

Examinar ¿escritura habilitada?

SI:

- *Establecer posición en el LCD de los parámetros de distancia*
- *Enviar información de distancia al LCD*
- *Establecer posición en el LCD de los parámetros de tiempo*
- *Enviar información de tiempo al LCD*

NO:

- *Establecer posición en el LCD de los parámetros de distancia*
- *Enviar información de distancia= “0 m” al LCD*
- *Establecer posición en el LCD de los parámetros de tiempo*
- *Enviar información de tiempo=”00:00:00” al LCD*
- *Llamar a Subrutina “Pausa”*

Fin de la Rutina

6.2.2.5.2 Configuración del Reloj Externo

Esta rutina se encarga de la interfaz de comunicación entre el microcontrolador y el reloj externo DS1307. Una vez establecida la interfaz de comunicación el microcontrolador puede manipular las diferentes variables del reloj externo (segundo, minuto, hora, día, mes, año) .Estas variables también son utilizadas para ser visualizadas en el LCD.

Rutina "Configuración de Reloj"

- *Generar un código de START*
  - *Enviar instrucción de escritura "$HD0"*
  - *Enviar instrucción de lectura "$HD1"*

- *Recibir Fecha/Hora del DS1307*
  - *Generar un código de START "$HD1"*
  - *Recibir información del reloj externo DS1307*
  - *Almacenar variables en: $date, $times (Variables internas del microcontrolador)*

- *Configurar Fecha/Hora del DS 1307*
  - *Generar un código de START "$HD0"*
  - *Conversión a decimal las variables (segundo, minuto, hora, día, mes, año)*
  - *Enviar las variables al DS1307*

Fin de la Rutina

6.2.2.6 Instalar y adecuar la unidad de Memoria

En una primera parte este bloque instala la unidad de almacenamiento que se haya insertado, dicho en otras palabras habilita la escritura y lectura de la información por medio del USBwiz. Este proceso es automático, si existiese alguna variación en el estado de la memoria habilitada, instala cualquier unidad de almacenamiento disponible considerando el orden de prioridad (USB0, USB1, SD).

La segunda parte de esta subrutina prepara la unidad de almacenamiento, creando dos carpetas. La primera una carpeta de nombre “REGISTRO” que almacenara todos los recorridos que el usuario efectué, dentro de esta se crea la segunda carpeta que a su vez almacenará el archivo utilizado para guardar los diferentes parámetros del automotor. El nombre de la segunda carpeta corresponde a la fecha de su creación y el nombre del archivo corresponde a la hora de su creación.

6.2.2.6.1 Instalar Unidad actual

La Rutina “Instalar unidad actual” en un primer momento habilita la lectura/escritura de la unidad de almacenamiento y asigna valores a las variables: Varfmref (Variable de referencia de unidad instalada), Varcarpeta (Variable que referencia si el nombre de la carpeta corresponde a la unidad instalada).

En una segunda instancia la rutina verifica si la unidad instalada corresponde al valor de referencia y si el nombre de la carpeta se debe conservar. Cuando cualquiera de estos dos parámetros no se cumplen, se debe instalar nuevamente la unidad de almacenamiento actual; caso contrario, este proceso solo sirve de verificación.

Rutina “Instalar unidad actual”

Examinar ¿existe alguna unidad instalada?

SI:

- *Verificar la unidad instalada actual es igual a la unidad instalada anterior (Varfmrev=VarFM)?*

  Si:

  - *Asignar variable Varcarpeta =”no” (el nombre de la carpeta creada no debe cambiar)*

N0:

- *Asignar variable Varcarpeta =”si” (el nombre de la carpeta creada debe cambiar)*
- *Actualizar variable “VarFM” con el valor de la unidad actual (VarFM=Var2 o VarFM=Var3 o VarFM=Var4)*
- *Actualizar variable Varfmrev=VarFM (Variable asignada para revisión, cambio de unidad instalada)*

N0:

- *Asignar variable Varcarpeta =”si” (el nombre de la carpeta creada debe cambiar)*
- *Actualizar variable “VarFM” con el valor de la unidad actual (VarFM=Var2 o VarFM=Var3 o VarFM=Var4)*
- *Actualizar variable Varfmrev=VarFM (Variable asignada para revisión, cambio de unidad instalada)*

Fin de la Rutina

6.2.2.6.2 Crear carpeta y archivo en la unidad de almacenamiento

Después de comprobar que existe alguna unidad instalada y la escritura se encuentra habilitada, la rutina se encarga de establecer contacto con el USBwiz, para que pueda crear las carpetas y el archivo en el dispositivo de almacenamiento. Si no cumpliese con los requisitos establecidos la rutina no realiza ninguna acción.

El nombre asignado para la carpeta es la fecha de su creación y el nombre del archivo es la hora de creación de la carpeta. Otro proceso importante que cumple esta rutina es acceder al archivo creado y guardar un encabezado de los parámetros que serán almacenados.

Rutina "Crear Carpeta/Archivo"

Examinar ¿existe alguna unidad instalada?

SI:

- *Examinar el nombre de la carpeta debe cambiar (Varcarpeta ="no")?*

SI:

- *Crea una carpeta en la memoria instalada (Carpeta REGISTR0, esta almacenara todos los recorridos del automotor)*
- *Ingresa a la carpeta "REGISTR0" en la memoria instalada*
- *Se llama a la subrutina "Folder_file" (asigna un nombre a la carpeta y asigna un nombre al archivo)*
- *Crea una carpeta en la memoria instalada*
- *Ingresa a la carpeta actual en la memoria instalada*
- *Crea un archivo en la memoria instalada (La extensión del archivo es .CVS)*
- *Abre el archivo en la memoria instalada*
- *Escribe un encabezado en el archivo actual*
- *Cerrar y guardar la información del archivo actual*

N0:

- *No realizar ninguna acción*

N0:

- *No realizar ninguna acción*

Fin de la Rutina

6.2.2.7 Almacenar de la información en la memoria

En este bloque la información se almacenará en el archivo creado en la memoria flash o en la tarjeta SD. La información guardada corresponde a la velocidad instantánea del automotor, la distancia recorrida en cada lectura, el tiempo de recorrido por el automotor, el estado del automotor, y la hora de cada lectura.

6.2.2.7.1 Escribir en el Archivo

Una vez comprobado que existe una memoria instalada en el sistema, la rutina pueda acceder al archivo y guardar la información del recorrido del automotor, el archivo solo es abierto cuando se realiza la primera escritura. Mientras el automotor sigue su trayecto normal se continua con el proceso de almacenamiento en el archivo antes abierto, una vez realizadas treinta escrituras en el archivo este se cierra guardando sus cambios y repitiendo nuevamente el proceso. Este proceso se implementó para aumentar la cantidad de escrituras en el archivo y evitar el proceso de guardar el archivo continuamente, ya que esto último es lo que más tiempo consume.

Si no existiese ningún dispositivo de almacenamiento instalado en el sistema, esta rutina no realiza ninguna acción.

