

# **ESCUELA POLITÉCNICA DEL EJÉRCITO**

**SEDE LATACUNGA**



**CARRERA DE INGENIERÍA AUTOMOTRIZ**

**“DISEÑO E IMPLEMENTACIÓN DE UN MÓDULO DE  
ALMACENAMIENTO Y ANÁLISIS DE RPM PARA EL CAMBIO  
DE MARCHA EN UN VEHÍCULO DATSUN 160J”**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO AUTOMOTRIZ**

**ROSERO OBANDO FREDY ALEXANDER  
ZAMBRANO RENGEL SYLVIA VANESSA**

**Latacunga, febrero 2.008**

## CERTIFICACIÓN

Certifico que el presente proyecto fue realizado en su totalidad por los señores: Rosero Obando Fredy Alexander y Zambrano Rengel Sylvia Vanessa, como requerimiento para la obtención del título de Ingeniero Automotriz.

---

Fecha.

---

Ing. Juan Castro.

**DIRECTOR**

---

Ing. Julio Acosta.

**CODIRECTOR**

## **AGRADECIMIENTOS**

Agradezco a mis padres por confiar siempre en mí y brindarme su apoyo incondicional en todo momento, haciendo de esto mi principal herramienta para vencer las distintas adversidades que tuve en la vida universitaria. Gracias a mis maestros que durante todo este tiempo supieron impartirme sus conocimientos para hacer de mí y mis compañeros unos grandes profesionales.

Fredy Rosero O.

A mis padres Sylvia y Arturo por haberme motivado para continuar adelante y llegar con éxito a la culminación de este trabajo de investigación.

A mis hermanos Karina y Diego por confiar en mi.

A mi abuelitos Rosa y Alberto por sus sabias enseñanzas sobre humildad y como ser mejor persona.

A todos mis profesores por todas las enseñanzas recibidas.

A mis mejores amigos por siempre estar junto a mí y ser incondicionales.

A los compañeros de trabajo por compartir sus conocimientos y sabios consejos.

Gracias sinceras para todas las personas que de una u otra manera colaboraron con este trabajo y me apoyaron durante mi vida universitaria.

Sylvia Zambrano R.

## **DEDICATORIA**

El presente trabajo lo dedico a mis padres y a mis hermanas quienes con su esfuerzo y dedicación supieron hacer de mí un hombre de bien. También se lo dedico a Jimena quien con cariño y comprensión siempre ha estado junto a mí en los momentos de alegría y tristeza que he tenido durante estos últimos años.

Fredy Rosero O.

Dedico con mucho cariño este trabajo que sintetiza el esfuerzo, sacrificio, esmero, dedicación y el cumplimiento de mis anhelos y aspiraciones a Dios por haberme guiado, a mis adorados padres, ejemplos de amor, honradez y perseverancia; a mi hermano y hermana por ser parte de mi vida y confiar en mi y a toda mi familia ya que con esfuerzo, responsabilidad y estímulos se puede alcanzar las metas trazadas.

Sylvia Zambrano R.

## **ANTECEDENTES**

Desde hace años atrás los técnicos automotrices decidieron implementar el uso del tacómetro análogo, para poder medir el régimen de giro del motor y a la vez ayudar al conductor a saber en que instante debe realizar el cambio de marcha. Con el paso de los años se logró mejorar y optimizar el rendimiento de los motores de combustión interna, con lo que el tacómetro convencional dejó de ser una herramienta lo suficientemente útil para lograr tener un cajeadó óptimo y aprovechar al máximo la potencia de los nuevos motores. Los actuales tacómetros que tienen shift lite también se ven limitados, ya que su luz indicadora solo se activa a un número de revoluciones determinada, sin evaluar en que cambio de marcha se halla el vehículo. Además los datos de estas RPM no pueden ser almacenadas para su posterior análisis y evaluación; pues el historial de todos estos parámetros ayudarán al técnico automotriz a determinar las posibles causas que ocasionaron averías y deterioro en el motor del vehículo.

## **JUSTIFICACIÓN E IMPORTANCIA**

En la actualidad, en muchos de los casos el mal manejo e impericia del conductor hace que el vehículos sufra graves daños por sobre carga en el motor. El Proyecto planteado tiene la finalidad de implementar un módulo que en función de las RPM y el conocimiento de la marcha del vehículo, permita analizar de qué manera el conductor esta exigiendo al motor del vehículo. Con este módulo se pretende ayudar a los pilotos de competición quienes son los que necesitan optimizar su manejo, ya que cuando están en competencias los ruidos del ambiente y del vehículo en sí no les ayudan a cajar su vehículo a las RPM adecuadas, y su única herramienta de ayuda es el tacómetro, que muchas veces no es usado correctamente.

Con la realización de este proyecto se logrará monitorear el historial de las RPM y del cambio de marcha del vehículo durante un periodo determinado, con lo que los datos obtenidos podrán ser procesados y visualizados en una PC, en donde el ingeniero automotriz estará en capacidad de analizar los

resultados que se obtenga y así sugerir al conductor que corrija algún aspecto de su forma de manejo. Al final lo que se logrará es poseer una herramienta confiable que ayude a evaluar el estilo de manejo de un conductor y así evitar que el motor del vehículo sufra un temprano deterioro.

## **OBJETIVO GENERAL**

Diseñar e implementar un módulo de almacenamiento y análisis de rpm para el cambio de marchas y así mejorar el desempeño del conductor en un vehículo

## **OBJETIVOS ESPECÍFICOS**

- Seleccionar el microcontrolador, los sensores, las memorias, los conectores y demás elementos electrónicos para el desarrollo del proyecto.
- Diseñar e implementar una herramienta de fácil uso que ayude a los pilotos de competición a optimizar su estilo de manejo y evitar que destruyan sus motores de manera prematura.
- Obtener las curvas de torque y potencia reales del motor en un banco de pruebas, para el análisis de funcionamiento del motor y optimización del manejo del conductor.
- Lograr la comunicación entre el módulo instalado en el vehículo y una PC para la transmisión de datos respectiva.
- Desarrollar un software en LabView para la recepción y análisis de los datos de marcha y RPM del vehículo.
- Determinar si para un conductor es mucho más fácil visualizar las RPM en un tacómetro análogo o en un tacómetro digital.

## **METAS**

- Definir todos los elementos electrónicos que se van a usar para el desarrollo del proyecto en el transcurso de un mes.

- Diseñar el circuito electrónico del módulo y demás conexiones para su funcionamiento correcto en un plazo de dos meses.
- Crear un Software para el manejo de datos de las RPM y cambio de marcha del vehículo en el lapso de dos meses.
- Implementar la comunicación entre el módulo y una PC en el período de un mes.

## INTRODUCCIÓN

El vertiginoso avance tecnológico que se está logrando en la electrónica del vehículo, obliga y compromete a las personas vinculadas al campo Automotriz a tratar de crear nuevos dispositivos que ayuden al conductor a optimizar la forma de manejo del vehículo. Con frecuencia los conductores someten a elevados regimenes de giro a los motores de sus vehículos, por no realizar los cambios de marcha a las RPM adecuadas, causando deterioro prematuro en estos componentes.

Con el diseño e implementación de un tacómetro inteligente se logrará aprovechar al máximo la potencia y el torque del vehículo, pues los regimenes de giro del motor a los que se active la shift lite en cada cambio podrán ser programados individualmente. Además se podrá modificar cada uno de los regimenes dependiendo del terreno o situación geográfica en donde se esté conduciendo el vehículo.

El módulo electrónico del tacómetro, toma desde la bobina de ignición del vehículo tomará la señal de frecuencia de giro del motor, esta señal será ingresada a un microcontrolador del módulo, en donde esta frecuencia mediante conversiones matemáticas se transformará a RPM. Se ubicará cuatro sensores en la caja de cambios, la misma que ayudará el posicionamiento de la marcha. Todas las señales de entrada descritas entrarán al módulo, en donde el microcontrolador procesará los datos y luego almacenará estos en una memoria EEPROM. El conductor del vehículo a través del uso de un lcd y el teclado podrá tener una interfaz directa con el módulo.

Los datos almacenados en el módulo, a través de la comunicación serial serán transmitidos a una PC, y mediante el uso de un software desarrollado en Lab View podremos obtener las curvas de RPM en función de tiempo y posición de la marcha, el software del módulo además ayudará a conocer los valores promedio, máximo y mínimo de revoluciones del motor a las que está cajeando el conductor.

# ÍNDICE

<b>CONTENIDO</b>	<b>PÁG.</b>
Carátula.	i
Certificación.	ii
Agradecimiento.	iii
Dedicatoria.	iv
Antecedentes.	v
Justificación e Importancia.	v
Objetivo General.	vi
Objetivos Específicos.	vi
Metas.	vi
Introducción.	viii
Índice de Contenidos.	ix
Índice de Figuras.	xv
Índice de Tablas.	xx

## **I. CAPÍTULO MARCO TEÓRICO**

1.1 Fundamentos Metodológicos.	1
1.1.1 Tipos de Investigación.	1
1.1.2 Método de Investigación.	2
1.2 Los tacómetros.	2
1.2.1 Definición.	2
1.2.2 Generalidades de los tacómetros.	2
1.2.2.1 Uso y utilización.	2
1.2.2.2 Tipos de tacómetros que existe en el mercado automotriz.	4
1.2.2.3 Comparativa de tacómetros.	4
1.2.2.3.1 Características de un tacómetro interno.	4
1.2.2.3.2 Características de un tacómetro externo.	6
1.2.3 Tipos y señales de entrada de los tacómetros.	7
1.2.3.1 Generador de pulsos.	8

1.2.3.2	Base de tiempo.	8
1.2.3.3	Contadores.	8
1.2.4	Esquema de conexión de un tacómetro.	8
1.3	Sistema de Encendido Convencional.	9
1.3.1	Descripción y Componentes.	9
1.3.1.1	Bobina de encendido.	9
1.3.1.2	Resistencia previa.	10
1.3.1.3	Ruptor.	10
1.3.1.4	Condensador.	10
1.3.1.5	Distribuidor de encendido.	10
1.3.1.6	Variador de avance centrífugo.	10
1.3.1.7	Variador de avance de vacío.	10
1.3.1.8	Bujías.	10
1.3.2	Funcionamiento.	10
1.4	Generalidades de la caja de cambios.	13
1.4.1	Influencia de una caja de cambios en el funcionamiento del motor.	15
1.5	Curvas Características.	16
1.5.1	Introducción.	16
1.5.2	El Par motor.	16
1.5.2.1	Variables que influyen en el par motor.	18
1.5.3	Diagrama del Par Motor en un motor de 4 cilindros.	19
1.5.4	La Potencia y tipos.	21
1.5.4.1	Potencia de un motor.	21
1.5.4.2	Tipos de potencia.	21
1.5.4.2.1	Potencia efectiva de un motor.	22
1.5.4.2.2	Potencia fiscal.	22
1.5.4.2.3	Potencia másica.	22
1.5.4.2.4	Potencia específica.	23
1.5.5	Curva de Potencia.	23
1.5.6	Consumo Específico.	23
1.5.7	Gráficos de Curvas características.	25

## II. CAPÍTULO

### ANÁLISIS DE ELEMENTOS ELECTRÓNICOS

2.1 Sensores.	27
2.1.1 Introducción.	27
2.1.2 Sensores de Retro.	28
2.2 Filtrado de señales.	28
2.2.1 Filtros.	28
2.2.2 Tipos de filtro.	29
2.3 Memorias programables.	32
2.3.1 Ventajas de la EEPROM.	32
2.3.2 Memoria EEPROM 24LC512.	33
2.4 El RTC DS1307.	34
2.5 El CI 4093.	35
2.6 Comunicación serial.	36
2.6.1 Conceptualización.	36
2.6.2 Tipos de Comunicaciones Seriales.	36
2.6.2.1 Comunicación serial síncrona.	36
2.6.2.2 Comunicación serial asíncrona.	37
2.6.3 Modos de Transmisión de Datos.	38
2.6.3.1 Simples.	38
2.6.3.2 Half-duplex.	38
2.6.3.3 Full-duplex.	38
2.6.3.4 Full/full-duplex.	38
2.6.4 Descripción de la norma RS232.	38
2.6.5 Descripción del Bus I2C.	39
2.6.5.1 Introducción.	39
2.6.5.2 Descripción de las señales.	40
2.6.5.3 Protocolo de comunicación del bus I2C.	40
2.6.6 El Max 232.	43
2.6.7 Conector D9 del PC.	44
2.7 Lcd y teclados.	45

### III. CAPÍTULO

#### DISEÑO Y DIAGRAMACIÓN DE CIRCUITOS ELECTRÓNICOS

3.1 Definición de etapas de funcionamiento del módulo.	48
3.1.1 Inicialización y configuración.	48
3.1.2 Obtención de número de revoluciones del motor.	49
3.1.3 Obtención de la posición de marcha.	49
3.1.4 Interpretación y procesamiento de los datos recolectados.	50
3.1.5 Visualización de datos procesados.	50
3.1.6 Interface con la PC para descarga de datos obtenidos.	50
3.2 Selección de sensores.	50
3.3 Selección de circuitos integrados.	51
3.4 Selección del microcontrolador.	52
3.5 Selección de la memoria externa para almacenamiento de datos.	53
3.6 Selección de LCD, teclados y displays.	54
3.7 Implementación de la comunicación serial.	54
3.7.1 Comunicación serial PIC a PIC.	54
3.7.2 Comunicación serial pic a PC con CI Max232.	55
3.7.3 Comunicación I2C pic con RTC y memoria EEPROM.	56
3.8 Programación del microcontrolador.	57
3.8.1 Diagrama de Flujo para obtención y visualización de datos de RPM.	58
3.8.2 Diagrama de Flujo para descarga de datos de EEPROM.	60
3.9 Diagrama de circuitos del módulo.	60
3.9.1 Adquisición de datos.	61
3.9.2 Almacenamiento de datos y RTC.	61
3.9.3 Interfaz LCD, teclado y PIC maestro.	62
3.9.4 Interfaz displays y PIC esclavo.	62
3.9.5 Interfaz PIC maestro y MAX232.	63
3.9.6 Interfaz con shift lite.	64
3.9.7 Circuito de alimentación regulado a 5V.	64
3.10 Diseño y construcción del circuito impreso.	65
3.11 Diseño del software de visualización y análisis de datos en LabView.	65

3.11.1 Introducción.	65
3.11.2 Programa para descargar datos desde una memoria EEPROM.	67
3.11.3 Presentación de la pantalla principal.	73
3.11.4 Presentación de las ventanas del programa.	74
3.12 Instalación del módulo e interfaces en el vehículo.	89

#### **IV. CAPÍTULO**

### **PRUEBAS EXPERIMENTALES DE OPERACIÓN DEL MÓDULO DE ALMACENAMIENTO Y ANÁLISIS DE LAS RPM Y CAMBIO DE MARCHA DE UN VEHÍCULO**

4.1 Prueba de funcionamiento de los sensores de marcha.	94
4.2 Prueba de recepción de señales en el módulo.	96
4.3 Prueba de comparación de lecturas de RPM entre el módulo y el multímetro de uso automotriz.	99
4.4 Prueba de comunicación entre el módulo y el PC.	100
4.5 Descripción de manejo del módulo.	101
4.6 Prueba y descripción de funcionamiento del software de visualización y análisis de datos en LabView.	105
4.6.1 Pruebas de la Interface de Descarga	105
4.6.1.1 Prueba del Hardware.	106
4.6.1.2 Prueba de Software.	107
4.6.2 Prueba de recolección de datos (Excel) .	110
4.6.2 Descripción y comprobación de funcionamiento del software de visualización en LabView.	111

#### **V. CAPÍTULO**

### **CONCLUSIONES Y RECOMENDACIONES**

5.1. Conclusiones.	119
5.2. Recomendaciones.	121
<b>BIBLIOGRAFÍA.</b>	<b>122</b>

- ANEXO A. HOJA DE DATOS DE TORQUE Y POTENCIA DEL DATSUN 160J OBTENIDAS EN BANCO DE PRUEBAS.**
- ANEXO B. GRÁFICAS DE CURVA DE TORQUE Y POTENCIA DEL DATSUN 160J OBTENIDAS EN BANCO DE PRUEBAS.**
- ANEXO C. CIRCUITO COMPLETO DEL MÓDULO DE ANÁLISIS Y ALMACENAMIENTO.**
- ANEXO D. DATASHEET CD4093 (Compuerta NAND Schmitt -Trigger).**
- ANEXO E. DATASHEET DS1307 (Reloj en tiempo real).**
- ANEXO F. DATASHEET 24LC512 (Memoria EEPROM).**
- ANEXO G. CÓDIGO DE PROGRAMACIÓN REALIZADO EN PIC BASIC.**

# ÍNDICE DE FIGURAS

<b>CONTENIDO</b>	<b>PÁG.</b>
<b>I. CAPÍTULO</b>	
Figura 1.1. Circuito de funcionamiento de un tacómetro.	3
Figura 1.2. Diagrama típico de conexión.	9
Figura 1.3. Circuito de un tacómetro.	9
Figura 1.4. Esquema de un sistema de encendido convencional.	10
Figura 1.5. Esquema eléctrico del sistema de encendido.	11
Figura 1.6. Misión del ruptor.	12
Figura 1.7. Variación de corriente y tensión.	13
Figura 1.8. Constitución de una caja de cambios.	13
Figura 1.9. Esquema de los elementos relacionados con el par motor.	18
Figura 1.10. Variación del brazo del par.	19
Figura 1.11. Ejemplos de curvas de par motor de motores de gasolina.	20
Figura 1.12. a) de un motor con el par muy constante.	
b) de un motor con el par máximo a altas revoluciones.	23
Figura 1.13. Ejemplos de curvas del consumo específico.	25
Figura 1.14. Ejemplo de gráficas de curvas características.	26
<b>II. CAPÍTULO</b>	
Figura 2.1. Modelos de interruptores de reversa.	28
Figura 2.2. Concepto de filtro eléctrico.	29
Figura 2.3. Funciones de transferencia ideales y reales de los filtros.	31
Figura 2.4. Memoria 24LC512.	33
Figura 2.5. Descripción de pines de la EEPROM.	34
Figura 2.6. Reloj en tiempo real.	34
Figura 2.7. El circuito integrado 4093.	35
Figura 2.8. Estructura carácter que se transmite en serial asíncrona.	37
Figura 2.9. Inicio de la comunicación I2C.	41
Figura 2.10. La escritura datos en la memoria EEPROM.	42
Figura 2.11. Fin de la comunicación I2C.	42

Figura 2.12. Esquema del MAX232.	43
Figura 2.13. Conectores DB9 del PC.	44
Figura 2.14. LCD.	45
Figura 2.15. Estructura interna el teclado.	47

### III. CAPÍTULO

Figura 3.1. Interface LCD, microcontrolador y teclado.	48
Figura 3.2. Esquema recolección de número de revoluciones del motor.	49
Figura 3.3. Ubicación de sensores de posición en la caja de cambios.	49
Figura 3.4. LCD y displays.	50
Figura 3.5. Ubicación del sensor para detectar la posición de la palanca de cambios.	51
Figura 3.6. Distribución de pines del PIC 16F877A.	52
Figura 3.7. Memoria EEPROM 24LC512.	53
Figura 3.8. LCD, display y teclado respectivamente.	54
Figura 3.9. Comunicación entre PIC esclavo y PIC maestro.	55
Figura 3.10. Conexión del CI MAX232.	56
Figura 3.11. Conexión I2C PIC con RTC y memoria EEPROM.	57
Figura 3.12. Diagrama de flujo para obtención y visualización de datos de RPM.	59
Figura 3.13. Diagrama de flujo para descarga de datos de memoria EEPROM.	60
Figura 3.14. Circuito de adquisición de datos.	61
Figura 3.15. Circuito de almacenamiento de datos y RTC.	61
Figura 3.16. Circuito de interface LCD, teclado y PIC maestro.	62
Figura 3.17. Circuito de interface displays y PIC maestro.	62
Figura 3.18. Multiplexacion de displays.	63
Figura 3.19. Circuito de interface PIC maestro y MAX232.	63
Figura 3.20. Circuito de interface con shift lite.	64
Figura 3.21. Circuito de alimentación regulada a 5V.	64
Figura 3.22. Plantilla para la construcción del circuito impreso.	65
Figura 3.23. Código de la secuencia cero.	68
Figura 3.24. Código de la secuencia uno.	68

Figura 3.25. Código de la secuencia dos.	69
Figura 3.26. Código de la secuencia tres.	69
Figura 3.27. Código para guardar los títulos.	69
Figura 3.28. Código para almacenar los datos hora y fecha en Excel.	71
Figura 3.29. Código para almacenar los datos regímenes del motor en Excel.	71
Figura 3.30. Código para almacenar el número de cilindros y número de marcha en Excel.	71
Figura 3.31. Código para separar los datos RPM de la primera marcha.	72
Figura 3.32. Código para guardar los datos en Excel.	73
Figura 3.33. Ventana Principal.	74
Figura 3.34. Ventana de obtención de datos.	75
Figura 3.35. Código para la creación de arreglo de datos.	76
Figura 3.36. Código para el acceso de los datos de tiempo, marcha y RPM.	76
Figura 3.37. Código para unir múltiples arreglo de datos, desplegadas en una tabla.	77
Figura 3.38. Ventana que muestra las curvas RPM vs. Tiempo y Marcha vs. Tiempo.	77
Figura 3.39. Programa para la realización de curvas.	78
Figura 3.40. Programa de separación de datos.	79
Figura 3.41. Bloque de comparaciones.	79
Figura 3.42. Ventana que muestra las curvas Torque vs. RPM y Potencia vs. RPM.	80
Figura 3.43. Ventana de evaluación de uso de la shift lite.	81
Figura 3.44. Ventana de valores máximos y mínimos.	81
Figura 3.45. Obtención de los valores máximos y mínimos.	82
Figura 3.46. Ventana que muestra valores promedios y promedio general.	82
Figura 3.47. Programa para obtener los valores promedios RPM.	83
Figura 3.48. Programa para obtener los valores promedios torque.	83
Figura 3.49. Ventana se simulación revoluciones, torque y potencia.	84
Figura 3.50. Programa para indicar la simulación de potencia y torque.	84

Figura 3.51. Programa para mostrar las RPM en el tacómetro.	86
Figura 3.52. Comparación de marcha y activación del led.	86
Figura 3.53. Ventana para consultar reporte de datos.	87
Figura 3.54. Programa para adquirir reporte de datos.	88
Figura 3.55. Programa para guardar el reporte de datos.	88
Figura 3.56. Conexión en bobina.	89
Figura 3.57. Conexión en batería.	89
Figura 3.58. Interruptor doble tiro.	90
Figura 3.59. Instalación de los sensores de posición de marcha.	90
Figura 3.60. Implementación de cableado hacia los sensores de posición.	90
Figura 3.61. Montaje de los shift lite.	91
Figura 3.62. Potenciómetro shift lite.	91
Figura 3.63. Soque pcb.	92
Figura 3.64. Acondicionamiento del tablero del vehículo.	92
Figura 3.65. Implementación del cableado.	92
Figura 3.66. Placa y alojamiento.	93
Figura 3.67. Placa en alojamiento	93
Figura 3.68. Placa instalada en vehículo.	93

#### **IV. CAPÍTULO**

Figura 4.1. SerialCom, prueba de transmisión serial.	100
Figura 4.2. Mensajes de visualización del módulo.	101
Figura 4.3. Visualización de la primera pantalla.	101
Figura 4.4. Ejemplo de desplazamiento con la tecla A.	102
Figura 4.5. Ejemplo de desplazamiento en el teclado.	103
Figura 4.6. SerialCom. prueba de hardware	106
Figura 4.7 Cadena de datos recibida en LabView.	107
Figura 4.8. Página de datos archivos “reporte”	110
Figura 4.9. Datos recolectados en Excel (reportes generales)	111
Figura 4.10. Funcionamiento de la ventana “inicio “	112
Figura 4.11. Funcionamiento de la ventana “obtención datos “	112
Figura 4.12. Funcionamiento de la ventana “curvas RPM “	113

Figura 4.13. Funcionamiento de la ventana “curvas RPM2 “	113
Figura 4.14. Funcionamiento de la ventana “evaluar Shif Lite “	114
Figura 4.15. Funcionamiento de la ventana “valores max.min “	115
Figura 4.16. Funcionamiento de la ventana “valores promedio “	115
Figura 4.17. Funcionamiento de la ventana “simulación “	116
Figura 4.18. Funcionamiento de la ventana “consulta“	117
Figura 4.19. Reportes Generales c:\consulta.xls	117
Figura 4.20. Reportes Individuales c:\reportes.xls	118

## ÍNDICE DE TABLAS

<b>CONTENIDO</b>	<b>PÁG.</b>
Tabla I.1. Características de un tacómetro interno.	5
Tabla I.2. Características de un tacómetro interno.	7
Tabla II.1. Ejemplo de filtros pasivos RC.	30
Tabla II.2. Descripción de pines del conector DB9.	44
Tabla II.3. Descripción de pines de una LCD.	46
Tabla III.1. Instrucciones de Pic Basic.	57
Tabla IV.1. Combinaciones de los estados de los sensores.	95
Tabla IV.2. Resultados de la prueba externa.	95
Tabla IV.3. Resultado de la prueba de recepción y visualización de señales en el módulo.	96
Tabla IV.4. Factor de multiplicación.	98
Tabla IV.5. Resultados de la prueba de ralentí.	99
Tabla IV.6. Resultados de la prueba de comparación de lecturas de RPM.	99
Tabla IV.7. Funciones de los submenús.	103

# I. CAPÍTULO

## MARCO TEÓRICO.

### 1.1 FUNDAMENTOS METODOLÓGICOS.

#### 1.1.1 Tipo de Investigación.

Dentro de los tipos de investigación se utilizará el exploratorio y el descriptivo por las siguientes consideraciones:

Al principio es de tipo Exploratorio, porque permite familiarizarnos con el problema y recopilar la información inicial para articular y formular las variables que se tendrán durante el desarrollo del proyecto. Los estudios exploratorios se efectúan cuando el objetivo es examinar un tema o problema de "investigación poco estudiado o que no ha sido abordado antes". Constituyéndose así en un estudio preliminar, una primera aproximación al problema.

Luego será de tipo Descriptivo, se conoce como se encuentra la situación del problema, es decir describirlo, utilizando técnicas de recolección de la información que ya pudo haber sido abordado en estudios anteriores.

Aplicando los conceptos de los tipos de investigación mencionados anteriormente en el proyecto a desarrollar, podemos decir que estos están vinculados de la siguiente manera:

**Exploratorio:** El estudio pretendió realizar un primer acercamiento al funcionamiento de un tacómetro digital y programación de la shift lite en vehículos de competición y convencionales; siendo el acceso hacia esta información muy limitada, ya que los manuales y datos técnicos de los tacómetros existentes en el mercado son muy reservados y forman parte de la

propiedad intelectual de cada una de las fábricas que los construye y ensambla.

**Descriptivo:** Mediante este paso se logró aclarar y comprender la información recolectada, pues a través del objeto de estudio y las relaciones de éste con otros objetos se fortaleció los conocimientos en el funcionamiento de elementos electrónicos, protocolos de comunicación y lenguajes de programación; siendo ya mucho más fácil acceder a esta información pues todo sustento teórico ya se lo encuentra en muchas fuentes bibliográficas.

### **1.1.2 Método de Investigación.**

Los principales métodos utilizados fueron:

El método inductivo - deductivo, como método general, por cuanto se partió de una hipótesis o idea que fue desarrollada y verificada en el transcurso del proyecto para llegar a las conclusiones y recomendaciones, muy útiles para personas que también deseen elaborar proyectos similares.

## **1.2 LOS TACÓMETROS.**

### **1.2.1 Definición.**

Se conoce como tacómetros, a los instrumentos que sirven para medir la velocidad de rotación de piezas giratorias (polea del cigüeñal). El tacómetro es un dispositivo que mide las revoluciones por minuto (RPM) del rotor de un motor o una turbina, velocidad de superficies y extensiones lineales. Casi universalmente están calibrados en revoluciones por minuto (RPM), aunque para fines particulares los hay con otras escalas.

### **1.2.2 GENERALIDADES DE LOS TACÓMETROS.**

#### **1.2.2.1 Uso y utilización.**

El uso del tacómetro como se muestra en la figura 1.1, permite obtener el mayor torque del motor sin tener que forzar al máximo, del mismo modo permite realizar los cambios de velocidad con precisión, aumentando el

rendimiento de gasolina y la durabilidad del motor; también nos permite obtener una visualización precisa del corte o límite máximo de revoluciones permitido. Son utilizados para llevar un registro de las velocidades del elemento que se tenga en estudio, que permita saber si está trabajando de forma adecuada. También se pueden emplear para conocer distancias recorridas por ruedas, engranes o bandas.

Uno de los mejores ejemplos es el tacómetro ubicado en el panel de instrumentos del automóvil. Lo rápido que el motor del automóvil está girando es medido en revoluciones por minuto por el instrumento llamado tacómetro o cuentarevoluciones. La mayoría de los autos están optimizados en sus transmisiones de modo estándar y cuando es necesario hacerlos funcionar a las mejores revoluciones posibles para conseguir una mejor aceleración y economizar combustible, entonces el indicador que ayuda a saber lo que está ocurriendo es el tacómetro. Si el vehículo está equipado con transmisión automática entonces el tacómetro se usa poco. Si la aguja del tacómetro se mueve sobre la zona roja entonces el motor está corriendo muy rápido.



**Figura 1.1. Funcionamiento de un tacómetro.**

El indicador en la zona roja es una alerta o una alarma para el conductor, para salvaguardar al motor de daños por exceso de su movimiento de rotación.

La mayoría de los tacómetros muestran simples dígitos indicadores con pequeñas marcas que se leen en revoluciones por minuto x 1000. Entonces lo que se lee en el display hay que multiplicarlo por mil para calcular el número correcto de revoluciones por minuto.

### 1.2.2.3 Tipos de tacómetro que existen en el mercado automotriz.

Básicamente existen dos tipos de tacómetros en el mercado automotriz.

Uno es el tacómetro integrado, que forma parte del cluster (tablero de instrumentos de medición) en el cual se observan cuatro relojes básicos principalmente (gasolina, tacómetro, velocímetro y temperatura del motor). Los tacómetros integrados en cluster (tablero de instrumentos) resultan ser más precisos y representan una métrica real de los ciclos del motor, aunque no tienen lo ostensible de un tacómetro externo.

El otro tipo de tacómetro es el externo, para este tipo existen en varias medidas, que van desde 2" para montaje en columna de instrumentos hasta 5" para montaje sobre tablero. En este tipo de tacómetros hay una gama muy amplia de características, que van desde la iluminación secuencial de 7 colores hasta la integración de medidores extras. (Presión y temperatura de aceite, temperatura de agua, amperímetro, etc.). Los tacómetros externos se vienen usando desde hace mucho tiempo en autos de carreras, principalmente en aquellos motores altamente revolucionados.

### 1.2.2.4 Comparativa de tacómetros.

#### 1.2.2.4.1 Características de un tacómetro interno (Tabla I.1).

**Precisión:** Estos tacómetros han sido especialmente diseñados para representar exactamente el número de revoluciones del motor. Aprovechándose al 100%<sup>(1)</sup>.

**Funcionalidad:** Estos tacómetros se limitan a medir el número de revoluciones del motor y podemos asociarlo fácilmente con la velocidad del automóvil para realizar los cambios. De ahí fuera no integran otra funcionalidad.

**Instalación:** La instalación de estos dispositivos es simple y garantizada, ya que solo se requiere de un solo polo que lleve los pulsos de la ECU al tablero (cluster) y sustituye al actual sin necesidad de adaptaciones, encaja perfectamente.

---

(1) <http://tallermecanico.com>

**Métrica:** En base a la precisión, podemos decir que este tacómetro integra una métrica exacta y adecuada, debido a que la medición es de rango completo (todo el arco) y la escala está perfectamente asociada al movimiento, obteniendo el valor real constante en la medición.

