

ESCUELA POLITECNICA DEL EJÉRCITO

DEPARTAMENTO DE ELECTRICA Y ELECTRONICA

CARRERA DE INGENIERIA EN ELECTRONICA,

AUTOMATIZACION Y

CONTROL

PROYECTO DE GRADO PARA LA OBTENCION DEL

TITULO EN

INGENIERIA

**“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA
ALTERNATIVO DE INYECCIÓN ELECTRÓNICA DE
GAS NATURAL COMPRIMIDO (GNC) PARA UN MOTOR
DE COMBUSTIÓN INTERNA A GASOLINA DE OCHO
CILINDROS EN V”**

PAREDES URQUIZO ALVARO RENAN

CONDO VELASTEGUI PAULO CESAR

SANGOLQUI – ECUADOR

2009

CERTIFICACIÓN

Por medio de la presente certificamos que el proyecto de grado para la obtención del título en Ingeniería Electrónica titulado “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA ALTERNATIVO DE INYECCIÓN ELECTRÓNICA DE GAS NATURAL COMPRIMIDO (GNC) PARA UN MOTOR DE COMBUSTIÓN INTERNA A GASOLINA DE OCHO CILINDROS EN V” fue desarrollado en su totalidad por los señores PAREDES URQUIZO ALVARO RENAN y CONDO VELASTEGUI PAULO CESAR.

Atentamente,

Ing. Rodolfo Gordillo
DIRECTOR

Ing. Paúl Ayala
CODIRECTOR

AGRADECIMIENTO

A Dios sobre todas las cosas por darnos la oportunidad de vivir para ser siempre mejores.

Al Dr. Galo Paredes y esposa Lcda. Gladis García por confiar siempre en mí y brindarme su apoyo en todos los aspectos de mi vida. (Álvaro)

Al Ing. Wilson Chiluisa gerente de DYCEL Cia. Ltda. Y toda su familia por su incondicional apoyo incondicional moral y económico a lo largo de toda mi carrera. (Álvaro)

Al director y codirector de tesis quienes apoyaron y fueron respaldo constante para el desarrollo del mismo.

A nuestro amigo David Santana quien constituyo un apoyo importante para el desarrollo del presente trabajo.

A todos nuestros maestros con quienes compartimos gratos momentos en el aula de clases y fuera de ellas.

Al Ing. Benjamin Condo Velastegui por confiar siempre en mí y brindarme su apoyo en todos los aspectos de mi vida. (Paulo)

DEDICATORIA

Este trabajo dedicamos a las personas mas importantes de nuestras vidas nuestros padres: Luis Paredes Arroyo y Sarita Noemí Urquizo asi como a Angel Ramiro Condo Ortiz y Blanca Cecilia Velastegui Altamirano en agradecimiento a su esfuerzo, cariño y sacrificio; de manera muy especial también dedicamos este trabajo a las personas quienes día a día han luchado con nosotros por conseguir este logro, a nuestras esposa y compañera Adriana Pamela Paredes y Teresa Marlene Sambonino asi como a la luz de nuestros ojos nuestras hijas Domenica y Mishell quienes son la inspiración para lograr todos nuestros objetivos.

PRÓLOGO

Debido al alto costo de la gasolina a los motores que funcionan con este combustible se los ha estado desechando lo cual significa desperdiciar muchos recursos en pequeñas y medianas empresas que no están en posibilidades de adquirir un nuevo vehículo o maquinaria que funcionen con motores a diesel.

Al observarse este problema se ha tomado como alternativa utilizar un sistema electrónico de inyección de GNC para impulsar estos motores puesto que este combustible alternativo es compatible con el sistema de los motores a gasolina dando a estos la oportunidad de seguir trabajando sin mayor perjuicio económico ya que la adaptación no implica un gasto significativo.

Como características el sistema permitirá la dosificación del GNC por medio de un inyector de gas el cual funcionará de acuerdo con las necesidades de la carga del motor por medio de la utilización de sensores de temperatura, presión y posición que indiquen las condiciones en que se encuentra el motor en cada instante los mismos que entregarán señales a un módulo electrónico el mismo que tendrá una respuesta a cada situación de funcionamiento.

El desarrollo de este proyecto está orientado a la mejora de las condiciones de seguridad para el manejo del gas como combustible de impulsión de motores aspecto muy importante que han dejado de lado los constructores de sistemas a gas utilizados actualmente.

Por otra parte la utilización de GNC para la impulsión de motores produce menos emisiones de gases contaminantes como son el CO y HC lo cual es una razón de suma importancia para la construcción de este prototipo.

ÍNDICE DE CONTENIDO

CAPITULO I

1.- INYECCIÓN ELECTRÓNICA.....	1
1.1.- Características de funcionamiento del sistema	2
1.2.- Elementos del sistema.....	4
1.3.- Descripción de los Elementos del Sistema.....	4
1.3.1.- Elementos motrices y/o mecánicos.....	4
1.3.2.- Elementos del sistema de combustible.....	5
1.3.3.- Elementos de obtención de señal.....	7
1.3.4.- Elementos de interfase.....	18
1.3.5.- Elementos de control.....	20
1.4.- Condiciones especiales del sistema.....	22

CAPITULO II

2.- SISTEMA CONTROLADOR.....	24
2.1.- Introducción.....	24
2.2.- Descripción del controlador digital de señales (DSC).....	26
2.4.- Comunicación de tarjeta electrónica con sensores.....	27
2.5.- Comunicación de tarjeta electrónica con actuadores.....	30

CAPITULO III

3.- DISEÑO DEL SISTEMA DE INYECCION DE COMBUSTIBLE.....	31
---	----

3.1.- Diseño Integral del sistema.....	31
3.2.- Disposición de los elementos sensores.....	32
3.3.- Disposición de válvula de salida de combustible del tanque de almacenamiento.....	32
3.4.- Disposición de válvula de ingreso de combustible a la admisión del motor	32
3.5.- Diseño del Controlador.....	33
3.5.1.- Consideraciones y dimensionamiento.....	33
3.5.2.- Diagrama de Bloques.....	38
3.5.3.- Diagrama de Flujo.....	39
3.5.4.- Diseño del software del sistema.....	40
3.5.4.1.- Introducción al desarrollo del software.....	40
3.5.4.2.- Configuración del hardware del dsc.....	41
3.6.- Programación de la tarjeta de control.....	43
3.6.1.- Configuración del hardware del dsc.....	44
3.7.- Programa del dsc.....	48
3.8.- Pruebas, detección y corrección de errores.....	52

CAPITULO IV

4.- INSTALACIÓN DEL SISTEMA

4.1.- Integración tarjeta de control – sensores y actuadores.....	62
4.2.- Instalación del sistema en el área disponible del vehiculo.....	65
4.3.- Pruebas y ajuste.....	69

4.3.1.- Pruebas.....69

4.3.2.-Corrección de errores.....73

CAPITULO V

5.- Análisis Económico.....74

CAPITULO VI

6.- Conclusiones Y Recomendaciones.....75

6.1.- Conclusiones.....75

6.2.- Recomendaciones.....77

ANEXOS.....78

CAPITULO I

INYECCIÓN ELECTRÓNICA

1.- INTRODUCCION

El cambio de carburadores a sistema de inyección se debe a que este último proporcionó una mejor manera de resolver los estándares de economía y emisión de combustible establecidos en el ámbito mundial.

Uno de los aspectos más importante es el hecho de que la inyección del combustible es un sistema más versátil para la salida de éste. Esto elimina la mayoría de los problemas del arranque en frío asociado a los carburadores. La inyección electrónica del combustible también se integra con mayor facilidad con los sistemas de control automatizados del motor, porque los inyectores se controlan más fácilmente que un carburador mecánico con agregaciones electrónicas.

Su importancia radica en su mejor capacidad para dosificar el combustible y crear un mezcla aire / combustible, muy próxima a la estequiométrica (14,7:1 para la gasolina), lo que garantiza una muy buena combustión con reducción de los porcentajes de gases tóxicos a la atmósfera. La relación estequiométrica es la proporción exacta de aire y combustible que garantiza una combustión completa de todo el combustible.

El sistema consta fundamentalmente de sensores, una unidad electrónica de control y actuadores o accionadores [3].

El funcionamiento se basa en la medición de ciertos parámetros de funcionamiento del motor, como son: la temperatura del refrigerante, el estado de carga

(sensor PAM), revoluciones del motor y sensor de posición de la mariposa (TPS) ; estas señales son procesadas por la unidad de control, dando como resultado señales que se transmiten a los accionadores (inyectores) que controlan la inyección de combustible y a otras partes del motor para obtener una combustión mejorada.

1.1.- CARACTERÍSTICAS DE FUNCIONAMIENTO DEL SISTEMA

Todo sistema de inyección electrónica requiere de varios sensores que detecten los valores importantes que deben ser medidos, para que con esta información se pueda determinar a través de un computador el tiempo de actuación de los inyectores y con ello inyectar la cantidad exacta de combustible.

Es necesario conocer algunos conceptos básicos acerca de los tipos de inyección para poder comprender cuales son las características de funcionamiento de este sistema.

✓ Inyección directa:

Es aquella, en la que los inyectores se encuentran e inyectan la gasolina directamente dentro del cilindro.

✓ Inyección Indirecta:

Es aquella, en la que los inyectores se encuentran fuera del cilindro, inyectando la gasolina al cilindro al abrirse la válvula de admisión.

✓ Diferencia entre inyección Continua e Intermitente:

En la inyección continua, los inyectores introducen el combustible de forma continua en el colector de admisión, ya dosificada y a presión; mientras que en la inyección intermitente, los inyectores introducen el combustible de forma intermitente, con lo que cada inyector se abre y cierra continuamente.

Entonces con estas consideraciones podemos describir de una manera mas clara que este proyecto se trata de una inyección electrónica intermitente indirecta monopunto la cual consta de las siguientes partes.

INYECCION ELECTRONICA DE COMBUSTIBLE

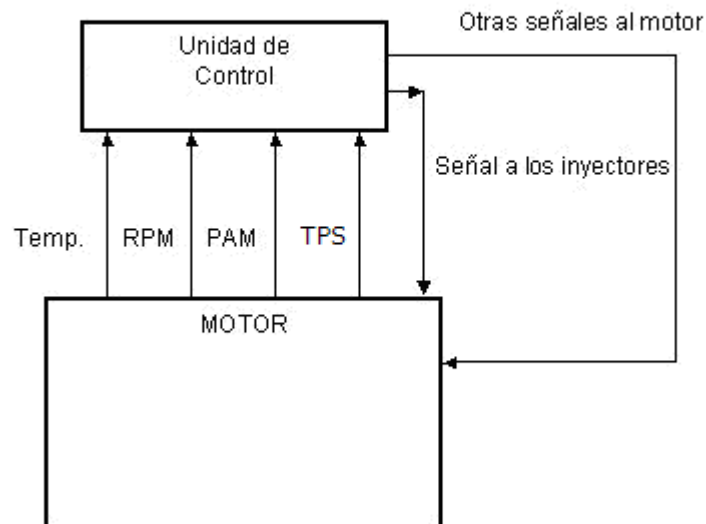


Figura 1.1 diagrama de la inyección electrónica

- ✓ Ventajas de la inyección electrónica de combustible
 - Un mayor control de la mezcla aire-combustible.
 - Bajos niveles de emisión de gases tóxicos.
 - Mejor rendimiento del motor.
 - Menor consumo de combustible.
 - Mayor regularidad del funcionamiento del motor.
 - Se mejora el arranque.
 - Se mejora la marcha en frío.
 - Se mejoran las transiciones.

1.2 ELEMENTOS DEL SISTEMA

1.3 DESCRIPCIÓN DE LOS ELEMENTOS DEL SISTEMA

1.3.1 ELEMENTOS MOTRICES Y/O MECÁNICOS

- **Inyectores**

Son electro válvulas solenoides que tienen características de apertura – cierre muy bien determinados que pueden trabajar a altas frecuencias y son las encargadas de entregar el gas al motor.

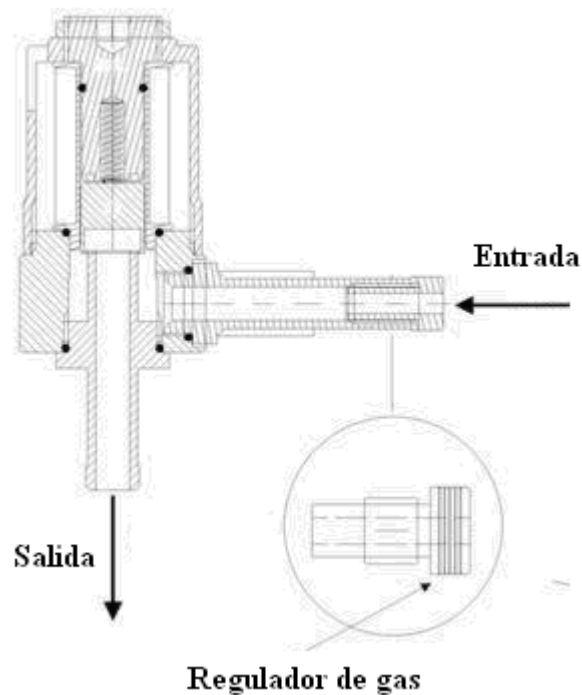


Figura 1.2 diagrama del inyector

- **Interruptor de selección de combustible**

Es un interruptor de tipo balancín con posición intermedia que se utiliza para hacer el cambio de combustible así si arrancamos a gasolina pasamos el interruptor a la mitad el cual cierra las dos válvulas hasta que se termine la gasolina del carburador para

evitar que se mezclen los dos combustibles y en el instante que se acabe dicha reserva conmutamos a gas y el motor funciona con este nuevo combustible.

1.3.2 ELEMENTOS DEL SISTEMA DE COMBUSTIBLE

- **Tanque de gas vehicular**

Este es un deposito cilíndrico con bases para montar en el vehiculo con capacidad de almacenamiento de 42 Kg. de gas construido con paredes de acero muy gruesas bajo normas de seguridad extrema en las cuales se incluyen colisiones a 84 Km\h exposición a elevadas temperaturas y presiones superiores a las establecidas por el fabricante.



Figura 1.3 Tanque de gas

- **Multiválvula**

Este dispositivo contiene tres servicios los cuales le dan su nombre: medidor de nivel, válvula de recarga y válvula de paso de gas al motor.



Figura 1.4 Multiválvula

- **Cañería de bronce**

Estas constituyen las líneas de transmisión de combustible; son de cobre puesto que soportan presión y tienen buenas características de resistencia para trabajar en la intemperie.

- **Vaporizador regulador de presión**

Es el encargado de transformar el gas licuado que recibe del depósito en gas y baja la presión para que pueda ser utilizado, tiene un sistema de calentamiento que ocupa la misma línea de refrigeración del motor la cual lo mantiene a la misma temperatura del mismo



Figura 1.5 Vaporizador regulador de presión

- **Electroválvulas**

Realizan la apertura y cierre de los dos tipos de combustible de manera inversa es decir la una se abre y la otra se cierra y viceversa para impedir que los combustibles se mezclen [2].



Figura 1.6 Electroválvulas

1.3.3 ELEMENTOS DE OBTENCIÓN DE SEÑAL

- **Sensor de temperatura del refrigerante**

Como el motor de combustión interna no se mantiene en el mismo valor de temperatura desde el inicio de funcionamiento, ya que se incrementa, las condiciones de funcionamiento también variarán notablemente, especialmente cuando la temperatura es muy baja, debiendo vencer las resistencias de sus partes móviles; adicionalmente un buen porcentaje del combustible inyectado es desperdiciado en las paredes del múltiple de admisión, de los cilindros y debido a la mala combustión, por lo que requerimos inyectar una cantidad adicional de combustible en frío y reducir paulatinamente este caudal hasta llegar al ideal en la temperatura óptima de funcionamiento.

Esta señal informa al computador la temperatura del refrigerante del motor, para que este pueda enriquecer automáticamente la mezcla aire – combustible cuando el motor está frío y la empobrezca paulatinamente en el incremento de la temperatura, hasta llegar a la temperatura ideal de trabajo, momento en el cual se mantiene la mezcla ideal.

Para ello se utiliza una resistencia NTC (NEGATIVE TEMPERATURE COEFICIENT), que como su nombre lo indica, es una resistencia de coeficiente negativo de temperatura. Esto quiere decir que la resistencia del sensor irá disminuyendo con el incremento de la temperatura medida, o lo que es lo mismo, que su conductibilidad irá aumentando con el incremento de temperatura, ya que cuando está frío el sensor, su conductibilidad es mala y aumenta con el incremento de temperatura.

El sensor está encapsulado en un cuerpo de bronce, para que pueda resistir los agentes químicos del refrigerante y tenga además una buena conductibilidad térmica. Está localizado generalmente cerca del termostato del motor, lugar que adquiere el valor máximo de temperatura de trabajo y entrega rápidamente los cambios que se producen en el refrigerante. En su parte anterior tiene un conector con dos pines eléctricos, aislados del cuerpo metálico.

Dependiendo del sistema, existen dos posibilidades de señal que puede entregar el sensor de temperatura:

- **Alimentación positiva:**

El sensor recibe en uno de sus pines una alimentación de 5 voltios de referencia, tensión eléctrica que la envía el computador una tensión ascendente de información hasta calentarse, momento en el cual le entrega una tensión mayor, pudiendo llegar cerca de los 5 voltios de alimentación. Esta señal se envía por el segundo pin del sensor hacia el computador, el cual identifica esta tensión variable en temperatura medida del refrigerante, entregando a los inyectores una cantidad de combustible ideal en cada etapa de calentamiento.

- **Alimentación negativa:**

Como en el primer caso, en otros sistemas se utiliza una alimentación negativa lo que significa que el primer pin del sensor tiene una conexión de tierra o MASA. Cuando el sensor está frío, la alta resistencia interior permite enviar una señal negativa muy pequeña por el segundo pin, dirigida al computador, pero seguirá incrementándose acorde al aumento de temperatura del motor.

Como se podrá notar, el tipo de señal que se envía al computador solamente dependerá del tipo de alimentación que se le entregue al sensor, el cual se encarga de enviar una señal variable de esta alimentación, progresiva con el aumento de temperatura.

Veamos en el esquema la constitución interna básica del sensor:

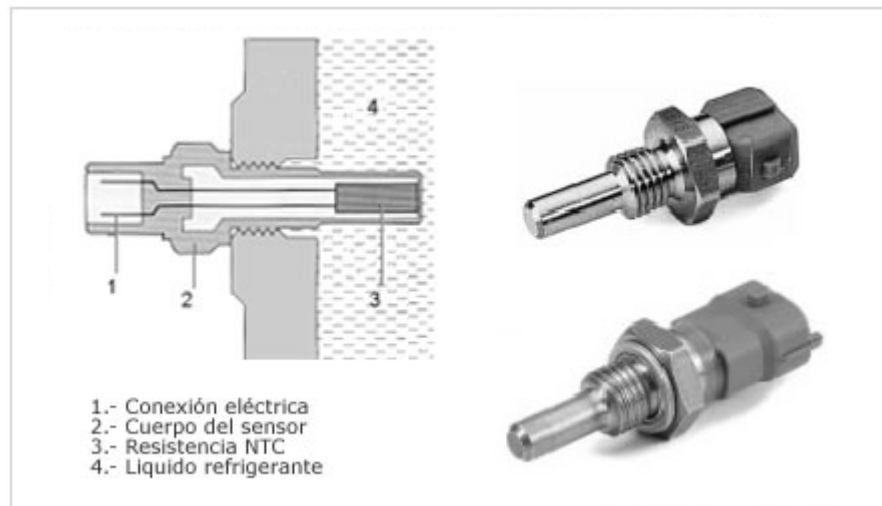


Figura 1.7 Esquema del sensor de temperatura del líquido refrigerante

Adicionalmente podemos decir que como el sensor se basa para su trabajo en la característica de su material, todos los sensores utilizados tendrán las características similares, con la diferencia mayor localizada en el tamaño, su diseño, la forma de la rosca y del conector, pero siempre tendrá características de medición muy similares, por no decir idénticas entre cualquier procedencia.

Es por ello que podemos asegurar, que una tabla de valores que relaciona la temperatura del sensor con la resistencia que nos entrega de nuestro ejemplo deberá coincidir en su mayor parte en todos los sistemas y marcas de Inyección electrónica y algunas Mecánicas con ayuda de la Electrónica. Esta señal sirve de información, como se dijo, para que el computador determine la mezcla exacta que debe inyectar, controlando a los inyectores del sistema; cuando se ha instalado un sensor en una inyección mecánica con ayuda eléctrica, también sirve para poder controlar con exactitud el caudal inyectado [3].

A continuación podemos observar la gráfica, donde se puede ver claramente que la resistencia del sensor disminuye con el incremento de la temperatura.

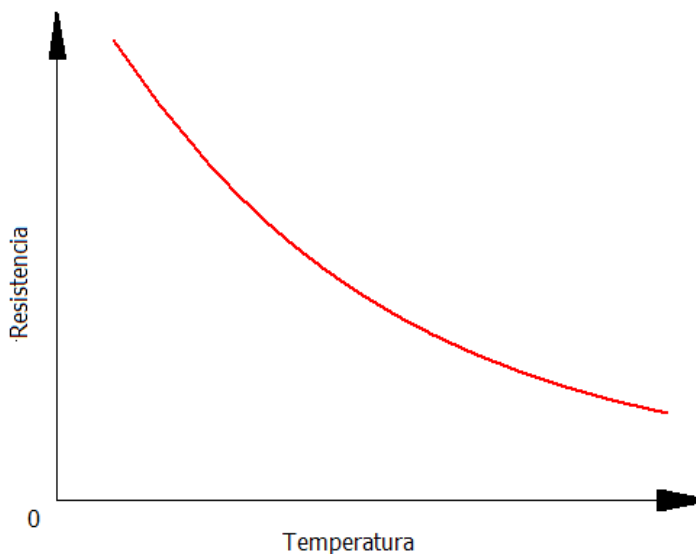


Figura 1.8 Grafica de los valores del sensor de temperatura

- **Señal de revoluciones del motor**

Uno de los datos más importantes que se requiere en un sistema de inyección, así como para el sistema de Encendido del motor de Combustión Interna, es justamente la señal del Número de Revoluciones a las cuales gira el motor.

Esta señal es tan importante debido a que el caudal de combustible que debe inyectarse está relacionado directamente con el número de combustiones que cada uno y el total de cilindros debe realizar.

Se entenderá que por cada combustión existen tres elementos relacionados para lograrlo, que son: una cantidad de aire aspirado, una cantidad de combustible relacionado exactamente (mezcla ideal) con este aire y un elemento capaz de inflamar la mezcla, que en este caso es la "chispa eléctrica" que logra combustionarla. Con la información del número de revoluciones, el Computador sabe el número de veces que debe inyectarse en combustible y la cantidad relacionada con el aire aspirado.

Existen algunas formas utilizadas para enviar una señal de revoluciones y en este momento las mencionamos.

- **Señal enviada por la Bobina de encendido.**

Al igual que la señal que requiere un tacómetro, instrumento electrónico que mide el número de revoluciones del motor, se puede enviar al Computador la misma señal, tomada de! mismo lugar que se ha tomado para este instrumento.

El lugar común del cual se ha tomado esta señal es el borne negativo de la bobina de encendido, es decir el contacto en el cual se interrumpe el bobinado primario de encendido, interrumpido por el "platino" o contacto del ruptor del sistema.

Como el platino debe interrumpir el campo magnético de la bobina un número de veces igual al número de cilindros que posee el motor, la señal resulta perfecta para información del número de revoluciones, ya que el tacómetro en el primer caso y el Computador en el siguiente, toman el número de pulsos recibidos y lo divide para el número de cilindros que posee el motor.

Con esta señal dividida, se sabe exactamente el número de vueltas o revoluciones a las que gira el motor, información que sirve en el caso de un Sistema de Inyección para determinar el caudal de Inyección por vuelta .[3]

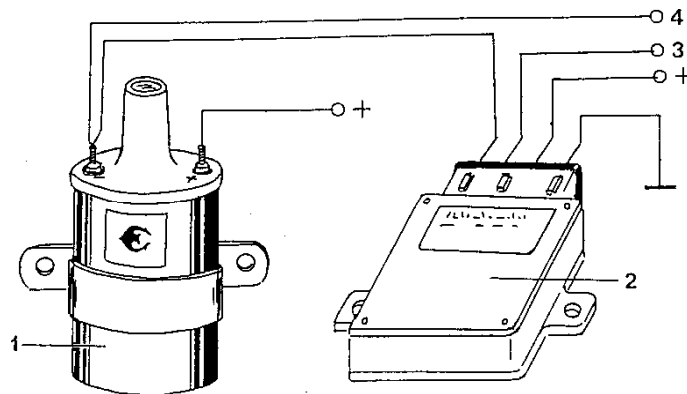
- **Señal enviada por el módulo de encendido**

Cuando el sistema de encendido tradicional por contactos (platinos) fue suplantado por un sistema de encendido electrónico, al no tener una señal pulsante de un contacto, se optó por tomar la señal del módulo de encendido, el cual cumple una función similar al de su antecesor, pero utilizando la electrónica.

Esta forma de pulsos lo crea el módulo, para formar el campo magnético primario de la bobina de encendido, para luego interrumpirla, logrando con ello realizar un pulso en el mismo borne, de forma idéntica al anterior: este pulso es enviado al Tacómetro en el caso de medición de revoluciones para el tablero de instrumentos y también al computador en el caso del Sistema de Inyección.

Como se notará, este pulso puede estar tomado tanto del módulo de encendido, como del lugar donde este actúa, que es el negativo de la bobina de encendido [3].

En el esquema que vemos a continuación se ve la forma de conexión de esta señal de revoluciones del motor.



1. Bobina de encendido
2. Módulo de encendido
3. Señal de RPM
4. Señal al Tacómetro

Figura 1.9 Grafica de modulo de encendido electronico

- **Señal enviada por un sensor inductivo en el volante del cigüeñal**

Como las señales de revoluciones del motor pueden adquirir errores, debido especialmente a la forma de actuar el módulo electrónico del caso anterior, quien recibe a su vez la señal de un generador de pulsos inductivo, de un generador Efecto HALL o de un sensor fotoeléctrico, en los sistemas precedentes se ha optado por tomar esta señal de un lugar más exacto.

Podemos entender este inconveniente, debido a que una parte mecánica es la encargada de mantener o alojar al sensor antes mencionado y como las partes mecánicas están expuestas a desgastes, falta de mantenimiento u otro tipo de daños, la señal que se envía al módulo de encendido sería también errada o por mejor decirlo, no muy exacta.

Por estas razones se ha diseñado un sistema completamente electrónico, el cual se basa en la información generada por un sensor inductivo, el mismo que genera señales de corriente alterna, tantas veces como número de dientes (de la rueda fónica o piñón) pasen junto a él.

Esta información es "traducida" por el computador, quien se encarga de adelantar o retardar el punto de encendido electrónicamente y envía una señal de comando al amplificador, que en este caso es el módulo de encendido. El módulo, a su vez, controla la formación y la interrupción del campo magnético de la bobina de encendido, sirviendo adicionalmente en los sistemas de Inyección como contador de revoluciones y con ello se modificará el caudal de entrega en los inyectores [3].

En la figura se puede ver la forma del sensor inductivo y de la rueda fónica o polea dentada.

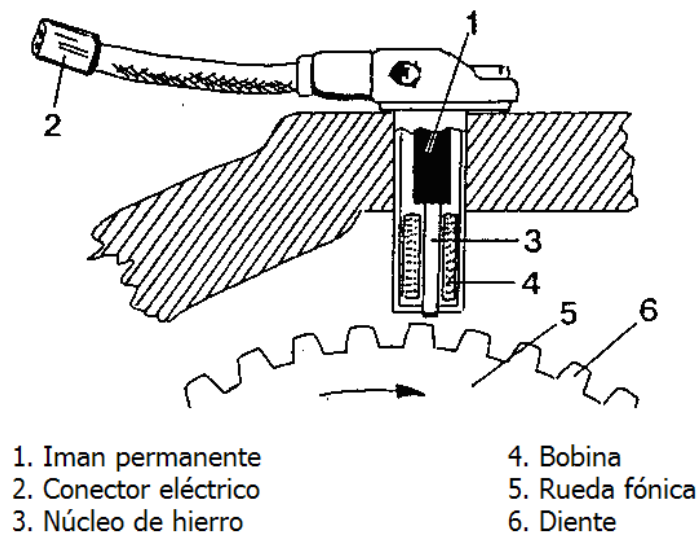


Figura 1.10 Grafica de sensor inductivo y rueda fónica

- **Señal de revoluciones enviada por el pick-up del distribuidor**

Tal como en el caso anterior, la señal de revoluciones puede ser tomada de un sensor inductivo en el distribuidor de encendido; a este sensor se lo denomina "pick-up" y se basa en los principios similares al sensor anterior, con la diferencia de que el eje del distribuidor tiene una rueda dentada, la cual corta el campo magnético del sensor, enviando tañías señales como número de dientes que posea esta rueda.

El computador identifica como una vuelta o revolución del motor al número de pulsos alternos generados en media vuelta del distribuidor, ya que este gira a mitad de vueltas del cigüeñal.

La razón básica de utilizar este tipo de señal se basa en evitar en determinado motor un nuevo diseño para el sensor, utilizando su antigua estructura, modificando únicamente al diseño del distribuidor tradicional, como lo podemos apreciar en el esquema [3].

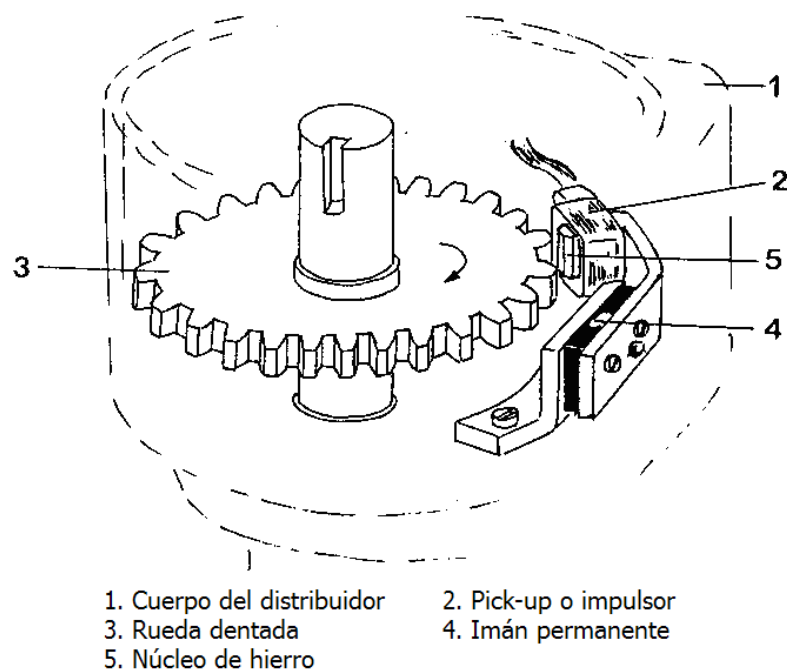


Figura 1.11 Grafica de sensor en el distribuidor

- **Señal fotoeléctrica proveniente del distribuidor**

Así como en las señales inductivas provenientes de un sensor alojado en el distribuidor anteriormente mencionado, algunos fabricantes utilizan otros tipos de sensores, como por ejemplo sensores a base de fotodiodos o sensores sensibles a la luz.

Se basan en la emisión de una luz infrarroja, captándola con un fotodiodo. Para ello, está alojado en el eje del distribuidor una lámina ranurada o perforada; el sensor está localizado opuesto al emisor de luz y el disco o lámina ranurada y al girar permite

cada vez que exista una ranura o una perforación, se obture o se abra la emisión de la luz con respecto al lector o sensor fotoeléctrico.

Si este disco posee por ejemplo 50 ranuras en su periferia, el sensor detectará 50 señales por cada vuelta del disco, enviando esta señal al computador, quien determina con ello el número de revoluciones de giro del motor [3].

En el esquema se puede observar la característica del sensor.

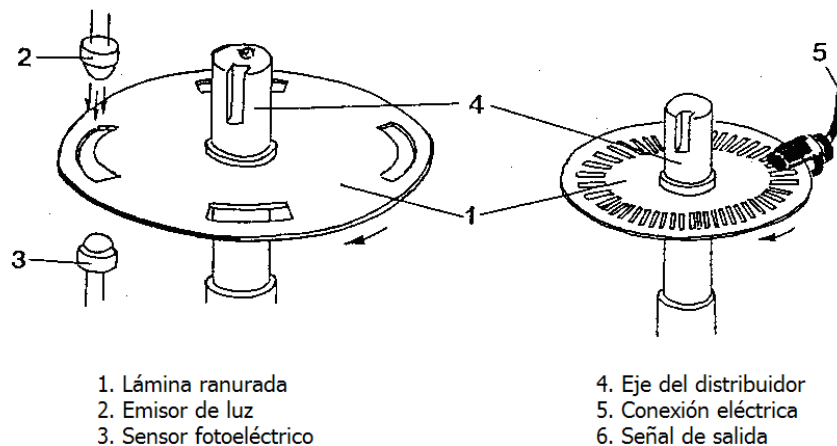


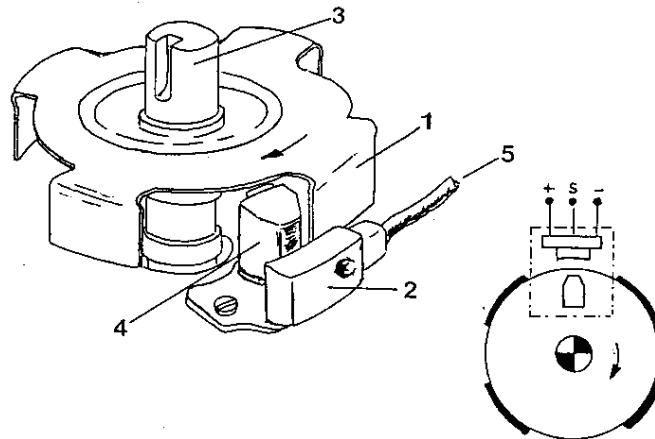
Figura 1.12 Gráfica de sensor óptico en el distribuidor

- **Sensor efecto hall localizado en el distribuidor**

Este sensor tiene antecedentes de su utilización en los sistemas de encendido electrónico, pero algunos fabricantes lo utilizan también como información adicional del número de revoluciones del motor, basándose en el número de pulsos o señales que este sensor pueda entregar.

El sistema Hall se basa en el principio de conductibilidad de una pastilla semiconductor, cuando se enfrenta a ella un campo magnético, es decir, si las líneas magnéticas de un Imán permanente está cercano o enfrenteado a esta "pastilla", ella se convierte en conductora eléctrica, emitiendo una señal hacia el computador. En cambio cuando una pantalla (disco) interrumpe o tapa esta acción del campo magnético del imán, la "pastilla" deja de conducir o enviar esta señal.

El número de veces que se envíe esta señal, dependerá únicamente del número de ventanas que posea el disco o pantalla obturadora en su periferia, la misma que también está alojada en el eje del distribuidor, forma que la podemos observar en la figura [3]



- | | |
|-------------------------|--------------------|
| 1. Pantalla obturadora | 2. Pastilla HALL |
| 3. Eje del distribuidor | 4. Imán permanente |
| 5. Conector eléctrico | |

Figura 1.13 Gráfica de sensor hall en el distribuidor

- **Señal de la posición de la mariposa de aceleración**

Esta señal se la obtiene de un potenciómetro eléctrico, el cual está alojado en el cuerpo de la mariposa de aceleración y el cual recibe e) movimiento de la aleta a través del mismo eje, de tal manera que la resistencia variable del potenciómetro esté relacionada de acuerdo a la posición en la cual se encuentre la mariposa.

A este potenciómetro se lo alimenta con una tensión de referencia, la cual generalmente es de 5 Voltios, provenientes de un regulador de voltaje del mismo computador. Cuando la mariposa de aceleración se encuentra en su posición de reposo, la cantidad de tensión que se envía como señal será de unas cuantas décimas de voltio y esta señal se irá incrementando paulatinamente, de acuerdo al incremento en el movimiento de la mariposa, hasta llegar al tope de la escala, la cual nos dará un valor cercano a los 5 Voltios de la referencia.

Esta señal variable de tensión eléctrica se dirige de regreso al computador, el cual identifica esta tensión como una posición exacta de la mariposa de aceleración.

Adicionalmente algunos sistemas tienen un interruptor que conecta dos contactos en la posición de reposo de la mariposa, los cuales pueden estar alojados en el mismo cuerpo del potenciómetro; este interruptor sirve como una señal adicional para que el computador determine la desaceleración del motor, lo cual sirve para que se corte el envío de combustible hacia los inyectores en esta etapa, reduciendo enormemente el consumo de combustible del motor [1] [3].

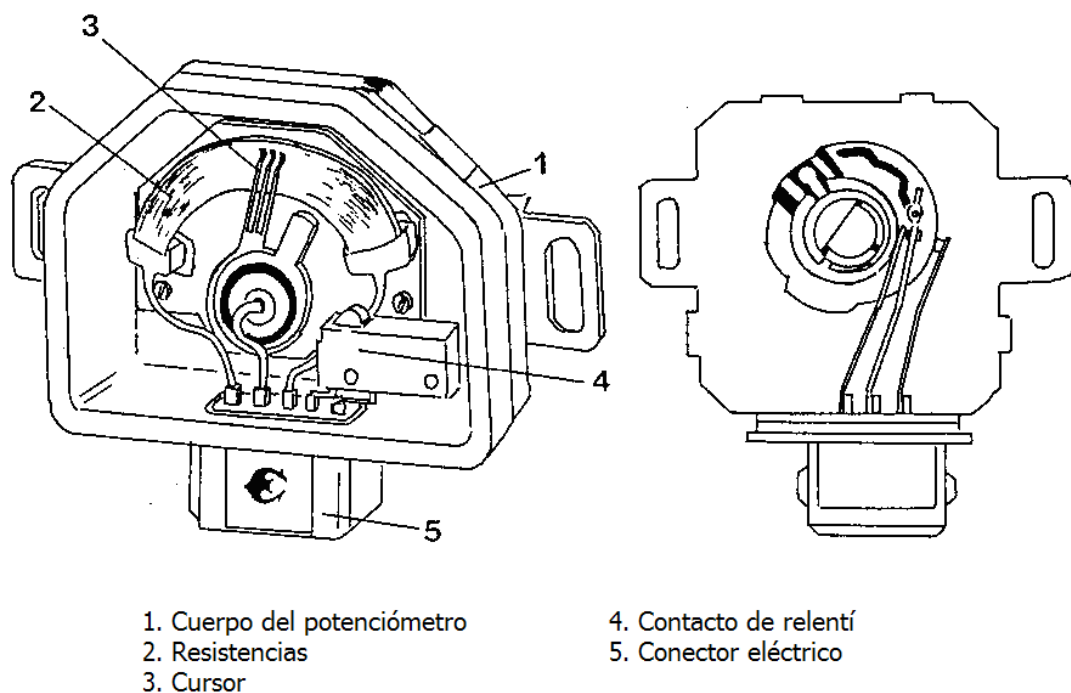


Figura 1.14 Gráfica del sensor de posición

- **Sensor de presión absoluta**

Este sensor toma las medidas de presión mediante un captador sensible al vacío producido en el múltiple de admisión.