Rutina "Escribir Archivo"
Examinar ¿existe alguna unidad instalada?
SI:
Aumentar contador2

Examinar, Contador2=1?
Si:
- *Abre Archivo Actual de la memoria instalada*
- *Llamar subrutina "Enviar_info" (Escribe información en el archivo actual)*

N0:
Examinar, Contador2<30?
Si:
- *Escribe información en el archivo actual*
- *Llamar subrutina "Enviar_info" (Escribe información en el archivo actual)*

N0:
- *Cierra y guarda la información del archivo actual*
- *Reinicia contador2 (contador2=0)*

N0:
No realizar ninguna acción
Fin de la Rutina

6.2.2.8 Subrutinas Utilizadas

Las subrutinas son procesos secundarios que sirven de complemento para el programa del microcontrolador y de ayuda a las rutinas para que realicen las funciones por las cuales fueron creadas. Las subrutinas implementadas en el programa del microcontrolador son las siguientes:

6.2.2.8.1 Limites de las Variables del Reloj "Limites"

Esta subrutina tiene como finalidad establecer los límites numéricos de las variables utilizadas en el reloj en tiempo real. Además muestra los parámetros del reloj en el LCD del sistema cuando se está igualando.

Subrutina "Limites"

- *Asignar variables límites al rango (límite superior, límite inferior)*
- *Actualizar las variables en el LCD*

Fin de la Subrutina

6.2.2.8.2 Obtener el nombre del archivo y de la carpeta " Folder_file"

Esta subrutina toma los valores de hora y fecha de creación de la carpeta, para que sean asignados como nombres de la carpeta y del archivo creado en el dispositivo de almacenamiento, respectivamente.

Subrutina "Folder_file"

- *Asigna el valor de la fecha a la variable "foldername"*
- *Asigna el valor de la hora a la variable "filename"*

Fin de la Subrutina

6.2.2.8.3 Igualar Reloj del USBwiz y del Microcontrolador "Igualar_Reloj"

Esta subrutina es controlada por una interfaz externa conformada por un conjunto de pulsadores Esta interfaz facilita la configuración o igualación del reloj del

sistema. Al acabar el proceso de esta subrutina el reloj del sistema (DS1307) y el reloj del USBwiz son configurados con la hora y fecha que el usuario desee.

Subrutina "Igualar_Reloj"

- *Actualizar Variables*
  (Segundo, minuto, hora, día, mes, año)

- *Definir valor de segundo*
- *Llamar a la subrutina "limites"*
  (Delimita el rango de los segundos)

- *Definir, valor de minuto*
- *Llamar a la subrutina "limites"*
  (Delimita el rango de los minuto)

- *Definir valor de hora*
- *Llamar a la subrutina "limites"*
  (Delimita el rango de las horas)

- *Definir valor de día*
- *Llamar a la subrutina "limites"*
  (Delimita el rango de los días)

- *Definir valor de mes*
- *Llamar a la subrutina "limites"*
  (Delimita el rango de los meses)

- *Definir valor de año*
- *Llamar a la subrutina "limites"*
  (Delimita el rango de los años)

- *Configurar el valor actual de reloj/fecha del microcontrolador*
- *Configurar el valor actual de reloj/fecha del USBwiz*

Fin de la Subrutina

6.2.2.8.4 Enviar información "Enviar_info"

Esta subrutina es utilizada para enviar la información depurada de cada lectura hacia el USBwiz. Cuando se desea que el USBwiz reciba información, primero se tiene que enviar un comando de escritura, identificando cuantos caracteres se van a enviar y a que unidad, posteriormente se envía toda la información que el USBwiz recibirá. Un parámetro importante que hay que tomar en cuenta es el tiempo que se demora el USBwiz en captar los datos y escribir en el archivo de destino. Para eso se implemento una pausa al final de esta subrutina en espera del siguiente comando.

Subrutina "Enviar_info"

- *Enviar comando de escritura por el puerto UART.*
- *Llamar subrutina "Espacio" (Añade espacios necesarios en cada variable).*
- *Enviar variable referente a la Hora/Fecha.*
- *Enviar variable referente a la Velocidad.*
- *Enviar variable referente a la Distancia.*
- *Enviar variable referente a la Actividad del automotor.*
- *Enviar variable referente a la Tiempo.*
- *Llamar subrutina "Pausa" (Añade pausa en espera del siguiente comando).*

Fin de la Subrutina

6.2.2.8.5 Generar Pausa "Pausa"

Esta subrutina incorpora una pausa utilizando dos lazos repetitivos con la instrucción "nop". El tiempo que dura la pausa es determinada por los contadores que se utilizan en cada lazo.

Subrutina. Pausa

Crear Lazo1: incrementar Contador1
Crear Lazo2: incrementar Contador2
Instrucción de pausa "nop"
Continuar Lazo2
Continuar Lazo1

Fin de la Subrutina

6.2.2.8.6 Generar espacios de relleno "Espacio"

Al enviar información al USBwiz se debe determinar la longitud de cada trama enviada. Cada trama tiene que tener una longitud fija, tomando en cuenta esta condición, el programa del microcontrolador incorpora esta subrutina para rellenar los espacios faltantes en cada variable.

Subrutina "Espacio"

- *Medir longitud de caracteres de la variable "acción"*
  Establecer, el número de espacios necesarios para enviar con la variable "acción"

- *Medir longitud de los caracteres de la variable "velocidad"*
  Establecer, el número de espacios necesarios para enviar con la variable "velocidad"

- *Medir longitud de los caracteres de la variable "distancia"*
  Establecer, el número de espacios necesarios para enviar con la variable "distancia"

- *Medir longitud de los caracteres de la variable "tiempo"*
  Establecer, el número de espacios necesarios para enviar con la variable "tiempo"

Fin de la Subrutina

6.2.2.8.7 Obtener valor del Timer "Tiempo"

Esta subrutina utiliza el timer1 para su proceso. Primero se guarda el valor del timer1 en una variable llamada "vartiempo", esta variable se utiliza como el tiempo entre lectura y lectura. Al finalizar la subrutina se reinicia el timer1 para una nueva lectura.

Subrutina "Tiempo"

- *Almacenar el valor del Timer1 en la variable "vartiempo"*
- *Adecuar la variable "vartiempo" (Multiplicar por una constante dependiendo de la escala que se utilice en el timer1)*
- *Reiniciar Timer1 (Timer1=0)*

Fin de la Subrutina

6.2.2.9 Interrupciones

Una interrupción es un proceso que detiene momentáneamente el programa del microcontrolador para saltar a una parte específica del programa. Las interrupciones son utilizadas para realizar procesos que no pueden esperar el orden continuo del programa.

Para permitir el uso de interrupciones es necesario habilitarlas al inicio del programa. Cada interrupción habilita a un subprograma que se efectuará cuando se invoque la interrupción. Una vez culminada la ejecución del subprograma, se retorna a la parte del programa donde se detuvo. Las interrupciones que se utilizaron en el programa del microcontrolador son dos interrupciones externas y una interrupción serial que se describen a continuación:

6.2.2.9.1 Interrupción de Comandos Serial "Serial_IN"

Esta interrupción es utilizada para identificar los eventos que suceden en el USBwiz, por ejemplo cuando algún byte es recibido por el puerto UART del microcontrolador este activa la interrupción.

El proceso que es activado por esta interrupción almacena los bytes recibidos por el puerto UART, y los compara con los eventos que produce el USBwiz, generalmente estos eventos son la inserción o remoción de los diferentes dispositivos de almacenamiento.

Interrupción "Serial_IN"

- *Almacenar la información entrante por el puerto UART del microcontrolador (Respuestas del USBwiz)*
- *Comparar la información entrante por el puerto UART*
- *Seleccionar el evento correspondiente a cada respuesta del USBwiz*
- *Reiniciar el buffer de la interrupción serial*

Retornar al Programa

6.2.2.9.2 Desinstalar Unidad de almacenamiento

Esta interrupción externa posibilita desinstalar cualquier dispositivo de almacenamiento habilitado en el sistema. Cuando se activa este proceso se guarda los cambios realizados en el archivo en uso, para después instalar otra unidad que este en el sistema. De no existir ninguna memoria insertada se asigna a la variable "VarFM" un valor correspondiente a vacio (Sin dispositivo de almacenamiento en el Sistema).