**Ajustes:** Este es el punto más débil de estos tacómetros, debido a que sólo realizan, con precisión absoluta, la medición de las revoluciones.

**Estética:** En proporción a la facilidad de instalación y a que sustituye de manera directa el cluster original, la estética de este dispositivo es perfecta. No tiene cables ni perforaciones extras a la vista. Además tiene la funcionalidad completa de iluminación, misma que viene desde atrás del tablero dándole una mayor visibilidad con contornos, métrica y tipografía iluminados.

**Tabla I.1. Características de un tacómetro interno.**

<b>Características</b>	<b>E</b>	<b>B</b>	<b>R</b>	<b>M</b>	<b>P</b>
Precisión	X				
Funcionalidad			X		
Instalación	X				
Métrica		X			
Ajustes				X	
Estética	X				
Calificación	<b>7</b>				

Donde:

E: Excelente.

B: Buena.

R: Regular.

M: Mala.

P: Pésima.

#### 1.2.2.4.2 Características de un tacómetro externo (Tabla I.2).

**Precisión:** Los tacómetros externos, en su mayoría, vienen topados a 11,000 rpm y esto sólo permite tener un rango máximo del 60% de la lectura. Igualmente el desplazamiento de la aguja por la escala no permite disfrutar en amplitud la respuesta del testigo (shift lite) <sup>(2)</sup>.

**Funcionalidad:** Los tacómetros externos incorporan otras mediciones y opciones que los hacen ser más funcionales e integrales, por ejemplo: presión y temperatura de aceite combinado con la temperatura del agua; desde luego incorpora un testigo luminoso para ejecución de cambios. Todos estos aspectos no vienen integrados en un cluster convencional, donde típicamente se cuenta con indicadores para gasolina, revoluciones, velocidad y temperatura del motor.

**Instalación:** El fabricante sólo incluye una descripción general de la instalación del producto, debido a que la instalación depende directamente del tipo y marca de vehículo. Es un proceso que requiere de otros componentes, tiempo y dedicación.

**Métrica:** En proporción al tamaño de los tacómetros externos, la métrica y distribución de escala son muy abiertos, por lo que no son del todo precisos. Principalmente cuando únicamente damos un vistazo al tacómetro de vez en vez y nos conformamos con ver una posición que se incline sobre los 6.5 de los 11 números representados para esperar la luz y aplicar el cambio.

**Ajustes:** Definitivamente los tacómetros externos incorporan más funciones y con ello más ajustes. Los más comunes o los mínimos aceptados por el usuario resultan ser el ajuste de disparo de la luz para el cambio, la medición variable, la iluminación secuencial en varios colores y la memorización de la lectura de un arrancón.

**Estética:** Debido al tamaño de este dispositivo, muchas veces se ve desproporcionado con respecto al auto, además este tacómetro requiere de montaje en el tablero, siendo inminente la perforación del mismo, tanto para los cables como para fijar la base. Esto resta estética en el instrumento de medición.

---

(2) <http://tallermecanico.com>

**Estética:** Debido al tamaño de este dispositivo, muchas veces se ve desproporcionado con respecto al auto, además este tacómetro requiere de montaje en el tablero, siendo inminente la perforación del mismo, tanto para los cables como para fijar la base. Esto resta estética en el instrumento de medición.

**Tabla I.2. Características de un tacómetro externo.**

<b>Características</b>	<b>E</b>	<b>B</b>	<b>R</b>	<b>M</b>	<b>P</b>
Precisión		X			
Funcionalidad	X				
Instalación			X		
Métrica		X			
Ajustes	X				
Estética		X			
Calificación	<b>8</b>				

Donde:

E: Excelente.

B: Buena.

R: Regular.

M: Mala.

P: Pésima.

### **1.2.3 Tipos y señales de entrada de los tacómetros digitales.**

En los tacómetros digitales, el circuito consta de tres secciones para la visualización de las revoluciones por minuto, las cuales se convertirán tomando en cuenta que se basa en el cierre y apertura de los platinos del vehículo, las tres secciones son:

- Generador de pulsos.
- Base de tiempo.
- Contadores.

**1.2.3.1 Generador de pulsos:** Los platinos son los encargados de aportar, en primera instancia los pulsos, cuando ejecutan su trabajo, al abrir y cerrar sus contactos para generar el alto voltaje en la bobina y luego esta, a las bujías; esto provoca la generación de corrientes inducidas con picos de alto voltaje.

Se debe de atenuar y convertir estas corrientes en pulsos cuadrados que sean compatibles con los circuitos digitales que procesarán toda la información y que nos darán la lectura final.

La sección de entrada (conformada por resistencias y condensadores), encargada de recolectar la información analógica; forman un filtro pasa-bajos para que toda señal generada por la bobina sea eliminada, el zener (puede estar entre 2 y 4 voltios) limita la corriente de entrada en los pines del circuito integrado<sup>(3)</sup>.

**1.2.3.2 Base de tiempo:** Ya se tiene los pulsos, pero aún falta algo para que el tacómetro funcione a plenitud, se necesita un circuito que sincronice el sistema para que las lecturas correspondan exactamente a ciclos por minuto.

Este circuito es la base de tiempos para la toma de datos, habilitar las memorias y borrar de los contadores las lecturas (retorno a cero).

**1.2.3.3 Contadores:** Consiste en circuitos contadores, configurados de tal forma para que puedan mostrar los cientos y miles de las revoluciones por minuto (RPM), las decenas y centenas se obviaron porque un motor no revoluciona tan bajo. Finalmente las lecturas son visualizadas en el display con la ayuda de resistencias como limitadoras de corriente para cada uno de los segmentos de dicho display.

#### **1.2.4 Esquema de conexión de un tacómetro.**

El tacómetro de un vehículo suele ser del tipo electrónico que posee mayor precisión. En la figura 1.2 un diagrama típico de conexión de un tacómetro o cuenta vueltas para el automóvil de sistema de encendido convencional.

---

(3) <http://electronica2000.com>

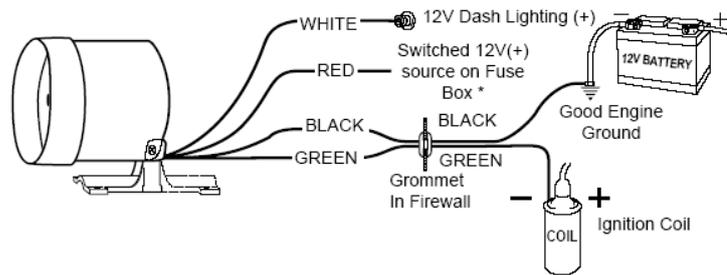


Figura 1.2. Diagrama de conexión en encendido convencional.

En la figura 1.3 se muestra un diagrama de conexión de un tacómetro con sistema de inyección electrónica pues la señal digital de frecuencia para el tacómetro se la toma desde la ECU del automóvil.

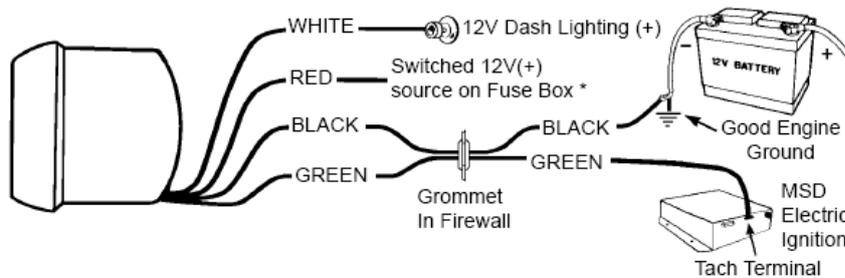


Figura 1.3. Diagrama de conexión en encendido electrónico.

### 1.3 SISTEMA DE ENCENDIDO CONVENCIONAL.

#### 1.3.1 Descripción y Componentes.

El sistemas de encendido por bobina es el más sencillo de los sistemas, en el se cumplen todas las funciones que se le piden a estos dispositivos. En la figura 1.4 se representa el esquema de un sistema de encendido convencional, compuesto por los siguientes elementos:

**1.3.1.1 Bobina de encendido (también llamado transformador):** Su función es acumular la energía eléctrica de encendido que después se transmite en forma de impulso de alta tensión a través del distribuidor a las bujías.

**1.3.1.2 Resistencia previa:** Se utiliza en algunos sistemas de encendido (no siempre). Se pone en cortocircuito en el momento de arranque para aumentar la tensión de arranque.

**1.3.1.3 Ruptor (también llamado platinos):** Cierra y abre el circuito primario de la bobina de encendido, que acumula energía eléctrica con los contactos del ruptor cerrados que se transforma en impulso de alta tensión cada vez que se abren los contactos.

**1.3.1.4 Condensador:** Proporciona una interrupción exacta de la corriente primaria de la bobina y además minimiza el salto de chispa entre los contactos del ruptor que lo inutilizarían en poco tiempo.

**1.3.1.5 Distribuidor de encendido (también llamado delco):** Distribuye la alta tensión de encendido a las bujías en un orden predeterminado.

**1.3.1.6 Variador de avance centrífugo:** Regula automáticamente el momento de encendido en función de las revoluciones del motor.

**1.3.1.7 Variador de avance de vacío:** Regula automáticamente el momento de encendido en función de la carga del motor.

**1.3.1.8 Bujías:** Contiene los electrodos que es donde salta la chispa cuando recibe la alta tensión, además la bujía sirve para hermetizar la cámara de combustión con el exterior.

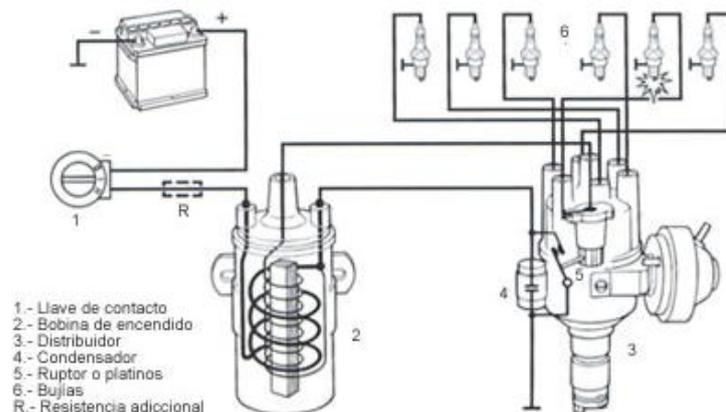
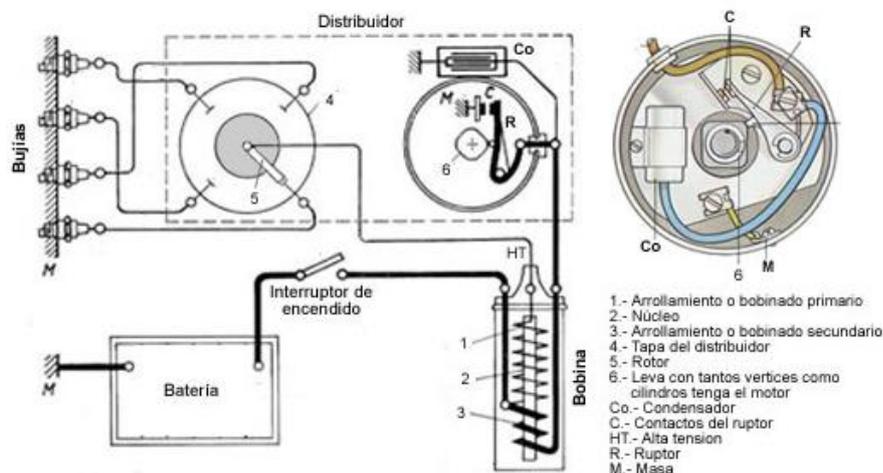


Figura 1.4. Esquema de un sistema de encendido convencional.

### 1.3.2 Funcionamiento.

Una vez que giramos la llave de contacto a posición de contacto el circuito primario es alimentado por la tensión de batería, el circuito primario está

formado por el arrollamiento primario de la bobina de encendido y los contactos del ruptor que cierran el circuito a masa. Con los contactos del ruptor cerrados la corriente eléctrica fluye a masa a través del arrollamiento primario de la bobina. De esta forma se crea en la bobina un campo magnético en el que se acumula la energía de encendido. Cuando se abren los contactos del ruptor la corriente de carga se deriva hacia el condensador que está conectado en paralelo con los contactos del ruptor. El condensador se cargara absorbiendo una parte de la corriente eléctrica hasta que los contactos del ruptor estén lo suficientemente separados evitando que salte un arco eléctrico que haría perder parte de la tensión que se acumulaba en el arrollamiento primario de la bobina. En la figura 1.5 se ve el esquema eléctrico del sistema de encendido convencional. Es gracias a este modo de funcionar, perfeccionado por el montaje del condensador, que la tensión generada en el circuito primario de un sistema de encendido puede alcanzar momentáneamente algunos centenares de voltios<sup>(4)</sup>.

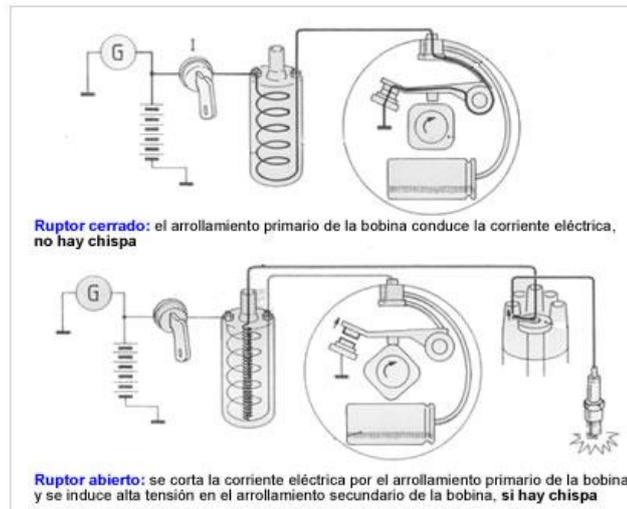


**Figura 1.5. Esquema eléctrico del sistema de encendido.**

La misión del ruptor es cerrar y abrir el circuito del arrollamiento primario (Figura 1.6). Debido a que la relación entre el número de espiras del bobinado primario y secundario es de 100/1 aproximadamente se obtienen tensiones entre los electrodos de las bujías entre 10 y 15000 Voltios.

(4) [http://mecanicavirtual.org/encendido\\_convecnional](http://mecanicavirtual.org/encendido_convecnional)

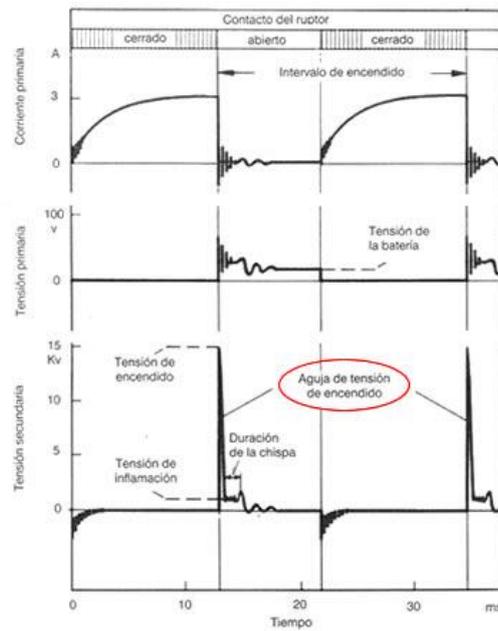
Una vez que tenemos la alta tensión en el secundario de la bobina esta es enviada al distribuidor a través del cable de alta tensión que une la bobina y el distribuidor. Una vez que tenemos la alta tensión en el distribuidor pasa al rotor que gira en su interior y que distribuye la alta tensión a cada una de las bujías.



**Figura 1.6. Misión del ruptor.**

En la figura 1.7 se han representado las variaciones de corriente y tensión (primaria y secundaria de sus circuitos correspondientes) en función del tiempo. En la curva correspondiente a la corriente primaria, pueden verse las oscilaciones y los cambios de sentido de esta en el momento de abrirse los contactos del ruptor. Las mismas oscilaciones se producen en la tensión primaria.

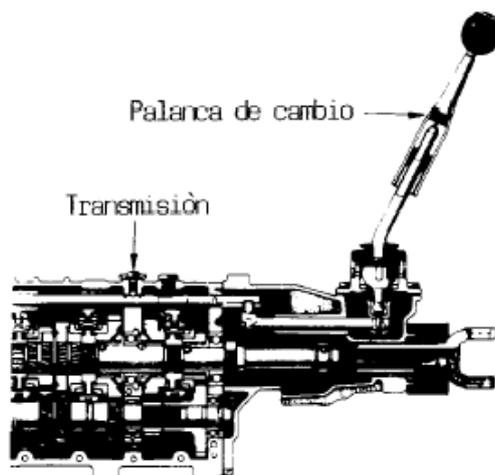
En la curva correspondiente a la tensión secundaria, pueden observarse el máximo valor alcanzado por la tensión de encendido y la subida brusca de la misma (aguja de tensión), para descender también bruscamente al valor de inflamación, en un cortísimo espacio de tiempo. La tensión de inflamación es ondulada, debido a las variaciones de flujo en el primario. La duración de la chispa supone un corte espacio de tiempo en que los contactos del ruptor permanecen abiertos.



**Figura 1.7. Variaciones de corriente y tensión (primaria y secundaria de sus circuitos correspondientes) en función del tiempo.**

#### **1.4 GENERALIDADES DE LA CAJA DE CAMBIOS.**

La caja de cambios es un elemento de transmisión que se interpone entre el motor y las ruedas para modificar el número de revoluciones de las mismas e invertir el sentido de giro cuando las necesidades de la marcha así lo requieran (Figura 1.8). Actúa por tanto, como transformador de velocidad y convertidor mecánico de par.



**Figura 1.8. Constitución de una caja de cambios.**

Si un motor de explosión transmitiera directamente el par a las ruedas, probablemente sería suficiente para que el vehículo se moviese en terreno llano. Pero al subir una pendiente, el par resistente aumentaría, entonces el motor no tendría suficiente fuerza para continuar a la misma velocidad, disminuyendo esta gradualmente, el motor perdería potencia y llegaría a pararse; para evitar esto y poder superar el par resistente, es necesario colocar un órgano que permita hacer variar el par motor, según las necesidades de la marcha. En resumen, con la caja de cambios se "disminuye" o "aumenta" la velocidad del vehículo y de igual forma se "aumenta" o "disminuye" la fuerza del vehículo <sup>(5)</sup>.

Como el par motor se transmite a las ruedas y origina en ellas una fuerza de impulsión que vence las resistencia que se opone al movimiento, la potencia transmitida ( $W_f$ ) debe ser igual, en todo momento, a la potencia absorbida en llanta; es decir:

$$W_f = \frac{C_m \cdot n}{716,2} = \frac{C_r \cdot n_1}{716,2}$$

$$C_m \cdot n = C_r \cdot n_1 \quad \text{(Ec. 1.1)}$$

Donde:

$C_m$ : Par desarrollado por el motor.

$C_r$ : Par resistente en las ruedas.

$N$ : Número de revoluciones en el motor.

$n_1$ : Número de revoluciones en las ruedas.

Si no existiera la caja de cambios el número de revoluciones del motor ( $n$ ) se transmitiría íntegramente a la ruedas ( $n = n_1$ ), con lo cual el par a desarrollar por el motor ( $C_m$ ) sería igual al par resistente en las ruedas ( $C_r$ ). Según esto si en algún momento el par resistente ( $C_r$ ) aumentara, habría que aumentar igualmente la potencia del motor para mantener la igualdad  $C_r = C_m$ .

---

(5) [http://mecanicavirtual.org/cajas\\_de\\_cambio\\_manuales](http://mecanicavirtual.org/cajas_de_cambio_manuales)

En tal caso, se debería contar con un motor de una potencia exagerada, capaz de absorber en cualquier circunstancia los diferentes regímenes de carga que se originan en la ruedas durante un desplazamiento.

La caja de cambios, por tanto, se dispone en los vehículos para obtener, por medio de engranajes, el par motor necesario en las diferentes condiciones de marcha, aumentado el par de salida a cambio de reducir el número de revoluciones en las ruedas. Con la caja de cambios se logra mantener, dentro de unas condiciones óptimas, la potencia desarrollada por el motor.

#### **1.4.1 Influencia de la caja de cambios en el funcionamiento del motor.**

En los vehículos, la caja de cambios o caja de velocidades es el elemento encargado de acoplar el motor y el sistema de transmisión con diferentes relaciones de engranes o engranajes, de tal forma que la misma velocidad de giro del cigüeñal puede convertirse en distintas velocidades de giro en las ruedas. El resultado en la ruedas de tracción generalmente es la reducción de velocidad de giro e incremento del torque.

La razón por la que se necesita hacer cambios es la siguiente: cuando enciende el motor este empieza a dar vueltas manteniendo estable la cantidad de revoluciones, cuando acelera las revoluciones aumentan y el motor se siente con más fuerza, pero el problema radica en que si mantiene trabajando el motor en altas revoluciones este calentará, gastará más combustible y la vida útil del motor no pasaría de unas cuantas horas. La caja de cambios sirve para administrar las revoluciones del motor y darle mayor desplazamiento para esto se vale de sincronizadores y engranes, y de la misma forma funciona el cambio de reversa.

En función de que la velocidad transmitida a las ruedas sea mayor, la fuerza disminuye, suponiendo que el motor entrega una potencia constante: dado que potencia es trabajo por unidad de tiempo y a su vez, trabajo es fuerza por distancia, una distancia mayor (derivada de la mayor velocidad) tiene por consecuencia una fuerza menor. De esta manera la caja de cambios permite

que se mantenga la velocidad de giro del motor y por lo tanto la potencia y par más adecuado a la velocidad a la que se desee desplazar el vehículo.

## **1.5 CURVAS CARACTERÍSTICAS.**

### **1.5.1 Introducción.**

En la técnica, se recurre con frecuencia a la representación gráfica de la relación que tienen dos variables. Este procedimiento proporciona una información inmediata de los valores que adquiere una de las variables al modificarse los valores que toma la otra; mediante una curva referida a dos ejes perpendiculares, en cada uno de los cuales, a una escala conveniente, se marcan los posibles valores que van a adoptar cada una de las variables <sup>(6)</sup>.

El conjunto de ejes y curvas representativas se denomina diagrama.

Las curvas características de los motores, son las siguientes:

- Par motor.
- Potencia.
- Consumo específico.

### **1.5.2 El Par Motor.**

Toda fuerza aplicada a un brazo de palanca origina en este un par.

En el motor de explosión, llamamos par motor al esfuerzo de giro aplicado al codo del cigüeñal por la fuerza de la explosión que le transmite el conjunto biela-pistón. Cuanto mayor sea la presión de empuje sobre el pistón, mayor será el par. La figura 1.9 presenta el esquema de un émbolo unido al cigüeñal mediante la biela, en un momento determinado del comienzo de la expansión (explosión). La parte señalada en esta figura como cigüeñal es su brazo de manivela (brazo del cigüeñal), que es la parte que une la muñequilla con el eje del cigüeñal (cuyo centro es O).

---

(6) Manual CEAC del Automóvil, pág 46, Editorial CEAC S.A., España, 2003.

Cuando el émbolo estaba en el PMS, la muñequilla estaba en el punto b. Al bajar el émbolo, la muñequilla gira, alrededor del centro del cigüeñal (O), desde b hasta a. La presión (P), representada por las flechas que actúan sobre el émbolo, multiplicada por la superficie de éste dan la fuerza actuante.

$$F = P \cdot S \quad (\text{Ec. 1.2})$$

La dirección que tiene esta fuerza F es la del movimiento del émbolo, esto es,, dirigida verticalmente, hacia el centro de giro del cigüeñal (O). Sin embargo, la biela está inclinada, y es a través de ella por donde se transmite la fuerza (F1). Debido a esta inclinación, aparece una fuerza lateral (F2) que empuja al émbolo sobre la pared del cilindro en esa zona. Las fuerzas F1 Y F2 son las componentes de F.

La distancia d es la que separa la fuerza F1 del centro O, siendo perpendicular a la línea de dirección de F1. En la figura no es la línea gruesa (brazo del cigüeñal) que une el punto a (muñequilla) con el centro O, sino la fina que aparece debajo. La distancia d será máxima cuando el brazo del cigüeñal forme un ángulo recto con la biela.

$$Mt = F1 \cdot d \quad (\text{Ec. 1.3})$$

Es el valor del par en ese instante. La distancia d es el brazo de palanca con que actúa F1 sobre el cigüeñal.

El par máximo se encuentra a un régimen intermedio, nunca al régimen máximo del motor; el objetivo de los fabricantes en la actualidad es conseguir un par máximo lo más uniforme posible a lo largo de todos los regímenes de giro. El punto donde se consigue el par máximo es el que consigue la mejor combustión.

Los valores del par motor, se expresan en: Newton-metro (N.m) o kilogramo fuerza-metro (kgf.m).

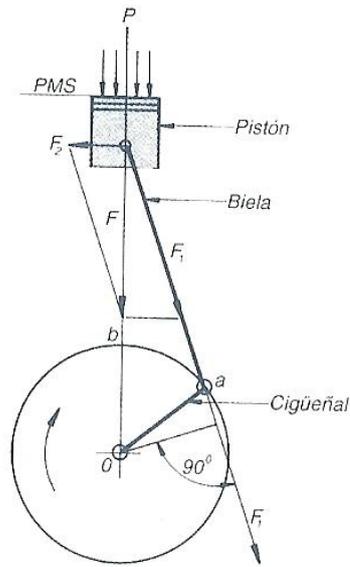


Figura 1.9. Esquema de los elementos relacionados con el par motor.

### 1.5.2.1 Variables que influyen en el par motor.

En el par motor influyen dos variables:

- La fuerza sobre el émbolo.
- La distancia entre el centro del cigüeñal y la dirección de la biela.

- Las fuerzas sobre el émbolo dependen de la presión que haya en la porción de cilindro que queda por encima de él, también dependen del tiempo del ciclo y de la posición que ocupa el émbolo, variando constantemente.

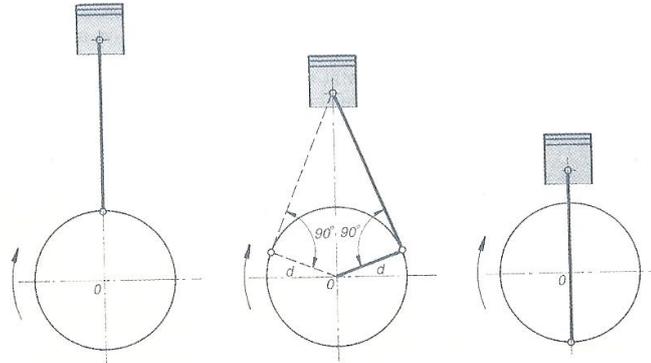
Pero las variaciones que sufre la presión dependen del grado de llenado del cilindro, y éste a su vez es consecuencia, de los avances y retrasos a la apertura y cierre de las válvulas y del número de revoluciones.

Los avances y retrasos no varían sus valores durante el funcionamiento, por lo tanto, para un determinado motor, el grado de llenado, en condiciones meteorológicas y de alturas normales, es función del número de revoluciones

- La otra variable es el brazo del par (Figura 1.10).

Cuando el émbolo está en el PMS, la distancia  $d$  es nula, ya que la prolongación de la biela pasa por el centro del cigüeñal (Figura 1.10, izquierda). Cuando la biela forma un ángulo recto con el brazo del cigüeñal, la distancia  $d$  es igual a la longitud de dicho brazo y es máxima (Figura 1.10, centro). Al llegar

el émbolo al PMI, la biela pasa por el centro del cigüeñal y la distancia  $d$  vuelve a ser nula (Figura 1.10, derecha). En cada media vuelta del cigüeñal, el brazo del par pasa por todos los valores intermedios entre el 0 (nulo) y el máximo <sup>(7)</sup>. Estos valores son independientes del número de revoluciones.



**Figura 1.10. Variación del brazo del par ( $d$ ) en función de la inclinación de la biela.**

### 1.5.3 Diagrama del Par Motor en un motor de 4 cilindros.

Si se van obteniendo los valores del par para cada número de revoluciones y se van trasladando a un diagrama y las distancias sobre el eje horizontal sean las revoluciones a las que se obtienen, uniendo todos los puntos hallados se llega a una curva con el aspecto de las de la figura 1.11, llamada curva del par motor. En esta figura se han representado dos curvas corresponden a un motor de gasolina.

La curva de la figura 1.11 (a) corresponde a un motor para turismo pequeño o medio. Alcanza un par considerable a 1800 RPM y se mantiene casi constante hasta las 4800 RPM, con un máximo de 12 kgf.m (unos 118 N.m) a 3200 RPM. El vehículo que llevara montado este motor sería muy apto para circular por ciudad, haciendo poco uso del cambio de velocidades, ya que con una marcha larga aguantaría bien pequeñas velocidades, y su comportamiento en carretera sería bueno con un régimen de giro moderado y una potencia también moderada.

---

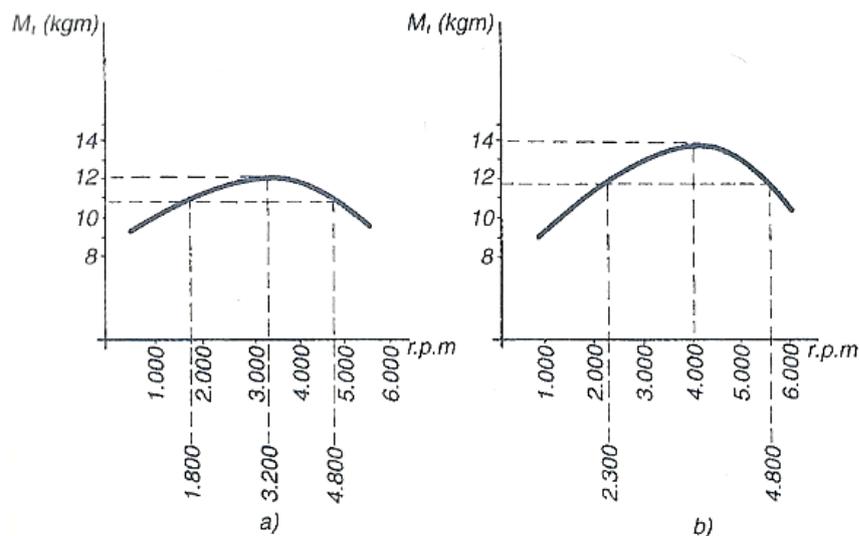
(7) Manual CEAC del Automóvil, pág 46, Editorial CEAC S.A., España, 2003.

La curva de la figura 1.11 (b), corresponde también a un turismo pero con pretensiones más deportivas. Alcanza un par suficiente a 2300 RPM, que se mantienen hasta las 5600 RPM dentro de unos límites aceptables, alcanzando un par máximo de 14 kgf.m (unos 137 N.m) a 4000 vueltas. La *curva* es más puntiaguda.

El coche que equipe este motor tendrá mayores prestaciones que el anterior, pero siempre con el motor más revolucionado, haciendo uso del cambio más frecuentemente para llevar el motor entre las 3200 RPM y el régimen máximo. En los adelantamientos, tendrá que hacer uso de una marcha más corta, para que el motor suba de revoluciones, y la maniobra será más brillante y rápida. La potencia máxima resulta mayor que en el motor anterior pero a mayor número de revoluciones.

Un motor deportivo alcanzará el par máximo a un número de revoluciones más alto y la curva será más aguda. Para sacarle el máximo rendimiento, tiene que girar a velocidades cercanas al régimen máximo, lo que conlleva un mayor uso del cambio de marchas.

Cada tipo de motor tiene su curva de par característica y de ella se puede deducir cual es su comportamiento.



