



Instituto Tecnológico de Estudios Superiores de Monterrey

Campus Ciudad de México
División de Ingeniería y Arquitectura

Ingeniería Electrónica y Comunicaciones

Departamento de Ingeniería Eléctrica y Electrónica

“Analizador de Impacto”

Autores:

Jazmín Caloca Berumen 922147
José Sergio Aguirre Zurita 704881

Asesor :

Dr. Guillermo Alfonso Parra R.



ITESM
CAMPUS CIUDAD DE MÉXICO
BIBLIOTECA

México D.F. a 12 de mayo de 2003.

ÍNDICE GENERAL.

1 ANTECEDENTES: ACCIDENTES, COLISIONES EN AUTOMÓVILES.....	3
1.1 INTRODUCCIÓN.....	3
1.2 ACCIDENTES DE TRÁNSITO.....	4
1.2.1 <i>Definición</i>	4
1.2.2 <i>Factores que influyen en los accidentes de tránsito</i>	4
1.2.3 <i>Tipos de colisiones</i>	4
1.2.4 <i>Datos relevantes y "La hora de oro"</i>	5
1.3 TIPOS DE COLISIONES.....	7
1.3.1 <i>Fundamentos Teóricos</i>	7
1.4 ORGANISMOS Y NORMAS.....	11
1.4.1 <i>Euro NCAP y la NHTSA</i>	11
1.4.2 <i>Tipos de pruebas</i>	12
2 INVESTIGACIÓN: SENSORES Y OBDII.	15
2.1 INTRODUCCIÓN.....	15
2.2 SENSORES.....	16
2.2.1 <i>DEFINICIÓN</i>	16
2.2.2 <i>ECSU (Electronic Crash Sensor Unit)</i>	16
2.2.3 <i>Galgas o extensiómetros</i>	17
2.2.4 <i>Filamentos de cobre</i>	17
2.2.5 <i>Sensores de velocidad</i>	18
2.2.6 <i>Sensores de contacto</i>	19
2.3 OBD II.....	19
2.3.1 <i>¿QUÉ ES OBD II?</i>	19
2.3.2 <i>¿Cómo funciona el OBD II?</i>	20
2.3.3 <i>Protocolos OBD II</i>	20
2.3.4 <i>CLASIFICACIONES DE SAE</i>	21
2.3.5 <i>El protocolo J1850</i>	22
3 DISEÑO Y DESARROLLO.	31
3.1 INTRODUCCIÓN.....	31
3.2 ORGANIZACIÓN DEL PROYECTO.....	32
3.2.1 <i>Diseño de comunicación con el Bus del vehículo, OBDII</i>	32
3.2.2 <i>Diseño del módulo de adquisición y procesamiento de datos</i>	35
3.2.3 <i>Diseño de Programa</i>	43
3.2.4 <i>Integración del Sistema</i>	54
3.3 DESARROLLO E IMPLEMENTACIÓN.....	54
3.3.1 <i>Introducción</i>	54
3.3.2 <i>Desarrollo de tarjetas</i>	55
3.3.3 <i>Implementación</i>	58
4 RESULTADOS	60
4.1 INTRODUCCIÓN.....	60
4.2 PRUEBAS.....	60
4.2.1 <i>Comunicación con el bus del vehículo</i>	60
4.2.2 <i>Tarjeta del microcontrolador</i>	62
4.2.3 <i>El programa</i>	63
4.3 INTEGRACIÓN DEL SISTEMA.....	73

5 CONCLUSIONES, PERSPECTIVAS, RECOMENDACIONES Y TRABAJO FUTURO.
75

5.1	INTRODUCCIÓN.....	75
5.2	CONCLUSIONES.....	75
5.3	PERSPECTIVAS.....	76
5.4	TRABAJO FUTURO	76

ÍNDICE DE TABLAS.

TABLA 2.3.1,	DEFINICIÓN DE LOS BITS DEL ENCABEZADO.....	24
TABLA 2.3.2,	CONFIGURACIÓN DEL BIT DE NORMALIZACIÓN	28
TABLA 2.3.3,	TIEMPOS DEL PULSO VPW.....	29
TABLA 2.3.4,	PARÁMETROS DE DC DEL PVW.....	29
TABLA 2.3.5,	NIVELES DE VOLTAJE	30
TABLA 3.2.1,	CÓDIGOS DE MODOS DE ACUERDO A SAE.....	34
TABLA 3.2.2	39
TABLA 3.2.3	40

1 Antecedentes: accidentes, colisiones en automóviles.

1.1 Introducción

Los accidentes determinan de manera definitiva el destino de las personas a las que les suceden, jóvenes, adultos y adultos mayores, todos están expuestos a los accidentes cuando conducen un automóvil.

El efecto de los accidentes puede cambiar la vida de los ocupantes del vehículo pues al resultar con lesiones permanentes o con la pérdida de la vida desencadenarían en su familia dolor y cambios importantes.

Muchos automovilistas consideran que por usar un carro lujoso, están exentos de los accidentes, de manera que no toman en cuenta las precauciones (el límite de velocidad, el uso del cinturón de seguridad, el volumen del radio) y se confían, y al sucederles algo como al resto de la gente, pueden morir. De tal forma, se puede pensar, que a pesar de todo, *los accidentes suceden* e inmediatamente a ellos se hace imperativa la llegada de los servicios de urgencia y aseguradoras para beneficio de los automovilistas.

La seguridad y los organismos que la regulan en el mundo, tienen una gran importancia, pues debido a lo que resulta de las pruebas de choque realizadas por los diferentes organismos reguladores de normas de seguridad para automotores, los consumidores pueden darse cuenta de la calidad del producto que adquieren. Si cumplen con los estándares mínimos que han impuesto los organismos de regulación podrán sentirse más seguros, sin que por eso signifique que son inmunes a fallas eventuales, factores ambientales o del conductor.

Debido a lo anterior, en este primer capítulo se presentan algunos datos relevantes sobre los accidentes automovilísticos, como factores que influyen, lo que sucede en la persona al haber una colisión, los tipos de choques "físicos" que hay y sus características, y las pruebas que realizan los organismos de seguridad para dar una calificación de seguridad al automóvil.

1.2 Accidentes de tránsito

1.2.1 Definición

Es el accidente que se produce en una vía abierta a la circulación o tiene su origen en la misma, a consecuencia de la cual una o varias personas resultan muertas o heridas o se producen daños materiales y en el cual al menos un vehículo en movimiento está implicado.¹

1.2.2 Factores que influyen en los accidentes de tránsito

Encontrar la causa de los accidentes de tránsito es una tarea amplia, pero a partir de una investigación dada a conocer por el Instituto Nacional de Medicina Legal y Ciencias Forenses, en Colombia, se han podido determinar algunos patrones globales, entre los que se mencionan los siguientes hechos:

1. Características de persona: tales como edad (jóvenes y ancianos), género (más frecuentes en varones), consumo de alcohol y/o de sustancias psicoactivas previo al hecho, fatiga y sueño, problemas emocionales e inestabilidad afectiva y enfermedades (diabetes, epilepsia, defectos visuales mal corregidos y otros) así como inexperiencia en la conducción.
2. Características del vehículo: relacionadas con el tipo de automotor, sus características estructurales (cinturón de seguridad, apoya cabezas en cada asiento del automóvil, bolsas de aire, almohadillado de ciertas zonas interiores, defectos en neumáticos, frenos o en luces entre otros), y su antigüedad.
3. Características ambientales: como el estado de la vía (deficiente pavimentación, obstáculos próximos al borde de la vía, falta de señalización, falta de iluminación), del tiempo (lluvia, viento, niebla, congestión) y del lugar (áreas urbana o rural).²

1.2.3 Tipos de colisiones

Para estudiar las causas de las lesiones, un equipo interdisciplinario que incluya patólogos, detectives, ingenieros y expertos en accidentes deben analizar las lesiones y el estado de los vehículos. Esto permite saber qué va a pasar con los cuerpos y cuáles van a ser las lesiones. Por otra parte, cada accidente suministra información a partir de las lesiones, que permitirá a los expertos efectuar sugerencias y recomendaciones que seguirán su camino y (en el mejor de los casos) estas recomendaciones llegarán al fabricante, y este modificará sus diseños de acuerdo a ellas, para mejorar la seguridad. Además, cabe resaltar, que **los accidentes no son accidentales**. No se producen al azar, sino que resultan de una secuencia de eventos que los desencadenan.

¹ Gestal JJ. Accidentes. Medicina preventiva y salud pública. Novena edición. Barcelona: Ediciones Científicas y Técnicas, S.A.;1991.& Instituto Nacional de Medicina Legal y Ciencias Forenses. Comportamiento de las lesiones de causa externa, Colombia 1996. Santafé de Bogotá: Imprenta Nacional de Colombia;1996.

² Gili M, Murillo F, Perea-Milla E. Epidemiología y prevención de los traumatismos. En: Salud Pública. Primera edición. Madrid: McGraw Hill – Interamericana;1998.

A partir del análisis antes mencionado, se encontró que, durante un choque se producen **tres colisiones**

- 1ª La colisión del automóvil contra un objeto.
- 2ª La colisión del cuerpo humano contra las partes del auto.
- 3ª La colisión de los órganos internos con los huesos y paredes del cuerpo.

Esta tercera colisión es la que causa las **verdaderas** y más graves lesiones, incluidas las intromisiones de objetos externos dentro del cuerpo. Las lesiones cerebrales se producen generalmente en esta etapa, por la turbulencia que se produce en el líquido que rodea el cerebro, y por los violentos movimientos de sus partes y golpes por los rebotes contra las paredes de la cavidad craneana.³

1.2.4 Datos relevantes y “La hora de oro”

Los accidentes de tránsito constituyen a la vez, una parte de la vida diaria y un progresivo desastre global que destruye vidas y medios de sustento, obstaculiza el desarrollo y deja a millones en la mayor vulnerabilidad.

En el siglo XIX ocurrió la primera muerte de un peatón, la cual ocurrió en 1896. El primer conductor murió en 1898 y a partir de ahí, los autos han cobrado unos 30 millones de vidas. En 1990 se calculó que los accidentes de tránsito constituyen la novena causa de muerte, matando por lo menos a 500,000 personas por año, aunque algunos calculan que el número de bajas llega al millón y a cerca de 15 millones de heridos.

Al evaluar el impacto a largo plazo de los accidentes y tomando en cuenta el correspondiente número de heridos, un importante estudio de la “carga global de la enfermedad” pronosticó que para el 2020 los choques en carretera alcanzarán el tercer lugar en la tabla de muerte e incapacidad de la población mundial, apenas detrás de las depresiones clínicas y las enfermedades cardíacas, pero arriba de las infecciones respiratorias, la tuberculosis, la guerra y el VIH.

Las campañas de seguridad vial, un mejor cuidado del vehículo, la legislación para el uso del cinturón de seguridad y las medidas de dirección del tránsito, han hecho que durante 30 años disminuyan las muertes ocasionadas por accidentes de tránsito, pero a pesar de esto, el índice de muertes per cápita en los EUA, por ejemplo, es de 15.5 por cada 100,000, mientras que en México, es de 5.7 por cada 100,000.

Generalmente, los accidentes en carretera se clasifican como fatales si la víctima muere dentro de los 30 días posteriores al accidente. Haciendo a un lado otros factores, las instalaciones inadecuadas contribuyen a elevar los niveles de mortalidad. Un estudio de TRL indica que los índices de mortalidad en el mundo en desarrollo están asociados a la cantidad per cápita de camas y enfermeras en los hospitales.

³ Segurivial en Argentina: <http://www.segurivial.com.ar/articulos/segurid.htm#inseguros>

Mientras cualquier mejora en las facilidades médicas podría ayudar a reducir el número de personas heridas en accidentes de carretera que posteriormente mueren, algunas necesidades tienen carácter de urgencia, como ocurre a menudo en los desastres de impacto repentino. Las víctimas de los accidentes de tránsito tienen una gran oportunidad de sobrevivir si son atendidos dentro de la “hora de oro” posterior al accidente, y reciben buenos primeros auxilios, más servicios de ambulancia y paramédicos, una opción cara para países en desarrollo.

A medida que los cinturones de seguridad, las bolsas de aire y otras medidas de seguridad reducen las muertes instantáneas y los doctores se vuelven más expertos en salvar vidas, en los países del Norte la atención se está trasladando a las heridas que incapacitan, como aquellas en piernas y pies. Para apoyar la investigación se ha estado utilizando la simulación de accidentes por computadora; por ejemplo, un trabajo efectuado por el Centro Nacional de Análisis de Choques, de la Universidad George Washington, indica que un relleno en los pisos de los vehículos podría reducir significativamente las lesiones de las piernas.

Otra investigación de los Estados Unidos muestra factores de riesgo según el tipo de vehículo y la combinación de marcas específicas con accidentes, y revela no solamente el mayor peligro que representan ciertos autos deportivos, sino también cómo es que los vehículos más grandes y pesados son más seguros, pero únicamente para sus ocupantes. Los vehículos más pequeños y livianos, y quienes viajan en ellos, se ven mucho más afectados al chocar con vehículos más grandes; resultado infortunado éste, dada la presión existente a favor de una mayor eficiencia en el consumo de combustible y una y menor contaminación.

Los beneficios de un vehículo mejor diseñado y el uso de los cinturones de seguridad, parece ser similar en diferentes países. Se esperan muchos más logros, como las bolsas de aire para impactos de costado y las de tipo “roll-over”. Sin embargo, estos avances podrían tener poca repercusión en la tasa total de mortalidad de los países en desarrollo, ya que según sean las leyes del país importador, puede ser que los accesorios de seguridad, como por ejemplo los cinturones, no sean instalados en los vehículos que se venden en los países en desarrollo.

Algunos estudios sugieren que las medidas de seguridad, desde los cinturones hasta el diseño de los vehículos y la construcción de carreteras, llevan a las personas a compensar corriendo más riesgos, conduciendo más rápido y yendo a corta distancia de los otros autos.

Uno de los factores clave en tal comportamiento es que la mayoría de los motociclistas son hombres, quienes tienen los índices más altos en todas las formas de acciones riesgosas, accidentes y muertes violentas. Otro factor es la ilusión de control en el ambiente cada vez más envolvente del auto, con aire acondicionado, música y asientos confortables; los pasajeros de trenes, botes o aviones, no tomarían el mismo nivel de riesgo.

Desde que el mundo en desarrollo tiene una mayor proporción de usuarios de carreteras vulnerables que todavía se desplazan por ellos, esto podría indicar que una mayor “seguridad” de los vehículos podría tener poco impacto - o potencialmente podría hacer que hasta aumenten las cantidades totales de muertes y heridos.

Los teléfonos portátiles y otras formas de comunicación inalámbrica podrán ofrecer la notificación automática de colisión, contacto inmediato con los servicios de emergencia, información dentro del vehículo acerca de las condiciones del camino, así como una mayor seguridad para el conductor y la posibilidad de reportar sobre congestionamientos, choques y conductores ebrios.

Conforme el mundo ingresa a su segundo siglo con autos, los expertos en transportes y seguridad ven muchas formas de que la curva creciente del costo humano y financiero del tránsito pueda por lo menos hacerse más lenta, tales como el empleo de paneles frontales que sean "amigables con los peatones", renovar la capacitación de los conductores después de un accidente, el traslado de los usuarios de carretera a las crecientes áreas libres de tránsito de la ciudad, producir mejores materiales educativos y hasta incentivar a los conductores que no hayan chocado.

Para las agencias internacionales, el imperativo de acción también está claro: en la víspera de un crecimiento masivo del tránsito en el mundo en desarrollo, los choques ya le cuestan a los países del Sur casi tanto como toda la ayuda que reciben. Los accidentes de tránsito perjudican el progreso, al matar o herir a las personas económicamente activas, al buscar a los más vulnerables y, según se pronostica, al hacer más daño con las muertes e incapacidades que producen, que muchas de las amenazas a la salud que actualmente reciben asistencia prioritaria.⁴

1.3 Tipos de colisiones.

1.3.1 Fundamentos Teóricos

Un *choque ó impacto* ocurre cuando dos cuerpos chocan (colisión), uno contra otro, durante un intervalo de tiempo muy corto, haciendo que se ejerzan fuerzas (impulsivas) relativamente grandes entre los cuerpos. El vector normal común a la superficie de contacto durante el choque es la *línea de impacto*. Por ejemplo, el choque de un martillo contra un clavo o el de un palo de golf contra la pelota.

En general, hay dos tipos de impacto o choque:

Central : ocurre cuando la dirección del movimiento de los centros de masa de las dos partículas que chocan es a lo largo de la *línea de impacto* (Figura 1.3.1)

-
- ⁴ Jones-Lee, M.W. et al. *The value of preventing non-fatal road injuries: Findings of a willingness to pay national sample survey*. TRL CR330, Transport Research Laboratory, 1993.
 - *Scientific American*. "The Future of Transportation". October 1997 special issue.
 - World Health Organization. *World Health Statistics*. Geneva: WHO, 1990, 1993. ISBN: 92-4-067930-8; ISSN: 0250-3794
 - International Road Federation: <http://web.eunet.ch/irf>
 - *Scientific American*: <http://www.sciam.com>
 - <http://www.disaster.info.desastres.net/federacion/imd98/capitulo2.htm>

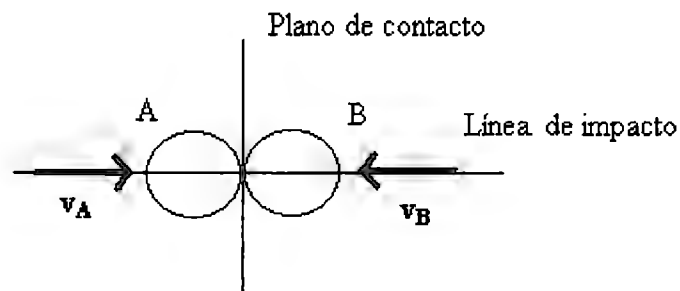


Figura 1.3.1

Oblicuo : ocurre cuando el movimiento de una ó de ambas partículas es según cierto ángulo con la *línea de impacto* (Figura 1.3.2).

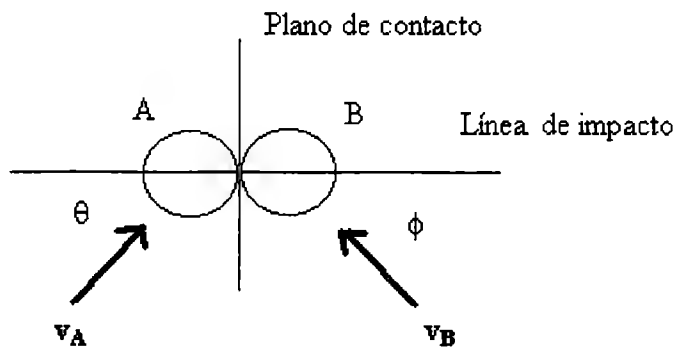


Figura 1.3.2

Tanto en uno como en el otro pueden ocurrir dos casos: que después del choque no se haya producido ninguna deformación permanente en los cuerpos que chocan; en este caso el choque se llama *perfectamente elástico*; y que se produzca una *deformación permanente*, entonces se llama *inelástico*.

Cualquiera que sea el tipo de choque, siempre se conserva el momento lineal del sistema, ya que el choque consiste en una interacción entre los cuerpos que chocan. Estas interacciones son fuerzas internas del sistema.

En el choque perfectamente elástico, además del momento lineal, se conserva también la energía cinética.

En el choque inelástico solamente se conserva el momento lineal. Parte de la energía cinética se convierte en el trabajo necesario para producir la deformación.

Impacto ó choque central.- Sean dos partículas A y B, de masa m_A y m_B que se mueven sobre la misma recta con las velocidades conocidas $(v_A)_1$ y $(v_B)_1$. Si $(v_A)_1$ es mayor que $(v_B)_1$, A acabará chocando con B. A consecuencia del choque, las dos partículas se *deforman*, y al final del *periodo de deformación*, las dos tienen la misma velocidad v . Sigue a ello el *periodo de restitución*, al final del cual, en función de las intensidades de las fuerzas de choque y de las características de los materiales, las partículas recobrarán sus formas originales o quedarán deformadas permanentemente.