Interrupción "Desinstalar"

- *Guardar información del archivo actual*
- *Cerrar archivo actual del dispositivo de almacenamiento*
- *Identificar la unidad instalada (VarFM=Var2 o VarFM=Var3 o VarFM=Var4)*
- *Desinstalar la unidad actual.*

- *Examinar ¿existe alguna unidad insertada?*
  SI:
  - *Instalar la unidad insertada al sistema, siguiendo la prioridad (USB0, USB1, SD)*

  NO:
  - *Reiniciar el valor de VarFM=" " (Sin dispositivo de almacenamiento en el Sistema)*

Retornar al programa

6.2.2.9.3 Habilitar escritura en el dispositivo de almacenamiento "Iniciar & Parar"

Esta interrupción funciona como un interruptor. Al ser accionado por primera vez, habilita la escritura en el dispositivo de almacenamiento utilizando una variable de referencia " Var_ini_par=1" . Cuando se desea finalizar el almacenamiento, esta interrupción externa es accionada y la variable de referencia cambia " Var_ini_par=0" dejando deshabilitado la escritura en la dispositivo de almacenamiento. Una vez que se deshabilita el almacenamiento se guardan los cambios del archivo y la memoria puede ser removida del sistema o aguardar que se habilite el almacenamiento nuevamente.

Interrupción "Iniciar & Parar"

Examinar ¿existe alguna unidad instalada?

SI:

- *Examinar está habilitada la escritura en el dispositivo de almacenamiento (Var_ini_par=1)?*
  - *SI:*
    - *Deshabilitar la escritura (Var_ini_par=0)*
  - *NO:*
    - *Habilitar la escritura (Var_ini_par=1)*

NO:

- *No realizar ninguna acción*

Retornar al programa

6.2.3 PROGRAMACIÓN EN EL USBWIZ-OEM

La interfaz que se implementa entre el USBwiz y el microcontrolador es utilizada para enviar comandos y recibir respuestas por parte del microcontrolador. Los diversos comandos utilizados por el microcontrolador le posibilitan acceder a Memorias Flash USB y Tarjetas Flash.

6.2.3.1 Comandos (USBwiz)

Se puede enviar varios comandos seguidos al USBwiz hasta que la FIFO se llene El USBwiz revisa los comandos recibidos individualmente siguiendo el orden en que llegaron. Una vez procesados los diferentes comandos, envía respuestas de cada uno de ellos.

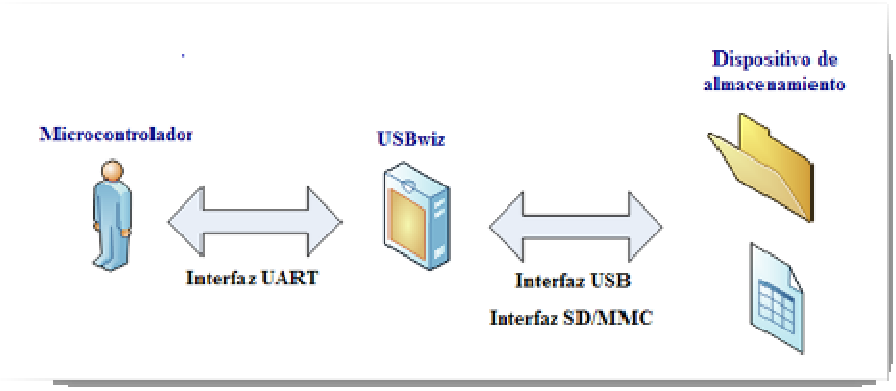


Figura 2.65 “Interfaz de comunicación entre el USBwiz-OEM y el Microcontrolador”

Todos los comandos enviados por el microcontrolador hacia el USBwiz utilizan un formato ASCII.

6.2.3.1.1 Comandos del Cargador de Arranque

Los comandos descritos a continuación son referentes al firmware del USBwiz. El comando “R” reinicia el firmware del USBwiz, y por otra parte el comando “V” retorna que versión de firmware se está utilizando.

Comando	Uso	Nota:
R	Carga y corre el firmware del USBwiz.	Si el cargador de Arranque retorna el comando “BL”. El USBwiz necesita ser reiniciado
V	Retorna la versión Cargada	Retorna el valor de la versión en ASCII

Tabla 2.9 “Comandos del Cargador de Arranque”

6.2.3.1.2 "IT" Inicializar el reloj en tiempo Real (RTC).

El USBwiz tiene incorporado un reloj en tiempo real, que es usado para guardar la fecha en los archivos creados. Los usuarios necesitan especificar si el USBwiz estará corriendo con el RTC que usa el oscilador interno del USBwiz o el que usa un oscilador de 32Khz y una batería independiente. La ventaja de correr el RTC en un oscilador de 32Khz y con una batería, es que el reloj no perderá su información cuando la fuente de poder principal se pierda en el USBwiz.

Formato:	IT t	Donde "t" es el modo que correrá el RTC. Puede ser "S" si se utiliza una batería externa o puede ser "B" si se utiliza el oscilador del USBwiz.
Ejemplo:	IT B	Esta configuración habilita al USBwiz utilizar el RTC con el oscilador interno del USBwiz.

Tabla 2.10 "Utilización del Comando IT Inicializar el reloj en tiempo Real "

6.2.3.1.3 "GT" Obtener parámetros del reloj en tiempo real.

Respecto a la hora actual, el valor devuelto por el USBwiz puede ser un número en hexadecimal o una cadena de caracteres en ASCII.

Formato:	GT f !00	"f" puede ser: F: Obtener el Valor en ASCII X: Obtener el valor en Hexadecimal
Ejemplo 1:	GT F !00 01/20/2006 - 06:21:03 !00	Obtener la hora actual en formato ASCII
Ejemplo 2:	GT X !00 34210000 !00	Obtener la hora actual en formato hexadecimal

Tabla 2.11 "Utilización del Comando GT, Obtener el reloj en tiempo real"

6.2.3.1.4 "BR" Configurar velocidad de Transmisión del Puerto UART

La velocidad por defecto del puerto UART es de 9600. Si bien esta velocidad de transmisión es extremadamente lenta algunos sistemas no soportan velocidades más altas. La tasa de transferencia se puede ajustar a diferentes velocidades estándar. El comando BR configura el divisor interno del registro del UART, de esta forma posibilita la configuración a cualquier tasa de transferencia estándar. Para calcular el valor del divisor se usa la formula $\left(\frac{OSC \times 4}{Baud\ Rate \times 16}\right)$ .El oscilador "OSC" que usa el USBwiz es 14745600.

Formato:	BR vvvv	vvvv: Palabra en Hexadecimal (Equivalente a la velocidad de Transmisión
Ejemplo:	BR 0020	Velocidad de Transmisión 115200

Tabla 2.12 "Utilización del Comando BR, configurar velocidad de transmisión del puerto UART"

A continuación se presenta la Tabla 2.13 con la Equivalencia de las velocidades "Baud Rate"

Baud Rate	Divisor en Decimal	Divisor en Hexadecimal (Utilizado en el comando BR)
9600	384	0180
19200	192	00C0
38400	96	0060
57600	64	0040
115200	32	0020
921600	4	0004

Tabla 2.13 "Equivalencia de Velocidades"

6.2.3.1.5 "ST" Configuración del reloj en tiempo Real

Para configurar una hora o fecha actualizada en el RTC del USBwiz se debe utilizar el comando ST. La resolución del RTC es de 2 segundos.