**Figura 1.11. Ejemplos de curvas del par motor de motores de gasolina.**

## 1.5.4 La Potencia y tipos.

### 1.5.4.1 Potencia de un motor.

El combustible que se introduce en el interior de los cilindros, posee una energía química, que con la combustión se transforma en energía calorífica, de la cual una parte es convertida en trabajo mecánico. Este trabajo es el producto de una fuerza por un espacio recorrido bajo la aplicación de la misma.

$$P = \frac{W}{t} \quad (\text{Ec. 1.4})$$

La potencia desarrollada por un motor depende de sus características constructivas, así como de una serie de factores que inciden en el proceso de transformación de la energía.

- Relación de compresión y grado de calidad del ciclo.
- Grado de llenado de los cilindros.
- Cilindrada.
- Número de cilindros.
- Número de revoluciones.

La unidad de potencia es el Wat (vatios).

También suele utilizarse como unidad de potencia el caballo (CV o bien HP)

1 CV = 0,7355 KW <sup>(8)</sup>.

### 1.5.4.2 Tipos de potencia.

Las diferentes clases de potencia a considerar son:

- Potencia efectiva.
- Potencia fiscal.
- Potencia másica.
- Potencia específica.

---

(8) Manual CEAC del Automóvil, pág 55, Editorial CEAC S.A., España, 2003.

**1.5.4.2.1 Potencia efectiva de un motor:** Es la que realmente tiene en un banco de pruebas, pero para poder comparar motores entre sí, en qué condiciones se ha medido cada uno. Hay dos sistemas principales de medir la potencia: el SAE y el DIN.

En el SAE (de origen norteamericano) la potencia con el motor libre de accesorios que consuman potencia, esto es: sin alternador, ventilador, sin bomba de agua, sin filtro de aire, sin silenciador de escape, etc.

En cambio, en el DIN (de origen alemán) la prueba se realiza con el motor al completo. Lógicamente, la potencia de un mismo motor resulta mayor con las normas SAE que con las DIN. Cuando en los catálogos y manuales se hace la mención de la potencia, esta suele venir acompañada de las siglas de la norma.

**1.5.4.2.2 Potencia fiscal:** Es la que resulta de aplicar la fórmula:

$$P = 0,08 \cdot 0,785 \cdot D^2 \cdot R^{0,6} \cdot N \quad (\text{Ec. 1.5})$$

Donde:

P: Potencia fiscal expresada en caballos fiscales

D: Diámetro de los cilindros expresado en cm.

R: Carrera también en cm.

N: Número de cilindros.

Como se aprecia, ni siquiera entra en la fórmula el número de revoluciones, con lo que resulta una forma arbitraria de determinar la potencia.

Esta fórmula es la que se aplica en España para determinar el impuesto sobre la circulación de vehículos. De ello se deriva que dos automóviles de diferente potencia real puedan tener una misma potencia fiscal (por ejemplo, un motor diesel de gran cilindrada pero con 65 CV de potencia y uno de gasolina de una cilindrada similar pero con 150 CV).

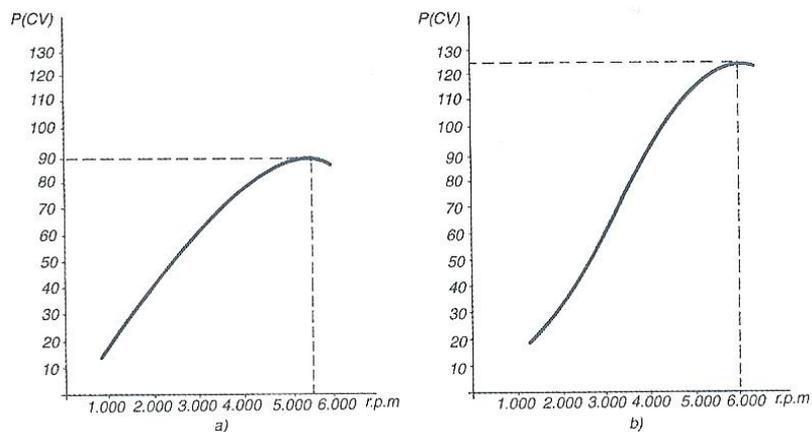
**1.5.4.2.3 Potencia másica:** Es la relación existente entre el peso del motor y su potencia efectiva. Se trata de una potencia relativa que da una idea del aprovechamiento mecánico. Se expresa en kg/CV.

**1.5.4.2.4 Potencia específica:** Es el número de CV que se obtienen por cada litro de cilindrada. Se trata de otra potencia relativa.

### 1.5.5 Curva de Potencia.

Las variaciones de la potencia con el número de revoluciones se pueden representar mediante una curva en un diagrama de dos ejes: el vertical para las potencias y el horizontal para las RPM.

En la figura 1.12 se representan dos curvas de potencia diferenciadas: una corresponde a un motor con el par muy constante (Figura 1.12 a) y la otra es propia de un motor con el máximo del par a revoluciones más bien altas (Figura 1.12 b). El motor con una curva como la figura 1.12 (a), tendrá una potencia moderada, aún siendo de gran cilindrada, pero resulta un motor muy cómodo, con poco uso del cambio de marchas; mientras el de la curva figura 1.12 (b), tendrá una potencia máxima mayor que el anterior, pero para obtener de él un buen rendimiento es necesario llevarlo muy acelerado, y para eso hay que utilizar el cambio con frecuencia.



**Figura 1.12. a) de un motor con el par muy constante.**

**b) de un motor con el par máximo a altas revoluciones.**

### 1.5.6 Consumo Específico.

El consumo específico es la cantidad de combustible consumido por cada unidad de trabajo desarrollado por el motor. Se expresa por Cs y se mide en g/CV.h (gramos por caballo vapor hora) y en g/kW.h (gramos por kilovatio hora).

1 CV es igual a 75 kgm/s. Si se multiplica por 3600 segundos que tiene una

hora, se tiene que  $1 \text{ CV.h} = 270000 \text{ kgm}^{(9)}$ .

Por otro lado, se tiene que  $1 \text{ kW}$  son unos  $102 \text{ kgm/s}$ , que multiplicándolo por  $3600 \text{ dan}$ .

$1 \text{ kW.h} = 367200 \text{ kgm}$ .

La curva del consumo específico es la que completa el conjunto de las curvas características de un motor, estando también en relación con las RPM.

Los valores en altura de la curva se trasladan al eje vertical del diagrama para obtener los consumos; los valores del eje horizontal corresponden a las RPM.

Teóricamente, el consumo específico debería ser igual a cualquier número de revoluciones, siendo independiente de la cilindrada del motor. Si un motor gira más deprisa consume más combustible, pero también proporciona mayor cantidad de trabajo, igualmente si un motor tiene mayor cilindrada gasta más, pero también es más potente.

En la práctica esto no es así, sino que la riqueza de la mezcla varía según las necesidades del motor. En la figura 1.13 se representa una curva de consumo específico genérica para un motor de gasolina.

A un bajo número de revoluciones, debido a la poca depresión que se origina en la admisión, el llenado del cilindro es menor y los restos de gases de escape mayores. En este ambiente, es necesario enriquecer la mezcla para aprovechar el aire, y como el trabajo producido es pequeño, el consumo específico es alto.

A velocidades de giro medias (velocidad de cruce del automóvil), la riqueza de la mezcla se puede reducir por debajo de la relación estequiométrica o teórica, ya que el motor trabaja en buenas condiciones con esfuerzos moderados. El consumo específico disminuye.

A altas revoluciones, las explosiones se suceden con mucha rapidez, y pese a la refrigeración del motor, la temperatura en la cámara de explosión se eleva en exceso, pudiendo afectar a las piezas que trabajan en peores condiciones de refrigeración, como la bujía, la válvula de escape y zonas cercanas a ellas. Una mezcla rica arde con menor temperatura, protegiendo las piezas citadas y además preserva de la detonación.

---

(9) Manual CEAC del Automóvil, pág 56, Editorial CEAC S.A., España, 2003.

Si a estas circunstancias se añade que la relación aire-gasolina de máxima potencia es un poco superior a la teórica, no es de extrañar que se aumente la proporción de gasolina y el consumo específico se eleve en esta zona de la curva. Esta es una de las causas de que los coches, a grandes velocidades, tengan consumos mayores.

En el consumo real de un automóvil, que se suele expresar en litros por cada 100 km (l/100 km), influyen, además del consumo específico, las resistencias a la marcha, el peso del vehículo, la forma de conducir, el trazado de la carretera, etc.

A mayor velocidad, el trabajo suministrado por el motor es mayor y también el gasto.

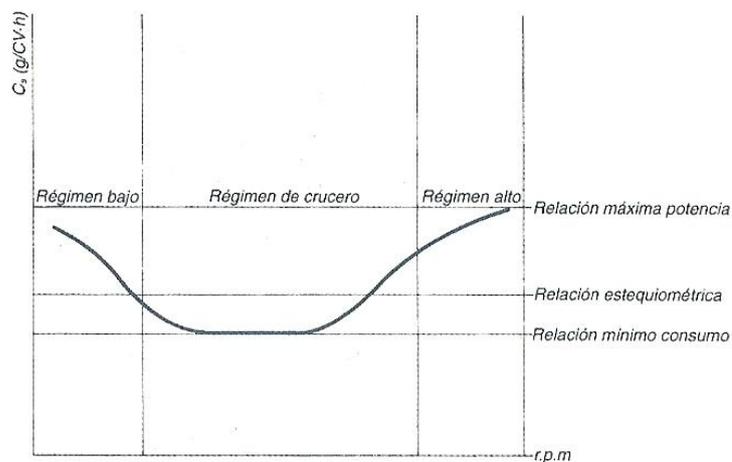


Figura 1.13. Ejemplo de curva del consumo específico de un motor de gasolina.

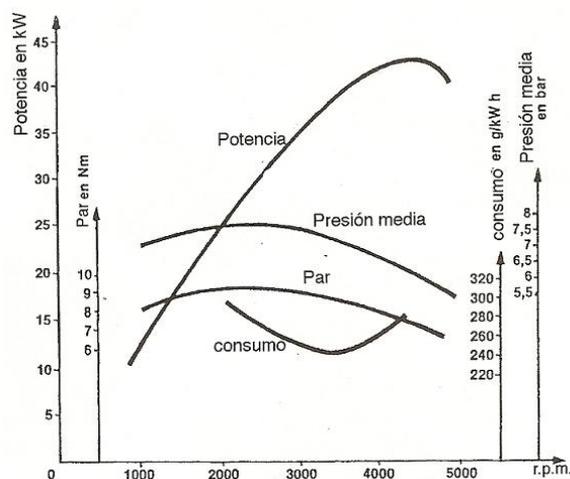
### 1.5.7 Gráficos de Curvas Características.

En la figura 1.14 están representados las gráficas completas con las tres curvas características de un motor de gasolina (par, potencia y consumo de gasolina).

Para conocer las variaciones de potencia, par motor y consumo de combustible en cada régimen del motor se utilizan las curvas características del motor.

Estos gráficos aparecen con frecuencia en los manuales de reparación, y sirven para comparar con ellos los resultados de las pruebas efectuadas en los bancos medidores de potencia o frenos dinamométricos en condiciones de

máxima alimentación, es decir, a plena carga, y por las diferencias apreciadas se sabe el estado del motor o la eficiencia de las reparaciones efectuadas <sup>(10)</sup>.



**Figura 1.14. Ejemplo de gráficas completas con las tres curvas características de un motor de gasolina (par, potencia y consumo de gasolina).**

(10) Manual CEAC del Automóvil, pág 58, Editorial CEAC S.A., España, 2003.

## **II. CAPÍTULO**

### **ANÁLISIS DE ELEMENTOS ELECTRÓNICOS.**

#### **2.1 SENSORES.**

##### **2.1.1 Introducción.**

El sensor (también llamado sonda o transmisor) convierte una magnitud física (temperatura, revoluciones del motor, etc.) o química (gases de escape, calidad de aire, etc.) que generalmente no son señales eléctricas, en una magnitud eléctrica que pueda ser entendida por la unidad de control. La señal eléctrica de salida del sensor no es considerada solo como una corriente o una tensión, sino también se consideran las amplitudes de corriente y tensión, la frecuencia, el periodo, la fase o asimismo la duración de impulso de una oscilación eléctrica, así como los parámetros eléctricos "resistencia", "capacidad" e "inductancia".

El sensor se puede presentar como un "sensor elemental" o un "sensor integrado" este último estaría compuesto del sensor propiamente dicho más la parte que trataría las señales para hacerlas comprensibles por la unidad de control. La parte que trata las señales generadas por el sensor (considerada como circuitos de adaptación), se encarga en general de dar a las señales de los sensores la forma normalizada necesaria para ser interpretada por la unidad de control.

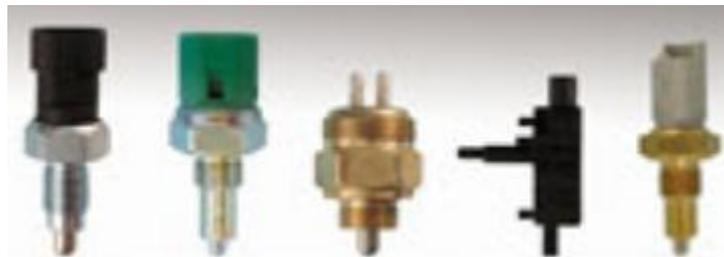
En su calidad de elementos periféricos, los sensores y actuadores constituyen las interfaces entre el vehículo con sus complejas funciones de transmisión, frenado, tren de rodaje, carrocería, conducción y navegación, y la unidad electrónica de controladora digital como unidad de tratamiento. Un circuito de

adaptación se encarga en general, de dar a las señales la forma normalizada necesaria para la unidad de control ECU (Electronic Control Unit).

Existen un gran número de circuitos de adaptación integrados, a la medida de los sensores y ajustados a los vehículos respectivos.

### **2.1.2 Sensores de Retro.**

Los trompos de reversa se los puede encontrar en varios modelos en el mercado automotriz tal como se muestra en la figura 2.1. Estos funcionan como un pulsador normalmente cerrado, tienen un vástago de pulsación, el cual al ser presionados pasan a estar en circuito abierto es decir ya no conduce corriente. Cuando están enroscados en la caja de cambios y la marcha de reversa es enclavada este vástago no es presionado por lo que se halla en estado de conductividad; mientras que cuando cualquier otra marcha es enclavada el vástago está presionado en esta de no conductividad.



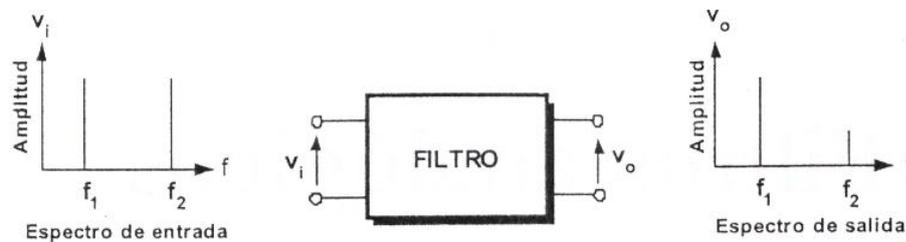
**Figura 2.1. Modelos de interruptores de retro**

## **2.2 FILTRADO DE SEÑALES.**

### **2.2.1 Filtros.**

Un filtro eléctrico es un elemento capaz de atenuar determinada frecuencia o gama de frecuencias de una señal eléctrica que pasa a través de él, pudiendo modificar tanto su amplitud como su fase. Se diseña un sistema que suprime una banda de frecuencia específica pero deja otra banda inalterada. Así, el espectro de frecuencia es "filtrado" para obtener el espectro requerido.

En la figura 2.2 se presenta el concepto básico de un filtro de manera gráfica; la componente del espectro no deseada de frecuencia  $f_2$ , al pasar por el filtro se ve atenuada, mientras que la frecuencia deseada  $f_1$  mantiene su amplitud.



**Figura 2.2. Concepto de filtro eléctrico.**

Un filtro se caracteriza por una función de transferencia  $H(s)$ . Esta función se puede llegar a expresar matemáticamente en forma de fracción mediante las transformaciones en frecuencia adecuadas.

$$H(s) = \frac{v_o(s)}{v_i(s)} \quad \text{Ec. (2.1)}$$

### 2.2.2 Tipos de filtro.

Los filtros se clasifican en los distintos grupos que a continuación se indica.

1) Según el tipo de señal procesada:

**Filtro analógico:** Diseñado para el tratamiento de señales analógicas. Según la tecnología empleada en su realización pueden ser:

- **Filtro pasivo:** Es el constituido únicamente por componentes pasivos como condensadores, bobinas y resistencias. En la tabla II.1 se muestran ejemplos de filtros pasivos RC.

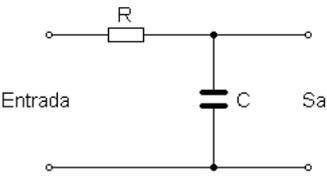
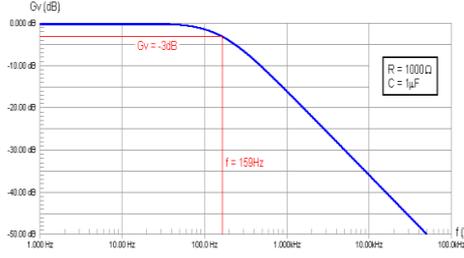
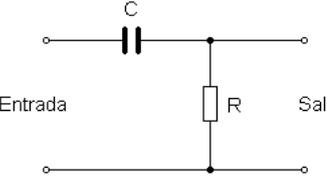
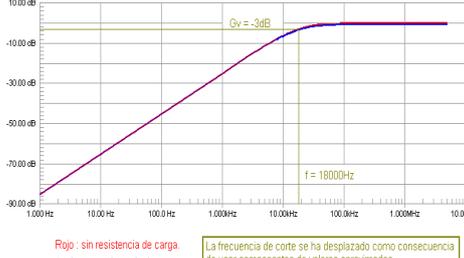
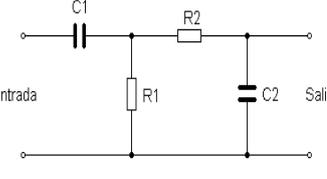
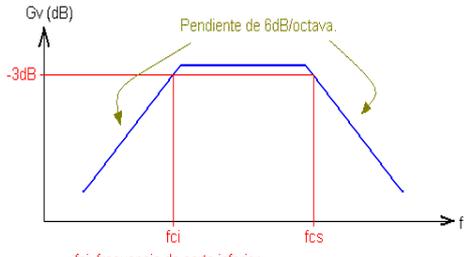
- **Filtro activo:** Es aquel que puede presentar ganancia en toda o parte de la señal de salida respecto a la de entrada. En su implementación se combinan elementos activos y pasivos; siendo frecuente el uso de amplificadores operacionales, que permite obtener resonancia, una buena adaptación de

impedancias entre etapas y condiciona su funcionamiento dinámico al tener limitaciones de ancho de banda.

- **Filtros de capacidad conmutada;** La señal de entrada es muestreada y procesada en un tiempo discreto. Las resistencias se reemplazan por interruptores MOS y condensadores integrados en el propio chip. La frecuencia de corte está determinada y es proporcional a la frecuencia de un reloj externo.

**Filtro digital:** Diseñado para el tratamiento de señales digitales. La señal analógica es cuantificada previamente mediante un convertidor analógico-digital y posteriormente procesada mediante algoritmos.

Tabla II.1 Ejemplos de filtros pasivos RC.

Tipo de filtro.	Respuesta frecuencial.	Ecuación.
<p>Paso bajo.</p>  <p>Entrada <span style="margin-left: 150px;"></span> Salida</p>		$f_c = \frac{1}{2 \cdot \pi \cdot R \cdot C}$
<p>Paso alto.</p>  <p>Entrada <span style="margin-left: 150px;"></span> Salida</p>	 <p>Rojos: sin resistencia de carga. Azul: con resistencia de carga.</p> <p>La frecuencia de corte se ha desplazado como consecuencia de usar componentes de valores aproximados.</p>	$f_c = \frac{R}{2 \cdot \pi \cdot L}$
<p>Paso banda.</p>  <p>Entrada <span style="margin-left: 150px;"></span> Salida</p>	 <p>Pendiente de 6dB/octava.</p> <p>-3dB</p> <p>fci frecuencia de corte inferior. fcs: frecuencia de corte superior.</p>	$C_2 = \frac{1}{2 \cdot \pi \cdot f_{cs} \cdot X_{C2}}$

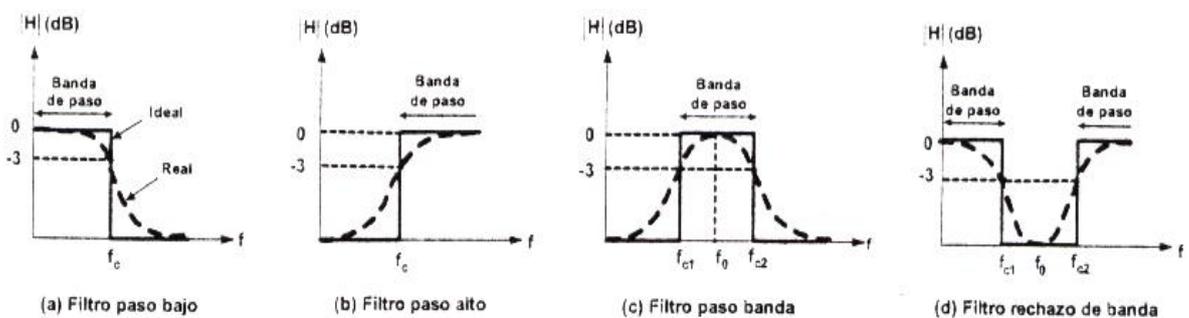
2) Según la función que lleven a cabo:

**Filtro paso bajo:** Dejan pasar todas las frecuencias por debajo de una determinada frecuencia de corte,  $f_c$ , atenuando las frecuencias por encima de dicha frecuencia de corte. La figura (2.3 a), muestra la curva de la ganancia de esta función en función de la frecuencia. Para  $f=f_c$ , la ganancia cae 3 dB, con respecto a la ganancia máxima, que por comodidad se ha supuesto unidad (es decir, 0 dB).

**Filtro paso alto:** Es el que permite el paso de frecuencias desde una frecuencia de corte  $f_c$  determinada hacia arriba, sin que exista un límite superior especificado, figura (2.3 b).

**Filtro paso banda:** Son aquellos que permiten el paso de componentes frecuenciales contenidos en un determinado rango de frecuencias, comprendido entre una frecuencia de corte superior y otra inferior. Es decir, dejan pasar aquellas frecuencias comprendidas entre la frecuencia de corte inferior  $f_{c1}$ , y la frecuencia de corte superior,  $f_{c2}$ , atenuando las demás, figura (2.3 c). La frecuencia  $f_0$  se denomina frecuencia central.

**Filtro elimina banda:** También llamado filtro rechaza banda, es el que dificulta el paso de componentes frecuenciales contenidos en un determinado rango de frecuencias, comprendido entre una frecuencia de corte inferior ( $f_{c1}$ ) y superior ( $f_{c2}$ ), dejando pasar las otras (figura 2.3 d).



**Figura 2.3. Funciones de transferencia ideales y reales de los filtros.**

## **2.3 MEMORIAS PROGRAMABLES.**

Actualmente las modernas técnicas de circuitos integrados permiten combinar miles e incluso millones de puertas dentro de un solo encapsulado. Esto ha llevado a la fabricación de diseños más complejos como los dispositivos lógicos programables, memorias y microprocesadores, que proporcionan dentro de un solo chip circuitos que requieren gran cantidad de componentes discretos.

Las memorias son dispositivos de almacenamiento de datos binarios de largo o corto plazo. La memoria de acceso aleatorio (RAM, Random Access Memory) almacena datos temporalmente; la memoria de sólo lectura (ROM, Read Only Memory) los guarda de manera permanente. La ROM forma parte del grupo de componentes llamados dispositivos lógicos programables (PLD, Programmable Logic Devices), que emplean la información almacenada para definir circuitos lógicos. Estos dispositivos son capaces de proveer el medio físico para almacenar información, y aunque ésta es su tarea fundamental (más del 90 % de las memorias se dedican a este fin) también se pueden utilizar para la implementación de circuitos combinatoriales y pueden sustituir la mayor parte de la lógica de un sistema.

Se puede considerar una memoria como un conjunto de  $M$  registros de  $N$  bits, cada uno de ellos. Estos registros ocupan las posiciones desde el valor 0 hasta  $M-1$ . Para acceder a cada registro es necesaria una lógica de selección, en general para cada registro se pueden realizar procesos de lectura y de escritura.

### **2.3.1 Ventajas de la EEPROM.**

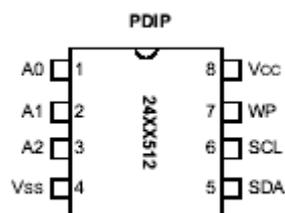
- Las palabras almacenadas en memoria se pueden borrar de forma individual.
- Para borrar la información no se requiere luz ultravioleta.
- Las memorias EEPROM no requieren programador.
- De manera individual se puede borrar y reprogramar eléctricamente grupos de caracteres o palabras en el arreglo de la memoria.
- Para reescribir no se necesita hacer un borrado previo.

### 2.3.2 Memoria EEPROM 24LC512.

La memoria EEPROM es programable y borrable eléctricamente y su nombre proviene de la sigla en inglés Electrical Erasable Programmable Read Only Memory. Actualmente estas memorias se construyen con transistores de tecnología MOS (Metal Oxide Silice) y MNOS (Metal Nitride-Oxide Silicon). La posibilidad de programar y borrar las memorias a nivel de bytes supone una gran flexibilidad, pero también una celda de memoria más compleja. Además del transistor de puerta flotante, es preciso un segundo transistor de selección. El tener 2 transistores por celda hace que las memorias EEPROM sean de baja densidad y mayor costo. La programación requiere de tiempos que oscilan entre  $157 \mu s$  y  $625 \mu s$  por byte <sup>(11)</sup>.

Las principales características de la memoria 24LC512 son:

- Capacidad de Almacenamiento: 512 Kbits (64 Kbytes).
- Voltaje de Operación: 2,5 – 5,5 V.
- Control: %1010dd0.
- Tamaño de dirección: 2 bytes.
- Comunicación: compatibilidad I2C.
- Tiempo de ciclo de escritura: 5ms. máximo.



**Figura 2.4. Memoria 24LC512**

En la figura 2.4 se muestra el esquema de descripción de pines de la memoria EEPROM.

---

(11) Datasheet Memoria Eeprom 24LC512

Donde:

CE = CHIP ENABLE: Permite Activar el Circuito Integrado

OE = OUTPUT ENABLE: Permite Activar la Salida del Bus De Datos.

WE = WRITE ENABLE: Permite Activar la escritura del Bus De Datos.

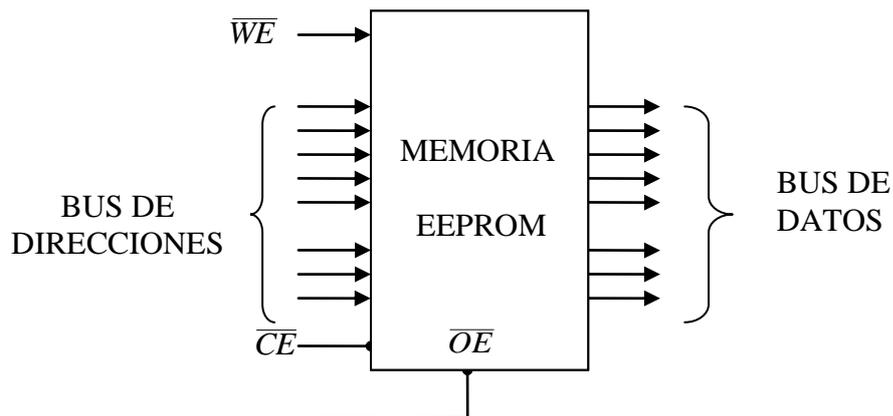


Figura 2.5. Descripción de pines de la EEPROM.

## 2.4 El RTC DS1307.

El DS1307 es un dispositivo de 8 pines al que se le conecta un cristal de cuarzo estándar, de bajo costo, a 32.768kHz entre los pines 1 y 2 para proveer tiempo base exacto. Es un reloj de tiempo real exacto, el cual automáticamente, mantiene el tiempo y la fecha actual, incluyendo compensación para meses con menos de 31 días y saltos de año. La figura 2.6 muestra la distribución de pines del RTC.

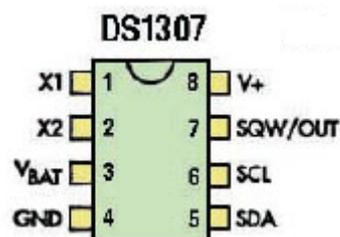


Figura 2.6. Reloj en tiempo real.

Además se le puede conectar al pin3, baterías de respaldo de 3 voltios, asegurando que se mantendrá el tiempo a la fecha aunque esté desconectada la fuente de tensión del circuito principal. El circuito integrado automáticamente detecta que se ha removido la energía en el circuito principal y se conectan las baterías de respaldo cuando es requerido. La batería de respaldo puede durar hasta 10 años y se coloca en la misma base de circuito impreso. El circuito integrado DS1307 además tiene dos características interesantes; el pin 7 es una salida de colector abierto, que puede ser programada para hacer “flash” cada 1Hz es decir cada segundo; internamente el circuito integrado también tiene 56 bytes de memoria RAM para propósito general.

## 2.5 El CI 4093.

Este circuito integrado consta de 4 compuertas NAND del tipo Schmitt-Trigger (fig. 2.7) las que nos ayudarán a obtener una señal de frecuencia perfectamente cuadrada. Para comprender las características de su funcionamiento, tome imaginariamente un Schmitt-Trigger que tenga su salida en nivel bajo, y aplique a su entrada un pulso cualquiera, cuyo voltaje esté subiendo muy lentamente, algo así como una onda senoidal. Al comienzo de haber aplicado la señal de entrada, no hay ningún cambio apreciable en el estado bajo de su salida; pero, llegado cierto momento, la salida del Schmitt-Trigger se invierte al estado alto, y continúa así aunque el nivel de voltaje en el pulso siga aumentando. Al punto o valor del voltaje de la señal ascendente que produjo el cambio en la salida, se le conoce como POSITIVE GOING THRESHOLD (umbral en el sentido positivo).

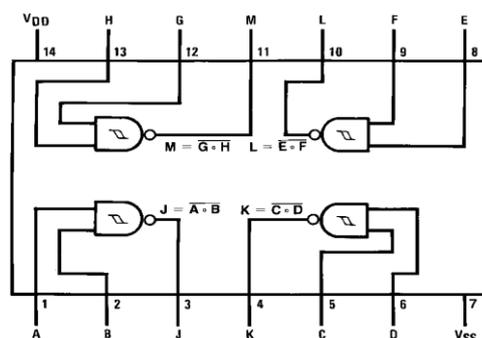


Figura 2.7. El circuito integrado 4093.

Suponga ahora que la señal comienza a bajar, a disminuir su voltaje. Llegará el instante en el cual la salida del Schmitt-Trigger cambiará abruptamente al nivel bajo original, punto al que se ha denominado NEGATIVE GOING THRESHOLD (umbral en el sentido negativo). Los puntos o niveles de umbral son distintos para el sentido positivo y para el sentido negativo de la señal gatilladora, y a su diferencia se le conoce como HISTÉRESIS.

## **2.6 COMUNICACIÓN SERIAL.**

### **2.6.1 Conceptualización.**

Existen dos formas de realizar una comunicación binaria, la paralela y la serial. La comunicación paralela tiene la ventaja de que la transferencia de datos es más rápida, pero el inconveniente es que se necesita un cable por cada bit de dato, lo que encarece y dificulta el diseño de las placas, otro inconveniente es la capacitancia que genera los conductores por lo que la transmisión se vuelve defectuosa a partir de unos pocos metros.

La comunicación serial en cambio es mucho más lenta debido a que transmite bit por bit pero tiene la ventaja de necesitar menor cantidad de hilos, y además se puede extender la comunicación a mayor distancia, por ejemplo; en la norma RS232 a 15 mts., en la norma RS422/485 a 1200 mts. y utilizando un modem, pues a cualquier parte del mundo.