El captador está constituido por un diafragma realizado en materia aislante dentro del cual están emplazadas unas resistencias que forman un puente de medida.

El puente de resistencias este formado por sensores piezoeléctricos que son sensibles a las deformaciones mecánicas.

El diafragma esta unido mediante un tubo al colector de admisión de manera que las variaciones de presión actúan directamente sobre el diafragma provocando su deformación. Esta deformación actúa sobre el puente de resistencias variando la tensión de salida.

La tensión de salida del puente es ajustada a las escalas de trabajo deseadas de manera que se obtiene una tensión final de salida comprendida entre 0 y 5 V. siguiendo de manera lineal las variaciones de presión [3].

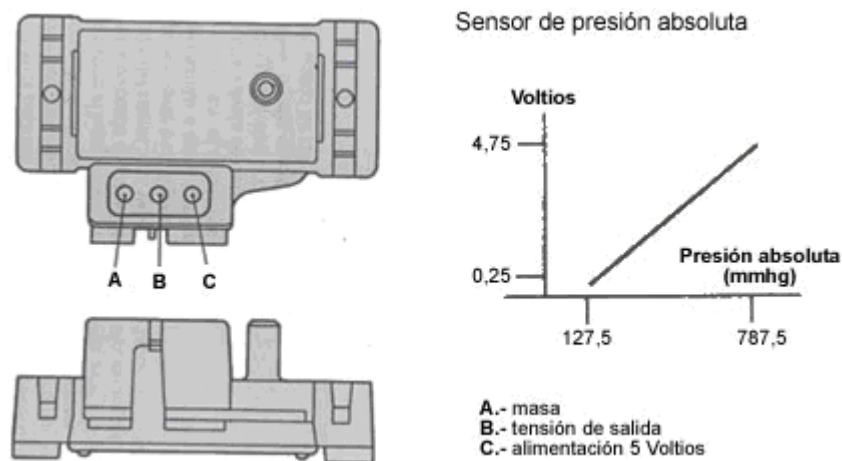


Figura 1.15 Gráfica del sensor de presión absoluta

1.3.4 ELEMENTOS DE INTERFASE

- **Detector de cruce por cero**

Este circuito es la parte fundamental del diseño de la tarjeta de control puesto que de aquí tenemos exactamente el sincronismo de la parte mecánica con la electrónica; se toma la señal del sensor inductivo que esta dentro del distribuidor la cual se asemeja a una senoidal y esta ingresa al siguiente detector de cruce por cero

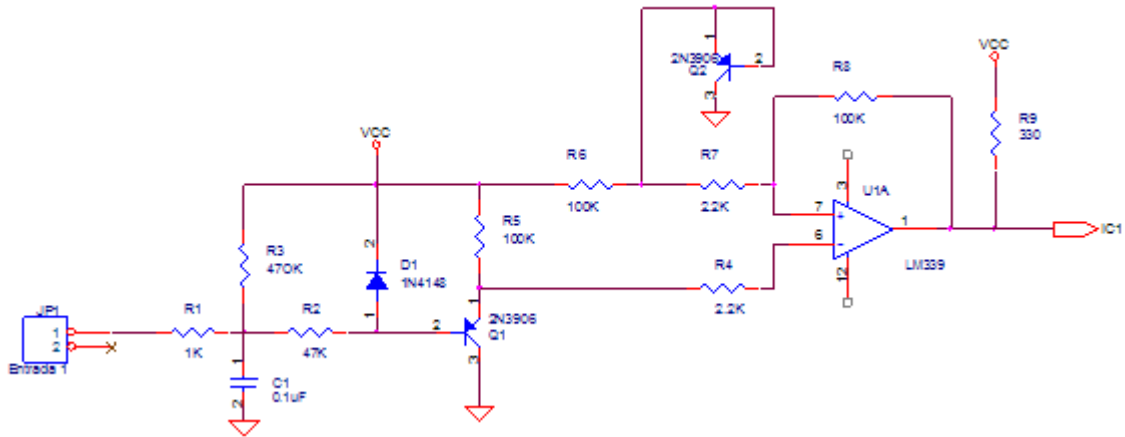


Figura 1.16 Diagrama del detector de cruce por cero

Luego obtenemos una señal cuadrada perfecta que sigue a la frecuencia de funcionamiento del motor como se muestra a continuación:

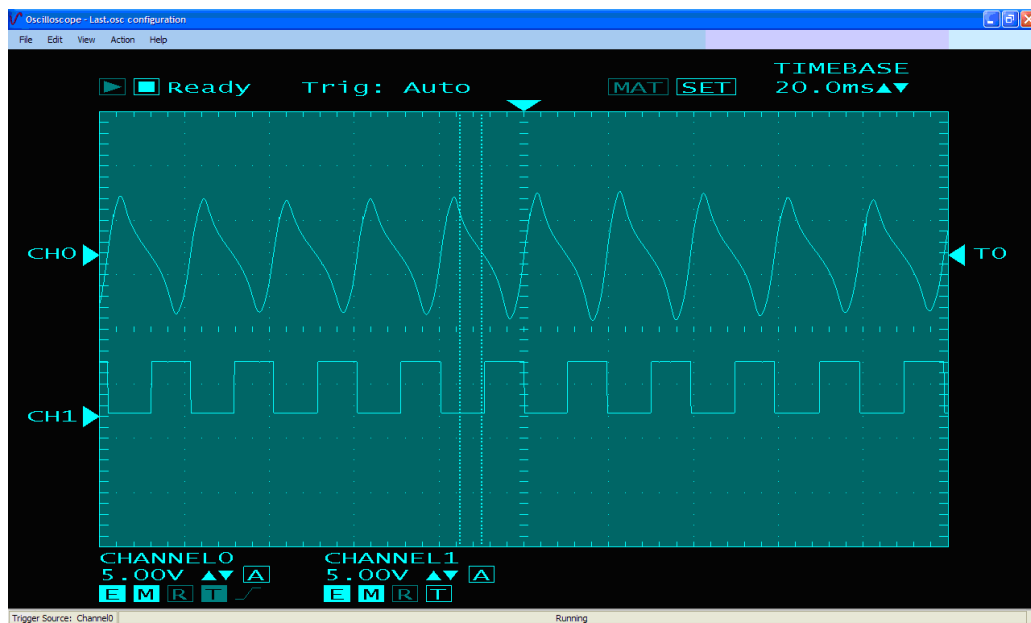


Figura 1.17 Pantalla del detector de cruce por cero

1.3.5 ELEMENTOS DE CONTROL

- Amplificador de corriente

Para poder excitar de forma correcta los inyectores utilizamos unos amplificadores de corriente en configuración Darlington por las prestaciones de alta corriente que estos ofrecen con la cual cumplimos con el requerimiento especificado en la placa de los inyectores [1].

El amplificador Darlington (frecuentemente llamado amplificador compuesto) es una conexión muy popular de dos transistores de unión bipolar para funcionar como un solo transistor “superbeta”, la conexión Darlington se muestra en la figura 1.13. La principal característica de la conexión Darlington es que el transistor compuesto actúa como una sola unidad con una ganancia de corriente que es el producto de las ganancias de corriente de dos transistores por separado.

Si se hace coincidir a los dos transistores de tal modo que $b_1 = b_2 = b$. La conexión Darlington suministra una ganancia de corriente de:

$$\beta_D = \beta'$$

Una conexión de transistor Darlington proporciona un transistor con una ganancia de corriente de varios miles.

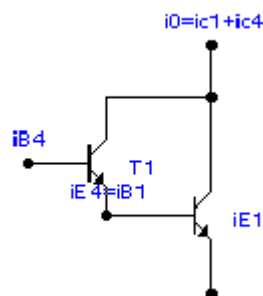


Figura 1.18 Esquema de configuración darlington

Un circuito Darlington básico se ilustra en la figura 1.14 Se hace uso de un transistor Darlington con muy alta ganancia de corriente, D. La corriente de base puede calcularse a partir de:

$$I_B = \frac{V_{CC} - 2V_{BE}}{R_B + \beta_D R_E}$$

Mientras que la ecuación anterior es la misma para un transistor regular, el valor de β_D es mucho mayor, y el valor de V_{BE} es más grande. La corriente de emisor es por lo tanto:

$$I_E = (\beta_D + 1)I_B = \beta_D I_B$$

Los voltajes de cd son:

$$V_E = I_E R_E$$

$$V_B = V_E + 2V_{BE}$$

La ganancia de corriente del amplificador Darlington, mostrado en la figura 1.14 puede ser calculada a partir de:

$$A_i = \frac{i_e}{i_{B2}} = \frac{i_{C1} + i_{C2}}{i_{B2}} = \frac{\alpha_1 i_{E1} + \alpha_2 i_{E2}}{i_{B2} (\beta_{D2})}$$

Donde:

$$\begin{aligned} \frac{i_{E1}}{i_{B2}} &= \frac{i_{E1}}{i_{B1}} \frac{i_{B1}}{i_{B2}} \frac{i_{E2}}{i_{B2}} \\ &= (h_{fe_1} + 1)(1)(h_{fe_2} + 1) = (h_{fe_1} + 1)(h_{fe_2} + 1) \end{aligned}$$

Así:

$$A_i = \alpha_1 (h_{fe_1} + 1)(h_{fe_2} + 1) + \alpha_2 (h_{fe_2} + 1)$$

Si los dos transistores son idénticos

$$A_i \approx h_{fe}^2$$

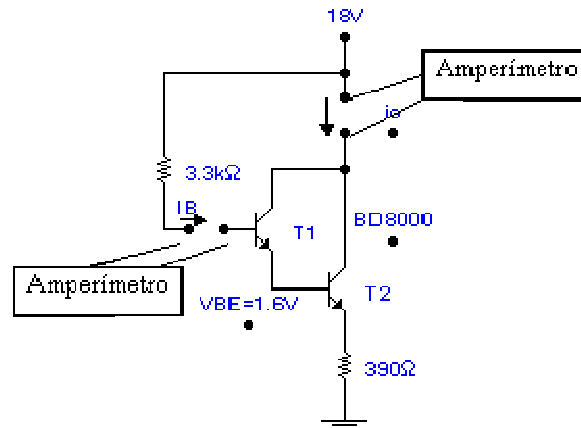


Figura 1.19 Amplificador darlington básico

Haciendo uso de estas consideraciones de diseño usamos el transistor TIP 110 que es un transistor Darlington encapsulado con otro transistor conectado en cascada también en la misma configuración con diodos de protección de corrientes inversas así:

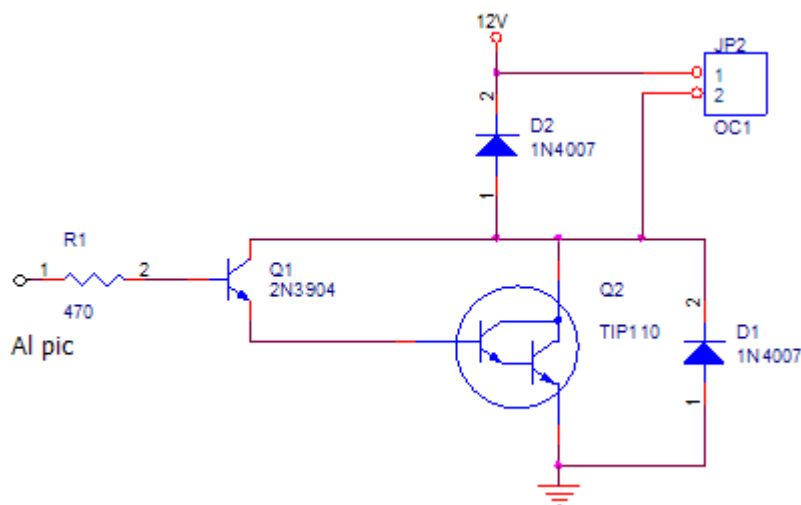


Figura 1.20 Amplificador de corriente

1.4 CONDICIONES ESPECIALES DEL SISTEMA

Un aspecto importante que se debe tomar en cuenta es que el sistema requiere de una condición especial para realizar el cambio de combustible, la cual consiste en dejar que se consuma toda la gasolina que queda en la cámara del carburador, de esta manera se evita que se mezclen los dos combustibles y produzca una contraexplosión.

Por otra parte es importante que el arranque se lo realice con el combustible original en este caso gasolina, por motivo de que la regulación de gas se realiza por medio de un medidor de presión (MAP), y en el arranque este presenta datos no válidos.

Para este caso utilizamos la misma señal que envía el sensor inductivo, que se coloco en el distribuidor al momento de cambiar el sistema de platino por un sistema de encendido electrónico, lo cual nos oblijo a que se realice este cambio [2].

CAPITULO II

SISTEMA CONTROLADOR

2.1 INTRODUCCION

En el siguiente capítulo se expone las especificaciones definitivas y detalladas que tendrá el hardware del sistema y el diseño que le corresponde. Para el sistema de inyección electrónica a gas debe cumplir con ciertos requerimientos de hardware para trabajar como tal. Desde el punto de vista de características funcionales, el sistema tendrá un hardware muy similar a un controlador embebido con adquisición y procesamiento de datos digitales ya que debe cumplir con lo requerimientos generales (Re1-Re6) y aquellos relacionados a la tecnología disponible en nuestro país y que definitivamente permiten la realización del proyecto (Re7-Re10).

REQUERIMIENTOS GENERALES	
Re1	Ser autosuficiente al cumplir la tarea que se le encomienda, esto es, el sistema debe ser capaz de controlar los sensores que se vayan a instalar de tal manera que la dosificación de combustible sea exacta para que la mezcla aire gas sea eficiente y nos permita un funcionamiento eficaz.
Re2	La ejecución de la aplicación debe realizarse en tiempo real, lo que significa que el sistema no debe presentar ningún tipo de retardo al variar los parámetros de los sensores
Re3	Debe incluirse al menos un canal de comunicación por medio del cual se pueda configurar el equipo o realizar otras funcionalidades.

Re4	Es necesario contar con entrada y salida analógica que puedan manejar rangos de señal típicos en la industria (en amplitud y frecuencia).
Re5	Debe tener entradas y salidas digitales capaces de soportar voltajes entre 5 y 12V.
Re6	El circuito debe acondicionar el voltaje de alimentación para que todos los componentes se energicen correctamente.
REQUERIMIENTOS TECNOLÓGICOS	
Re7	Se procurará utilizar la tecnología y componentes más modernos en la medida de las posibilidades.
Re8	La tecnología de placa debe ser máximo de doble lado y que todos los componentes sean “through hole” para permitir un ensamblaje fácil y la reparación de la placa a lo largo de su vida útil.
Re9	En lo posible se intentará realizar el proyecto con elementos disponibles en el mercado ecuatoriano, caso contrario se importarán las partes.
Re10	Los valores de los componentes pasivos deben ser nominales.

Tabla. 2.1. Requerimientos del Hardware

De los seis primeros puntos expuestos que definen el hardware se obtienen los 5 bloques básicos que componen el sistema de inyección electrónica a gas:

Requerimientos	Bloque funcional encargado		Referencia Planos
1 y 2	Procesador/Controlador e Interfase	B1	A2.1
3	Comunicaciones	B2	A2.1
4	I/O Analógicas	B3	A2.1 y A2.2

5	I/O Digitales	B4	A2.1
6	Fuente de poder	B5	A2.2

Tabla. 2.2. Bloques funcionales según requerimientos

La Figura 2.1 es el diagrama general del hardware del sistema, la explicación y justificación de cada bloque se detalla conforme se avanza en el capítulo. En la Tabla 2.2 se señala el lugar en los planos o esquemáticos (Anexo A2) del sistema de inyección electrónica a gas donde se encuentra cada bloque funcional [21].

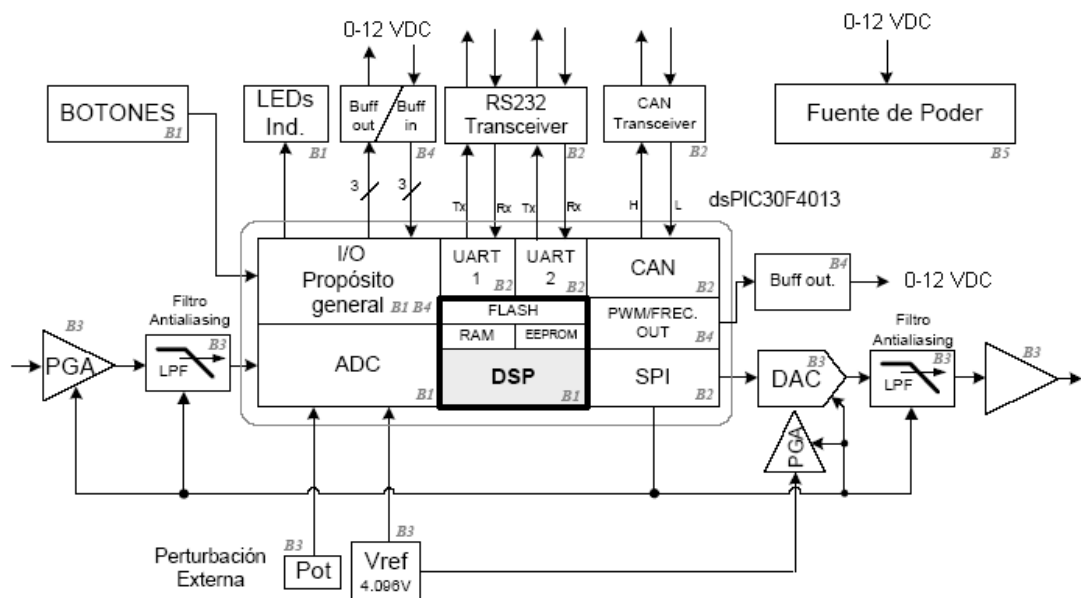


Figura. 2.1. Diagrama general del Hardware del sistema de inyección electrónica a gas

2.2 Descripción del controlador digital de señales (DSC)

✓ Aspectos generales del (DSC)

El dsPIC30F es un procesador RISC de 16bits de alto desempeño que ofrece la funcionalidad de un DSP con las capacidades fundamentales de control de un microcontrolador. Posee una arquitectura Harvard modificada lo que permite tener un rendimiento de hasta 30 millones de operaciones por segundo (MIPS). Integra un buen grupo de periféricos y elementos que permiten una fácil implementación de aplicaciones

como control de motores, reconocimiento de voz, medidores de energía, controladores avanzados entre otros.

En el anexo A3.1 [21] se incluye el resumen del fabricante en cuanto a las características del DSC y en el Manual de Referencia [22] una descripción completa de funcionalidades. A continuación se da un resumen de las características del chip dsPIC30F4013 que cumplen con los requerimientos planteados en el punto 2.1.

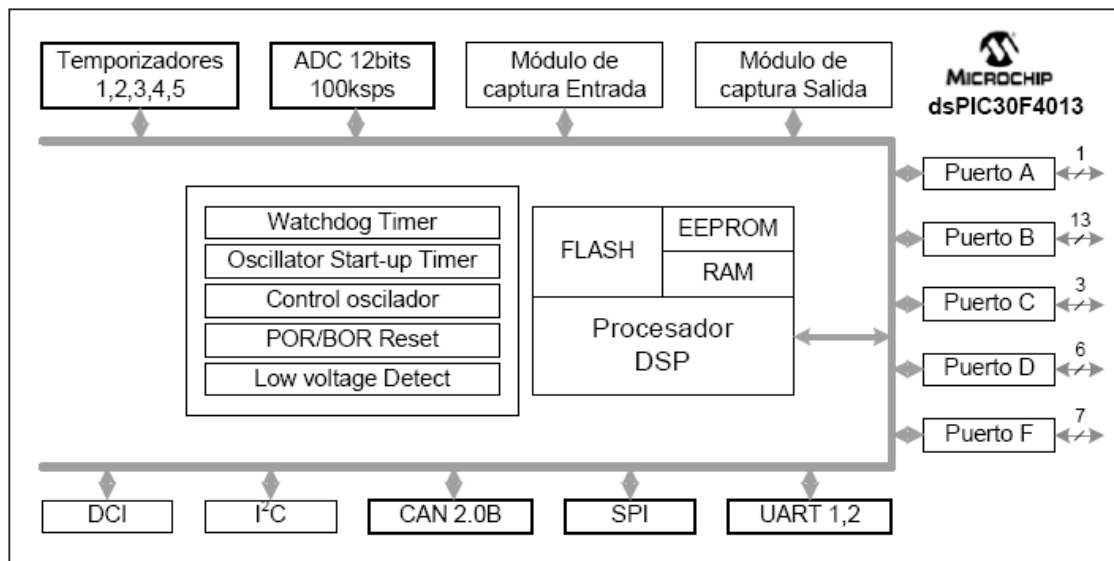


Figura. 2.2. Diagrama general del DSC dsPIC30F4013

2.3. Capacitores de desacoplo

Estos capacitores van ubicados lo más cerca posible de los pines de alimentación de cada integrado (IC). Su función es importante en la supresión de ruido y el correcto funcionamiento de los componentes. Tienen la función de filtrar el ruido que hay en el sistema hacia el componente o evitar que el ruido que genera el componente se disperse por el sistema y afecte otros componentes. Para el caso del sistema, todos los integrados tienen un capacitor de 0.1uF en sus pines de alimentación y para los AMP-OPs se incluye un capacitor de 10uF en paralelo a los de 0.1uF.

2.4 COMUNICACIÓN CON SENSORES

- ✓ Tipos de comunicaciones en el sistema de inyección electrónica a gas

El sistema utiliza dos de los periféricos dedicados a comunicación que tiene el DSC y cumplen las siguientes funciones (Tabla 2.3), cabe señalar que para futuros proyectos el CAN es un protocolo que cada vez tiene mayor aceptación en vehículos y maquinarias en general.

Módulo de comunicación	Utilización en el sistema de inyección electrónica a gas
UART1 Universal Asynchronous Receiver Transmitter	Configurado para trabajar en los pines alternos es el canal de comunicación por el cual se descarga el programa con ayuda del bootloader y se realiza la configuración del sistema de inyección electrónica a gas y además es el que, con un cable cruzado convencional, se comunica con cualquier equipo CTU (Control Terminal Unit) con interfase RS232, por ejemplo una PC.
UART2	Se da la posibilidad de que con un cable especial se tenga un segundo canal asíncrono RS232. Con un cable cruzado se tienen las señales de control CTS y RTS.
CAN2.0B Controller Area Network	Aunque no se utilizará en el sistema de inyección electrónica a gas, es conveniente dejar la extensión de proyección para futuras aplicaciones. El CAN es un protocolo que cada vez tiene mayor aceptación en la industria.

Tabla. 2.3. Designación de los canales de comunicación en el sistema.

✓ Interfase para UART1 y UART2}

Para poder lograr los niveles de voltaje requeridos por la interfase RS232 es necesario incorporar al sistema de inyección electrónica a gas un transceiver para acoplar correctamente las señales. Otra función importante que debe cumplir este elemento es la protección del circuito de sobrevoltajes y cortocircuitos.

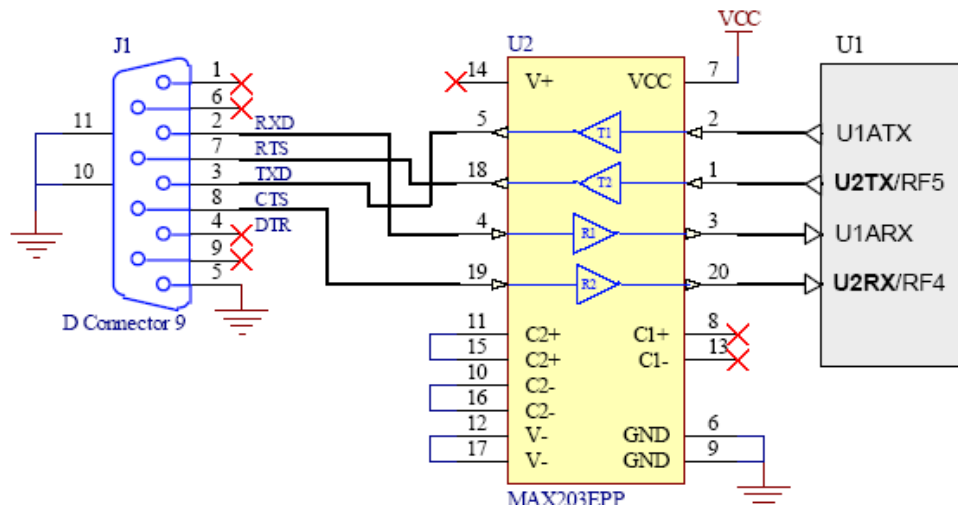


Figura. 2.3. (A2.1)

Para esta aplicación se seleccionó el transceiver MAX232. Contiene 2 drivers de salida y 2 drivers de entrada que acoplan señales TTL (0-5V) con los voltajes de la interfase RS232 ($\pm 12V$).

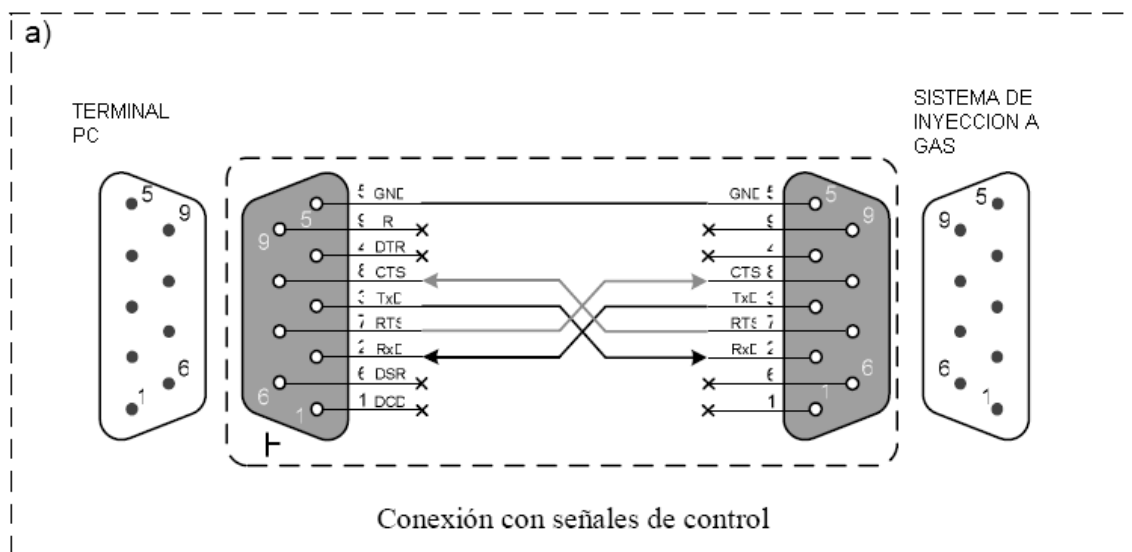


Figura. 2.4. Conexiones posibles por medio del puerto serial.

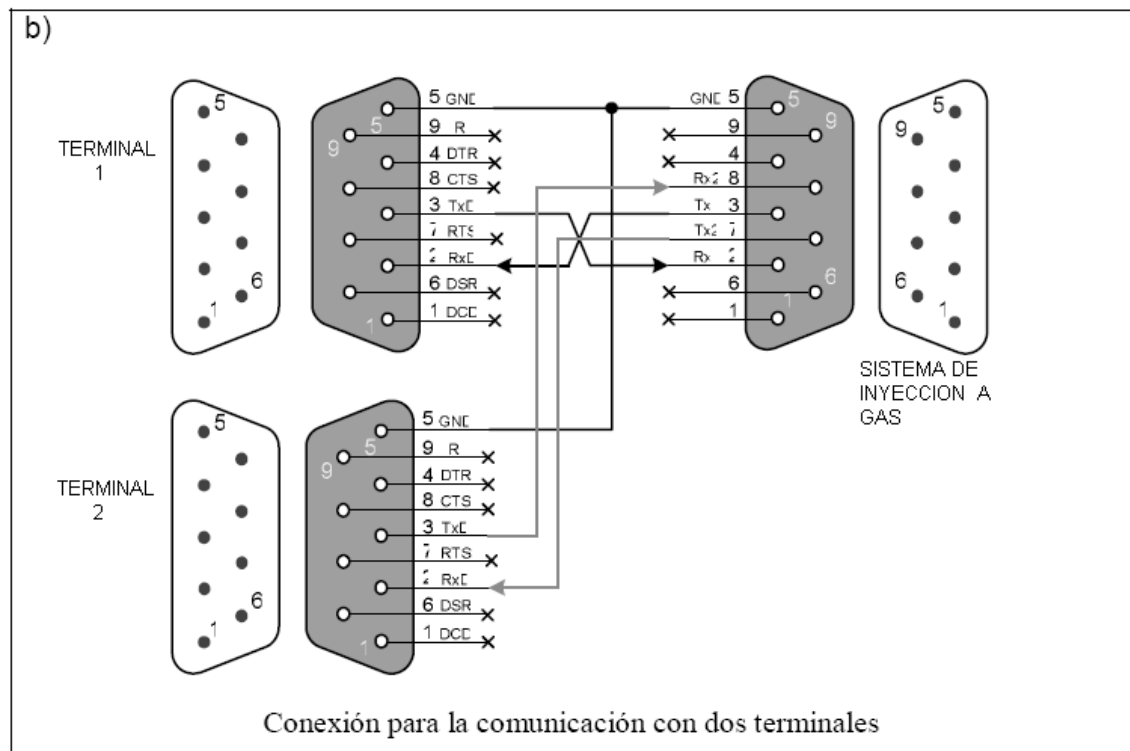


Figura. 2.5. Conexiones posibles por medio del puerto serial.

El conector utilizado es un DB9 macho. Las posibles conexiones se muestran en la Figura 2.4.y 2.5. La compatibilidad necesaria para un adaptador USB-Serial se garantiza con los niveles de voltaje presentado por el MAX232. Este garantiza una interfase EIA/TIA-232E y en los adaptadores USB-Serial por lo general cumplen con esta especificación.

2.5. COMUNICACIÓN CON ACTUADORES

- **Entradas y salidas analógicas**

- ✓ Aspectos generales del circuito analógico

El diseño del circuito analógico juega el papel más importante en la calidad del sistema de adquisición de datos. Para el sistema de inyección electrónica a gas se tendrá entradas analógicas y discretas, con rangos de señal unipolar de 0 - 12 V. y 0 – 5 V. Si

bien se encuentra en la industria sistemas bipolares, se sacrifica esta característica en el sistema de inyección electrónica a gas en aras de lograr un circuito más sencillo

CAPITULO III

DISEÑO DEL SISTEMA DE INYECCION DE COMBUSTIBLE

3.1.- DISEÑO INTEGRAL DEL SISTEMA

- **Controlador e interfase**
 - ✓ Selección de componentes

La selección del procesador/controlador es clave para un buen desarrollo del proyecto y, como se indicó en la Tabla 2.1, Re1 y Re2, se deben cumplir ciertas exigencias del mismo. Ya que para el desarrollo de este proyecto se necesita que el procesador tenga la mayor cantidad de funciones incorporadas en un solo chip, esto es, que al menos se encuentre incorporado al procesador, la memoria RAM y Flash, comunicaciones (Re3) y salidas/entradas digitales (Re5), con el fin de poder hacer modificaciones para diferentes fabricantes de motores en los cuales se aplicara el sistema.

Se buscó entre distintos fabricantes de procesadores para aplicaciones embebidas que demandan cierto nivel de complejidad matemática, como Texas Instruments, Freescale y Microchip. Todos estos fabricantes tienen una línea de productos denominados controladores digitales de señales (DSC), los cuales incorporan la robustez e integración de un microcontrolador con la capacidad computacional de un DSP. Para la selección se tomaron en cuenta no solo aspectos de potencia computacional sino de facilidad en conseguir los elementos, forma de programarlos, disponibilidad de herramientas de desarrollo y empaques DIP. La selección final fue por el DSC de Microchip de la familia dsPIC30F y específicamente el dispositivo dsPIC30F4013.

3.2 ELECCION DE ELEMENTOS SENSORES

Según lo estudiado en el capítulo 1 decidimos utilizar, solo los sensores necesarios para poder obtener la información que requiere el controlador para poder dosificar el gas de manera precisa. Los cuales son:

- **MAP:** Nos indica la carga del motor.
- **TPS:** Indica el instante en que hay variación instantánea de aceleración.
- **SENSOR DE TEMPERATURA:** Indica cuando el motor esta frío para que el controlador enriquezca la mezcla y active la válvula de vacío para elevar las RPM mientras se calienta.
- **SENSOR INDUCTIVO :** Nos proporciona la señal de sincronismo

3.3 DISPOSICIÓN DE LA VÁLVULA DE SALIDA DE COMBUSTIBLE DEL TANQUE DE ALMACENAMIENTO.

Esta válvula esta ubicada necesariamente en el tanque de almacenamiento de combustible puesto que el medidor de nivel del mismo esta incorporado en esta porque es de sistema mecánico de flotador por esta razón se la llama multiválvula, pero sin dejar de lado las normas de manipulación del gas como combustible vehicular [300].

3.4 DISPOSICION DE VALVULA DE INGRESO DE COMBUSTIBLE A LA ADMISION DEL MOTOR

Estas válvulas dosificadoras conocidas como inyectores se ubican en la cabeza del carburador puesto que se trata de un diseño monopunto como se muestra en la figura 3.1

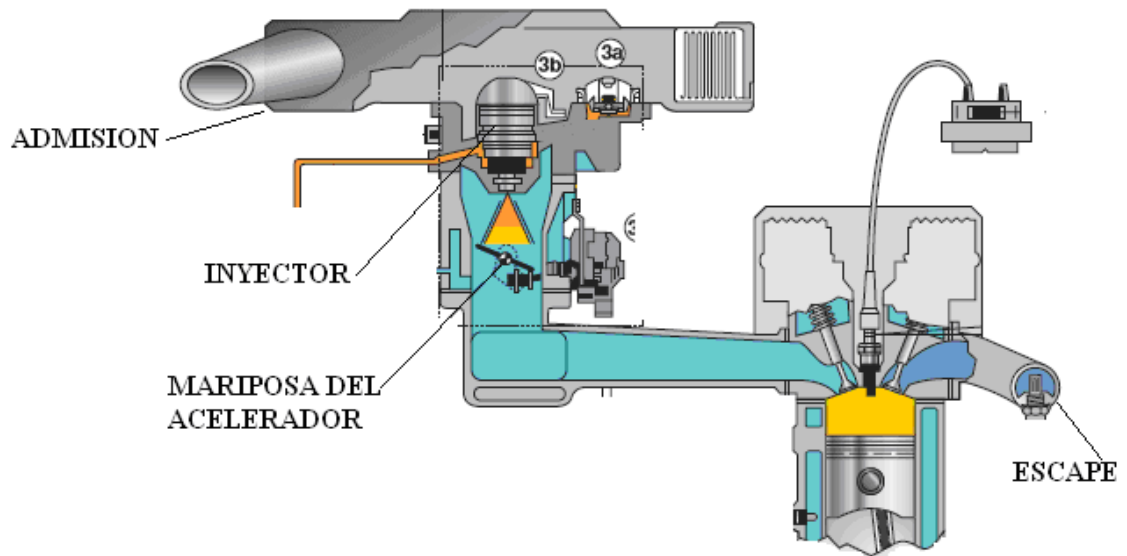


Figura 3.1 Ubicación de inyectores

3.5 DISEÑO DEL CONTROLADOR

3.5.1. Consideraciones y dimensionamiento

- **Elaboración de la placa pcb**

- ✓ **Parámetros de diseño**

Por facilidad en la elaboración de la placa se estableció de antemano que todos los elementos utilizados serían “through hole” y la placa utilizada de un lado con recubrimiento de cobre de 1 oz y sustrato tipo FR4. El plano de la placa se encuentra en el anexo A4. Algunos de los lineamientos que se siguieron al hacer la placa PCB fueron [31], [32].

1. Se trata de alejar lo más posible los elementos de comunicación y entradas/salidas digitales del circuito analógico con el fin de evitar interferencias y ruidos.

2. Los capacitores de desacoplo deben ir ubicados lo más cerca posible a los pines de alimentación de los componentes.
3. El tamaño de la placa debe ser lo más reducida posible con el fin de abaratar costos.
4. Se busca que el largo de los trazos sea lo más pequeño posible.
5. En lo posible se trazan las líneas Tx y Rx paralelas para las comunicaciones USART y CAN en caso de usarlas.
6. Los reguladores lineales no van a estar sometidos a grandes demandas de corriente, se calcula que la carga total está entre los 200 y 300 mA, por lo que se destina cierta área de cobre como disipador para ambos reguladores. Figura 3.2

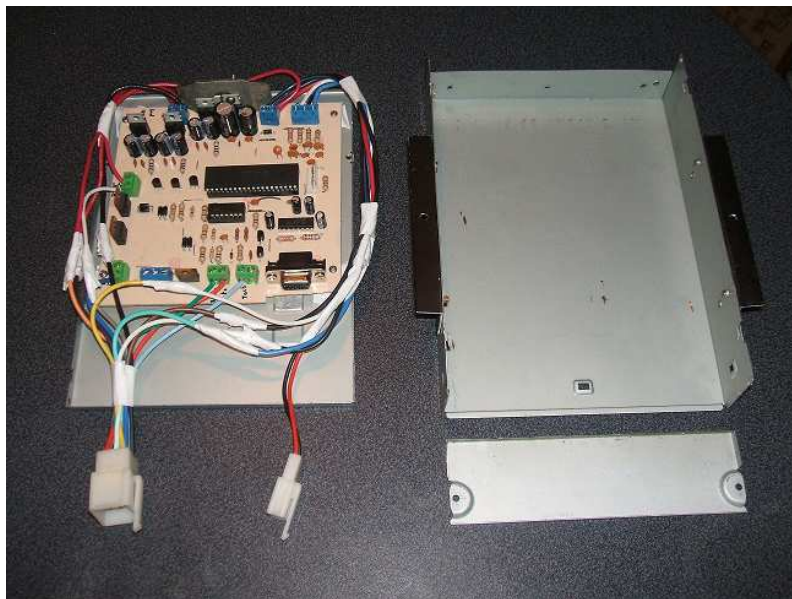


Figura 3.2 Controlador

✓ **Lista de materiales y costos referenciales del proyecto**

La lista de los componentes, cantidades, precios unitarios se encuentran en el anexo A5. Se estima que el costo de materiales del sistema es de 2455 dólares y alrededor de 1000 dólares incluyendo el ensamblaje.