Formato:	ST t	Donde "t" es una palabra (DWORD) que representa a la fecha y hora que se desea configurar el RTC.
Ejemplo:	ST 34210000	RTC configurado a: 1/1/2006 00:00:00

Tabla 2.14 "Utilización del Comando ST, Configuración del reloj en tiempo Real"

A continuación se presenta la Tabla 2.15 explicando el formato de la Palabra (DWORD) que representa la fecha/hora, la longitud de esta palabra en código binario es de 32bit.

Bits(s)	Campo	Descripción
31..25	Año1980	Años desde 1980
24..21	Mes	1..12
20..16	Día	1..31
15..11	Hora	0..23
10..5	Minuto	0..59
4..0	Segundo	Segundos divididos para 2 (0..30)

Tabla 2.15 "DWORD que representa la Fecha/Hora"

6.2.3.1.6 "FM" Instalar una unidad de almacenamiento en el sistema

El USBwiz puede instalar hasta 3 dispositivos de almacenamiento al mismo tiempo.

Formato.	FM d	Donde "d" es el dispositivo que se desea instalar: 0: Dispositivo USB0 1: Dispositivo USB1 S: Dispositivo SD/MMC
Ejemplo:	FM 0	Dispositivo USB0 Instalado en el sistema

Tabla 2.16 "Utilización del Comando FM, Instalar un archivo de sistema"

6.2.3.1.7 "MD" Crear un directorio (Carpeta)

Este comando es utilizado para crear un directorio en cualquier dispositivo de almacenamiento instalado en el sistema.

Formato:	MD foldername	"foldername" es el nombre que se le va asignar al directorio creado (El nombre no puede sobrepasar los 8 caracteres)
Ejemplo:	MD PRUEBA1	Crear una carpeta con el nombre PRUEBA1

Tabla 2.17 "Utilización del Comando MD, Crear un directorio"

6.2.3.1.8 "CD" Cambio de Directorio o Acceso al directorio

Cambia la carpeta de acceso actual. La Carpeta debe existir para poder utilizar este comando.

Formato:	CD foldername	"foldername" es el nombre del directorio al que se desea ingresar (el archivo tiene que existir o se producirá un error
Ejemplo:	CD PRUEBA1	Ingresa a la carpeta cuyo nombre es PRUEBA1

Tabla 2.18 "Utilización del Comando CD, Cambio de Directorio"

6.2.3.1.9 "WF" Escribir en un archivo

El comando "WF" es utilizado para escribir en cualquier archivo de una unidad instalada. Para utilizar este comando es necesario que el archivo este abierto.

Después de escribir en el archivo, el USBwiz emite un código de respuesta de éxito (¡00) y una confirmación de cuantos bytes fueron escritos en el archivo (Hexadecimal). Si bien el tamaño de cada trama es arbitrario se recomienda tramas menores a 100 bytes.

Formato:	WF n>ssssssss (Respuesta retornada) !00 $aaaaaaaa !00	n: Representa la unidad donde se encuentra el archivo abierto para su escritura. ssssssss :El número de caracteres que se desea escribir en el archivo (Hexadecimal) $aaaaaaa: Confirmación del número de caracteres escritos en el archivo (Hexadecimal)
Ejemplo:	WF 1>10 !00 $00000010 !00	Escribir 16 bytes en el archivo abierto de la unidad USB1

Tabla 2.19 "Utilización del Comando WF, Escribir en un archivo"

6.2.3.1.10 "EV" Habilitar/Deshabilitar Eventos

El USBwiz puede detectar la remoción o inserción de las tarjetas SD y de los dispositivos USB. Todos los eventos son retornados con el símbolo % cuando son habilitados.

Formato	EV v !00	"v" puede ser: 0: Deshabilitar todos los eventos 1: Habilitar eventos de la tarjeta SD 2: Habilitar eventos de los dispositivos USB 3: Habilitar todos los eventos
Ejemplo	EV 3 !00	Habilitar todos los eventos

Tabla 2.20 "Utilización del Comando EV, Habilitar/Deshabilitar Eventos"

6.2.3.1.11 "OF" Abrir un archivo

El comando "OF – Open File" posee tres modos de operación:

- **Modo de operación "R":** En este modo el archive es abierto y solo se permite el acceso para su lectura. El archivo necesariamente tiene que existir para utilizar este comando.

- **Modo de operación 'W':** En este modo se puede crear un archivo y guardar información en este. Si se utilizara este modo para abrir un archivo existente se borra toda la información dentro del archivo y la sobrescribirá con nueva información entrante.

- **Modo de operación 'A':** En este modo se puede crear un archivo y guardar información en este. Con la diferencia que cuando se abre un archivo existente en este modo, no se borra la información anterior del archivo, sino que la nueva información se anexa al archivo.

Nota: El archivo creado o abierto puede tener cualquier tipo de extensión.

Formato:	OF nM>filename	"filename" representa al archivo que se desea abrir o guardar. "n" es la unidad instalada en el sistema (USB0=0, USB1=1, SD=2) "M" es el modo que se desea utilizar
Ejemplo1:	OF 1R>RESERVA.LOG	Abrir el archivo RESERVA.LOG instalado en la unidad USB1, solo para lectura.
Ejemplo2:	OF 0W>DATOS.TXT	Abrir el archivo DATOS.TXT instalado en la unidad USB0, borrar la información inicial y escribir una nueva.

Tabla 2.21 "Utilización del Comando OF, Abrir un archive para leer, escribir o crear archivo"

6.2.3.1.12 "RF" Lee la información desde un archivo

Este comando permite leer la información contenida por un archivo. Para la utilización de este comando es necesario que el archivo este abierto

Formato:	RF nM>D (Respuesta retornada) !00 sssssss $aaaaaaa	n: representa la unidad donde se encuentra el archivo abierto para su lectura M: El carácter de relleno que se utilizara si es necesario D: El numero de caracteres que se desea leer (Hexadecimal) sssssss: Los caracteres leídos en el archivo $aaaaaaa: Confirmación del numero de caracteres leidos
Ejemplo1:	RF 1Z>5 !00 ABCDE $00000005 !00	Leer 5 caracteres del archivo abierto en la unidad USB1, con carácter de relleno "Z" Se leyeron 5 caracteres ABCDE (No necesario carácter de relleno)
Ejemplo2:	RF 0Z>6 !00 FGHZZZ $00000003 !00	Leer 6 caracteres del archivo abierto en la unidad USB0, con carácter de relleno "Z" Se leyeron 3 caracteres FGHZZZ (Se necesito carácter de relleno Z)

Tabla 2.22 "Utilización del Comando RF, Lee la información desde un archivo"

6.2.3.1.13 ."CF" Cierra y Guarda los cambios en el archivo

Este comando cierra los archivos abiertos en cualquier dispositivo instalado en el sistema. Previamente se guarda la información de cada archivo que se encuentre abierto.