Para ilustrar el método de analizar la mecánica del choque, consideremos el caso del choque central de las partículas A y B (Figura 1.3.3, Figura 1.3.4, Figura 1.3.5, Figura 1.3.6, Figura 1.3.7)

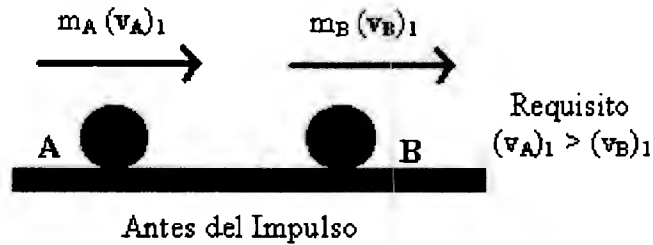


Figura 1.3.3

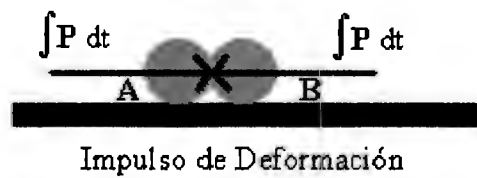


Figura 1.3.4

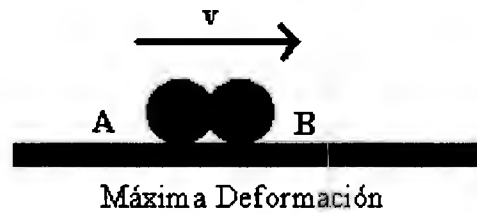


Figura 1.3.5

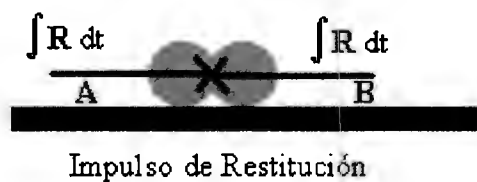


Figura 1.3.6

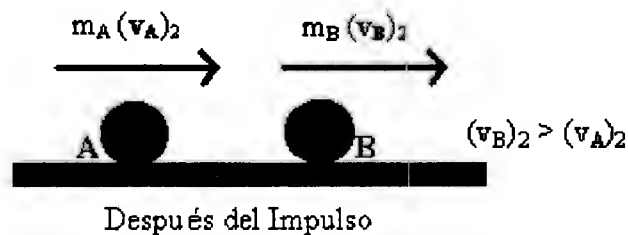


Figura 1.3.7

1. Las partículas tienen los momentos lineales iniciales $m_A (v_A)_1$ y $m_B (v_B)_1$ (Figura 1.3.3). Si $(v_A)_1 > (v_B)_1$ el choque ocurrirá eventualmente.
2. Durante el choque, las partículas experimentan un periodo de deformación de tal manera que ejercen un impulso de deformación igual pero opuesto una sobre la otra (Figura 1.3.4)

$$\int P dt$$

Ecuación 1.3-1

3. Solamente en el instante de máxima deformación ambas partículas se moverán con una velocidad común v (Figura 1.3.5)
4. Después ocurre un periodo de restitución, en cuyo caso, ya sea que las partículas regresarán a su forma original, ó se mantendrán permanentemente deformadas. El impulso de restitución

$$\int R dt$$

Ecuación 1.3-2

Igual pero opuesto, empuja a las partículas por separado, una de la otra (Figura 1.3.6).

Realmente, las propiedades físicas de dos cuerpos cualesquiera son tales que el impulso de deformación siempre es mayor que el impulso de restitución; es decir

$$\int P dt > \int R dt$$

5. Justo después de la separación las partículas tendrán las cantidades de movimiento $m_A (v_A)_2$ y $m_B (v_B)_2$ (Figura 1.3.7), donde $(v_B)_2 > (v_A)_2$.

En la mayoría de los problemas se conocen las velocidades iniciales de las partículas y es necesario determinar sus velocidades finales $(v_A)_2$ y $(v_B)_2$. Respecto a esto, el momento para el sistema de partículas se conserva, ya que durante el choque los impulsos internos de deformación y restitución se cancelan. De la (Figura 1.3.3) y (Figura 1.3.7), tenemos

$$m_A (v_A)_1 + m_B (v_B)_1 = m_A (v_A)_2 + m_B (v_B)_2$$

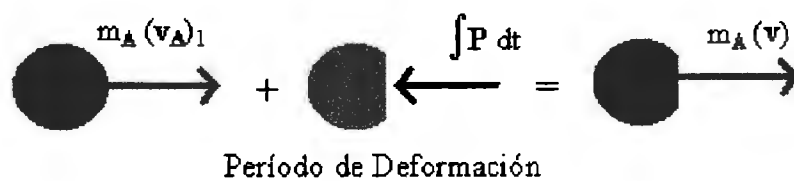


Figura 1.3.8

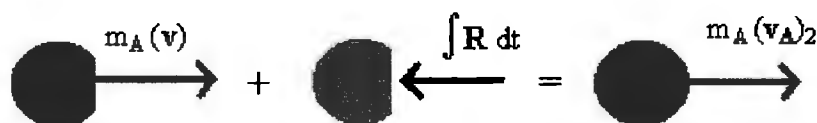


Figura 1.3.9

Para obtener una segunda ecuación, necesaria para despejar $(v_B)_2$ y $(v_A)_2$, debemos aplicar el principio del impulso y momento a cada partícula. Para ello, consideramos el movimiento de la partícula A durante el periodo de deformación y aplicamos el teorema del momento lineal. Como la única fuerza impulsiva que actúa sobre A durante ese periodo es la fuerza P ejercida por B (Figura 1.3.8), podemos escribir

$$m_A (v_A)_1 - \int P dt = m_A (v)$$

donde la integral comprende todo el periodo de deformación. Considerando ahora el movimiento de A durante el periodo de restitución, y representando por R la fuerza ejercida por B sobre A durante ese periodo (Figura 1.3.9), podemos escribir

$$m_A v - \int R dt = m_A (v_A)_2$$

En general, la fuerza R ejercida sobre A durante el periodo de restitución difiere de la fuerza P ejercida durante el periodo de deformación y el valor

$$\int R dt$$

de su impulso es menor que el valor

$$\int P dt$$

del impulso de P.

El cociente entre los módulos de los impulsos correspondientes, respectivamente, a los periodos de restitución y de deformación recibe el nombre de **coeficiente de restitución**, y se representa por e.⁵

1.4 Organismos y Normas

1.4.1 Euro NCAP y la NHTSA

La Euro NCAP provee a los consumidores del mercado automovilístico asesoría del rendimiento de la seguridad en algunos de los más populares autos que se venden en Europa.

Establecida en 1997, esta organización está ahora respaldada por cinco gobiernos europeos, la Comisión Europea y organizaciones automovilísticas y de consumidores que están en cada país de la Unión Europea. Euro NCAP se ha convertido de forma rápida en un catalizador para fomentar significativamente las mejoras en materia de seguridad dentro de los nuevos diseños automotrices.

Las pruebas de este organismo, aunque muy rigurosas, tienen un inconveniente: no contemplan muchos de los autos que se venden en México. Por eso, también se recurrió

⁵ Grupo de investigación español http://www.uco.es/investiga/grupos/labvirtual/tutoriales/choques/central/ce_elastico.htm

a la NHTSA (*National Highway Traffic Safety Administration*), el organismo que regula la seguridad en las carreteras de Estados Unidos para tratar de completar el panorama y poder presentar una visión acerca de los resultados de los autos en México en las pruebas a los choques más frecuentes. Casi ninguno compite en ambas, pues en Europa no se prueban coches que no sean fabricados en ése continente –pero los resultados que ofrecen son más generales, y en los Estados Unidos los resultados no ofrecen un panorama general, sino que otorgan una calificación a cada prueba.

Las legislaciones otorgan estatutos mínimos que sirven como un estándar para marcar la seguridad de los nuevos coches. Es el objetivo de Euro NCAP alentar a los fabricantes a exceder estos requerimientos mínimos.

1.4.2 Tipos de pruebas

A continuación se presentan los impactos evaluados por estos organismos:

Impacto frontal

Se realiza a 64 km/h contra una barrera deformable situada con un desplazamiento del 40% sobre el frontal total del vehículo. **El choque simula un impacto entre dos coches que circulan en sentido contrario a 55 km/h.**

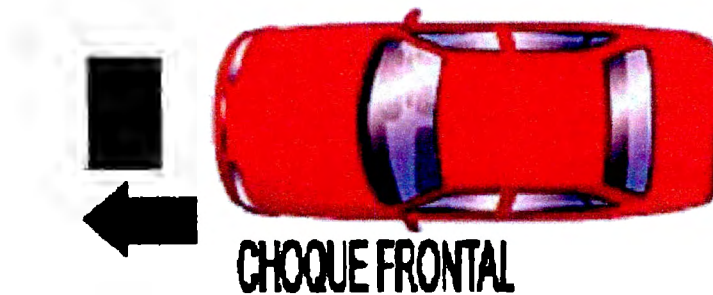


Figura 1.4.1

Impacto lateral

Una barrera deformable choca a una velocidad de 50 km/h contra la puerta del conductor. El muro va montado sobre un carro que avanza de forma transversal sobre el vehículo. El punto del choque donde impacta el punto medio de la barrera es aquel en el que, estaría situada la cadera de un conductor varón en el 95% de los casos.

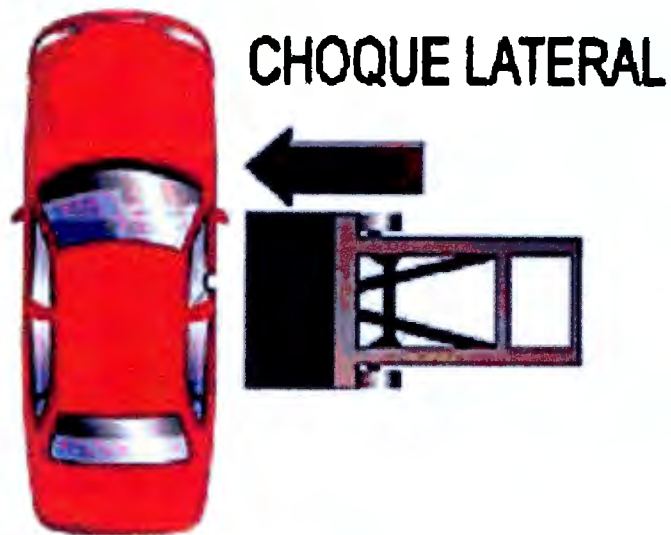


Figura 1.4.2

Impacto contra un peatón

El ensayo evalúa el impacto de la pierna contra la defensa, del muslo contra el borde del cofre y de la cabeza contra este mismo. **Los tres simulan un peatón que es impactado por el vehículo a 40 km/h.** Para la cabeza, la prueba se subdivide en dos: uno representado a un adulto y otro a un niño.



Figura 1.4.3

Prueba del poste

Las formas de accidentes varían de país a país. En Europa, aproximadamente la cuarta parte del total de los daños fatales ocurren en colisiones laterales. Varias de estas lesiones suceden cuando **un auto impacta lateralmente a otro**, pero en algunos países como Alemania es más común que este tipo de accidentes sucedan **cuando un auto golpea un poste o un árbol.**⁶

⁶ Euroncap: http://www.euroncap.com/tests_side_impact.htm

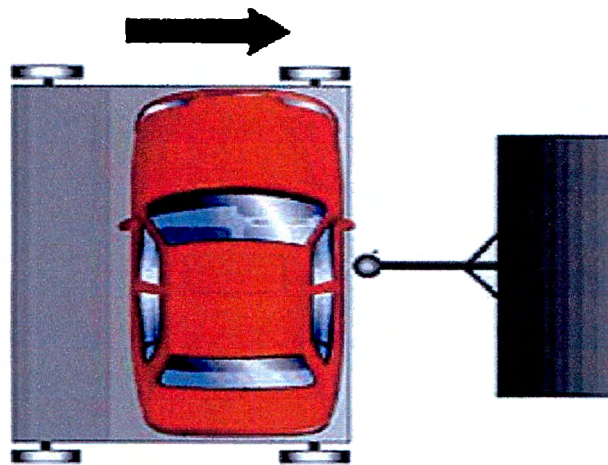


Figura 1.4.4

2 Investigación: sensores y OBDII.

2.1 Introducción

En el presente capítulo que corresponde a desarrollo de la base teórica, incluye la investigación de los diferentes tipos de sensores que se consideraron para el proyecto, de los cuales algunos se descartaron por diferentes motivos, entre los cuales se menciona, el exceder las necesidades del proyecto, no cumplir con el objetivo, el costo. Sin embargo, el resto se usó y son los elementos con base en los cuales se realizó el diseño del sistema.

Los sensores son uno de los pilares importantes del proyecto ya que es por medio de estos que se determinarán datos de ayuda con respecto a la colisión, dónde comenzó dicha colisión y cuánto daño causó al auto.

Otro punto muy importante que se realizó en este capítulo es el que corresponde a la investigación del puerto del automóvil con estándar OBDII que sirvió en el desarrollo del proyecto para tomar la velocidad del carro. Aquí se determinó qué es, cómo funciona, las clasificaciones, los protocolos que existen según la marca del vehículo y las características de funcionamiento del protocolo que se utiliza en el proyecto.

Este capítulo es de vital importancia pues en él, están siendo evaluados los elementos del sistema y con esto se pretende garantizar que el proyecto funcionará de la manera más sencilla y económica pero perfectamente capaz de cumplir con los objetivos del proyecto. De esta manera se garantiza que sea viable una implementación total por parte de una aseguradora, sin que se eleven de forma importante los costos y pueda brindarse el servicio desde pólizas bajas.

2.2 Sensores

2.2.1 DEFINICIÓN

La medición de magnitudes mecánicas, térmicas, eléctricas y químicas se realiza empleando dispositivos denominados sensores y transductores. El sensor es sensible a los cambios de la magnitud a medir, como una temperatura, una posición o una concentración química. El transductor convierte estas mediciones en señales eléctricas, que pueden alimentar a instrumentos de lectura, registro o control de las magnitudes medidas. Los sensores y transductores pueden funcionar en ubicaciones alejadas del observador, así como en entornos inadecuados o impracticables para los seres humanos.⁷

A continuación se numeran algunos de los sensores presentes en el mercado:

2.2.2 ECSU (Electronic Crash Sensor Unit)

Ventajas

Es un sensor que se usa para detectar choques y brindar protección, que tiene características muy aproximadas a los que se busca, pues brinda memoria de 60 segundos, sobre la aceleración que tuvo el auto antes del choque, se le pueden configurar los cambios en la aceleración y el frenado.

Desventajas

El kit tiene más de lo que se necesita, y además es costoso. Se desperdiciarían algunas de sus funciones, haciendo que no fuera costeable (a menos que se logrará la implementación desde el ensamblaje y este kit tuviera varias funciones)

Fabricante:

- ◆ Safety System Inc.

Características principales:

- ◆ Voltajes de alimentación: 12 a 30 VDC
- ◆ Corriente de -125 mA, de entrada
- ◆ 6 puertos de salida para bolsas de aire
- ◆ Capacidad de programación de cambios en aceleración y frenado
- ◆ Opera normalmente hasta 30 segundos después de haber perdido alimentación
- ◆ Graba hasta 60 segundos de muestras en cada evento a una tasa de 2000 muestras por segundo.
- ◆ 6 salidas (2 grupos de 3) con corriente de 6 amp. mínimo a una carga de 1 ohm durante 15 segundos.

⁷"Electrónica", Enciclopedia Microsoft® Encarta® 98 © 1993-1997 Microsoft Corporation. Reservados todos los derechos.

2.2.3 Galgas o extensiómetros.

Las galgas se utilizan para medir deformaciones lineales o esfuerzos en los materiales.

Ventajas

Los extensiómetros son dispositivos pasivos que funcionan cambiando la resistencia al deformar el material sobre el que están colocados y que al pasar un voltaje por ellos y al deformar la superficie de donde se encuentran dan un cambio en la resistencia que tienen. Para detectar cambios horizontales y verticales se necesita de dos, una que detecte los que son horizontales y otra los verticales, ya que estos dispositivos solo detectan cambios en una sola dimensión.

Desventajas

Las galgas o extensiómetros fueron desechados pues lo que miden es el grado de vibración que tiene un material, es decir la resistencia que tiene el material ante deformaciones.

Lo que se pretende medir es el estado en el que se encuentra el carro, es decir, que deformaciones tuvo y en qué parte y no la vibración (o microdeformación) que tuvo en cierto metal o lugar del carro, así que sería muy difícil establecer parámetros que marcaran cuando se llegó al límite de deformación y se tiene un daño notable.

Además, el costo de los extensiómetros varía entre \$200 pesos y \$1800 pesos por lo que no es viable, ya que se necesitarían de más de 2 y no aportan la información que se necesita.

Fabricantes:

- ◆ Advanced Custom Sensors, Inc.
- ◆ Mason Industries, Inc

Características principales:

- ◆ Voltajes de excitación (Vdc y Vac rms) 10 nominal, 15 máximo
- ◆ Resistencia de entrada (ohms) 1200 ±300
- ◆ Resistencia de salida (ohms) 1200 ±300

2.2.4 Filamentos de cobre.

Los filamentos de cobre, no son como tal un sensor, son hilos de cobre que tienen conductividad y cierta elasticidad.

Ventajas

Los filamentos son una opción barata, los hilos se deben colocar en principio sobre el chasis, motor y parabrisas, haciendo un circuito en cada uno de ellos, estos circuitos en todo momento tendrían un voltaje y solo al romper el filamento se rompería la conductividad.

La apertura del circuito es lo que ayudaría a detectar el estado del auto.

Desventajas

La mano de obra y el pegamento son dos cosas que se tendrían que buscar que fueran óptimas si se desea usar esta opción pues si el circuito está mal colocado o se despega con facilidad podría reportar datos incorrectos.

El costo de la mano de obra, por otro lado sería de importancia para que no se elevaran los costos generales de esta opción.

Proveedor

- ◆ Condulimex

Características principales

- ◆ Flexibilidad, adherencia y resistencia a la abrasión excelentes.
- ◆ Buena estabilidad química.
- ◆ Excelentes propiedades dieléctricas.
- ◆ Resistencia al gas freón 12 y 22

2.2.5 Sensores de velocidad.

Dentro de los sensores de velocidad, destacan 2 principales tipos:

- ◆ Los magnéticos. Generalmente se instalan en la transmisión o en las ruedas y se generan pulsos proporcionales a la velocidad del vehículo.
- ◆ Los ópticos. Se pueden colocar en cualquier parte que determine el movimiento del automóvil y tiene un mucho mayor margen de error, la señal puede o no venir en forma de pulsos dependiendo del modelo.

Las maneras de tomar la medición:

- ◆ Utilizar un sensor externo al automóvil.
- ◆ Acondicionar la señal del sensor de velocidad que ya viene incluido en el vehículo.

Ventajas

El sensor de giro de las llantas, proporciona a los sistemas auxiliares como el ABS y ASR, o GPS datos sobre velocidad de las ruedas para el cálculo de múltiples valores.

Desventajas

Podría ser difícil de colocar, o de acondicionar, ya que es probable que el dueño pierda la garantía del vehículo. El precio varía entre \$650 y \$900 moneda nacional, lo que aumentaría considerablemente el costo del sistema.

Proveedores:

- ◆ Quasarvigo, Inc.
- ◆ General Motors

- ◆ Siemens
- ◆ Toyota

2.2.6 Sensores de contacto.

Es un interruptor normalmente cerrado, barato, que puede ser calibrado de tal forma que al presionarse hasta el fondo se mande una señal de que ha sido accionado pues el circuito presentará discontinuidad. Este interruptor es comúnmente usado en el pedal de freno para accionar la luz trasera de los autos.

Ventajas

El precio de este interruptor es barato, se puede colocar en un lugar rígido para que garantice que sólo después de una colisión se activará. Existe un interruptor universal

Desventajas

Puede ser difícil de conectar y garantizar el buen funcionamiento del circuito todo el tiempo.

Fabricantes:

- ◆ 4B
- ◆ Ross- Tech

Características principales:

- ◆ Voltajes de excitación (Vdc y Vac rms) 12 - 240volts
- ◆ Resistencia de entrada (ohms) .04

2.3 OBD II

2.3.1 ¿QUÉ ES OBD II?