- **Polarización**

El DSC provee 3 pines para la alimentación digital ($V_{cc} = 5V$), 3 pines para la conexión a la tierra digital y un pin de polarización para el ADC (V_{cca}) junto con un pin para la conexión a la tierra analógica. Los capacitores de desacoplo se incluyen en la fuente de alimentación.

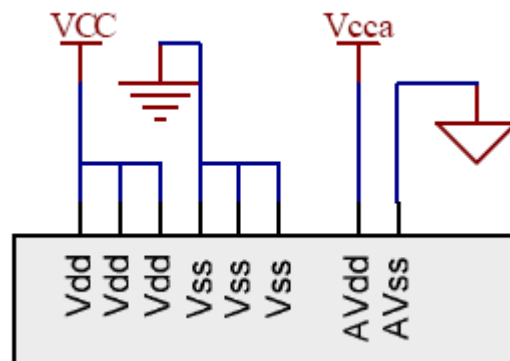


Figura. 3.3 (A2.1)

- **Reset**

El Reset esta conformado por R15 y el conjunto de pines JP2, el mismo que es manejado por el programador.

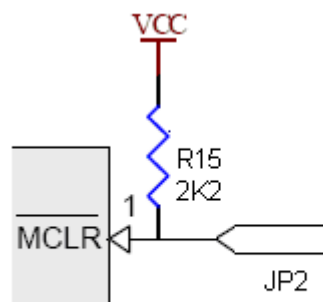


Figura. 3.4 (A2.1)

- **Oscilador**

Si para el caso es necesario trabajar con un cristal externo Y1 de 7.3728MHz (valor comercial). Este es el valor más cercano a 7.5MHz que es el máximo que permite el dispositivo si se habilita el PLL de 16x. Los capacitores C7 y C9 son necesarios para un correcto funcionamiento del cristal (Valores recomendados por el fabricante Microchip [33]). Con el cristal (HC49) se logra tener una precisión de ± 50 ppm.

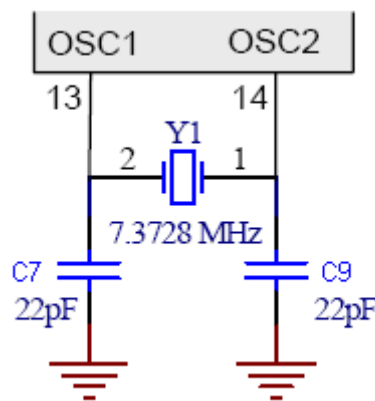


Figura. 3.5. (A2.1)

- **Interfase con el usuario**

Para el caso del sistema de inyección electrónica a gas se proporciona una interfase muy sencilla y básica con el usuario: dos LEDs indicadores, tres botones compactados en un solo selector de tres posiciones. La razón de esto es la enorme simplificación del circuito y del programa al obviar el uso de un LCD y teclado, aparte de que el uso de estos requiere muchas líneas de señal, insuficientes en el 30F4013.

- **Fuente de poder**

Según los requerimientos del fabricante de CI TTL necesitamos fuentes reguladas de 5Vcd, además los sensores utilizan de igual manera 5Vcd para funcionar el cual es una especificación de placa.

El tipo de fuente a usar es lineal y aunque su eficiencia es baja, es importante reconocer la baja generación de ruido lo cual es un atributo reconocido en el diseño de circuitos analógicos. La sencillez de diseño y la facilidad en conseguir las partes en el Ecuador son otros factores importantes por los que se seleccionó este tipo de fuente.

Las corrientes que necesitan cada uno de los dispositivos no son mayores a 500 mA utilizamos el integrado LM 7805 que es un regulador de 5Vcd con una corriente máxima de 1 amperio.

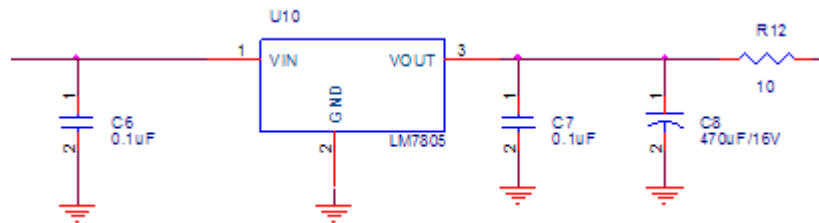


Figura 3.6. Fuente de poder de 5Vcd

- **Componentes**

Conector – Es un conector tipo jack hembra para la conexión del suministro de energía proveniente de la batería del carro ya sea de 12 o 24 Vdc.

Filtro 1 – C6, estabiliza el voltaje de salida del sistema eléctrico del automóvil ya sea de la batería o el regulador electrónico.

LM7805 – Regulador lineal de 5[V]/1.5[A]. Disminuye el voltaje de 12 a 5V.

Filtro 2 – C7 y C8, estabiliza y regula el voltaje que se entrega a los componentes digitales (Vcc).

3.5.2 DIAGRAMA DE BLOQUES

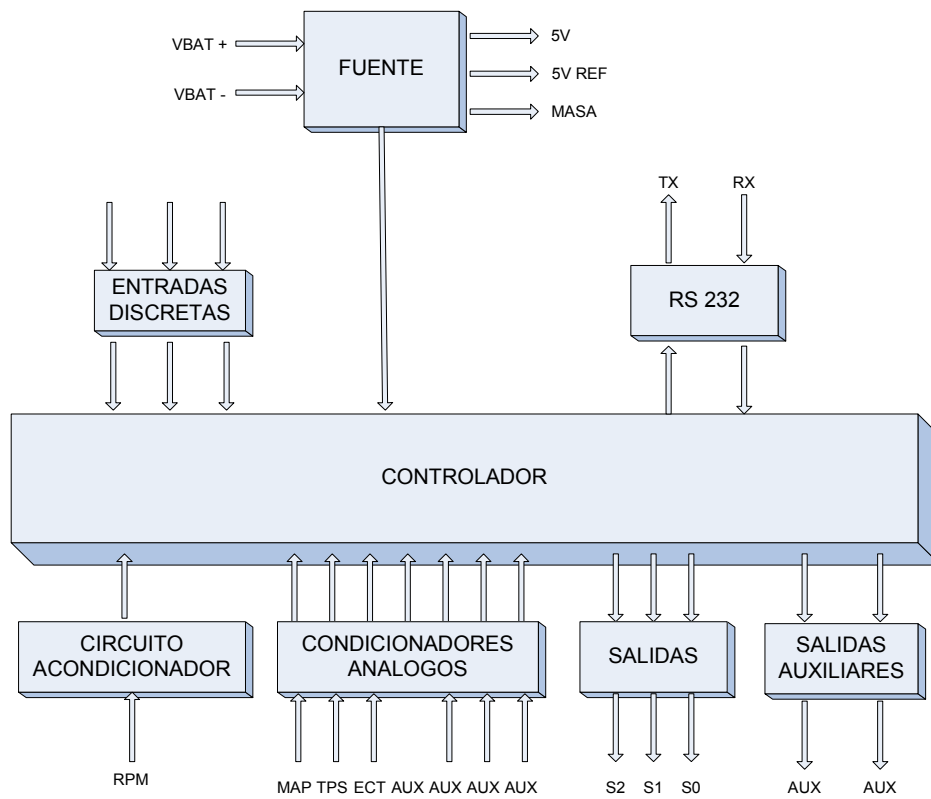


Figura 3.7 Diagrama de bloques

3.5.3 DIAGRAMA DE FLUJO

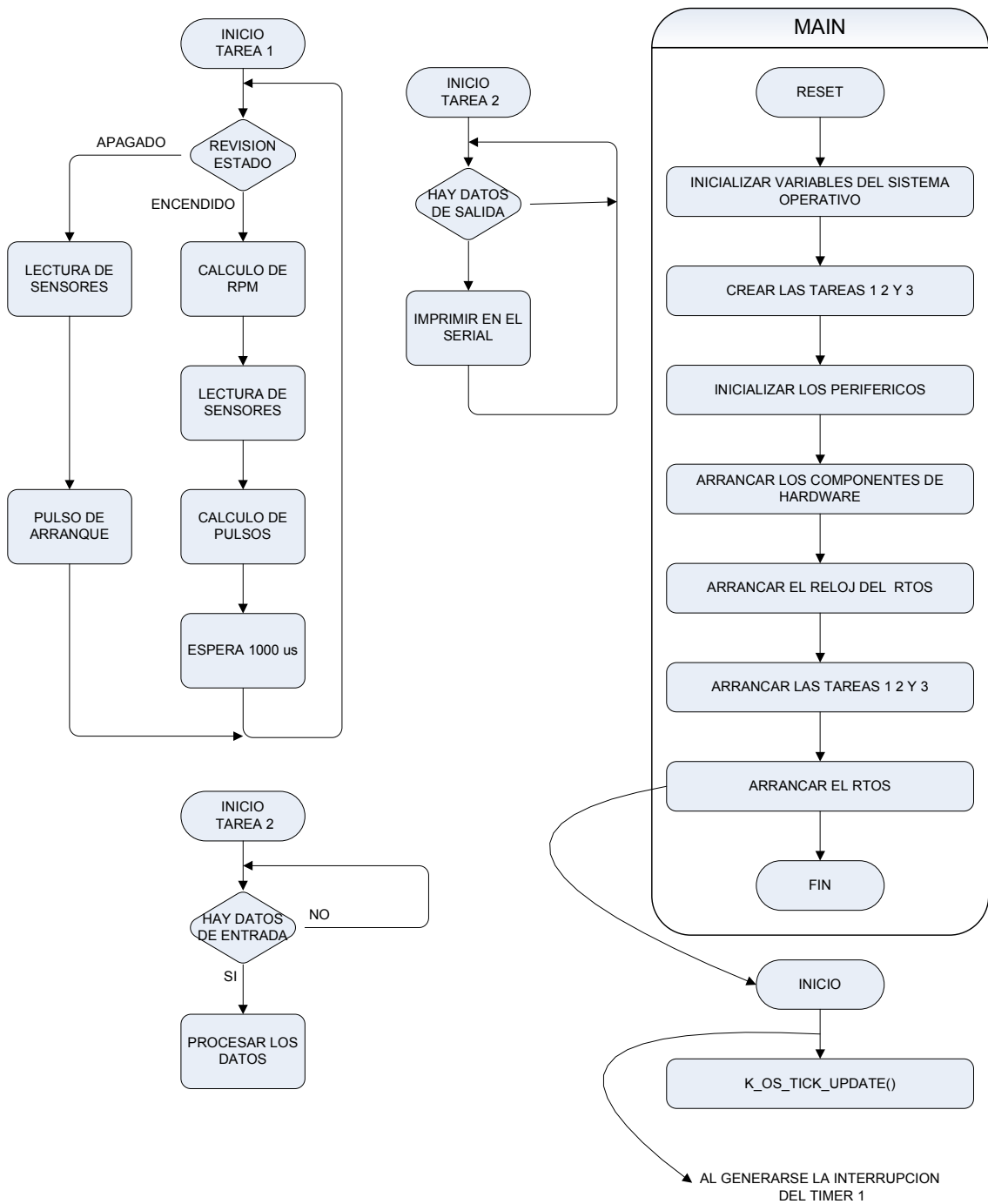


Figura 3.8 Diagrama de flujo

3.5.4 DISEÑO DEL SOFTWARE DEL SISTEMA

3.5.4.1 INTRODUCCIÓN AL DESARROLLO DEL SOFTWARE

✓ Descripción General y especificaciones

Como se detalló en la sección anterior, en este capítulo se imponen los requerimientos de Software y se detalla la manera en que se programó el controlador digital de señales (DSC) para trabajar correctamente en el sistema. Los principales requerimientos y funciones del programa embebido (PE) en el DSC se muestran en la *tabla 3.1*.

Funciones del programa embebido en el DSC	
FPE1	El PE debe administrar las a través de este se descargue de parámetros de calibración pruebas del hardware.
FPE2	El PE debe guardar los datos de configuración y calibración no volátil y permitir recuperar los datos de la misma.
FPE3	El PE realizará todos los cálculos matemáticos correspondientes a las funciones requeridas. Los flujos de estos cálculos y los parámetros de los mismos vendrán dados por los valores almacenados en la memoria no volátil del controlador.
FPE4	El PE es el encargado de gestionar los distintos controles e indicadores de usuario que posee la placa.
FPE5	El PE tiene que configurar todos los periféricos del DSC y los componentes externos programables de manera correcta y oportuna.
<i>Requerimientos del programa embebido en el DSC</i>	

RPE1	El programa debe aprovechar al máximo la arquitectura y los periféricos del dsPIC30F4013 de manera que se obtenga el máximo rendimiento.
RPE2	El programa no debe tener código con funciones que realizan esperas por medio de la ejecución de código inútil.
RPE3	La sección de código que realiza el cómputo del modelo a seguir debe ejecutarse con mayor velocidad que el mínimo ciclo de cómputo que se fije.
RPE4	El programa debe ser escrito de manera ordenada y comentada con el fin de facilitar la corrección, depuración y revisión de terceros.

Tabla. 3.1. Funciones y requerimientos generales del software embebido

3.5.4.2.- CONFIGURACIÓN DEL HARDWARE DEL DSC

✓ Herramientas de desarrollo de software para el DSC

El desarrollo del software se apoya en varias herramientas computacionales que permiten la programación, depuración y análisis del código que se genera. Es importante enumerarlas (*Tabla 3.2*) porque definen en gran manera el método de trabajo y la forma de programar. Adicionalmente es necesario mencionar los manuales, guías de programación y descripción del DSC proporcionados por el fabricante Microchip.

<p>Microchip MPLAB IDE v7.1</p>	<p>Programa para desarrollar código y realizar depuraciones del mismo. Incluye el ensamblador, el <i>Linker</i> y el <i>Librarian</i>. No tiene costo alguno.</p>
	<ul style="list-style-type: none"> • <i>MPLAB IDE® Users Guide (DS51519) [31]</i> • <i>MPLAB ASM30, MPLAB LINK30 and Utilities User's Guide(DS51317) [32]</i> • <i>MPLAB IDE Help</i>
<p>dsPIC Language Suite</p>	<p>Se refiere al compilador en C MPLAB C30. Este permite la creación de código en ANSI C y su código base aprovecha al máximo la arquitectura del procesador. Para este proyecto se utilizó la licencia estudiantil que no realiza ciertas optimizaciones en el código. Este compilador incluye librerías estándar del lenguaje C, funciones para el manejo de los periféricos del DSC y funciones matemáticas.</p>
	<ul style="list-style-type: none"> • <i>MPLAB® C30 C Compiler User's Guide (DS5284) [34]</i> • <i>dsPIC® Language Tools Libraries (DS51456) [35]</i>
<p>MPLAB SIM Software Simulator</p>	<p>Simulador del dispositivo que se ejecuta en una PC y permite ver, modificar e inyectar valores a la memoria. Trabaja bien con C o ensamblador. Viene integrado al MPLAB IDE.</p>
<p>CMX- Scheduler for dsPIC</p>	<p>Librerías que permiten la implementación de un sistema operativo en tiempo real (RTOS) en el dsPIC30F. Es gratis y completamente compatible con las herramientas de Microchip.</p>
	<ul style="list-style-type: none"> • <i>CMX – SHEDULER® User's Manual [36]</i>

Tabla 3.2 Herramientas de Software

Los documentos que contienen la información necesaria sobre la arquitectura, instrucciones y manejo de periféricos del DSC son:

- *dsPIC30F Family Reference Manual (DS70046)* [22]
- *dsPIC30F Programmer's Reference Manual (DS70030)* [37]
- *dsPIC30F Data Sheet General Purpose and Sensor Families (DS70083)* [38]
- *dsPIC30F3014, dsPIC30F4013 Data Sheet (DS70138)* [39]

3.6 PROGRAMACION DE LA TARJETA DE CONTROL

- **Herramientas de descarga de código**

Uno de los mayores impedimentos al probar un nuevo microcontrolador o DSC es la manera en que al dispositivo se lo programa, esto es, se le transfiere desde el computador el archivo con el código máquina (generalmente con extensión .hex) del programa que se ha compilado. Una de las grandes ventajas de los DSC de la familia Microchip es que se puede realizar este proceso con ayuda de un circuito sencillo que, por medio del puerto serial RS232, descarga el código al mismo. En el Anexo A6 se detalla el circuito (conocido como JDM Programmer).

Para la descarga de los archivos .hex por medio de este método se utiliza el programa WinPic800 [22]. El proceso anterior permite grabar todos los registros de memoria, pero es necesario sacar y poner el dispositivo en el grabador o equipar al circuito impreso final con este circuito. Un método más conveniente es el usar un *Bootloader* (cargador de arranque) que no es más que un pequeño código que se ejecuta justo después del arranque del dispositivo y controla la descarga del programa por medio de cualquier canal de comunicación del DSC. La ventaja radica en que se utiliza la misma infraestructura ya adecuada de la placa (en el sistema el puerto serial y sus drivers) y elimina la necesidad de sacar el dispositivo cada vez que se desea programar. Para este proyecto se utilizó el *Ingenia dsPICbootloader* [39]. Los pasos para implementar el bootloader son:

1. Adecuar el programa en ensamblador que proporciona Ingenia del DSC

para el modelo específico que se va a usar.

2. Con la ayuda de MPLAB IDE, compilar el archivo mencionado y generar el correspondiente archivo .hex.

3. Usando un grabador JDM (Anexo A6) y cargando el archivo .hex al programa WinPic800, descargarlo al DSC.

5. Instalar el programa *Ingenia dsPICbootloader*.

6. Suponiendo que el circuito de aplicación tiene correctamente implementado la comunicación RS232, seguir los pasos que se describen en el manual de operación del bootloader.

3.6.1 CONFIGURACIÓN DEL HARDWARE DEL DSC

✓ Configuración de Recursos y Periféricos del DSC

El dispositivo dsPIC30F4013 posee una serie de funcionalidades y periféricos que permiten cumplir con los requerimientos del sistema. La configuración de estos recursos viene dada principalmente por el hardware que se diseñó y que se detalló en el capítulo anterior.

✓ Oscilador

Es la unidad encargada de generar el reloj del dispositivo. Para el sistema se ha buscado la máxima velocidad por lo que se utiliza un cristal externo (XT) de 7.3728 MHz y adicionalmente se configura al PLL interno en aumento de 16x. Esto da un reloj interno de 117.9648 MHz. Cada ciclo de instrucción (Fcy) requiere 4 ciclos de reloj, por lo que la velocidad de procesamiento es de 29.4912 MIPS (Millones de instrucciones por segundo).

✓ Funciones del sistema

Se tiene deshabilitada las funciones de POR (Reset al arrancar), BOR, LVD (Detección de bajo voltaje) y Watchdog Timer.

✓ **Interrupciones y Traps**

Las interrupciones externas INTx se dejan deshabilitadas mientras que los Traps, o condiciones de error en la pila, en acceso a memoria, en operaciones matemáticas o en el oscilador se dejan habilitadas para indicar cualquier error de este tipo durante la ejecución del programa..

✓ **Convertor Análogo-Digital de 12 bits**

Para este convertor se habilitan dos canales: AN9 (Señal externa acondicionada) y AN10 (Potenciómetro de variación). El voltaje de referencia alto (VREF+) corresponde al voltaje de 4.096V y el voltaje de referencia bajo (VREF-) corresponde a tierra.

El reloj del módulo ADC corresponde al generador interno del mismo y trabaja a una velocidad de 1.03us (T_{AD}). Se deja configurado para que el módulo realice conversiones automáticas alternadas entre AN9 y AN10.

El tiempo de muestreo se fija como $5 T_{AD} = 5.15$ us que, sumado a los $14T_{AD}$ que se requiere en la conversión, resulta en un tiempo total de $19 T_{AD} = 19.57$ us para cada adquisición. Debido a que hay dos canales, el período de muestro para las dos señales es 39.14 us ó 25.55kHz. El buffer de resultados se lo configura dividido, de esta forma se obtiene la secuencia de conversión: AN9–ADCBUF0, AN10–ADCBUF1, AN9–ADCBUF8, AN10–ADCBUF9, AN9–ADCBUF0

De esta manera antes de leer desde el programa principal el valor de AN9 o AN10 se verifica en qué sector del buffer (ADCBUF0-7 ó ADCBUF8-15) se realiza la conversión actual y se procede a leer el valor del otro sector del buffer. Es así que se asegura la lectura de un valor correcto del ADC sin tener que sincronizar la lectura con el fin de la conversión.

- Puertos

Puerto A			
Pin	Sentido	Estado Inicial	Función
RA11	Entrada	-	Entrada digital externa I3

Puerto B			
Pin	Sentido	Estado Inicial	Función
RB0	Entrada	-	Configurados como entrada para que el módulo del ADC trabaje correctamente.
RB1	Entrada	-	
RB2	Entrada	-	
RB3	Entrada	-	
RB4	Entrada	-	Entradas de los pulsadores. Se deja la posibilidad de configurar los pull-up's internos para sacar las resistencias externas.
RB5	Salida	1	Salida que accionan los LEDs indicadores. Trabajan como drenadores de corriente.
RB6	Salida	1	
RB7	Salida	1	
RB8	Salida	0	Es la salida para realizar las pruebas de calibración del reloj.
RB9	Entrada	-	Configurados como entrada para que el módulo del ADC trabaje correctamente.
RB10	Entrada	-	
RB11	Salida	1	Activa/Desactiva el CS del DAC
RB12	Salida	1	Activa/Desactiva los CSs de los potenciómetros digitales.

Puerto D			
Pin	Sentido	Estado Inicial	Función
RD0	Salida	1	Salida digital externa O1
RD1	Salida	1	Salida digital externa O2
RD2	Salida	1	Salida digital externa O3
RD3	Salida	1	Salida digital externa O4/Salida PWM
RD8	Entrada	-	Entrada digital externa I1
RD9	Entrada	-	Entrada digital externa I2

Puerto F			
Pin	Sentido	Estado Inicial	Función
RF4	Entrada	-	Salida opcional para comunicación por RS232 usando handshake.(CTS) ó Rx del módulo UART2.
RF5	Salida	1	Entrada opcional para comunicación por RS232 usando handshake.(RTS) ó Tx del módulo UART2.

Tabla. 3.3. Configuración de los puertos

- **Módulo de comunicación SPI**

Este módulo se configura como maestro sin sincronización de trama, en modo de transmisión de 16 bits, con estado de espera en alto, captura de dato a la mitad del reloj y transmisión en el flanco positivo del reloj.

La frecuencia de operación se fija en 4.915 MHz para cumplir con los requerimientos de velocidad de los potenciómetros digitales. A esta velocidad del SPI es posible hacer que el DAC tenga un nuevo dato cada 3.26 us ó a 306.748 kHz.

- **Módulo UART1**

Este módulo está configurado para trabajar en los pines alternos (U1ATX y U1ARX). Aunque este módulo tiene una gran cantidad de funcionalidades, para el sistema este trabaja a 9600 kbps, sin paridad y con un bit de parada (8N1).

- **Módulo CAN**

No se utiliza en la aplicación del sistema de inyección a gas pero se deja la posibilidad que para futuras aplicaciones se implemente este módulo ya que el hardware está listo para ser habilitado.

- **Temporizadores**

El Timer1 trabaja como generador de ticks del sistema. Su interrupción es la encargada de invocar a K_OS_Tick_Update() que es la que temporiza al sistema operativo. El Timer2 es ocupado por el módulo de comparación de salida y se utiliza cuando se configura al sistema para sacar una señal de RPM. El Timer3 es ocupado por la función de escritura al DAC y permite regular el ancho de pulso en los inyectores

- **Funciones de configuración de los módulos**

Función de entrada	void iniPeripherals(void)	Archivo
ADC de 12 bits	.global _configADC	drivers.h, config.s
Puerto A	.global _configPortA	
Puerto B	.global _configPortB	
Puerto D	.global _configPortD	
Puerto F	.global _configPortF	
SPI	.global _configSPI	
UART1	.global _configUART	
Timer1	.global _configTimer1	
Timer2	void iniPWMFunction(void)	plant.h, plantcomp.c
Timer3	.global _configTimer3	drivers.h, config.s
OC4	void iniPWMFunction(void)	plant.h, plantcomp.c

Tabla. 3.4. Funciones de configuración.

3.7. PROGRAMA DEL DSC

- **Breve descripción del sistema operativo multitarea**

Para el desarrollo del software del Emulador SISO se escogió basar el programa en un sistema operativo en tiempo real (RTOS) perteneciente a CMX, el CMX-Scheduler.

El CMX-Scheduler permite implementar un ambiente multitarea administrado con derecho a preferencia (*preemptive*) sobre los procesadores 24F/30F/33F de la familia Microchip. CMX provee las funciones necesarias y el sistema operativo para escribir aplicaciones multitarea de forma ordenada y bien diseñada. Esto evita que se tenga un código engorroso y lleno de bucles, condicionales y banderas lo que demora la detección de errores y la depuración en general. Adicionalmente ayuda a analizar y cumplir los tiempos necesarios para que el programa se comporte en tiempo real.

El sistema multitarea permite que en un solo procesador se ejecuten muchas tareas al conmutar concurrentemente las distintas tareas con el fin de obtener el efecto de que

todas operan simultáneamente. Cada tarea representa un proceso en particular y puede ser coordinada con otras para definir su flujo. El CMXScheduler posee las funciones [36] descritas en la *Tabla 3.5*:

K_Task_Create	Crea una nueva tarea
K_Task_Start	Inicia una tarea
K_Task_Wake	Despierta una tarea
K_Task_Wait	Se mantiene en espera a una tarea
K_Task_Kill	Se elimina una tarea
K_Task_Coop_Sched	Realiza un cambio cooperativo
K_Event_Wait	Espera por un evento
K_Event_Signal	Se señala un evento desde una tarea a otra
K_Event_Signal	Se señala un evento desde una interrupción a una tarea
K_Event_Reset	Resetea un evento para una tarea en particular.

Tabla. 3.5 Funciones del RTOS CMX-Scheduler

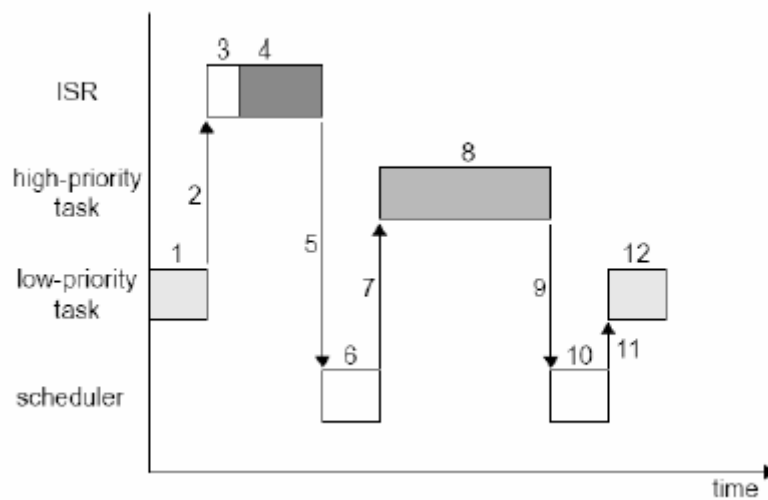


Figura. 3.9. Diagrama de tiempo de un RTOS con derecho de preferencia.

Una discusión completa sobre RTOS está fuera del alcance de este proyecto, por lo que se puede ampliar este tema con las referencias [31] y [32].

- **División funcional del programa**

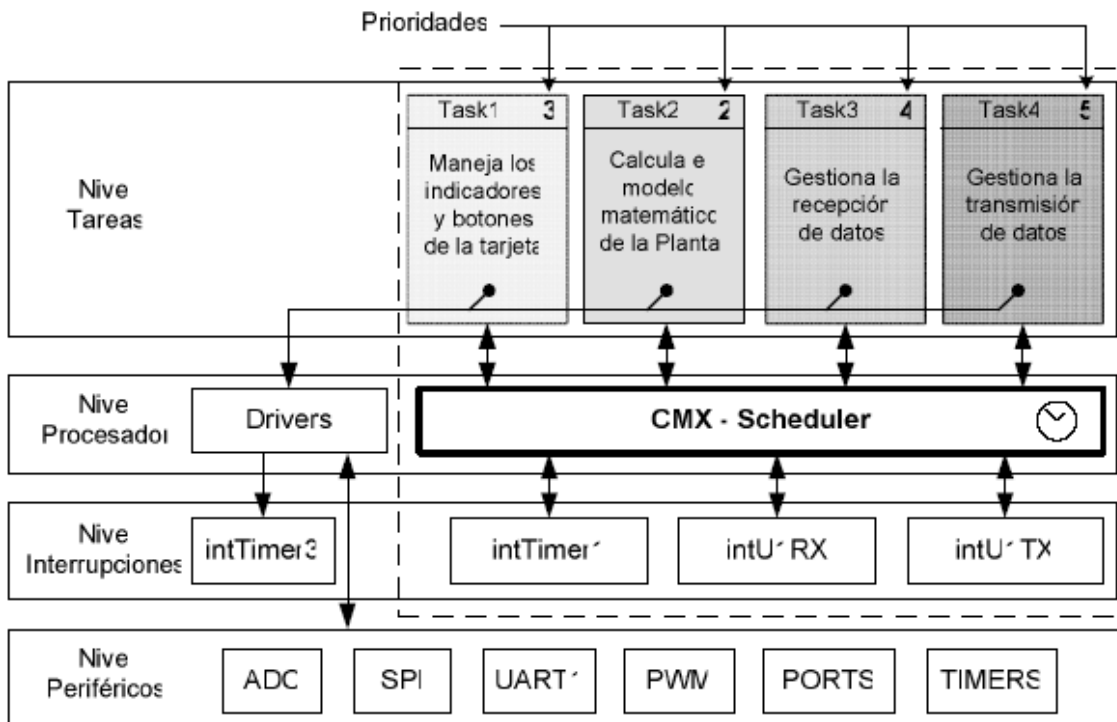


Figura. 3.10 Interacción entre los elementos de Software

La *Figura 3.10* permite tener una perspectiva de los distintos elementos que componen el software embebido en el sistema y las distintas interacciones entre sí. Los únicos procesos que se ejecutan realmente en paralelo son los de nivel de procesador y nivel de periféricos, éstos últimos permiten contar con procesos automáticos a nivel de hardware ya que alivian tareas como la adquisición de los datos analógicos (ADC), la temporización (Timers), el envío y recepción de datos por SPI ó UART y la generación de pulsos PWM. El tiempo de uso del procesador debe compartirse en los niveles de interrupción, procesador y tareas. Desde un punto de vista de software, el DSC está dividido en 4 Niveles de ejecución que debido al uso del RTOS hace posible suponer que se ejecutan simultáneamente. Las interrupciones son generadas siempre por los periféricos, el nivel de procesador corresponde al código de los drivers y el RTOS, mientras que el código de las tareas es administrado por el RTOS.

Las prioridades de las 4 tareas se encuentran indicadas en la *Figura 3.10* (mientras menor el número, mayor es la prioridad). La ejecución de cada una de ellas es

arbitrada por el CMX-Scheduler, el cual actualiza el estado de las variables de cada tarea, los temporizadores y eventos cada vez que se invoca a `K_OS_Tick_Update()` al producirse la interrupción del Timer1. Por esta razón la resolución de los timers del RTOS viene dada por la frecuencia de llamado a la función de actualización (tick del OS).

Mientras mayor número de llamadas a esta función se tenga en un segundo, más resolución en las tareas de espera se logra pero a su vez se pierde tiempo útil del procesador. Para el caso del sistema se estableció una frecuencia de tick de 10kHz en la cual el uso del RTOS “roba” aproximadamente el 1% del tiempo del procesador (trabajando a 29.4912 MIPS). Cada tarea cumple las siguientes funciones:

Task1: Es la encargada de realizar la revisión de los estados de los sensores. Actualiza las banderas indicadoras y, si las condiciones son correctas, inicia la Task2.

Task2: Esta tarea se crea cuando el sistema empieza a recibir señal de RPM, Carga los datos desde la EEPROM que configura las funciones de retardo. Un timer del sistema operativo controla la frecuencia del bucle. Lee el ADC y escribe al DAC con el resultado del cálculo; también actualiza el ciclo de trabajo del generador PWM.

Task3: Espera la interrupción debida a la recepción de un nuevo byte por el UART. Invoca a la función que procesa los datos del serial.

Task4: Esta tarea esta definida en el programa para satisfacer posteriores requerimientos.

- **Comunicaciones en el Sistema**

- ✓ **Recepción y transmisión de datos**

Los diagramas de flujo de la transmisión y recepción de datos se muestran en la *Figura 3.9* Este flujo es aplicable sea cual sea el protocolo de la aplicación. Para el caso

de la recepción, el manejo lógico de la comunicación (protocolo) es manejada por la función UartMain(),



Figura. 3.11 Diagrama de comunicaciones

3.8. PRUEBAS, DETECCIÓN Y CORRECCIÓN DE ERRORES

- Descripción del capítulo

En este capítulo se presentan las pruebas realizadas al de inyección electrónica a gas y los resultados de las mismas. Se describen dos tipos de pruebas: las primeras obtienen el comportamiento del sistema como equipo de adquisición de datos y las características del mismo

En primeras instancias se tomo la señal de rpm de la bobina de encendido teniendo las siguientes señales en los inyectores y sincronismo. Como mostramos en las figuras 3.12 a 3.17

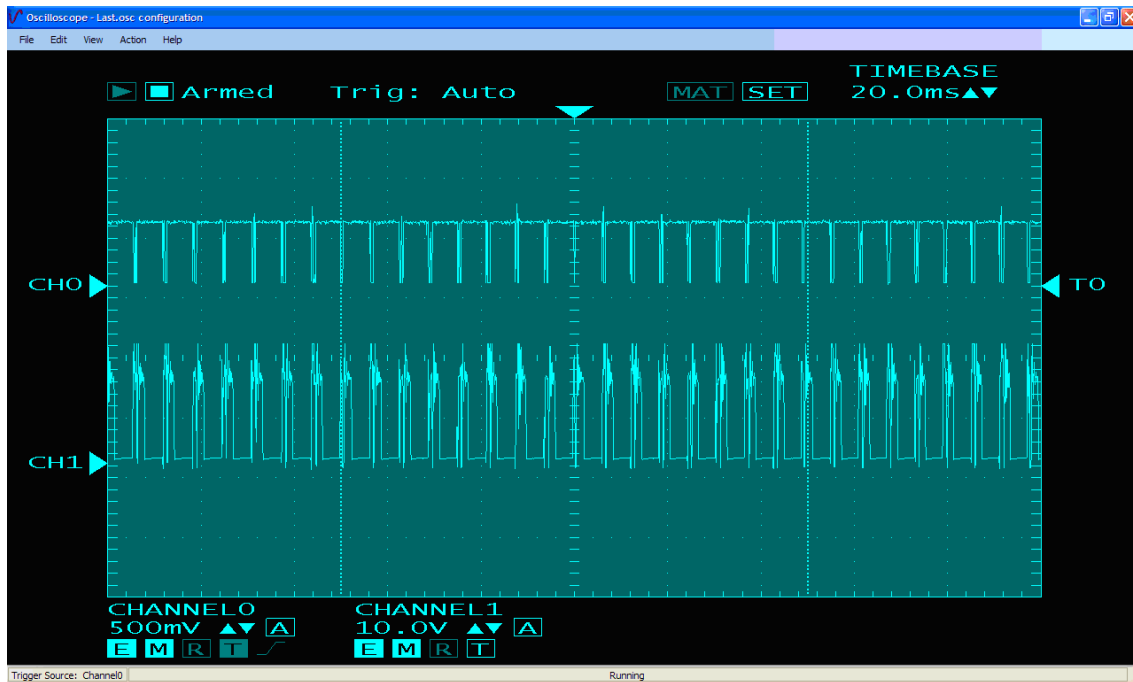


Figura. 3.12 señal del PC osciloscopio

Podemos observar que existe mucho ruido producido por efecto de la autoinducción en la bobina de encendido.