Formato:	CF n	"n" es la unidad instalada que contiene el archivo abierto (USB0=0, USB1=1, SD=2)
Ejemplo:	CF 0	Cierra y guarda los archives abiertos en la unidad USB0

Tabla 2.23 "Utilización del Comando CF, cierra y guarda los cambios en el archivo"

6.2.3.2 Repuestas o Eventos (USBwiz)

Una vez que el USBwiz recibe y procesa cualquier comando, este remite una respuesta Las respuestas emitidas por el USBwiz pueden ser confirmando el éxito del proceso del comando o informando de algún error en la transacción utilizada.
A continuación se presenta la Tabla de las principales respuestas emitidas por el USBwiz.

Valor (HEX)	Descripción
00	Operación procesada satisfactoriamente
09	Falla en firmware
11-14, 13-24,	Fallo en lectura del dispositivo, Se recomienda formatear desde el PC.
15	La conexión con dispositivos FAT12 no es soportada.
25, 35	Dispositivo Lleno.
4F	Archivo actual en uso, se debe cerrar el archivo
52	El nuevo nombre del Archivo y existe
60	Fallo en la inicialización del controlador Host ISP1160.
61 – 6D	No se complemento la transacción USB.
6E	Fallo en la comunicación del controlador Host ISP1160.
78	Dispositivo USB no responde.
79	Dispositivo USB corrupto.
7A	Descriptor USB corrupto.
81-83, 85-86	Dispositivo de almacenamiento masivo USB, fallo.
90	Dispositivo de almacenamiento masivo USB, no está listo
91	Dispositivo de almacenamiento masivo USB, usa protocolos
92	Dispositivo de almacenamiento masivo USB, usa subclass no
A3	Nombre invalido
A4	Numero invalido
A5	No se pudo completar la solicitud de escritura
A7	No se pudo inicializar el dispositivo.
A8	El comando requerido contiene parámetros incorrectos.
AA, D4	El Checksum es erróneo.
AB	El archivo de sistema FAT no está instalado.
AF, D3, FD	Error interno.

Valor (HEX)	Descripción
B2	ID del vendedor no es válida.
B3	ID del producto no es válido.
D7	Archivo de firmware invalido.
D8, DE	Comando desconocido
DF	Interfaz nula.

Tabla 2.24 "Respuestas, Emitidas por el USBwiz"

Los eventos en el USBwiz son interrupciones que generan un código que se emite por la interfaz serial. Los eventos utilizados por el USBwiz se describen en la Tabla 2.25.

Valor (HEX)	Descripción
01	Tarjeta SD/MMC Insertada
02	Tarjeta SD/MMC Removida
20	Fallo inicialización con el ISP1160.
30	Dispositivo USB insertado en el Pórtico 0
31	Dispositivo USB removido en el Pórtico 0
32	Dispositivo USB insertado en el Pórtico 1
33	Dispositivo USB removido en el Pórtico 1

Tabla 2.25 "Eventos del USBwiz"

En el próximo capítulo se hablará sobre la instalación del módulo electrónico en una camioneta Chevrolet LUV la cual fue escogida para realizar las pruebas del funcionamiento del módulo electrónico.

CAPÍTULO 3

IMPLEMENTACIÓN DEL HARDWARE DEL SISTEMA

CAPITULO 3

IMPLEMENTACIÓN DEL HARDWARE DEL SISTEMA

En el presente capitulo se explica el modo correcto de instalar el sistema propuesto en un automotor, tomando muy en cuenta la obtención de la señal de velocidad del automotor y la fuente de poder utilizada por el módulo electrónico. Se presentará información que facilitará el entendimiento de la instalación del sistema y porque el de su ubicación. El automotor escogido para este fin en una camioneta Chevrolet LUV del año 2005.

7.1 AUTOMOTOR PROPUESTO PARA REALIZAR LAS PRUEBAS

Chevrolet (también conocida como "Chevy") es una marca de automóviles de Estados Unidos perteneciente al grupo General Motors. En los últimos 8 años esta compañía se ha convertido en líder en la fabricación de automóviles, con un nivel de penetración mundial del 17.7%. En América latina, gracias a las fuertes inversiones y diversos modelos comercializados, han reafirmado su liderazgo como fabricante de automotores. Dentro de esta región, General Motors registró un crecimiento en siete mercados, incluyendo Ecuador, donde la empresa consiguió el nivel de penetración más alto en todo el mundo, con un 46%. [14]

El alto nivel de penetración y el vasto listado de modelos de la marca Chevrolet hacen atractivo a esta marca comercial de automotores para usarse como el automotor predeterminado en la implementación del módulo electrónico fabricado en el presente proyecto. El modelo específico de automotor en el cual se instalará el módulo electrónico es una camioneta Chevrolet LUV año 2005. Esta camioneta recoge las características de los automóviles que actualmente se comercializan en el mercado Ecuatoriano.

[14] http://www.lukor.com/not-neg/empresas/0601/09115250.htm

En los últimos años los automotores implementan en su fabricación una computadora interna que registra y controla su funcionamiento, para adquirir la información necesaria del automotor la computadora usa diferentes dispositivos de control, como por ejemplo el velocímetro y sensores internos ubicados en el motor.

En el proyecto presente se utilizará la señal que genera el velocímetro hacia el módulo de control del motor (ECM[15] – Cerebro del automóvil) para adquirir la medida referente a la velocidad del automotor.

7.2 OBTENCIÓN DE LA SEÑAL DE VELOCIDAD DEL AUTOMOTOR (SEÑAL GENERADA POR EL VELOCÍMETRO)

Al observar la camioneta Chevrolet LUV en su panel frontal posee un velocímetro analógico, sin embargo este modelo de automotor posee un sensor digital para obtener los parámetros de velocidad. La señal emitida por el sensor de velocidad es procesada por el velocímetro ubicado en el panel frontal de la cabina y posteriormente enviada al Módulo de control del motor (Cerebro del Automóvil o ECM). En la **Figura 3.66** se puede ver el velocímetro analógico instalado en la camioneta Chevrolet LUV.

Figura 3.66 "Velocímetro analógico ubicado en el panel frontal"

[15] **ECM** es el acrónimo de "Enterprise Content Management". Traducido al español Módulo de control del motor. También conocido como cerebro del automóvil.

En la camioneta Chevrolet LUV el sensor de velocidad está ubicado en la caja de cambios del automotor. Si se toma de referencia el armazón de la camioneta, el sensor se encuentra en la parte inferior derecha del automotor muy próximo al neumático derecho delantero, como se puede ver en la **Figura 3.67**.