“Sistema de diagnósticos a bordo de segunda generación”, por sus siglas en inglés, es un sistema computarizado de diagnósticos suministrado en vehículos de 1996 y más modernos de origen americano, asiático o europeo vendidos en los Estados Unidos. Fue adoptado en 1996 como parte de un mandato del gobierno para reducir las emisiones de contaminantes por los vehículos. Básicamente, el OBD II es un programa sofisticado que corre en la computadora principal del vehículo. Está diseñado específicamente para detectar fallas en los diferentes sistemas del vehículo (inyección de combustible, encendido, catalizador, sistema de recirculación de gases de escape, etc.). Si se descubre un problema, la computadora ilumina la luz MIL (rotulada “Check Engine”) para avisarle al conductor y fija un código de diagnóstico de problema (DTC) para identificar en cuál sistema o circuito ocurrió el problema. Una herramienta de diagnóstico especial, es

necesaria para extraer esta información de diagnóstico con el objeto de reparar correctamente los vehículos de 1996 ó más modernos.⁸

2.3.2 ¿Cómo funciona el OBD II?

Éste es un conector que se encuentra en alguna parte del tablero, dependiendo del modelo y fabricante del vehículo, el conector tiene la siguiente forma:

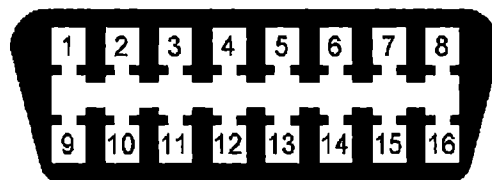


Figura 2.3.1
Conector OBD II

Donde los conectores tienen el siguiente orden:

- Pin 2 - J1850 Bus+
- Pin 4 – Tierra del Chassis
- Pin 5 – Tierra de la Señal
- Pin 6 - CAN High (J-2284)
- Pin 7 - ISO 9141-2 K Line
- Pin 10 - J1850 Bus
- Pin 14 - CAN Low (J-2284)
- Pin 15 - ISO 9141-2 L Line
- Pin 16 – Voltaje de la Batería

2.3.3 Protocolos OBD II

Se puede hacer notar que existen 3 variantes del OBD II, es decir, se manejan 3 protocolos de comunicación distintos:

- SAE J1850 VPW (Variable Pulse Width Modulation)
- ISO 9141
- SAE J1850 PWM (Pulse Width Modulation)

Donde el primero es utilizado por General Motors y Pick Ups, el segundo por la Chrysler, todos los vehículos Europeos y casi todos los Asiáticos, y el último es usado por la Ford.

⁸ Innova Electronics Corporation: http://www.iequus.com/pics/3100_Flyer_Spanish.pdf

Los dispositivos que leen esta información se llaman Scanners y tienen diferentes costos y capacidades, ya que la información de cada protocolo es entregada en diferentes pines del conector OBDII:

J1850 VPW—El conector debe tener contacto metálico en los pines 2, 4, 5, y 16, pero no con 10.

ISO 9141-2— El conector debe tener contacto metálico en los pines 4, 5, 7, 15, y 16.

J1850 PWM—El conector debe tener contacto metálico en pines 2, 4, 5, 10, y 16.

Esto depende del fabricante del vehículo, en caso de tener contactos metálicos en los pines correspondientes a los tres protocolos, entonces es probable que se utilice el estándar SAE J1979.

Con lo cual, se necesita saber cómo funcionan los tres protocolos para poder obtener la información que se va a requerir para el proyecto

2.3.3.1 SAE J1850 VPW:

Este protocolo utiliza una arquitectura abierta, de bajo costo y topología de bus de un solo nivel. Este protocolo soporta dos alternativas:

Método PWM de 41.6 Kb/s

Método VPW de 10.4 Kb/s

En el caso del primer método, utiliza modulación VPM (Variable Pulse Width Modulation), con frames, símbolos, capa física, etc. de J1850.

2.3.4 CLASIFICACIONES DE SAE

SAE ha definido tres clasificaciones distintas de protocolos, Clase A, Clase B y Clase C. La Clase A es la primer clasificación y es la que maneja la menor tasa de transmisión a 10 Kb/s, esta clase soporta actuadores y pequeños sensores.

La segunda clase de acuerdo a la clasificación de SAE es la Clase B. Ésta soporta tasas de hasta 100 Kp/s y típicamente soporta intermódulos, control en tiempo no real y comunicaciones. La utilización de la Clase B puede eliminar sensores redundantes y otros elementos de sistema proveyendo un significado a la transferencia de datos entre nodos.

La tercer Clase, o Clase C tiene capacidad de transmisión de datos hasta de 1 Mb/s. Esta clase es utilizada para aplicaciones críticas, de control en tiempo real, y facilita el control distribuido vía transmisiones de alta velocidad conectados a sistemas de control.

2.3.5 El protocolo J1850

CLASE A:

El protocolo J1850 es una red de comunicación de datos que comparte información paramétrica en frames (mensajes) entre los módulos electrónicos conectados al bus de datos.

La codificación de Ancho de Pulso Variable (Variable Pulse Width Modulation) es un esquema que trabaja a 10.4 Kb/s. Esta codificación utiliza comunicación que depende de símbolos dependientes del tiempo. Por ejemplo, un 1 binario no es necesariamente un alto en bus. En VPW, un uno binario es un símbolo que denota una transición de un potencial bajo a uno alto, o viceversa; pero el símbolo nos indica si es un estado alto o bajo por el tiempo que se mantiene en esa posición, por ejemplo si un estado alto se mantiene por 64 microsegundos, entonces es un uno.

Este protocolo utiliza dos estados:

- Dominante, cuando el voltaje en el bus de datos es mayor a 4.25 V y hasta 20 V
- Pasivo, cuando el voltaje incluyendo ruido están debajo de los 3.5 V

EL CONCEPTO DE ARBITRAJE ES:

El estándar SAE J1850 ofrece una asignación de red para cada usuario basada en un concepto llamado arbitraje. Este concepto es un proceso para determinar cuál de los dos o más nodos puede continuar transmitiendo cuando ambos o más nodos comienzan a transmitir de forma simultánea. El bus J1850 es asíncrono, sin maestro y peer-to-peer, que ofrece acceso a la red equivalente para cualquier nodo. Un atributo importante del bus J1850 es que un nodo transmite tipo punto-multipunto o broadcast. Esto significa que no sólo los otros reciben su mensaje, sino que el mismo nodo ve su propio mensaje de regreso. Todos los mensajes son asíncronos por naturaleza. Siendo asíncronos, se fuerza a cualquier dispositivo que envía mensajes a determinar cuándo puede ser iniciada la transmisión de dicho mensaje, y realizada sin secuencias predefinidas entre los mensajes.

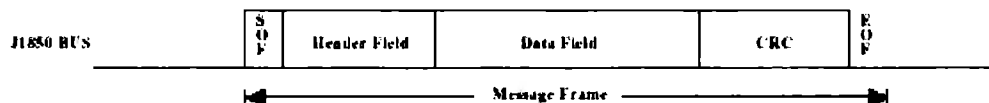


Figura 2.3.2, Arbitraje

El protocolo J1850 soporta arbitraje CSMA/CR, el cual es de tipo "non-colliding" que soporta enlaces sin maestro. Antes de que algún nodo trate de enviar algún dato, primero escucha el bus, por un pequeño periodo de tiempo. Si el bus está ocupado, entonces el nodo espera hasta que el mensaje actual esté completo para volver a intentar enviar sus datos. A este proceso de escuchar antes de enviar, se le llama Sensado de Portadora (Carrier Sense – CS-). Como el protocolo J1850 es peer-to-peer, se ofrece un acceso a la red de forma equivalente a cada nodo, lo que se llama Acceso Múltiple (Multiple Access -MA-). Acceso

Múltiple significa que más de un nodo puede comenzar a transmitir al mismo tiempo, y se resuelve el problema de quién tiene el control de bus a través de la priorización de los mensajes de un símbolo activo sobre uno pasivo.

Cuando los nodos transmiten un símbolo pasivo pero aparece un símbolo activo transmitido en el bus, entonces hay otro nodo que está transmitiendo y tiene mayor prioridad. Cualquier nodo que transmite un símbolo pasivo al bus J1850, pero observa a un símbolo activo, pierde el arbitraje y cambia su función de transmisor a receptor. El nodo que gana el arbitraje sigue transmitiendo, verificando cada bit del mensaje y eliminando cuando sea necesario hasta que un solo nodo quede. Cuando el chequeo está listo para cada bit, se le llama arbitraje bit a bit.

La siguiente figura muestra cómo funciona el bus del J1850:

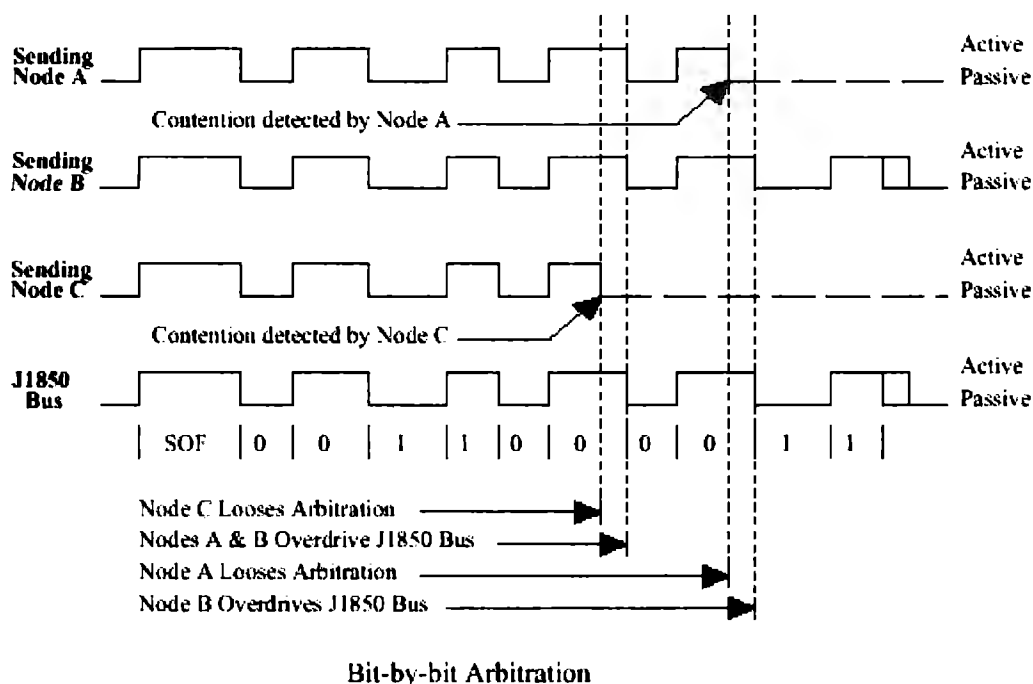


Figura 2.3.3, Arbitraje bit a bit

Porque el nodo sobreviviente siempre se queda con el control del bus con símbolos activos, cuando los otros nodos enviaron símbolos pasivos, la integridad del mensaje queda intacto. Los demás nodos que perdieron el arbitraje intentarán poner otra vez sus mensajes una vez que el mensaje en curso termina, haciendo arbitraje para tener el control del bus. Los mensajes con prioridad más alta serán los que contengan más bits activos al inicio de su mensaje que otros nodos. Una opción inherente con CSMA/CR es aquella que no es posible tener retardos arreglados entre los diferentes tramas de mensajes. Sin embargo, este método de no corrupción de colisiones, o CSMA/CR no es correcta en la base del protocolo J1850.

INICIO DEL FRAME (SOF –START OF FRAME-)

Recordando que un bus J1850 libre es tirado a tierra. El inicio de cualquier trama de mensaje comienza con el periodo predefinido llamado Inicio de Trama (SOF). El periodo SOF es crucial para sensar la portadora y para propósitos de arbitraje. Un símbolo SOF se define como "potencial alto" en el bus J1850 para un periodo nominal de 200µs. Después del símbolo SOF, los bits símbolos representan bytes de datos que son transmitidos. Cualquier dato desde un byte hasta siete bytes pueden ser transmitidos.

EL CAMPO DEL ENCABEZADO:

El primer byte está designado para ser el campo de encabezado. Sin embargo, el campo de encabezado puede ser modificado para ser de uno o tres bytes de longitud. El campo de encabezado contiene información crucial acerca de qué debería esperar el nodo receptor en el mensaje que le precede. Por ejemplo, de cuántos bytes será el encabezado y cuántos bytes de datos contiene el campo de datos que se envía en particular. Los bytes de datos son referenciados como el campo de datos. El campo de mensaje dinámico VPW J1850 puede contener desde uno hasta 7 bytes de datos. Ambos, el campo de encabezado y el de datos deben contenerse dentro del límite de uno hasta 7 bytes de datos de la trama del mensaje VPW J1850.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P ₂	P ₁	P ₀	H-bit	K-bit	Y-bit	Z ₁	Z ₀

Figura 2.3.4, byte de encabezado

Bit	Meaning	Value	Meaning
PPP	Message Priority	000	Highest Priority
		111	Lowest Priority
H	Header Type	0	Three Byte Header
		1	Single Byte Header
K	In-Frame Response (IFR)	0	IFR Required
		1	IFR Not Allowed
Y	Addressing Mode	0	Functional Addressing
		1	Physical Addressing
ZZ	Specific Message Type	00	The meaning for these values are dependent on K & Y above. Refer to SAE J2178/1 document for more details.
		01	
		10	
		11	

Tabla 2.3.1, Definición de los bits del encabezado

El byte completo es utilizado como identificador del mensaje con un solo byte de encabezado. Los tres primeros bits más significativos son prioritarios. Estos tres bits son capaces de designar ocho niveles de prioridad, con un 000 binario se define la máxima prioridad y con un 111 binario, la prioridad más baja. Estos tres bits sirven fácilmente al proceso de arbitraje sin tener que influenciar otros bits del encabezado con otras asignaciones.

Siguiendo los bits prioritarios del encabezado, o el bit H. Este bit indica si este encabezado es de 1 o tres bytes. Si el bit H es puesto a "cero", entonces se espera un encabezado de

tres bytes. Si el bit H se pone en "uno", entonces el encabezado sólo contendrá un solo byte.

El siguiente bit del encabezado, designa si In-Frame Response (IFR), es requerido o no. Este bit es conocido como el bit K. Si el bit K es puesto a "cero", entonces un IFR es requerido. Si el bit K es puesto a "uno", entonces el IFR no es requerido. Más adelante se explica mejor este bit.

Siguiendo con el bit K, este bit es utilizado para designar el tipo de direccionamiento. Este bit es llamado el bit Y. Existen dos tipos de direccionamiento que son acomodados con el protocolo J1850. Uno es un tipo funcional de direccionamiento, el otro es un tipo de direccionamiento físico. El resto del mensaje es interpretado dependiendo del cómo es puesto este bit. Si el bit Y es puesto a "cero", entonces el direccionamiento funcional será utilizado. Si el bit Y es puesto a "uno", entonces el direccionamiento físico será utilizado.

El direccionamiento funcional es un esquema de direccionamiento que etiqueta los mensajes basados en su código de operación o contenido. El direccionamiento físico es un esquema que etiqueta los mensajes de acuerdo a su dirección física de acuerdo a su fuente y/o destino(s). El direccionamiento físico es independiente de la localización geográfica, el conector, y/o la asignación de identificación del cable. Para más detalles del funcionamiento de estos tipos de direccionamiento, es necesario referirse a los documentos 1 a 4 de SAE J2178.

Los últimos dos bits del encabezado simple o de un byte son los bits de tipo de mensaje, o bits ZZ. Estos dos bits le dicen a cualquier nodo(s) receptor qué formato tomará el resto del mensaje. Estos dos bits, combinados con el bit Y y el bit K, ofrecen hasta diez y seis combinaciones diferentes tipos de mensajes que pueden ser transmitidos vía el encabezado de un solo byte. El bit Z1 ayuda a designar si el mensaje usa direccionamiento extendido o no. Si el bit Z1 es puesto a "uno", entonces el(los) nodo(s) receptor(es) saben que el quinto byte del mensaje en cuestión es la dirección extendida de esta función. El bit Z0 indica si hay datos contenidos en el mensaje o no. Si el bit Z0 es puesto a "cero", entonces se esperan datos en el mensaje. Si el bit Z0 es puesto a "uno", entonces no hay datos contenidos en el mensaje.

Un encabezado de tres bytes es casi lo mismo. Con un encabezado de tres bytes, el bit H es puesto a "cero" para indicar que este encabezado es de tres bytes. Los otros bits operan de una forma muy similar. El segundo y el tercer byte del encabezado contienen la dirección destino. La dirección destino puede ser de direccionamiento funcional o físico. El tercer y último byte del encabezado de tres bytes contiene la dirección física de la fuente del mensaje. Porque la dirección fuente debe ser siempre única, el arbitraje siempre es resultado al final del tercer byte aunque sean utilizados encabezados de tres bytes.

BITS DE DATOS

Los bits de datos que son enviados en el campo de datos consisten en "ceros" y "unos" representados por símbolos. Un símbolo pasivo de un bit es de longitud de 128ms de potencial bajo en el bus J1850. Un símbolo dominante de un bit es de 64us de duración de potencial bajo en el bus J1850 y un cero dominante es de 128us de longitud de potencial alto en el bus J1850. Estos símbolos requieren solamente de una transición por bit, y se pueden mezclar las combinaciones indefinidamente.

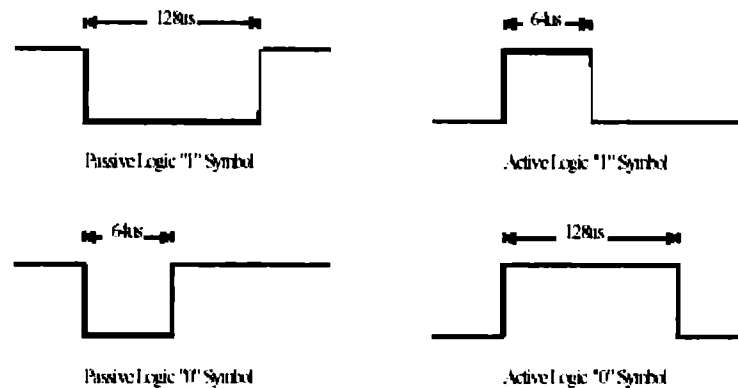


Figura 2.3.5

Tiempos de los bits símbolos

CHEQUEO CÍCLICO REDUNDANTE (CRC –CYCLICAL REDUNDANCY CHECK-)

Agregado al final de cada mensaje VPW J1850 está el byte de chequeo de error (CRC). Este, junto con los once bytes de datos, hace la longitud máxima del mensaje de doce bytes. Uno de los esquemas de protección de error más efectivos en el estándar SAE J1850 es su CRC. El total del mensaje, no incluye el símbolo SOF, es tratado como “una gran palabra”. Esta “gran palabra” es dividida por un “polinomio” especial. Los 1’s complementan los cálculos del “polinomio” los cuales son agregados al final del mensaje en forma de byte, y transmitido. Todos los nodos receptores hacen cálculos con un “polinomio” similar para recibir los mensajes, incluyendo la inversión del CRC. Si no han ocurrido errores, entonces el CRC resultante en el receptor siempre igualará un cierto valor, (C3 hex).⁹

FIN DE DATOS (EOD –END OF DATA-)

Inmediatamente después del haber transmitido el byte CRC, un símbolo de Fin de datos (EOD) es transmitido. El símbolo EOD consiste en un símbolo de bajo potencial de 200us de duración en el bus J1850. Directamente después del símbolo EOD, el nodo receptor, puede optar por responder inmediatamente el mensaje. A esta respuesta se le llama, IFR (In-Frame Response) y contribuye a las opciones de manejo de errores que soporta este protocolo. Sin embargo, si no se desea IFR, entonces el símbolo de EOD se permite que vaya en el símbolo EOF (End Of Frame). El símbolo EOF no es otra cosa que el símbolo EOD que ha sido cambiado de 200us a 280us de duración. El símbolo EOF denota el término de la trama del mensaje en particular.

⁹ Los detalles del proceso de los cálculos del CRC, se pueden encontrar en el documento SAE J1850 REV: MAY94, iniciando en la página 15.