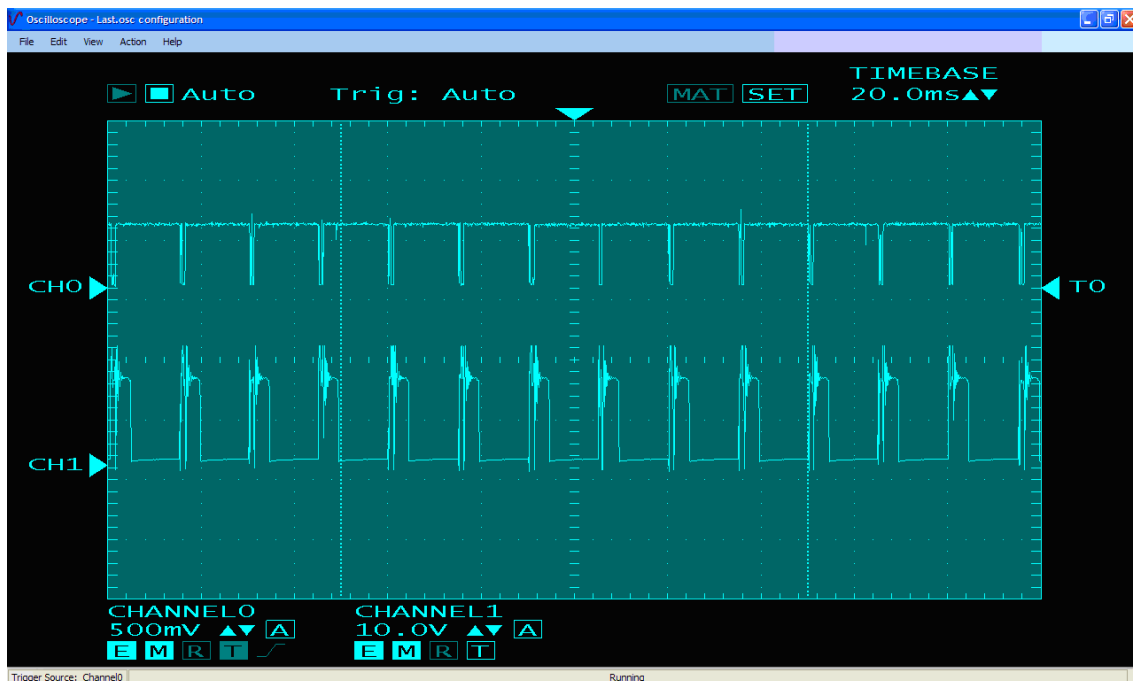


Figura. 3.13 señal del PC osciloscopio

La misma señal con $T=5\text{ms/div}$

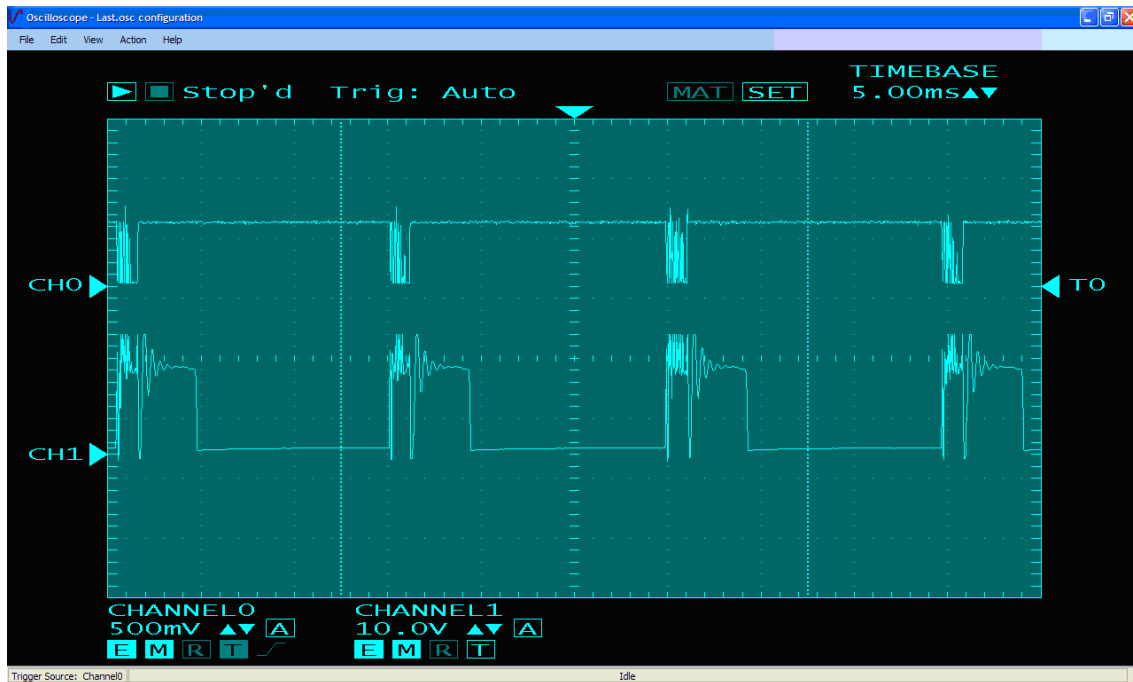


Figura. 3.14 señal del PC osciloscopio

Ahora tenemos en los dos canales del DAQ la señal entregada en los inyectores en dos escalas donde podemos observar que el ruido que ingresa al controlador no nos permite hacer ningún tipo de control como se muestra en las figuras

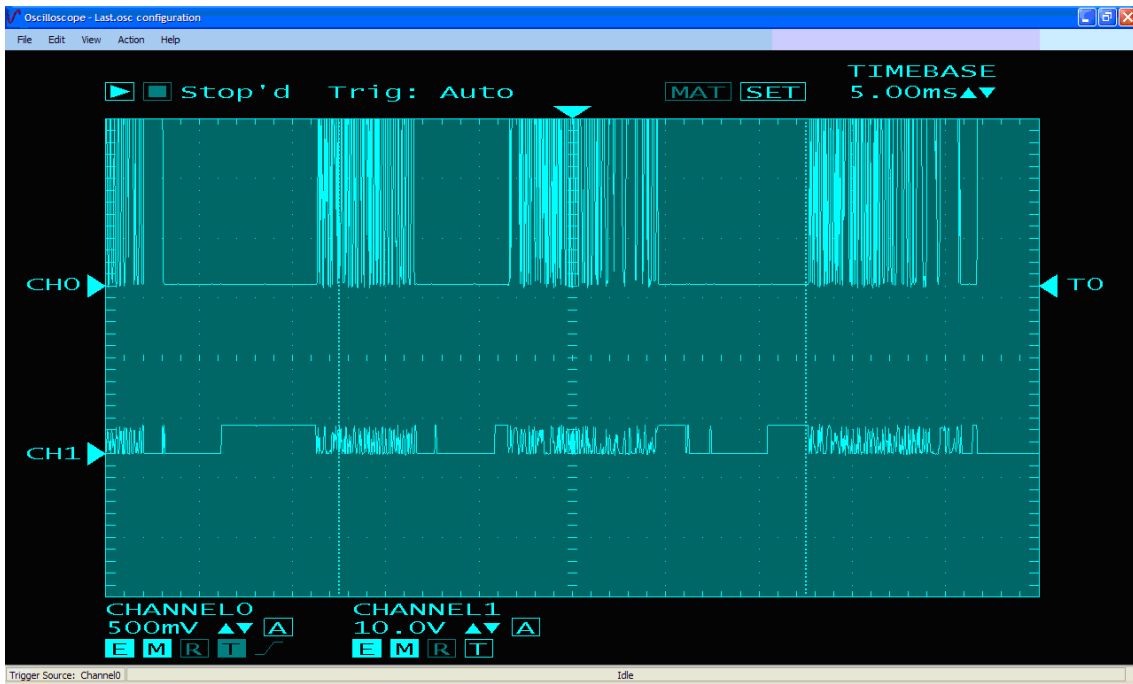


Figura. 3.15 señal del PC osciloscopio

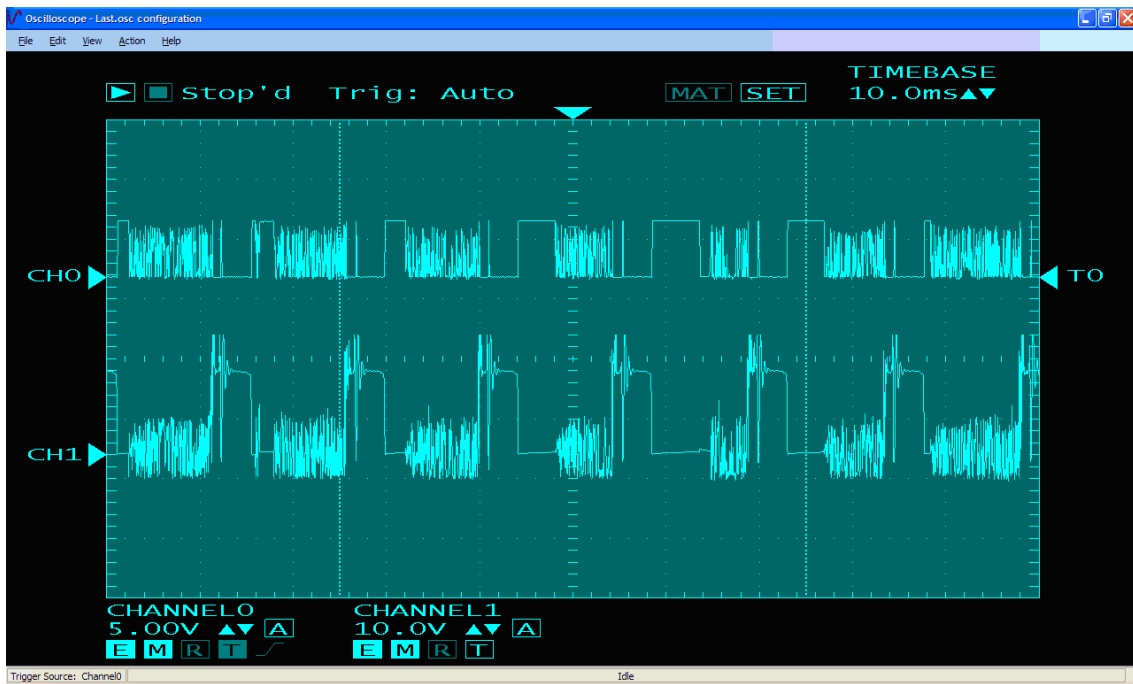


Figura. 3.16 señal del PC osciloscopio

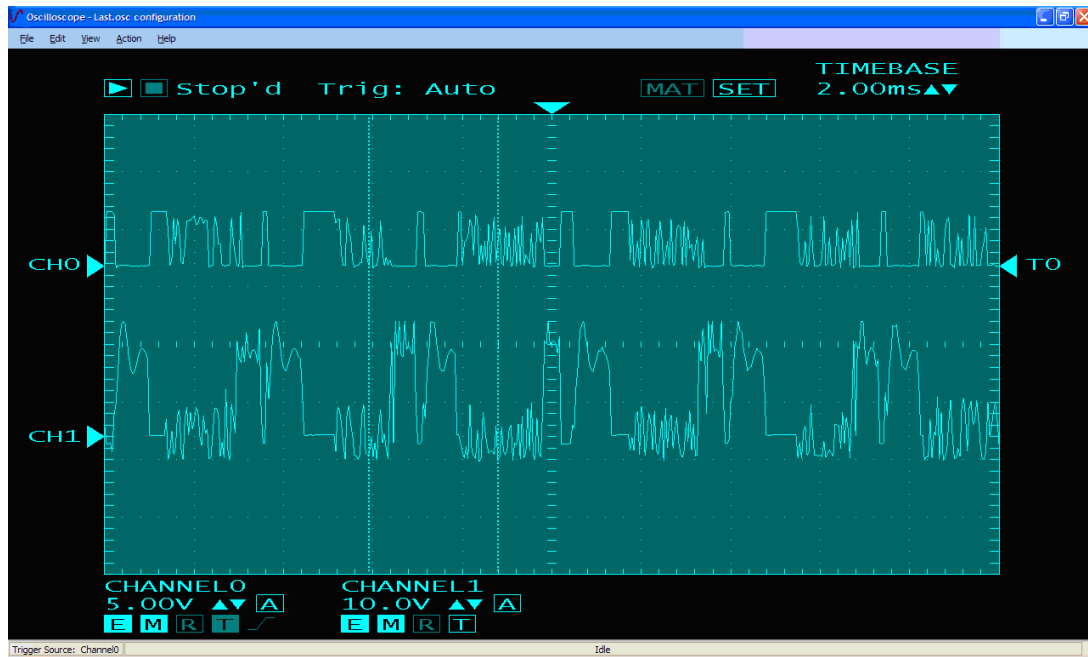


Figura. 3.17 señal del PC osciloscopio

Por otra parte también se intento utilizar otra señal entregada por el modulo electrónico de encendido puesto que en primeras pruebas con osciloscopio analógico se mostraba pura pero con el DAQ podemos notar que de la misma manera posee ruido a mas de tener un offset o componente de cd. Figura 3.18

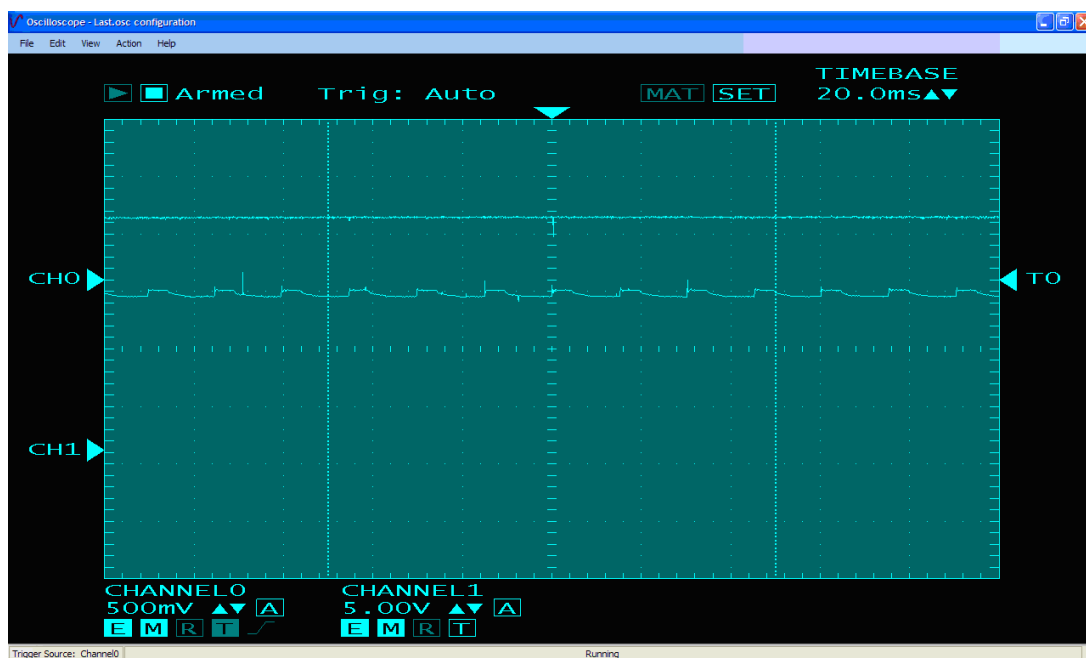


Figura. 3.18 señal del PC osciloscopio

El problema más grande fue que se perdió el sincronismo debido al ruido generado por el sistema de ignición electrónico generando disparos falsos. Figura 3.19

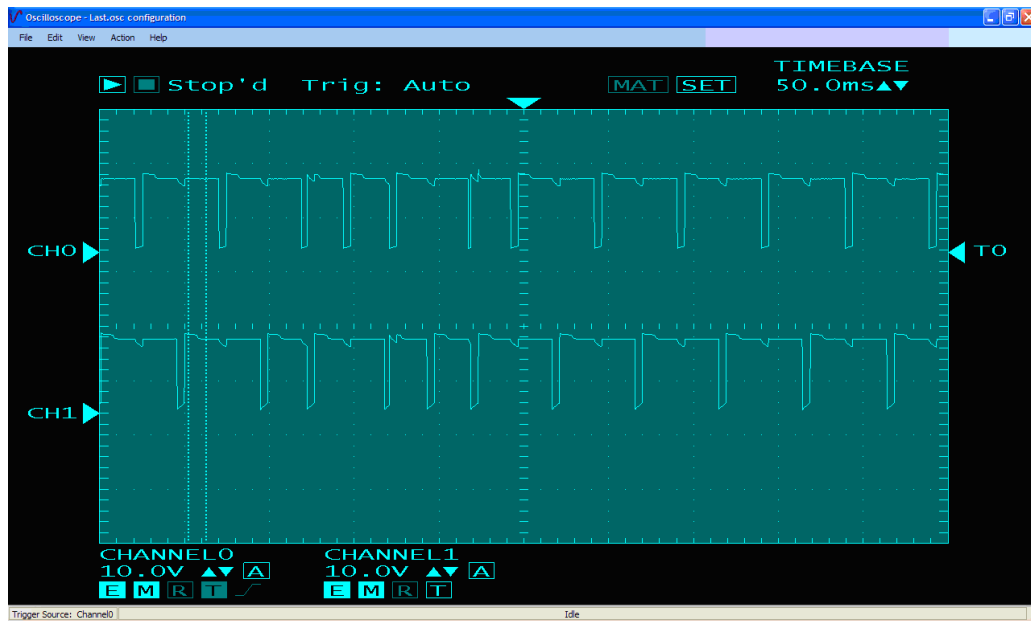


Figura 3.19 Pantalla señal de inyectores

En consecuencia los disparos en los inyectores se generan con incertidumbre. Figura 3.20

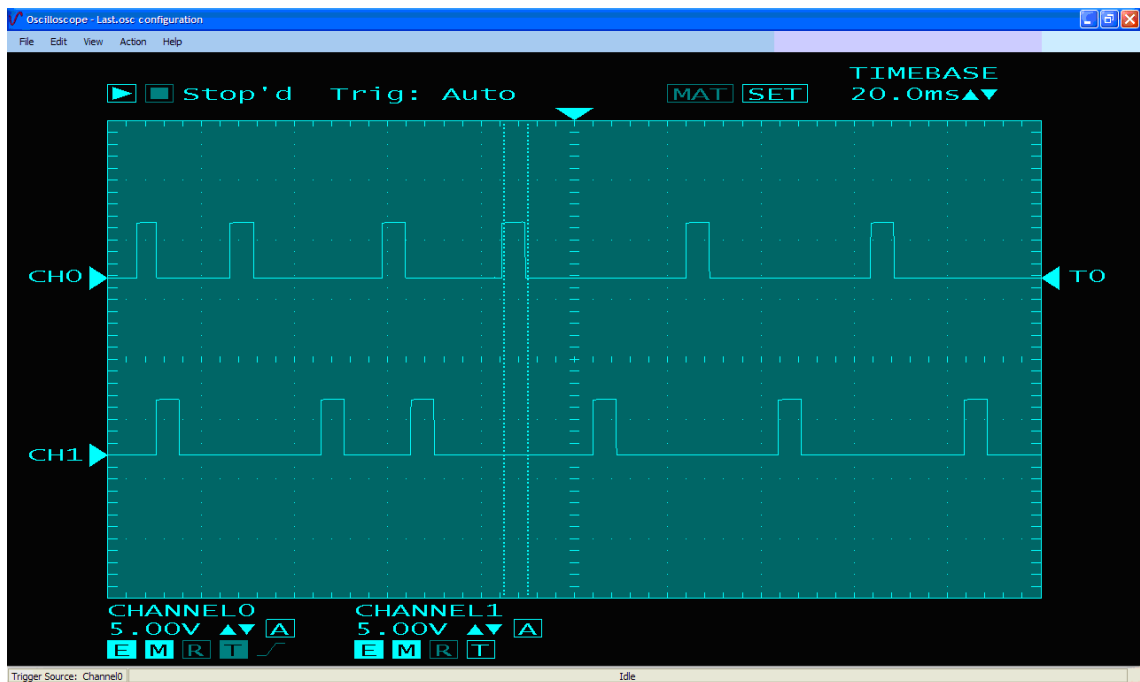


Figura 3.20 Pantalla señal de inyectores

En la búsqueda de señal adecuada tomamos la señal directamente del mismo sensor inductivo del sistema de encendido electrónico y pudimos darnos cuenta que esta señal era mas pura como se muestra en las figuras 3.21 y 3.22

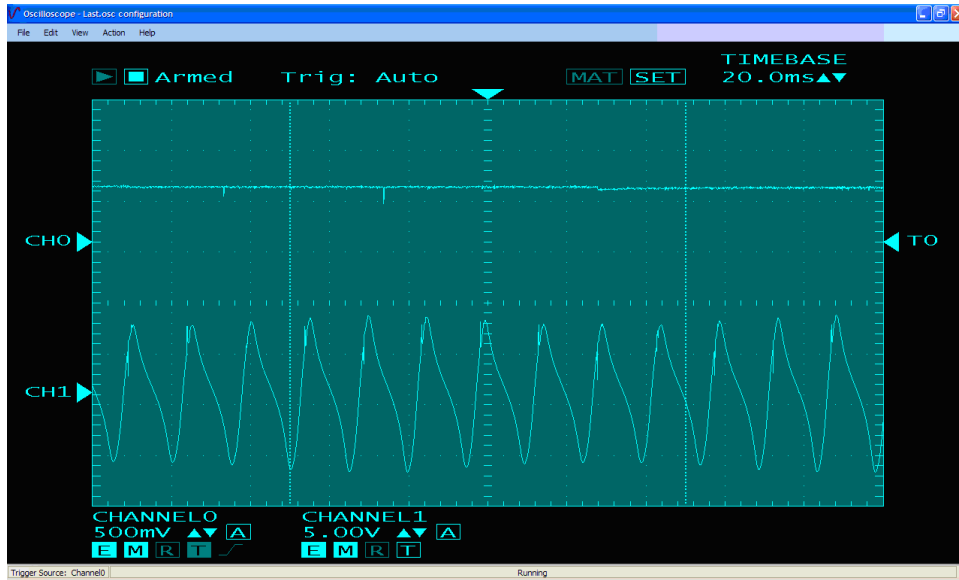


Figura. 3.21 señal del PC osciloscopio

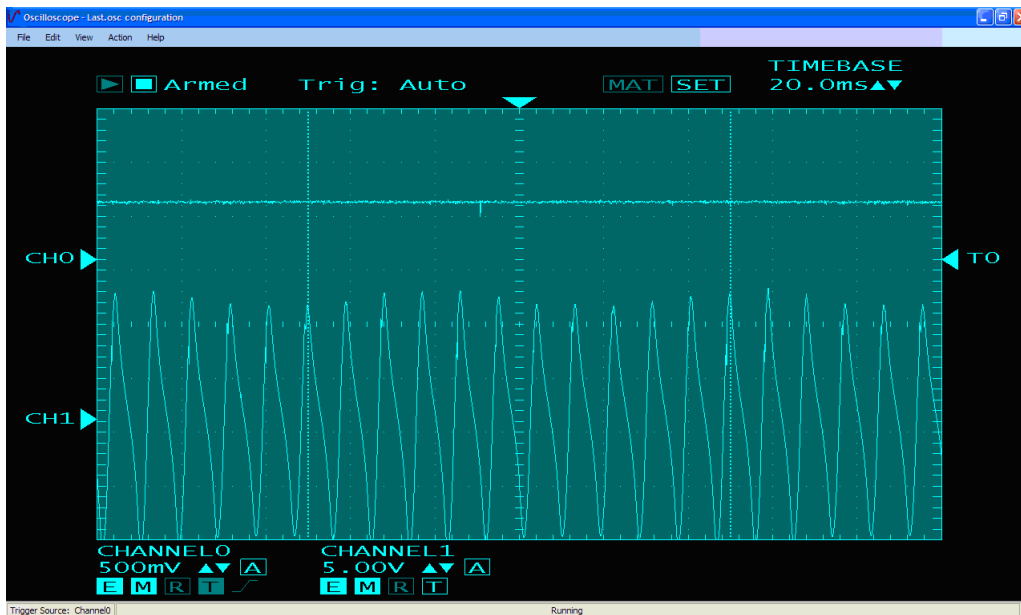


Figura. 3.22 señal del PC osciloscopio

Para solucionar el problema del ruido se inserto un detector de cruce por cero el cual nos entrega una señal cuadrada perfecta figura 3.23

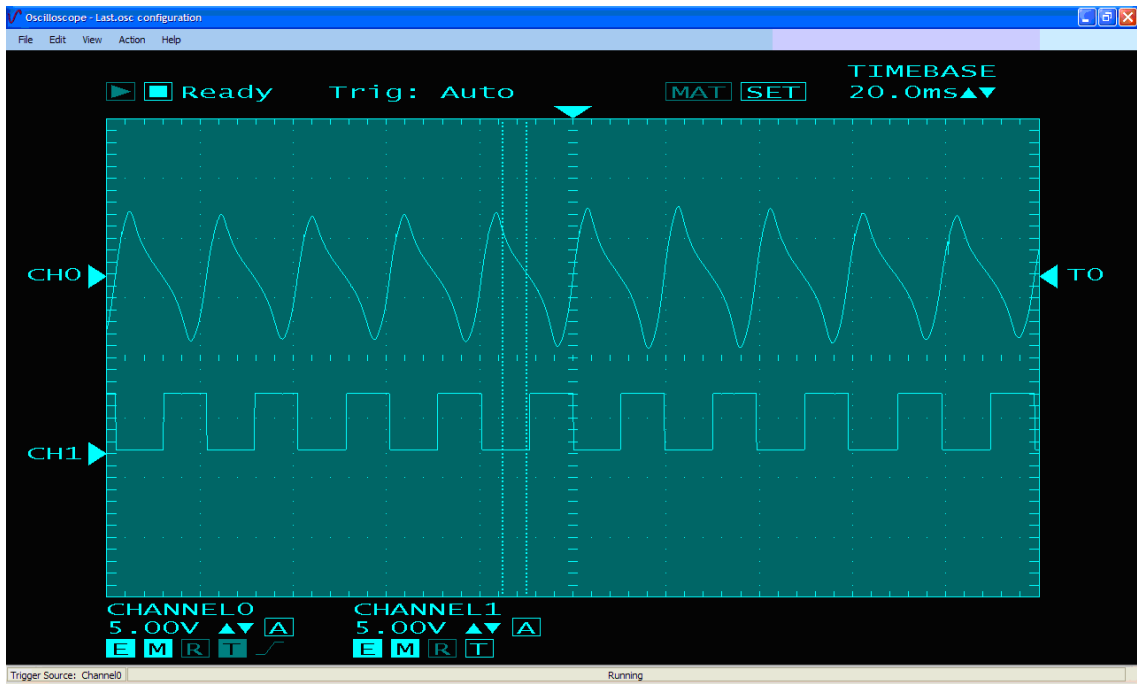


Figura 3.23 Pantalla detector de cruce por cero

Con esta señal que ya es perfecta se soluciono los problemas de falsos disparos en los inyectores figura 3.24

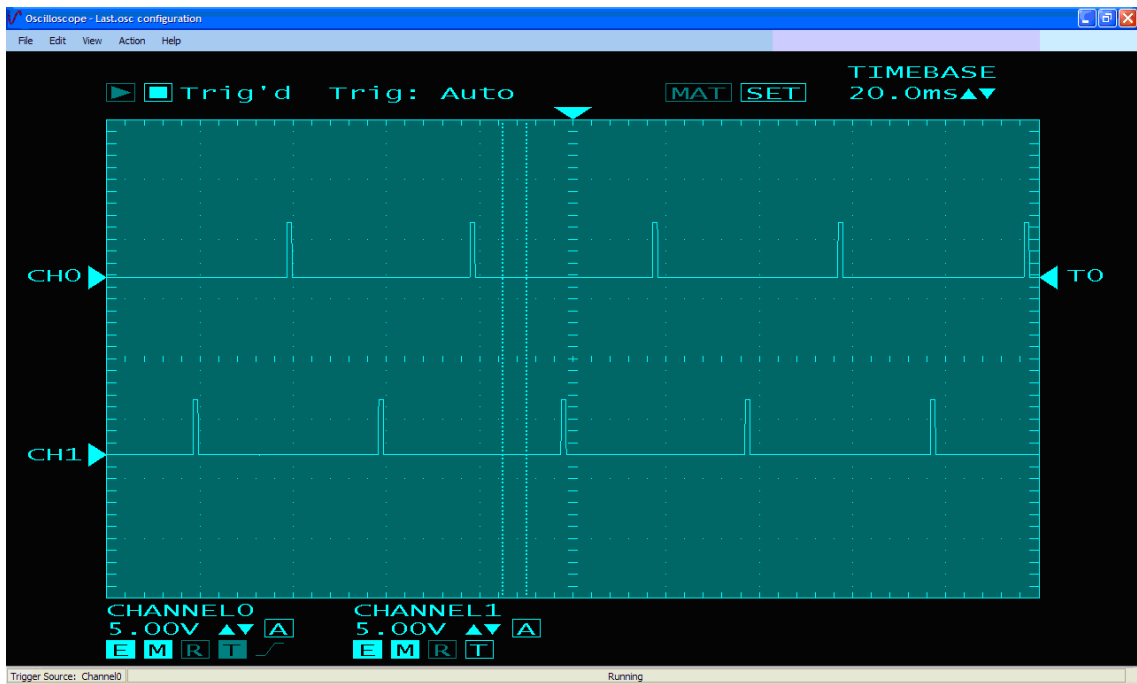


Figura 3.24 Pantalla Señal de inyectores

Podemos notar que la señal esta alternándose correctamente saltando un ciclo de sincronismo cada inyector figura 3.25

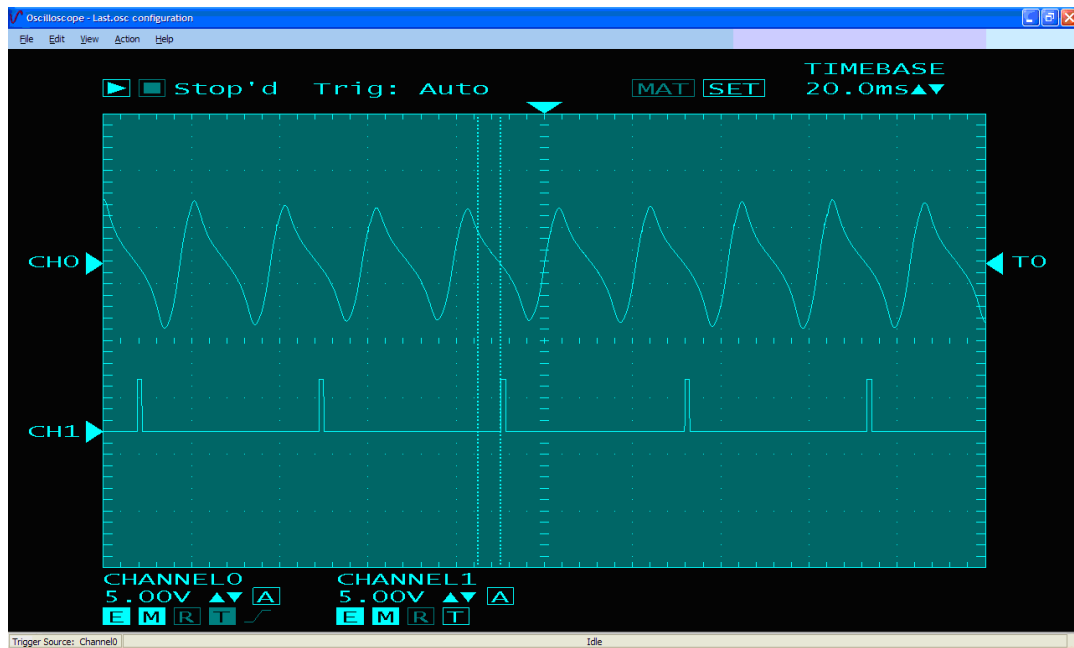


Figura 3.25 Pantalla inyector y sensor inductivo

Asi como también logramos que el disparo de cada inyector sea en el cero preciso figura 3.26

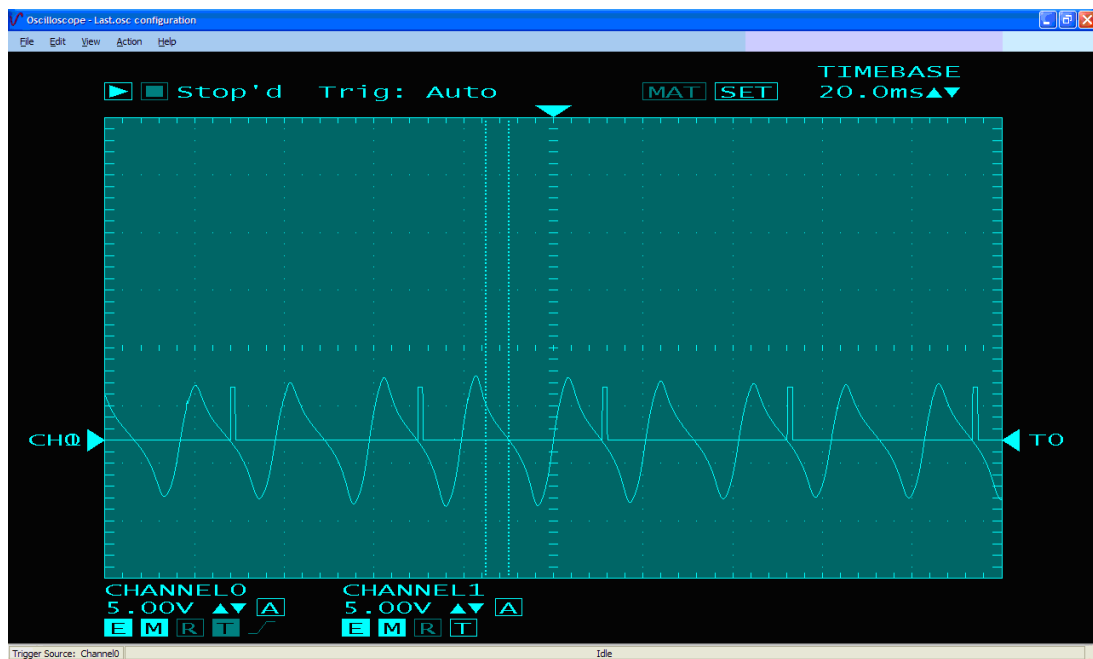


Figura 3.26 Pantalla inyector y sensor inductivo

Otra proceso de pruebas importantes se realizo descargando los datos enviados por los sensores a través del puerto RS-232 para generar y cargar la tabla de datos que contenga la curva característica de funcionamiento de cada sensor

CAPITULO IV

INSTALACIÓN DEL SISTEMA

En primera instancia se colocó todo lo que corresponde al sistema de combustible ya que para comenzar a trabajar con la electrónica debe estar toda la parte mecánica lista tal como se muestra en la figura 4.1 [2].

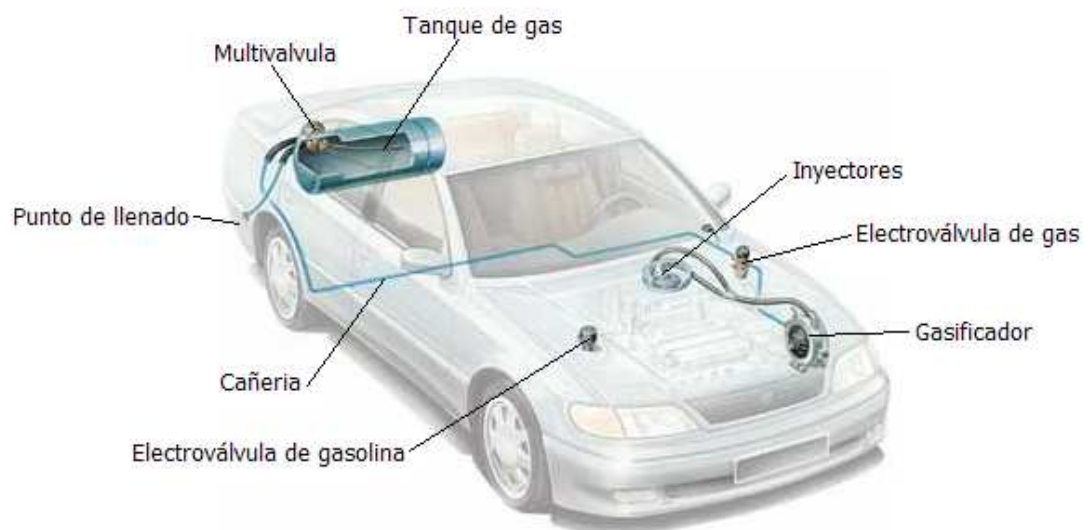


Figura 4.1 integración del sistema de gas

4.1 INTEGRACIÓN DE TARJETA DE CONTROL , SENSORES Y ACTUADORES

Una vez instalados todos los elementos del sistema de control procedemos a conectarlos con la tarjeta de control se ha utilizado cable de distintos calibres de acuerdo al consumo de corriente de cada dispositivo como se muestra en la tabla 4.1.

ELEMENTO	CORRIENTE MAXIMA	CALIBRE
IAT	100 mA	18 AWG
TPS	80 mA	18 AWG
MAP	20 mA	18 AWG
RPM	100 mA	18 AWG
INYECTORES	3.5 A	16 AWG
ELECTROVALVULAS	500 mA	16 AWG
ALIMENTACION	4 A	16 AWG
CONMUTADOR	4 A	18 AWG

Tabla 4.1 Designación de cable

Cada sensor y actuador esta provisto por un conector genérico de 4 pines macho con su respectivo conector hembra que lo conecta con la unidad de control como se muestra en la figura 4.2.

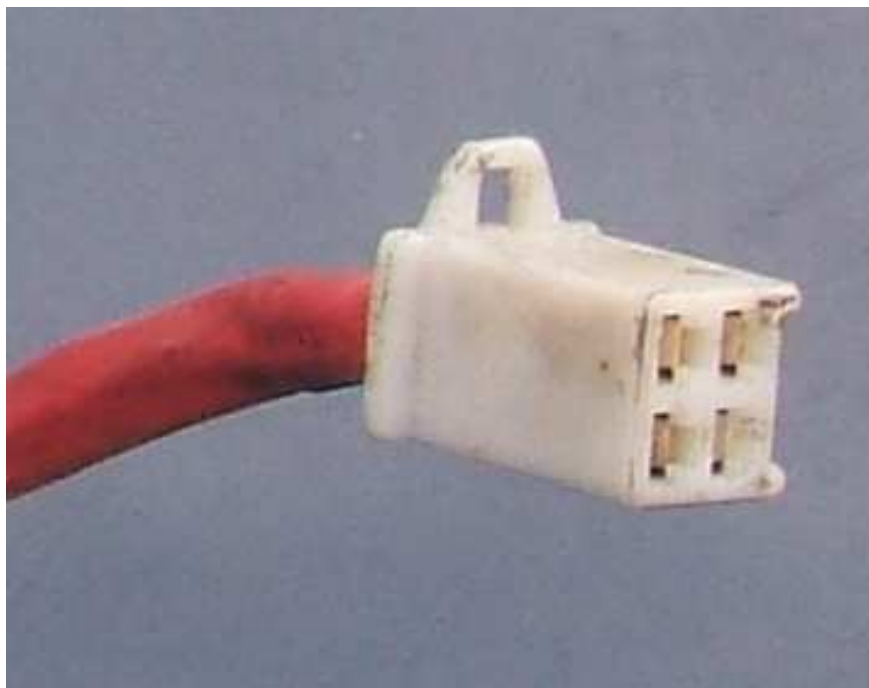


Figura 4.2 conectores de sensores y actuadores

Los inyectores vienen originalmente con sus propios conectores los cuales se indican en la figura 4.3.



Figura 4.3 conectores de inyectores

El conector de la tarjeta de control es un conector de 12 pines macho con su respectivo conector hembra en el cual están conectados todos los sensores y actuadores del sistema el cual se muestra en la figura 4.4.

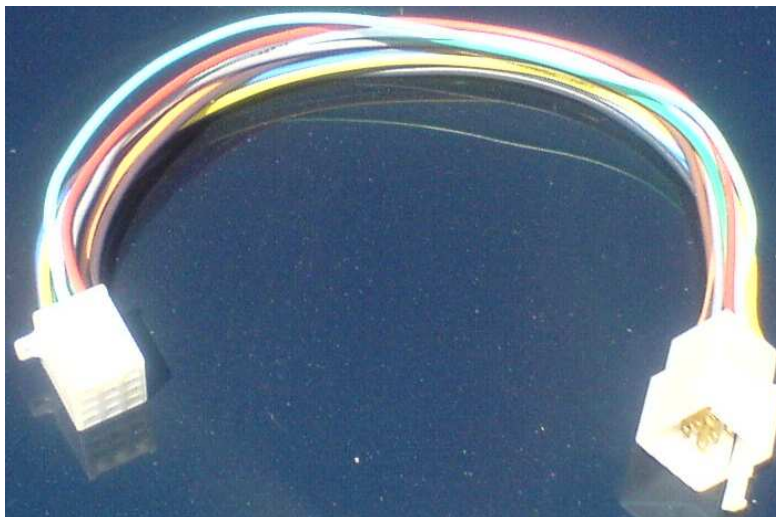


Figura 4.4 conector deL modulo de control

4.1.1 DIAGRAMA DE BLOQUES

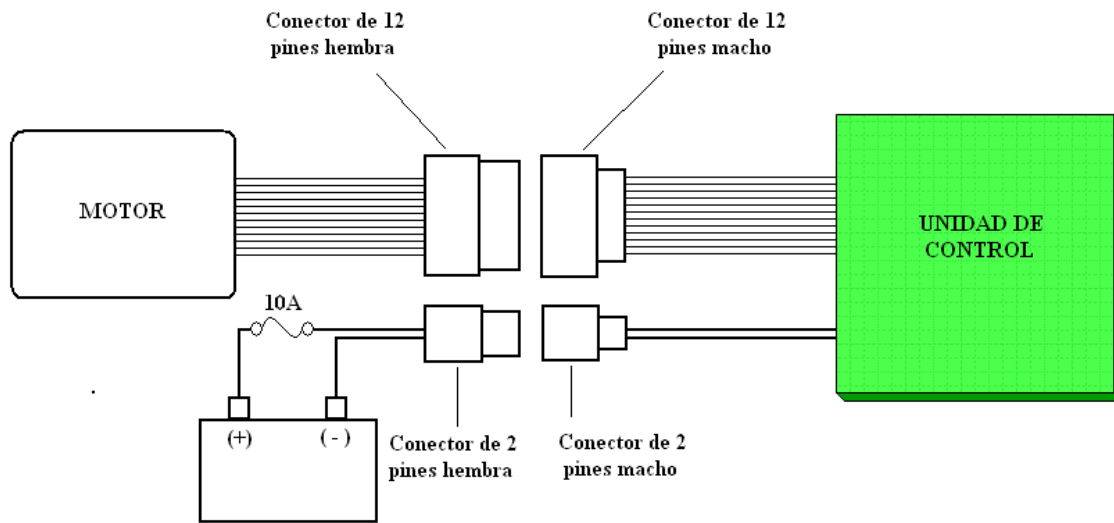


Figura 4.5 Diagrama de bloques

4.2 INSTALACIÓN DEL SISTEMA DE CONTROL ELECTRÓNICO EN EL ÁREA DISPONIBLE DEL VEHICULO

- Tarjeta de control



Figura 4.6 modulo de control

- **Sensor de temperatura del refrigerante**

Este sensor esta instalado en el gasificador ya que se aprovecho el diseño original del mismo puesto que trae su propio sensor de temperatura el mismo que sensa la temperatura del mismo circuito refrigerante del motor como se muestra en la figura 4.7.