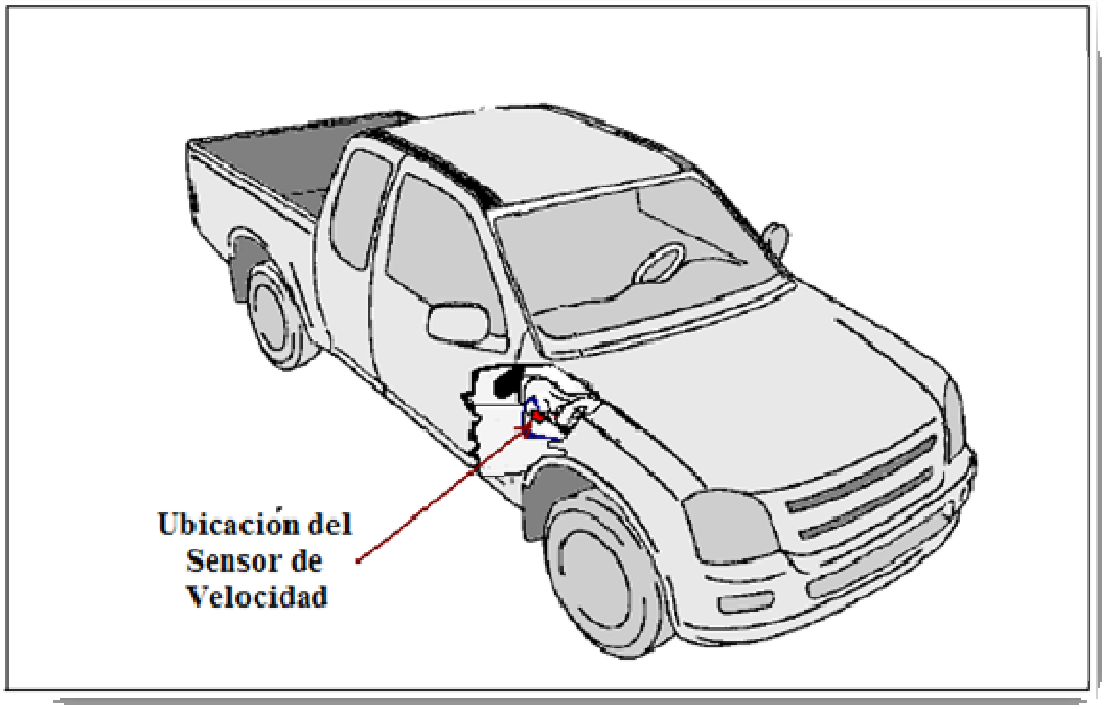


Figura 3.67 "Ubicación del sensor de velocidad tomando referencia el armazón del automotor"

La **Figura 3.68** muestra la ubicación del sensor de velocidad desde la vista inferior del automotor. El sensor de velocidad de la camioneta está instalado en la parte interna de la caja de cambios y desinstalarlo provocaría averías en su funcionamiento, por lo que se prefirió tomar la señal generada por el velocímetro del automotor. Otro factor importante que influyó en la obtención de la señal desde el velocímetro fue que al tomar la señal generada por el sensor de velocidad, la frecuencia de la señal depende de la marcha en que se encuentre el

automotor y provocaría la utilización de otra referencia para obtener la señal de la velocidad real del automotor.

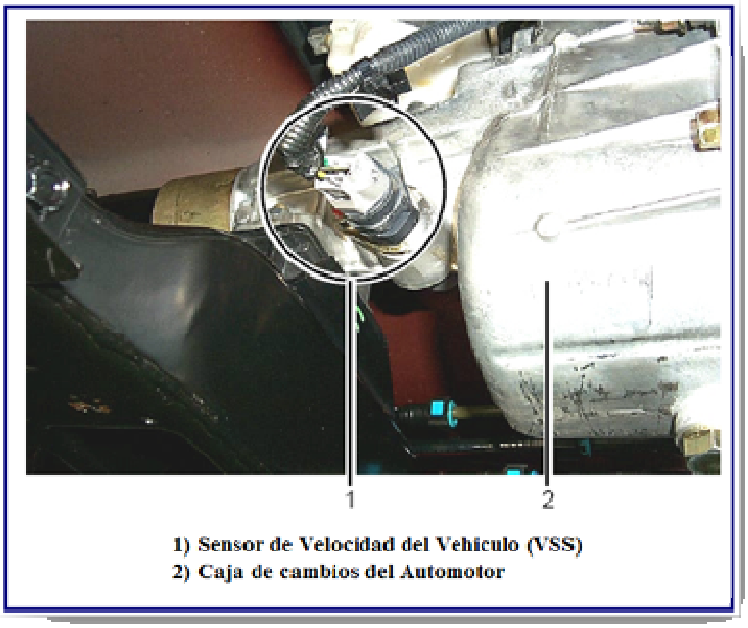


Figura 3.68 "Ubicación del sensor de velocidad en la transmisión"

Como ya se mencionó, si bien el sensor de velocidad genera una señal cuya frecuencia es proporcional a la velocidad del automotor, esta señal también depende de la marcha que el automóvil tenga en ese momento. Este problema es solucionado por el velocímetro, el cual procesa la señal del sensor de velocidad y la compara con la marcha en la que se encuentra el vehículo en ese momento. Una vez adecuada la señal del sensor de velocidad, el velocímetro envía la señal acondicionada hacia el módulo de control del motor (ECM). Las diferentes características del sensor de velocidad se obtuvieron del "Manual de Servicio Motor C24SE (Automotor Chevrolet LUV 2005)" cuyo extracto se anexa en el presente documento (ANEXO D).

En la **Figura 3.69** se puede observar la conexión que existe entre el sensor de velocidad, el velocímetro y el módulo de control del motor. La conexión de cada dispositivo es realizado por cables cuyo color es igual a los que se muestran en la **Figura 3.69**

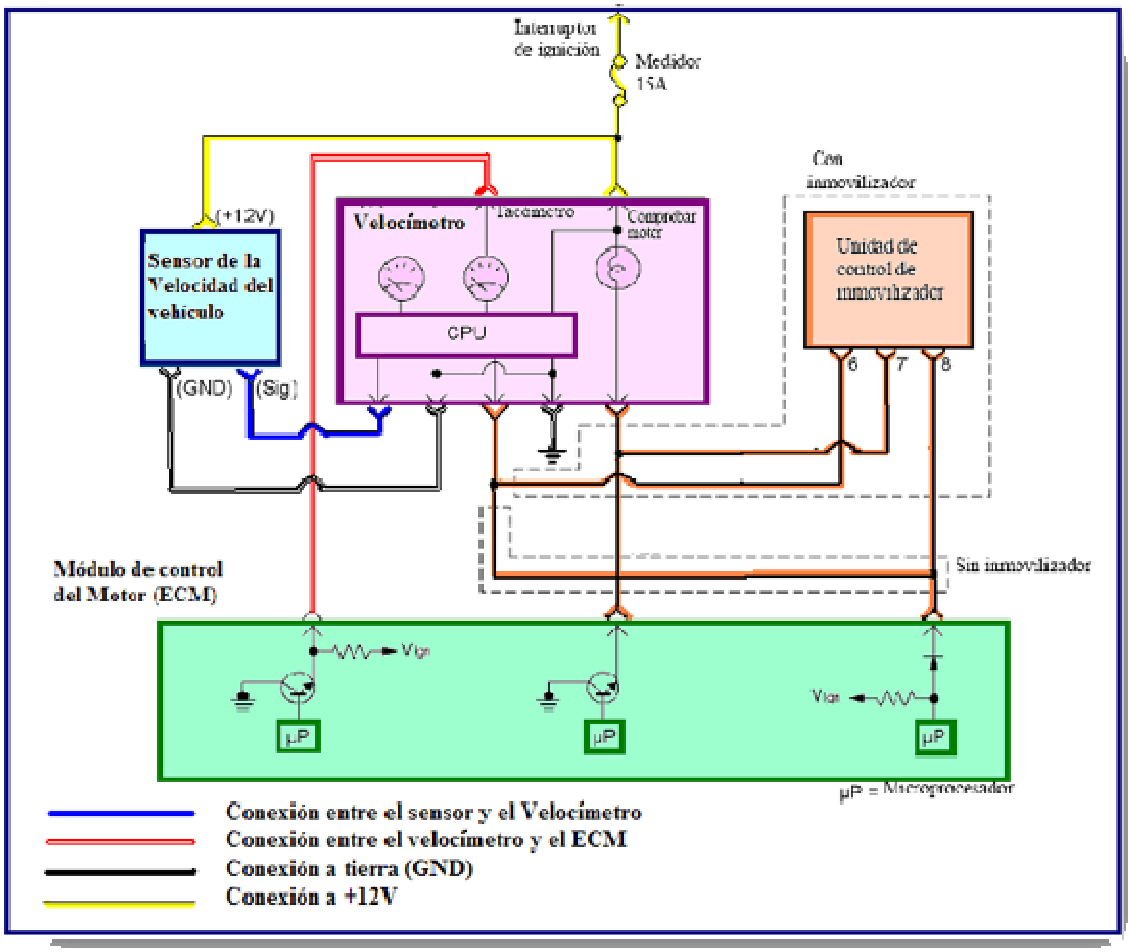


Figura 3.69 "Conexión del sensor de velocidad, el velocímetro y el Módulo de control del motor (ECM)"