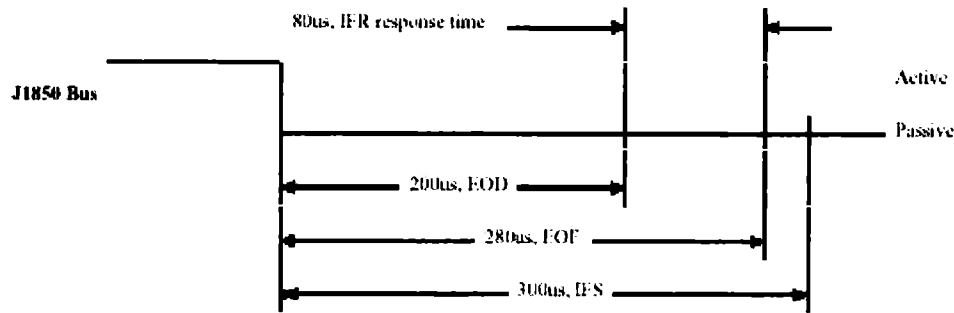


Figura 2.3.6
Símbolo EOD versus Símbolo EOF

RESPUESTA DENTRO DEL FRAME (IFR –IN-FRAME RESPONSE-)

El IFR provee una plataforma para los nodos que reciben de forma remota para activar el reconocimiento de una transmisión. Los nodos receptores añaden al final una respuesta al mensaje transmitido por los nodos transmisores. IFR permite incrementar la eficiencia en la transmisión de mensajes desde que un nodo receptor puede responder dentro del mismo mensaje.

La respuesta de nodo transmisor para un IFR se denota como el bit K en el encabezado del nodo transmisor. Todos los nodos receptores requieren de verificar el bit K y responderlo. Esta respuesta debe venir dentro del tiempo de un símbolo EOD o EOF, el cual es de aproximadamente 80 us.

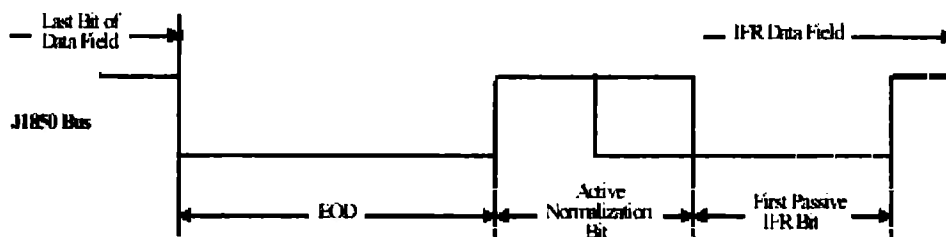


Figura 2.3.7
Símbolos de IFR y Normalización

El inicio de una respuesta IFR es denotada por la transmisión por los nodos receptores del Bit de Normalización (NB). El NB provee una separación activa entre el símbolo pasivo EOD y el primer bit de datos de la respuesta IFR. El NB puede tomar una o dos formas. Cada forma denota si el IFR incluye su propio CRC o no. La interpretación del NB es dependiente del estándar de la industria se requiera. De conformidad con GM. Un símbolo NB “cero dominante”denota que el IFR incluye CRC. Sin embargo, se cree que GM no requiere IFR para ninguna implementación J1850. De acuerdo a la conformidad de Chrysler es exactamente lo opuesto de GM, y es la opción de configuración para la mayoría de los sistemas que utilizar IFR.

	GM (NB=1)	Chrysler NB=0)
IFR does not include CRC	active long NB	active short NB
IFR includes CRC	active short NB	active long NB

Tabla 2.3.2, Configuración del Bit de Normalización

El IFR puede tomar una de las cuatro formas distintas etiquetadas como Tipo 0, Tipo 1, Tipo 2, y Tipo 3. Un IFR Tipo 0 está definido como el que no requiere ningún tipo de respuesta. Un IFR Tipo 1 consiste en un solo byte de respuesta desde un solo respondedor. El IFR Tipo 2 son respuestas de un solo byte desde varios respondedores. El IFR no está completado hasta que todos los respondedores aplicables hayan tenido oportunidad para responder. El IFR Tipo 3 es una respuesta que contiene múltiples bytes de un solo respondedor, y un CRC es agregado al final. Después del símbolo EOF, un periodo de gracia de bus desocupado es dado antes que otros nodos comiencen a transmitir nuevos datos. Este tiempo es llamado IFS (Inter Frame Separation) – Separación entre Tramas- y se denota por un “potencial bajo” en un tiempo de 300us seguidos de un símbolo EOF. Y, un IFS puede ser drenado indefinidamente en un tiempo tan largo como el que el bus J1850 está desocupado.

Otro símbolo de interés es el Break (BRK). Este símbolo denotado por un “potencial alto” en el bus J1850 de al menos 768us de longitud. La longitud máxima del símbolo BRK es aproximadamente 5 ms. El símbolo BRK puede ser inmediatamente transmitido el bus J1850 en cualquier momento. Los símbolos BRK causan que cualquier otro nodo que esté transmitiendo pierda el arbitraje y cese su transmisión. El símbolo BRK manda a todos los nodos en un estado de “listo para recibir”. El símbolo BRK está reservado para diagnóstico y herramientas de escaneo.

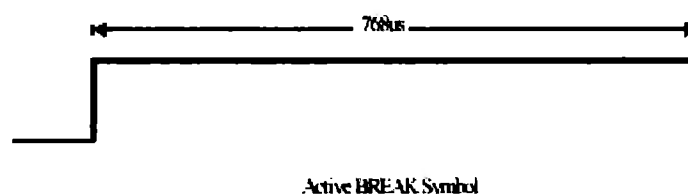


Figura 2.3.8

Símbolo Break en el J1850

El objetivo del estándar SAE J1850 es la tolerancia a fallos. Cualquier interferencia o disturbio menor a 8us de longitud es digitalmente filtrado como ruido no deseado. Aunque el filtrado de 8us fuerza a un retraso para todos los datos entrantes, la mayoría del ruido es eliminado en la red J1850.

CAPA FÍSICA

La capa física de 10.4 Kb/s VPW consiste en un solo cable manejado por voltaje puesto de forma aleatoria son restricciones de ruteo. Para más ayuda en el sistema de multicanalización, referirse a SAE J1211 y para especificaciones de componentes a SAEJ1879.

La longitud de la red en el vehículo es de 25 metros de largo. La longitud fuera del vehículo puede ser máximo de 5 metros de largo. Esto significa que la longitud total de la

red puede ser de hasta 40 metros. El número máximo de unidades estándar de carga, incluyendo fuera de la unidad, es de 32 nodos. Esto es asumiendo que el número de nodos fuera del vehículo es de una unidad (el scanner). Una unidad de carga consiste en 10Kohms y 470 pF. Las mediciones de carga fuera del vehículo son hechas en el conector de diagnóstico SAE J1962. La carga mínima fuera del vehículo es de 10.6Kohms. La capacitancia no debe exceder de los 500 pF. Las lecturas de capacitancia deben ser hechas entre la señal del cable del bus y la tierra del chasis del vehículo.

La capa de aplicación del J1850 requiere un símbolo de bit "cero" siempre domine sobre un símbolo bit "uno". Apegándose a las definiciones de símbolos del J1850 VPW de 10.4Kp/s, podemos verificar que esto sucede. El límite máximo del número de bytes en un mensaje (sin incluir los delimitadores SOF, EOD, EOF e IFS), es de 12. Los requerimientos de tiempo de VPW dados anteriormente están dibujados en el documento SAE J1850 REV MAY94, página 35. Incluidos los parámetros de VPW DC.

Symbol	T _{Min}	T _{nom}	T _{max}	R _{min}	R _{max}	units
Passive "0" Active "1"	≥ 49	64	≤ 79	> 34	≤ 96	μs
Passive "1" Active "0"	≥ 112	128	≤ 145	> 96	≤ 163	μs
Active "SOF"	≥ 182	200	≤ 218	> 163	≤ 239	μs
Passive "EOD"	≥ 182	200	≤ 218	> 163	≤ 239	μs
Passive "EOF"	≥ 261	280	N/A	> 239	N/A	μs
Passive "IFS"	≥ 280	300	N/A	> 280	N/A	μs
Active "BREAK"	≥ 280μs	300μs	≤ 5ms	> 239μs	≤ 1.0 s	N/A

Tabla 2.3.3, Tiempos del Pulso VPW

Parameter	Symbol	Min	Typ	Max	Units
Input High Voltage	V _{ih}	4.25	—	20.00	Volts
Input Low Voltage	V _{il}	—	—	3.5	Volts
Output High Voltage	V _{oh}	6.25	—	8.0	Volts
Output Low Voltage	V _{ol}	0.00	—	1.5	Volts
Absolute Ground Offset Voltage	V _{gs}	0.00	—	2.00	Volts
Network Resistance	R _{load}	315	—	1,575	ohms
Network Capacitance	C _{load}	2,470	—	16,544	pF
Network Time Constant	T _{load}	—	—	5.2	μs
Signal Transition Time	T _t	—	—	18.0	μs
Node Resistance (unit load)	R _{ul}	—	10,600	—	ohms
Node Capacitance (unit load)	C _{ul}	—	470	—	pF
Node Leakage Current	I _{leak}	—	—	10	μA

Tabla 2.3.4, Parámetros de DC del PVW

La descripción física del VPW de 10.4Kb/s es lo suficientemente robusta para sobrevivir muchas fallas catastróficas, y trabajar sin daños y con la habilidad de recobrar la operación normal automáticamente al desaparecer la falla. Tolerancia a fallas en circuitos abiertos o cortos circuitos a tierra, cortos circuitos hacia voltaje, o cortos circuitos entre voltaje y tierra, son todos superados y recobrada la operación normal. La interferencia a las comunicaciones no es un obstáculo si alguno de los nodos de la red falla.

Todos los pulsos en la red J1850 de 10.4 Kb/s son ondas bien formadas. La onda es redondeada de las esquinas para las transiciones del bus, quitando todas las componentes de señales de alta frecuencia no deseadas. Los anchos de los pulsos transmitidos son afectados por las tolerancias de oscilación de los nodos, y variaciones en los retrasos en la transmisión entre los nodos y el medio del bus. Estos aspectos deben ser considerados en el diseño de un nodo J1850.



Figura 2.3.9

Forma de Onda de SAE J1850 VPW

Value	Value Description
$V_{oh,min}$	Minimum output high voltage. This is the highest trip point with 2V receiver noise.
$V_{ih,min}$	Minimum input high voltage. This is the highest trip point with no offset noise.
V_i	Ideal receiver trip point.
$V_{il,max}$	Maximum input high voltage. This is the lowest trip point with no offset noise.
$V_{ol,max}$	Maximum output low voltage. This is the highest trip point with 2V source noise.
$T_{t,max}$	Maximum signal transition time.

Tabla 2.3.5, Niveles de Voltaje

El poder leer la información del bus de datos del vehículo permite que sea posible diseñar equipos que tengan la capacidad de interactuar con el mismo de forma externa, incluso manipular el automotor.

El diseño de un proyecto, cualquiera que éste sea, es una de las partes más importantes, ya que todo lo que será el producto se define en esta etapa. Para el proyecto de Impacto, teniendo una documentación clara de los antecedentes que sirven de base para la comprensión de la problemática, la siguiente etapa descrita en este capítulo es el diseño de cómo se hace la conexión con el vehículo para obtener la información necesaria, cómo se procesan los datos y cómo se integran las partes.

La investigación de tipos de choque permite que la posición de los sensores puedan ser colocados de forma estratégica dentro del vehículo para poder detectar el tipo de choque que sucede al momento del impacto.

Tomando la decisión de los sensores, entonces podemos trabajar en el diseño de la tarjeta que tendrá en control de éstos, la cual será a través de un microcontrolador como se describe en el siguiente capítulo.

3 Diseño y Desarrollo.

3.1 Introducción

El capítulo que corresponde al diseño es quizá el más importante, ya que en él se presenta la integración de los antecedentes y de la investigación (capítulo I y II). De tal forma que a continuación se describirán los métodos que se utilizan en el proyecto para evaluar los choques y la comunicación del sistema con el automóvil. Además es en este capítulo donde también se presentan las tarjetas de implementación de dicho diseño para que en capítulos posteriores se verifiquen las pruebas realizadas a ellas y los resultados obtenidos del diseño.

Para lograr el objetivo del proyecto, se determinó, que es necesario contar con un método directo, un método indirecto y un dispositivo capaz de hacer la toma de decisiones para la transmisión.

El método directo consiste en varios sensores colocados en el automóvil, que en todo momento estén monitoreando el estado del vehículo.

El método indirecto se realiza por medio de la medida de la desaceleración del auto.

Y para la toma de decisiones, se cuenta con un microprocesador que es capaz de evaluar si ocurrió o no impacto y con base a lo anterior, grabarlo y transmitirlo.

La primera sección de este capítulo es la organización del proyecto donde se describe a grandes rasgos cómo funcionará el sistema completo, donde se divide en dos grandes bloques: La comunicación con el vehículo para obtener la velocidad en cualquier momento y el sensado y procesamiento de los datos.

La comunicación con el vehículo se hace a través del bus de datos del mismo, con un circuito adicional utilizando un Pic.

El sensado y procesamiento de los datos los hacemos con una serie de sensores de "choque" que son interruptores normalmente abiertos, y los procesamos con un

microcontrolador 89C52, el cual tiene un algoritmo para sensor e informar cuándo ha ocurrido un choque.

3.2 Organización del Proyecto

En el capítulo anterior, se realizó un trabajo de investigación de sensores que fueran de utilidad para el proyecto, adicionalmente también se abordó el tema acerca del puerto del automóvil y sus estándares, pero no se hizo mención en alguna parte de cómo se esperaba ensamblar los sensores y los datos del bus del carro en el proyecto, por lo que a continuación se presenta de forma general el diseño.

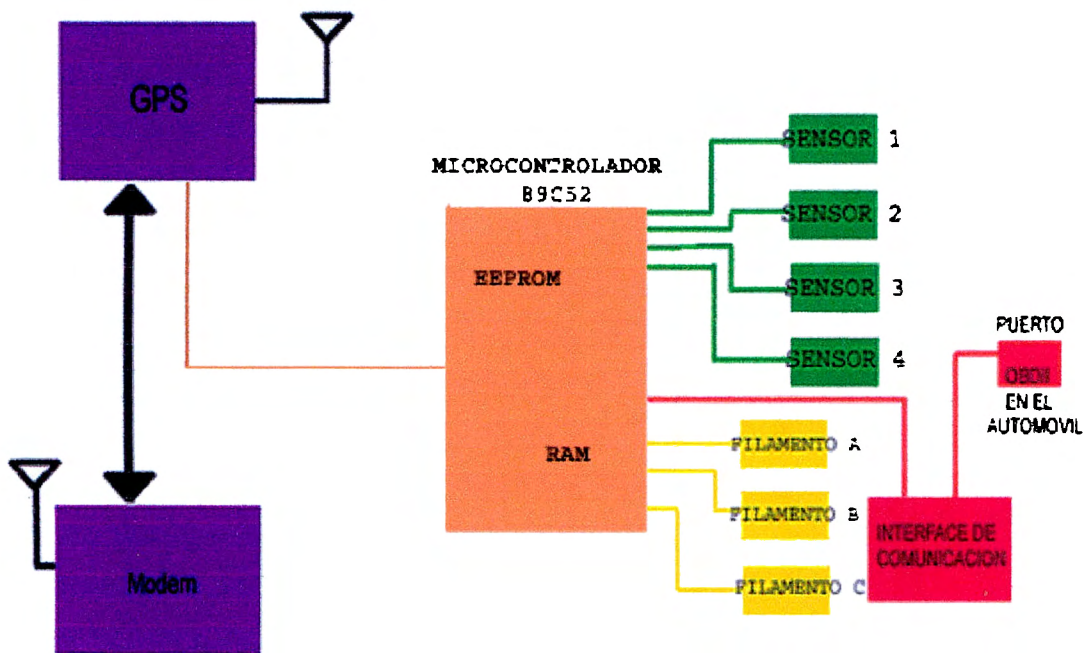


Figura 3.2.1

En esta figura se puede notar la forma del ensamblaje de los métodos directo e indirecto (sensores/filamentos e interfaz de comunicación) ya que es por medio de un microcontrolador que se evalúan los datos monitoreados por los sensores y a la par del cambio de aceleración se determina el envío de la señal de alarma que determina que es necesaria la presencia de los servicios de urgencia y aseguradora.

3.2.1 Diseño de comunicación con el Bus del vehículo, OBDII

Para poder comunicarse al bus de datos, existen algunos Circuitos Integrados que funcionan como interfaz entre el bus y una computadora personal o una PDA.

En este caso se utiliza un PIC programado por ELM Electronics, donde este circuito sirve para entrar al bus de datos del J1850 y obtener la información que se requiere. El circuito es el ELM 322, el cual es un PIC de Microchip Technology de la familia PIC12C5XX. Con las siguientes características:

- Bajo consumo de energía

- Capacidad de alta corriente a la salida – hasta 25 mA
- Controlado por cristal para mayor exactitud
- Configurable con comandos AT
- Salida estándar con caracteres ASCII
- Puerto RS232 de alta velocidad
- Soporte del protocolo J1850 de 10.4 Kb/s<

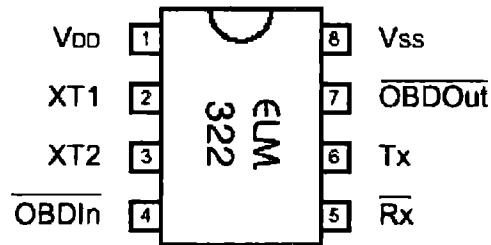


Figura 3.2.2
Diagrama de Conexión del Chip

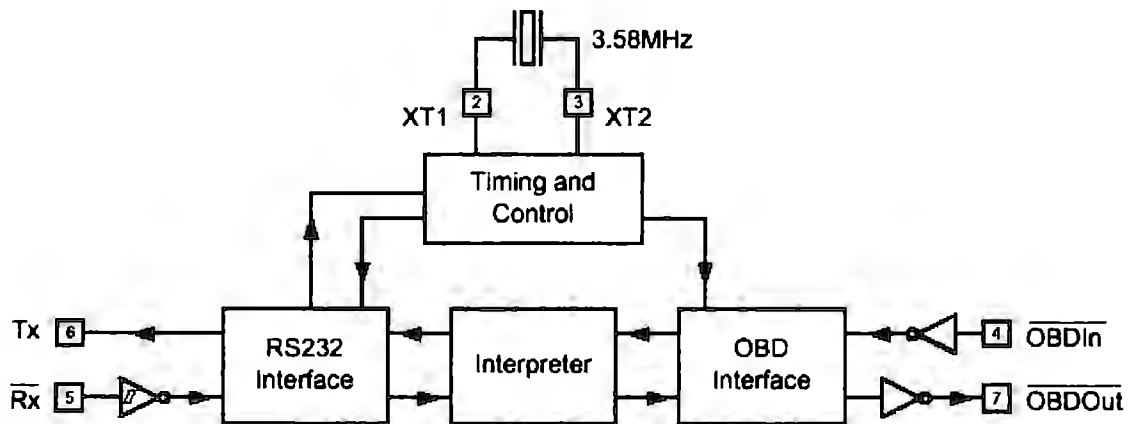


Figura 3.2.3
Diagrama a Bloques

DESCRIPCIÓN DE LOS PINES:

VDD (pin 1)

Este pin es la alimentación positiva, este pin debe ser siempre el punto más positivo. Las conexiones internas a este pin son usadas para reiniciar el microprocesador, por lo que no se requiere un reset externo.

XT1 (pin 2) y XT2 (pin 3)

Un cristal de 3.579545Mhz de televisión NTSC es conectado entre estos dos pines. Capacitores de carga de 27 pF son conectados normalmente entre estos pines y tierra.

OBDIn (pin 4) Activado en bajo

Los datos del bus OBD llegan a este pin, con un nivel bajo lógico representa el estado activo (y un alto el pasivo).

Rx (pin 5) Activado en bajo

Este pin se conecta directamente a la señal de Transmisión de la computadora o PDA, en este caso el MC89C52.

Tx (Pin 6)

Salida de Transmisión al la entrada RS232 de la computadora, se utiliza típicamente con un transistor PNP para aumentar la corriente.

VSS (pin 8)

Tierra del circuito. Este pin siempre debe ser el más negativo del circuito.

COMUNICACIÓN AL BUS DE DATOS

De acuerdo a las normas establecidas por SAE, la comunicación con los vehículos debe ser en formato Hexadecimal.