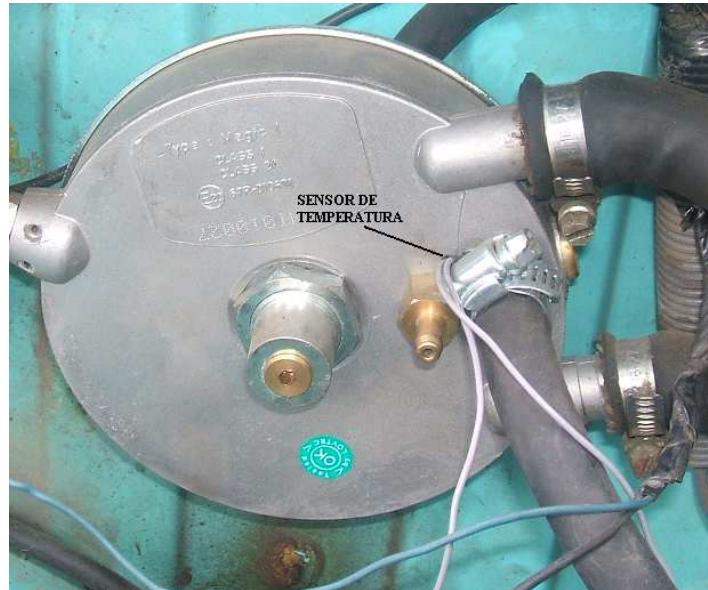


Figura 4.7 Sensor de temperatura del refrigerante

- **Sensor de posición de la mariposa**

Este sensor requiere de una adaptación especial para cada tipo de carburador , en este caso se construyo una base que sujeta el sensor y lo comunica con el eje de la mariposa que regula el flujo de aire que entra al motor como se muestra en la figura 4.8.



Figura 4.8 Sensor TPS

Como vemos en la misma base del sensor se monto el sistema de avance de vacio para lograr acelerar el motor en frio quedando ya instalado como se indica en la figura 4.9.



Figura 4.9 Sensor TPS y avance de vacio

- **Sensor de presión**

Este sensor debe estar ubicado en la parte mas alta del compartimento del motor puesto que se lo protegerá de la humedad y su manguera debe estar vertical para

prevenir acumulación de condensaciones que obstruyan el paso del vacío como mostramos en la figura 4.10[3].



Figura 4.10 Sensor MAP

- **Sensor inductivo**

Este sensor esta incorporado en el distribuidor de encendido el cual ya fue modificado anteriormente para eliminar el platino como se indica en la figura 4.11



Figura 4.11 Sensor inductivo

4.3 PRUEBAS Y AJUSTE

4.3.1 PRUEBAS

Aquí tenemos la señal del detector de cruce por cero que ingresa al pic y la salida de un inyector.

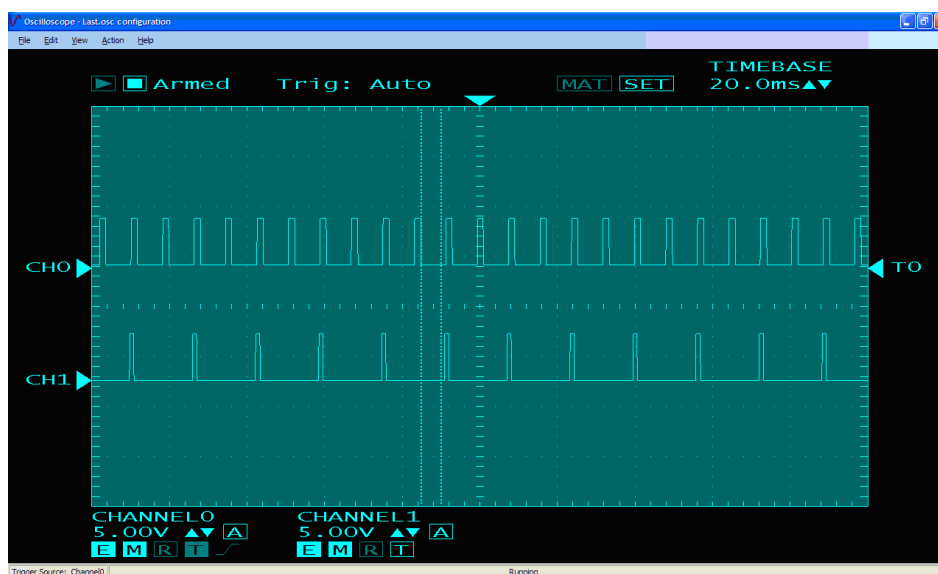


Figura 4.12 Pantalla sincronismo e inyector

Señal de los dos inyectores con ancho de pulso abierto.

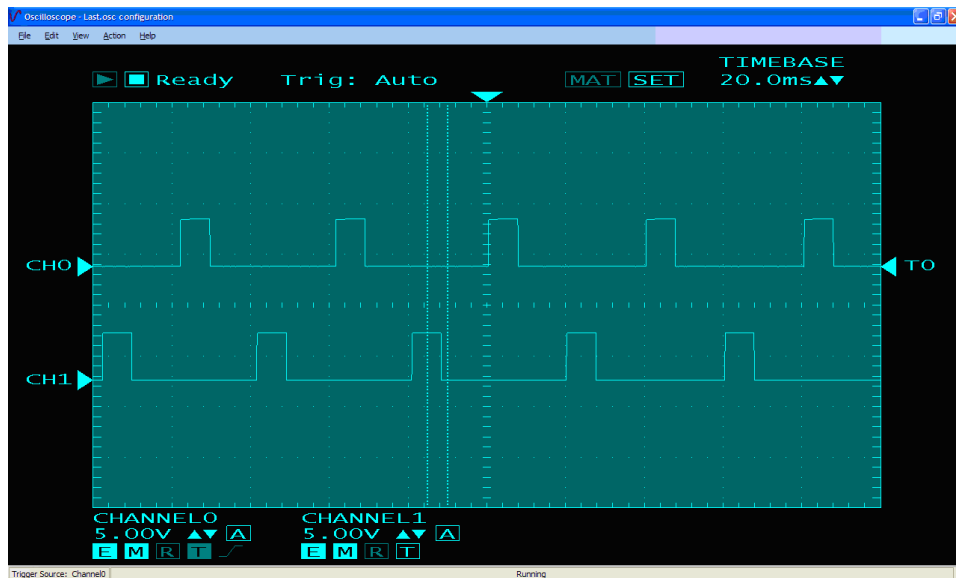


Figura 4.13 Pantalla señal de inyectores

Señal que arroja el sensor de presión MAP y sensor de revoluciones .

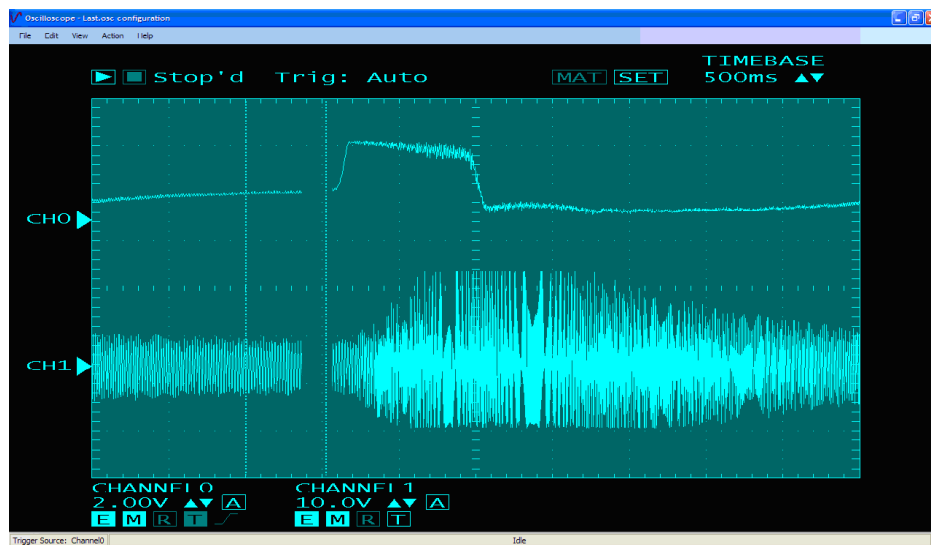


Figura 4.14 Pantalla señal MAP 1

Señal del MAP con dos intervalos de aceleración y sensor inductivo.

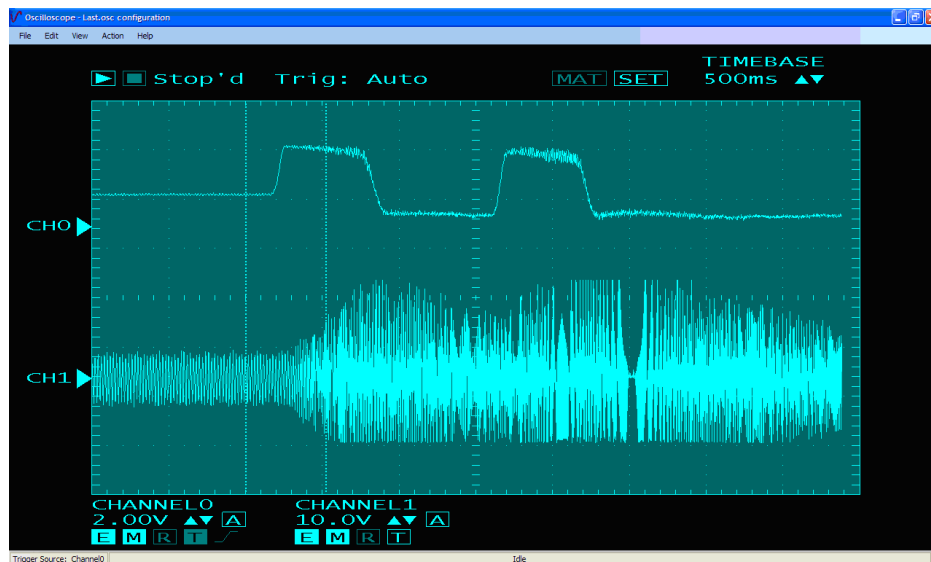


Figura 4.15 Pantalla señal MAP 2

Señal del MAP co el acelerador pisado y constante en 4000 rpm por 8 segundos y sensor inductivo.

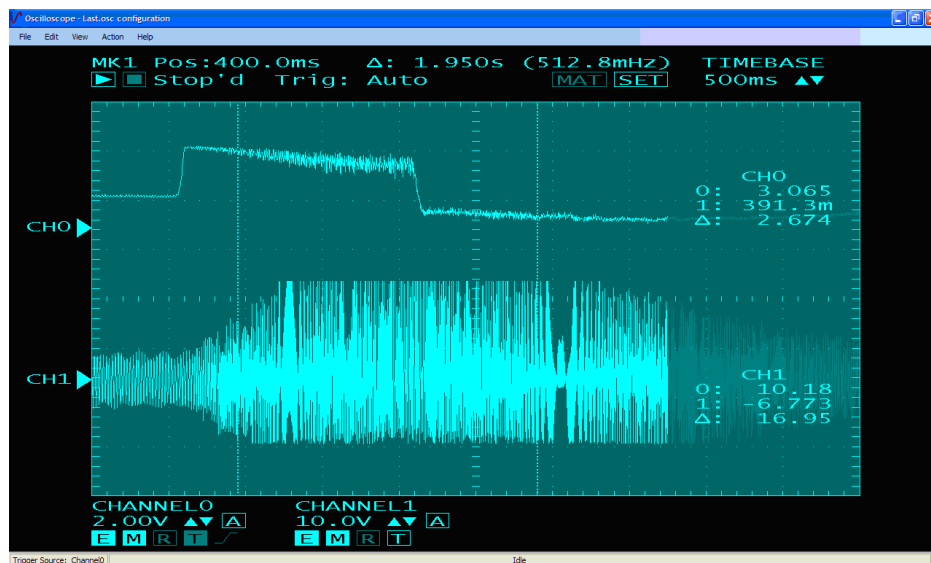


Figura 4.16 Pantalla señal MAP 3

Acelerando lento hasta 3000 rpm y mantenemos en este valor de rpm .

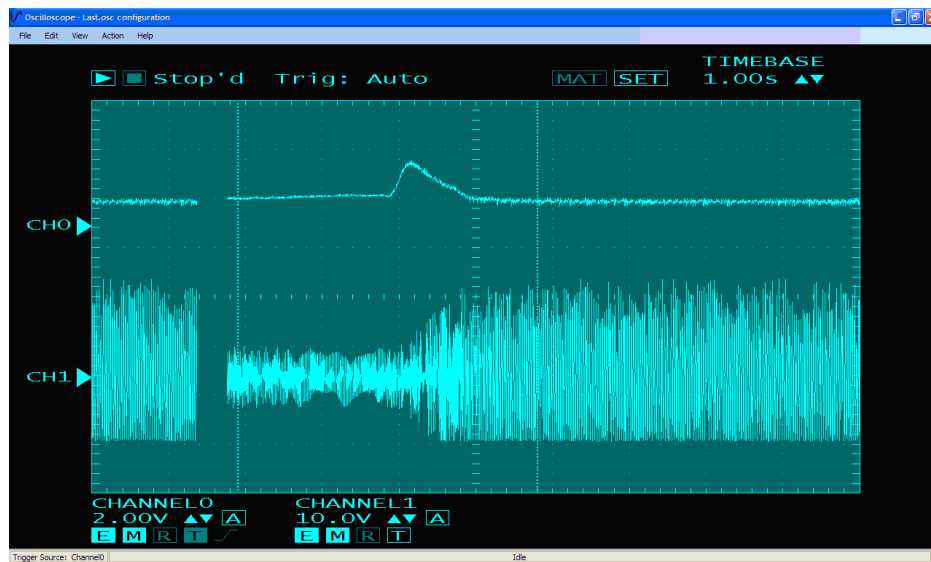


Figura 4.17 Pantalla señal MAP 4

Comenzamos en 1500 rpm constante y subimos lentamente a 4000 rpm constante.

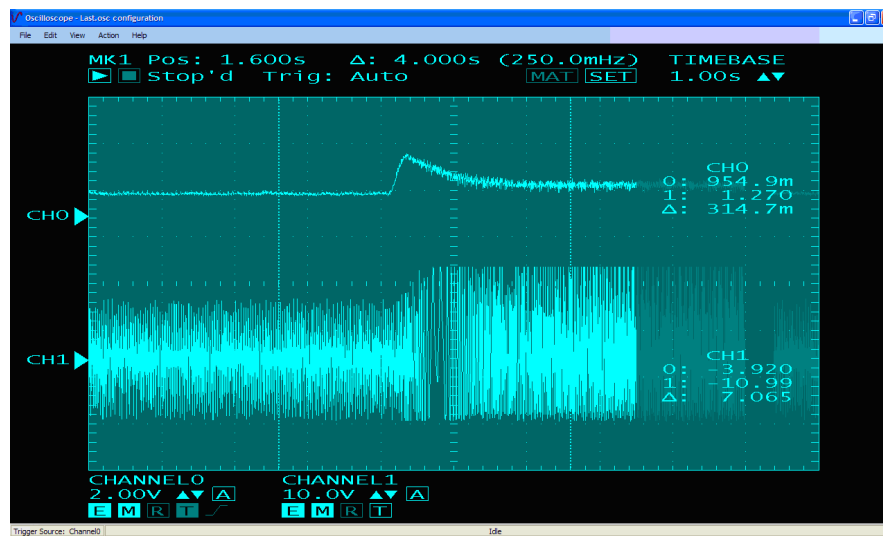


Figura 4.18 Pantalla señal MAP 5

Señal del MAP en el arranque y sensor inductivo.

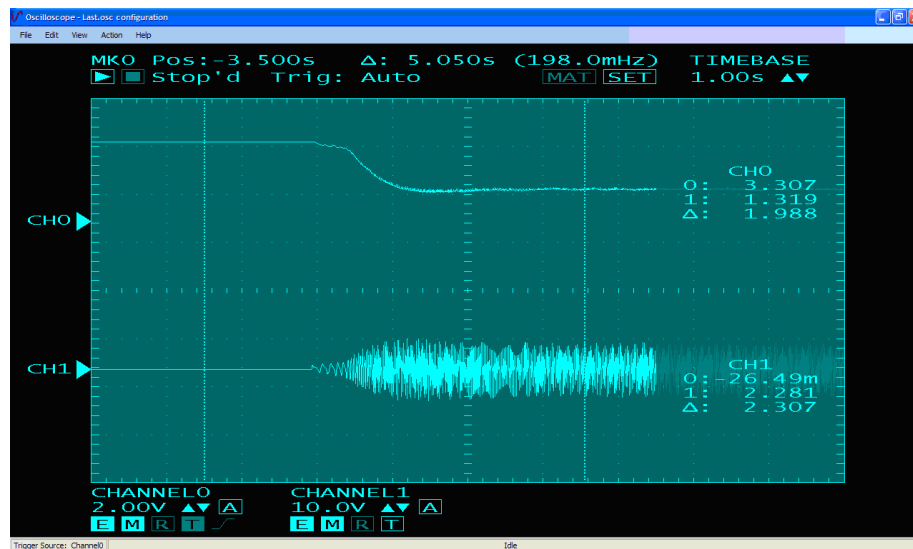


Figura 4.19 Pantalla señal MAP 6

Señales del detector de cruce por cero en el arranque Figura 4.20

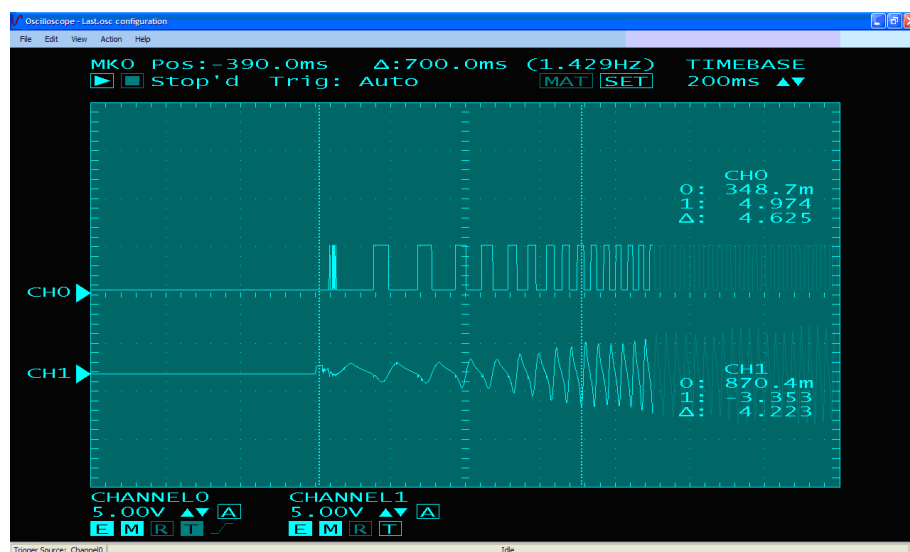


Figura 4.20 Pantalla detector de cruce por cero en arranque

4.3.2 CORRECCION DE ERRORES

Una vez concluida la instalación del sistema y arrancado el motor del vehiculo con los parámetros teóricos programados en el controlador realizamos una nueva carga de tabla de datos para compensar en algunos puntos en los que las RPM se notaba

defectuosos, se logro tener un funcionamiento uniforme del mismo y se pudo mantener encendido el motor en marcha lenta.

Principalmente las correcciones se realizaron en la tabla de datos del MAP puesto que es el sensor de mayor prioridad.

Por otra parte al activar la válvula de vacío el actuador acelero a un valor de 3000 RPM, lo cual es demasiado alto siendo el valor necesario 1000RPM, por lo que se procedio a su calibración.

Otro punto importante fue la calibración por primera vez del TPS el cual debe estar en un valor de 0,5Vdc, con el pedal suelto (0%). [3]

CAPITULO V

ANÁLISIS ECONÓMICO

Una de las principales razones para la realización de este proyecto fue el lograr revivir a los viejos motores a gasolina por medio de la utilización de el gas de uso comercial en estos motores para lo cual hacemos el siguiente análisis costo beneficio en el cual hay que tomar muy en cuenta que los precios son de uno en uno puesto que no se tiene ningún descuento al no ser una producción en serie sino solo para este prototipo. Anexo 5

Luego de esto sabemos por pruebas realizadas que con el sistema de gas el rendimiento del vehiculo es de 270 Km / 45 Kgr de gas teniendo en cuenta que el cilindro comercial de 45 Kgr cuesta (\$23,49) [51] entonces el rendimiento del vehiculo es (\$11.4 Km) por cada dólar en comparación al rendimiento cuando el motor funciona con gasolina que es de 10 Km / Galón teniendo en cuenta que el precio actual del galón de gasolina es de 1.50 dólares el rendimiento por dólar es de 6.666 Km, con este análisis se tiene en cuenta que el rendimiento es del 100%

Por otra parte teniendo en cuenta que si la producción del sistema se realiza en serie se estima que el valor comercial sería de \$1500 y que el ahorro de combustible mensual es del 50% se estimò un valor de gas al mes de \$120, lo cual quiere decir que con el ahorro logrado al cabo de un año estará pagado el sistema, en su totalidad.

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

El sistema permite la exacta dosificación del combustible por medio de la comunicación de sensores con el modulo de control.

En el caso del controlador se lo eligió por las grandes prestaciones y las posibilidades de ampliar las características del sistema.

La inclusión del firmware bootloader en el DSC simplifica y agiliza proceso de descarga de código al dispositivo, evitando el maltrato empaque debido a la extracción e inserción constante desde el zócalo.

Mientras mas grande es la cilindrada del vehiculo se requiere que el caudal que entrega el gasificador sea mayor ya sea aumentando la presión o aumentando el área de salida del gas.

Cuando el motor tiene un cubicaje alto, aumenta el número de cilindros, entonces la frecuencia en el sensor inductivo es mayor puesto que hay más inyecciones en un ciclo de funcionamiento de un motor de combustión interna.

El rendimiento del motor mejora notablemente en función al costo del combustible respecto al número de kilómetros recorridos.

Las válvulas de inyección son fabricadas para un tiempo de trabajo especificado en su ficha técnica caso contrario se averían comenzando por el recalentamiento y posteriormente a su avería total.

El uso del DSC es indispensable para alcanzar la precisión en los cálculos que se desea gracias a su velocidad de funcionamiento y la arquitectura de DSP. Este desempeño computacional no es posible con un microcontrolador de 8 bits.

El uso de la comunicación serial por RS232 resulta ser la mejor opción para conectar dos equipos debido a la difusión, disponibilidad de periféricos en los microcontroladores y drivers para el manejo de los mismos desde una PC.

Debido a que las computadoras portátiles actuales no incluyen un puerto RS232 se incluyó compatibilidad para trabajar con adaptadores USB a RS232.

Un correcto uso de planos a tierra permite disminuir notablemente el ruido en el sistema. A este criterio se suma la correcta ubicación de los elementos.

A pesar de que el compilador de ANSI C Microchip C30 es muy completo, es necesario conocer bien el lenguaje ensamblador del DSC para realizar ciertas operaciones y aún más si se desea implementar rutinas que optimicen el tiempo de ejecución.

Se estructuró de mejor manera el código y se simplificó el desarrollo del programa embebido gracias al uso del sistema operativo en tiempo real CMX-Scheduler, cuyo uso es sencillo y el esfuerzo en su aprendizaje compensa por las ventajas que trae en la programación. Hay que tener especial cuidado en la asignación de memoria a las distintas tareas.

6.2 RECOMENDACIONES

Dentro de la planeación del proyecto es imprescindible realizar la selección del controlador en función de los requerimientos planteados tomando en cuenta que pueden surgir otras necesidades imprescindibles de hardware y software.

El diseño del programa de funcionamiento se debe hacer a base de tareas paralelas y variables asociadas.

Es estrictamente necesario considerar sistemas de protección eléctrica, para garantizar el buen desempeño de los elementos que constituyen el sistema y la vida útil de los mismos.

Se recomienda dar capacitación al personal encargado del manejo del sistema, ya que conlleva una gran responsabilidad la manipulación del gas como combustible automotriz.

La instalación del presente proyecto requiere adaptaciones de mucha precisión por lo que deben ser realizadas por mano de obra calificada.

Cada vez es más difícil conseguir componentes nuevos que vengan en empaques tipo DIP. De hecho el DSC que se usó para este proyecto es el último que el fabricante lanzó con este empaque y las nuevas generaciones de estos dispositivos vienen en empaques de montaje superficial. Esto obliga a que la ESPE busque nuevas formas de realizar los prototipos a parte del clásico protoboard para que la enseñanza pueda moverse hacia los nuevos dispositivos comerciales.

Muy importante tomar las medidas de seguridad respectivas con respecto a la manipulación del gas especialmente en lo que se trata a fugas debido a la presión que se maneja por esto obligatoriamente se usa cañería de cobre.

REFERENCIAS BIBLIOGRÁFICAS

CAPITULO I

[1] FUNDAMENTOS DE ELECTRONICA. Robert L. Boylestad, Louis Nashelsky, Prentice-Hall Hispanoamericana, S.A.1997.

[2] TÉCNICAS DE G.N.C. / G.L.P. Circuitos, Puesta a Punto, Reparación y Fallas, Prentice-Hall Hispanoamericana, S.A 2002

[3]La web de los estudiantes de automoción, sensores y actuadores www.mecanicavirtual.com

CAPITULO II

[21] MICROCHIP Technology Inc. dsPIC30F3014, dsPIC30F4013 Data Sheet DS70138C, USA, 2004.

[22] MICROCHIP Technology Inc. dsPIC30F Family Reference Manual, DS70046C, USA, 2004.

[23] AMPLIFICADORES OPERACIONALES Y CIRCUITOS INTEGRADOS LINEALES. Robert F. Coughlin, Frederick F. Driscoll, Prentice-Hall Hispanoamericana, S.A 1993

[24] MOTOR DE COMBUSTIÓN INTERNA, ESTRUCTURA Y FUNCIONAMIENTO

www.wikipedia.org/w/index.php?title=Especial:Recentchanges&feed=atom.

[25] Información técnica sobre electrónica. www.electronica2000.com

CAPITULO III

[31] JONES David L., PCB Design Tutorial,

<http://alternatezone.com/electronics/files/PCBDesignTutorialRevA.pdf>, 2004.

[32] BAKER Bonnie, MICROCHIP Technology Inc., Layout Tips for 12-Bit A/D

[33] PECK Kim, MICROCHIP Technology Inc. PICmicro™ Microcontroller Oscillator Design Guide, AN588, USA, 1997. Converter Application, AN688, USA, 2004.

[34] SERVICIOS ESPECIALIZADOS GLP, Productos www.gessaonline.com

[35] Inyectores de gas, www.hlprophane.com

[37] MICROCHIP Technology Inc., dsPIC30F Data Sheet General Purpose and Sensor Families, DS70083F, USA, 2004.

[38] MICROCHIP Technology Inc., dsPIC30F3014, dsPIC30F4013 Data Sheet, DS70138, USA, 2004.

[39] <http://perso.wanadoo.es/siscobf/winpic800.htm>,

[300] Normas para el funcionamiento de motores a gas
www.enargas.gov.ar

CAPITULO IV

[41] OGATA, Katsuhiko, Ingeniería de Control Moderna, Tomo 1, 2da edición Prentice-Hall, México 1993, 1024p.

[42] SMITH, Steven W., The Scientist and Engineer's Guide to Digital Signal Processing, Tomo1, 2da edición, California Technical Publishing, San Diego 1999, 630p.

[43] OGATA, Katsuhiko, Sistemas de Control en Tiempo Discreto, Tomo 1, 2da edición, Prentice-Hall, México 1996, 745p.

CAPITULO V

[51] Direccion nacional de hidrocarburos
www.dnh.gov.ec

ANEXOS

SISTEMA DE INYECCION ELECTRONICA A GAS

ANEXO A1

MANUAL DE OPERACION DEL
SISTEMA DE INYECCION ELECTRONICA

ÍNDICE

MANUAL DE OPERACIÓN

1- INTRODUCCIÓN	2
2- COMPONENTES DEL SISTEMA DE INYECCION ELECTRONICA	2
3- CARACTERÍSTICAS DEL HARDWARE DEL SISTEMA	3
3.1 ENERGIZACIÓN DE LA PLACA	3
3.2 COMUNICACIONES	3
3.3 ENTRADAS/SALIDAS DIGITALES	3
3.4 CONTROLES E INDICADORES PARA EL USUARIO	4
3.5 ADICIONALES	4
4- TUTORIAL	5
4.1 DESCRIPCIÓN DEL HARDWARE DEL SISTEMA	5
4.2 CONEXIÓN CON EL HARDWARE	8
6.- DEFINICIÓN	23
6.1 CONEXIÓN DE LOS EQUIPOS	24
6.2 INICIO DE FUNCIONAMIENTO	29

MANUAL DE OPERACIÓN DEL SISTEMA DE INYECCION ELECTRONICA

1- INTRODUCCIÓN

Descripción General

El sistema de inyección electrónica es una solución de Hardware y Software que permite el uso de combustibles alternos, para que el parque automotor no sea dependiente de los combustibles derivados del petróleo los cuales tarde o temprano escasearán, pese a que este proyecto esta funcionando con GLP (gas licuado de petróleo) las modificaciones para GNC (gas natural comprimido) serian únicamente de software es decir solo el tiempo de inyección.

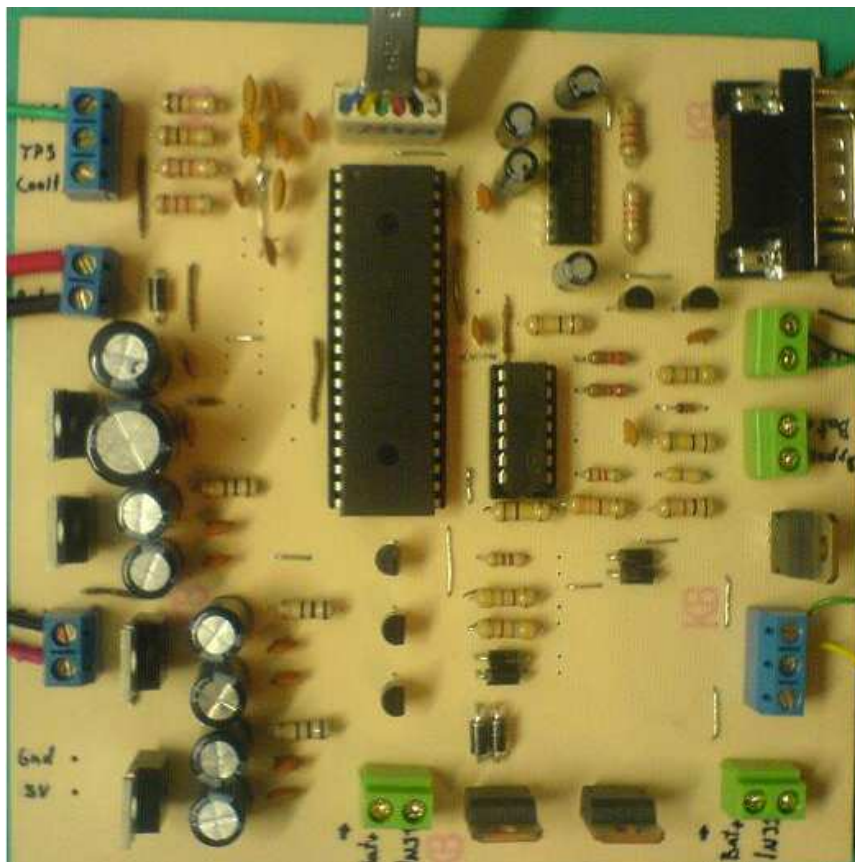


Figura. M1. Hardware del sistema y sus componentes

2- COMPONENTES DEL SISTEMA DE INYECCION ELECTRONICA

El sistema viene con los siguientes componentes:

Hardware del emulador SISO (Circuito impreso con sus componentes).

- Adaptador conector de 12VDC.
- Borneras para salida de sensores
- Cable de interfase RS-232.
- Firmware de carga al arranque (bootloader) precargado.
- Programa de carga al arranque Ingenia dsPicbootloader.

3- CARACTERÍSTICAS DEL HARDWARE DEL SISTEMA

3.1 Energización de la placa

- Reguladores incluidos para voltajes de placa de +5V y por medio de un adaptador externo de 12 Vdc accesible por un jack estándar.
- Consumo de 300mA máx.
-

3.2 Comunicaciones

- Dos puertos seriales RS-232 accesibles por un solo conector DB9 macho.
- Comunicación CAN 2.0B accesible por borneras NO HABILITADO.
-

3.3 Entradas/Salidas digitales

- Tres entradas y cuatro salidas digitales con rangos programables (por polarización externa) entre 5 y 24 V.
- Las salidas pueden absorber 200mA (colectores abiertos).

3.4 Controles e indicadores para el usuario

Interna a la cabina se tiene un selector de tres posiciones para el cambio de gas a gasolina cada una con los respectivos leds indicadores.

3.5 Adicionales

Conector para acceder a las señales analógicas y demás sensores con el fin de facilitar la calibración del hardware.

Todos los chips están colocados en zócalos para facilitar la reparación o actualización de los componentes de la placa.

4.- TUTORIAL

4.1 Descripción del Hardware del sistema

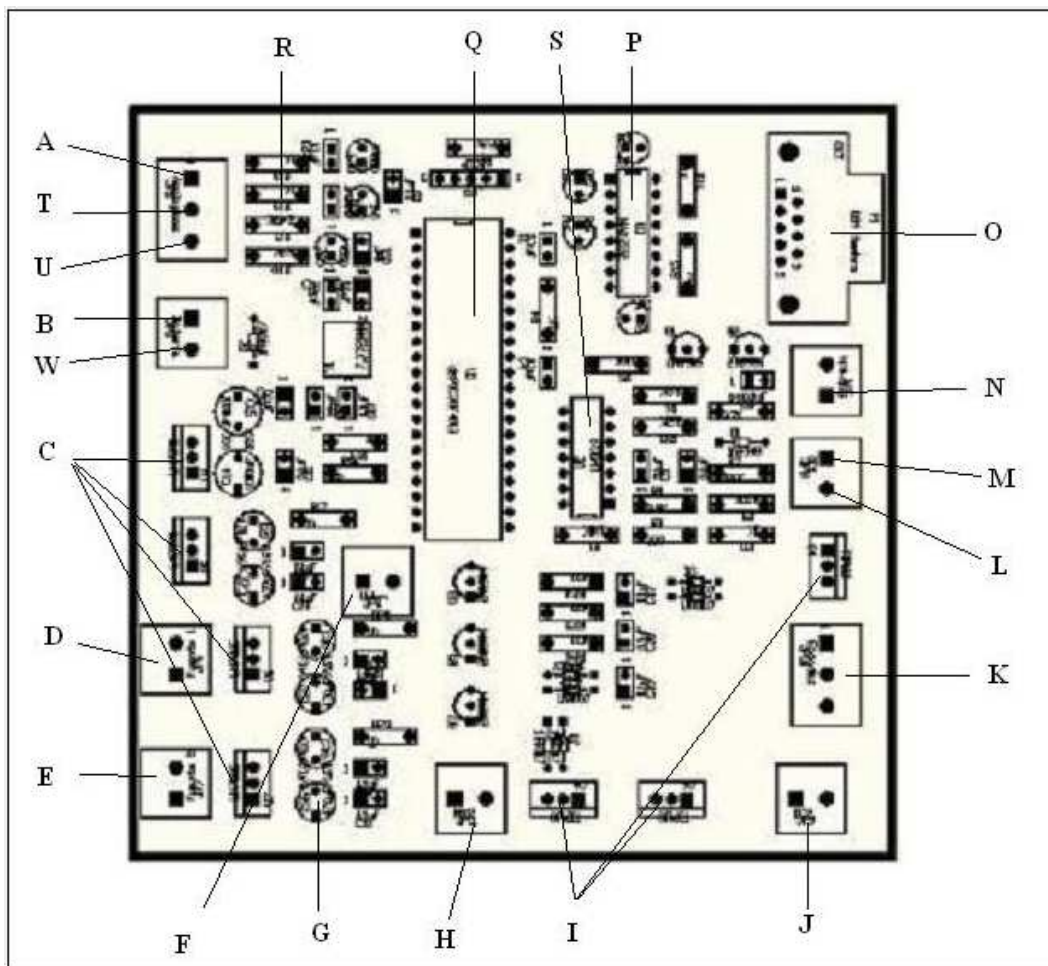


Figura M2 – Elementos del hardware del sistema

Conectores e Interfase

A.- MAP

T.- TPS

U.- SENSOR DE TEMPERATURA

W-B Entrada de voltaje de batería a la fuente de la placa ($V+ = 12$ dc).

D -E.- Reguladores de voltaje LM7805 ($V_{out} = 5$ Vdc)

D -E.-Salida de voltaje para MAP ($V+ = 5$ dc).

D -E.-Salida de voltaje para TPS ($V+ = 5$ dc).

Q.- Controlador digital de señales (Procesador del sistema).dispositivo dsPIC30F4013.