La conexión entre el sensor de velocidad y el velocímetro es realizada con tres conductores, dos de ellos proveen la fuente de alimentación del sensor (Cable Negro = GND, Cable Amarillo = VCC) y el restante la salida de la señal del sensor (Cable Azul). Para el proyecto presente, la señal de velocidad del automotor se tomará de la conexión que existe entre el velocímetro y el ECM (Conductor color rojo/blanco).

En las **Figura 3.70** y **Figura 3.71** se puede observar el conjunto de conductores que establecen la conexión entre el sensor de velocidad del automotor, el velocímetro y el módulo de control del motor.

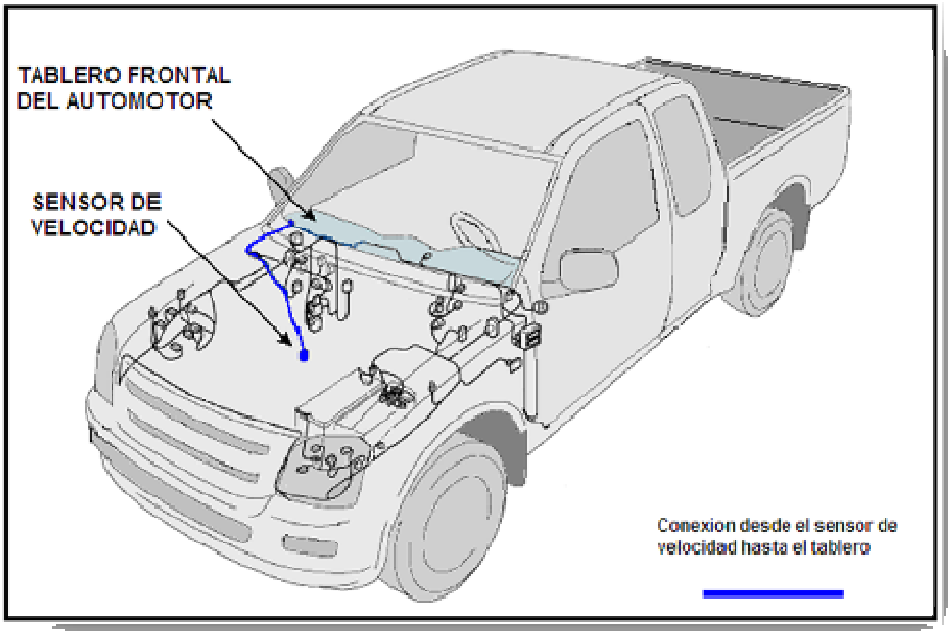


Figura 3.70 "Conexión del sensor de velocidad hacia el tablero"

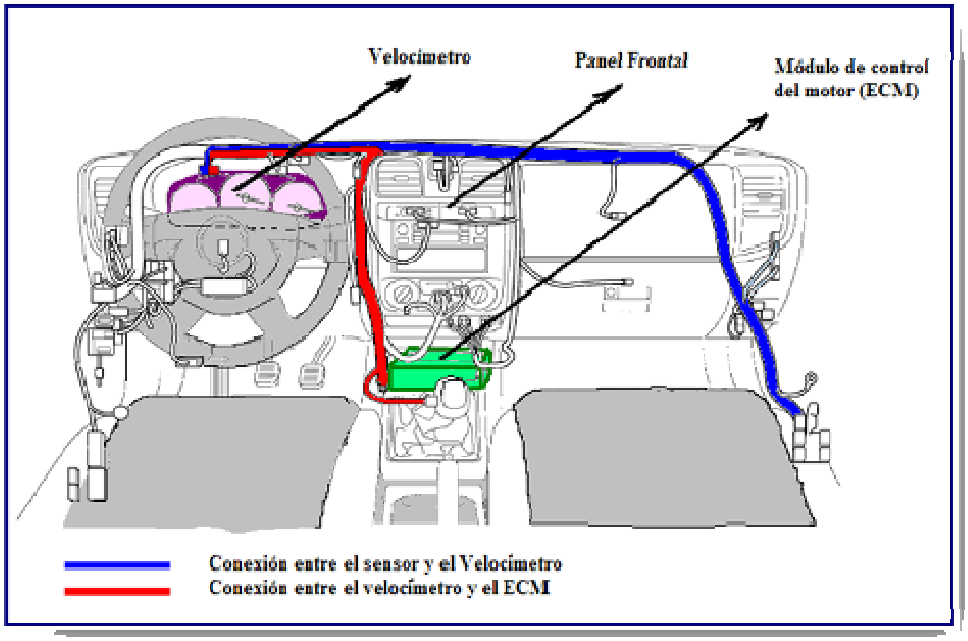


Figura 3.71 "Cableado del panel frontal en el automotor"

Una vez establecida la conexión que se establece entre el velocímetro y el módulo de control del motor (ECM), se procede a realizar la conexión entre el velocímetro y el módulo electrónico diseñado en el proyecto.

7.2.1 CONEXIÓN DEL VELOCIMETRO AL MÓDULO ELECTRONICO

Como ya se mencionó anteriormente, el velocímetro envía su información al módulo de control del motor por medio de un conductor color rojo/blanco. Este conductor pasa por el panel frontal ubicado en la cabina del automóvil, Basta con ubicar el conductor de la señal del velocímetro para realizar el empalme correspondiente.

Si bien la señal que genera el velocímetro se podría obtener realizando un empalme en el conductor que va hacia el ECM, el acceso al conector del ECM es dificultoso y la obtención de la señal por este medio podría causar daños en la sistema de conexión del motor. Tratando de evitar posibles daños que pudiesen surgir con el ECM, la señal se tomará directamente del velocímetro a pesar de los inconvenientes propios de la instalación.

En la **Figura 3.72** se muestra la ubicación del ECM en el panel frontal de la cabina.

Figura 3.72 "Ubicación del Módulo de control del motor"

En la Figura 3.8 se puede observar los pasos necesarios para desmontar el velocímetro del panel frontal de la cabina.

Figura 3.73 "Pasos para desmontar el Velocímetro del Panel frontal de la cabina"

- **Paso1.-** Identificar los diferentes tornillos y contactos que sujetan al tablero en la cabina del automotor.

- **Paso2.-** Desmontar la protección del ECM y remover las protecciones del volante del vehículo.

- **Paso3.-** Desmontar la cubierta de protección del velocímetro.

- **Paso4.-** Remover el velocímetro de su ubicación en el tablero del automotor

En la **Figura 3.74** se puede observar la conexión que se realiza en el velocímetro para obtener la señal de la velocidad del automotor. Esta señal (S. OUT) será la entrada del módulo electrónico del proyecto que se está desarrollando.