Siguiendo el estándar, para comunicarse al vehículo, es necesario introducir primero el modo, el cual es el que indica el tipo de información a enviar, seguido de la información. Existen nueve modos de acuerdo al estándar de SAE:

01	Muestra la información en ese momento
02	Muestra la información del frame
03	Muestra los códigos de error
04	Limpia los códigos de error y valores establecidos
05	Resultados de prueba, sensores de oxígeno
06	Resultados de prueba, monitoreados no continuos
07	Resultados de prueba, monitoreados continuos
08	Modo especial de control
09	Requerimientos de información del vehículo.

Tabla 3.2.1, Códigos de modos de acuerdo a SAE

La interfaz para conectarse entre el vehículo y el microcontrolador 89C52 se compone del chip ELM 322 y una serie de componentes adicionales. La Figura 3.2.4 es el diagrama esquemático:

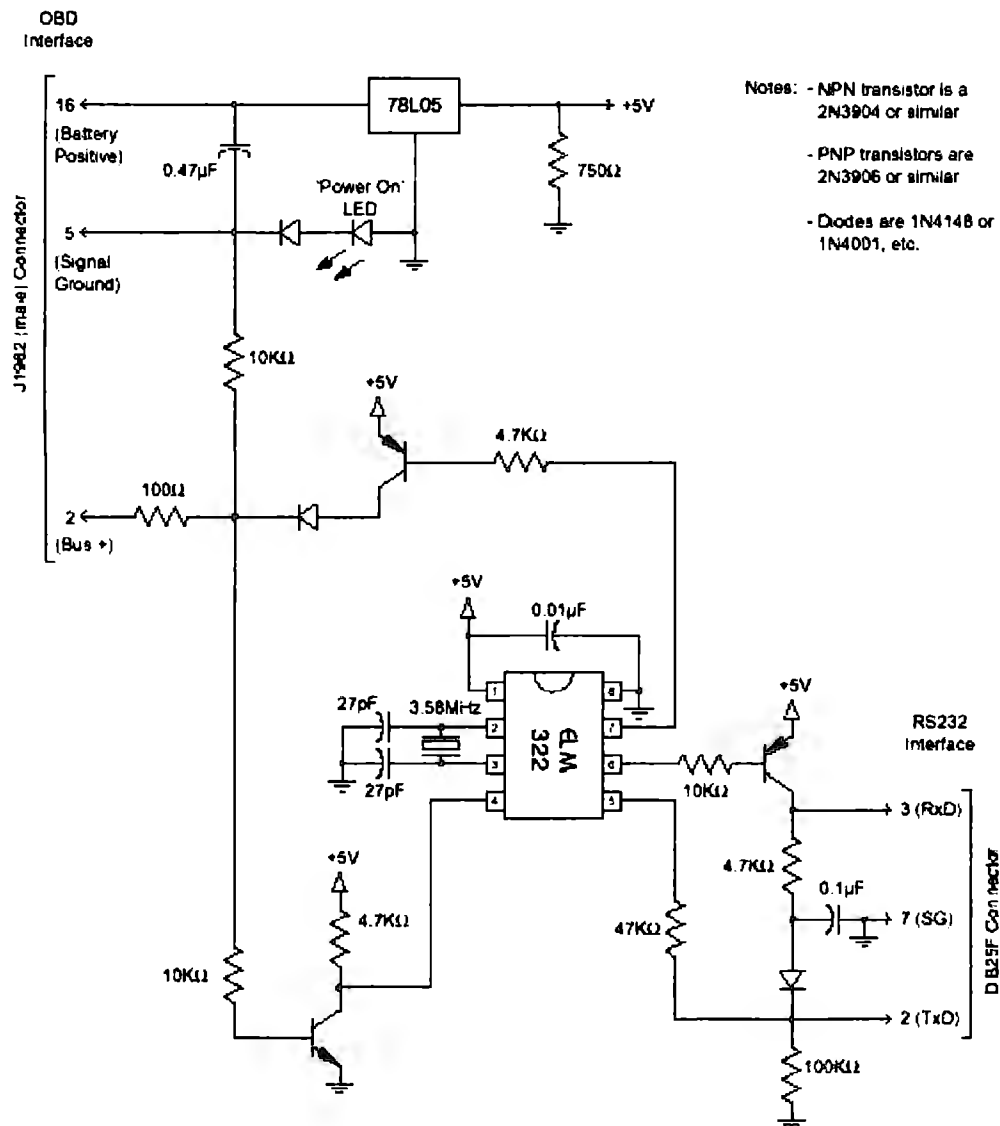


Figura 3.2.4

Diagrama esquemático de la interfaz con el bus J1850 VPW

3.2.2 Diseño del módulo de adquisición y procesamiento de datos

3.2.2.1 Microcontrolador

El microcontrolador es un Atmel 89C52 que pertenece a la familia del 8051 por lo que se otorga un espacio en este capítulo para conocer las características de dicha familia con el fin de facilitar la comprensión del diseño y el programa.

Las principales ventajas que tiene son:

- Cmos y CPU de 8-bit
- 8K Bytes de memoria Flash reprogramable (In-System)
- Resistencia: 1000 Ciclos de Escritura/Borrado.

- ☑ Operación estática completa: 0 Hz a 24MHz.
- ☑ 3 niveles de seguro memoria programable.
- ☑ 256 X 8-bit de RAM interna.
- ☑ 32 líneas de entrada/salida programables.
- ☑ 3 temporizadores/contadores de 16-bit.
- ☑ 8 fuentes de interrupción.
- ☑ Canales programables seriales.
- ☑ Baja potencia parado y modo de Baja potencia.

Diagrama de bloques del microcontrolador:

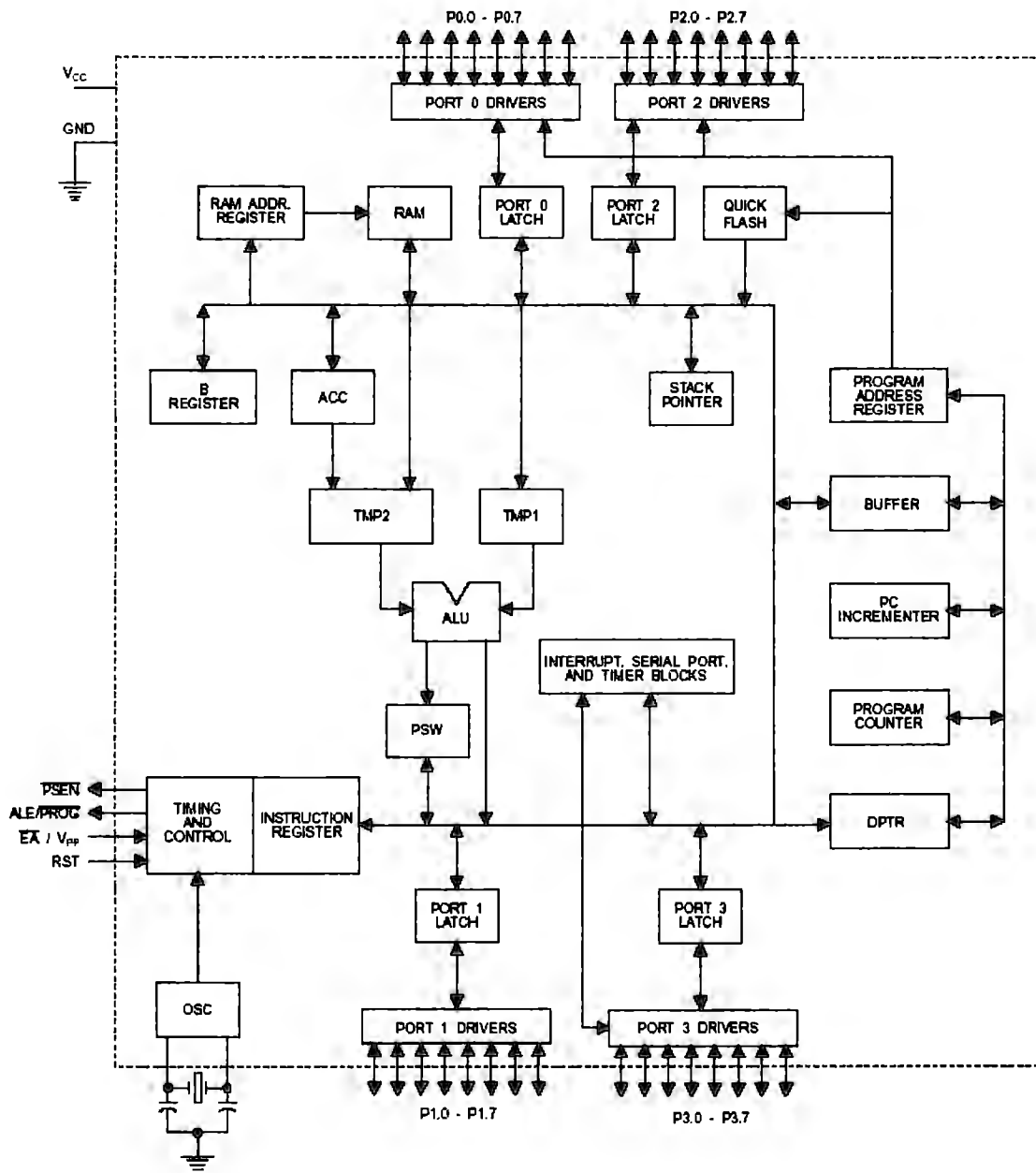


Figura 3.2.5

El Microcontrolador 89C52 forma parte de la familia de Microcontroladores 8051 de Intel, los cuales cuentan internamente con una CPU de 8 bits, memoria, lógica de interrupciones, contadores y temporizadores y líneas de entrada/salida digitales.

Todos los miembros de la familia comparten una arquitectura interna común, contando con el mismo conjunto de instrucciones; modos de direccionamiento y espacios de memoria. Las principales diferencias entre los miembros de la familia 8051 son la cantidad de memoria interna; la cantidad y tipos de funciones de entrada/salida y la tecnología de fabricación y debido a esto se usa en la mayoría de literatura al respecto el termino 8051 para referirse colectivamente a todos los miembros de la familia.

Las mínimos componentes del 8051 son los siguientes:

- Una CPU de 8 bits.
- 32 líneas de entrada/salida agrupadas en 4 puertos.
- Memoria Externa de Programa de 64 Kbytes.
- Memoria Externa de Datos de 64 Kbytes.
- Memoria Interna de Datos de 256 bytes.
- 2 temporizadores/contadores de 16 bits.
- 5 fuentes de interrupción con dos niveles de prioridad.
- Procesador Booleano.
- Puerto Serie asíncrono full-duplex.

Los pines del 89C52

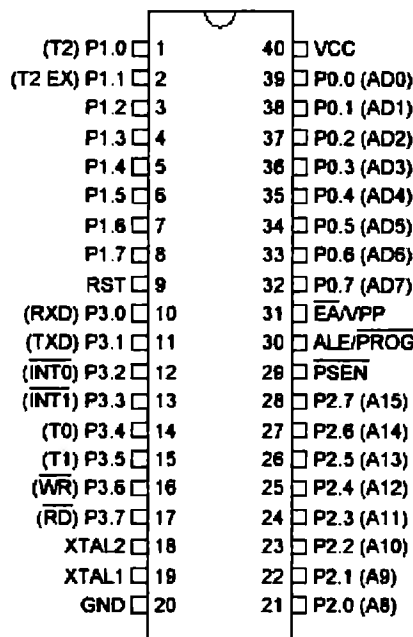


Figura 3.2.6

Por ser un Microcontrolador de 8 bits, la memoria del 8051 se organiza en bytes y casi todas sus instrucciones operan sobre bytes. El registro principal es el llamado Acumulador. Se pueden acceder otros operandos usando uno de los siguientes modos direccionamiento disponibles en el 8051: registro implícito; directo; indirecto o inmediato. Los operandos residen en uno de los cinco espacios de memoria con los que cuenta el 8051.

Los cinco espacios de memoria del 8051 son los siguientes: Memoria de Programa; Memoria Externa de Datos; Memoria Interna de Datos; Registros de Funciones Especiales y Memoria de Bit.

El espacio de la Memoria de Programa contiene todas las instrucciones, datos, tablas y cadenas de caracteres (strings) usadas en los programas. Esta memoria se direcciona principalmente usando el registro de 16 bits Program Counter (PC) y el registro de 16 bits llamado Data Pointer (DPTR). El tamaño máximo de la Memoria de Programa es de 64 Kbytes. Algunos miembros de la familia 8051 integran dentro del circuito integrado alguna cantidad de memoria ROM o EPROM como parte del espacio de la Memoria de Programa.

La Memoria Externa de Datos contiene todas las variables y estructuras de datos que no caben en la memoria interna del Microcontrolador. Esta memoria se direcciona principalmente por el registro de 16 bits Data Pointer (DPTR), aunque también se puede direccionar un banco de Memoria Externa de Datos de 256 bytes usando los dos primeros registros de propósito general (R0 y R1) con los que cuenta el 8051. El tamaño máximo de la Memoria Externa de Datos es de 64 Kbytes y, como ya se ha mencionado, solo se puede acceder usando el modo indirecto de direccionamiento con los registros DPTR, R0 y R1.

El espacio de Memoria Interna de Datos funcionalmente es la memoria de datos más importante, ya que ahí es donde residen cuatro bancos de registros de propósito general; la pila o stack del programa; 128 bits de los 256 bits de un área de memoria direccionable por bit y todas las variables y estructuras de datos operadas directamente por el programa. El tamaño máximo de la Memoria Interna de Datos es de 256 bytes, sin embargo, los diferentes miembros de la familia 8051 integran cantidades diferentes de este espacio de memoria en el mismo circuito integrado.

La familia 8051 contiene un espacio para los denominados Registros de Funciones Especiales (Special Function Registers-SFR) destinado para los puertos de entrada/salida, temporizadores y puerto serie del circuito integrado. Estos registros incluyen al Stack Pointer; al registro de la palabra de estado del programa (Program Status Word-PSW) y al Acumulador. La cantidad máxima de Registros de Funciones Especiales es 128, aunque depende también de la cantidad y tipo de funciones periféricas integradas al miembro en particular de la familia 8051. Todos los Registros de Funciones Especiales tienen direcciones mayores a 127 y se ubican en los 128 bytes superiores de la Memoria Interna de Datos. Estas dos áreas de la Memoria Interna de Datos se diferencian por el modo de direccionamiento usado para accederlas. Los Registros de Funciones Especiales solo se pueden acceder usando el modo de direccionamiento Directo, mientras que los 128 bytes superiores solo se pueden acceder con el modo de direccionamiento Indirecto.

Por otra parte, el espacio de Memoria de Bit se usa para almacenar variables y banderas de un bit, existiendo en el 8051 instrucciones específicas que operan solo en el espacio de Memoria de Bit. El tamaño máximo de la Memoria de Bit es de 256 bits, 128 de los bits comparten su espacio con 16 bytes del espacio de la Memoria Interna de Datos y los otros 128 bits lo hacen con los Registros de Funciones Especiales. Los bits sólo se pueden acceder usando las instrucciones de bit del 8051 y el modo de direccionamiento Directo.

El 8051 tiene un conjunto completo de instrucciones lógicas y aritméticas, que incluyen multiplicación y división. El 8051 es un buen microcontrolador para procesamiento de bits, en ocasiones llamado procesamiento Booleano. Usando la bandera de Acarreo (Carry Flag) del registro PSW como un acumulador de un bit, el 8051 puede realizar operaciones de movimiento y operaciones lógicas entre los bits del espacio de Memoria de Bit y la bandera de Acarreo. Los bits del espacio de Memoria de Bit también se pueden usar como banderas de propósito general en las instrucciones de salto y de prueba de bits.

Descripción de los pines.

Vcc

Fuente de voltaje

Gnd
Tierra

Puerto 0

Es un puerto de E/S bi-direccional de drenaje abierto es un puerto de salida, cada pin puede aceptar una entrada TTL. Cuando 1s son escritos en los pines del puerto 0, el puerto puede ser usado con impedancias altas de entrada.

El puerto 0 puede ser configurado para ser multiplexado en orden-bajo de bus de direcciones de datos durante el acceso a programa externo y memoria de datos, el P0 tiene un paro interno.

El P0 además recibe el código de bytes durante la programación flash y salidas del código bytes durante la verificación del programa. El paro externo es requerido durante la verificación del programa.

Puerto 1

Este es un puerto 8-bits bidireccional de E/S con paro interno. Cuando 1s son escritos en los pines del puerto, hay niveles altos por el paro interno y pueden ser usados como entradas. Como entradas, los pines del puerto 1 son parados externamente en bajo y se generará una fuente de corriente (I_{IL}) por el paro interno.

Adicionalmente, P1.0 y P1.1 pueden ser configurados como temporizador/contador como 2 contadores externos de entrada (P1.0/T2) y el temporizador/contador de entrada 2 de disparo (P1.1/T2EX), respectivamente y se muestra en la siguiente tabla.

También este puerto recibe la parte baja de bytes de direcciones durante la programación y verificación de Flash.

Port Pin	Alternate Functions
P1.0	T2 (external count input to Timer/Counter 2), clock-out
P1.1	T2EX (Timer/Counter 2 capture/reload trigger and direction control)

Tabla 3.2.2

Puerto 2

Este es un puerto 8-bits bidireccional de E/S con paro interno. Cuando 1s son escritos en los pines del puerto, hay niveles altos por el paro interno y pueden ser usados como entradas. Como entradas, los pines del puerto 2 son parados externamente en bajo y se generará una fuente de corriente (I_{IL}) por el paro interno.

El puerto 2 emite el orden alta de bytes de direcciones durante la búsqueda del programa de memoria externa y durante el acceso a los datos externos que usan 16-bits de direcciones (MOVX @ DPTR). En esta aplicación, el puerto 2 usa fuertes paros cuando emiten 1s. Durante el acceso a datos externos en memoria que usan direcciones de 8-bits (MOVX @R1), el puerto 2 emite el contenido del P2 en el registro de función especial.

Puerto 3

Este es un puerto 8-bits bidireccional de E/S con paro interno. Cuando 1s son escritos en los pines del puerto, hay niveles altos por el paro interno y pueden ser usados como entradas. Como entradas, los pines del puerto 3 son parados externamente en bajo y se generará una fuente de corriente (I_{IL}) por el paro interno.

El puerto 3 además sirve para varias funciones especiales que son ventajas del 89C52, mostradas en la siguiente tabla.

El puerto 3 puede recibir algunas señales de control para programación y verificación de flash.

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{\text{INT0}}$ (external interrupt 0)
P3.3	$\overline{\text{INT1}}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{\text{WR}}$ (external data memory write strobe)
P3.7	$\overline{\text{RD}}$ (external data memory read strobe)

Tabla 3.2.3

RST

Entrada de reset. Una entrada en alto en este pin por 2 ciclos de máquina mientras el oscilador está corriendo reinicia el dispositivo.

ALE/ PROG

La dirección "latch"habilitada es en el pulso de salida por corrimiento del byte de bajo nivel de la dirección durante el acceso a memoria externa. Este pin también es el pulso de entrada de programa (PROG) durante la programación Flash.

En una operación normal, ALE es emitida como una tasa de 1/6 la frecuencia de oscilación y puede ser utilizada por temporizadores externos o propósitos de reloj.

XTAL 1

Entrada al amplificador oscilador inversor y entrada al reloj interno de circuito de operación.

XTAL 2

Salida del amplificador oscilador inversor.

Después de conocer el funcionamiento interno del microcontrolador se presenta a continuación el diseño del Hardware necesario como "cerebro" encargado de la toma de decisiones del sistema.