O.- CTU (Control Terminal Unit) DB9 macho

R.- Resistencias de polarización

S.- CI LM 339 amplificadores operacionales

P.- CI RS 232 Transceiver del puerto

L – J Salida de señal de inyectores

M.- Vcc de inyectores y actuador

H.- Señal de salida para salida de arranque en frío

I.-TIP 110 transistores de potencia para inyectores

K.- Salida para pruebas de señal de inyectores

N.- Señal de ingreso de tacómetro RPM

4.2 Conexión con el Hardware

- Conectar el sistema con la PC usando el cable RS-232 de interfase proporcionado.
- En *Puerto* se despliegan todos los puertos seriales disponibles para comunicarse.
- Conectar el socket de alimentación de dos pines figura M3.
- Conectar el socket de control de 12 pines figura M3.

6- DEFINICION

6.1 Conexión de los equipos

Para asegurar un correcto funcionamiento de los equipos y evitar daños es necesario verificar lo siguiente:

- La polarización de la entrada y salida analógica debe ser correcta. El sistema solo trabaja con voltaje +12V.
- Verificar que la batería de alimentación sea capaz de suministrar suficiente corriente para los dos equipos.
- Verificar las conexiones de acuerdo a la figura M4 de instalación de periféricos

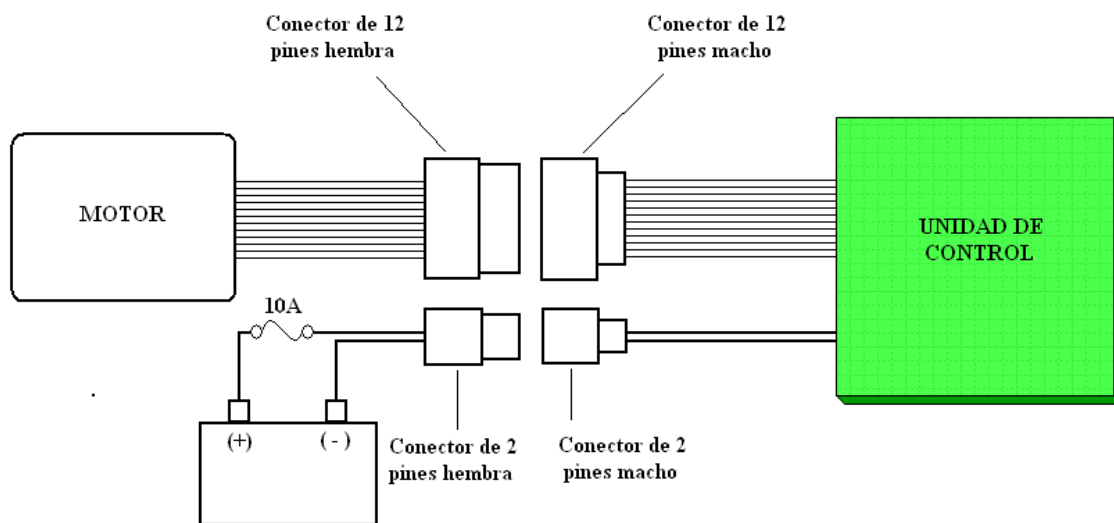


Figura M3 Conexiones del sistema

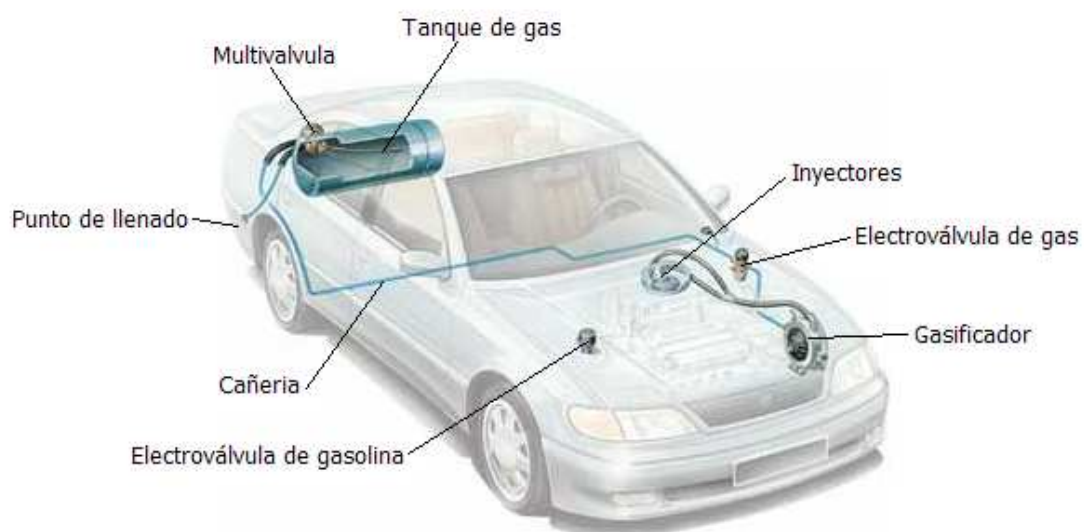


Figura M4 – Elementos principales del sistema

6.1 Inicio de funcionamiento

Uno de los aspectos principales para el funcionamiento del sistema es realizar el cambio de combustible, la cual consiste en dejar que se consuma toda la gasolina que queda en la cámara del carburador, de esta manera se evita que se mezclen los dos combustibles y produzca una contraexplosión

Por otra parte es importante que el arranque se lo realice con el combustible original en este caso gasolina, para lo cual se debe poner el swich de selección en GAS figura M5 por motivo de que la regulación de gas se realiza por medio de un medidor de presión (MAP), y en el arranque este presenta datos no válidos.

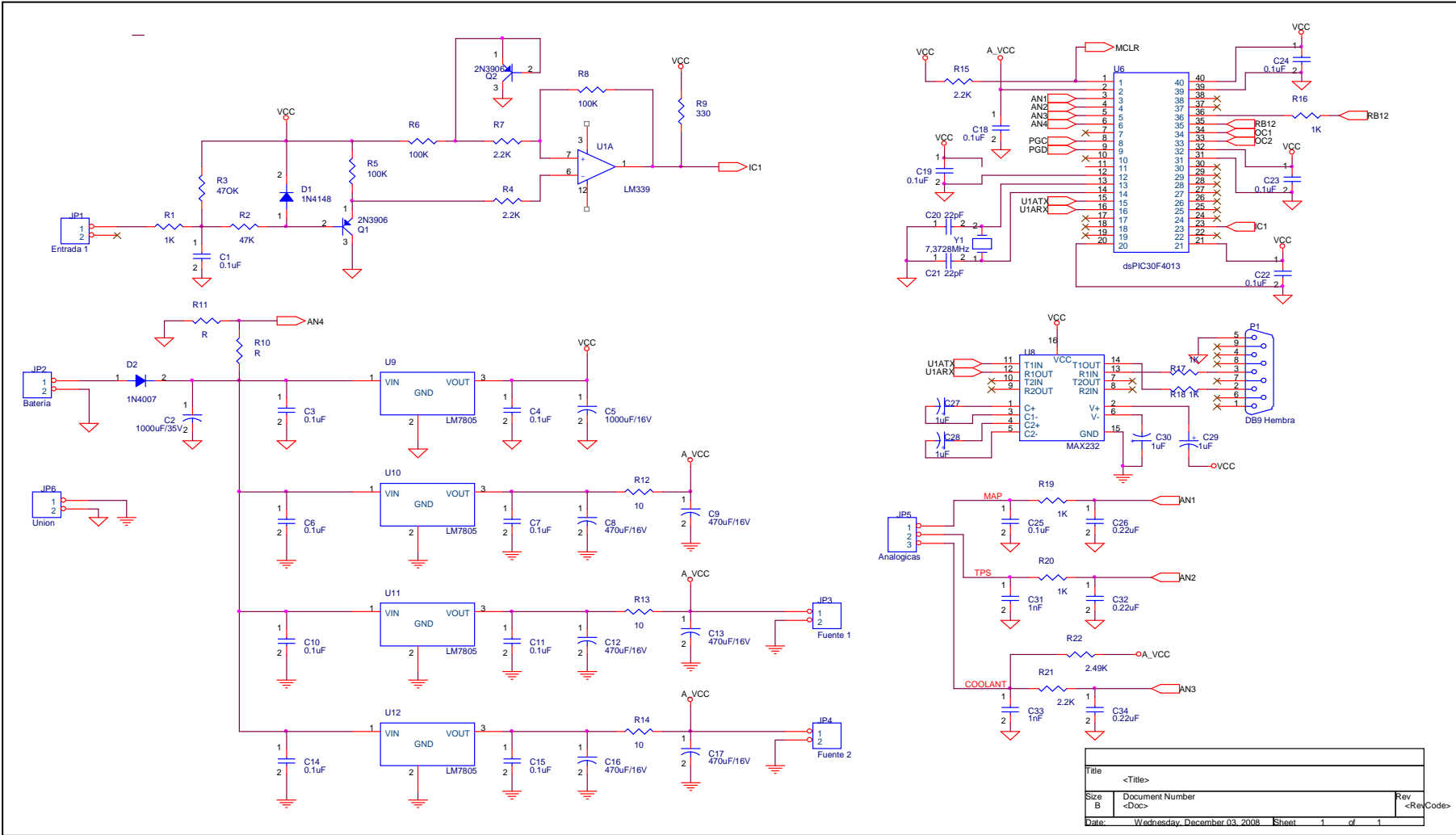
Luego de que el vehiculo esta encendido a gasolina se debe colocar el swich en la posición apagado para que se termine de consumir la gasolina que quedo en el carburador para hacer el cambio respectivo a gas y que no se mezclen los dos combustibles



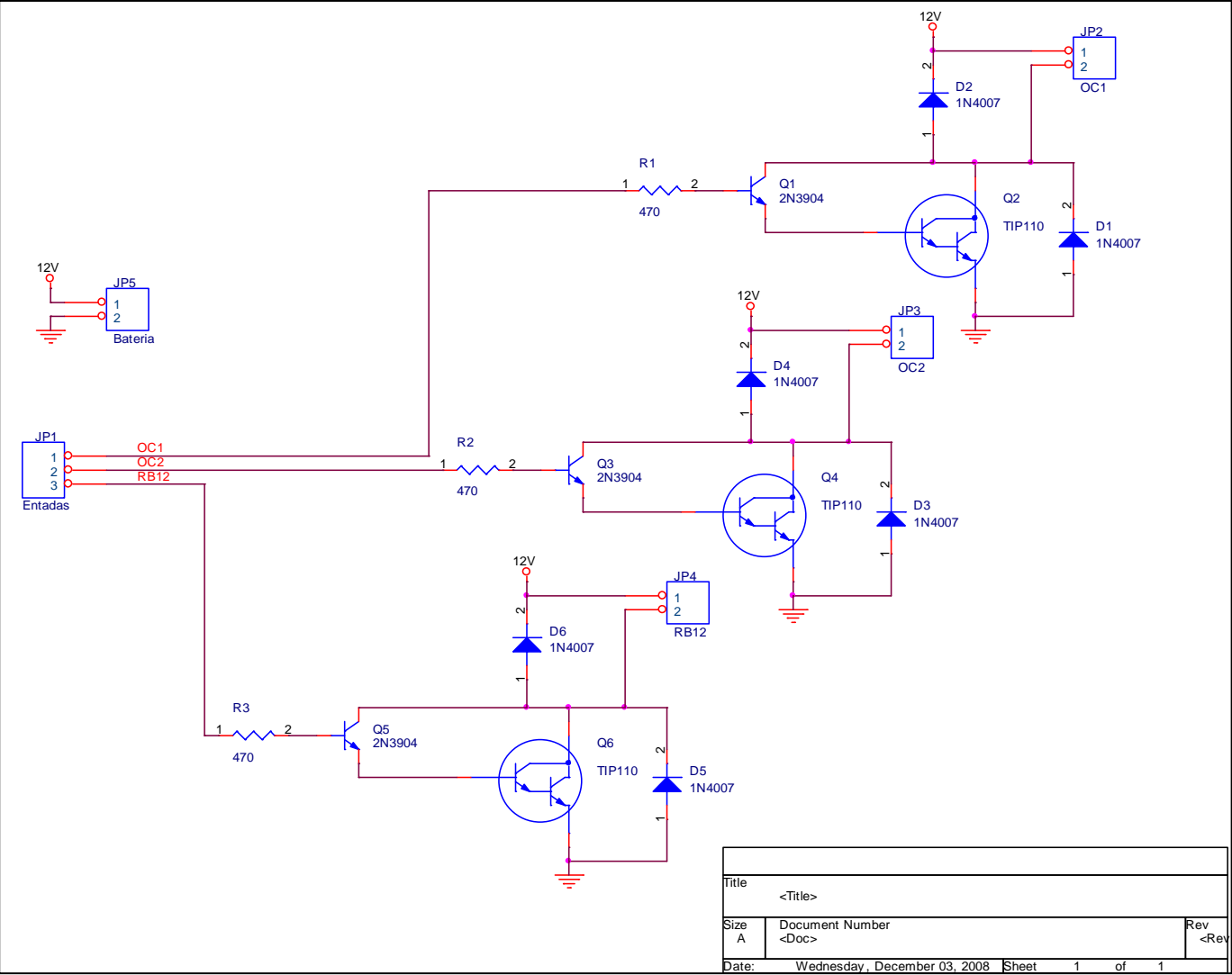
Figura M4 – Swich de selección de cambio gas gasolina

SISTEMA DE INYECCION ELECTRONICA A GAS
ANEXO A2
DIAGRAMAS DEL SISTEMA

ANEXO A2.1



ANEXO A2.2



SISTEMA DE INYECCION ELECTRONICA A GAS
ANEXO A3
DATA SHEETS

Anexo A3.1



dsPIC30F3014/4013

dsPIC30F3014/4013 High-Performance Digital Signal Controllers

Note: This data sheet summarizes features of this group of dsPIC30F devices and is not intended to be a complete reference source. For more information on the CPU, peripherals, register descriptions and general device functionality, refer to the *dsPIC30F Family Reference Manual* (DS70046). For more information on the device instruction set and programming, refer to the *dsPIC30F Programmer's Reference Manual* (DS70030).

High-Performance Modified RISC CPU:

- Modified Harvard architecture
- C compiler optimized instruction set architecture
- Flexible addressing modes
- 84 base instructions
- 24-bit wide instructions, 16-bit wide data path
- Up to 48 Kbytes on-chip Flash program space
- 2 Kbytes of on-chip data RAM
- 1 Kbyte of non-volatile data EEPROM
- 16 x 16-bit working register array
- Up to 30 MIPS operation:
 - DC to 40 MHz external clock input
 - 4 MHz-10 MHz oscillator input with PLL active (4x, 8x, 16x)
- Up to 33 interrupt sources:
 - 8 user selectable priority levels
 - 3 external interrupt sources
 - 4 processor traps

DSP Features:

- Dual data fetch
- Modulo and Bit-reversed modes
- Two 40-bit wide accumulators with optional saturation logic
- 17-bit x 17-bit single cycle hardware fractional/integer multiplier
- All DSP instructions are single cycle
 - Multiply-Accumulate (MAC) operation
- Single cycle ± 16 shift

Peripheral Features:

- High current sink/source I/O pins: 25 mA/25 mA
- Up to five 16-bit timers/counters; optionally pair up 16-bit timers into 32-bit timer modules
- Up to four 16-bit Capture input functions
- Up to four 16-bit Compare/PWM output functions
- Data Converter Interface (DCI) supports common audio Codec protocols, including I²S and AC'97
- 3-wire SPI™ module (supports 4 Frame modes)
- I²C™ module supports Multi-Master/Slave mode and 7-bit/10-bit addressing
- Up to two addressable UART modules with FIFO buffers
- CAN bus module compliant with CAN 2.0B standard

Analog Features:

- 12-bit Analog-to-Digital Converter (ADC) with:
 - 100 Ksps conversion rate
 - Up to 13 input channels
 - Conversion available during Sleep and Idle
- Programmable Low Voltage Detection (PLVD)
- Programmable Brown-out Detection and Reset generation

Special Microcontroller Features:

- Enhanced Flash program memory:
 - 10,000 erase/write cycle (min.) for industrial temperature range, 100K (typical)
- Data EEPROM memory:
 - 100,000 erase/write cycle (min.) for industrial temperature range, 1M (typical)
- Self-reprogrammable under software control
- Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Flexible Watchdog Timer (WDT) with on-chip low power RC oscillator for reliable operation
- Fail-Safe Clock Monitor operation:
 - Detects clock failure and switches to on-chip low power RC oscillator

Anexo A3.1

dsPIC30F3014/4013

Special Microcontroller Features (Cont.):

- Programmable code protection
- In-Circuit Serial Programming™ (ICSP™)
- Selectable Power Management modes:
 - Sleep, Idle and Alternate Clock modes

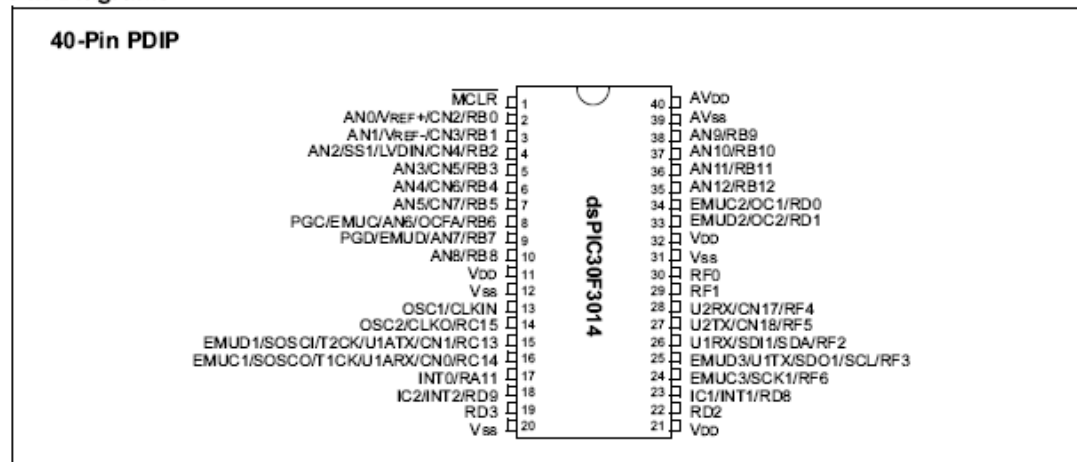
CMOS Technology:

- Low power, high speed Flash technology
- Wide operating voltage range (2.5V to 5.5V)
- Industrial and Extended temperature ranges
- Low power consumption

dsPIC30F3014/4013 Controller Family

Device	Pins	Program Memory		SRAM Bytes	EEPROM Bytes	Timer 16-bit	Input Cap	Output Comp/Std PWM	Codec Interface	A/D 12-bit 100 Ksps	UART	SPI™	I ² C™	CAN
		Bytes	Instructions											
dsPIC30F3014	40/44	24K	8K	2048	1024	3	2	2	-	13 ch	2	1	1	0
dsPIC30F4013	40/44	48K	16K	2048	1024	5	4	4	AC'97, I ² S	13 ch	2	1	1	1

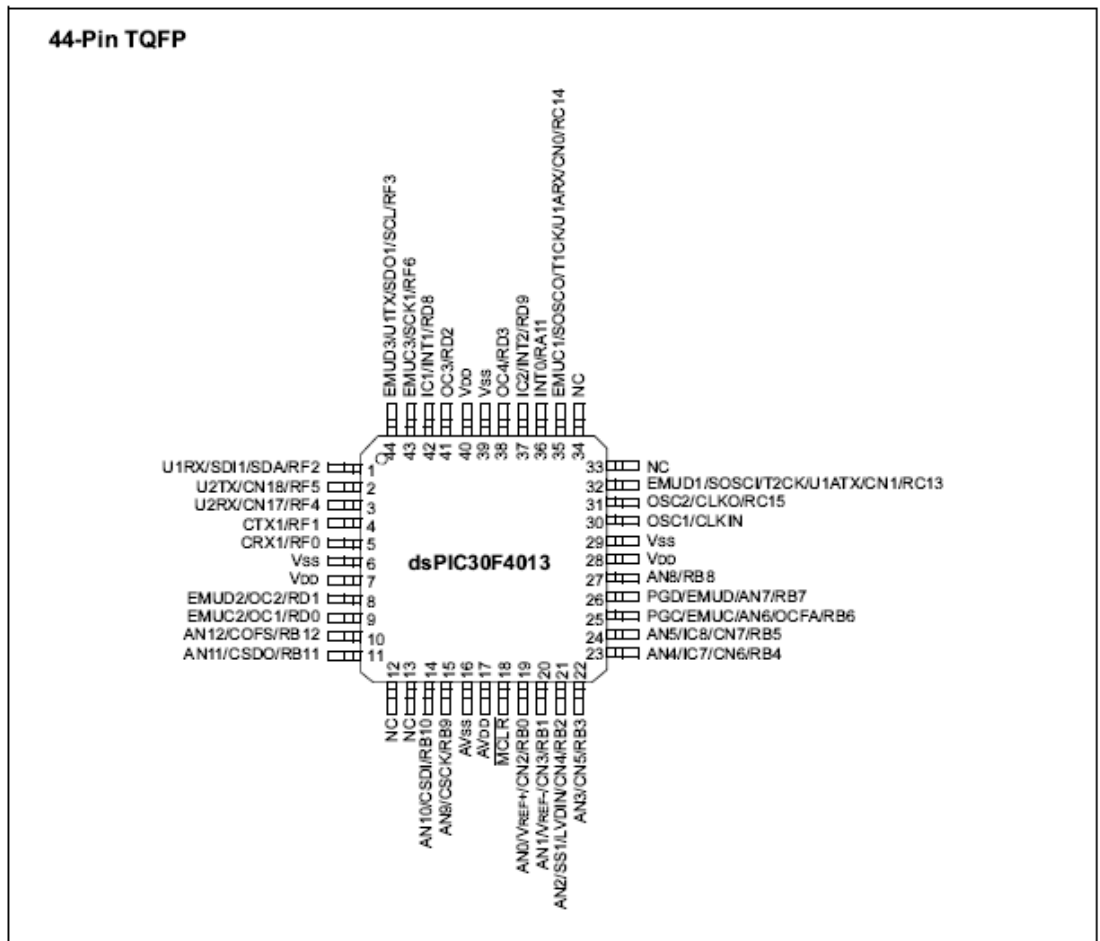
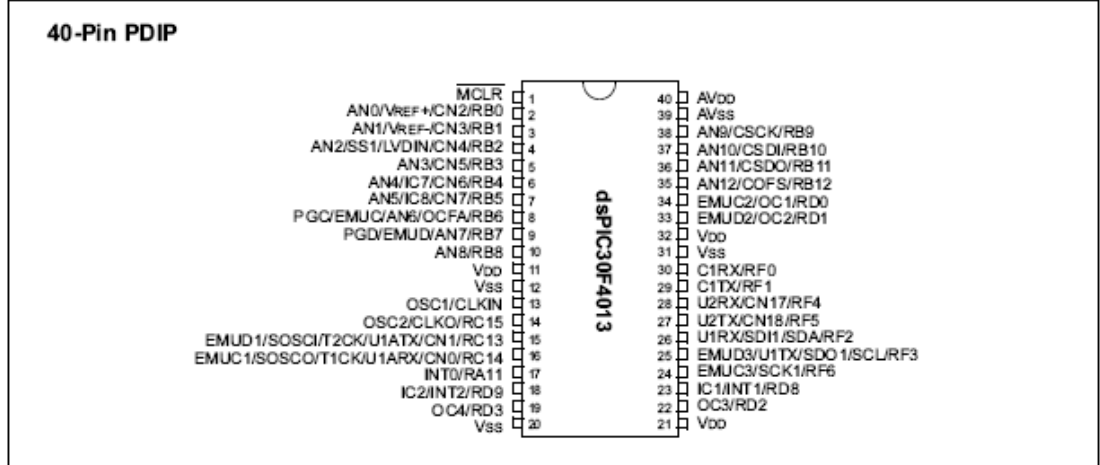
Pin Diagrams



Anexo A3.1

dsPIC30F3014/4013

Pin Diagrams (Continued)



Note: For descriptions of individual pins, see Section 1.0.

Anexo A3.2

19-4929 Rev 15, 1/05



+5V-Powered, Multichannel RS-232 Drivers/Receivers

MAX220-MAX249

General Description

The MAX220-MAX249 family of line drivers/receivers is intended for all EIA/TIA-232E and V.28/V.24 communications interfaces, particularly applications where $\pm 12V$ is not available.

These parts are especially useful in battery-powered systems, since their low-power shutdown mode reduces power dissipation to less than 5 μ W. The MAX225, MAX233, MAX235, and MAX245/MAX246/MAX247 use no external components and are recommended for applications where printed circuit board space is critical.

Applications

Portable Computers
Low-Power Modems
Interface Translation
Battery-Powered RS-232 Systems
Multidrop RS-232 Networks

AutoShutdown and UCSP are trademarks of Maxim Integrated Products, Inc.

Next-Generation Device Features

- ♦ For Low-Voltage, Integrated ESD Applications
MAX3222E/MAX3232E/MAX3237E/MAX3241E/
MAX3246E: +3.0V to +5.5V, Low-Power, Up to
1Mbps, True RS-232 Transceivers Using Four
0.1 μ F External Capacitors (MAX3246E Available
In a UCSPTM Package)
- ♦ For Low-Cost Applications
MAX221E: $\pm 15kV$ ESD-Protected, +5V, 1 μ A,
Single RS-232 Transceiver with AutoShutdown™

Ordering Information

PART	TEMP RANGE	PIN PACKAGE
MAX220CPE	0°C to +70°C	16 Plastic DIP
MAX220CSE	0°C to +70°C	16 Narrow SO
MAX220CWE	0°C to +70°C	16 Wide SO
MAX220CDE	0°C to +70°C	Dice*
MAX220EFE	-40°C to +85°C	16 Plastic DIP
MAX220ESE	-40°C to +85°C	16 Narrow SO
MAX220EWE	-40°C to +85°C	16 Wide SO
MAX220EJE	-40°C to +85°C	16 CERDIP
MAX220MJE	-55°C to +125°C	16 CERDIP

Ordering information continued at end of data sheet.
*Contact factory for dice specifications.

Selection Table

Part Number	Power Supply (V)	No. of RS-232 Drivers/Rx	No. of Ext. Caps	Nominal Cap. Value (μ F)	SHDN & Thres. State	Rx Active in SHDN	Data Rate (kbps)	Features
MAX220	+5	2/2	4	0.047/0.33	No	—	120	Ultra-low-power, industry-standard pinout
MAX222	+5	2/2	4	0.1	Yes	—	200	Low-power shutdown
MAX223 (MAX213)	+5	4/5	4	1.0 (0.1)	Yes	✓	120	MAX241 and receivers active in shutdown
MAX225	+5	5/5	0	—	Yes	✓	120	Available in SO
MAX230 (MAX200)	+5	5/0	4	1.0 (0.1)	Yes	—	120	5 drivers with shutdown
MAX231 (MAX201)	+5 and +7.5 to +13.2	2/2	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; same functions as MAX232
MAX232 (MAX202)	+5	2/2	4	1.0 (0.1)	No	—	120 (64)	Industry standard
MAX232A	+5	2/2	4	0.1	No	—	200	Higher slew rate, small caps
MAX233 (MAX203)	+5	2/2	0	—	No	—	120	No external caps
MAX233A	+5	2/2	0	—	No	—	200	No external caps, high slew rate
MAX234 (MAX204)	+5	4/0	4	1.0 (0.1)	No	—	120	Replaces 1488
MAX235 (MAX205)	+5	5/5	0	—	Yes	—	120	No external caps
MAX236 (MAX206)	+5	4/3	4	1.0 (0.1)	Yes	—	120	Shutdown, three state
MAX237 (MAX207)	+5	5/3	4	1.0 (0.1)	No	—	120	Complements IBM PC serial port
MAX238 (MAX208)	+5	4/4	4	1.0 (0.1)	No	—	120	Replaces 1488 and 1489
MAX239 (MAX209)	+5 and +7.5 to +13.2	3/5	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; single-package solution for IBM PC serial port
MAX240	+5	5/5	4	1.0	Yes	—	120	DIP or flatpack package
MAX241 (MAX211)	+5	4/5	4	1.0 (0.1)	Yes	—	120	Complete IBM PC serial port
MAX242	+5	2/2	4	0.1	Yes	✓	200	Separate shutdown and enable
MAX243	+5	2/2	4	0.1	No	—	200	Open-line detection simplifies cabling
MAX244	+5	8/10	4	1.0	No	—	120	High slew rate
MAX245	+5	8/10	0	—	Yes	✓	120	High slew rate, int. caps, two shutdown modes
MAX246	+5	8/10	0	—	Yes	✓	120	High slew rate, int. caps, three shutdown modes
MAX247	+5	8/9	0	—	Yes	✓	120	High slew rate, int. caps, nine operating modes
MAX248	+5	8/8	4	1.0	Yes	✓	120	High slew rate, selective half-chip enables
MAX249	+5	8/10	4	1.0	Yes	✓	120	Available in quad flatpack package



Maxim Integrated Products 1

For pricing, delivery, and ordering information, please contact Maxim/Dallas Direct! at 1-888-629-4642, or visit Maxim's website at www.maxim-ic.com.

Anexo A3.3



www.fairchildsemi.com

MC78XX/LM78XX/MC78XXA

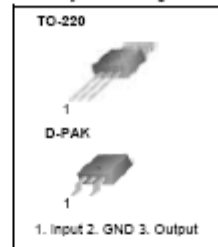
3-Terminal 1A Positive Voltage Regulator

Features

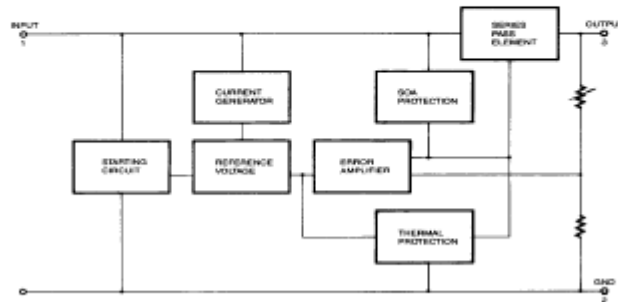
- Output Current up to 1A
- Output Voltages of 3, 6, 8, 9, 10, 12, 15, 18, 24V
- Thermal Overload Protection
- Short Circuit Protection
- Output Transistor Safe Operating Area Protection

Description

The MC78XX/LM78XX/MC78XXA series of three terminal positive regulators are available in the TO-220/D-PAK package and with several fixed output voltages, making them useful in a wide range of applications. Each type employs internal current limiting, thermal shut down and safe operating area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents.



Internal Block Diagram



Rev. 1.0.1

Anexo A3.3

MC78XXLM78XX/MC78XXA

Absolute Maximum Ratings

Parameter	Symbol	Value	Unit
Input Voltage (for $V_O = 5V$ to $18V$) (for $V_O = 24V$)	V_I V_{I1}	35 40	V
Thermal Resistance Junction-Cases (TO-220)	$R_{\theta JC}$	5	$^{\circ}C/W$
Thermal Resistance Junction-Air (TO-220)	$R_{\theta JA}$	65	$^{\circ}C/W$
Operating Temperature Range	T_{OPR}	0 ~ +125	$^{\circ}C$
Storage Temperature Range	T_{STG}	-65 ~ +150	$^{\circ}C$

Electrical Characteristics (MC7805/LM7805)

(Refer to test circuit, $0^{\circ}C < T_J < 125^{\circ}C$, $I_O = 500mA$, $V_I = 10V$, $C_I = 0.33\mu F$, $C_O = 0.1\mu F$, unless otherwise specified)

Parameter	Symbol	Conditions	MC7805/LM7805			Unit	
			Min.	Typ.	Max.		
Output Voltage	V_O	$T_J = +25^{\circ}C$	4.8	5.0	5.2	V	
		$5.0mA \leq I_O \leq 1.0A$, $P_O \leq 15W$ $V_I = 7V$ to $20V$	4.75	5.0	5.25		
Line Regulation (Note1)	Regline	$T_J = +25^{\circ}C$	$V_O = 7V$ to $25V$	-	4.0	100	mV
			$V_I = 8V$ to $12V$	-	1.6	50	
Load Regulation (Note1)	Regload	$T_J = +25^{\circ}C$	$I_O = 5.0mA$ to $1.5A$	-	9	100	mV
			$I_O = 250mA$ to $750mA$	-	4	50	
Quiescent Current	I_Q	$T_J = +25^{\circ}C$	-	5.0	8.0	mA	
Quiescent Current Change	ΔI_Q	$I_O = 5mA$ to $1.0A$	-	0.03	0.5	mA	
		$V_I = 7V$ to $25V$	-	0.3	1.3		
Output Voltage Drift	$\Delta V_O / \Delta T$	$I_O = 5mA$	-	-0.8	-	mV/ $^{\circ}C$	
Output Noise Voltage	V_N	$f = 10Hz$ to $100KHz$, $T_A = +25^{\circ}C$	-	42	-	$\mu V/V_O$	
Ripple Rejection	RR	$f = 120Hz$ $V_O = 8V$ to $18V$	62	73	-	dB	
Dropout Voltage	V_{Drop}	$I_O = 1A$, $T_J = +25^{\circ}C$	-	2	-	V	
Output Resistance	r_O	$f = 1KHz$	-	15	-	m Ω	
Short Circuit Current	I_{SC}	$V_I = 35V$, $T_A = +25^{\circ}C$	-	230	-	mA	
Peak Current	I_{PK}	$T_J = +25^{\circ}C$	-	2.2	-	A	

Note:

1. Load and line regulation are specified at constant junction temperature. Changes in V_O due to heating effects must be taken into account separately. Pulse testing with low duty is used.

Anexo A3.3

MC78XXLM78XX/MC78XXA


Electrical Characteristics (MC7806)(Refer to test circuit, $0^{\circ}\text{C} < T_J < 125^{\circ}\text{C}$, $I_O = 500\text{mA}$, $V_I = 11\text{V}$, $C_I = 0.33\mu\text{F}$, $C_O = 0.1\mu\text{F}$, unless otherwise specified)

Parameter	Symbol	Conditions	MC7806			Unit	
			Min.	Typ.	Max.		
Output Voltage	V_O	$T_J = +25^{\circ}\text{C}$	5.75	6.0	6.25	V	
		$5.0\text{mA} \leq I_O \leq 1.0\text{A}$, $P_O \leq 15\text{W}$ $V_I = 8.0\text{V to } 21\text{V}$	5.7	6.0	6.3		
Line Regulation (Note1)	Regline	$T_J = +25^{\circ}\text{C}$	$V_I = 8\text{V to } 25\text{V}$	-	5	120	mV
			$V_I = 9\text{V to } 13\text{V}$	-	1.5	60	
Load Regulation (Note1)	Regload	$T_J = +25^{\circ}\text{C}$	$I_O = 5\text{mA to } 1.5\text{A}$	-	9	120	mV
			$I_O = 250\text{mA to } 750\text{mA}$	-	3	60	
Quiescent Current	I_Q	$T_J = +25^{\circ}\text{C}$	-	5.0	8.0	mA	
Quiescent Current Change	ΔI_Q	$I_O = 5\text{mA to } 1\text{A}$ $V_I = 8\text{V to } 25\text{V}$	-	-	0.5	mA	
			-	-	1.3		
Output Voltage Drift	$\Delta V_O/\Delta T$	$I_O = 5\text{mA}$	-	-0.8	-	mV/ $^{\circ}\text{C}$	
Output Noise Voltage	V_N	$f = 10\text{Hz to } 100\text{kHz}$, $T_A = +25^{\circ}\text{C}$	-	45	-	$\mu\text{V}/V_O$	
Ripple Rejection	RR	$f = 120\text{Hz}$ $V_I = 9\text{V to } 19\text{V}$	59	75	-	dB	
Dropout Voltage	V_{Droop}	$I_O = 1\text{A}$, $T_J = +25^{\circ}\text{C}$	-	2	-	V	
Output Resistance	r_O	$f = 1\text{kHz}$	-	19	-	$\text{m}\Omega$	
Short Circuit Current	I_{SC}	$V_I = 35\text{V}$, $T_A = +25^{\circ}\text{C}$	-	250	-	mA	
Peak Current	I_{PK}	$T_J = +25^{\circ}\text{C}$	-	2.2	-	A	

Note:

1. Load and line regulation are specified at constant junction temperature. Changes in V_O due to heating effects must be taken into account separately. Pulse testing with low duty is used.