### **2.6.2 Tipos de Comunicaciones Seriales.**

Existen dos tipos de comunicaciones seriales: la síncrona y asíncrona.

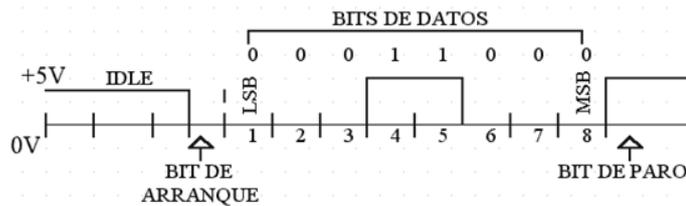
**2.6.2.1 Comunicación serial síncrona:** Además de una línea sobre la cual se transmitirán los datos se necesita de una línea la cual contendrá los pulsos de reloj que indicaran cuando un dato es válido.

Ejemplos de este tipo de comunicación son:

- I2C.
- ONE WIRE.
- SPI.

**2.6.2.2 Comunicación serial asíncrona:** No son necesarios los pulsos de reloj. La duración de cada bit está determinada por la velocidad con la cual se realiza la transferencia de datos.

La figura 2.8 muestra la estructura de un carácter que se transmite en forma serial asíncrona.



**Figura 2.8. Estructura de un carácter que se transmite en forma serial asíncrona.**

Normalmente cuando no se realiza ninguna transferencia de datos, la línea del transmisor se encuentra en estado de (IDLE) esto quiere decir en estado alto.

Para iniciar la transmisión de datos, el transmisor coloca esta línea en bajo durante determinado tiempo, lo cual se le conoce como bit de arranque (start bit) y a continuación empieza a transmitir con un intervalo de tiempo los bits correspondientes al dato, empezando siempre por el BIT menos significativo (LSB), y terminando con el BIT más significativo.

Si el receptor no está sincronizado con el transmisor, este desconoce cuándo se van a recibir los datos.

Por lo tanto el transmisor y el receptor deberán tener los mismos parámetros de velocidad, paridad, número de bits del dato transmitido y de BIT de parada.

En los circuitos digitales, cuyas distancias son relativamente cortas, se pueden manejar transmisiones en niveles lógicos TTL (0-5V), pero cuando las distancias aumentan, estas señales tienden a distorsionarse debido al efecto capacitivo de los conductores y su resistencia eléctrica. El efecto se incrementa a medida que se incrementa la velocidad de la transmisión.

Todo esto origina que los datos recibidos no sean igual a los datos transmitidos, por lo que no se puede permitir la transferencia de datos.

Una de las soluciones más lógicas es aumentar los márgenes de voltaje con que se transmiten los datos, de tal manera que las perturbaciones a causa de la línea se puede corregir.

### **2.6.3 Modos de Transmisión de Datos.**

Los modos de transmisión de datos se dividen en cuatro tipos y estos son:

**2.6.3.1 Simplex:** Es la transmisión que puede ocurrir en un solo sentido; sea sólo para recibir o solo para transmitir. Una ubicación puede ser un transmisor o un receptor, pero no ambos a la vez, un ejemplo claro es la radiodifusión, en donde la estación es el transmisor y los radios son los receptores.

**2.6.3.2 Half-duplex:** Se refiere a la transmisión que puede ocurrir en ambos sentidos pero no al mismo tiempo, en donde una ubicación puede ser un transmisor y un receptor, pero no los dos al mismo tiempo, un ejemplo son los llamados radios walking talking, en donde un operador presiona el botón y habla, luego suelta el botón y el otro usuario presiona el botón para contestar.

**2.6.3.3 Full-duplex:** Es la transmisión que puede ocurrir en ambos sentidos y al mismo tiempo, también se los conoce con el nombre de líneas simultaneas de doble sentido, una ubicación puede transmitir y recibir simultáneamente, siempre y cuando la estación a la que está transmitiendo también sea la estación de la cual está recibiendo un ejemplo es la telefonía móvil.

**2.6.3.4 Full/full-duplex:** Con este modo de transmisión es posible transmitir y recibir simultáneamente, pero no necesariamente entre las dos ubicaciones, es decir una estación puede transmitir a una segunda estación y recibir de una tercera estación al mismo tiempo. Esta transmisión se utiliza casi exclusivamente con circuitos de comunicación de datos.

### **2.6.4 Descripción de la norma RS232.**

Ante la gran variedad de equipos, sistemas y protocolos que existen surgió la necesidad de un acuerdo que permitiera a los equipos de varios fabricantes comunicarse entre sí. La EIA (Electronics Industry Association) elaboro la norma RS232, la cual define la interface mecánica, los pines, las señales y los protocolos que debe cumplir la comunicación serial.

La norma RS232 resuelve tres aspectos en la comunicación que se establece entre el DTE (Equipo Terminal de Datos), por ejemplo un PC y el DCE (Equipo para la comunicación de datos), por ejemplo un ratón.

**Características eléctricas de la señal:** Se establece que la longitud máxima entre el DTE y el DCE no debe ser superior a los 15 metros y la velocidad máxima de transmisión es de 20.000 bps<sup>(12)</sup>. Esta norma establece una señalización eléctrica bipolar:

- Un "1" lógico es un voltaje comprendido entre  $-5V$  y  $-15V$  en el transmisor y entre  $-3V$  y  $-25V$  en el receptor.
- Un "0" lógico es un voltaje comprendido entre  $+5V$  y  $+15V$  en el transmisor y entre  $+3V$  y  $+25V$  en el receptor.

**Características mecánicas de los conectores:** Se utiliza un conector de 25 pines (DB25), o de 9 pines (DB9), donde el identificador macho identifica al DTE y el conector hembra al DCE.

**Descripción funcional de las señales usadas:** Las señales están básicamente divididas en dos grupos:

- Señales primarias, utilizadas para la transferencia de datos.
- Señales secundarias, utilizadas para el control de la información que será transferida.

La norma RS232 está definida tanto para la transmisión síncrona como para la asíncrona, pero cuando se utiliza esta última se selecciona el conector de 25 pines.

## **2.6.5 Descripción del Bus I2C.**

### **2.6.5.1 Introducción.**

El bus I2C, es un estándar que facilita la comunicación entre microcontroladores, memorias y otros dispositivos con cierto nivel de "inteligencia", sólo requiere de dos líneas de señal y un común o masa.

---

(12) Datasheet Memoria Eeprom 24ILC512

Permite el intercambio de información entre muchos dispositivos a una velocidad aceptable, de unos 100 Kbits por segundo, aunque hay casos especiales en los que el reloj llega hasta los 3,4 Mhz.

Una característica del bus I2C es que con las mismas dos líneas (SDA y SCL) se puede comunicar con varios dispositivos, sin la necesidad de utilizar otros pines del microcontrolador que permiten ahorrar entradas.

La metodología de comunicación de datos del bus I2C es en serie y sincrónica. Una de las señales del bus marca el tiempo (pulsos de reloj) y la otra se utiliza para intercambiar datos.

#### **2.6.5.2 Descripción de las señales.**

**SCL (System Clock):** Es la línea de los pulsos de reloj que sincronizan el sistema.

**SDA (System Data):** Es la línea por la que se mueven los datos entre los dispositivos.

**GND (Masa):** Es el común de la interconexión entre todos los dispositivos "enganchados" al bus.

#### **2.6.5.3 Protocolo de comunicación del bus I2C.**

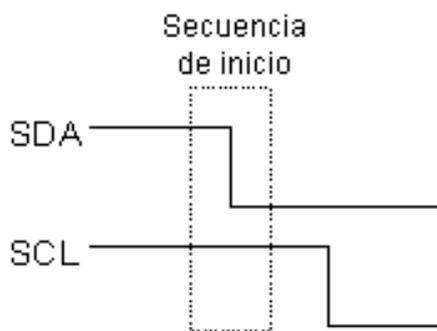
Como se dijo anteriormente el bus I2C permite la comunicación entre dispositivos como: microcontroladores, memorias EEPROM, RTC's (Reloj en Tiempo Real), utilizando una comunicación **serial síncrona** por medio de dos líneas una de datos (SDA) y una de reloj (SCL).

Habiendo varios dispositivos conectados sobre el bus, es lógico que para establecer una comunicación a través de él se deba respetar un protocolo. Existen dispositivos maestros y dispositivos esclavos. Sólo los dispositivos maestros pueden iniciar una comunicación.

Lo primero que ocurre en un bus I2C es que el dispositivo maestro (microcontrolador) envía una secuencia de inicio. Esto alerta a los dispositivos

esclavos (memoria EEPROM, reloj en tiempo real, etc.), poniéndolos a la espera de una transacción. Éstos quedan atentos para ver si se trata de una solicitud para ellos.

La condición inicial, de bus libre, es cuando ambas señales están en estado lógico alto. En este estado cualquier dispositivo maestro puede ocuparlo, estableciendo la condición de inicio (start). Esta condición se presenta cuando un dispositivo maestro pone en estado bajo la línea de datos (SDA), pero dejando en alto la línea de reloj (SCL). En la figura 2.9 se muestra el estado de las señales cuando se inicia la comunicación.



**Figura 2.9. Inicio de la comunicación I2C.**

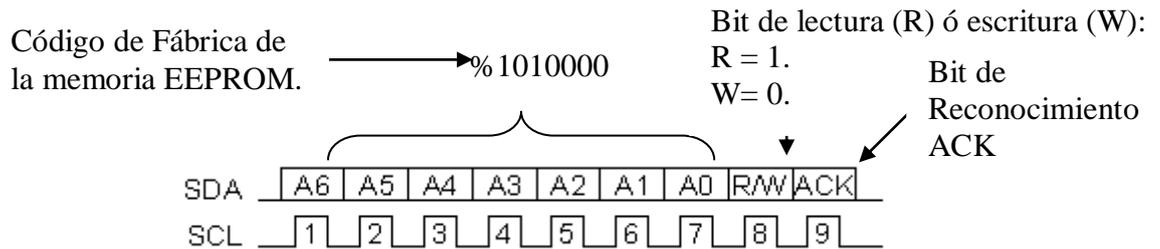
El primer byte que se transmite luego de la condición de inicio contiene siete bits que componen la dirección del dispositivo que se desea seleccionar, y un octavo bit que corresponde a la operación que se quiere realizar con él (lectura o escritura), en bajo (0 lógico) escritura y en alto (1 lógico) lectura.

Si el dispositivo cuya dirección corresponde a la que se indica en los siete bits (A0-A6) está presente en el bus, éste contesta con un bit en bajo, ubicado inmediatamente luego del octavo bit que ha enviado el dispositivo maestro.

Este bit de reconocimiento (ACK del inglés Acknowledge), en bajo le indica al dispositivo maestro que el esclavo reconoce la solicitud y está en condiciones de comunicarse.

Aquí la comunicación se establece en firme y comienza el intercambio de información entre los dispositivos.

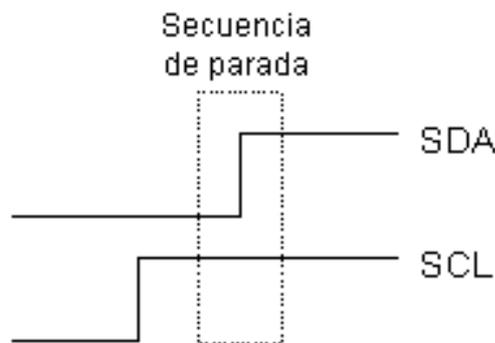
En la figura 2.10 se ilustra la utilización de una memoria EEPROM.



**Figura 2.10. La escritura datos en la memoria EEPROM.**

Si el bit de lectura/escritura (R/W) fue puesto en esta comunicación a nivel lógico bajo (escritura), el dispositivo maestro envía datos al dispositivo esclavo. Esto se mantiene mientras continúe recibiendo señales de reconocimiento, y el contacto concluye cuando se hayan transmitido todos los datos.

En la figura 2.11 se muestra el estado de las señales al finalizar la comunicación.



**Figura 2.11. Fin de la comunicación I2C.**

Se necesita solo 2 líneas para transmitir y recibir datos, estos son: para datos (SDA) y para la señal de reloj (SCL), esta forma de comunicación utiliza una

sincronía con un tren de pulsos que viaja en la línea SCL, de tal manera que en los flancos negativos se revisan los datos RX o TX. Su velocidad de transmisión puede ser de 100Kbits/seg. En el modo estándar, 400Kbits/seg. en el modo rápido y 3,4Mbits/seg. en alta velocidad. Cada dispositivo conectado al bus tiene un código de dirección seleccionable mediante software, por lo que existe una relación permanente Master/Slave. El Master es el dispositivo que inicia la transferencia en el bus y genera la señal de reloj (SCL), y el Slave es el dispositivo direccionado, sin embargo cada dispositivo reconocido por su código (dirección), puede operar como transmisor o receptor de datos; ya que la línea (SDA) es bidireccional.

### 2.6.6 El MAX232.

Este chip permite adaptar los niveles RS232 y TTL, permitiendo conectar un PC con un microcontrolador. Sólo es necesario este chip y 4 condensadores electrolíticos. El esquema se muestra en la figura 2.12.

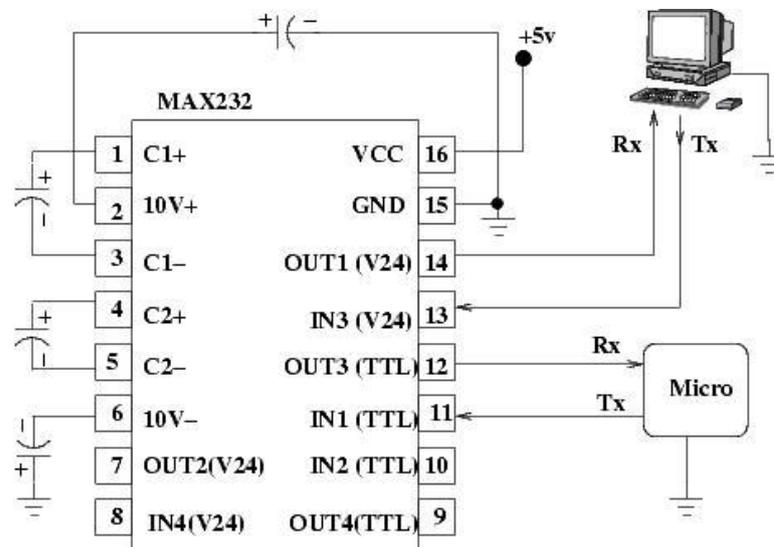


Figura 2.12. Esquema del MAX232.

En el MAX232 todos los condensadores deben ser de 1 microfaradio para llegar hasta 120 Kbps o de 100 nanofaradios para llegar hasta 64 Kbps. En la práctica para transmitir a 2400 bps los condensadores pueden ser de 10 microfaradios, 1 microfaradio o 0.1 microfaradios dan buenos resultados.

Sus características Eléctricas son:

- Vcc: de 4,5v a 5,5v.
- Consumo: 4 mA (15 mA con carga a la salida de 3 Kohm).
- Tensión de entrada máxima RS232: +/- 30v.
- Tensión de Salida RS232: +/- 15v.
- Tensión de salida típica de +/-8v con carga nominal de 5 Kohm en RS232.
- Las entradas se pueden dejar al aire.
- Entrada TTL al aire, se considera un "0" al invertirse en la salida.
- Entrada RS232 al aire, se considera un "1" al invertirse en la salida.
- Salida RS232: +/- 22 mA.

### 2.6.7 Conector DB9 del PC.

En los PCs hay conectores DB9 macho, de 9 pines, por el que se conectan los dispositivos al puerto serie. Los conectores hembra que se enchufan tienen una colocación de pines diferente, de manera que se conectan el pin 1 del macho con el pin 1 del hembra, el pin2 con el 2, etc. (Figura 2.13).

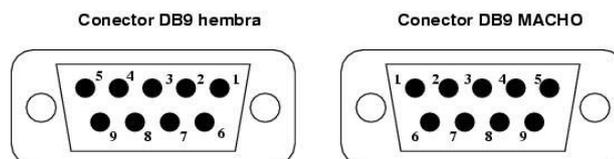


Figura 2.13. Conectores DB9 hembra y macho.

La información asociada a cada uno de los pines se representa en la tabla II.2.

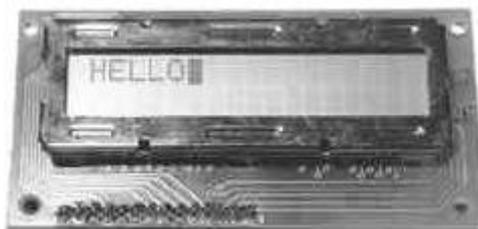
Tabla II.2. Descripción de pines del conector DB9.

Número de pin	Señal
1	DCD (Data Carrier Detect)

2	RX
3	TX
4	DTR (Data Terminal Ready)
5	GND
6	DSR (Data Sheet Ready)
7	RTS (Request To Send)
8	CTS (Clear To Send)
9	RI (Ring Indicator)

### 2.6.8 LCD y teclados.

Los módulos LCD son display de cristal líquido (Figura 2.14), son utilizados para mostrar mensajes al operario al estado de la máquina, o para dar instrucciones de manejo, mostrar valores, etc. El LCD permite la interface entre un dispositivo electrónico y su usuario, este puede mostrar cualquier carácter ASCII y consume mucho menos corriente que los displays de siete segmentos. Los LCD los podemos encontrar en algunas presentaciones tales como 2 líneas por 8 caracteres; 2 x16, 2x 20, 4x20, 4x40; estas presentaciones pueden tener (16 pines) o no backlight (14 pines).



**Figura 2.14. LCD.**

El LCD más popular es el 2x16 y su distribución de pines se representa en la tabla II.3.

**Tabla II.3. Descripción de pines de una LCD.**

Pin	Simb.	Descripción
1	Vss	Tierra de alimentación GND
2	Vdd	Alimentación de +5V CC
3	Vo	Ajuste del contraste de cristal liquido
4	RS	Selección del registro de control de datos
5	R/W	Lectura/ escritura en LCD
6	E	Habilitación
7	D0	Bit menos significativo(bus de datos bidireccional)
8	D1	
9	D2	
10	D3	
11	D4	
12	D5	
13	D6	
14	D7	Bit menos significativo(bus de datos bidireccional)
15	A	Alimentación de back light +3.5C 0 +5 V CC
16	K	Tierra GND de back Light

Los LCD se pueden conectar con el PIC con un bus de 4 u (bits, la diferencia está en el tiempo que se demora, pues la comunicación a 4 bits, primero envía los 4 bits más altos y luego los 4 bits más bajos, mientras que las de 8 bits envía todo al mismo tiempo, esto no es un inconveniente, si consideramos que la LCD trabaja en microsegundos, Pero la gran ventaja de hacer conexión a 4 bits, son los pocos cables que se deben conectar.

Los teclados matriciales son muy útiles para ingresar datos, ocupando una menor cantidad de pines a diferencia del uso de pulsadores individuales. Los teclados de 4 x 4 tiene la posibilidad de 16 combinaciones utilizando solo 8 pines, lo que nos deja ver que con esta matriz ahorramos un 50% de pines.

La estructura interna del teclado matricial se indica en la figura 2.15.

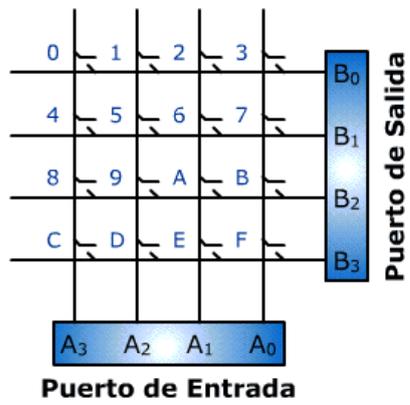


Figura 2.15. Estructura interna del teclado.

### III. CAPÍTULO

## DISEÑO Y DIAGRAMACIÓN DE CIRCUITOS ELECTRÓNICOS.

### 3.13 DEFINICIÓN DE ETAPAS DE FUNCIONAMIENTO DEL MÓDULO.

El módulo a desarrollar va a constar de las siguientes etapas:

- Inicialización y configuración.
- Obtención de número de revoluciones del motor.
- Obtención de la posición de marcha.
- Interpretación y procesamiento de los datos recolectados.
- Visualización de datos procesados.
- Interfaz con la PC para descarga de datos obtenidos.

#### 3.1.7 Inicialización y configuración.

En esta etapa, el módulo inicia su funcionamiento para luego permitir al usuario a través de la interface LCD, microcontrolador y teclado (Figura 3.1) realizar las configuraciones tales como ajuste de número de cilindros, numero de marchas del vehículo, hora y fecha del sistema, etc.

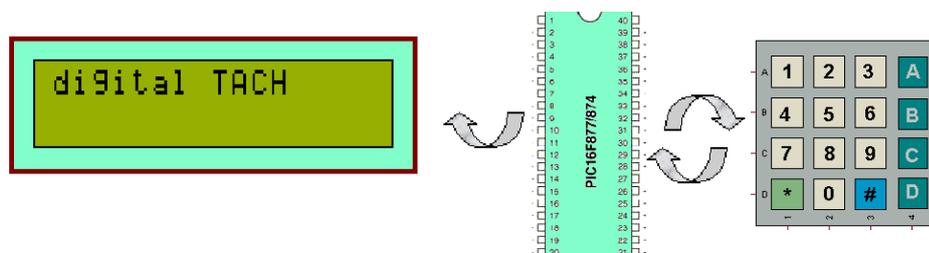


Figura 3.1. Interface LCD, microcontrolador y teclado.

### 3.1.8 Obtención de número de revoluciones del motor.

El módulo va a recolectar las revoluciones que tiene el motor a través del borne negativo de la bobina de encendido, este terminal pertenece al arrollamiento primario de la bobina el mismo que transmite en forma de impulso de corriente cada que el ruptor se abre y se cierra de acuerdo al giro que realiza el motor. La figura 3.2 muestra el esquema de recolección de número de revoluciones del motor.

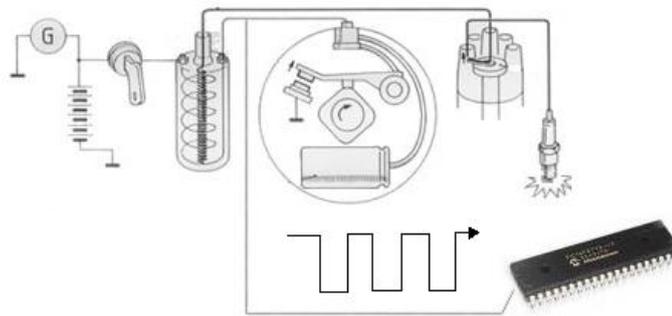


Figura 3.2. Esquema de recolección de número de revoluciones del motor.

### 3.1.9 Obtención de la posición de marcha.

Se recolectará el dato del número de marcha ubicando sensores de posición en la parte externa de la caja de cambios (Figura 3.3); estarán ubicados en un alojamiento por donde la palanca de la caja de cambios se desplazará en su funcionamiento normal. El dato recolectado será enviado al módulo para su respectivo procesamiento.

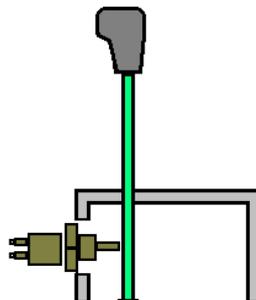


Figura 3.3. Ubicación de sensores en la caja de cambios.

### **3.1.10 Interpretación y procesamiento de los datos recolectados.**

Con el uso de un microcontrolador se recolectará los distintos datos mencionados anteriormente y se les dará el procesamiento de acuerdo a las necesidades que tenga el usuario, para luego interactuar con los dispositivos externos (teclado, display, shift lite) e internos (PIC esclavo, memoria EEPROM, reloj RTC, MAX 232).

### **3.1.11 Visualización de datos procesados.**

A través de la LCD y los displays (Figura 3.4) el usuario podrá observar los datos procesados tales como: revoluciones por minuto del motor (RPM), número de marcha del vehículo, hora y fecha de obtención de datos, y los distintos menús de configuración del módulo.



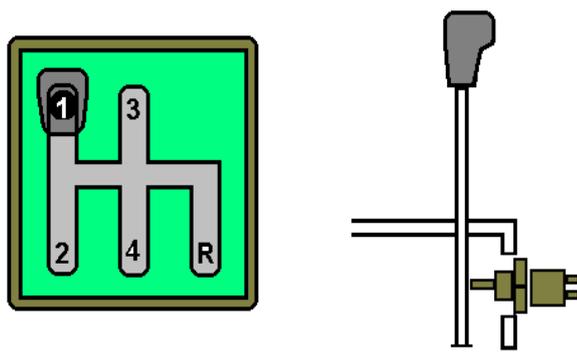
**Figura 3.4. LCD y displays.**

### **3.1.12 Interfaz con la PC para descarga de datos obtenidos.**

Los valores de RPM y posición de marcha almacenados en la memoria EEPROM del módulo podrán ser transferidos a través de un protocolo de comunicación hacia un Computador Personal, en donde mediante el uso del software LabView se puede obtener curvas de funcionamiento, valores máximos y mínimos, promedios, etc.

### **3.14 SELECCIÓN DE SENSORES.**

Se necesita conocer el número de marcha de la caja de cambios, por lo que en el diseño se ha decidido usar un sensor que detecte la posición de la palanca de cambios en su recorrido de funcionamiento, tal como se muestra en la figura 3.5.



**Figura 3.5. Ubicación del sensor para detectar la posición de la palanca de cambios.**

En el mercado se busco sensores electrónicos de contacto o posición, pero no hubo ninguno que se ajuste a nuestra aplicación, por lo que se opto en usar un elemento automotriz de fácil acceso en el mercado como son los trompos de retro, puede ser considerado como un sensor electromecánico. Estos funcionan como un pulsador normalmente cerrado, al momento que son presionados pasan a estar en circuito abierto es decir ya no conduce corriente.

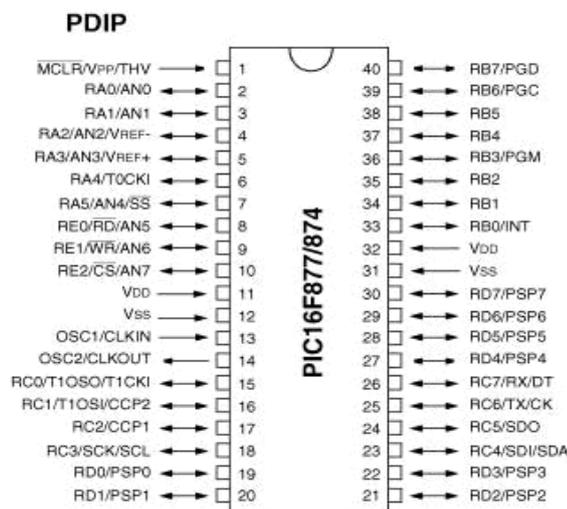
Estos trompos se los puede encontrar en algunos modelos en el mercado, se eligió emplear uno que tenga el vástago de pulsación lo suficientemente largo para que con el juego que tiene la palanca de cambios y el movimiento del vehículo no vaya a estar cambiando aleatoriamente de estado (on/off).

### **3.15 SELECCIÓN DE CIRCUITOS INTEGRADOS.**

Para obtener el dato de RPM del motor necesitamos convertir la frecuencia de chispazo de la bobina en una señal digital perfectamente cuadrada para que el microcontrolador sea capaz de contar el número de pulsos (ya sea en flanco ascendente o descendente) por minuto. Para lograr este propósito se usara una combinación de diodo, condensador, resistencia, zener y un CI 4093. Este circuito se lo encontrará en páginas posteriores.

### **3.16 SELECCIÓN DEL MICROCONTROLADOR.**

El módulo va a manejar e interactuar con displays, LCD, teclado, memoria externa EEPROM, reloj RTC, sensores de posición, e integrados; por lo que será necesario una gran cantidad de pines. El PIC 16F877A (Figura 3.6) es el que mejores se ajusta a la aplicación del módulo, ya que un solo PIC aun no es suficiente por la gran cantidad de pines que se necesitara. En este proyecto se utiliza dos microcontroladores de este tipo; un PIC será maestro mientras el otro será su esclavo.



**Figura 3.6. Distribución de pines del PIC 16F877A.**

A continuación se mencionan las características más relevantes del PIC 16F877A:

- Frecuencia Máxima 20 Mhz.
- Memoria de programa de 14.3K, soporta 8192 instrucciones.
- Memoria EEPROM de 256 bytes y Data Sram de 256 bytes.
- Comunicación mediante protocolo I2C.
- Comunicación Serial MSSP, Usart.
- Comunicación Paralelo PSP.
- Posee 33 pines de Entrada / Salida.
- 14 Interrupciones y 3 Timers.
- Un Juego de 35 Instrucciones.

- Arquitectura Harvard.

### **3.17 SELECCIÓN DE LA MEMORIA EXTERNA PARA ALMACENAMIENTO DE DATOS.**

Se decidió buscar una memoria externa EEPROM que cumpla con las siguientes características:

- Gran capacidad de almacenamiento.
- Se maneje con protocolo bus I2C.
- Fácil acceso y bajo costo.

La mejor opción que se tuvo fue una Memoria EEPROM 24LC512 de 512 Kbits (Figura 3.7), sus principales características técnicas se describen en el capítulo anterior. En esta memoria se ocuparan las 19 primeras localidades para almacenamiento de datos de configuración tales como numero de cilindros y marchas, régimen de revoluciones para cambio de marcha, hora y fecha del sistema. A partir de la localidad 20 y 21 se grabaran 1 dato de RPM y 1 dato de marcha respectivamente. El módulo recolecta estos dos datos cada medio segundo hasta llenar la capacidad de la memoria que es 64Kbytes.



**Figura 3.7. Memoria EEPROM 24LC512.**

Si a las 64000 localidades disponibles en la memoria le restamos las 19 iniciales tendremos aproximadamente 63980 localidades; si cada medio segundo usamos 2 localidades, el tiempo máximo de recolección de datos que tendrá el modulo será de 15995 segundos. Que equivale más o menos a 4 horas y 26 minutos.

### 3.18 SELECCIÓN DE LCD, TECLADO Y DISPLAYS.

Usaremos una LCD de 2 filas x 16 columnas que será suficiente para mostrar los distintos menús de configuración y datos mostrados por el módulo. La comunicación de la LCD con el PIC maestro será a 4 bits para ahorrar el número de pines. En la figura 3.8 se muestra los dispositivos utilizados.

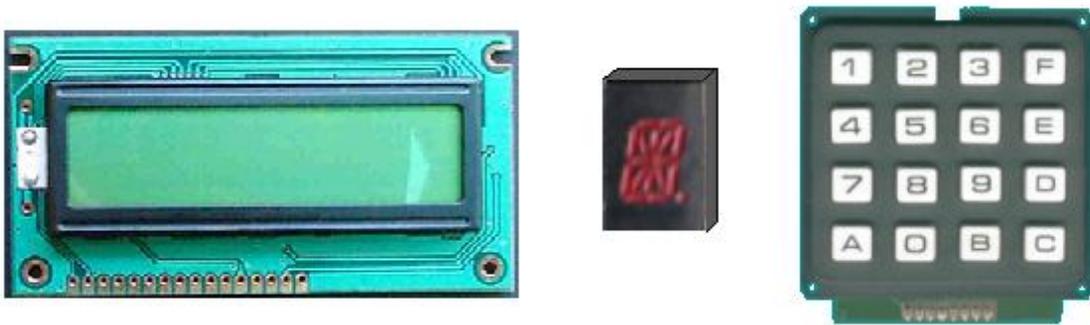


Figura 3.8. LCD, display y teclado respectivamente.