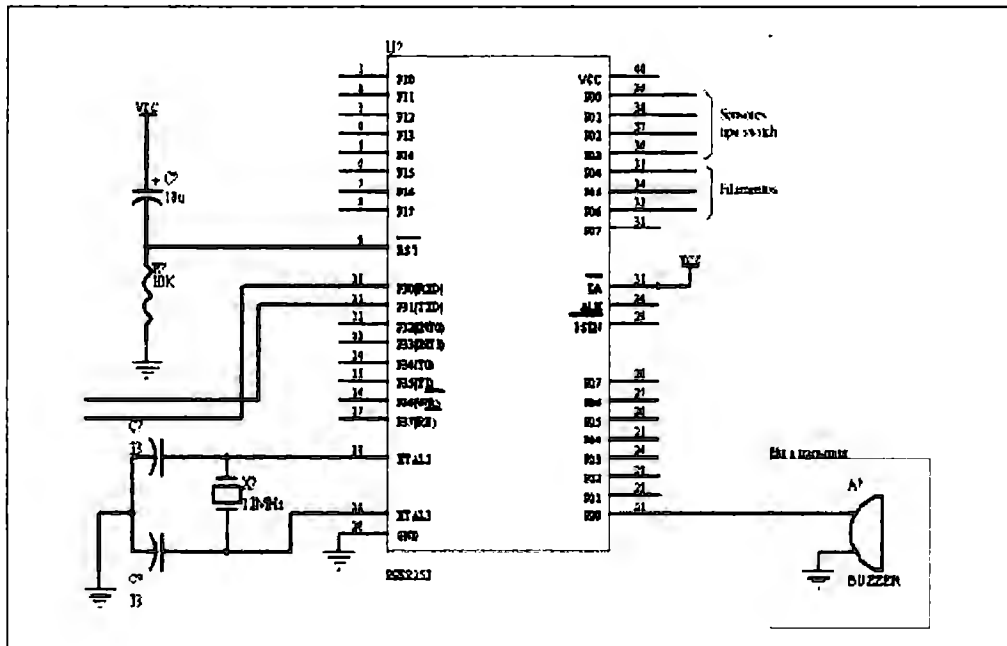


Figura 3.2.7

Los elementos integrados en esta parte son: el circuito de reset para limpiar el sistema y la reinicialización del mismo, el circuito para la oscilación, que en este caso para la velocidad requerida es de 11.0592 MHz, la comunicación serial hacia el MAX232 por el puerto 3 encargado de la transmisión serial el pin 0 (RX) de entrada serial y el pin 1 (TX) de salida serial y finalmente el bit del puerto 2 (P2.0) que enciende el buzzer para reportar a la central la necesidad de ayuda.

El puerto 0 que es el encargado de recibir la información, en los pines 0-3 se conectan los sensores de contacto y en los pines 4-6 se tienen conectados los filamentos y estas son las entradas del microcontrolador que se activan en bajo.

La señal EA en alto, indica que el se debe ejecutar la programación interna.

Finalmente aparece en los pines 20 y 40 la alimentación del microcontrolador que debe de ser TTL (5V) en el pin 40 y Tierra en el pin 20.

Bloque de Comunicación Serial

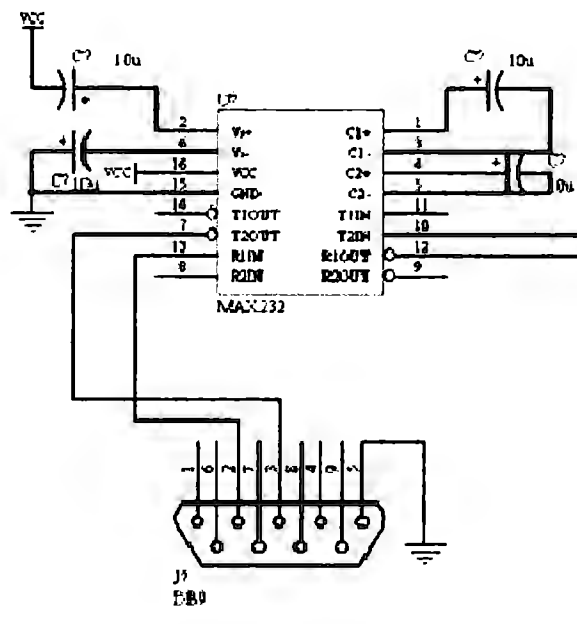


Figura 3.2.8

El bloque de comunicaciones Serial para la Computadora es un CI MAX 232 que condiciona los niveles de voltaje del microcontrolador con aquellas del puerto serie (normalmente COM 1) de una computadora o un PDA (PALM) en el estándar RS 232.

3.2.2.2 Sensores

Los sensores de contacto están conectados al microcontrolador con la siguiente configuración:

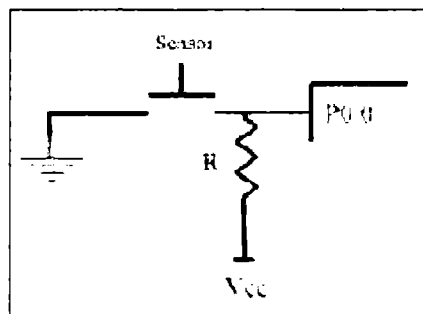


Figura 3.2.9

De tal forma que cuando el sensor no está presionado en el puerto del microcontrolador se puede leer un nivel alto, y al presionarse cambia a tierra la terminal del puerto.

Al ser colocados en la parte frontal del automóvil y en la parte posterior, son capaces de detectar las colisiones frontales y posteriores o ambas, pues con la deformación del

chasis (lugar donde se colocarán estos sensores) se presiona el sensor mandando pulso de bajada al microcontrolador.

Los filamentos en cambio, están enviando Vcc debido a que lo que se verifica en este caso es la continuidad del circuito, cuando este se rompe se detecta tierra en el pin del puerto, detectando el daño del automóvil.

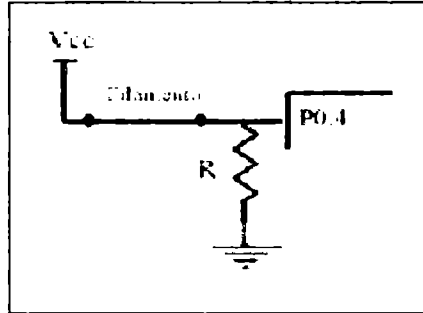
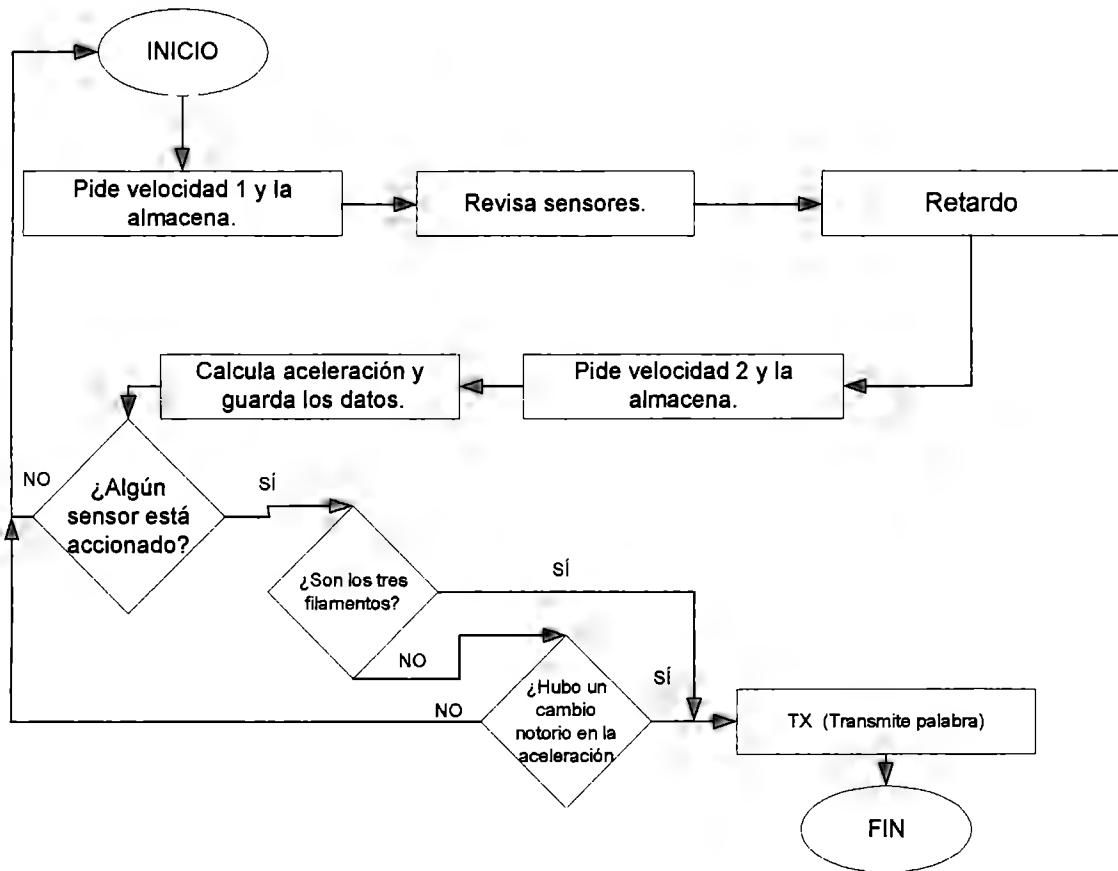


Figura 3.2.10

3.2.3 Diseño de Programa

A continuación se presenta el programa que se ejecuta para tomar las decisiones del analizador de impacto. En primer lugar en diagrama de flujo y posteriormente el código en lenguaje C.



El programa final que valida todos los casos de choque, y realiza el reporte, se presenta abajo.

En el programa, aparecen en bloques las diferentes partes, para que el lector comprenda con mayor facilidad el contenido. Cabe destacar, que al ejecutarlo este manda datos, pero no los despliega pues el despliegue de los mismos implicaría que se están mandando datos al bus del auto y podría desencadenar fallas futuras.

```
#include <c:\franklin\include\reg52.h>
#include <c:\franklin\include\stdio.h>

//definicion de prototipos
int aceleracion(int velfin,int velin,n);
int checa_vel();
void checa_sensor();

//void checa_sensor2();
void reporte();
void repor2();
void borra();
void noimp();

/***** habilita escritura a EEPROM *****/
#define HABILITA_EE() WMCON|=0x10

/***** deshabilita escritura a EEPROM *****/
#define DESHABILITA_EE() WMCON&=0x08

/***** espera a que se escriba dato en EEPROM *****/
//void escribe_eepromInt(int *direccion, int valor);

#define ESPERA_EEW() while((WMCON&0x02)==0)
#define ON 0
#define OFF 1
#define BUZZER P20
#define sensor1 P21
#define sensor2 P22
#define sensor3 P23
#define sensor4 P24
#define filamento1 P25
#define filamento2 P26
#define filamento3 P27

//variables globales
int n;
int filamento;
int sensorn;
int acel;
//int T;
//int TIEMPO;
int desacel;
```

```

int damage;

int acel2[15];

idata int sensorA;
idata int sensorB;
idata int sensorC;
idata int sensorD;
idata int filamentoA;
idata int filamentoB;
idata int filamentoC;

void inicializa()
{
    //T=0;
    desacel=30;
    BUZZER=1;
    //TIEMPO=0;
    WMCON=0x08;           /*habilita EEPROM interna como XRAM*/
    P20=0x0FF;
    TH0=0xFC;
    TL0=0x17;
    ET0=1;
    TR0=1;
    filamento=1;
    sensorn=1;
    damage=1;

}

/*
void escribe_eepromInt(int *direccion, int valor)
{
    unsigned char x;
        HABILITA_EE();
        x=(char)valor;
        *direccion=x;
        ESPERA_EEW();

        valor>>=8;
        x=(char)valor;
        *(direccion+1)=x;
        ESPERA_EEW();

        DESHABILITA_EE();
        //int numero, dato;
        //xdata int eenumero;

```

```

//numero=10;
//escribe_eepromInt(&eenumero, numero);
//dato=eenumero;
}

*/

void timercero() interrupt 1 using 2
{
TF0=0;
TH0=0xFC;
TL0=0x17;

//base de tiempo 1 milisegundo
// if (TIEMPO==1) //tiempo entre velocidades
// T;

}

void inicializa_serie()
{
SCON=0x52;
TMOD=0x20;
TR1=1;

TL1=0xFD; //FD 9600 bps@11.059MHz
TH1=0xFD; //FA 4600 bps@11.059MHz
}

int checa_vel(){
int cuenta;
int numero;
char DATO;
char DATO2;

cuenta=0;
n=0;
putchar('\n');
putchar('\n');
putchar('\n');
putchar('0');
putchar('1');
putchar('0'); //Manda dato de lectura de velocidad
putchar('D');
putchar('\n');

```

```

do{
    DATO=getchar();
    if (DATO == ' '){ //busca los espacios en la cadena

        n++;}
    if((DATO=='z') || (DATO=='Z')){reporte();}
    if (n==2){
        DATO2=getchar();
        if((DATO=='z') || (DATO=='Z')){reporte();}
        DATO=getchar(); // guarda velocidad en hexadecimal
        if((DATO=='z') || (DATO=='Z')){reporte();}

        getchar();
    }

}while(n!=2); //conversión de velocidad de hexadecimal a decimal
if ((DATO=='A') || (DATO=='a')){
    numero=10;
}
if ((DATO=='B') || (DATO=='b')){
    numero=11;
}
if ((DATO=='C') || (DATO=='c')){
    numero=12;
}
if ((DATO=='D') || (DATO=='d')){
    numero=13;
}
if ((DATO=='E') || (DATO=='e')){
    numero=14;
}
if ((DATO=='F') || (DATO=='f')){
    numero=15;
}

if (DATO=='0'){
    numero=0;
}
if (DATO=='1'){
    numero=1;
}
if (DATO=='2'){
    numero=2;
}
if (DATO=='3'){

```



```
numero=3;
    }
    if (DATO=='4'){
        numero=4;
    }
    if (DATO=='5'){
        numero=5;
    }
    if (DATO=='6'){
        numero=6;
    }
    if (DATO=='7'){
        numero=7;
    }
    if (DATO=='8'){
        numero=8;
    }
    if (DATO=='9'){
        numero=9;
    }

    cuenta=cuenta+numero;
    if ((DATO2=='A') || (DATO2=='a')){
        numero=160;
    }
    if ((DATO2=='B') || (DATO2=='b')){
        numero=176;
    }
    if ((DATO2=='C') || (DATO2=='c')){
        numero=192;
    }
    if ((DATO2=='D') || (DATO2=='d')){
        numero=208;
    }
    if ((DATO2=='E') || (DATO2=='e')){
        numero=224;
    }
    if ((DATO2=='F') || (DATO2=='f')){
        numero=240;
    }
    if (DATO2=='0'){
        numero=0;
    }
    if (DATO2=='1'){
        numero=16;
    }
    if (DATO2=='2'){
        numero=32;
    }
    if (DATO2=='3'){
        numero=48;
    }
```

```

    }
    if (DATO2=='4'){
        numero=64;
    }
    if (DATO2=='5'){
        numero=80;
    }
    if (DATO2=='6'){
        numero=96;
    }
    if (DATO2=='7'){
        numero=112;
    }
    if (DATO2=='8'){
        numero=128;
    }
    if (DATO2=='9'){
        numero=144;
    }

    cuenta=cuenta+numero;
    return(cuenta);

}

void checa_sensor(){
    if(damage==1){
        if((filamento1 == 0) && (filamento2 == 0) && (filamento3 == 0)){

            filamento=0;
            damage=0;
            filamentoA= filamento1;
            filamentoB= filamento2;
            filamentoC= filamento3;

            //          escribe_eepromInt(&filamentoA, filamento1);
            //          escribe_eepromInt(&filamentoB, filamento2);
            //          escribe_eepromInt(&filamentoC, filamento3);

        }

        if ((sensor1 == 0) || (sensor2 == 0) || (sensor3 == 0) || (sensor4 == 0)){

            sensorn=0;

```

```

        damage=0;
        sensorA=sensor1;
        sensorB=sensor2;
        sensorC=sensor3;
        sensorD=sensor4;

        //escribe_eepromInt(&sensorA, sensor1);
        //escribe_eepromInt(&sensorB, sensor2);
        //escribe_eepromInt(&sensorC, sensor3);
        //escribe_eepromInt(&sensorD, sensor4);

    }

}

int aceleracion(int velfin,int velin,n){
//escribe_eepromInt(&acel2[n],velin-velfin);
//escribe_eepromInt(&vel2[n],velfin);
//printf("acel:%d\n",velin-velfin);
return(velin-velfin);

}

void reporte(){
int opcion;
Ri=0;
printf(" *** Proyectos de ingenieria. ***\n");
printf(" Sistema de seguridad\n");
printf("« *** Menu de reporte ***\n »");
do
{
printf("\n*****\n");
printf("* 1.Reporte * \n");
printf("* 2.Limpia sensores * \n");
printf("* 3.no implementad * \n");
printf("* 4.no implementad * \n");
printf("* 5.no implementad * \n");
printf("* 6.Salir * \n");
printf("*****\n");

scanf("%d", &opcion);

switch (opcion)
{
case 1: repor2();break;

```

```

        case 2: borra(); break;
        case 3: noimp(); break;
        case 4: noimp(); break;
        case 5: noimp(); break;
    }
}while(opcion!=6);
RI=0;

}

void repor2()
{
int x;
x=1;

printf(" *** Proyectos de ing. ***\n");
printf("      Reporte de sensores \n");
printf(" *****\n");

    printf("sensor1:%d ,sensor2:%d\n",sensorA,sensorB);
    printf("sensor3:%d ,sensor4:%d\n",sensorC,sensorD);
printf("filamento1:%d ,filamento2:%d\n",filamentoA,filamentoB);
printf("filamento3:%d\n",filamentoC);
printf("damage:%d\n",damage);
printf("volcadura:%d\n",filamento);
    printf("choque:%d\n",sensorn);
    printf("desaceleracion:\n");
    while (x!=11){
    printf("%d,",acel2[x]);
    x++;
        }

printf("\n");
while(x!=16){
printf("%d,",acel2[x]);
x++;
    }

x=1;
printf("\n");

//    }

}

```

```

    void borra(){
    int i;
    for(i=1;i==15;i++){
    //escribe_eepromInt(&acel2[i], temp);
    //escribe_eepromInt(&vel2[i], temp);
    acel2[i]=0;

        }

    filamentoA=1;
    filamentoB=1;
    filamentoC=1;
    sensorA=1;
    sensorB=1;
    sensorC=1;
    sensorD=1;

    //escribe_eepromInt(&filamentoA, temp);
    //escribe_eepromInt(&filamentoB, temp);
    //escribe_eepromInt(&filamentoC, temp);
    //escribe_eepromInt(&sensorA, temp);

//escribe_eepromInt(&sensorB, temp);
//escribe_eepromInt(&sensorC, temp);
    //escribe_eepromInt(&sensorD, temp);
    damage=1;
    sensorn=1;
    filamento=1;

    }

void noimp(){
    printf("<<<FUNCION NO IMPLEMENTADA>>\n");
}

void main(){
    int k;
    int velin;
    int velfin;
    int i;

    k=0;
    velin=0;
    velfin=0;
    inicializa();
    borra();
    inicializa_serie();

    while(1){

```

```

//checa_sensor();
//checa_sensor2();
while(damage!=0){
    k++;
    if(k==16) {k=1;}
    checa_sensor();
    velin=checa_vel();
    // printf("velinicial:%d\n",velin);

    for (i=0;i==1000;i++){

    }

    velfin=checa_vel();
    // printf("velocidadfinal:%d\n",velfin);
    acel=aceleracion(velin,velfin,k);
    acel2[k]=acel;

    // printf("valor n:%d\n",k);

    if(filamento==0){
        BUZZER=0;
        //printf("volcadura\n");
        damage=0;

    }

    if(sensorn==0){
        if(acel>= desacel){
            BUZZER=0;
            //printf("choque\n");
            damage=0;

        }

    }

else if (damage==OFF) {

    P2=0xFF;
    sensorn=OFF;

    filamento=OFF;
    BUZZER=OFF;

    }

    if(RI==1)
    reporte();

    }

    if(RI==1)
    reporte();

```

}

}

3.2.4 Integración del Sistema

Después de realizar las tres etapas del diseño, se tuvo que acoplar al microcontrolador las otras 2 etapas (los sensores y la interfaz de comunicación con el auto) por lo que fue necesario alimentarlas y realizar las interfaces entre tarjetas.

3.2.4.1 Alimentación de tarjetas

Dicha alimentación es de 12V que es lo que da la batería del automóvil y es regulada por medio de 2 LM7805 (reguladores positivos de voltaje a 5V.) para brindar la alimentación TTL que los circuitos integrados necesitan, los 12V se obtienen del pin de alimentación del puerto OBDII por lo que siempre están alimentadas las tarjetas. Para que ambas tarjetas tengan en sus terminales los 12V se colocaron en paralelo.