Anexo A3.4




TIP110/111/112

TIP110/111/112

Monolithic Construction With Built In Base-Emitter Shunt Resistors

- Complementary to TIP115/116/117
- High DC Current Gain : $\beta_{DC} = 1000$ @ $V_{CE} = 4V, I_C = 1A$ (Min.)
- Low Collector-Emitter Saturation Voltage
- Industrial Use



TO-220

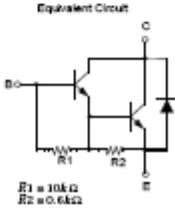
1.Base 2.Collector 3.Emmitter

NPN Epitaxial Silicon Darlington Transistor

Absolute Maximum Ratings $T_C = 25^\circ\text{C}$ unless otherwise noted

Symbol	Parameter	Value	Units
V_{CBO}	Collector-Base Voltage : TIP110	60	V
	: TIP111	80	V
	: TIP112	100	V
V_{CEO}	Collector-Emitter Voltage : TIP110	60	V
	: TIP111	80	V
	: TIP112	100	V
V_{EBO}	Emitter-Base Voltage	5	V
I_C	Collector Current (DC)	2	A
I_{CP}	Collector Current (Pulse)	4	A
I_B	Base Current (DC)	50	mA
P_C	Collector Dissipation ($T_C = 25^\circ\text{C}$)	2	W
	Collector Dissipation ($T_C = 25^\circ\text{C}$)	50	W
T_J	Junction Temperature	150	$^\circ\text{C}$
T_{STG}	Storage Temperature	- 65 ~ 150	$^\circ\text{C}$

Equivalent Circuit



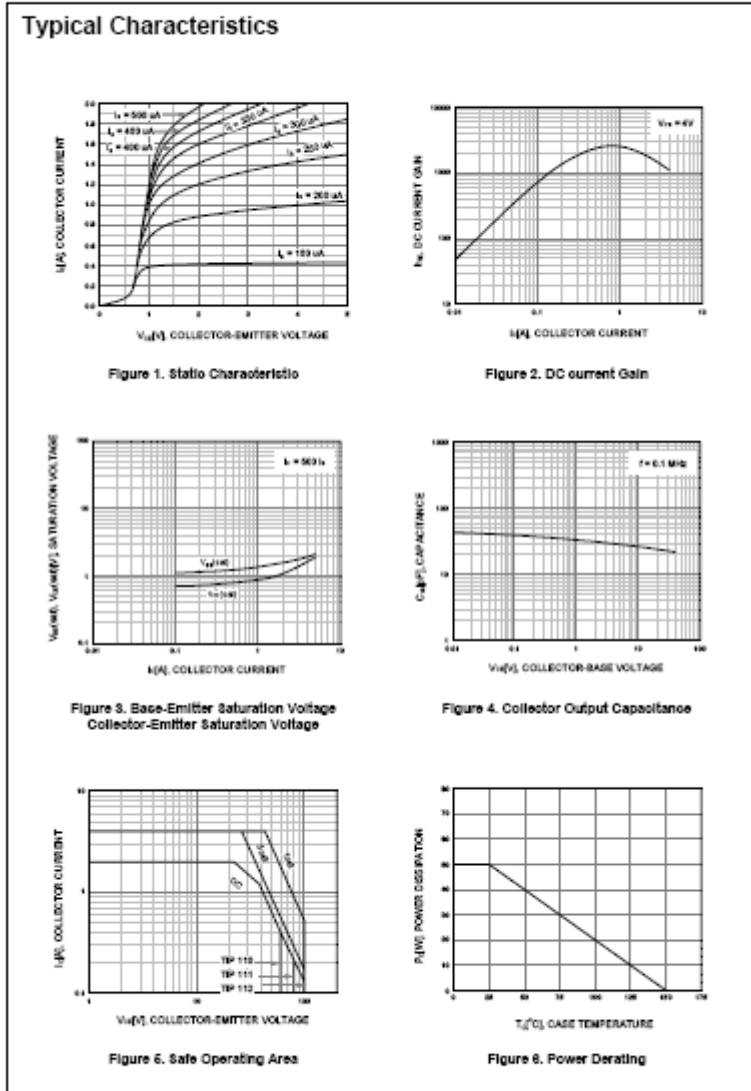
$R1 = 10k\Omega$
 $R2 = 0.6k\Omega$

Electrical Characteristics $T_C = 25^\circ\text{C}$ unless otherwise noted

Symbol	Parameter	Test Condition	Min.	Max.	Units
$V_{CE0(sus)}$	Collector-Emitter Sustaining Voltage	$I_C = 30mA, I_B = 0$: TIP110	60	V
			: TIP111	80	V
			: TIP112	100	V
I_{C0}	Collector Cut-off Current	$V_{CE} = 30V, I_B = 0$		2	mA
		$V_{CE} = 40V, I_B = 0$		2	mA
		$V_{CE} = 50V, I_B = 0$		2	mA
I_{C0}	Collector Cut-off Current	$V_{CE} = 60V, I_C = 0$: TIP110	1	mA
			: TIP111	1	mA
			: TIP112	1	mA
I_{E0}	Emitter Cut-off Current	$V_{BE} = 5V, I_C = 0$		2	mA
β_{DC}	DC Current Gain	$V_{CE} = 4V, I_C = 1A$	1000		
		$V_{CE} = 4V, I_C = 2A$	500		
$V_{CE(sat)}$	Collector-Emitter Saturation Voltage	$I_C = 2A, I_B = 8mA$		2.5	V
$V_{BE(on)}$	Base-Emitter ON Voltage	$V_{CE} = 4V, I_C = 2A$		2.8	V
C_{ob}	Output Capacitance	$V_{CE} = 10V, I_C = 0, f = 0.1MHz$		100	pF

©2011 Fairchild Semiconductor Corporation
Rev. A3, June 2011

Anexo A3.4



TIP110/111/112

Anexo A3.5

INTEGRATED CIRCUITS

DATA SHEET

**LM139/239/239A/339/339A/LM2901/MC
3302**
Quad voltage comparator

Product specification
IC11 Data Handbook

1995 Nov 27

Philips
Semiconductors



PHILIPS

Anexo A3.5

Philips Semiconductors

Product specification

Quad voltage comparator

LM139/239/239A/339/339A
/LM2901/MC3302

DESCRIPTION

The LM139 series consists of four independent precision voltage comparators, with an offset voltage specification as low as 2.0mV max for each comparator, which were designed specifically to operate from a single power supply over a wide range of voltages. Operation from split power supplies is also possible and the low power supply current drain is independent of the magnitude of the power supply voltage. These comparators also have a unique characteristic in that the input common-mode voltage range includes ground, even though they are operated from a single power supply voltage.

The LM139 series was designed to directly interface with TTL and CMOS. When operated from both plus and minus power supplies, the LM139 series will directly interface with MOS logic where their low power drain is a distinct advantage over standard comparators.

FEATURES

- Wide single supply voltage range $2.0V_{DC}$ to $35V_{DC}$ or dual supplies $\pm 1.0V_{DC}$ to $\pm 18V_{DC}$
- Very low supply current drain (0.8mA) independent of supply voltage (1.0mW/comparator at $5.0V_{DC}$)
- Low input biasing current 25nA
- Low input offset current $\pm 5nA$ and offset voltage
- Input common-mode voltage range includes ground
- Differential input voltage range equal to the power supply voltage
- Low output 250mV at 4mA saturation voltage
- Output voltage compatible with TTL, DTL, ECL, MOS and CMOS logic systems

APPLICATIONS

- A/D converters
- Wide range VCO
- MOS clock generator
- High voltage logic gate
- Multivibrators

ORDERING INFORMATION

DESCRIPTION	TEMPERATURE RANGE	ORDER CODE	DWG #
14-Pin Ceramic Dual in-Line Package (CerDip)	-55 to +125°C	LM139F	0581B
14-Pin Plastic Dual in-Line Package (DIP)	-25°C to +85°C	LM239AN	SOT27-1
14-Pin Plastic Dual in-Line Package (DIP)	-25°C to +85°C	LM239N	SOT27-1
14-Pin Plastic Small Outline (SO) Package	-25°C to +85°C	LM239D	SOT109-1
14-Pin Plastic Dual in-Line Package (DIP)	-40°C to +125°C	LM2901N	SOT27-1
14-Pin Plastic Small Outline (SO) Package	-40°C to +125°C	LM2901D	SOT109-1
14-Pin Plastic Dual in-Line Package (DIP)	0 to +70°C	LM339AN	SOT27-1
14-Pin Plastic Small Outline (SO) Package	0 to +70°C	LM339D	SOT109-1
14-Pin Plastic Dual in-Line Package (DIP)	0 to +70°C	LM339N	SOT27-1
14-Pin Plastic Small Outline (SO) Package	-40°C to +85°C	MC3302D	SOT109-1
14-Pin Ceramic Dual in-Line Package (CerDip)	-40°C to +85°C	MC3302F	0581B
14-Pin Plastic Dual in-Line Package (DIP)	-40°C to +85°C	MC3302N	SOT27-1
14-Pin Plastic Dual in-Line Package (DIP)	-55 to +125°C	LM139N	SOT27-1

PIN CONFIGURATION

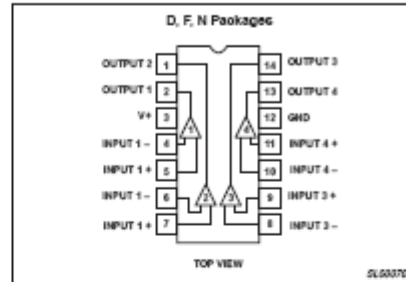


Figure 1. Pin Configuration

EQUIVALENT CIRCUIT

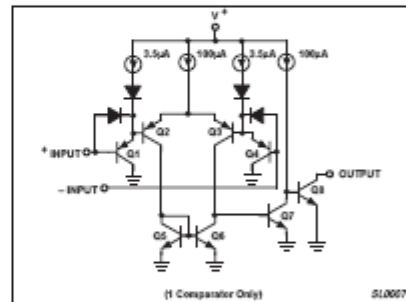




Figure 2. Equivalent Circuit

Anexo A3.6




1N/FDLL 914/A/B / 916/A/B / 4148 / 4448

1N/FDLL 914/A/B / 916/A/B / 4148 / 4448



DO-35



LL-34

THE PLACEMENT OF THE EXPANSION GAP HAS NO RELATIONSHIP TO THE LOCATION OF THE CATHODE TERMINAL.

COLOR BAND MARKING

DEVICE	1ST BAND	2ND BAND
FDLL914	BLACK	BROWN
FDLL914A	BLACK	GRAY
FDLL914B	BROWN	BLACK
FDLL916	BLACK	RED
FDLL916A	BLACK	WHITE
FDLL916B	BROWN	BROWN
FDLL4148	BLACK	BROWN
FDLL4448	BROWN	BLACK

Small Signal Diode

Absolute Maximum Ratings* T_a = 25°C unless otherwise noted

Symbol	Parameter	Value	Units
V _{RRM}	Maximum Repetitive Reverse Voltage	100	V
I _{RM}	Average Rectified Forward Current	200	mA
I _{FSM}	Non-repetitive Peak Forward Surge Current Pulse Width = 1.0 second Pulse Width = 1.0 microsecond	1.0 4.0	A A
T _{stg}	Storage Temperature Range	-65 to +200	°C
T _j	Operating Junction Temperature	175	°C

* These ratings are limiting values above which the serviceability of any semiconductor device may be impaired.

NOTES
 1) These ratings are based on a maximum junction temperature of 200 degrees C.
 2) These are steady state limits. The factory should be consulted on applications involving pulsed or low duty cycle operations.

Thermal Characteristics

Symbol	Characteristic	Max	Units
		1N/FDLL 914/A/B / 4148 / 4448	
P _D	Power Dissipation	500	mW
R _{θJA}	Thermal Resistance, Junction to Ambient	300	°C/W

Anexo A3.6

1N/FD/L 914/A/B / 916/A/B / 4148 / 4448

Small Signal Diode
(continued)

Electrical Characteristics $T_A = 25^\circ\text{C}$ unless otherwise noted

Symbol	Parameter	Test Conditions	Min	Max	Units
V_R	Breakdown Voltage	$I_R = 100 \mu\text{A}$ $I_R = 5.0 \mu\text{A}$	100 75		V V
V_F	Forward Voltage	1N914B/4448 1N916B 1N914S/16/4148 1N914A/S/16A 1N916B 1N914B/4448	$I_F = 5.0 \text{ mA}$ $I_F = 5.0 \text{ mA}$ $I_F = 10 \text{ mA}$ $I_F = 20 \text{ mA}$ $I_F = 20 \text{ mA}$ $I_F = 100 \text{ mA}$	620 730 1.0 1.0 1.0 1.0	mV mV V V V V
I_R	Reverse Current	$V_R = 20 \text{ V}$ $V_R = 20 \text{ V}, T_A = 150^\circ\text{C}$ $V_R = 75 \text{ V}$		25 50 5.0	nA μA μA
C_T	Total Capacitance	1N916A/B/4448 1N914A/B/4148	$V_R = 0, f = 1.0 \text{ MHz}$ $V_R = 0, f = 1.0 \text{ MHz}$	2.0 4.0	pF pF
t_r	Reverse Recovery Time	$I_F = 10 \text{ mA}, V_R = 6.0 \text{ V (50mA)}$, $I_R = 1.0 \text{ mA}, R_s = 100\Omega$		4.0	ns

Typical Characteristics

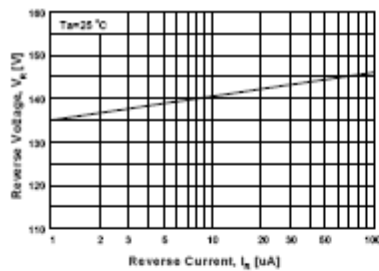


Figure 1. Reverse Voltage vs Reverse Current
BV - 1.0 to 100 μA

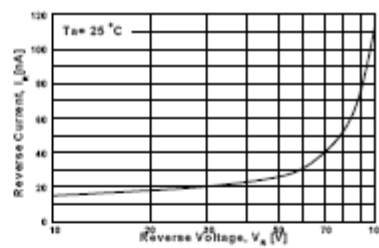


Figure 2. Reverse Current vs Reverse Voltage
IR - 10 to 100 V

GENERAL RULE: The Reverse Current of a diode will approximately double for every 10(10) Degree C increase in Temperature

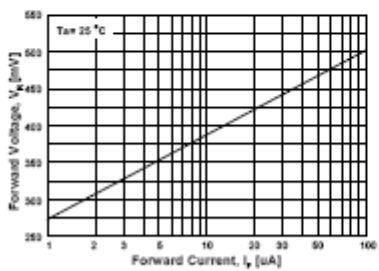


Figure 3. Forward Voltage vs Forward Current
VF - 1 to 100 μA

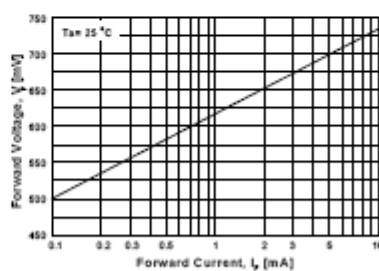


Figure 4. Forward Voltage vs Forward Current
VF - 0.1 to 10 mA

Anexo A3.7



2N3906



MMBT3906



PZT3906



2N3906 / MMBT3906 / PZT3906

PNP General Purpose Amplifier

This device is designed for general purpose amplifier and switching applications at collector currents of 10 μ A to 100 mA. Sourced from Process 66.

Absolute Maximum Ratings* TA = 25°C unless otherwise noted

Symbol	Parameter	Value	Units
V_{CE0}	Collector-Emmitter Voltage	40	V
V_{CB0}	Collector-Base Voltage	40	V
V_{EB0}	Emmitter-Base Voltage	5.0	V
I_C	Collector Current - Continuous	200	mA
T_j, T_{stg}	Operating and Storage Junction Temperature Range	-55 to +150	°C

*These ratings are limiting values above which the serviceability of any semiconductor device may be impaired.

NOTES:

- 1) These ratings are based on a maximum junction temperature of 150 degrees C.
- 2) These are steady state limits. The factory should be consulted on applications involving pulsed or low duty cycle operations.

Anexo A3.7

PNP General Purpose Amplifier
(continued)

Electrical Characteristics

TA = 25°C unless otherwise noted

Symbol	Parameter	Test Conditions	Min	Max	Units
OFF CHARACTERISTICS					
$V_{(BR)CEO}$	Collector-Emitter Breakdown Voltage*	$I_C = 1.0 \text{ mA}, I_B = 0$	40		V
$V_{(BR)CBO}$	Collector-Base Breakdown Voltage	$I_C = 10 \mu\text{A}, I_E = 0$	40		V
$V_{(BR)EBO}$	Emitter-Base Breakdown Voltage	$I_E = 10 \mu\text{A}, I_C = 0$	5.0		V
I_{B1}	Base Cutoff Current	$V_{CE} = 30 \text{ V}, V_{BE} = 3.0 \text{ V}$		50	nA
I_{C1X}	Collector Cutoff Current	$V_{CE} = 30 \text{ V}, V_{BE} = 3.0 \text{ V}$		50	nA

ON CHARACTERISTICS

β_{DC}	DC Current Gain *	$I_C = 0.1 \text{ mA}, V_{CE} = 1.0 \text{ V}$ $I_C = 1.0 \text{ mA}, V_{CE} = 1.0 \text{ V}$ $I_C = 10 \text{ mA}, V_{CE} = 1.0 \text{ V}$ $I_C = 50 \text{ mA}, V_{CE} = 1.0 \text{ V}$ $I_C = 100 \text{ mA}, V_{CE} = 1.0 \text{ V}$	60 80 100 60 30	300	
$V_{CE(sat)}$	Collector-Emitter Saturation Voltage	$I_C = 10 \text{ mA}, I_E = 1.0 \text{ mA}$ $I_C = 50 \text{ mA}, I_E = 5.0 \text{ mA}$		0.25 0.4	V
$V_{BE(sat)}$	Base-Emitter Saturation Voltage	$I_C = 10 \text{ mA}, I_E = 1.0 \text{ mA}$ $I_C = 50 \text{ mA}, I_E = 5.0 \text{ mA}$	0.65	0.85 0.95	V

SMALL SIGNAL CHARACTERISTICS

f_T	Current Gain - Bandwidth Product	$I_C = 10 \text{ mA}, V_{CE} = 20 \text{ V},$ $f = 100 \text{ MHz}$	250		MHz
C_{obe}	Output Capacitance	$V_{CE} = 5.0 \text{ V}, I_E = 0,$ $f = 100 \text{ kHz}$		4.5	pF
C_{ibe}	Input Capacitance	$V_{BE} = 0.5 \text{ V}, I_C = 0,$ $f = 100 \text{ kHz}$		10.0	pF
NF	Noise Figure (except MMPQ3905)	$I_C = 100 \mu\text{A}, V_{CE} = 5.0 \text{ V},$ $R_s = 1.0 \text{ k}\Omega, f = 10 \text{ Hz to } 15.7 \text{ kHz}$		4.0	dB

SWITCHING CHARACTERISTICS (except MMPQ3905)

t_d	Delay Time	$V_{CC} = 3.0 \text{ V}, V_{BE} = 0.5 \text{ V},$		35	ns
t_r	Rise Time	$I_C = 10 \text{ mA}, I_{B1} = 1.0 \text{ mA}$		35	ns
t_s	Storage Time	$V_{CC} = 3.0 \text{ V}, I_C = 10 \text{ mA}$		225	ns
t_f	Fall Time	$I_{B1} = I_{B2} = 1.0 \text{ mA}$		75	ns

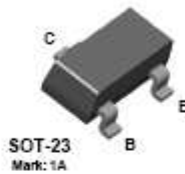
* Pulse Test: Pulse Width $\leq 300 \mu\text{s}$, Duty Cycle $\leq 2.0\%$

Spice Model

PNP (Is=1.41f Xti=3 Eg=1.11 Vaf=18.7 Bf=180.7 Ne=1.5 Ise=0 Ikf=80m Xtb=1.5 Br=4.977 Nc=2 Iso=0 Ikr=0 Rc=2.5 Cjc=9.728p Mjc=.5776 Vjc=.75 Fc=.5 Cje=8.063p Mje=.3677 Vje=.75 Tr=33.42n Tf=179.3p Itf=.4 Vtf=4 Xtf=6 Rb=10)

2N3906 / MMBT3906 / PZT3906

Anexo A3.8

**2N3904****MMBT3904****PZT3904**

2N3904 / MMBT3904 / PZT3904

NPN General Purpose Amplifier

This device is designed as a general purpose amplifier and switch. The useful dynamic range extends to 100 mA as a switch and to 100 MHz as an amplifier.

Absolute Maximum Ratings* T_A = 25°C unless otherwise noted

Symbol	Parameter	Value	Units
V _{CEO}	Collector-Emitter Voltage	40	V
V _{CBO}	Collector-Base Voltage	60	V
V _{EB0}	Emitter-Base Voltage	6.0	V
I _C	Collector Current - Continuous	200	mA
T _J , T _{stg}	Operating and Storage Junction Temperature Range	-55 to +150	°C

*These ratings are limiting values above which the serviceability of any semiconductor device may be impaired.

NOTES:

- 1) These ratings are based on a maximum junction temperature of 150 degrees C.
- 2) These are steady state limits. The factory should be consulted on applications involving pulsed or low duty cycle operations.

Thermal Characteristics T_A = 25°C unless otherwise noted

Symbol	Characteristic	Max			Units
		2N3904	**MMBT3904	**PZT3904	
P _D	Total Device Dissipation	625	350	1,000	mW
	Derate above 25°C	5.0	2.8	8.0	mW/°C
R _{θJC}	Thermal Resistance, Junction to Case	83.3			°C/W
R _{θJA}	Thermal Resistance, Junction to Ambient	200	357	125	°C/W

* Device mounted on FR-4 PCB 1.6" X 1.6" X 0.06"

** Device mounted on FR-4 PCB 38 mm X 18 mm X 1.5 mm; mounting pad for the collector lead min. 6 cm²

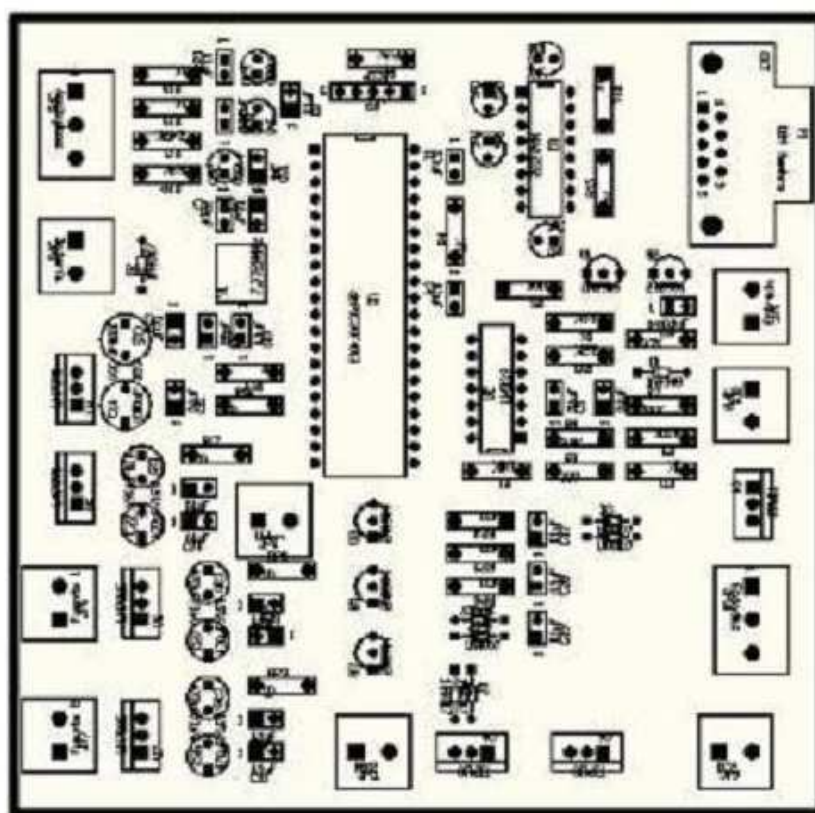
Anexo A3.8

NPN General Purpose Amplifier (continued)					
Electrical Characteristics <small>T_A = 25°C unless otherwise noted</small>					
Symbol	Parameter	Test Conditions	Min	Max	Units
OFF CHARACTERISTICS					
V _{BRCEO}	Collector-Emitter Breakdown Voltage	I _C = 1.0 mA, I _B = 0	40		V
V _{BRCSO}	Collector-Base Breakdown Voltage	I _C = 10 μA, I _E = 0	60		V
V _{BRBEO}	Emitter-Base Breakdown Voltage	I _E = 10 μA, I _C = 0	6.0		V
I _{BL}	Base Cutoff Current	V _{CE} = 30 V, V _{EB} = 3V		50	nA
I _{CIX}	Collector Cutoff Current	V _{CE} = 30 V, V _{EB} = 3V		50	nA
ON CHARACTERISTICS*					
h _{FE}	DC Current Gain	I _C = 0.1 mA, V _{CE} = 1.0 V I _C = 1.0 mA, V _{CE} = 1.0 V I _C = 10 mA, V _{CE} = 1.0 V I _C = 50 mA, V _{CE} = 1.0 V I _C = 100 mA, V _{CE} = 1.0 V	40 70 100 60 30	300	
V _{CE(sat)}	Collector-Emitter Saturation Voltage	I _C = 10 mA, I _B = 1.0 mA I _C = 50 mA, I _B = 5.0 mA		0.2 0.3	V
V _{BE(sat)}	Base-Emitter Saturation Voltage	I _C = 10 mA, I _B = 1.0 mA I _C = 50 mA, I _B = 5.0 mA	0.65	0.85 0.95	V
SMALL SIGNAL CHARACTERISTICS					
f _T	Current Gain - Bandwidth Product	I _C = 10 mA, V _{CE} = 20 V, f = 100 MHz	300		MHz
C _{obe}	Output Capacitance	V _{CE} = 5.0 V, I _E = 0, f = 1.0 MHz		4.0	pF
C _{ibe}	Input Capacitance	V _{EB} = 0.5 V, I _C = 0, f = 1.0 MHz		8.0	pF
NF	Noise Figure	I _C = 100 μA, V _{CE} = 5.0 V, R _S = 1.0kΩ, f = 10 Hz to 15.7kHz		5.0	dB
SWITCHING CHARACTERISTICS					
t _d	Delay Time	V _{CC} = 3.0 V, V _{BE} = 0.5 V,		35	ns
t _r	Rise Time	I _C = 10 mA, I _{BT} = 1.0 mA		35	ns
t _s	Storage Time	V _{CC} = 3.0 V, I _C = 10mA		200	ns
t _f	Fall Time	I _{BT} = I _{BS} = 1.0 mA		50	ns
<small>* Pulse Test: Pulse Width ≤ 300 μs, Duty Cycle ≤ 2.0%</small>					
Spice Model					
NPN (Is=6.734f Xti=3 Eg=1.11 Vaf=74.03 Bf=416.4 Ne=1.259 Ise=6.734 Ikf=66.78m Xtb=1.5 Br=.7371 No=2 Isc=0 Ikr=0 Rc=1 Cjc=3.638p Mjc=.3085 Vjc=.75 Fc=.5 Cje=4.493p Mje=.2593 Vje=.75 Tr=239.5n Tf=301.2p Ibf=4 Vtf=4 Xtf=2 Rb=10)					

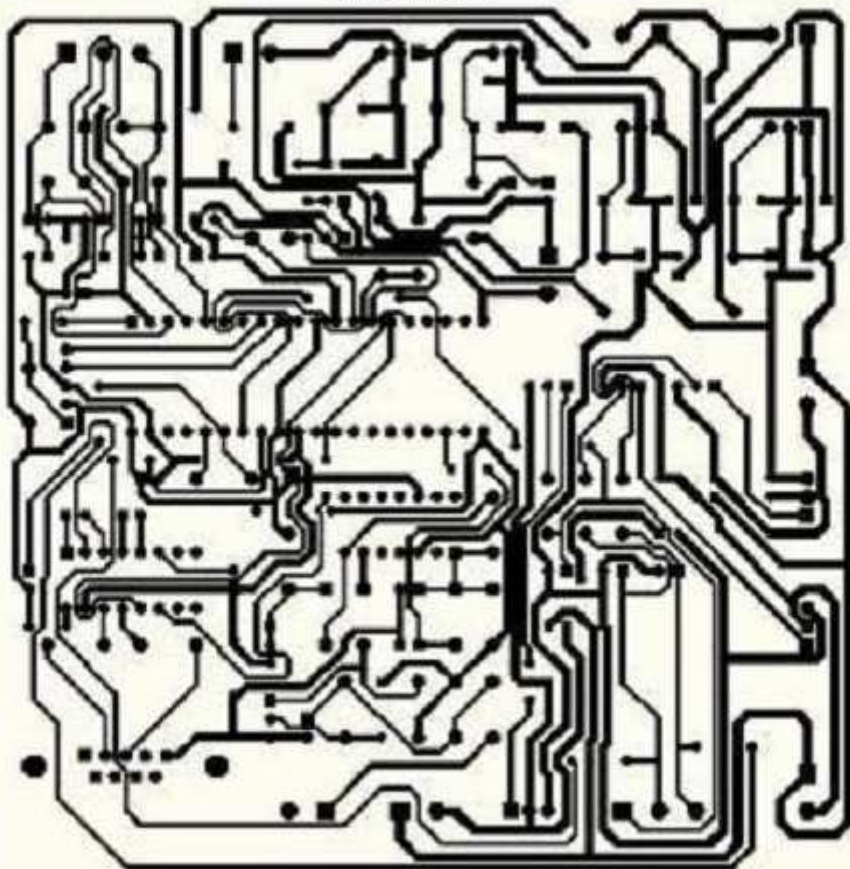
2N3904 / MMBT3904 / PZT3904

SISTEMA DE INYECCION ELECTRONICA A GAS
ANEXO A4
CIRCUITO IMPRESO

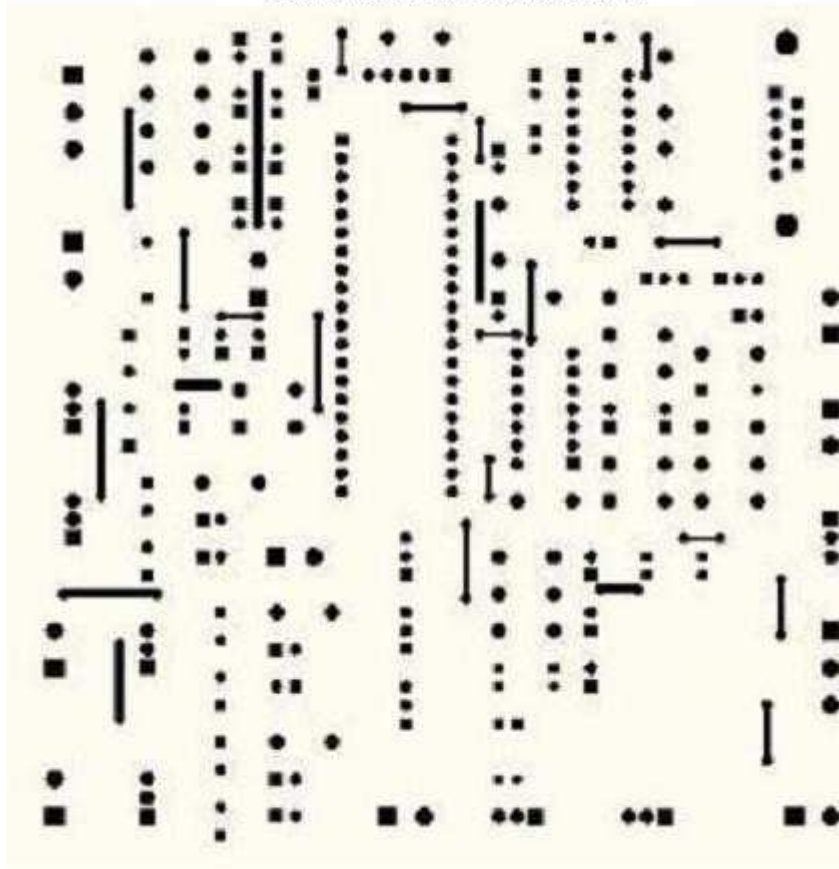
ESQUEMA PICTORICO DE LA
PLACA DE CONTROL



RUTEADO DE LA PLACA DE CONTROL



ESQUEMA DE PUENTES DE LA
TARGETA DE CONTROL

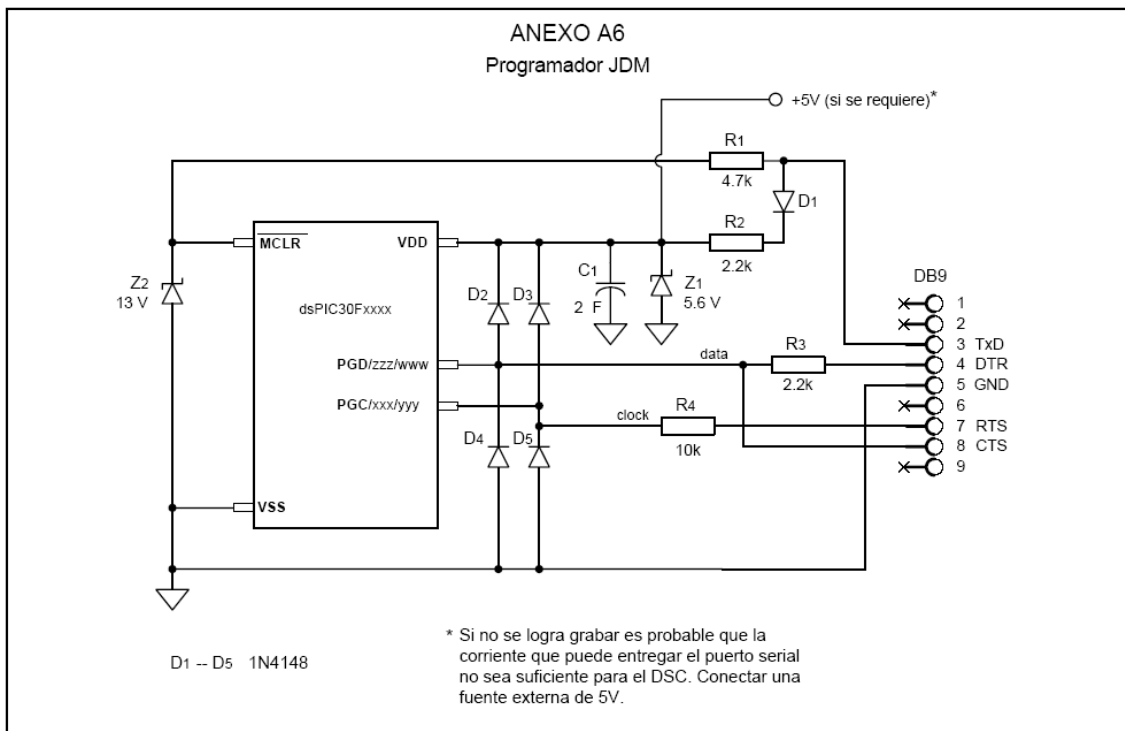


SISTEMA DE INYECCION ELECTRONICA A GAS
ANEXO A5
LISTA DE MATERIALES

ELEMENTO	CANTIDAD	VALOR	OBSERVACION
Multiválvula	1	150	
Electrofiltro de gas	1	40	
Llave conmutadora	1	25	Gasolina – gas
Regulador	1	280	Presión
Inyectores de gas	2	360	Eléctricos
Sensores	5	300	Genéricos
Electroválvulas	3	120	
Modulo Electrónico	1	950	A prueba de ruido
Relé de activación del sistema	1	15	
Cable , conectores , aislantes y otros	N/A	100	
Tanque	1	390	Norma ISO
Soporte para tanque de GNC	1	40	
Válvula manual para gas	1	35	
Manómetro	1	25	0-400 Bar.
Punto de llenado	1	60	
Tubería	N/A	40	Cobre
Mangueras	N/A	20	De presión
Adaptación de sensores	5	125	Desmontaje motor
Adaptación de partes y elementos	N/A	200	
Adaptación de la refrigeración para	1	80	Con el mismo

regulador de presión			sistema del motor
Avance de vacío	1	40	
Válvula de vacío de 3 vías dos posiciones	1	60	
TOTAL		3455	