El teclado que usaremos será una matriz de 4 x 4, tiene 16 teclas hexadecimales suficientes para permitir al usuario tener una interfaz amigable con la LCD y el módulo en sí. Para mostrar las revoluciones y posición de marcha en un tamaño lo suficientemente visible para el conductor, usaremos 5 displays de 16 segmentos, estos nos permitirán mostrar tanto números y letras mayúsculas con rasgos bien definidos; características a las que el display convencional de 7 segmentos está muy limitado.

### 3.19 IMPLEMENTACIÓN DE LA COMUNICACIÓN SERIAL.

#### 3.19.1 Comunicación serial PIC a PIC.

Como se dijo anteriormente el módulo utilizara dos PIC 16F877A, el uno será el maestro que coordinara en sí, todo el funcionamiento de los periféricos externos e internos del módulo; mientras que el otro será esclavo, simplemente recibirá los datos de posición de marcha y rpm desde el pic maestro y los mostrará al conductor a través del uso de 5 displays de 16 segmentos.

La comunicación entre PIC esclavo y PIC maestro será serial asíncrona en modo Simplex, quiere decir que el maestro solo enviará datos más no los

recibirá, y de manera análoga el PIC esclavo se limitará a recibir datos más no a enviar (Figura 3.9).

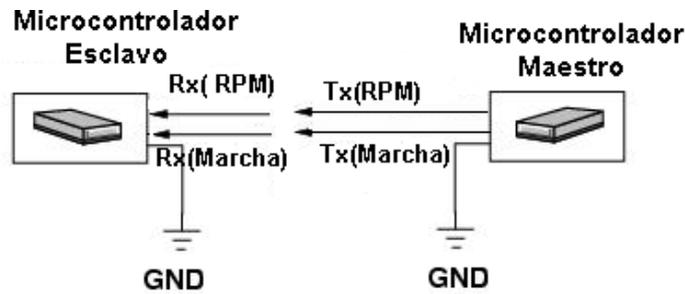


Figura 3.9. Comunicación entre PIC esclavo y PIC maestro.

Se usará dos líneas de transmisión de datos, una para transmitir las RPM y la otra para transmitir el número de marcha. Al ser comunicación serial asíncrona no necesita pulsos de reloj, en su lugar utiliza como mecanismo de referencia a la tierra, en donde la duración de cada bit es determinada por la velocidad de transmisión de datos que se debe definir previamente al momento de programar los microcontroladores.

### 3.19.2 Comunicación serial PIC a PC con CI MAX232.

El CI MAX232 es la solución para transmitir a mayor distancia, ya que incrementa los voltajes de 5V a +/-10V, gracias a un juego de capacitores que le ayuda a doblar voltajes, por lo que para su alimentación solo requiere una fuente de 5V, que puede ser la misma que usa el PIC. El Max232 dispone de 2 juegos de transmisores y receptores, de los cuales solo ocuparemos un par de ellos. En la figura 3.10 se aprecia la conexión del MAX232.

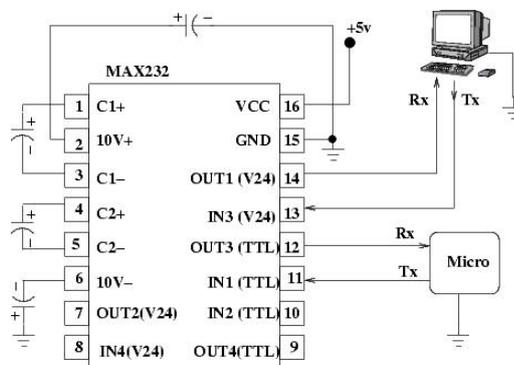


Figura 3.10. Conexión del CI MAX232.

El MAX232 en este caso nos ayudará a convertir los voltajes TTL del pic en voltajes de la norma RS232, quiere decir que si enviamos un estado lógico alto (5V), a la salida del Tout tendremos  $-10V$  y si enviamos un 0 lógico desde el PIC (0V), el MAX 232 enviará  $+10V$ , por lo tanto debemos invertir el dato de la salida del PIC y esto se lo consigue al enviando el dato como verdadero (T300, T1200, T2400, T9600).

### 3.19.3 Comunicación I2C PIC con RTC y memoria EEPROM.

Como se explicó en el capítulo anterior el bus I2C, permite la comunicación serial sincrónica con algunos dispositivos como RTC (reloj en tiempo real), memoria EEPROM, procesadores de señal, codificadores de video, sensores de temperatura, etc. El bus I2C necesita solo 2 líneas para transmitir y recibir datos, estos son para datos SDA y para la señal de reloj SCL.

Cada dispositivo conectado al bus tiene un código de dirección seleccionable mediante software, por lo que existe una relación permanente Master/Slave. El Master es el dispositivo que inicia la transferencia en el bus y genera la señal de reloj (SCL), y el Slave es el dispositivo direccionado, sin embargo cada dispositivo reconocido por su código (dirección), puede operar como transmisor o receptor de datos, ya que la línea (SDA) es bidireccional (Figura 3.11).

Código (dirección) del Reloj RTC DS 1307:  $\%11010000$ .

Código (dirección) de Memoria EERPOM 24LC512:  $\%10100000$ .

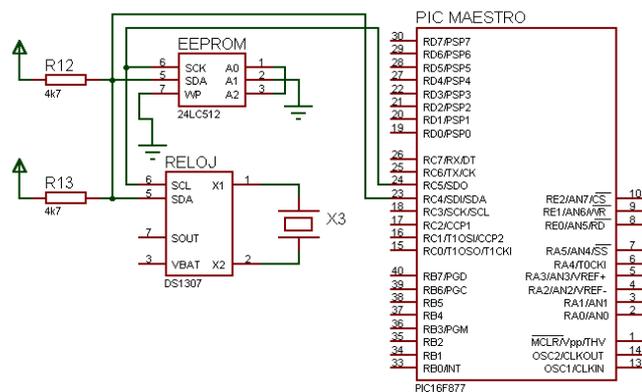


Figura 3.11. Conexión I2C PIC con RTC y memoria EEPROM.

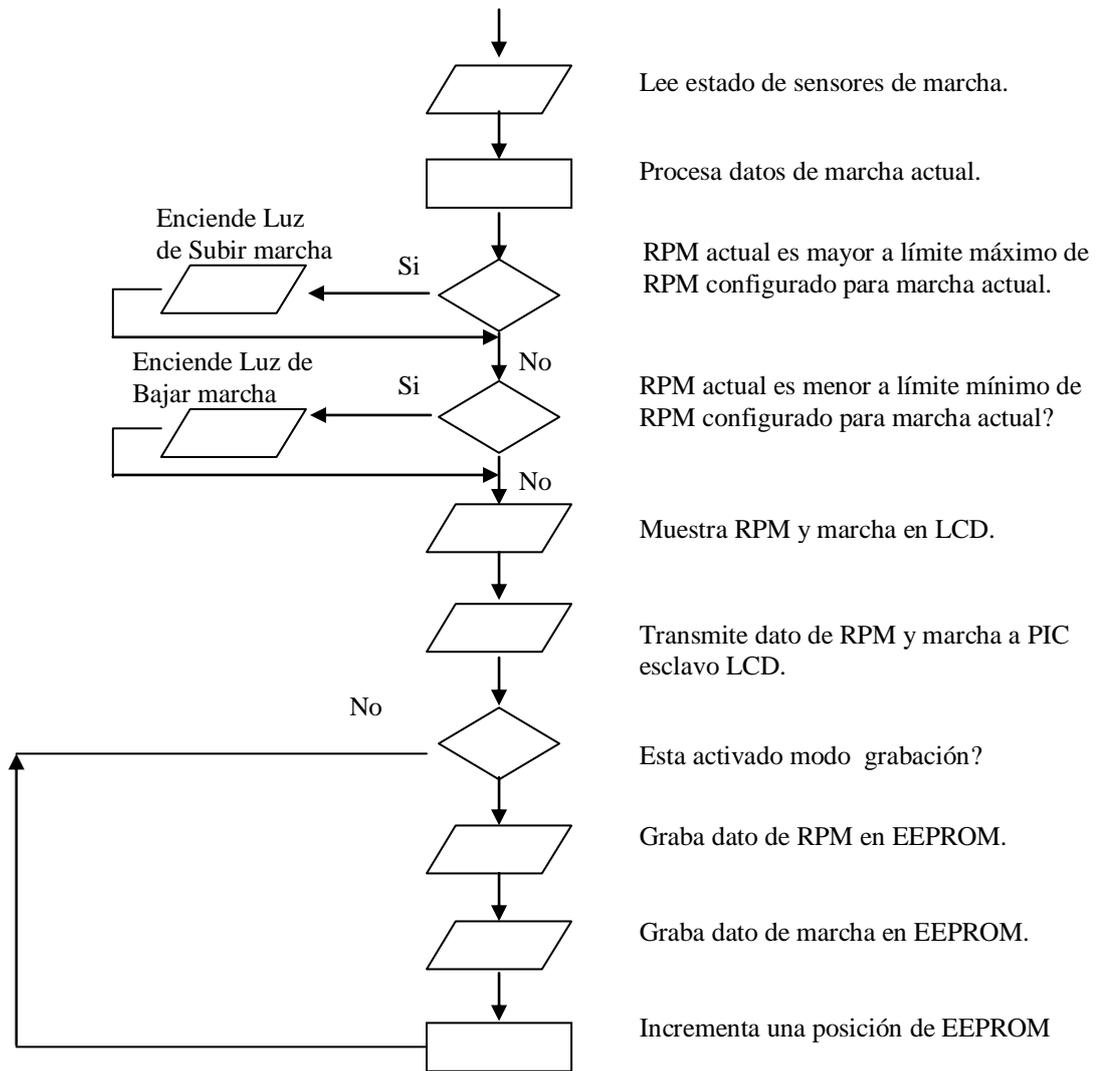
### 3.20 PROGRAMACIÓN DEL MICROCONTROLADOR.

Se decidió utilizar el software Micro Code Studio con su compilador Pic Basic Pro (PBP) para programar los microcontroladores de nuestro módulo. Pic Basic como su nombre lo dice es un lenguaje del tipo Basic de nueva generación, el mismo que hace que el programador maneje más fácil y rápido los microcontroladores de Micro Chip. En la tabla III.1 se describen las instrucciones más importantes que se usaron en la programación de los microcontroladores.

Tabla III.1. Instrucciones de Pic Basic.

INSTRUCCIÓN	DESCRIPCIÓN
LCDOUT Item{ , Item...}	Muestra caracteres en LCD.
COUNT Pin,Period,Var	Cuenta el número de pulsos en un Pin, durante un período periodo, y guarda el resultado en Var.
[var1] = [var2] MAX [var3]	Obtiene el valor máximo de entre dos variables y lo almacenado en una tercera.
EEPROM {Location ,} [ constante]	Define el contenido inicial en un chip EEPROM.
READ Address,Var	Lee el EEPROM incorporado en la dirección Address, y guarda el resultado en Var.
WRITE Address,Value	Graba valores Value en el EEPROM incorporado en la dirección Address especificada.

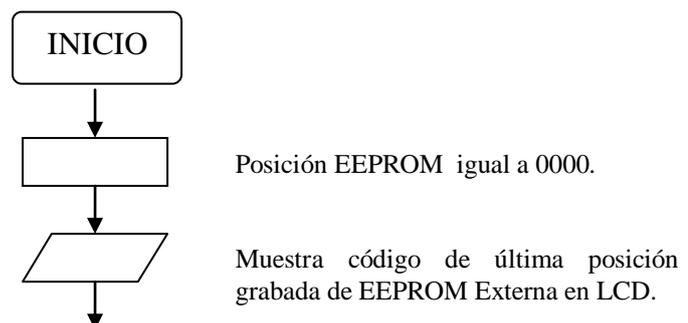


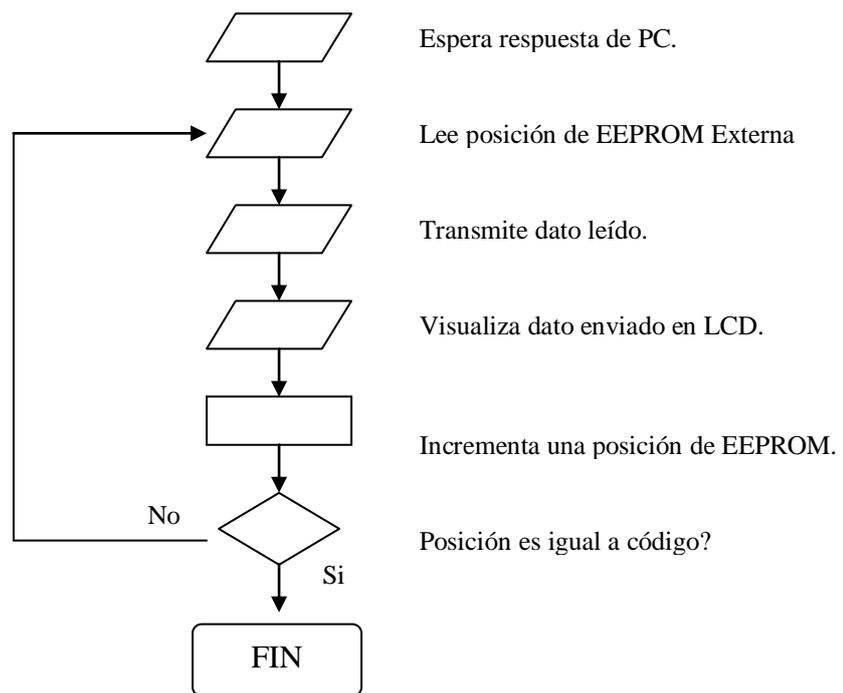


**Figura 3.12. Diagrama de flujo para obtención y visualización de datos de RPM.**

### 3.20.2 Diagrama de Flujo para descarga de datos de memoria EEPROM.

En la figura 3.13 se describe el proceso que realiza el microcontrolador para descargar los datos de la memoria EEPROM externa hacia un computador.





**Figura 3.13. Diagrama de flujo para descarga de datos de memoria EEPROM.**

### **3.21 DIAGRAMA DE CIRCUITOS DEL MÓDULO.**

A continuación se presenta los circuitos de todos los elementos electrónicos del módulo separados de acuerdo a la tarea que desempeñan en el módulo. En el anexo se adjunta un diagrama del circuito completo del módulo.

#### **3.21.1 Adquisición de datos (Figura 3.14).**

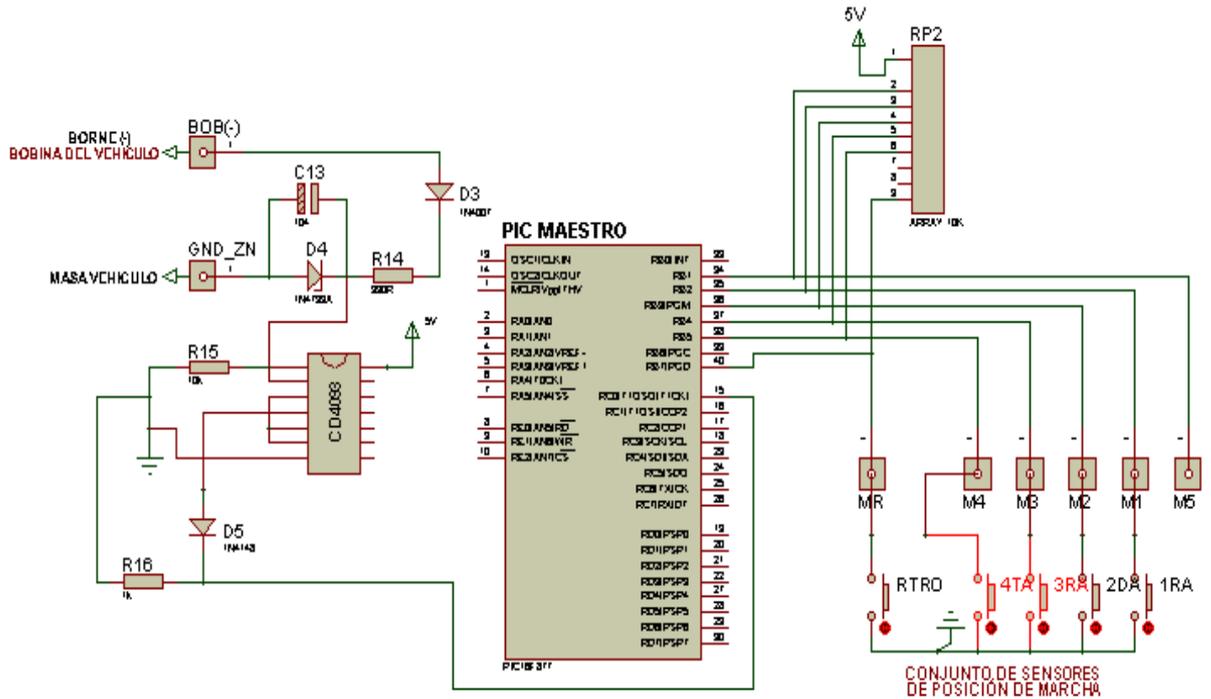


Figura 3.14. Circuito de adquisición de datos.

### 3.21.2 Almacenamiento de datos y RTC (Figura 3.15).

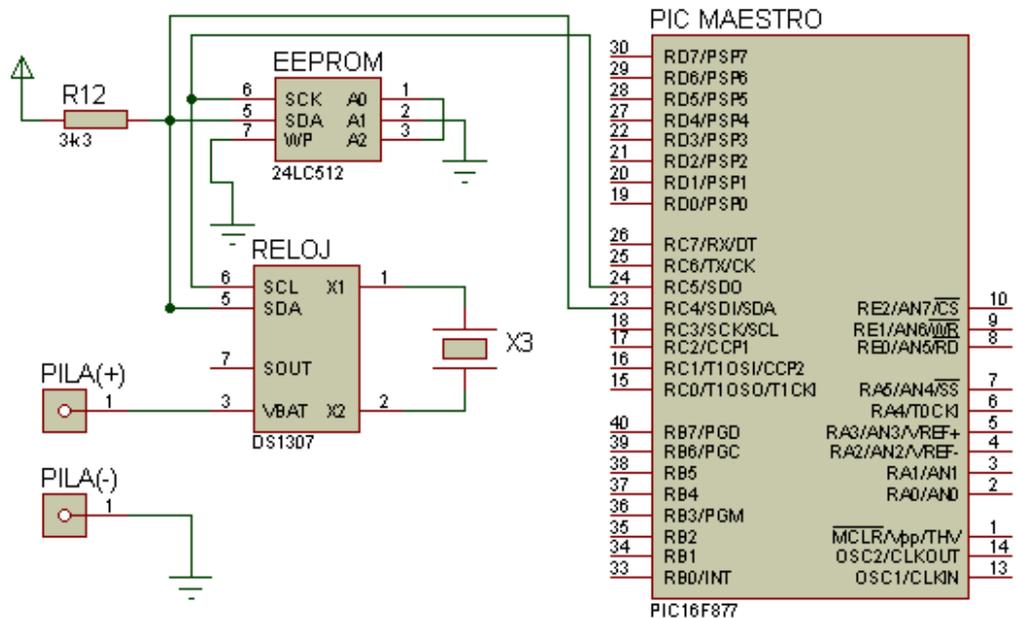


Figura 3.15. Circuito de almacenamiento de datos y RTC.

### 3.21.3 Interfaz LCD, teclado y PIC maestro (Figura 3.16).

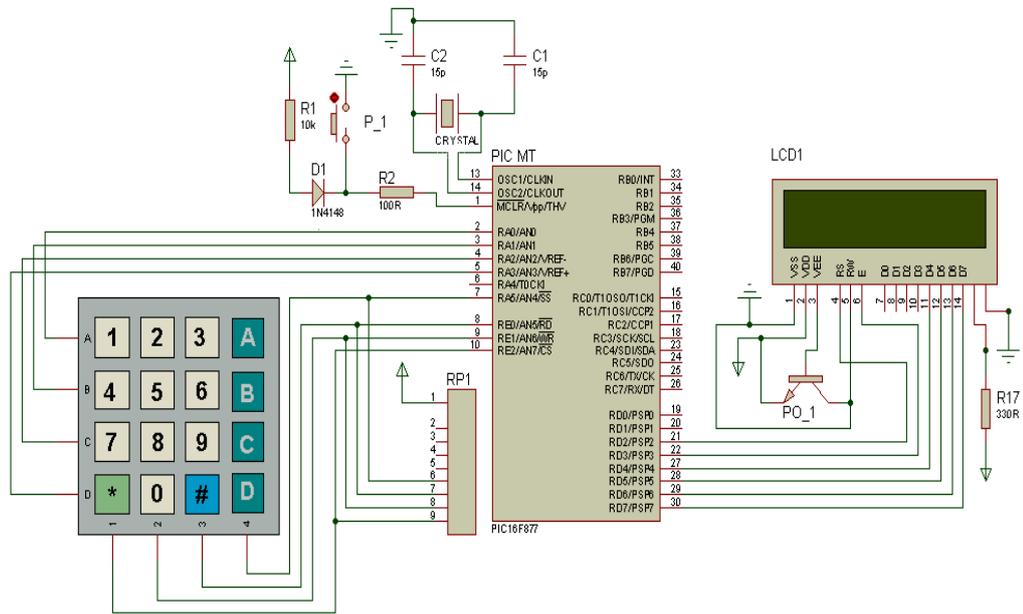


Figura 3.16. Circuito de interface LCD, teclado y PIC maestro.

### 3.21.4 Interfaz displays y PIC esclavo (Figuras 3.17 y 3.18).

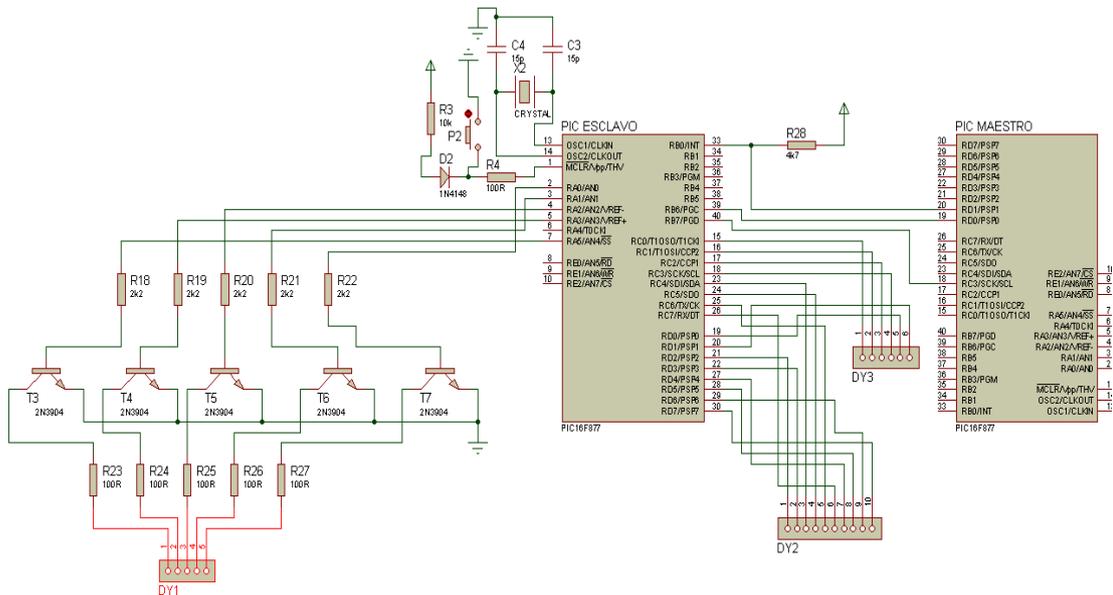


Figura 3.17. Circuito de interface displays y PIC esclavo.

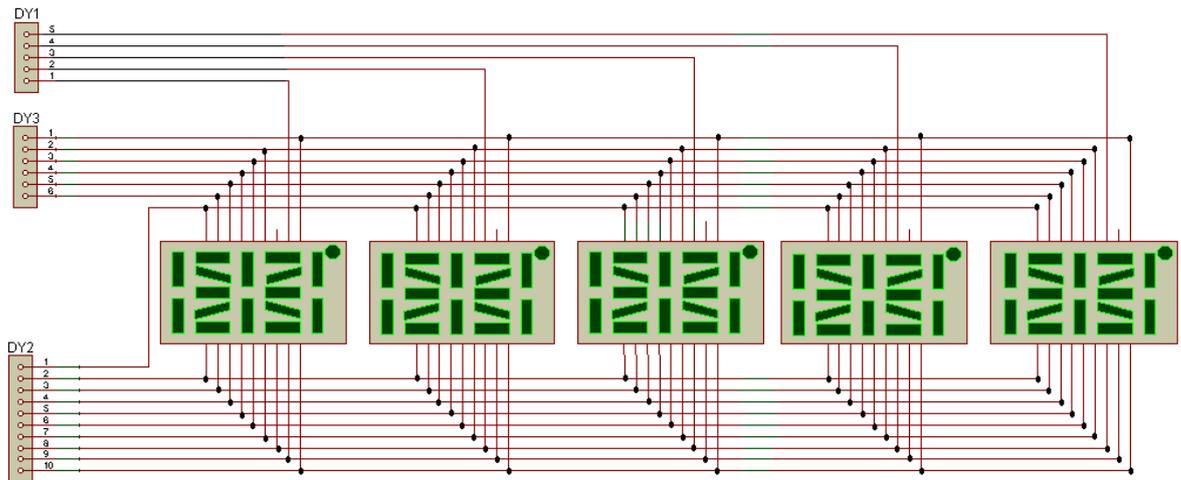


Figura 3.18. Multiplexación de displays.

### 3.21.5 Interfaz PIC maestro y MAX232 (Figura 3.19).

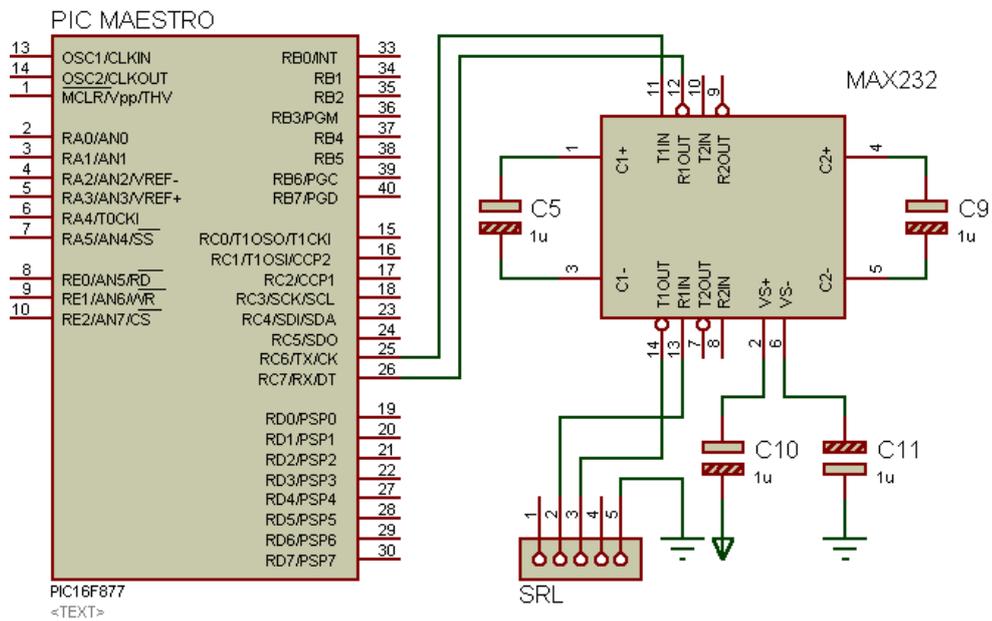


Figura 3.19. Circuito de interface PIC maestro y MAX232.

### 3.21.6 Interfaz con shift lite (Figura 3.20).

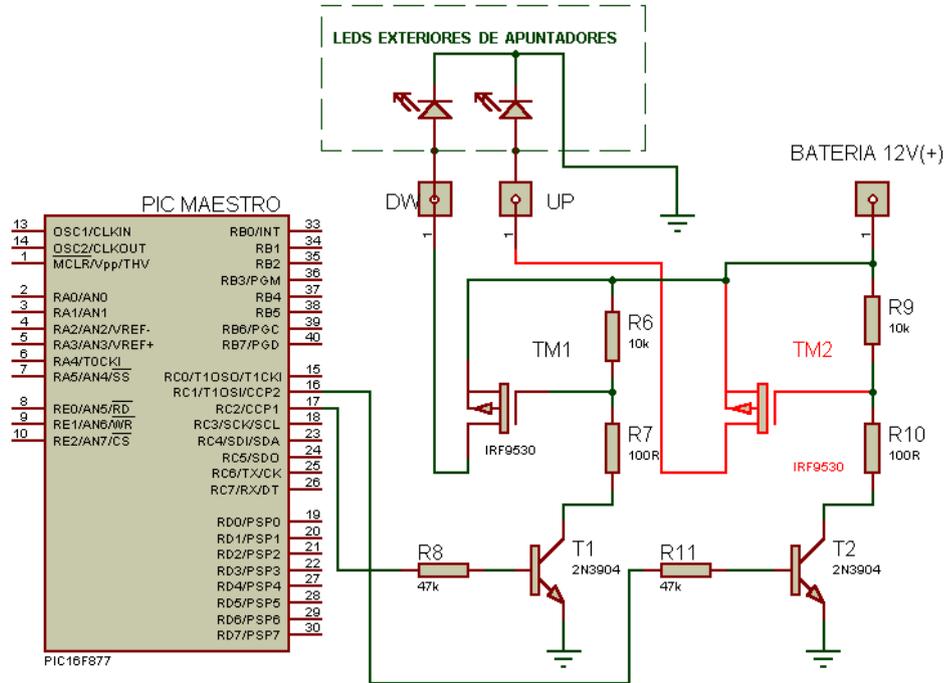


Figura 3.20. Circuito de interface con shift lite.

### 3.21.7 Circuito de alimentación regulado a 5V (Figura 3.21).

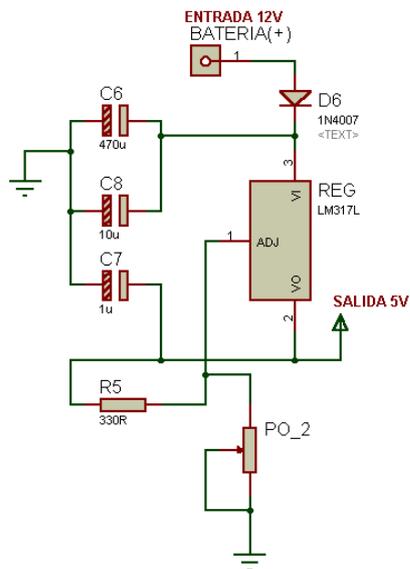


Figura 3.21. Circuito de alimentación regulada a 5V.

### 3.22 DISEÑO Y CONSTRUCCIÓN DEL CIRCUITO IMPRESO.

Para diseñar el circuito impreso se decidió usar el software Eagle. Luego de realizar los diagramas de todos los circuitos que consta el módulo, la plantilla que se obtuvo para la construcción del circuito impreso se muestra en la figura 3.22.