Figura 3.74 "Conexión del Velocímetro"

7.3 CONEXIÓN A LA BATERÍA DEL AUTOMOTOR

Aprovechando la batería del automotor se hizo una conexión hacia el módulo electrónico del proyecto. En la Figura 3.10 se muestra la ubicación de la batería en la camioneta Chevrolet LUV 2005 que se utiliza en la implementación del proyecto realizado.

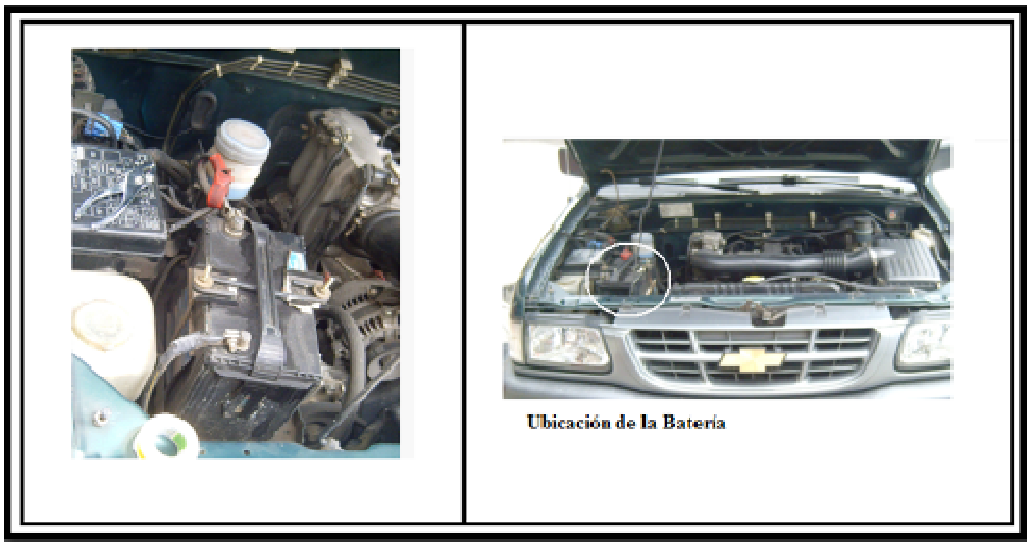


Figura 3.75 "Ubicación de la Batería"

La conexión se debe realizar con mucho cuidado evitando que los cables se junten para no producir corto circuito. En la Figura 3.11 se muestra el cable que corresponde a VCC y el cable que corresponde a GND.

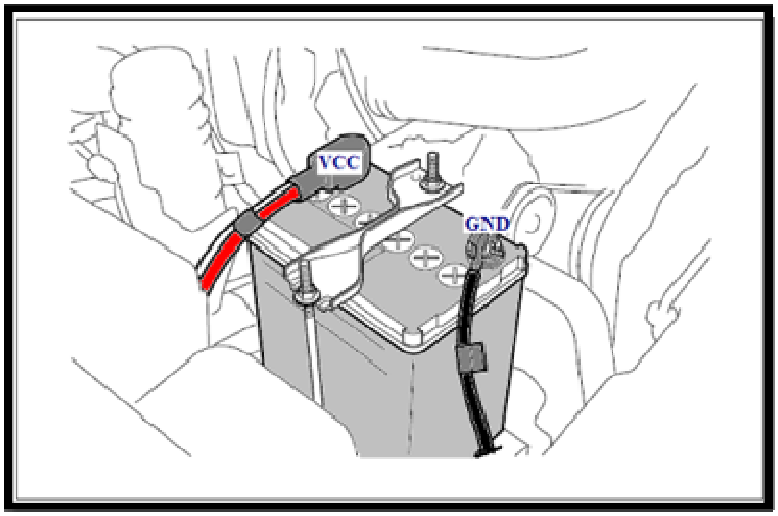


Figura 3.76 "Conexión de VCC y GND en la Batería"

La conexión se realiza desmontando los cables de poder de la batería para después colocar los conectores de la extensión que proveerá de energía al módulo electrónico. Los conectores deben estar fuertemente ajustados en los cables de poder de la batería para evitar pérdidas en la alimentación de energía tanto para el automotor como para el módulo electrónico. En la Figura 3.12 se muestra la colocación de los conectores en los cables de poder de la batería.

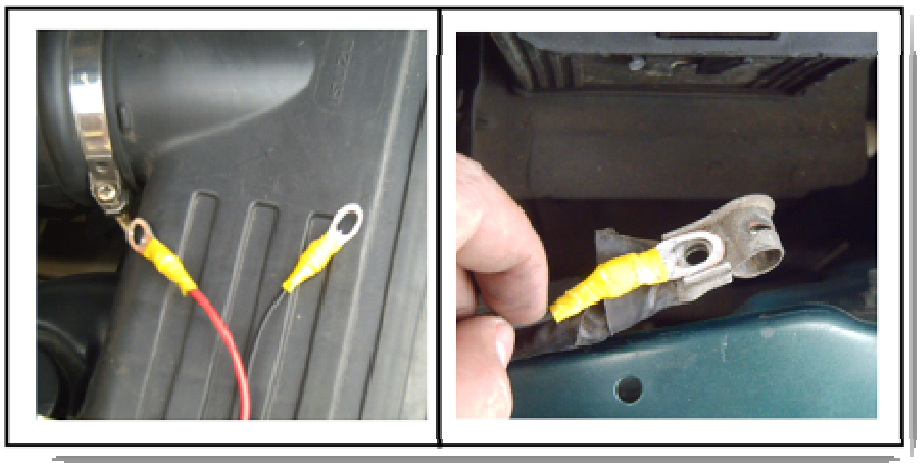


Figura 3.77 "Conectores colocados en el cable de poder de la batería (GND)"

En la Figura 3.13 se muestra la conexión realizada entre los cables de poder de la batería y la extensión utilizada para la alimentación del módulo electrónico.

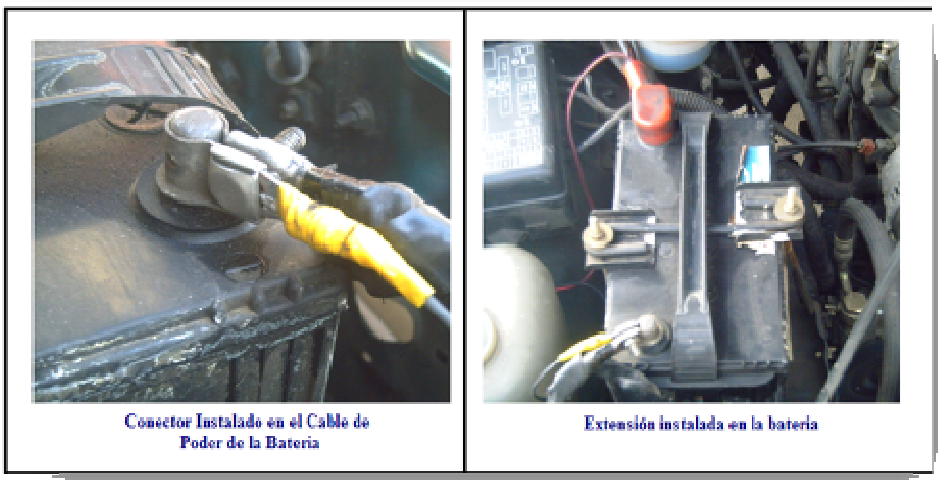


Figura 3.78 "Conexión realizada a la batería del automotor"