SISTEMA DE INYECCION ELECTRONICA A GAS
ANEXO A6
PROGRAMADOR JDM



SISTEMA DE INYECCION ELECTRONICA A GAS
ANEXO A7
CODIGO DEL PROGRAMA

PROGRAMA PRINCIPAL

```
#include "p30f4013.h"
#include "drivers.h"
#include "csfunc.h"
#include "reset.h"
#include "serial.h"
```

```
//----- Hardware basic configurations -----
//----- CONFIG BITS -----
```

```
_FOSC(CSW_FSCM_OFF & FRC_PLL16); //Run this project using the
internal FRC
```

```
//routed via the PLL in 16x multiplier mode
//For the 7.3728 MHz callibrated freq. we will
```

derive a

```
//throughput of  $7.3728e+6 * 16/4 = 29.4912$ 
```

MIPS(Fcy)

```
//,~135.63 nanoseconds instruction cycle
```

time(Tcy).

```
_FWDWT(WDT_ON & WDTPSA_1 & WDTPSB_10); //Watch-Dog Timer on -
20ms period.
```

```
_FBORPOR(MCLR_EN & PWRT_OFF); //Enable MCLR reset pin and turn
off the
```

```
//power-up timers.
```

```
_FGS(CODE_PROT_OFF); //Disable Code Protection
```

```
//----- Application specific Defs -----
```

```
#define INJECTOR1 LATDbits.LATD0 //active high
```

```
#define INJECTOR2 LATDbits.LATD1 //active high
```

```
#define LED1 LATBbits.LATB9 //active low
```

```
#define TACH1 PORTDbits.RD8 //tach input
```

```
#define DEBOUNCE_TIME 5 //in RTOS ticks
```

```
#define MIN_INJECTOR_PULSE 1000
```

```
#define LOW_MODE_THRES 4500
```

```
#define HIGH_MODE_THRES 5000
```

```
#define ON_OFF_INJECTOR_RECOVER 1500
```

```
//----- Application variables -----
```

```
unsigned int global_period;
```

```
unsigned int posedge_tach_time;
```

```
unsigned int posedge_tach_time_old;
```

```
unsigned int negedge_tach_time;
```

```
unsigned int rpm_period;
unsigned int rpm;
```

```
//----- Application functions declarations -----
```

```
//float DoAverage(int analog, int samples, int period);
void InjectorDriver(unsigned int time, unsigned int inj_start_comp);
```

```
//-----CMX-RTOS declarations-----
```

```
#define TACH_EVENT 0x01 /*task 3 event*/
#define EVENT2 0x01 /* task 2 event */
```

```
#define TASK1 1      /* task 1 index position within ROM TCB */
#define TASK2 2      /* task 2 index position within ROM TCB */
#define TASK3 3      /* task 3 index position within ROM TCB */
#define TASK4 4      /* task 4 index position within ROM TCB */
#define RTC_SCALE 1
```

```
int main(void);
void task1(void); //
void task2(void); //
void task3(void); //
void task4(void); //
```

```
#define C_INTERRUPT_SIZE 64 /* Size of interrupt stack. The CMX timer
task
```

```
scheduler, and int_action
```

```
function use this. */
```

```
struct {
    word16 interrupt_bytes[C_INTERRUPT_SIZE / 2];
    } int_storage;
```

```
word16 stacks[600];
```

```
/* Note: stack grows upwards (low to high memory) */
```

```
const ROM_TCB task_rom_list[] =
    /* tsk addr, system stk, priority */
```

```
{ main,0,0,
    task1,&stacks[0],5,
    task2,&stacks[200],3,
    task3,&stacks[364],2,
    task4,&stacks[564],4 };
```

```
//----- Tasks functions -----//
```

```
//task 1 - initialization steps
void task1(void)
{
    byte status;
    int read = 200;

    unsigned int i = 0;
    unsigned int j = 0;
    unsigned int long aux = 0;

    TMR5 = 0;
    PR5 = 0;

    status = K_Task_Wait(500); //wait 50 ms

    inj_timer = 1;

    status |= K_Task_Start(TASK3);
    status |= K_Task_Start(TASK2);

    IEC0bits.T2IE = 1;
    T2CONbits.TON = 1;

    while(1)
    {

        if(FlowFlags.NEW_TACH_EDGE && FlowFlags.TACH_LEVEL)
        {
            aux = (posedge_tach_time+posedge_tach_time_old)*(unsigned
long int)(21701); //21701 original
            rpm_period = aux/10000;
            rpm = (unsigned int)((unsigned long int)(60000000)/rpm_period);

            InjectorDriver(pulse_by_serial,0); //in microseconds
            FlowFlags.NEW_TACH_EDGE = 0;
        }

        j++;

        status = K_Task_Wait(10); //wait 1ms

    }

    K_Task_End();
}

void task2(void)
```

```
{
    byte status;
    unsigned char ancii[5];
    int i = 0;

    while(1)
    {
        //send by serial data every 100ms

        PrintInt(pulse_by_serial);
        WriteTxBuff(32);
        PrintInt(rpm);
        WriteTxBuff(32);
        PrintInt(ADCRead(1));
        WriteTxBuff(0x0D);
        WriteTxBuff(0x0A);
        if(U1STAbits.OERR)
            WriteTxBuff(39);
        status = K_Task_Wait(1000); //wait 100ms
    }

    K_Task_End();
}

void task3(void)
{
    byte status;
    unsigned char data;
    int result;
    int uart_counter = 0;

    //serial communication manager task

    while(1)
    {
        while(U1STAbits.URXDA) //instead of UR1DA
        {
            //check last time a char was received
            if (uart_counter > UART_TIMEOUT)
                result = Reset_stack_pointer();
            else
                uart_counter = 0;
            //chek if a new data is in the hardware buffer and copy
            //it to the RAM buffer
            data = (unsigned char)(U1RXREG);
            //check if the new character is the end of line 0x0D char.
            if(data == 0x0A)
```

```
        {
            //if 0x0A start message analysis
            result = Process_buffer_data();
            break; //finish reading data; first process data
        }
        else
        {
            //copy data to RAM buffer
            result = Push_new_char(data);
        }
    }
    //check if there is data to be transmitted
    result = WriteHWTxBuff();

    uart_counter++;

    status = K_Task_Wait(35);
}

K_Task_End();
}

void task4(void)
{
    //byte status;

    //K_Task_End();
}

//----- Main function -----//

int main(void)
{
    byte status;
    WDTSWDisable; //disable Watchdog timer during initialization
    K_OS_Init(); /* initialize ram and things */
    status = K_Task_Create(TASK1); /* create task 1 */
    status |= K_Task_Create(TASK2); /* create task 2 */
    status |= K_Task_Create(TASK3); /* create task 3 */
    status |= K_Task_Create(TASK4); /* create task 4 */

    //status |= K_Task_Start(TASK4); /* trigger task 4 */
    status |= K_Task_Start(TASK1); /* trigger task 1 */
    //status |= K_Task_Start(TASK3); /* trigger task 3 */
    //status |= K_Task_Start(TASK2); /* trigger task 1 */

    while (status) /* test status, if non zero then sit here. */
        ;
}
```

```
INTCON2bits.ALTIVT = 0;    // use standard vector table

// initialize hardware here
iniPorts();
iniADC();
iniUart();
iniTimers();
iniInputCapture();
iniContext();

T1CONbits.TON = 1;          //start RTOS tick clock
T5CONbits.TON = 0;
IEC0bits.T1IE = 1;
IEC1bits.T5IE = 1;
ADCON1bits.ADON = 1;
IEC0bits.IC1IE = 1;
LED1=0;
U1MODEbits.UARTEN = 1;
WDTSWEnable;              //Enable watchdog timer
K_OS_Start();             /* enter CMX RTOS */
}

//-----
// ----- Application functions -----
//-----

void InjectorDriver(unsigned int time, unsigned int inj_start_comp)
{
    if(time < 17760)        //maximum injector pulse
    {
        unsigned long int longperiod = 0;
        unsigned int period = 0;
        unsigned int pulse_mode_thres = 0;

        pulse_mode_thres = rpm_period - ON_OFF_INJECTOR_RECOVER -
inj_start_comp;           // Alternate

        //timer resolution 0.271us
        //time 0-16.66259ms, 0-61425.

        period = time; //initial period value

        if(time < MIN_INJECTOR_PULSE)    //minimum injector pulse
        {
            period = MIN_INJECTOR_PULSE;
        }
    }
}
```



```
if(time >= pulse_mode_thres) period = period/2;

period = period + inj_start_comp;

if(period >= pulse_mode_thres)
    period = pulse_mode_thres;

//Match to timer time scale

longperiod = (long int)(period);
longperiod = longperiod*69;
longperiod = period*3 + longperiod/100;

if(FlowFlags.TACH_LEVEL)
{
    global_period = (unsigned int)longperiod;

    //select mode type, 1 is alternating, 0 is simultaneous
    if(time < pulse_mode_thres)
    {
        if(!FlowFlags.INJECTION_MODE)
FlowFlags.CHANGED_MODE = 1;
        FlowFlags.INJECTION_MODE = 1;
    }
    else
    {
        if(FlowFlags.INJECTION_MODE)
FlowFlags.CHANGED_MODE = 1;
        FlowFlags.INJECTION_MODE = 0;
    }
}
}

/*-----
-----INTERRUPTS-----
-----*/

//OS tick generator
void __attribute__((interrupt,auto_psv))_T1Interrupt(void)
{
    K_OS_Intrp_Entry();
    K_OS_Tick_Update();
    asm ("CLRWDT"); // clears watch dog timer
    //LED1 = TACH1;
    if(FlowFlags.DEBOUNCE) debounce_counter++;

    IFS0bits.T1IF = 0; // reset timer 1 flag
    K_OS_Intrp_Exit();
}
```

```
//Input Capture 1 interrupt
void __attribute__((interrupt,auto_psv))_IC1Interrupt(void)
{
    //K_OS_Intrp_Entry();
    IEC0bits.IC1IE = 0;

    if(!TACH1) //SEÑAL DEL COMPARADOR {
        if(debounce_counter > DEBOUNCE_TIME) // ANTIRREBOTE
        {
            FlowFlags.TACH_LEVEL = 1;
            FlowFlags.NEW_TACH_EDGE = 1;
            posedge_tach_time_old = posedge_tach_time;
            posedge_tach_time = TMR2; //CAPTURA PARA MEDIR
                                   //TIMER 2

            TMR2 = 0;
            //T2CONbits.TON = 1;
        }
        debounce_counter = 0;
        FlowFlags.DEBOUNCE = 1;
    }
    else
    {
        if(debounce_counter > DEBOUNCE_TIME && inj_timer)
        {
            negedge_tach_time = TMR2;
            TMR5 = 0;
            PR5 = global_period;
            T5CONbits.TON = 1;
            FlowFlags.TACH_LEVEL = 0;
            FlowFlags.NEW_TACH_EDGE = 1;

            if(FlowFlags.INJECTION_MODE) //mode 1 - Alternating pulses
            {
                if(FlowFlags.ACTIVE_INJ)
                {
                    INJECTOR2 = 1;
                    INJECTOR1 = 0;
                }
                else
                {
                    INJECTOR1 = 1;
                    INJECTOR2 = 0;
                }
                FlowFlags.ACTIVE_INJ ^=1; //Toggle injector
            }
            else //mode 2 - Simultaneous pulses
            {
                INJECTOR2 = 1;
                INJECTOR1 = 1;
            }
        }
    }
}
```

```
        FlowFlags.CHANGED_MODE = 0;

    }

    debounce_counter = 0;
    FlowFlags.DEBOUNCE = 1;
}

IFS0bits.IC1IF = 0; //reset input capture 1 flag
IEC0bits.IC1IE = 1;
//K_OS_Intrp_Exit();
}

void __attribute__((interrupt,auto_psv))_T2Interrupt(void)
{
    //This interrupt is executed if the tach period is over 142ms
    posedge_tach_time = 65535;
    posedge_tach_time_old = 65535;
    rpm = 0;
    IFS0bits.T2IF = 0;
}

void __attribute__((interrupt,auto_psv))_T5Interrupt(void)
{
    //K_OS_Intrp_Entry();
    //This timer closes the injectors
    INJECTOR1 = 0;
    INJECTOR2 = 0;
    IFS1bits.T5IF = 0; // reset timer 5 flag
    //K_OS_Intrp_Exit();
}

//----- Fucture Interrupts -----

/*

//UART char transmit Interrupt
void __attribute__((__interrupt__))_U1TXInterrupt(void)
{
    K_OS_Intrp_Entry();
    K_Event_Signal(0,TASK4,TXUART_EVENT);
    IFS0bits.U1TXIF = 0;
    K_OS_Intrp_Exit();
}

//UART char receive Interrupt
void __attribute__((interrupt,auto_psv))_U1RXInterrupt(void)
{
```

```
    K_OS_Intrp_Entry();
    K_Event_Signal(0,TASK3,RXUART_EVENT);
    IFS0bits.U1RXIF = 0;
    K_OS_Intrp_Exit();
}

void __attribute__((interrupt,auto_psv))_CNInterrupt(void)
{
    K_OS_Intrp_Entry();
    K_Event_Signal(0,TASK2,CN_EVENT);
    IFS0bits.CNIF = 0;
    K_OS_Intrp_Exit();
}

*/}
```

DRIVERS

```
#include "p30f4013.h"
#include "drivers.h"
#include "adc12.h"
#include "timer.h"
#include "ports.h"
#include "uart.h"
#include "incap.h"
#include "outcompare.h"

#define ERROR_INDICATOR LATBbits.LATB9

volatile fFlags FlowFlags;
volatile int inj_timer;
volatile unsigned int debounce_counter;
volatile int pulse_by_serial;

/*----- Application pin map -----

//see Other Files -> pinout_pic4013 for dsPIC30F4013 pin distribution
-----*/

//----- Definitions -----

//----- Initialization functions -----

void iniPorts(void)
{
```

```

                                                                    //1->input, 0->output
LATA = 0b1000000000000;

TRISAbits.TRISA11 = 1; //Input - Reserved

LATB = 0b11111111111111;
TRISBbits.TRISB0 = 1; //Vref+
TRISBbits.TRISB1 = 1; //AN1 Analog input
TRISBbits.TRISB2 = 1; //AN2 Analog input
TRISBbits.TRISB3 = 1; //AN3 Analog input
TRISBbits.TRISB4 = 1; //AN4 Analog input
TRISBbits.TRISB5 = 1; //AN5 Analog input
TRISBbits.TRISB6 = 1; //AN6 Analog input
TRISBbits.TRISB7 = 1; //AN7 Analog input
TRISBbits.TRISB8 = 1; //AN8 Analog input
TRISBbits.TRISB9 = 0; //Indicator - Digital output
TRISBbits.TRISB10 = 0; //Indicator - Digital output
TRISBbits.TRISB11 = 1; //Input - Reserved
TRISBbits.TRISB12 = 1; //Input - Reserved

LATC = 0b1110000000000000;
TRISCbits.TRISC13 = 1; //Used by UART1_TX, altern pins
TRISCbits.TRISC14 = 1; //Used by UART1_RX, altern pins
TRISCbits.TRISC15 = 1; //Used by oscillator

LATD = 0b1100000000;
TRISDbits.TRISD0 = 0; //Output - Injector 1
TRISDbits.TRISD1 = 0; //Output - Injector 2
TRISDbits.TRISD2 = 0; //Output - Reserved
TRISDbits.TRISD3 = 0; //Output - Reserved
TRISDbits.TRISD8 = 1; //Digital input - Tach 1
TRISDbits.TRISD9 = 1; //Digital input - Reserved

LATF = 0b1111100;
TRISFbits.TRISF0 = 1; //Digital input? - Reserved
TRISFbits.TRISF1 = 1; //Digital input? - Reserved
TRISFbits.TRISF2 = 1; //Digital input? - Reserved
TRISFbits.TRISF3 = 1; //Digital input? - Reserved
TRISFbits.TRISF4 = 1; //Digital input? - Reserved
TRISFbits.TRISF5 = 1; //Digital input? - Reserved
TRISFbits.TRISF6 = 1; //Digital input? - Reserved

//Configure CN inputs
//CNPUx Register
//This case CN5 and CN7 used -> 0b10100000
ConfigCNPullups(0b10100000);

}

void iniADC(void)
```

```

{
    unsigned int Channel, PinConfig, Scanselect, Adcon3_reg, Adcon2_reg,
    Adcon1_reg;

    ADCON1bits.ADON = 0; /* turn off ADC */
    IEC0bits.ADIE = 0; //disable interrupt

    Channel =    ADC_CH0_POS_SAMPLEA_AN1                &
    //ADC_CH0_POS_SAMPLEA_ANx
                ADC_CH0_NEG_SAMPLEA_NVREF                &
//ADC_CH0_NEG_SAMPLEA_NVREF, ADC_CH0_NEG_SAMPLEA_AN1
                ADC_CH0_POS_SAMPLEA_AN2&
//ADC_CH0_POS_SAMPLEB_ANx
                ADC_CH0_NEG_SAMPLEB_NVREF;
//ADC_CH0_NEG_SAMPLEB_NVREF, ADC_CH0_NEG_SAMPLEB_AN1
                ADC_CH0_POS_SAMPLEA_AN3&
//ADC_CH0_POS_SAMPLEB_ANx
                ADC_CH0_NEG_SAMPLEB_NVREF;
//ADC_CH0_NEG_SAMPLEB_NVREF, ADC_CH0_NEG_SAMPLEB_AN1
                ADC_CH0_POS_SAMPLEA_AN4&
//ADC_CH0_POS_SAMPLEB_ANx
                ADC_CH0_NEG_SAMPLEB_NVREF;
//ADC_CH0_NEG_SAMPLEB_NVREF, ADC_CH0_NEG_SAMPLEB_AN1
                ADC_CH0_POS_SAMPLEA_AN5&
//ADC_CH0_POS_SAMPLEB_ANx
                ADC_CH0_NEG_SAMPLEB_NVREF;
//ADC_CH0_NEG_SAMPLEB_NVREF, ADC_CH0_NEG_SAMPLEB_AN1
                ADC_CH0_POS_SAMPLEA_AN6&
//ADC_CH0_POS_SAMPLEB_ANx
                ADC_CH0_NEG_SAMPLEB_NVREF;
//ADC_CH0_NEG_SAMPLEB_NVREF, ADC_CH0_NEG_SAMPLEB_AN1
                ADC_CH0_POS_SAMPLEA_AN7&
//ADC_CH0_POS_SAMPLEB_ANx
                ADC_CH0_NEG_SAMPLEB_NVREF;
//ADC_CH0_NEG_SAMPLEB_NVREF, ADC_CH0_NEG_SAMPLEB_AN1
                ADC_CH0_POS_SAMPLEA_AN8&
//ADC_CH0_POS_SAMPLEB_ANx
                ADC_CH0_NEG_SAMPLEB_NVREF;
//ADC_CH0_NEG_SAMPLEB_NVREF, ADC_CH0_NEG_SAMPLEB_AN1

    SetChanADC12(Channel); //Set channel properties

    ConfigIntADC12(ADC_INT_DISABLE);                //ADC_INT_ENABLE,
    ADC_INT_DISABLE                                //Config ADPCFG

    PinConfig =    ENABLE_AN0_ANA    &                //ENABLE_ANx_ANA,
    ENABLE_ALL_DIG, ENABLE_ALL_ANA
                ENABLE_AN1_ANA &
                ENABLE_AN2_ANA &

```

```

ENABLE_AN3_ANA &
ENABLE_AN4_ANA &
ENABLE_AN5_ANA &
ENABLE_AN6_ANA &
ENABLE_AN7_ANA &
ENABLE_AN8_ANA;

```

```

Scanselect = SKIP_SCAN_AN0 & SKIP_SCAN_AN9 & SKIP_SCAN_AN10
& //SKIP_SCAN_ANx, SCAN_NONE, SCAN_ALL
SKIP_SCAN_AN11 & SKIP_SCAN_AN12;

```

```

Adcon3_reg = ADC_SAMPLE_TIME_1 & //ADC_SAMPLE_TIME_x (x:0-31)
ADC_CONV_CLK_SYSTEM &
//ADC_CONV_CLK_INTERNAL_RC
ADC_CONV_CLK_32Tcy; //ADC_CONV_CLK_x (x:
Tcy/2, Tcy, 3Tcy/2, ..., 32Tcy)
//Tad =
Tcy(ADCS+1)/2; Tadmin = 677ns

```

```

Adcon2_reg = ADC_VREF_EXT_AVSS & //ADC_VREF_EXT_AVSS,
ADC_VREF_AVDD_EXT, ADC_VREF_EXT_EXT
ADC_SCAN_ON & //ADC_SCAN_ON
ADC_ALT_BUF_ON & //ADC_ALT_BUF_ON,
ADC_ALT_BUF_OFF
ADC_ALT_INPUT_OFF & //ADC_ALT_INPUT_ON,
ADC_ALT_BUF_OFF
ADC_SAMPLES_PER_INT_2;
//ADC_SAMPLES_PER_INT_x (x:1-16)

```

```

Adcon1_reg = ADC_MODULE_ON & //ADC_MODULE_OFF
ADC_IDLE_STOP & //ADC_STOP
ADC_FORMAT_INTG & //ADC_FORMAT_INTG,
ADC_FORMAT_SIGN_INT, ADC_FORMAT__FRACT,
ADC_FORMAT_SIGN_FRACT
ADC_CLK_AUTO & //ADC_CLK_MANUAL,
ADC_CLK_AUTO, ADC_CLK_TMR, ADC_CLK_INT0
ADC_AUTO_SAMPLING_ON;
//ADC_AUTO_SAMPLING_ON

```

```

OpenADC12(Adcon1_reg, Adcon2_reg, Adcon3_reg, PinConfig, Scanselect);

```

```

}

```

```

void iniTimers(void)

```

```

{

```

```

    unsigned int Period;

```

```

    //Timer 1

```

```

    Period = 2948; //Ttimer = (PRx + 1)*Tcy*Preescaler

```

```

                //PRx = Ttimer/(Tcy*Preescaler)-1

```

29.4912 MIPS //->PR1 = 2948 to obtain a 100us period @

```
OpenTimer1( T1_OFF &          //_ON
            T1_IDLE_STOP &    //_CON
            T1_GATE_OFF &     //_ON
            T1_PS_1_1 & //_1_1, _1_8, _1_64, _1_128
            T1_SYNC_EXT_OFF & //_ON
            T1_SOURCE_INT  //_EXT
            , Period);
```

//Timer 2 input capture reference
 Period = 65535;
 //PR1 = 65535 to obtain a max 142.22ms period with 2.17013us res. @ 29.4912
 MIPS

```
OpenTimer2( T2_OFF &          //_ON
            T2_IDLE_STOP &    //_CON
            T2_GATE_OFF &     //_ON
            T2_PS_1_64 &      //_1_1, _1_8, _1_64, _1_128
            T2_SOURCE_INT  //_EXT
            , Period);
```

//Timer 5 valve opening timer
 Period = 65535; //->PR1 = 65535 to obtain a max 17.77ms period with
 0.271us res. @ 29.4912 MIPS

```
OpenTimer5( T5_OFF &          //_ON
            T5_IDLE_STOP &    //_CON
            T5_GATE_OFF &     //_ON
            T5_PS_1_8 & //_1_1, _1_8, _1_64, _1_128
            T5_SOURCE_INT  //_EXT
            , Period);
```

}

void iniUart(void)

```
{
  unsigned int U1MODEvalue;
  unsigned int U1STAValue;
  unsigned int baud;
```

baud = 191; //191 -> 9600bps ; 95 -> 19200bps; 47 ->
 38400bps;@29.4912MIPS

```
//U1REG = Fcy/(16*baudrate)-1
```

```
U1MODEvalue = UART_EN      &      UART_IDLE_CON      &
UART_ALTRX_ALTTX &
```

```
UART_EN_WAKE & UART_DIS_LOOPBACK
&
```



```

                                UART_DIS_ABAUD & UART_NO_PAR_8BIT
&
                                UART_1STOPBIT;
U1STAValue =    UART_INT_TX_BUF_EMPTY &
                                UART_TX_PIN_NORMAL &
                                UART_TX_ENABLE &
                                UART_INT_RX_CHAR &
                                UART_ADR_DETECT_DIS &
                                UART_RX_OVERRUN_CLEAR ;
OpenUART1(U1MODEvalue, U1STAValue, baud);

ConfigIntUART1(    UART_RX_INT_DIS &
                                UART_RX_INT_PR5 &
                                UART_TX_INT_DIS &
                                UART_TX_INT_PR3);
}

void iniInputCapture(void)
{
    unsigned int config;

    config =    IC_IDLE_STOP & //IC_IDLE_CON
                IC_TIMER2_SRC & //IC_TIMER2_SCR
                IC_INT_1CAPTURE & //IC_INT_xCAPTURE
                IC_EVERY_EDGE    ;    //IC_EVERY_RISE_EDGE,
IC_EVERY_FALL_EDGE,

                //IC_EVERY_16_RISE_EDGE, IC_EVERY_4_RISE_EDGE
                //IC_INPUTCAP_OFF

    OpenCapture1(config);

    ConfigIntCapture1(IC_INT_ON & IC_INT_PRIOR_2);
}

/*void iniOutputCompare(void)
{

}*/

void iniContext(void)
{
    //interrupt priorities
    IPC0bits.T1IP = 1;
    IPC0bits.IC1IP = 2;
    IPC5bits.T5IP = 3;
    IPC1bits.T2IP = 4;

    //ini other functions
    ini_uart_internal_var();
}
```

```
//inivariabls
FlowFlags.ACTIVE_INJ =0;
FlowFlags.DEBOUNCE = 0;
FlowFlags.INJECTION_MODE = 1;
inj_timer = 0;
debounce_counter = 0;
pulse_by_serial = 100;
}

void iniInterrupts(void)
{

}

//----- Functionals -----//

unsigned int ADCRead(unsigned int channel)
{
    unsigned int val;

    if(ADCON2bits.BUFS)
    {
        val = ReadADC12(channel - 1);
    }
    else
    {
        val = ReadADC12(channel + 7);
    }

    return val;
}

//-----ERROR TRAPS-----
/*
Primary Exception Vector handlers:
These routines are used if INTCON2bits.ALTIPT = 0.
All trap service routines in this file simply ensure that device
continuously executes code within the trap service routine. Users
may modify the basic framework provided here to suit to the needs
of their application.
*/
void __attribute__((interrupt,auto_psv)) _OscillatorFail(void)
{
    INTCON1bits.OSCFAIL = 0;    //Clear the trap flag
    ERROR_INDICATOR = 1;
    while (1);
}

void __attribute__((interrupt,auto_psv)) _AddressError(void)
```

```
{
    INTCON1bits.ADDRERR = 0;    //Clear the trap flag
    ERROR_INDICATOR = 1;
    while (1);
}
void __attribute__((interrupt,auto_psv)) _StackError(void)
{
    INTCON1bits.STKERR = 0;    //Clear the trap flag
    ERROR_INDICATOR = 1;
    while (1);
}

void __attribute__((interrupt,auto_psv)) _MathError(void)
{
    INTCON1bits.MATHERR = 0;    //Clear the trap flag
    ERROR_INDICATOR = 1;
    while (1);
}

/*
Alternate Exception Vector handlers:
These routines are used if INTCON2bits.ALTIVT = 1.
All trap service routines in this file simply ensure that device
continuously executes code within the trap service routine. Users
may modify the basic framework provided here to suit to the needs
of their application.
*/

void __attribute__((interrupt,auto_psv)) _AltOscillatorFail(void)
{
    INTCON1bits.OSCFAIL = 0;
    ERROR_INDICATOR = 1;
    while (1);
}

void __attribute__((interrupt,auto_psv)) _AltAddressError(void)
{
    INTCON1bits.ADDRERR = 0;
    ERROR_INDICATOR = 1;
    while (1);
}

void __attribute__((interrupt,auto_psv)) _AltStackError(void)
{
    INTCON1bits.STKERR = 0;
    ERROR_INDICATOR = 1;
    while (1);
}

void __attribute__((interrupt,auto_psv)) _AltMathError(void)
```

```
{
    INTCON1bits.MATHERR = 0;
    ERROR_INDICATOR = 1;
    while (1);
}
```

SERIAL

```
#include "p30f4013.h"
#include "serial.h"
#include "drivers.h"
```

```
int uartTimeout=0;
```

```
unsigned char pCount=0;
unsigned char pLeng=0;
unsigned char aux=0;
unsigned char aux2=0;
```

```
unsigned char RxBuff[UARTBUFFLENGTH]; //Receive buffer
unsigned char TxBuff[UARTBUFFLENGTH]; //Transmit buffer
```

```
unsigned char EEBuff[32]; //EEPROM Buffer
unsigned int EEBuffindex; //EEPROM Buffer index
```

```
unsigned int msg_type = 0;
unsigned int msg_state = 0;
```

```
//transmit variables
```

```
unsigned char cont = 0;
unsigned char pND = 0; //pointer to the end of the FIFO buffer
unsigned char pTD = 0; //pointer to the beginning of the FIFO buffer
```

```
//receive variables
```

```
unsigned char pRxStack = 0;
```

```
void ini_uart_internal_var(void)
```

```
{
    pCount=0;
    pLeng=0;
    aux=0;
    aux2=0;
    msg_type = 0;
    msg_state = 0;
}
```

```
int Push_new_char(unsigned data)
```

```
{
    if(pRxStack < UARTBUFFLENGTH)
    {
```

```
        RxBuff[pRxStack] = data;
        pRxStack = pRxStack + 1;
        return(1);
    }
    else
        return(0); //0 if overflow
}

int Reset_stack_pointer(void)
{
    pRxStack = 0;
}

int Process_buffer_data(void)
{
    int result = 0;
    int number = 0;
    //check command
    if(pRxStack!=0)
    {
        //Check command #MAP
        if(RxBuff[0]==0x23 && RxBuff[1]==0x4D && RxBuff[2]==0x41 &&
RxBuff[3]==0x50)
        {
            msg_type = 1;
            msg_state = 1;
            result = 10;    //stop injection process
            EEBuffindex = 0;
        }
        //Check command #END
        if(RxBuff[0]==0x23 && RxBuff[1]==0x45 && RxBuff[2]==0x4E &&
RxBuff[3]==0x44)
        {
            msg_type = 0;
            msg_state = 0;
            result = 15;    //start injection process
        }
        //Transform 3 digits ASCII number to binary form
        if (msg_state == 1 && result == 0)
        {
            if(pRxStack == 2){number = RxBuff[0]-0x30;}
            if(pRxStack == 3){number = 10*(RxBuff[0]-0x30)+(RxBuff[1]-
0x30);}
            if(pRxStack == 4){number = 100*(RxBuff[0]-
0x30)+10*(RxBuff[1]-0x30)+(RxBuff[2]-0x30);}

            EEBuff[EEBuffindex] = number;
            EEBuffindex ++;
        }
        pRxStack = 0;
    }
}
```

```
    }
    return (result);
}

void WriteTxBuff(unsigned char data)
{
    if(cont < UARTBUFFLENGTH)
    {
        TxBuff[pND] = data;
        cont++;
        if(pND < UARTBUFFLENGTH)
            {pND++;}
        else
            {pND = 0;}
    }
}

int WriteHWTxBuff(void)
{
    while(!U1STAbits.UTXBF && cont!=0 )
    {
        U1TXREG = TxBuff[pTD];
        cont = cont-1;
        if(pTD < UARTBUFFLENGTH)
        {
            pTD++;
            /*if(pTD == pND)
            {
                pTD = 0;
                pND = 0;
            }*/
            return(1);
        }
        else
        {
            pTD = 0;
            return(0);
        }
    }
}
```

```
void PrintInt(int num)
{
    int dat;
    int zero = 0;

    if(num >= 10000)
```

```
{
    dat = (unsigned int) (num/10000);
    WriteTxBuff(dat+48);
    num = num - dat*10000;
    zero = 4;
}
if(num >= 1000)
{
    dat = (unsigned int) (num/1000);
    WriteTxBuff(dat+48);
    num = num - dat*1000;
    zero = 3;
}
if(num >= 100)
{
    if(zero == 4)
        WriteTxBuff(48);
    dat = (unsigned int) (num/100);
    WriteTxBuff(dat+48);
    num = num - dat*100;
    zero = 2;
}
if(num >= 10)
{
    if(zero == 4){
        WriteTxBuff(48);
        WriteTxBuff(48);}
    if(zero == 3)
        WriteTxBuff(48);
    dat = (unsigned int) (num/10);
    WriteTxBuff(dat + 48);
    num = num - dat*10;
    zero = 1;
}
if(zero == 4){
    WriteTxBuff(48);
    WriteTxBuff(48);
    WriteTxBuff(48);}
if(zero == 3){
    WriteTxBuff(48);
    WriteTxBuff(48);}
if(zero == 2)
    WriteTxBuff(48);

dat = (unsigned int) (num);
WriteTxBuff(dat + 48);
}
```

INDICE DE FIGURAS

Figura 1.1 Diagrama de la inyección electrónica	3
Figura 1.2 Diagrama del inyector	4
Figura 1.3 Tanque de gas	5
Figura 1.4 Multiválvula	5
Figura 1.5 Vaporizador regulador de presión	6
Figura 1.6 Electroválvulas	6
Figura 1.7 Esquema del sensor de temperatura del líquido refrigerante	9
Figura 1.8 Gráfica de los valores del sensor de temperatura	10
Figura 1.9 Gráfica de módulo de encendido electrónico	12
Figura 1.10 Gráfica de sensor inductivo y rueda fónica	13
Figura 1.11 Gráfica de sensor en el distribuidor	14
Figura 1.12 Gráfica de sensor óptico en el distribuidor	15
Figura 1.13 Gráfica de sensor hall en el distribuidor	16
Figura 1.14 Gráfica del sensor de posición	17
Figura 1.15 Gráfica del sensor de presión absoluta	18
Figura 1.16 Diagrama del detector de cruce por cero	19
Figura 1.17 Pantalla del detector de cruce por cero	19
Figura 1.18 Esquema de configuración darlington	20
Figura 1.19 Amplificador darlington básico	21
Figura 1.20 Amplificador de corriente	22
Figura. 2.1. Diagrama general del Hardware del sistema de inyección electrónica a gas	26
Figura. 2.2. Diagrama general del DSC dsPIC30F4013	27
Figura. 2.3. (A2.1)	29
Figura. 2.4. Conexiones posibles por medio del puerto serial.	29
Figura. 2.5. Conexiones posibles por medio del puerto serial.	30
Figura 3.1 Ubicación de inyectores	33
Figura 3.2 Controlador	34
Figura. 3.3 (A2.1)	35
Figura. 3.4 (A2.1)	35
Figura. 3.5. (A2.1)	36
Figura 3.6. Fuente de poder de 5Vcd	37

Figura 3.7 Diagrama de bloques	38
Figura 3.8 Diagrama de flujo	39
Figura. 3.9. Diagrama de tiempo de un RTOS con derecho de preferencia.	49
Figura. 3.10 Interacción entre los elementos de Software	50
Figura. 3.11 Diagrama de comunicaciones	52
Figura. 3.12 señal del PC osciloscopio	53
Figura. 3.13 señal del PC osciloscopio	53
Figura. 3.14 señal del PC osciloscopio	54
Figura. 3.15 señal del PC osciloscopio	55
Figura. 3.16 señal del PC osciloscopio	55
Figura. 3.17 señal del PC osciloscopio	56
Figura. 3.18 señal del PC osciloscopio	56
Figura 3.19 Pantalla señal de inyectores	57
Figura 3.20 Pantalla señal de inyectores	57
Figura. 3.21 señal del PC osciloscopio	58
Figura. 3.22 señal del PC osciloscopio	58
Figura 3.23 Pantalla detector de cruce por cero	59
Figura 3.24 Pantalla Señal de inyectores	59
Figura 3.25 Pantalla inyector y sensor inductivo	60
Figura 3.26 Pantalla inyector y sensor inductivo	60
Figura 4.1 integracion del sistema de gas	62
Figura 4.2 conectores de sensores y actuadores	63
Figura 4.3 conectores de inyectores	64
Figura 4.4 conector deL modulo de control	64
Figura 4.5 Diagrama de bloques	65
Figura 4.6 modulo de control	65
Figura 4.7 Sensor de temperatura del refrigerante	66
Figura 4.8 Sensor TPS	67
Figura 4.9 Sensor TPS y avance de vacio	67
Figura 4.10 Sensor MAP	68
Figura 4.11 Sensor inductivo	69
Figura 4.12 Pantalla sincronismo e inyector	69
Figura 4.13 Pantalla señal de inyectores	70
Figura 4.14 Pantalla señal MAP 1	70
Figura 4.15 Pantalla señal MAP 2	71

Figura 4.16 Pantalla señal MAP 3	71
Figura 4.17 Pantalla señal MAP 4	72
Figura 4.18 Pantalla señal MAP 5	72
Figura 4.19 Pantalla señal MAP 6	73
Figura 4.20 Pantalla detector de cruce por cero en arranque	73

INDICE DE TABLAS

Tabla. 2.1. Requerimientos del Hardware	25
Tabla. 2.2. Bloques funcionales según requerimientos	26
Tabla. 2.3. Designación de los canales de comunicación en el sistema.	28
Tabla. 3.1. Funciones y requerimientos generales del software embebido	41
Tabla 3.2 Herramientas de Software	42
Tabla. 3.3. Configuración de los puertos	46
Tabla. 3.4. Funciones de configuración.	48
Tabla. 3.5 Funciones del RTOS CMX-Scheduler	49
Tabla 4.1 Designación de cable	63

GLOSARIO

ADC Analog to Digital Converter Conversor análogo – digital

bootloader Cargador de arranque - Pequeño programa que se ejecuta después del arranque de un procesador y que tiene como función principal administrar la descarga de código y grabar el mismo en la memoria del dispositivo.

BOR Brown-Out Reset Circuito que fuerza a un microprocesador o semejante a resetearse cuando el voltaje de alimentación cae por debajo de un límite preprogramado.

DAC Digital to Analog Converter Conversor digital – analógico.

DSC Digital Signal Controller Controlador digital de señales - Dispositivo que integra la capacidad y arquitectura de un DSP con la integración y fácil desarrollo de un microcontrolador.

DSP Digital Signal Processor Procesador Digital de señales –es un procesador o microprocesador que posee un juego de instrucciones, un hardware y un software optimizados para aplicaciones que requieran operaciones numéricas a muy alta velocidad y son usado en aplicaciones que requieren comportamiento en tiempo real.

EEPROM Electrically-Erasable Programmable Read-Only Memory Memoria sólo de lectura programable, borrrable eléctricamente. emulador Dispositivo o programa diseñado para hacer trabajar (imitar) a una computadora como otro dispositivo. Un emulador se refiere tanto a un equipo de hardware como a un software.

Filtro antialiasing Circuito analógico que trabaja como filtro pasa bajos con el fin de que los datos digitalizados no se corrompan por la presencia de frecuencias superiores a la mitad de la frecuencia de muestro en la señal de entrada. filtro IIR

Infinite Impulse Response Filtro digital cuya respuesta al impulso es infinita debido a la presencia en sus cálculos de resultados anteriores.

FLASH Tecnología de memoria no volátil mucho más barata que la EEPROM pero con la desventaja de que se graba/borra solo por bloques.

GUI Graphical User Interface Interfaz gráfica de Usuario – Programa que sirve de nexo entre un equipo y un usuario para configurar y visualizar información del dispositivo.

GUIDE Graphical User Interface Development Environment Programa incluido en matlab que permite la creación de interfases gráficas de usuario.

HIL Hardware In the Loop Hardware en el lazo – Técnica por la cual se sustituye un equipo físico por un sistema digital que se comporta como el equipo físico real.

LVD Low Voltage Detect Detección de bajo voltaje – Circuito que avisa a un microcontrolador o semejante si hay una caída de voltaje con el fin de respaldar información o tomar otras acciones posible una descripción matemática para un sistema físico.

OPAMP Operational Amplifier Amplificador Operacional

PCB Printed Circuit Board Placa de circuito impreso

PGA Programmable Gain Amplifier Amplificador de ganancia programable – Circuito discreto o integrado que por medios digitales permite fijar la ganancia de un amplificador analógico.

FIFO First In – First Out Mecanismo lógico de almacenamiento de datos donde el primer elemento en ser ingresado en memoria es el primero en ser extraído

Planta Conjunto de sistemas mecánicos, eléctricos, químicos, etc., que realizan algún proceso.

POR Power-On Reset Circuito que fuerza a un microcontrolador o semejante a estar en estado de reset por un tiempo después de encenderse hasta que el voltaje de alimentación se estabilice.

Preemptive Característica de un sistema operativo que permite escribir una tarea como si esta fuera la única que se ejecuta en el procesador. Esto indica que el sistema operativo está completamente a cargo de despachar la ejecución de las distintas tareas según su prioridad.

PWM Pulse Width Modulation Modulación por ancho de pulso

RAM Random Access Memory Memoria de acceso aleatorio

RISC Reduced Instruction Set Computer Computadora con Conjunto de Instrucciones

RS232 (también conocido como EIA RS-232C) es una interfaz que designa una norma para el intercambio serial de datos binarios.

RTOS Real Time Operating System Sistema operativo en tiempo real - es un sistema operativo que ha sido desarrollado para aplicaciones de tiempo real. Como tal, se le exige corrección en sus respuestas bajo ciertas restricciones de tiempo. Si no las respeta, se dirá que el sistema ha fallado. Para garantizar el comportamiento correcto en el tiempo requerido se necesita que el sistema sea predecible (determinista).

Scheduler Código de programa que es la base de un sistema operativo y es el encargado de administrar la creación, ejecución y destrucción de las distintas tareas definidas, juntamente con los mensajes que estas comparten.

SISO Single Input – Single Output Una sola entrada, una sola salida - Descripción de un sistema por medio de una única ecuación cuya variable dependiente (salida) es función únicamente de una variable independiente (entrada).

Sistemas embebidos Sistema de propósito específico en que una computadora está

completamente encapsulada por el dispositivo que controla y que ejecuta tareas predefinidas con requerimientos bien delimitados y específicos.

SPI Serial Peripheral Interface Interfase de periféricos serial - Estándar de comunicación serial síncrona de alta velocidad muy difundido para controlar periféricos, principalmente a nivel de placa.

Through hole Se refiere al esquema utilizado por componentes electrónicos que involucran el uso de pines que se insertan dentro de agujeros taladrados en el circuito impreso y que se sueldan en el lado opuesto de la placa.

Tiempo real Modo de trabajo de un programa en el que la reacción (cómputos respecto al contexto) ante una acción no demora más que el retardo permitido para una aplicación dada, de tal manera que se perciba “una reacción instantánea del dispositivo”.

UART Universal Asynchronous Receiver Transmitter. Transmisor/Receptor Universal Asíncrono. Hardware que permite enviar datos de manera serial conteniendo la información de sincronización en la misma trama de datos.

ZOH Abreviación para Retenedor de orden cero Forma matemática de indicar el proceso de conversión análoga-digital en el que se retiene el valor de entrada hasta la próxima muestra.

GNC Gas Natural Comprimido.

GLP Gas Licuado de Petróleo.

CUBICAJE Medida de volumen total de aire aspirado por un motor en todos sus cilindros

CILINDRAJE Cubicaje.

TPS Sensor de posición de la mariposa.

MAP Sensor de presión.

ECT Sensor de temperatura.

Cuerpo de mariposa parte donde se aloja la mariposa del acelerador.

MARCHA LENTA RPM minima a la que gira el motor sin apagarse.

RPM Revoluciones por minuto.

DISTRIBUIDOR Parte del motor en donde se direcciona la chispa de encendido a la bujía que se encuentra con su pistón en compresión.

MODULO DE ENCENDIDO Circuito electrónico que genera la chispa que entra al distribuidor.

INYECTOR Válvula on – off que posee características de tiempo muy pequeñas

VACIO Fuerza con la que el motor succiona el aire por la admisión.

FECHA DE ENTREGA:

**COORDINADOR DE LA CARRERA DE
INGENIERÍA ELECTRÓNICA Y CONTROL**

**ALVARO RENAN
PAREDES URQUIZO**

**PAULO CESAR
CONDO VELASTEGUI**