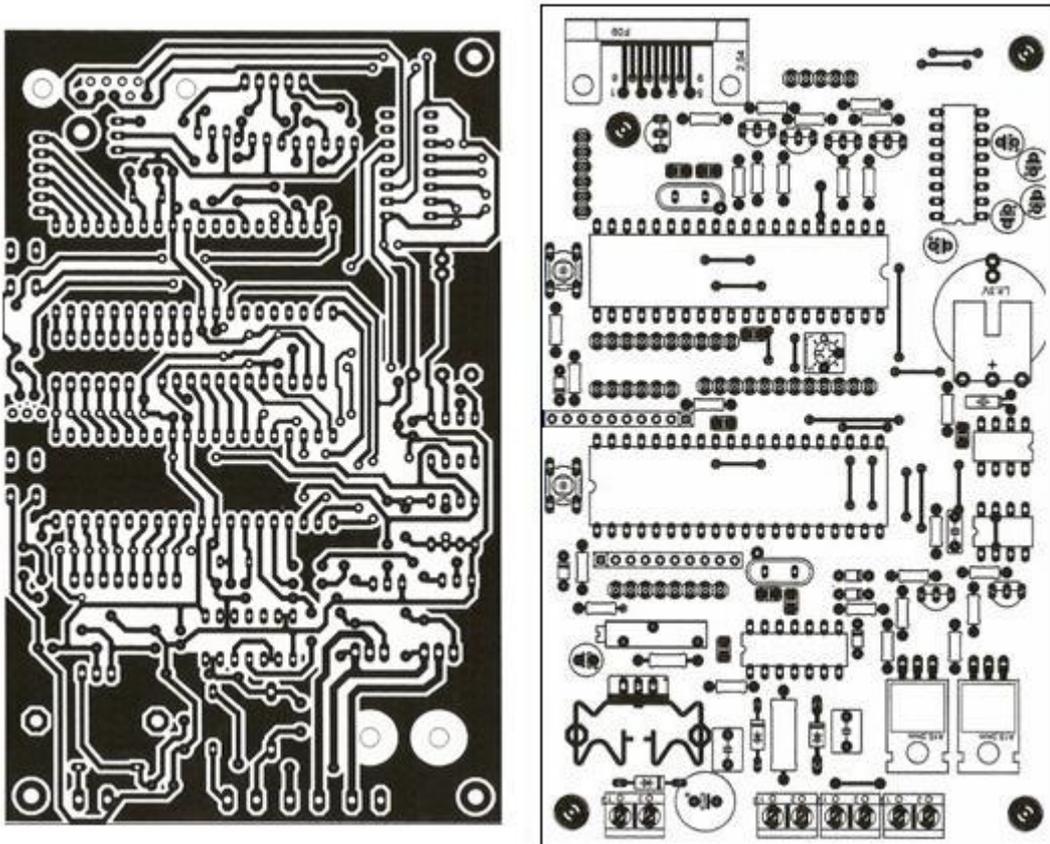


Figura 3.22. Plantilla para la construcción del circuito impreso.

### 3.23 DISEÑO DEL SOFTWARE DE VISUALIZACIÓN Y ANÁLISIS DE DATOS EN LABVIEW.

#### 3.11.1 Introducción.

LabView es un lenguaje de programación de alto nivel, de tipo gráfico y enfocado al uso en instrumentación. Pero como lenguaje de programación, debido a que cuenta con todas las estructuras, puede ser usado para elaborar

cualquier algoritmo que se desee, en cualquier aplicación, como en análisis, telemática, juegos, manejo de textos, etc. Cada programa realizado en LabView será llamado Instrumento Virtual (VI). El lenguaje más apto para realizar proyectos computarizados se lo ha considerado al software LABVIEW por vanas razones como:

- Es muy simple de manejar, debido a que está basado en un nuevo sistema de programación gráfica.
- Es un programa enfocado hacia la instrumentación virtual, por lo que cuenta con numerosas herramientas de presentación, con gráficas, botones, indicadores y controles, los cuales son muy esquemáticos y de gran elegancia. Estos serían complicados de realizar en bases como C++ o Visual Basic donde el tiempo para lograr el mismo efecto sería muchas veces mayor.
- Es un programa de mucho poder donde se cuentan con librerías especializadas para manejos de DAQ (Sistemas de adquisición de datos), Redes, Comunicaciones, Análisis Estadístico, Comunicación con Bases de Datos (Útil para una automatización de una empresa a nivel total).
- Con este las horas de desarrollo de una aplicación por ingeniero, se reducen a un nivel mínimo.
- Como se programa creando subrutinas en módulos de bloques, se pueden usar otros bloques creados anteriormente como aplicaciones por otras personas.
- LABVIEW permite utilizar tarjetas de adquisición de datos para comunicarnos con periféricos externos al computador, lo cual tiene grandiosas ventajas, pero los costos de estas tarjetas son muy elevados.
- LABVIEW permite comunicarse con los puertos de comunicación que posee el computador, tanto con el puerto paralelo, como con el puerto serial, siendo esta una de las grandes ventajas ya que el microcontrolador PIC puede aceptar un dato serial enviado desde el computador.

### **3.11.2 Programa para descargar datos desde una memoria EEPROM.**

Para descargar los datos desde la memoria EEPROM externa se necesita contar con dos elementos: la tarjeta de descarga denominada interface de descarga de datos (ítem 3.1.6) y un programa en LabView 7.1.

En el presente proyecto se usa las librerías de VISA donde se encuentran los modos de configuración del puerto serial, modo para lectura y el modo para cerrar el puerto de comunicación serial. El puerto serial recibe un dato de 8 bits a la vez, a diferentes velocidades de transmisión; el módulo de transmisión y recepción trabajan a 9600 baudios (el baudio se expresa como el número de bits recibidos en el puerto por cada segundo).

Para este fin se decidió utilizar una estructura de secuencia de cuatro pasos; a continuación se muestra el código estructurado dentro de cada ventana de la estructura de secuencia.

La secuencia “cero” se muestra en la figura 3.23, para esta secuencia son necesarios un For Loop, VISA Configure Serial Port, VISA Write, VISA Read, Concatene Strings, Feedback Node, y un Reversed Strng. El For Loop determina el número de caracteres a recibir por el puerto serial, en esta secuencia es igual al valor 14 debido a que corresponde al contenido de siete variables que se requiere para la fecha y hora de grabación. El VISA Configure Serial Port establece configuraciones del puerto serial como: velocidad de transmisión en baudios, el nombre del recurso que contiene la información sobre el tipo de interface I/O. El VISA Write escribe la cadena de caracteres al dispositivo especificado según el instante en que reciba el buffer la letra C; El VISA Read determina la cantidad de caracteres a leer en cada interacción del For Loop. EL Concatene Strings va situando los datos recibidos uno detrás de otro, de tal forma que al final se tiene toda la cadena de datos pero al revés. El Reversed String se encarga de volver la cadena a su forma original.

En resumen esta secuencia se encarga de recibir todos los caracteres leídos desde la memoria EEPROM en forma de cadena, almacenando en el indicador denominado Datos1.

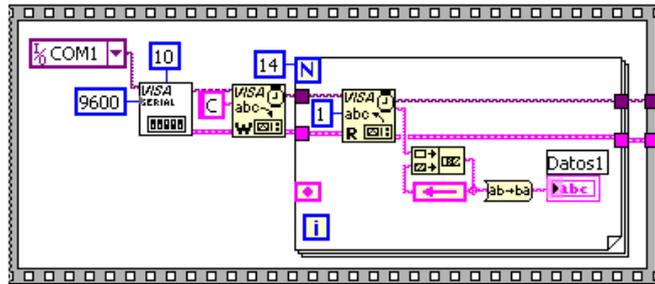


Figura 3.23. Código de la secuencia cero.

La secuencia “uno”, “dos” y “tres” se muestra en las figuras 3.24, 3.25 y 3.26 respectivamente; similar a la secuencia cero con la diferencia del número de caracteres a recibir por el puerto serial.

En la secuencia “uno” el número de caracteres a recibir es igual al valor 22 correspondiendo al contenido de once variables que se requiere para los regimenes de marcha y la máxima RPM. Esta información es almacenada en una cadena denominado Datos2.

En la secuencia “dos” recibe 4 caracteres correspondiendo al contenido de dos variables que se requiere para el número de cilindros y el número de marchas; para luego ser almacenados en una cadena denominado Datos3.

Finalmente la secuencia “tres” recibe el valor numérico que contiene el resto de caracteres enviado por la memoria EEPROM, se establece el número de veces que se ejecute el ciclo For Loop; estos datos son almacenados en un indicador tipo string denominado Cadena correspondiendo a los valores de RPM y marchas. Esta secuencia también utiliza el VISA Close que cierra la sesión del dispositivo especificado.

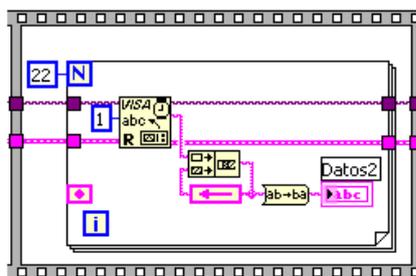


Figura 3.24. Código de la secuencia uno.

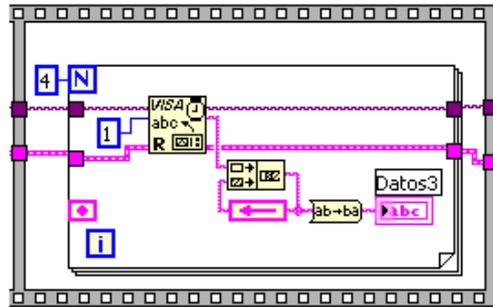


Figura 3.25. Código de la secuencia dos.

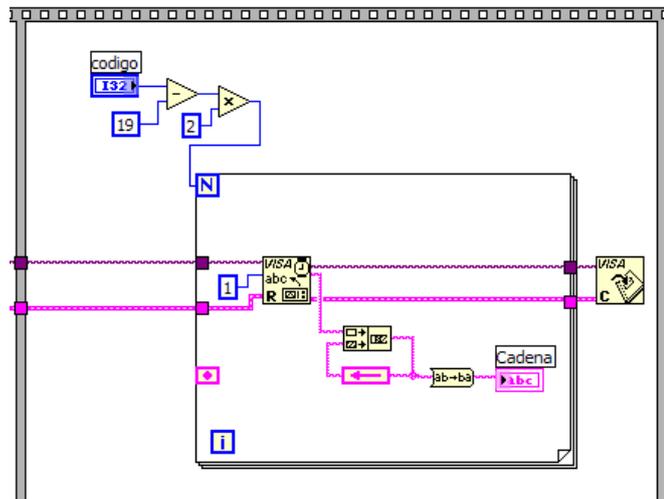


Figura 3.26. Código de la secuencia tres.

En la figura 3.27 se muestran el código para guardar los títulos en Excel; primero se necesita agrupar los datos en una matriz de 1 x 11, el programa consta de la herramienta llamada Build Array, esta crea la matriz; con el fin de usar el icono denominado Write Spreadsheet el mismo que necesita que los datos estén en forma de Array.

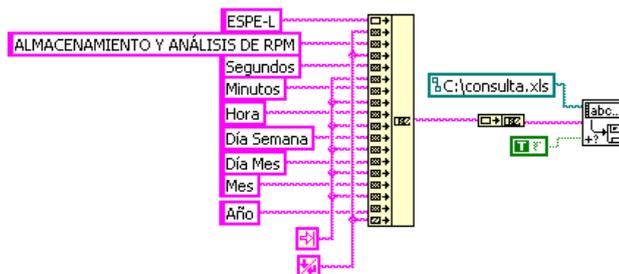


Figura 3.27. Código para guardar los títulos.

En las figuras 3.28, 3.29 y 3.30 se muestran las siguientes funciones:

- a) Obtener cuatro cadenas, la primera cadena de 14 caracteres (Datos1), la segunda cadena de 22 caracteres (Datos2), la tercera cadena de 4 caracteres (Datos3) y la cuarta cadena (Cadena) contiene el resto de caracteres que es enviado por la memoria EEPROM.
- b) Separar las cadenas en subgrupos de dos.
- c) Transformar de formato string a número para realizar comparaciones según la marcha valores de marcha con sus respectivas RPM almacena en indicadores.
- d) Tomar los valores de cada subgrupo y las almacena en una base de datos en Excel.

A continuación se detalla las herramientas utilizadas para realizar cada una de estas funciones, para realizar la separación de los datos en subgrupos de dos dígitos, se utiliza el icono denominado String Subset; este icono necesita que se ingrese la cadena de string, en el presente caso la primera cadena de catorce caracteres correspondiente a siete variables; además de un número de inicio, que indica el comienzo a realizar la separación y otro valor que determina cuántos caracteres va a separar, estos parámetros son llamados offset y lenght respectivamente.

Para obtener cadenas de dos dígitos se optó por sumar 10 a cada valor leído desde la I2C, debido a la facilitación de datos en el momento en que los datos son enviados desde el PIC hasta LabView; por lo que dentro del For Loop hay que restar el valor sumado para obtener el valor real.

Una vez separada la cadena, es necesario convertir estos datos a número para poder restar 10; sólo en el bloque de diagrama de la figura 3.29 una vez restados los datos son multiplicados por 100 para obtener los datos reales de regimenes del motor. Luego son añadidos al archivo generado en Excel.

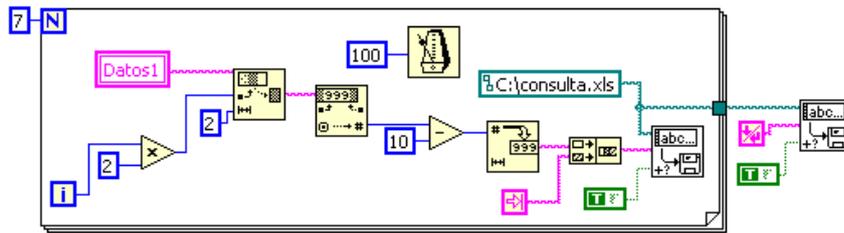


Figura 3.28. Código para almacenar los datos hora y fecha en Excel.

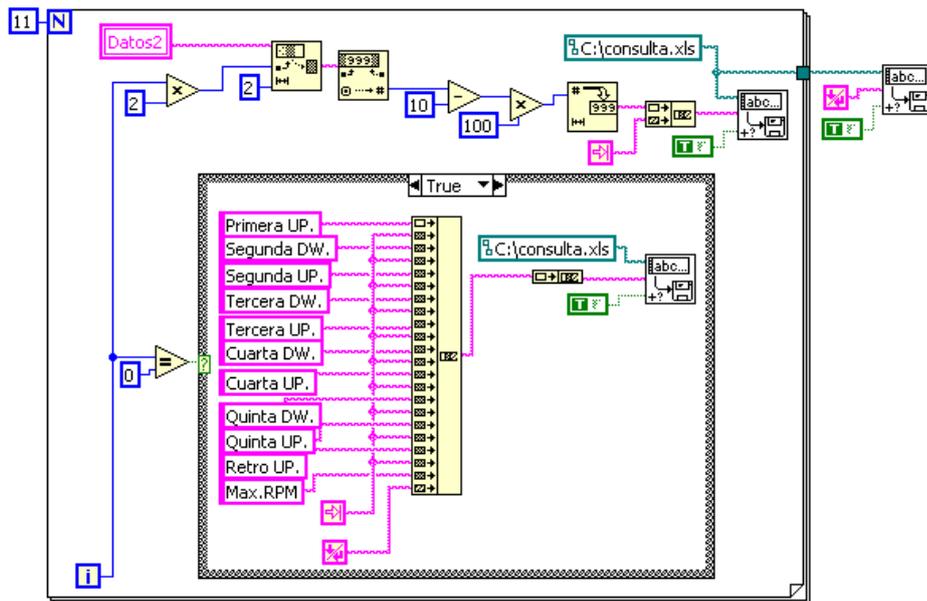


Figura 3.29. Código para almacenar los datos regímenes del motor en Excel.

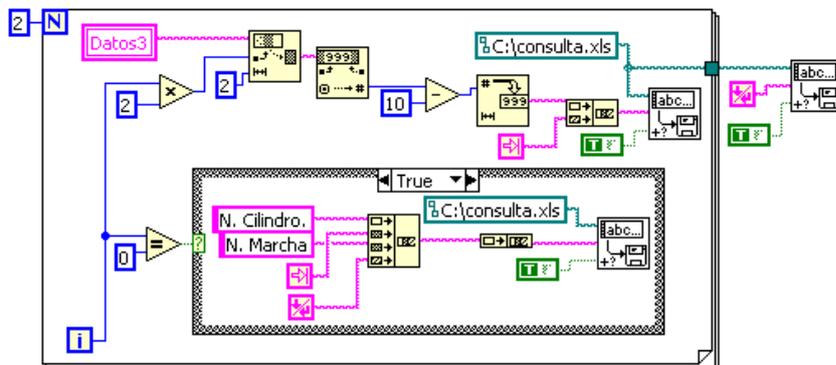


Figura 3.30. Código para almacenar el número de cilindros y número de marcha en Excel.

El diagrama de la figura 3.31 indica el código para separar los datos RPM de la primera marcha; para este propósito se toma el número de caracteres del

indicador denominado Cadena dividiendo esta longitud para cuatro, y con este valor se especifica el número de veces que se va a ejecutar el ciclo For. Dentro de este ciclo, la terminal de iteración (i) es la que contiene el número de veces que se está ejecutando el ciclo.

Este ciclo For se encarga de separar los datos del indicador Cadena en grupos de cuatro correspondiente a RPM y marcha, y este grupo a su vez es separado en dos valores indicando solo la marcha; esto lo hace mediante un registro de desplazamiento y la herramienta String Subset.

Los valores de marcha son de formato string y se convierte a datos numéricos para ser restados 10; adquiriendo así los datos reales de marcha. Estos valores son comparados con 1 que correspondiente a la marcha primera, con 2 a la segunda marcha, con 3 a la tercera, con 4 a la cuarta, con 5 a la quinta, con 23 a neutro y con 27 a retro.

Cada vez que una comparación de marcha es verdadera, los datos de RPM y marcha son separados solo en RPM convirtiéndolos en caracteres numéricos, para ser restados 10 y multiplicados por 100, finalmente se agrupan en un indicador tipo string correspondiente a las RPM de la marcha; en este caso de la primera marcha.

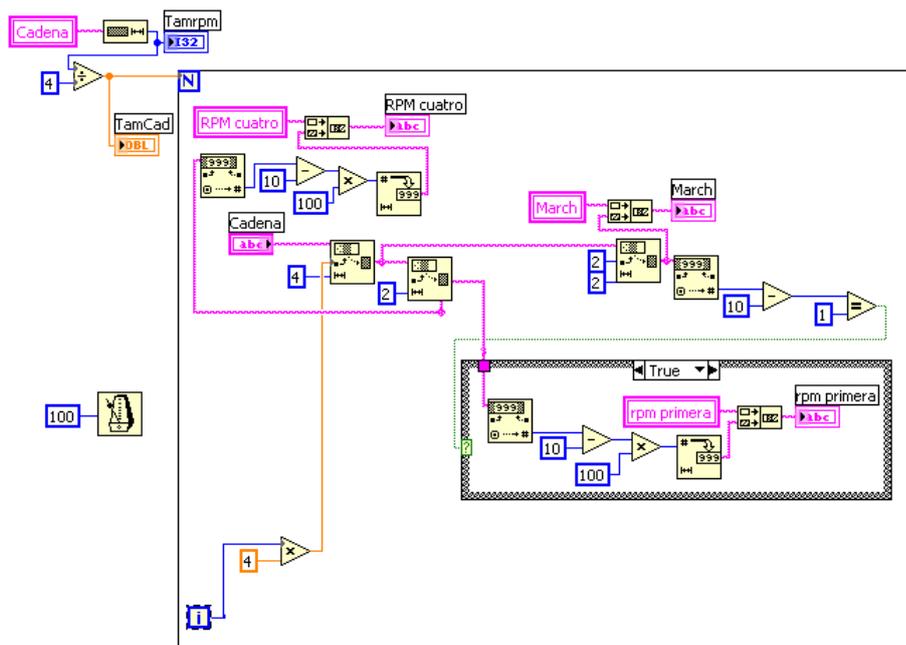


Figura 3.31. Código para separar los datos RPM de la primera marcha.

La figura 3.32 muestra los indicadores de las RPM de la primera marcha, RPM de la segunda marcha, RPM de la tercera, RPM de la cuarta, RPM de la quinta y RPM de retro; se toma el indicador denominado Tamrpm, que indica el número de caracteres de todas las RPM, este valor se divide para dos y con este número se especifica las veces que se va a ejecutar el ciclo For. La terminal de iteración (i) es multiplicada por cuatro cada vez que se ejecute el ciclo; este parámetro determina el comienzo de separación de datos y agrupa en subconjuntos de cuatro almacenados en una matriz de 1 x 12, una vez separados los datos son añadidos al archivo creado en Excel por cada iteración.

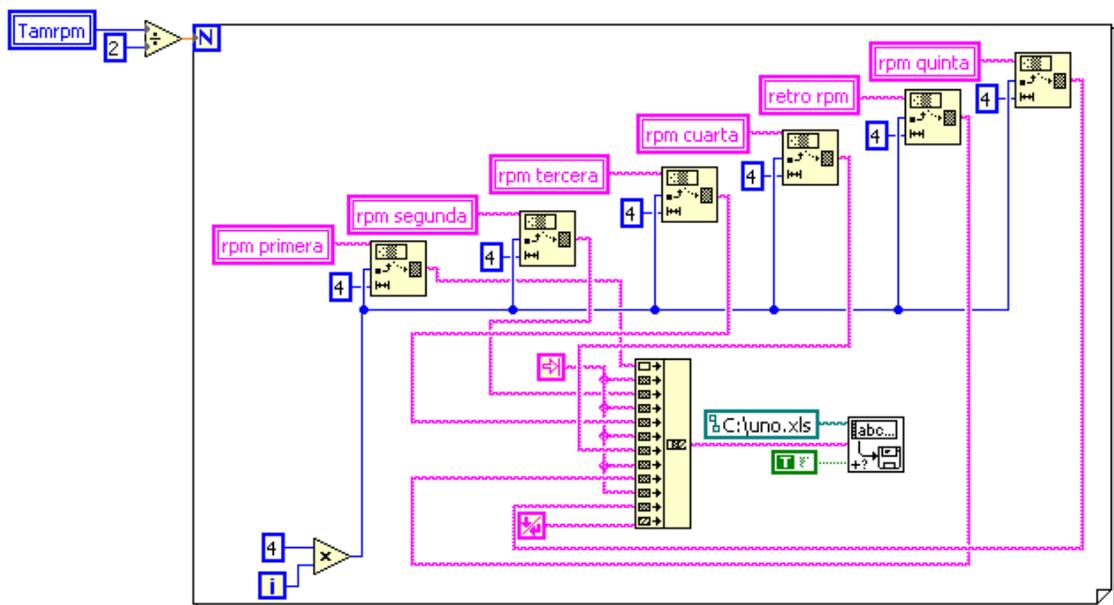


Figura 3.32. Código para guardar los datos en Excel.

### 3.11.3 Presentación de la pantalla principal.

En la pantalla principal se encuentra un menú en el que se puede elegir indistintamente cualquier opción para lo cual se utiliza un Tab control con diez páginas, las cuales han sido denominadas “PORTADA”, “INICIO”, “OBTENCIÓN DATOS”, “CURVAS RPM”, “CURVAS RPM2”, “EVALUAR SHIFT LITE”, “VALORES MAX.MIN.”, “VALORES PROMEDIO”, “SIMULACIÓN”, “CONSULTA”. La ventaja que ofrece la herramienta Tab control es que permite distribuir de mejor manera los espacios y de esta forma obtener una mejor

visualización de los indicadores. En cada página se cuenta con los respectivos indicadores numéricos y gráficos para cada variable.

Esta pantalla se puede observar en la figura 3.33.



Figura 3.33. Ventana Principal.

#### 3.11.4 Presentación de las ventanas del programa.

A continuación se presenta cada una de las ventanas que compone el software desarrollado.

En la figura 3.34 se presenta el panel frontal de la página "OBTENCIÓN DATOS", que indica los datos adquiridos en la memoria I2C al momento que se presiona el botón leer memoria como son: fecha y hora de grabación, datos generales (número de cilindros, número de marchas, máximas RPM), regímenes de marcha y la creación de una tabla. La primera columna contiene el parámetro de tiempo; la segunda columna indica la marcha; y la tercera columna contiene las RPM que se graba cada medio segundo.



Figura 3.34. Ventana de obtención de datos.

A continuación se presenta el diagrama de bloque de la página “OBTENCIÓN DATOS”, para este fin se decidió utilizar una estructura Case numérica, enseguida se muestra el código estructurado.

El Case “cero” se muestra en la figura 3.35, para este caso son necesarios un Read From Spreadsheet File, esta herramienta permite leer un archivo, leer la cantidad de líneas o columnas especificadas y convertir los datos a un arreglo de dos dimensiones; también es necesario un Number to Fraccional String, esta herramienta convierte un arreglo de valores numéricos a un arreglo de cadenas que son desplegadas en una tabla; la función Delete from Array propios de LabView permite eliminar un elemento o una porción en el arreglo de datos y la función Index Array permite tener acceso a un elemento.

Todas estas funciones permite la creación de arreglos de dos dimensiones para ser mostrados en la ventana de obtención de datos.

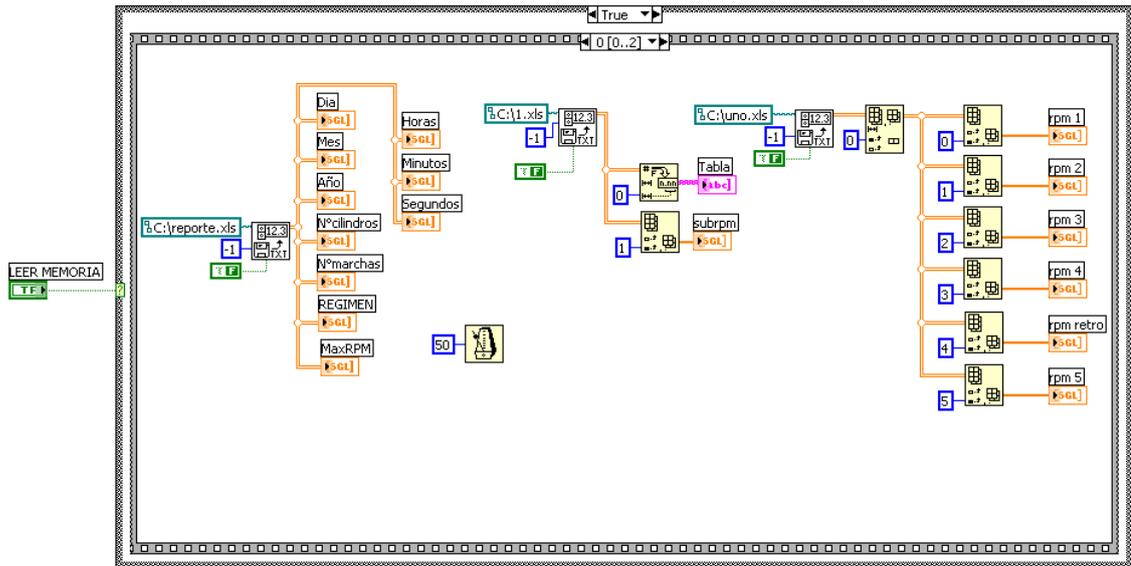


Figura 3.35. Código para la creación de arreglo de datos.

El Case “uno” se muestra en la figura 3.36, para este caso son necesarios un Read From Spreadsheet File, un Delete from Array y un Index Array. Esta secuencia se encarga de leer los archivos creados, permitiendo tener el acceso a los elementos necesarios para el arreglo de datos; estos elementos se colocan en columna obteniendo así los valores de las columnas de tiempo, marcha y RPM.

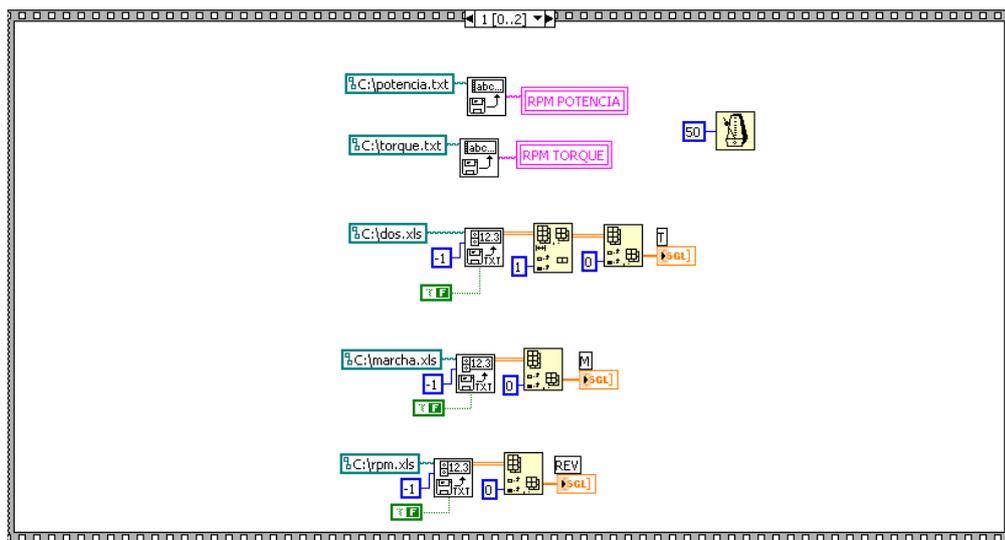


Figura 3.36. Código para el acceso de los datos de tiempo, marcha y RPM.

El Case “dos” se muestra en la figura 3.37, para este caso son necesarios un Build Array que une múltiples arreglos; un Number to Fraccional String, esta función convierte un arreglo de valores numéricos a un arreglo de cadenas que son desplegadas en una tabla.

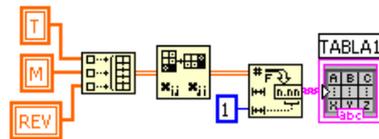


Figura 3.37. Código para unir múltiples arreglo de datos, desplegadas en una tabla.

La figura 3.38 presenta el panel frontal de la página “CURVAS RPM”, que muestra las curvas RPM vs. Tiempo y Marcha vs. Tiempo en el instante que se presiona el botón graficar.

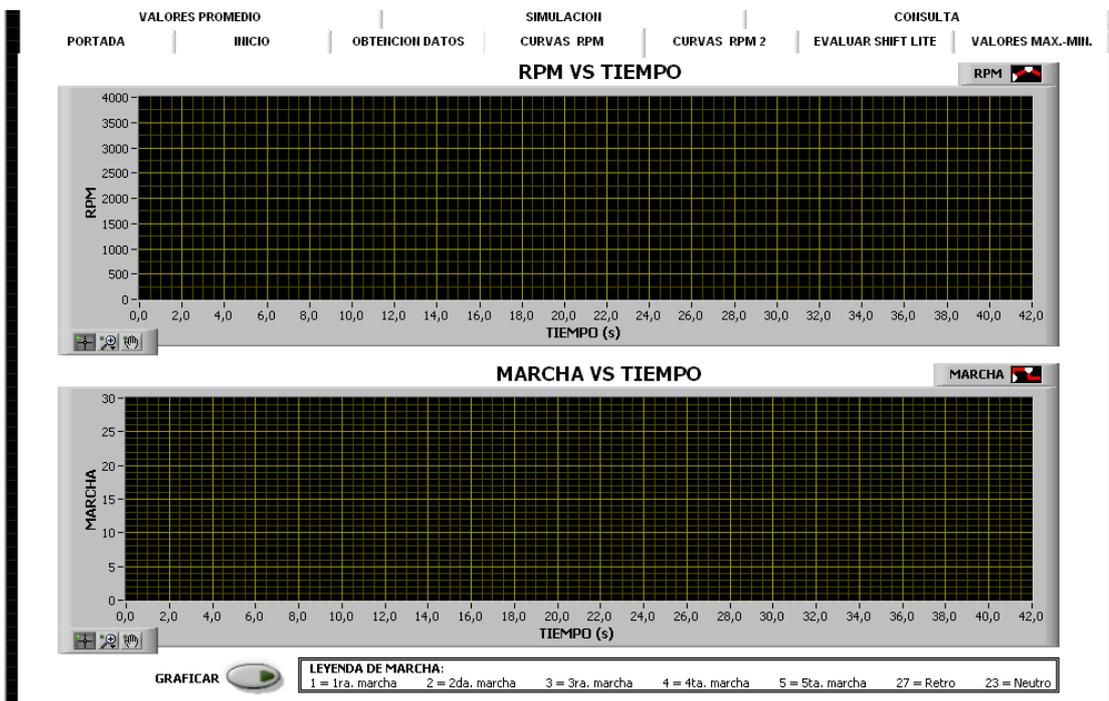
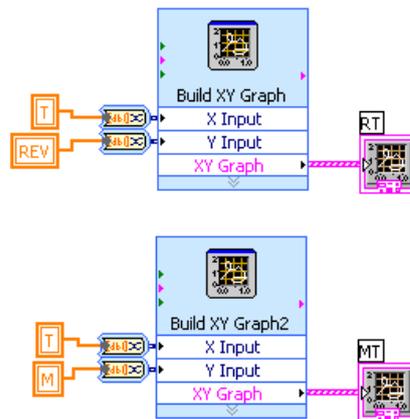


Figura 3.38. Ventana que muestra las curvas RPM vs. Tiempo y Marcha vs. Tiempo.