3.2.4.2 Alimentación de emergencia

Es una batería conectada a los circuitos que por medio de un relevador entra en funcionamiento en caso de que la alimentación convencional fallara. Dicha batería es de 9V recargable con un circuito que constantemente la está cargando para que en cualquier falla de energía entre y funcione correctamente por tiempo prolongado.

3.2.4.3 Conectividad

Para la conectividad se tuvieron que realizar 3 cables.

El primero de los cables es la interfaz entre la computadora y la tarjeta del microcontrolador que por medio del puerto serial se comunican, este es un cable directo con conectores DB9 en ambas terminales, pero para conectarse a la computadora es hembra y al microcontrolador macho.

Para la comunicación entre el bus del automóvil y el microcontrolador también fue necesario realizar un cable serial con conectores DB9M-DB9M de tal manera que las tarjetas se pudieran conectar para obtener la información del bus del automóvil.

La última interfaz es la que conecta la tarjeta OBDII con el puerto del mismo nombre, que se realizó por medio de terminales y headers para su correcta y fácil conexión.

3.3 Desarrollo e implementación

3.3.1 Introducción

Después de contar con un diseño integral del proyecto, es necesario realizar pruebas que constaten su buen funcionamiento por lo que es necesaria la implementación del diseño. Se llevó a cabo el desarrollo de ambas tarjetas, primero con el fin de obtener la velocidad del automóvil por medio del envío de un código por la hyperterminal de la computadora y después realizar esta misma lectura y la de los sensores por medio de la tarjeta del microcontrolador.

En ambos casos las tarjetas necesitan funcionar óptimamente para realizar de forma integral el proyecto, ya que con la buena integración de estas se realizan pruebas que validen y robustezcan el sistema evitando fallas eventuales no contempladas en el diseño de Hardware y software.

3.3.2 Desarrollo de tarjetas

3.3.2.1 OBDII

La Figura 3.3.1 muestra la tarjeta de comunicación con el OBDII. Los resultados obtenidos en la computadora al ser conectada dicha tarjeta y cómo se obtuvo la velocidad se muestran de la Figura 3.3.4 a la Figura 3.3.7.

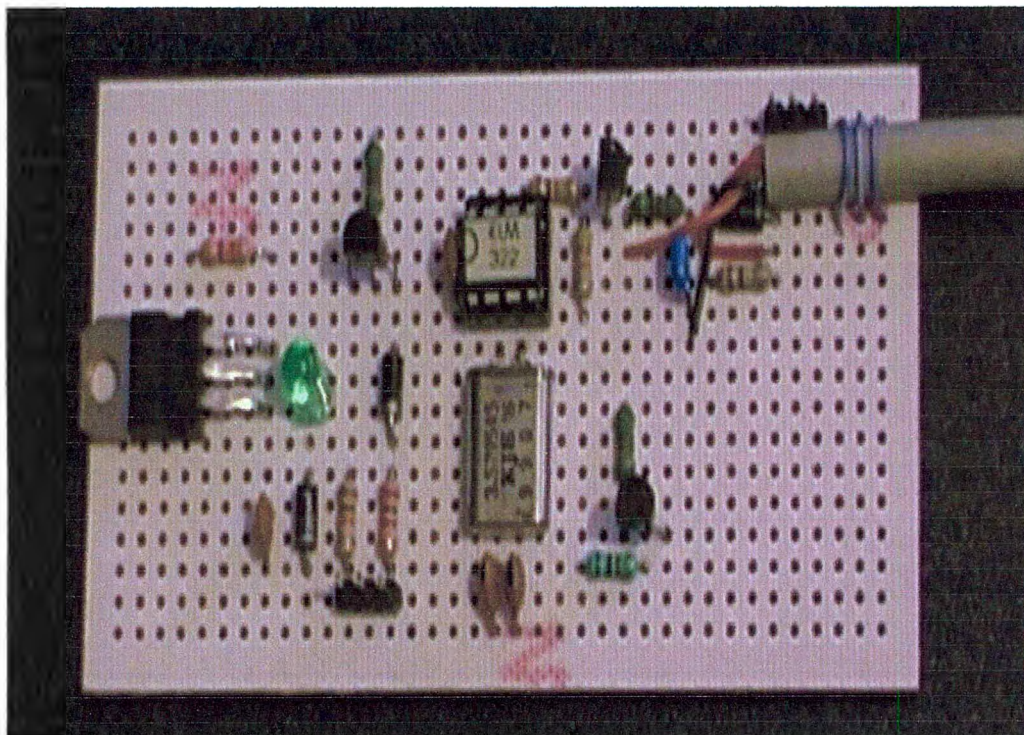


Figura 3.3.1

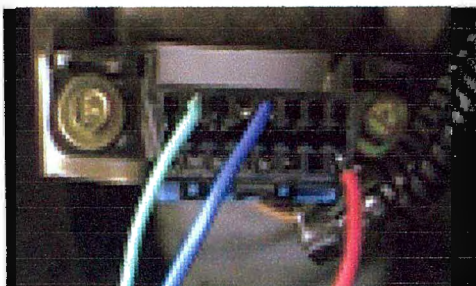


Figura 3.3.2

El bus del automóvil, Puerto OBD II



Figura 3.3.3

Conexión serial, Interfaz entre la computadora y el bus del automóvil.

La lectura del puerto OBDII del automóvil donde se ha obtenido la velocidad en hexadecimal que es equivalente a kilómetros por hora. A continuación se presentan 2 pruebas que validan el funcionamiento de la tarjeta y el código.

A 50km/h

Hyperterminal: > 010d //Código para obtener la velocidad en vehículos General Motors
>1 0D 31// Últimos 2 dígitos velocidad en km/h en Hexadecimal.
31h = 49d



Figura 3.3.4



Figura 3.3.5

A 35 km/h

Hyperterminal: > 010d //Código para obtener la velocidad en vehículos General Motors
>1 0D 31// Últimos 2 dígitos velocidad en km/h en Hexadecimal.
23h = 35d



Figura 3.3.6

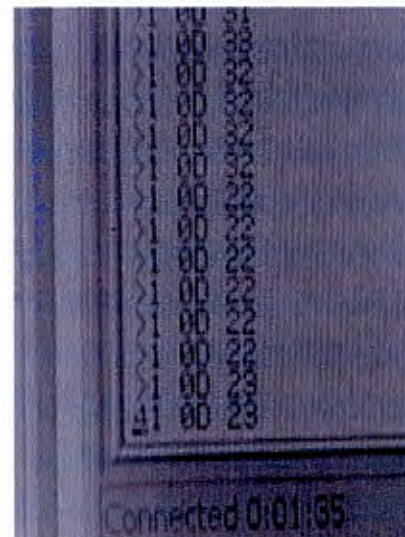


Figura 3.3.7

Ambas pruebas se realizaron en un vehículo General Motors Pontiac.

3.3.2.2 Microcontrolador

La tarjeta del microcontrolador aparece a continuación:

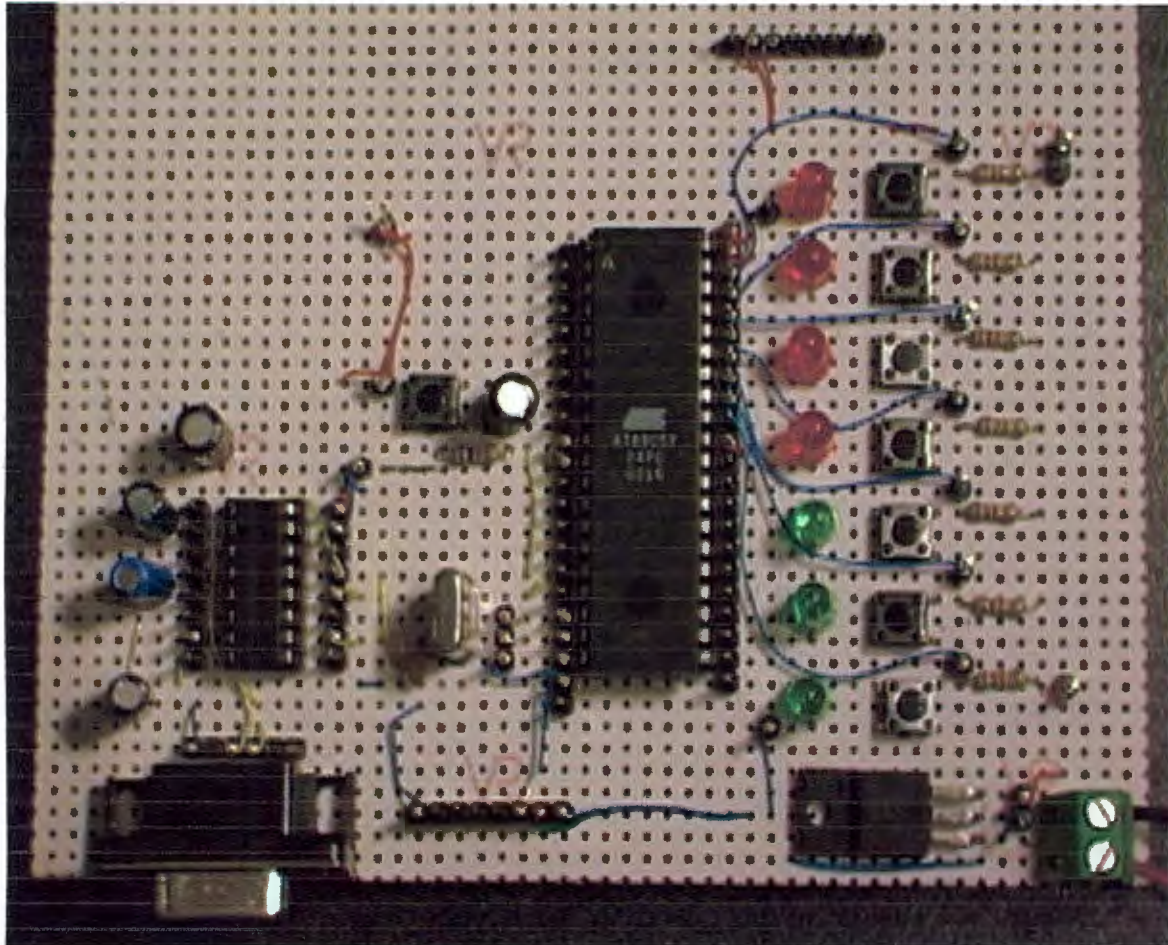


Figura 3.3.8

Esta tarjeta comprende la alimentación, la comunicación serial y los sensores además del microcontrolador con su configuración pertinente.

A continuación se presentan imágenes que corresponden a la ejecución del programa.

El menú permite varias acciones y a este se puede acceder presionando la tecla "z", entre las funciones que puede desempeñar, se encuentran las de presentar el estado actual de todos los sensores y la tabla de aceleraciones de los últimos segundos, el limpiar el estado de los sensores para nuevas mediciones y salir del programa.

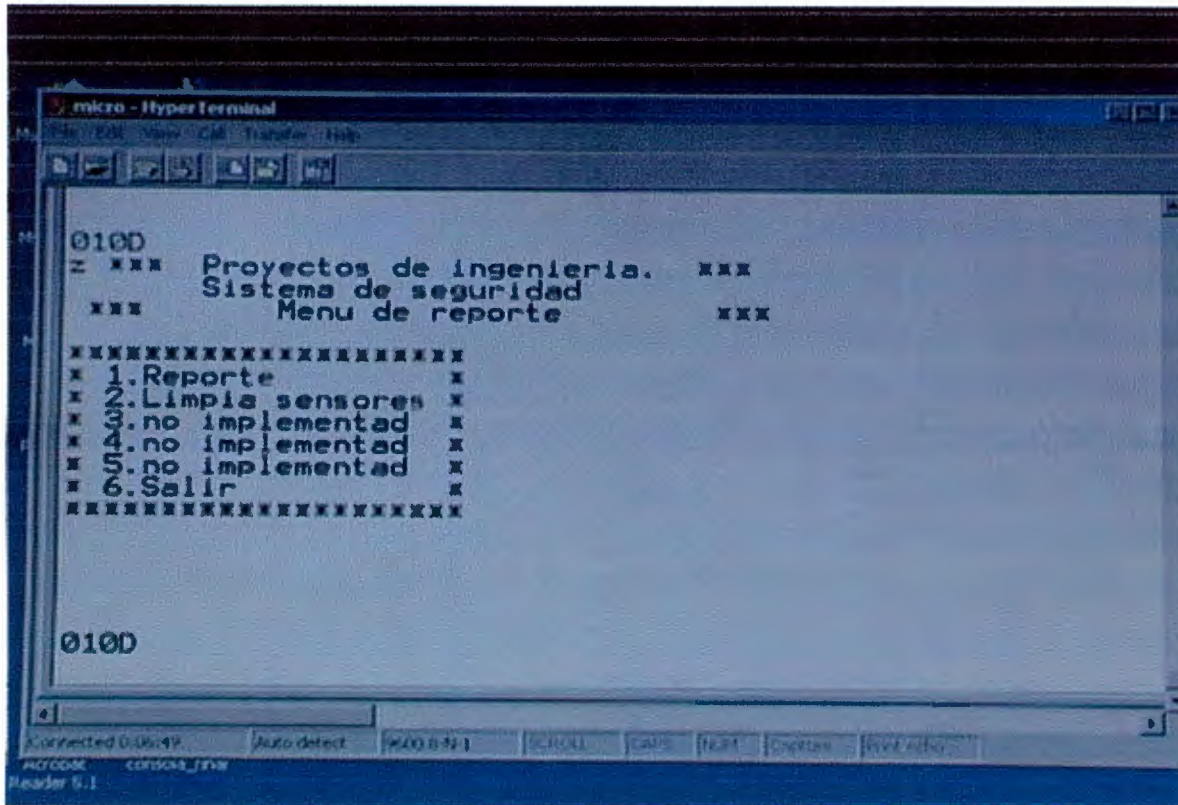


Figura 3.3.9

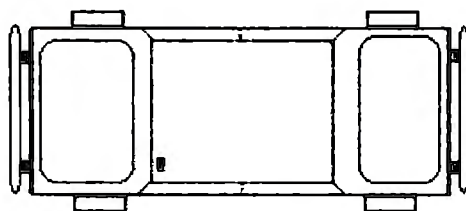
3.3.3 Implementación.

En la Figura 3.3.10, se presenta la ubicación que deben tener los elementos del sistema para su implementación.

Adicionalmente, se debe mencionar el lugar donde se colocarán las tarjetas, el GPS y la batería de emergencia, que serán:

Para las tarjetas el lugar ideal es la parte baja del tablero por ser uno de los lugares más seguros del automóvil, el GPS y la batería de alimentación alternativa deben de colocarse debajo del asiento del conductor, por las misma razón anterior.

Método directo.



Sensores de contacto.

Filamentos de Cobre.

Método Indirecto.

Puerto OBD II.

Figura 3.3.9

El conocimiento de los diferentes elementos que forman parte del diseño es fundamental para corregir fallas en la implementación y el funcionamiento, y en el caso de haberlas poder corregirlas con mayor rapidez y facilidad.

El terminar el capítulo hablando sobre el lugar donde se deben colocar las diferentes piezas que constituyen los métodos directo e indirecto, además de donde se colocarán las tarjetas, hace que el lector pueda tener claro el diseño (donde van los elementos) y cuales son los requerimientos mínimos del sistema que se armó.

En el capítulo IV, se abordarán temas relacionados con las diferentes pruebas que se llevaron acabo con las tarjetas y el auto, ya que aunque el diseño se encuentra terminado por completo es muy importante conocer los resultados obtenidos con lo presentado en este capítulo.

De esta manera en el siguiente capítulo se verá la integración del sistema y sus diferentes resultados según el tipo de impacto sufrido.

4 Resultados

4.1 Introducción

Las pruebas efectuadas a este proyecto se documentan de acuerdo a las etapas que se han terminado.

El primer circuito terminado en este proyecto fue la tarjeta de comunicación con el bus del vehículo. En esta etapa se logró una comunicación entre el bus del vehículo y una computadora a través de un puerto serie de ésta última.

Después de esta tarjeta, se terminó la tarjeta del microcontrolador permitiendo verificar los resultados del programa realizado.

Los cables, los sensores y la alimentación también fueron valorados y verificados en su funcionamiento.

4.2 Pruebas

4.2.1 Comunicación con el bus del vehículo

En esta etapa, se logró comunicar una computadora con el vehículo a través del puerto serial de la computadora. El bus utilizado fue el J1850 implementado en los vehículos General Motors.

Con la aplicación de Hyper Terminal de Windows, se logró comunicar la computadora con el bus y comparar la velocidad del velocímetro del vehículo con la obtenida en la computadora. El vehículo que se utilizó para estas pruebas fue un sunfire de GM, el cual estaba levantado con gatos hidráulicos con lo cual se pudo acelerar sin necesidad de estar en movimiento.

Al inicio, se realizaron pruebas con la interfase de comunicación con el bus del auto que ayudaron a determinar, la velocidad a la que se debía hacer la transmisión, si la señal

tenía algún bit de paridad y que controlaba el flujo, por lo que se obtuvo que para el sistema era necesaria la siguiente configuración.

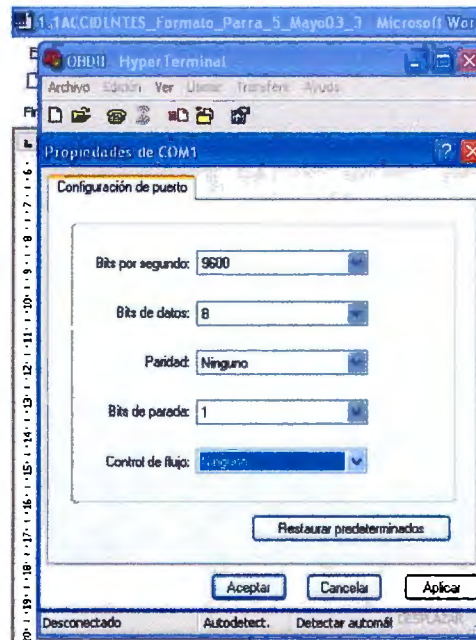


Figura 4.2.1

Dicha configuración corresponde al puerto COM1 (serial) de la computadora y puede o no tener eco para recibir la información.

Después realizamos la prueba presentada en el capítulo 3 que en esta ocasión se presenta nuevamente para mostrar la comunicación de la tarjeta.

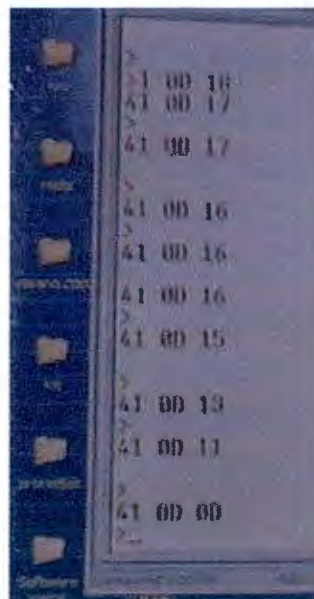


Figura 4.2.2

Aquí se puede observar como varía la velocidad del automóvil de 24 Km/h hasta 13 Km/h y el modo en el que la información se muestra, al principio se conecta automáticamente al bus del auto y después de enviarle la palabra 010d obtenemos 8 bits de los cuales los

últimos 2 son la velocidad en hexadecimal y en km/h, así mismo envía también el bus del auto un salto de carro posterior a la respuesta de los datos por lo que con esta prueba, se determinó que era necesario el leer sólo los últimos tres bits para obtener la velocidad y el salto de carro y de esta forma estar enviándole después de cada salto de carro un nuevo requerimiento de la velocidad.

4.2.2 Tarjeta del microcontrolador

A continuación de las pruebas realizadas a la primera tarjeta, se procedió a terminar la tarjeta correspondiente a la toma de decisiones.

Lo que se necesitaba verificar en esta tarjeta era:

1. La alimentación de la tarjeta fuera la correcta +5V y tierra reguladas con un 7805.
2. El circuito de reset funcionará correctamente para el microcontrolador.
3. El cristal oscilará a 11.059 MHz para obtener la velocidad de 9600 baudios necesaria para la comunicación serial con la computadora.
4. La comunicación serial se realizara correctamente con la configuración hecha al MAX 232 encargado de dicha tarea.
5. Los sensores cumplieran con su objetivo enviando un pulso en alto cuando no se encontraban accionados y al accionarlos un pulso en bajo.
6. El envío del pulso después de validar algún siniestro.
7. El buen funcionamiento del programa.

De lo anterior se encontraron los siguientes resultados:

La tarjeta se alimentaba al igual que la del bus del automóvil con una batería de 12 V, pero la regulación resultó de 4.6V al microcontrolador por lo que resulto satisfactorio en un 92%.

En lo concerniente al circuito de reset, no se encontró ningún problema.

Al evaluar el cristal, resultó que había variaciones en la frecuencia de oscilación que fluctuaban entre los 11.2 y los 11 MHz, por los que se pensó en cambiar los capacitores de 33pF a otro valor cercano, pero al realizar una prueba preliminar se mostró que resultaba igual por lo que se aceptó como válido su funcionamiento.

Para comprobar la comunicación serial, primero se programo en el microcontrolador un programa sencillo cuya única función fuera cumplir con la comunicación y al probarlo, no se tenían los resultados esperados, por lo que se dedujo que el cable con el que se estaba realizando la comunicación era el motivo de la falla, después de recortarlo a aproximadamente 1m de largo se obtuvieron buenos resultados y se pudo seguir adelante.

Los sensores cumplieron con su objetivo siguiendo la configuración del capítulo anterior sin ningún contratiempo.

El envío de un pulso de bajada después de detectar algún siniestro funcionó correctamente y permitió realizar varias pruebas con el programa.

4.2.3 El programa

Para poder validar el funcionamiento del programa hubo que tener terminada la tarjeta del microcontrolador y la comunicación lo que permitió hacer pruebas, para determinar lo que era necesario realizar.

En primer lugar se presenta el código de prueba donde se despliega y verifica la conversión de las velocidades a km/h, y el cálculo de la aceleración.

```
#include <c:\franklin\include\reg52.h>
#include <c:\franklin\include\stdio.h>

//definicion de prototipos
int aceleracion(int velfin,int velin,n);
int checa_vel();
void checa_sensor();

//void checa_sensor2();
void reporte();
void repor2();
void borra();
void noimp();

/***** habilita escritura a EEPROM *****/
#define HABILITA_EE() WMCON|=0x10

/***** deshabilita escritura a EEPROM *****/
#define DESHABILITA_EE() WMCON&=0x08

/***** espera a que se escriba dato en EEPROM *****/
//void escribe_eepromInt(int *direccion, int valor);
#define ESPERA_EEW() while((WMCON&0x02)==0)
#define ON 0
#define OFF 1
#define BUZZER P20
#define sensor1 P21
#define sensor2 P22
#define sensor3 P23
#define sensor4 P24
#define filamento1 P25
#define filamento2 P26
#define filamento3 P27

//variables globales
int n;
int filamento;
int sensorn;
int acel;
//int T;
//int TIEMPO;
```



```

int desacel;
int damage;

int acel2[15];

idata int sensorA;
idata int sensorB;
idata int sensorC;
idata int sensorD;
idata int filamentoA;
idata int filamentoB;
idata int filamentoC;

void inicializa()
{
    //T=0;
    desacel=30;
    BUZZER=1;
    //TIEMPO=0;
    WMCON=0x08;           /*habilita EEPROM interna como XRAM*/
    P20=0x0FF;
    TH0=0xFC;
    TL0=0x17;
    ET0=1;
    TR0=1;
    filamento=1;
    sensorn=1;
    damage=1;

}

/*
void escribe_eepromInt(int *direccion, int valor)
{
    unsigned char x;
        HABILITA_EE();
        x=(char)valor;
        *direccion=x;
        ESPERA_EEW();

        valor>>=8;
        x=(char)valor;
        *(direccion+1)=x;
        ESPERA_EEW();

        DESHABILITA_EE();
        //int numero, dato;

```

```

//xdata int eenumero;
//numero=10;
//escribe_eepromInt(&eenumero, numero);
//dato=eenumero;
}

*/

void timercero() interrupt 1 using 2
{
TF0=0;
TH0=0xFC;
TL0=0x17;

//base de tiempo 1 milisegundo
// if (TIEMPO==1) //tiempo entre velocidades
// T;

}

void inicializa_serie()
{
SCON=0x52;
TMOD=0x20;
TR1=1;

TL1=0xFD; //FD 9600 bps@11.059MHz
TH1=0xFD; //FA 4600 bps@11.059MHz
}

int checa_vel(){
int cuenta;
int numero;
char DATO;
char DATO2;

cuenta=0;
n=0;
putchar('\n');
putchar('\n');
putchar('\n');
putchar('0');
putchar('1');
putchar('0'); //Manda dato de lectura de velocidad
putchar('D');

```

```

putchar('\n');

do{
    DATO=getchar();
    if (DATO == ' '){ //busca los espacios en la cadena

        n++;}
        if((DATO=='z') || (DATO=='Z')){reporte();}
    if (n==2){
        DATO2=getchar();
    if((DATO=='z') || (DATO=='Z')){reporte();}
        DATO=getchar(); // guarda velocidad en hexadecimal
        if((DATO=='z') || (DATO=='Z')){reporte();}

            getchar();
        }

}while(n!=2); //conversion de velocidad de hexadecimal a decimal
if ((DATO=='A') || (DATO=='a')){
    numero=10;
}
if ((DATO=='B') || (DATO=='b')){
    numero=11;
}
if ((DATO=='C') || (DATO=='c')){
    numero=12;
}
if ((DATO=='D') || (DATO=='d')){
    numero=13;
}
if ((DATO=='E') || (DATO=='e')){
    numero=14;
}
if ((DATO=='F') || (DATO=='f')){
    numero=15;
}

if (DATO=='0'){
    numero=0;
}
if (DATO=='1'){
    numero=1;
}
if (DATO=='2'){
    numero=2;
}

```

```
if (DATO=='3'){
numero=3;
}
if (DATO=='4'){
numero=4;
}
if (DATO=='5'){
numero=5;
}
if (DATO=='6'){
numero=6;
}
if (DATO=='7'){
numero=7;
}
if (DATO=='8'){
numero=8;
}
if (DATO=='9'){
numero=9;
}

cuenta=cuenta+numero;
if ((DATO2=='A') || (DATO2=='a')){
numero=160;
}
if ((DATO2=='B') || (DATO2=='b')){
numero=176;
}
if ((DATO2=='C') || (DATO2=='c')){
numero=192;
}
if ((DATO2=='D') || (DATO2=='d')){
numero=208;
}
if ((DATO2=='E') || (DATO2=='e')){
numero=224;
}
if ((DATO2=='F') || (DATO2=='f')){
numero=240;
}
if (DATO2=='0'){
numero=0;
}
if (DATO2=='1'){
numero=16;
}
if (DATO2=='2'){
numero=32;
}
if (DATO2=='3'){
```

```
numero=48;
    }
    if (DATO2=='4'){
        numero=64;
    }
    if (DATO2=='5'){
        numero=80;
    }
    if (DATO2=='6'){
        numero=96;
    }
    if (DATO2=='7'){
        numero=112;
    }
    if (DATO2=='8'){
        numero=128;
    }
    if (DATO2=='9'){
        numero=144;
    }

    cuenta=cuenta+numero;
    return(cuenta);
}

void checa_sensor(){
    if(damage==1){
        if((filamento1 == 0) && (filamento2 == 0) && (filamento3 == 0)){

            filamento=0;
            damage=0;
            filamentoA= filamento1;
            filamentoB= filamento2;
            filamentoC= filamento3;

            //          escribe_eepromInt(&filamentoA, filamento1);
            //          escribe_eepromInt(&filamentoB, filamento2);
            //          escribe_eepromInt(&filamentoC, filamento3);

        }

        if ((sensor1 == 0) || (sensor2 == 0) || (sensor3 == 0) || (sensor4 == 0)){
```

```

        sensorn=0;
        damage=0;
        sensorA=sensor1;
        sensorB=sensor2;
        sensorC=sensor3;
        sensorD=sensor4;

        //escribe_eepromInt(&sensorA, sensor1);
        //escribe_eepromInt(&sensorB, sensor2);
        //escribe_eepromInt(&sensorC, sensor3);
        //escribe_eepromInt(&sensorD, sensor4);

    }

}

int aceleracion(int velfin,int velin,n){
//escribe_eepromInt(&acel2[n],velin-velfin);
//escribe_eepromInt(&vel2[n],velfin);
printf("acel:%d\n",velin-velfin);
return(velin-velfin);

}

void reporte(){
int opcion;
RI=0;
printf(" *** Proyectos de ingenieria. ***\n");
printf(" Sistema de seguridad\n");
printf("« *** Menu de reporte ***\n »");
do
{
printf("\n*****\n");
printf(" * 1.Reporte * \n");
printf(" * 2.Limpia sensores * \n");
printf(" * 3.no implementad * \n");
printf(" * 4.no implementad * \n");
printf(" * 5.no implementad * \n");
printf(" * 6.Salir * \n");
printf("*****\n");

scanf("%d", &opcion);

```

```

        switch (opcion)
        {
            case 1: repor2();break;
            case 2: borra(); break;
            case 3: noimp(); break;
            case 4: noimp(); break;
            case 5: noimp(); break;
        }
    }while(opcion!=6);
    RI=0;

}

void repor2()
{
int x;
x=1;

printf(" *** Proyectos de ing. ***\n");
printf("      Reporte de sensores \n");
printf(" *****\n");

    printf("sensor1:%d ,sensor2:%d\n",sensorA,sensorB);
    printf("sensor3:%d ,sensor4:%d\n",sensorC,sensorD);
printf("filamento1:%d ,filamento2:%d\n",filamentoA,filamentoB);
printf("filamento3:%d\n",filamentoC);
printf("damage:%d\n",damage);
printf("volcadura:%d\n",filamento);
    printf("choque:%d\n",sensorn);
    printf("desaceleracion:\n");
    while (x!=11){
        printf("%d,",acel2[x]);
        x++;
    }

printf("\n");
while(x!=16){
printf("%d,",acel2[x]);
x++;
}

x=1;
printf("\n");

// }

}

```

```

    void borra(){
    int i;
    for(i=1;i==15;i++){
    //escribe_eepromInt(&acel2[i], temp);
    //escribe_eepromInt(&vel2[i], temp);
    acel2[i]=0;

        }

    filamentoA=1;
    filamentoB=1;
    filamentoC=1;
    sensorA=1;
    sensorB=1;
    sensorC=1;
    sensorD=1;

    //escribe_eepromInt(&filamentoA, temp);
    //escribe_eepromInt(&filamentoB, temp);
    //escribe_eepromInt(&filamentoC, temp);
    //escribe_eepromInt(&sensorA, temp);

    //escribe_eepromInt(&sensorB, temp);
    //escribe_eepromInt(&sensorC, temp);
    //escribe_eepromInt(&sensorD, temp);
    damage=1;
    sensorn=1;
    filamento=1;

    }

void noimp(){
    printf("<<<FUNCION NO IMPLEMENTADA>>>\n");
}

void main(){
    int k;
    int velin;
    int velfin;
    int i;

    k=0;
    velin=0;
    velfin=0;
    inicializa();

```



```

inicializa_serie();

while(1){
    //checa_sensor();
    //checa_sensor2();
while(damage!=0){
    k++;
    if(k==16) {k=1;}
    checa_sensor();
    velin=checa_vel(); printf("velinicial:%d\n",velin);

    for (i=0; i==1000; i++){

        }

    velfin=checa_vel();
    printf("velocidadfinal:%d\n",velfin);
    acel=aceleracion(velin,velfin,k);
    acel2[k]=acel;
    // printf("valor n:%d\n",k);

    if(filamento==0){
        BUZZER=0;
        printf("volcadura\n");
        damage=0;

        }

    if(sensorn==0){
        if(acel>= desacel){
            BUZZER=0; printf("choque\n"); damage=0;
        }

        }

    else if (damage==OFF) {

        P2=0xFF;
        sensorn=OFF;

        filamento=OFF;
        BUZZER=OFF;

        }

    if(RI==1)
        reporte();

```

```

    }

    if(RI==1)
    reporte();

}

```

Así, el programa anterior demostró su capacidad de convertir las velocidades en hexadecimal a velocidades en km/h y el cálculo de la aceleración del vehículo. Además de verificar cuales son los sensores activados siguiendo el diagrama de flujo del programa presentado en el capítulo concerniente a diseño.

Además se creó el reporte del estado de todos los sensores con el programa y un menú.

```

printf(" *** Proyectos de ing. ***\n");
printf("      Reporte de sensores \n");
printf(" *****\n");

printf("sensor1:%d ,sensor2:%d\n",sensorA,sensorB);
printf("sensor3:%d ,sensor4:%d\n",sensorC,sensorD);
printf("filamento1:%d ,filamento2:%d\n",filamentoA,filamentoB);
printf("filamento3:%d\n",filamentoC);
printf("damage:%d\n",damage);
printf("volcadura:%d\n",filamento);
printf("choque:%d\n",sensorn);

```

Donde al validar algún siniestro damage se enciende (todo se enciende en bajo) y queda activado, y muestra el resultado de volcadura o choque según lo que haya ocasionado el daño.

4.3 Integración del sistema.

Después realizar todas las pruebas anteriormente mencionadas se integraron las tarjetas y el programa final para llevar acabo una prueba final.

Cuando se realizó esta prueba, se presentaron dificultades, sobretodo para obtener imágenes fijas, pues en esta ocasión el carro se encontraba en movimiento y por lo tanto la cámara vibraba, pero a continuación se presenta la imagen de la conexión de las tarjetas con el puerto OBDII.

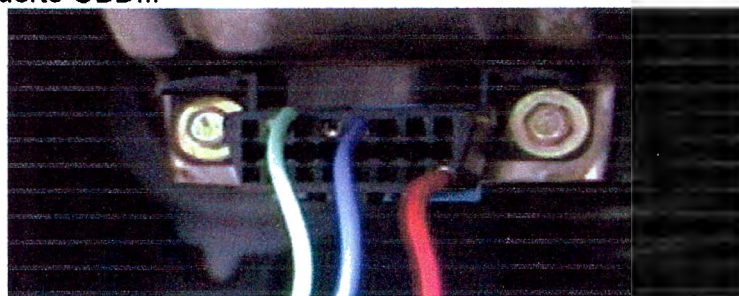


Figura 4.3.1

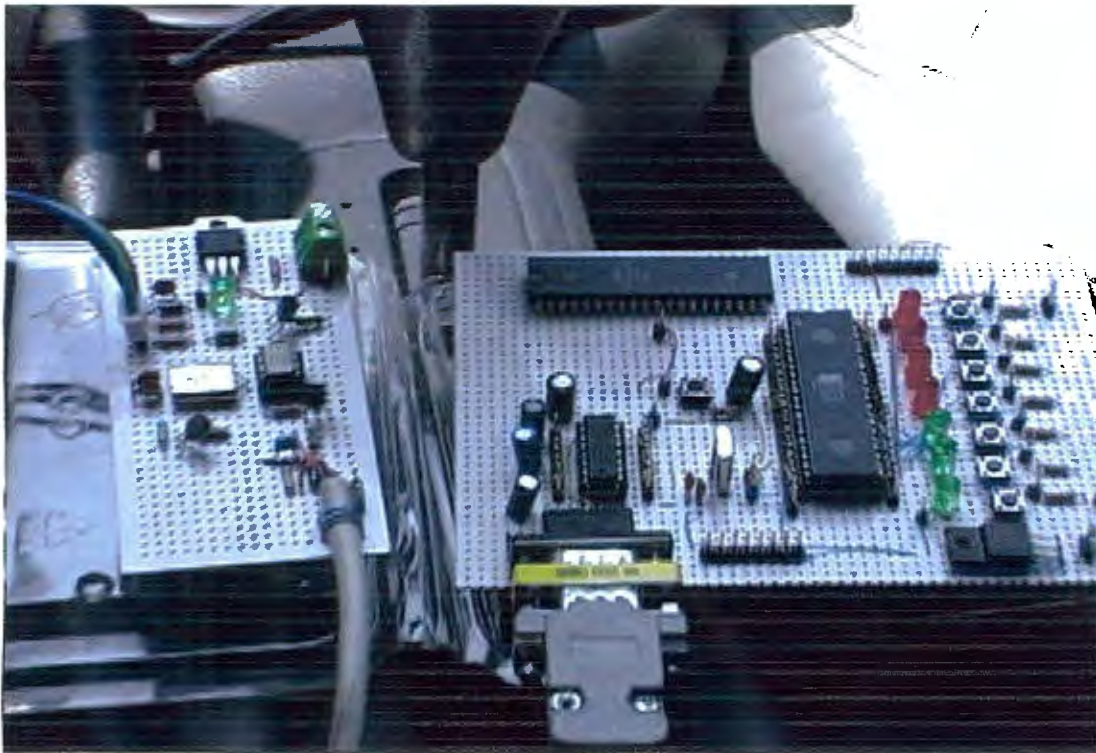


Figura 4.3.2

Con esta integración, se pudieron realizar pruebas para validar los diferentes tipos de siniestros y la correcta toma de aceleraciones.

Posteriormente, se conectó la tarjeta del microcontrolador a la computadora para verificar que sí estuvieran las diferentes aceleraciones y los sensores accionados, y al obtener resultados satisfactorios se concluyó con los objetivos del proyecto.

Adicionalmente, se realizaron algunas pruebas con el GPS para constatar la flexibilidad y modularidad del sistema y su correcto funcionamiento.

Al concluir con esto, fue necesario conocer el rumbo del proyecto a futuro y las conclusiones obtenidas al realizarlo, por lo se creo el siguiente capítulo.

5 Conclusiones, perspectivas, recomendaciones y trabajo futuro.

5.1 Introducción

Las aplicaciones para este proyecto son especialmente para compañías que se dedican al rastreo de vehículos, así como compañías fabricantes de autos. Es decir, el sistema tiene mercado a nivel aseguradoras y fabricantes de autos que deseen ensamblarlo en serie.

Después de un año de desarrollo del proyecto, se obtuvo experiencia acerca de la elaboración, diseño e implementación del mismo. Aprendiendo acerca de protocolos, estándares y aplicaciones del bus de los automóviles.

Adicionalmente se obtuvo un mayor conocimiento sobre los microcontroladores y la programación, con todo lo anterior, se obtuvieron las siguientes conclusiones:

5.2 Conclusiones

- El sistema se muestra útil en los casos en los cuales los accidentes suceden.

En cualquier caso, todos están siempre expuestos a los accidentes pues por más medidas de seguridad que se toman a diario, siempre se ven rebasadas por fallas humanas, por lo que el sistema es válido siempre ya que en cualquier momento ocurre un siniestro.

- Se espera que el sistema aumente la velocidad de atención a los medios de urgencias después de un siniestro grave. Igualmente la asistencia de las aseguradoras será más pronta.

Si se sabe que bastan con menos de 10 minutos sin respirar para que el cerebro, resulte dañado, es muy importante que los servicios de urgencia a diario mejoren su tiempo de respuesta y salven vidas a toda costa por lo que el proyecto, es un pequeño precio por un bien mucho mayor.

- Se demostró que es posible integrar a un automóvil un sistema sencillo y de bajo costo capaz de analizar el impacto y avisar a quien corresponda en caso necesario.

Como prioridad siempre se contempló el mantener el costo del proyecto bajo para lograr que su implantación sea una posibilidad grande. De modo que se pensó en la solución mejor, y no en la más enredada y costosa, ya que una solución simple cumplía perfectamente con el objetivo del proyecto.

- La modularidad del sistema permite su adaptación y escalabilidad, permitiendo así la integración de dispositivos como GPS vía celular o radio.

Con motivo de la gran variedad que existe en el mercado de GPS, se considero que el sistema resultará universal para todos los equipos GPS y así continuar con el objetivo de que todos los automóviles puedan además de ser localizados en algún caso de robo o secuestro, estén seguros en los casos de siniestro.

5.3 Perspectivas

Es necesario buscar la manera de hacer el proyecto aun más universal, buscando la manera de unificar los estándares de comunicación con el bus de automóvil para que sea válido para todas las marcas. En perspectiva, se empieza a abrir campo para el sistema debido a la necesidad de ventajas competitivas que necesitan las empresas de seguros para darle un valor agregado a sus productos.

5.4 Trabajo Futuro

Al demostrar su eficacia, se espera que alguna ensambladora o aseguradora, contemple instalarlo en serie.