A continuación se presenta el diagrama de bloque de la página “CURVAS RPM”, donde se utiliza una estructura Case numérica.

En el Case “cero” es necesario dos XY Graph con sus respectivos Build XY Graph, permitiendo una gráfica con múltiples valores como un plano cartesiano de las curvas RPM vs. Tiempo y Marcha vs. Tiempo. Estas herramientas se muestran en la figura 3.39.



**Figura 3.39. Programa para la realización de curvas.**

En los Case “uno” hasta el “seis” su programación varía según los datos guardados en la variable local correspondiente a cada marcha. La secuencia que se muestra en la figura 3.40, se encarga de separar los datos de las RPM de la marcha correspondiente, en este caso las RPM de la primera marcha en grupos de dos; una vez separados los datos son almacenados en un indicador tipo string (subrpm1). Dentro del ciclo principal se coloca otro ciclo For, de igual manera separa la cadena denominada RPM POTENCIA (indicador que contiene los valores de potencia según las RPM) en subconjuntos de seis y almacenados en el indicador subRPM.

La figura 3.41 es la continuación de la figura 3.40, la figura 3.41 se utiliza una estructura Case Booleana, en este caso se compara los valores de los indicadores subrpm1 y subRPM, si son iguales realiza las comparaciones requeridas para obtener todas las potencias de las RPM de la primera marcha.

Para determinar los valores de la potencia de las demás marchas su programación es la misma, variando los indicadores que contiene las RPM de cada marcha.

Para adquirir los valores de torque según las RPM de cada marcha, es preciso una programación similar a las figuras 3.40 y 3.41, diferenciándose en el control RPM POTENCIA por el control denominado RPM TORQUE.

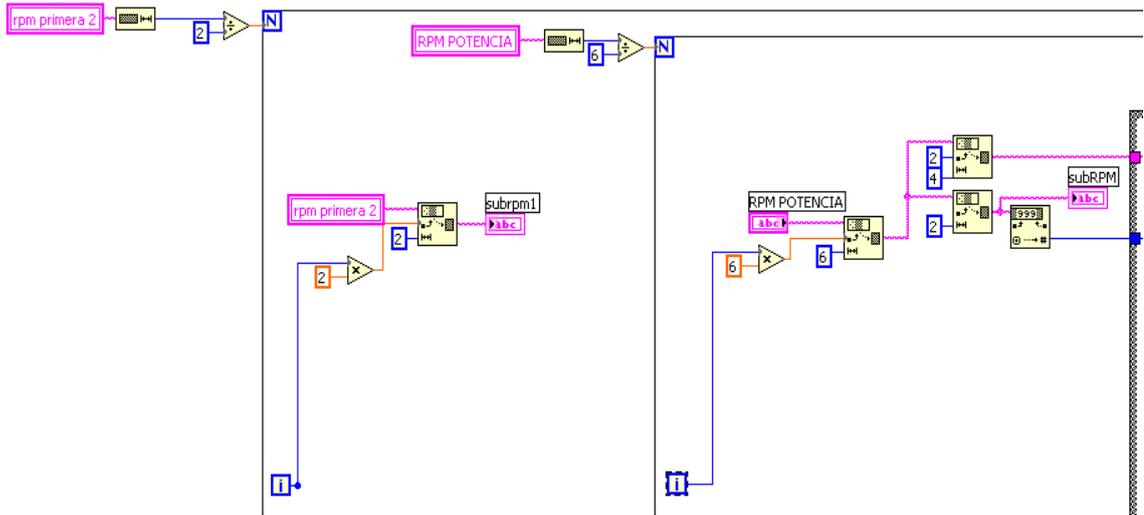


Figura 3.40. Programa de separación de datos.

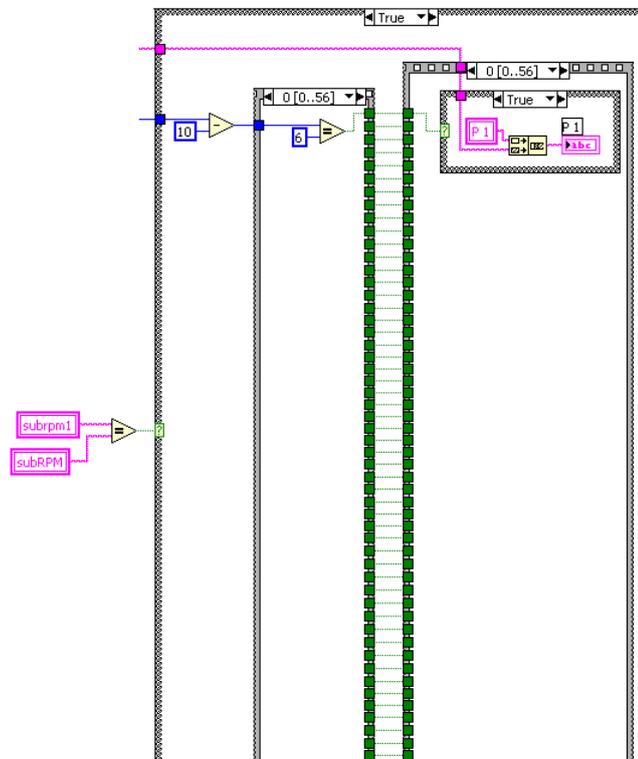


Figura 3.41. Bloque de Comparaciones.

La figura 3.42 muestra el panel frontal de la página “CURVAS RPM2”, indicando las curvas Torque vs. Tiempo y Potencia vs. Tiempo en el momento que se presiona el botón graficar. En su programación se usa las herramientas XY Graph con los datos conseguidos anteriormente.

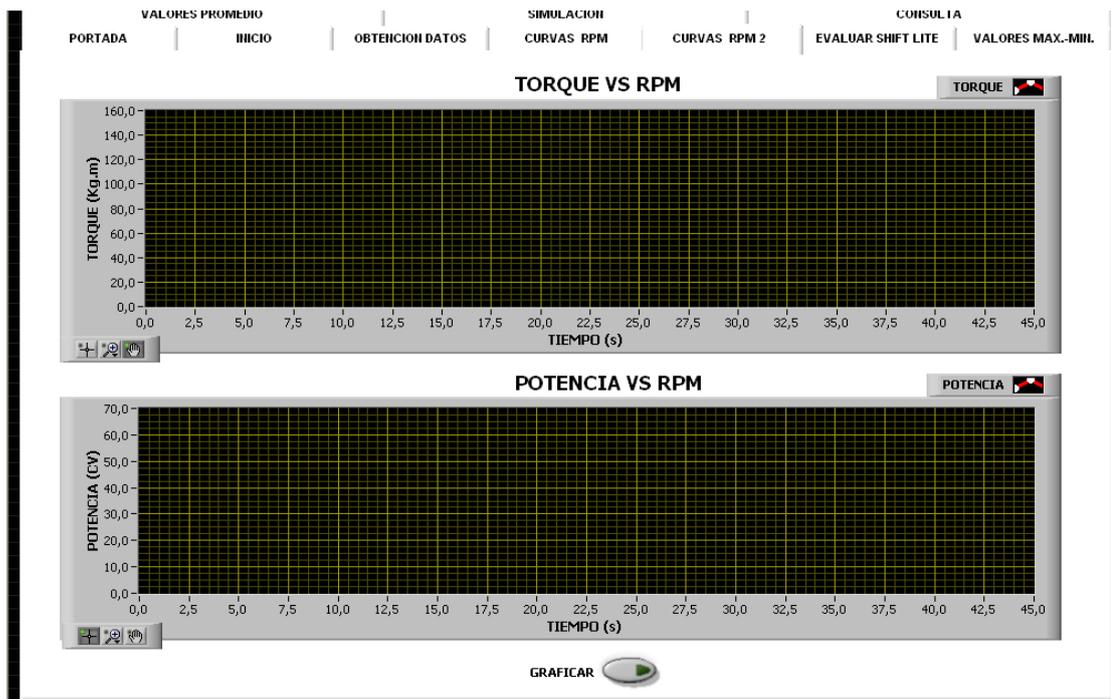


Figura 3.42. Ventana que muestra las curvas Toque vs. Tiempo y Potencia vs. Tiempo.

El panel frontal de la página “EVALUAR SHIFT LITE” se muestra en la figura 3.43, según los regímenes de marcha se obtiene la evaluación de régimen del giro del motor de cada marcha (3=Motor sobrerrevolucionado, 2=Motor a régimen de giro normal y 1=Motor bajo revoluciones) indicadas en tablas, la evaluación general de las marchas y total presentadas en los indicadores numéricos. En el bloque de código se usa condiciones de igualdad, estableciendo condiciones de rango para cada marcha.

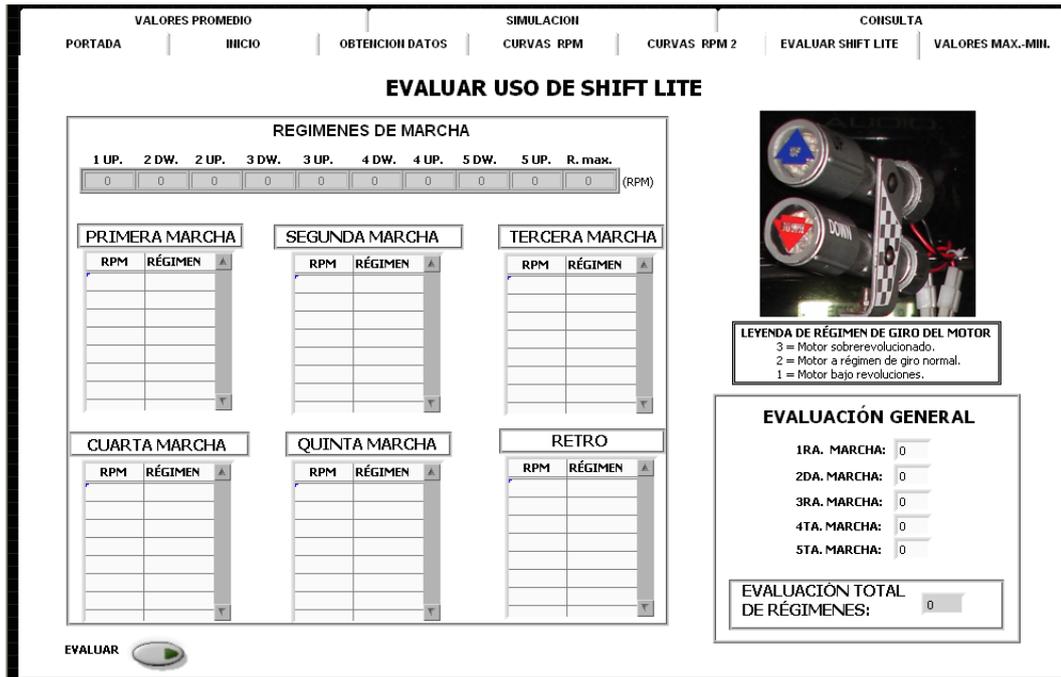


Figura 3.43. Ventana de evaluación de uso de la shift lite.

El panel frontal de la página “VALORES MAX. MIN.” se indica en la figura 3.44, en esta página se adquiere los valores máximos - mínimos de RPM, torque y potencia de cada marcha y los valores generales en el momento que se presiona el botón leer.

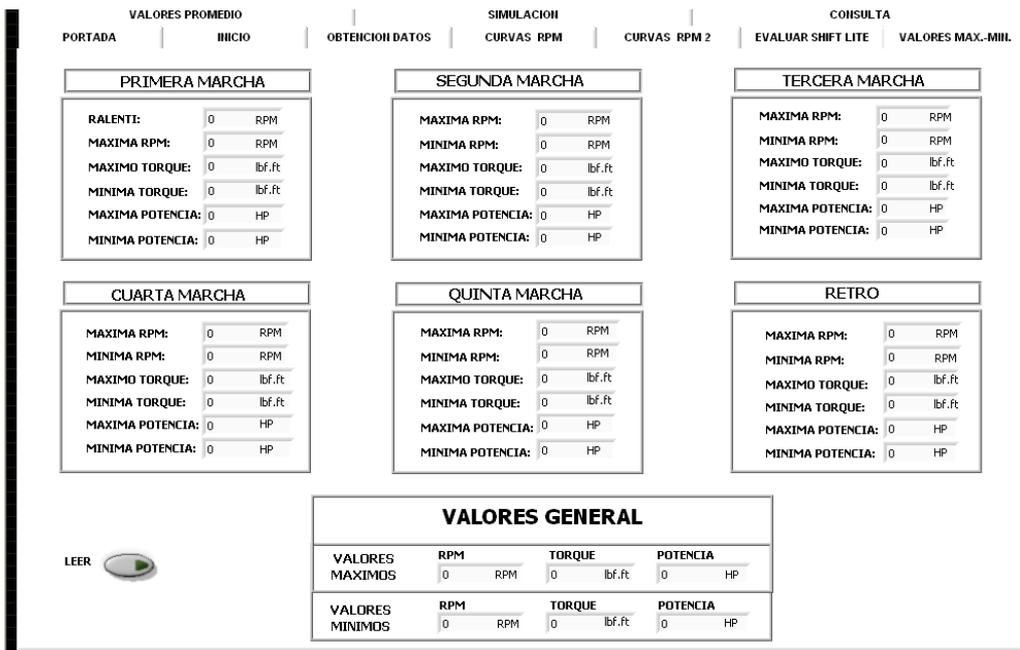


Figura 3.44. Ventana de valores máximos y mínimos.

En la figura 3.45 se utiliza la función propia de LabView para la obtención de los valores máximos - mínimos según las variables de RPM, torque y potencia de cada una de las marchas.

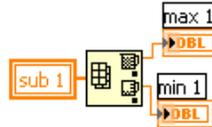


Figura 3.45. Obtención de los valores máximos y mínimos.

En la figura 3.46 se indica el panel frontal de la página “VALORES PROMEDIO”, en esta ventana se consigue los valores promedio de RPM, torque y potencia correspondientes a cada marcha; también se adquiere el promedio general de estas variables; en el momento que se presiona el botón obtener.

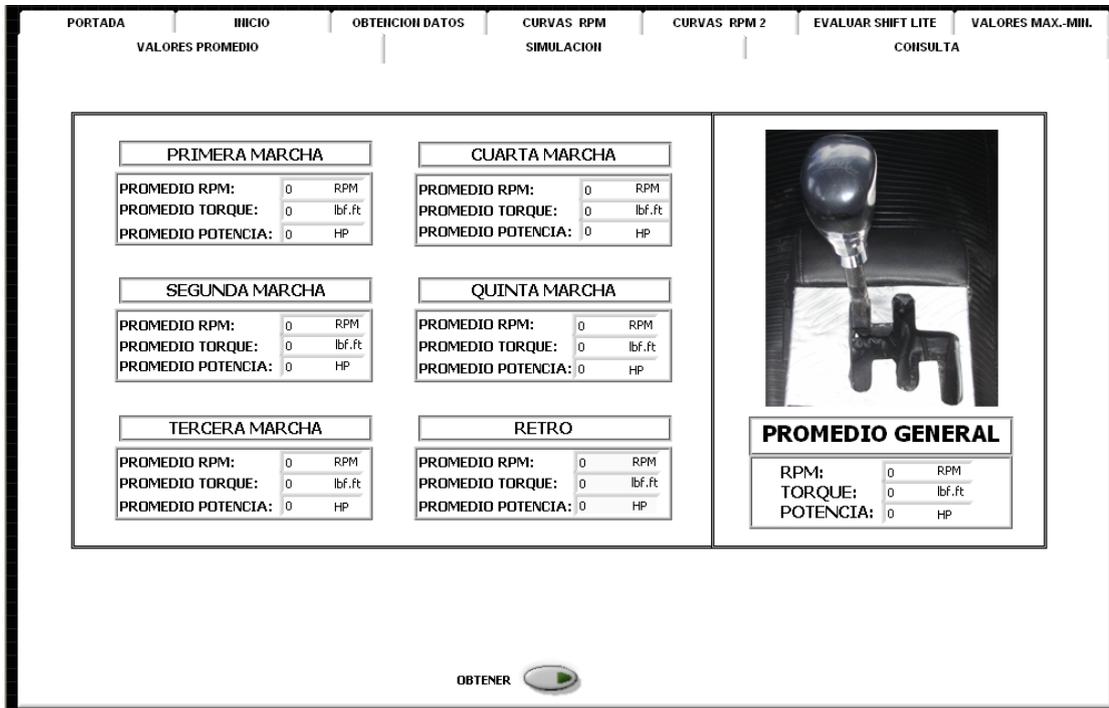
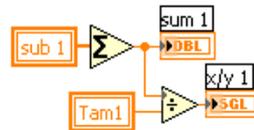


Figura 3.46. Ventana que muestra valores promedios y promedio general.

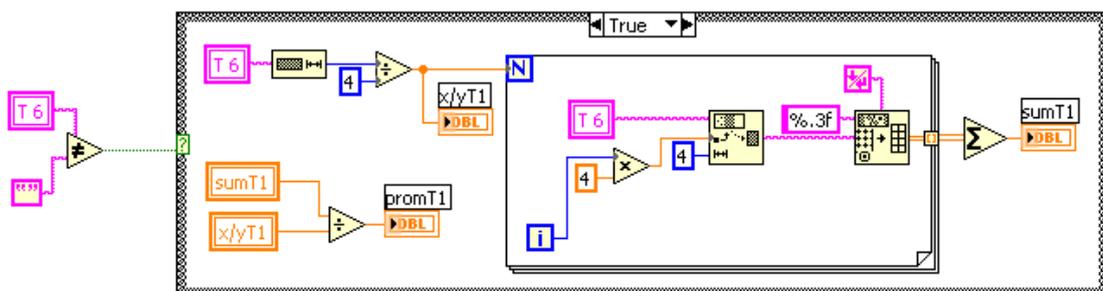
Para adquirir los valores promedios RPM, se realiza la sumatoria de todas las RPM de cada una de las marchas y luego se divide según los elementos que contenga; esto se esquematiza en la figura 3.47.



**Figura 3.47. Programa para obtener los valores promedios RPM.**

Para conseguir los valores promedio torque, se separa los datos que contiene el control (control que muestra los datos de torque según las RPM de cada marcha) en grupos de cuatro, una vez separados los caracteres se realiza la sumatoria de todas los valores de torque de cada una de las marchas y luego se divide según los elementos que contenga; esto se esquematiza en la figura 3.48.

Para adquirir los valores de promedio potencia, es necesaria una programación similar a las figuras 3.48, diferenciándose en el control que muestra los datos de torque por el control que indica los valores de potencia.



**Figura 3.48. Programa para obtener los valores promedios torque.**

La ventana de simulación se indica en la figura 3.49, esta página realiza una simulación en el instante que se ingrese el tiempo en segundos y se presione el botón inicio según los datos grabados, mostrando la marcha en que se encuentra con su respectiva revolución, torque y potencia.

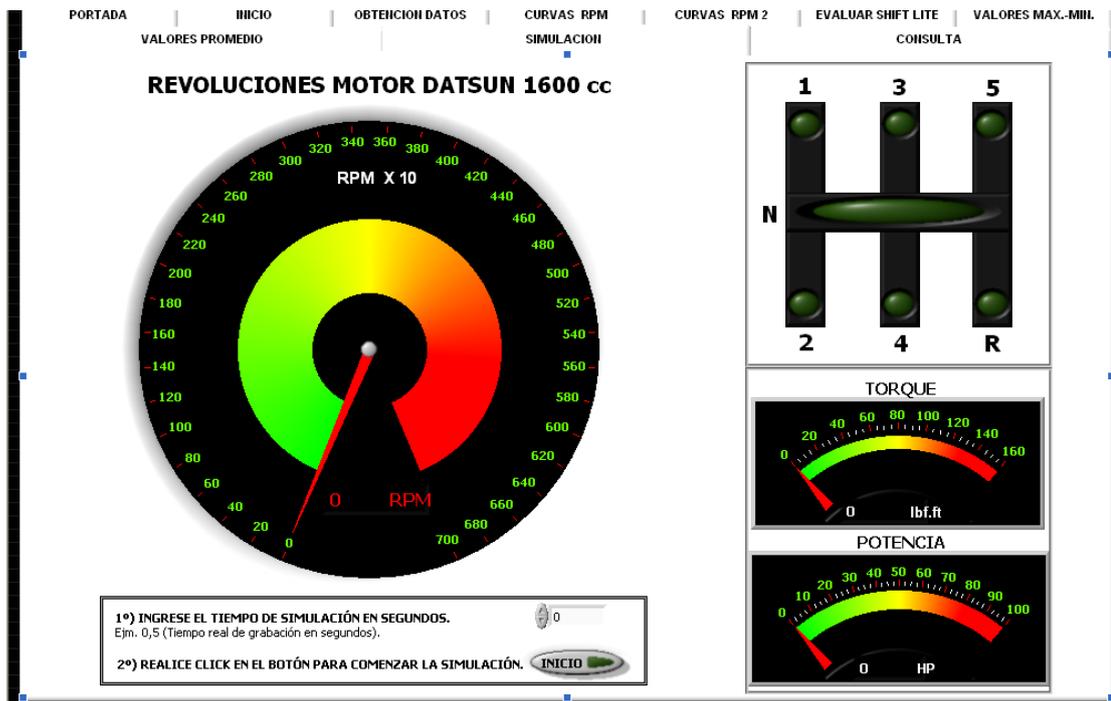


Figura 3.49. Ventana de simulación revoluciones, torque y potencia.

Para mostrar la simulación de potencia y torque, es necesaria la separación de los datos que contiene los controles respectivos en grupos de cuatro, luego se utiliza la función que convierte un arreglo de cadenas a un arreglo de valores numéricos para finalmente ser mostrados en los indicadores Potencia y Torque, esto se indica en la figura 3.50.

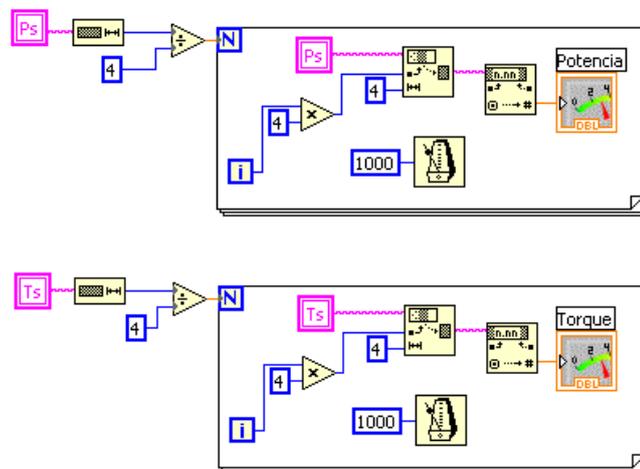


Figura 3.50. Programa para indicar la simulación de potencia y torque.

El diagrama de bloques de la figura 3.51 muestra el código para separar los datos RPM, para este fin el control denominado TamCad especifica el número de veces que se va a ejecutar el ciclo For.

Este ciclo For se encarga de separar los datos del indicador Cadena en grupos de cuatro correspondiente a RPM y marcha, y este grupo a su vez es separado en dos valores indicando solo la marcha.

Para adquirir los datos reales de marcha es necesario ser restados 10; estos valores son comparados con 1 que correspondiente a la marcha primera, 2 a la segunda marcha, 3 a la tercera, 4 a la cuarta, 5 a la quinta, 23 a neutro y 27 a retro. Cada vez que una comparación de marcha es verdadera, entra a una estructura Case y activa el led correspondiente; esto se observa en la figura 3.52.

Los datos de las RPM son restados 10 y multiplicados a la vez por 10, para ser mostrados en el indicador denominado Tac, también se muestra un indicador llamado RPM que indica las RPM que lee el tacómetro, valor que es multiplicado por 100; se considera algunos parámetros como la desactivación del botón INICIO y el botón RPM en el momento que se termine la simulación. Se debe considerar que la figura 3.52 es la continuación de la figura 3.51.

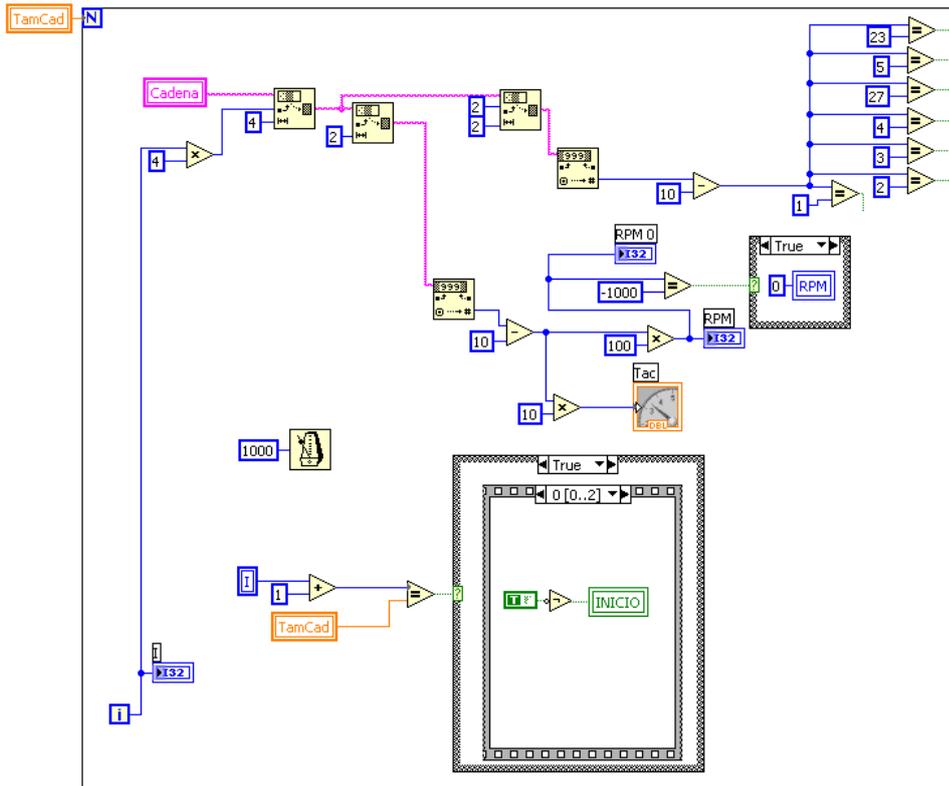


Figura 3.51. Programa para mostrar las RPM en el tacómetro.

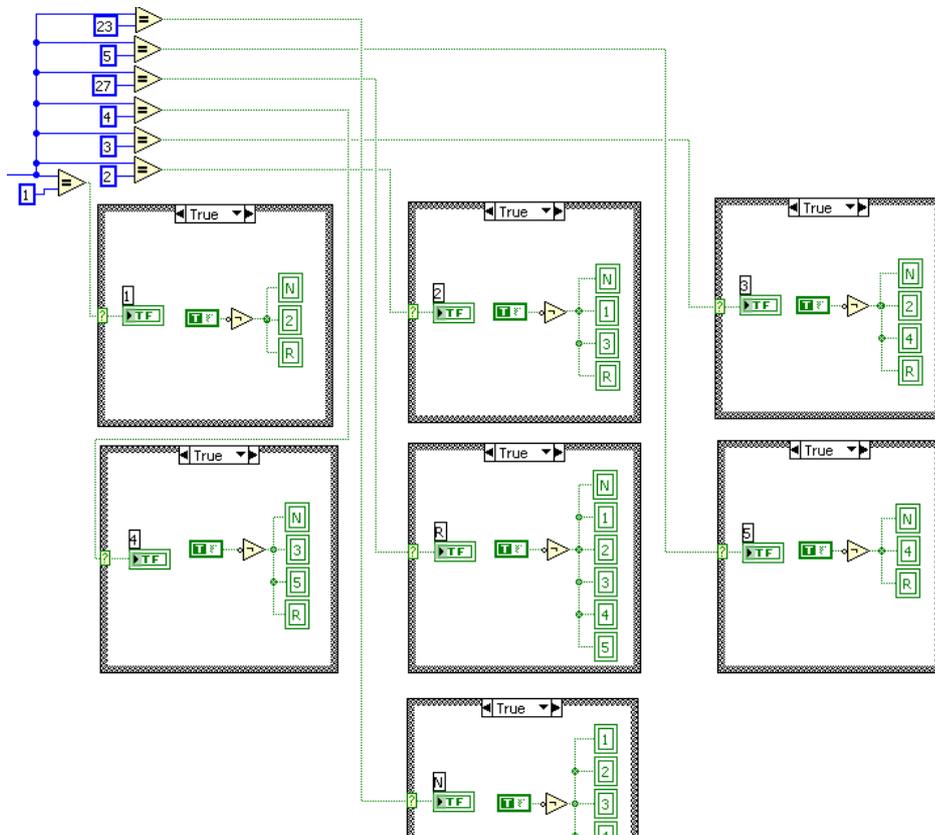


Figura 3.52. Comparación de marcha y activación del led.

En la figura 3.53 se presenta el panel frontal de la página “CONSULTA”, este presenta tres opciones, la primera permite adquirir el reporte de datos en una tabla con los siguientes ítems: tiempo en segundos, marcha, RPM, torque en Lbf.ft y potencia en HP; la segunda opción guarda estos datos en los reportes generales (un reporte guardado tras de otro reporte guardado en un mismo archivo) generado desde LabView (C:\consulta.xls) y la tercera opción guarda estos datos en un reporte individual ingresando en éste la dirección y el nombre del archivo.

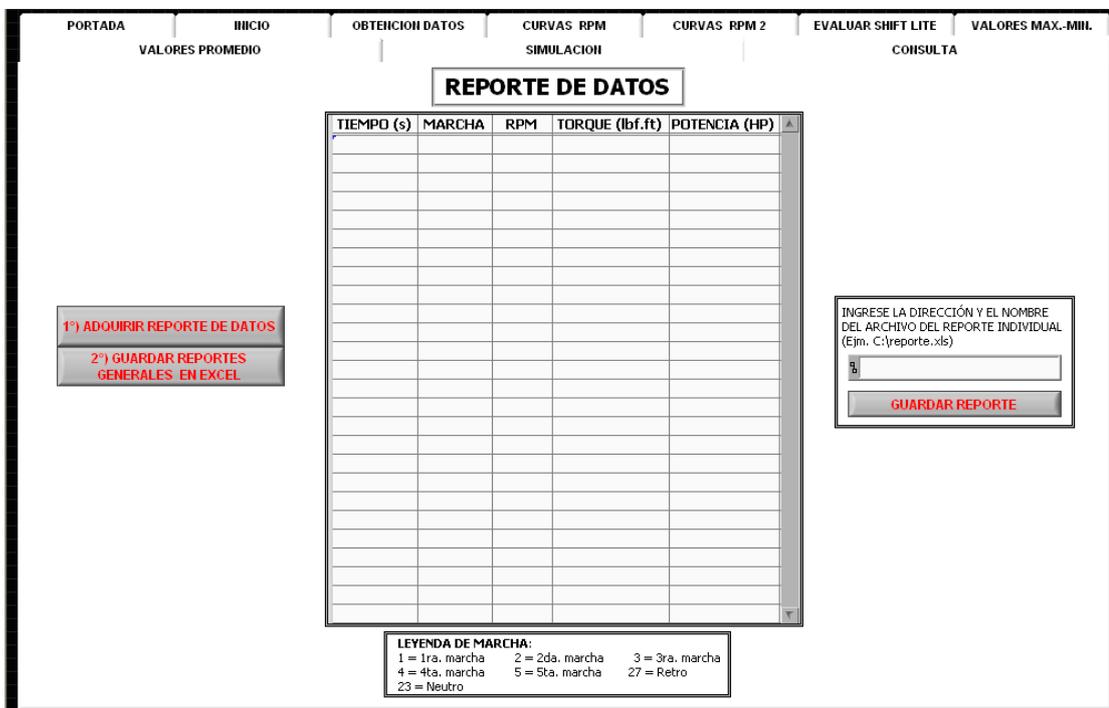


Figura 3.53. Ventana para consultar reporte de datos.

Para almacenar los datos del reporte es necesario una estructura Case Booleana, es decir, en el momento que se presiona el botón denominado DATOS realice una agrupación en una matriz de 1 x 5; estos valores son ordenados según se requiera; también se requiere la función que convierte un arreglo de valores numéricos a un arreglo de cadenas que son desplegadas en una tabla (Figura 3.54).

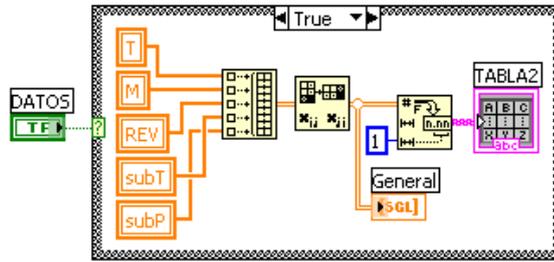


Figura 3.54. Programa para adquirir reporte de datos.

La figura 3.55 muestra los indicadores de tiempo, marcha, RPM, torque y potencia; se toma el indicador denominado RPM cuatro, que indica el número de caracteres de todas las RPM, este valor se divide para cuatro y con este número se especifica las veces que se va a ejecutar el ciclo For. La terminal de iteración (i) es multiplicada por el valor que se requiere; este parámetro determina el comienzo de separación de datos y agrupa en subconjuntos almacenados en una matriz de 1 x 9 estos datos son almacenados al archivo creado en Excel.

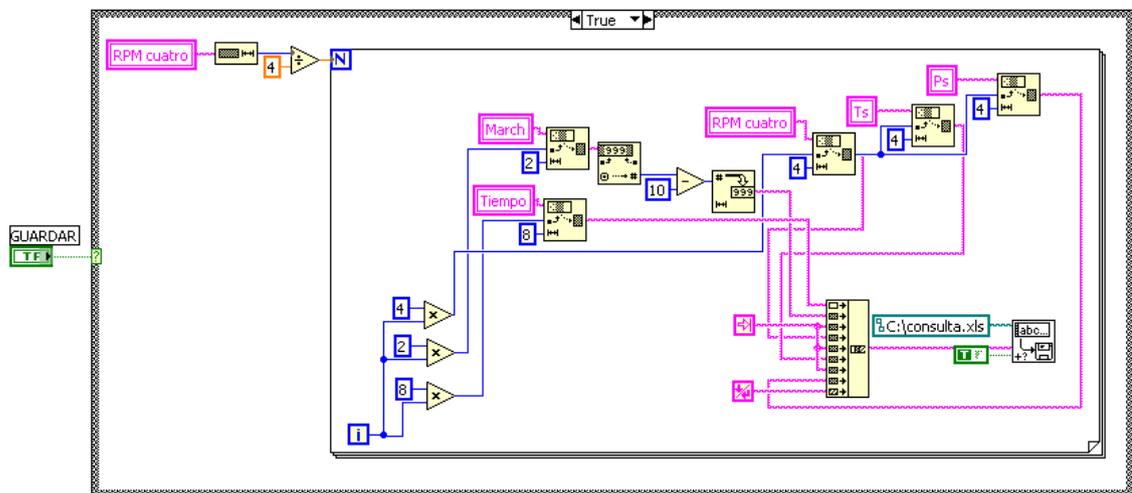


Figura 3.55. Programa para guardar el reporte de datos.

### 3.24 INSTALACIÓN DEL MÓDULO E INTERFACES EN EL VEHÍCULO.

A continuación se enumeran los pasos para el montaje en el vehículo:

1. Se instaló el cableado de alimentación de 12V desde la batería hacia el interior del vehículo, esta alimentación será la que usará el módulo tanto para su funcionamiento como para el manejo de sus interfaces (Figura 3.56).



Figura 3.56. Conexión en bobina.

2. Se conectó un cable desde el borne (-) de la bobina hacia la cabina del vehículo para que el módulo con esta señal pueda determinar las RPM a las que gira el motor del vehículo (Figura 3.57).



Figura 3.57. Conexión en batería.

3. Se colocó un interruptor de doble tiro, el mismo que servirá para prender o apagar el módulo. La una línea que se corta es la de alimentación positiva de 12V, mientras que la otra es la línea de señal del borne (-)

de la bobina, esto para ayudar a que no se autoinduzca corriente en el módulo cuando este haya sido apagado (Figura 3.58).



**Figura 3.58. Interruptor doble tiro.**

4. Previa la construcción de su soporte, se instaló los sensores de posición de marcha. Este conjunto fue empotrado al interior del vehículo por donde la palanca de cambios realiza su desplazamiento normal (Figura 3.59).



**Figura 3.59. Instalación de los sensores de posición de marcha.**

5. Se implementó el cableado de alimentación de masa hacia todos los sensores, como también el cableado de retorno con las señales de salida de estado (activo/inactivo) de cada uno de los sensores, los mismos que ayudarán al módulo a detectar la posición de la palanca de cambios (Figura 3.60).



**Figura 3.60. Implementación de cableado hacia los sensores de posición.**

6. Se montó los shift lite (luz indicadora) sobre el tablero del vehículo, consta de 3 cables, uno de masa común, un positivo para la luz de subir marcha y otro positivo para la luz de bajar marcha (Figura 3.61).



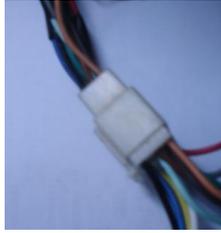
**Figura 3.61. Montaje de los shift lite.**

7. Los dos cables positivos se conectaron hacia el módulo para que se alimenten de acuerdo a su funcionamiento. Al cable de masa común se le implementó un potenciómetro para que el conductor pueda graduar el brillo de los leds de la luz indicadora de acuerdo a su comodidad (Figura 3.62).



**Figura 3.62. Potenciómetro shift lite**

8. Los cables de alimentación, de señal de bobina, de luces indicadoras y de señal de sensores de posición que se conectan al módulo se conectaron a un mismo socket de 12 pines para que sea mucho más fácil su montaje y desmontaje (Figura 3.63).



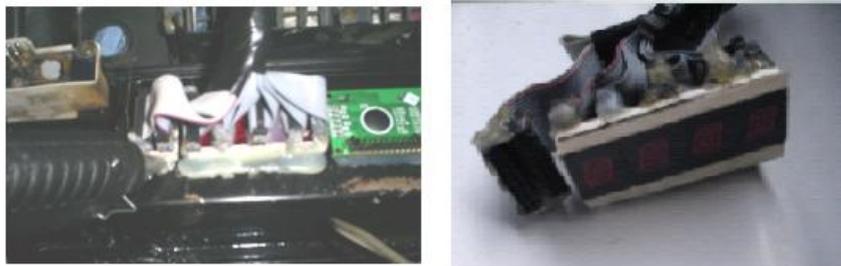
**Figura 3.63. Soque pcb.**

9. Se acondicionó el tablero del vehículo para poder montar los displays, la LCD y el teclado que son las interfaces entre el conductor y el módulo (Figura 3.64).



**Figura 3.64. Acondicionamiento del tablero del vehículo.**

10. Implementamos todo el cableado desde los displays, LCD y teclado hacia el módulo para su funcionamiento (Figura 3.65).



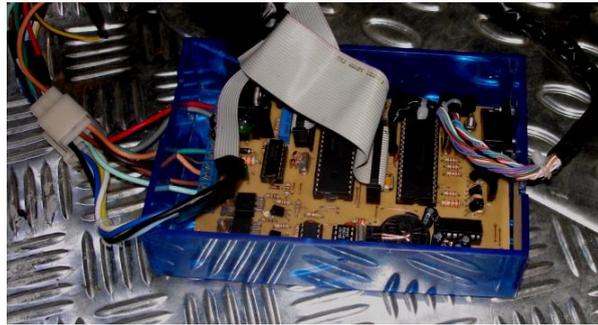
**Figura 3.65. Implementación del cableado.**

11. Una vez que se construyó la placa electrónica del módulo y se soldaron todos sus elementos, se fijó esta placa en una caja metálica para su alojamiento (Figura 3.66).



**Figura 3.66. Placa y alojamiento.**

12. Se conectó el socket con señales de alimentación y entrada, el socket para el teclado, el socket para LCD y los sockets para los displays multiplexados (Figura 3.67).



**Figura 3.67. Placa en alojamiento.**

13. Como último paso se sujetó la caja de alojamiento del módulo a la carrocería del vehículo (Figura 3.68).



**Figura 3.68. Placa instalada en vehículo.**

## **IV. CAPITULO**

### **PRUEBAS EXPERIMENTALES DE OPERACIÓN DEL MÓDULO DE ALMACENAMIENTO Y ANÁLISIS DE LAS RPM Y CAMBIO DE MARCHA DE UN VEHÍCULO.**

#### **4.7 PRUEBA DE FUNCIONAMIENTO DE LOS SENSORES DE MARCHA.**

El sensor de marcha simplemente es un interruptor del tipo normalmente cerrado, quiere decir que cuando su vástago no es pulsado el sensor conducirá, mientras cuando el vástago del sensor es pulsado deja de conducir. Todos los sensores de marcha están alimentado con masa es decir 0 lógico; de esta manera el módulo detecta la marcha, pues cuando la palanca de cambios está enclavada en una determinada marcha el vástago del sensor de esa marcha es pulsado, deja de conducir y el módulo receptorá un 1 lógico ya que todas las señales de entrada de los sensores tienen resistencias Pull\_Up. Mientras que el resto de sensores que no son pulsados estarán en estado de conducción, entonces el modulo los reconocerá como un 0 lógico.

Cabe indicar que en la marcha de retro se maneja lógica inversa pues su sensor está montado directamente en la caja de cambios, cuando el conductor enclava retro el vástago del sensor deja de estar pulsado y deja de conducir, el módulo identificará un 1 lógico; mientras que cuando la palanca de cambios este en cualquier otra marcha el vástago es pulsado y conduce entonces el sensor enviara un 0 lógico al módulo.

Entonces podemos resumir que si el sensor conduce tendremos un 0 lógico y cuando no conduce tendremos un 1 lógico en el módulo. En la tabla IV.1 se muestra las combinaciones de los estados de los sensores para cada una de las marchas.

**Tabla IV.1. Combinaciones de los estados de los sensores.**

ESTADO LÓGICO DE LOS SENSORES DE MARCHA					MARCHA ACTIVA
1RA.	2DA.	3RA.	4TA.	RETRO	
1	0	0	0	1	Primera Velocidad
0	1	0	0	1	Segunda Velocidad
0	0	1	0	1	Tercera Velocidad
0	0	0	1	1	Cuarta Velocidad
0	0	0	0	0	Retro
0	0	0	0	1	Neutro

La prueba externa que se hará para ver si los sensores eléctricos están trabajando correctamente de acuerdo a la tabla anterior, será enclavar cada marcha y medir continuidad con un multímetro entre masa y cada una de las líneas de salida de los sensores que están en el socke principal.

Luego de realizar esta prueba se obtuvieron los siguientes resultados (Tabla IV.2).

**Tabla IV.2. Resultados de la prueba externa.**

ESTADO DEL SENSOR	Sensor 1ra.		Sensor 2da.		Sensor 3ra.		Sensor 4ta.		Sensor Retro		FUNCIONAMIENTO
	Conduce	No conduce	Conduce	No conduce							
1ra		x	x		x		x			x	correcto

2da	X			x	x		x			x	correcto
3ra	X		x			x	x			x	correcto
4ta	X		x		x			x		x	correcto
Retro	X		x		x		x		x		correcto
Ninguna	X		x		X		x			x	correcto

El funcionamiento de los sensores en cada una de las marchas fue el correcto, están listos para ser implementados en el módulo

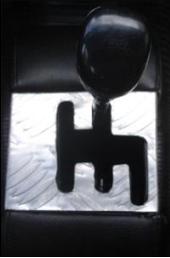
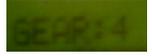
#### 4.8 PRUEBA DE RECEPCIÓN Y VISUALIZACIÓN DE SEÑALES EN EL MÓDULO.

Una vez que instalamos el módulo en el vehículo podemos empezar estas pruebas, primero enclavaremos cada una de las marchas del vehículo y se verificará que tanto el valor mostrado en la LCD y en el display corresponda al valor de marcha que el conductor haya enclavado ese momento.

Los resultados que se obtuvieron se expresan en la tabla IV.3.

**Tabla IV.3. Resultado de la prueba de recepción y visualización de señales en el módulo.**

Marcha Enclavada	GRÁFICAS DE:			FUNCIONAMIENTO
	Palanca de Cambios	Display del Módulo	Lcd del Módulo	
1ra				correcto

2da				correcto
3ra				correcto
4ta				correcto
Neutro				correcto
Retro				correcto

La prueba de muestreo de marcha fue cumplida satisfactoriamente.

La siguiente prueba será visualizar el valor de RPM en los 4 display y en la LCD del módulo y verificar que la señal obtenida del chispazo de bobina sea multiplicada por el respectivo factor, de acuerdo al número de cilindros, ya que en la configuración del módulo el conductor puede escoger el número de

cilindros de su vehículo ya sea 3, 4, 6, 8 cilindros para motores gasolina, o 2 para el caso de motores diesel. El cálculo de las RPM viene dado por la siguiente fórmula (Ec 4.1):

$$N = 120 \left( \frac{f}{\#cilindros} \right) \quad (\text{Ec. 4.1})$$

Donde:

N: Revoluciones por minutos.

f: Número de chispazos por segundo.

Entonces para conocer las RPM del vehículo multiplicaremos el número de chispazos por segundo por un factor  $\left( \frac{120}{\#cilindros} \right)$ , dependiendo el número de cilindros el factor de multiplicación se indica en la tabla IV.4.

**Tabla IV.4. Factor de multiplicación.**

Tipo de Motor	Factor Multiplicación
3 cilindros	40
4 cilindros	30
6 cilindros	20
8 cilindros	15
Diesel	60

Luego de conocer los factores de multiplicación procedemos a realizar la prueba en ralentí del vehículo para cada uno de los cilindros. Cabe recalcar que el módulo fue diseñado para aumentar su valor entre centena y centena. Los valores que se obtuvieron se muestran en la tabla IV.5.

**Tabla IV.5. Resultados de la prueba de ralentí.**

RÉGIMEN MOTOR	f(#puls/seg)	3 cilindros RPM	4 cilindros RPM	6 cilindros RPM	8 cilindros RPM	Diesel RPM
Ralentí	28-32	1100-1300	800-1000	500-600	400-500	1700-1900

Como se puede observar en la tabla a un mismo régimen de giro del motor del vehículo se puede obtener una lectura de RPM diferente, dependiendo de la configuración de número de cilindros que se hizo. Los valores obtenidos de RPM en la tabla si variaron correctamente de acuerdo al factor de número de cilindros.

#### **4.9 PRUEBA DE COMPARACIÓN DE LECTURAS DE RPM ENTRE EL MÓDULO Y EL MULTÍMETRO DE USO AUTOMOTRIZ.**

Existen algunos instrumentos externos para medir las RPM del vehículo tales como tacómetros análogos o multímetros de uso automotriz. Decidimos usar un multímetro de uso automotriz para comparar sus lecturas de RPM con las lecturas de nuestro módulo y así determinar si el valor de RPM que nos da el módulo son correctas. En la tabla IV.6 se expresan los valores que se obtuvieron.

**Tabla IV.6. Resultado de la prueba de comparación de lecturas de RPM.**

Lectura de RPM en:		Diferencia de RPM entre los instrumentos
Módulo	Multímetro Automotriz	
900	810	80
1500	1430	70
2200	2110	90
3300	3130	70
Promedio Diferencia RPM		77.5 RPM

Como se puede observar en la tabla, el número de revoluciones en que difieren ambos valores de lectura tomados está entre 70-90 rpm aproximadamente, esta diferencia mínima se da por el redondeo en centenas que hace el módulo o ya sea por los ruidos eléctricos que se generan al interior del vehículo.

#### 4.10 PRUEBA DE COMUNICACIÓN ENTRE EL MÓDULO Y EL PC.

##### 4.4.1. Pruebas de la transmisión serial.

Con la ayuda del programa SerialCom incluido en la carpeta de MicroCode Studio se recibió una cadena de caracteres, con lo que se pudo comprobar el correcto funcionamiento de la interfaz serial.

En la figura 4.1 se muestra la llegada de datos al SerialCom.

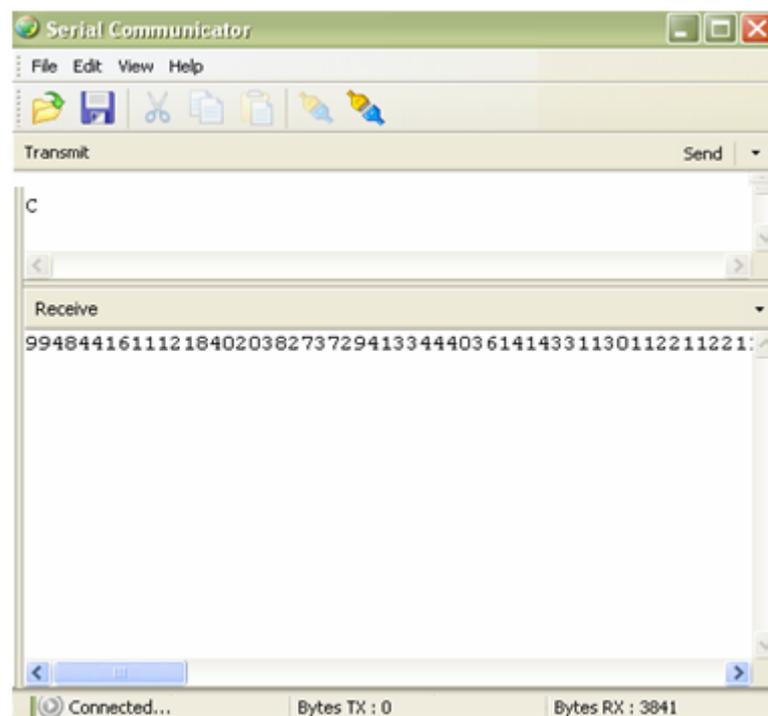


Figura 4.1. SerialCom, pruebas de la transmisión serial.

Cada una de las cadenas correspondientes optó por sumar 10 al valor leído por el conversor ADC.

Los datos en cadenas se la examinarán en el ítem correspondiente a las pruebas del software (ítem 4.6.2).

#### 4.11 DESCRIPCIÓN DE MANEJO DEL MÓDULO.

A continuación se indica rápidamente las funciones que tiene el teclado en cada uno de los menús que nos despliega el módulo en la LCD.

Cuando encendemos el módulo observaremos este par de pantallas que serán los mensajes de inicialización del módulo (Figura 4.2).

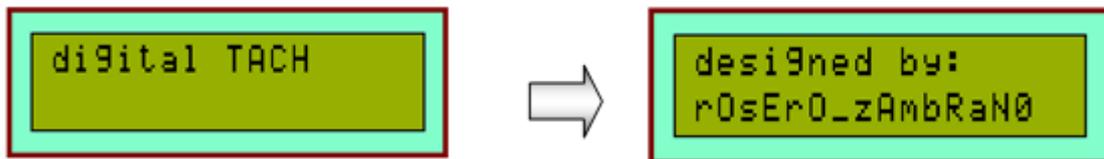


Figura 4.2. Mensajes de visualización del módulo.

La primera pantalla de menús será la figura 4.4.



Figura 4.3. Visualización de la primera pantalla.

En todos los menús las funciones de teclado son las siguientes:



Sirve para desplazarnos al menú anterior.



Nos ayuda a desplazarnos al siguiente menú.



Equivale a un (+) sirve para aumentar el número de RPM en la configuración de regímenes de cambio de marcha de shift lite.



Equivale a un (-) sirve para disminuir el número de RPM análogamente a la tecla anterior.

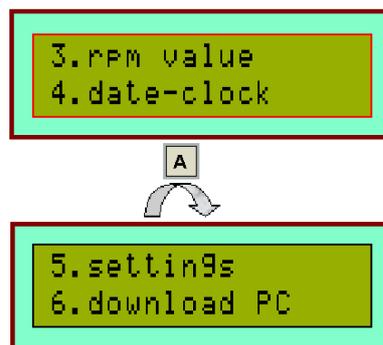


Es similar a la tecla ESC de un computador nos ayuda a salir de un menú sin guardar los cambios y nos ubica en un nivel superior de menú.



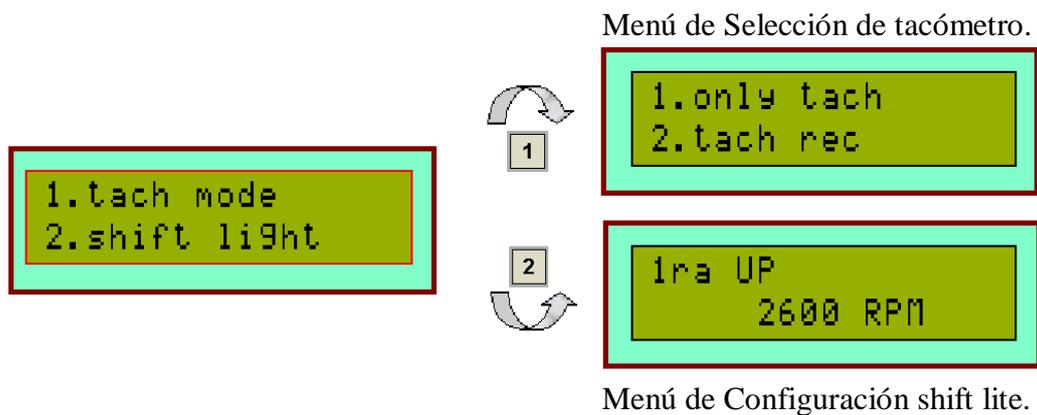
Su función es la de un ENTER, en un menú inferior nos ayuda a guardar los últimos cambios y nos ubica en un menú superior.

Un ejemplo de desplazamiento desde arriba hacia abajo con la tecla A lo indicamos en la figura 4.4.



**Figura 4.4. Ejemplo de desplazamiento con la tecla A.**

Ahora, las teclas del 1 al 9 sirven para que el usuario ingrese en un menú inferior dependiendo del número que lleve la etiqueta de la opción que se desee escoger. En la figura 4.5 tenemos un ejemplo, usando la tecla 1 ingresamos a la selección del tipo de tacómetro y con la tecla 2 ingresamos a la configuración de la shift lite.

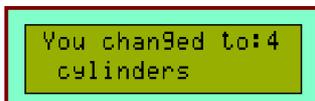


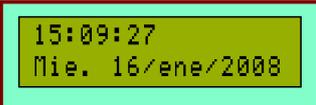
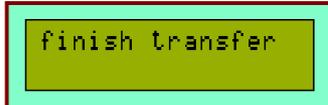
**Figura 4.5. Ejemplo de desplazamiento en el teclado.**

En la tabla IV.7 describimos las funciones de los principales submenús.

**Tabla IV.7. Funciones de los submenús.**

Nombre del Menú	Descripción	Gráfica
Only Tach	Aquí nos indica las RPM y número de marcha del vehículo.	
Tach Rec	La misma función que Only Tach pero además guarda cada 0.5 seg. las RPM y marcha del vehículo en la memoria EEPROM del módulo.	
2 UP	Aquí se configura el número de RPM a la que se quiere que la luz de subir marcha se encienda dependiendo de la marcha, en este ejemplo 2da. con las teclas C y D se sube y baja las RPM respectivamente. Luego con la tecla # grabamos el cambio.	
2 DOWN	Análogamente que en 2 up,	

	<p>aquí se configura el numero de RPM a la que se quiere que la luz de bajar marcha se encienda. El manejo de teclas es el mismo que 2 up.</p>	
1,3,4,5 UP Y Rtro	<p>Misma función y funcionamiento que 2 up, solo cambia la marcha a configurar.</p>	
1,3,4,5 DOWN	<p>Misma función y funcionamiento que 2 down, solo cambia la marcha a configurar.</p>	
Rpm_Value	<p>En este menú nos indicará la máxima revolución que alcanzó el motor, la última vez que se uso el tacómetro.</p>	
No cylinders	<p>Aquí el usuario puede configurar el número de cilindros de su vehículo. Si quiere modificarlo simplemente debe presionar el número de tecla que corresponda a su número de cilindros. Son validos 1 (diesel) 3, 4, 6 y 8 cilindros. Usando tecla “#” grabamos el nuevo valor, caso contrario con “*” salimos del menú sin guardar los cambios.</p>	 
No Gears	<p>El usuario puede configurar el número de marchas de su vehículo. Usando tecla que</p>	

	<p>corresponda al número de marchas cilindros. Son validos 4 y 5 marchas. Usando tecla “#” grabamos el nuevo valor, caso contrario con “*” salimos del menú sin guardar los cambios.</p>	
Date_ Clock	<p>En esta opción el usuario puede visualizar la hora y fecha actual. Además puede modificar estos parámetros usando el teclado. Con 1 la hora, con 2 los minutos; con 4 el día de la semana; con 5 día del mes, con 6 el mes; con B el año. El modulo automáticamente evalúa los años bisiestos y días de 30 y 31 días. Con la tecla # grabamos los cambios</p>	 
Down load PC	<p>Nos sirve para transferir todos los datos de la memoria EEPROM del módulo hacia el computador. El código que nos da debemos ingresarlo en el computador y se adquirirá los datos.</p>	 

## 4.12 PRUEBA Y DESCRIPCIÓN DE FUNCIONAMIENTO DEL SOFTWARE DE VISUALIZACIÓN Y ANÁLISIS DE DATOS EN LABVIEW.

### 4.6.1 Pruebas de la Interface de Descarga.

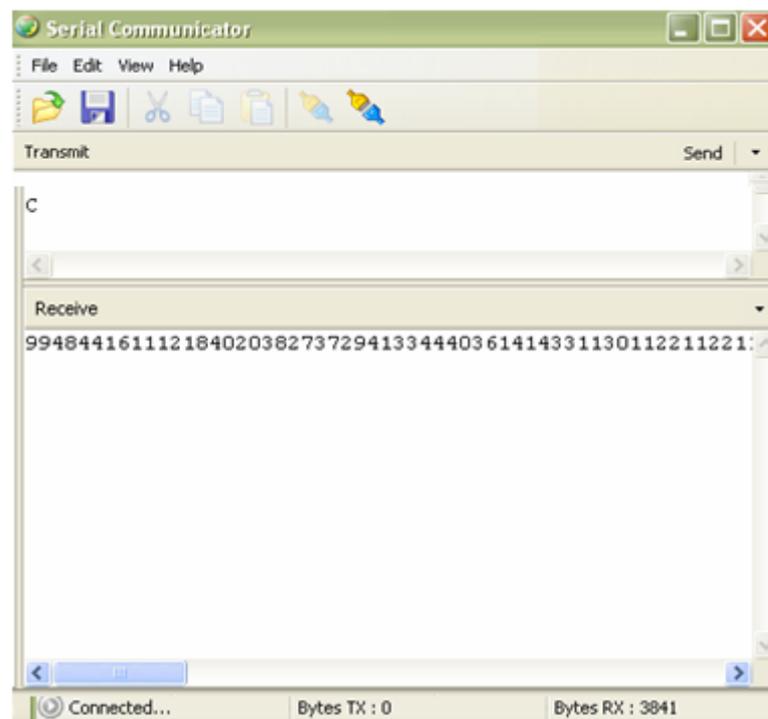
La interface de descarga tiene el objetivo de recuperar los datos guardados en la memoria I2C y enviarlos hacia un programa hecho en LabView, el mismo

que manipula convenientemente estos datos. El módulo de descarga se divide en dos partes hardware y software.

#### 4.6.1.1 Prueba del Hardware.

Esta prueba fue realizada con el fin de comprobar que el módulo esté enviando los datos y consistió en lo siguiente:

- a) Abrir el programa SerialCom.
- b) Pulsar el botón de inicio de transmisión.
- c) Revisar que la cadena de datos no presente símbolos extraños.
- d) Si la cadena de datos de principio a fin es numérica, se podrá concluir que el hardware está funcionando correctamente. En la figura 4.6 se puede observar la cadena de datos recibida.



**Figura 4.6. SerialCom, prueba del hardware.**

En la figura 4.6 se observa que la cadena de datos no presenta errores, en la parte inferior de la misma se puede ver la configuración del puerto para la recepción de datos (9600-8-N-1), lo cual significa 9600 bps de velocidad, cada digito a ocho bits, sin paridad, un BIT de parada.

Se puede concluir que el hardware funciona correctamente.

#### 4.6.1.2 Prueba del software.

Para comprobar el correcto funcionamiento del software se decidió seguir los siguientes pasos.

- a) Abrir el programa en LabView y ponerlo en modo de ejecución.
- b) Pulsar el botón de transferencia de datos para realizar la descarga.
- c) Comprobar que la cadena de datos recibida en LabView sea la misma que se recibió en el programa SerialCom de MicoCode Studio.
- d) Abrir el archivo creado y comprobar que los datos estén separados correctamente.

En la figura 4.7 se muestra la cadena de datos recibida en LabView y la figura 4.6 que contiene la cadena recibida en el SerialCom.

VALORES PROMEDIO | SIMULACION | CONSULTA  
PORTADA | INICIO | OBTENCION DATOS | CURVAS RPM | CURVAS RPM 2 | EVALUAR SHIFT LITE | VALORES MAX.-MIL.

 **"DISEÑO E IMPLEMENTACIÓN DE UN MÓDULO DE ALMACENAMIENTO Y ANÁLISIS DE RPM PARA EL CAMBIO DE MARCHA DE UN VEHÍCULO DATSUN 160J"**

**DESCARGA DE DATOS CON ENLACE A EXCEL**

1º INGRESE EL CÓDIGO DE TRANSFERENCIA:  2º PULSE EL BOTÓN PARA DESCARGAR:

**RECEPCIÓN DE DATOS:**

Datos1 (hora y fecha: son 14 caracteres)  Datos2 (régimenes de marcha): son 22 caracteres  Datos3 (datos generales: son 4 caracteres)

Cadena (RPM y marchas)

Figura 4.7. Cadena de datos recibida en LabView.

Si se compara las cadenas recibidas en estos dos programas, se observa que son iguales, por lo que se puede concluir que el software recibe correctamente los datos desde el puerto serie.

En la figura 4.8 se observa como fueron separados y guardados los datos en la página de Excel (C:\reporte.xls):

Se puede apreciar en la figura 4.8 como los datos que anteriormente en la figura 4.7 se recibió en forma continua, ahora se los tiene por separado y transformados, se observa claramente que los primeros catorce dígitos (primer cuadro de texto denominado Dato1) son separados en longitudes de dos y restados diez. Estos datos corresponden a los segundos, minutos, hora, día de del la semana, día del mes, mes y año. Para comprobar la veracidad de los datos del primer cuadro de texto (Datos1) de la figura 4.7 (99484416111218), deberán observarse en Excel desde la cuarta fila y columnas correspondientes de la siguiente manera: Segundos (=59), Minutos (=26), Hora (=22), Día Semana (=6), Día Mes (=1), Mes (=2) y Año (=8).

En el segundo cuadro de texto llamado Datos2 (4020382737294133444036) de la figura 4.7, el número de caracteres a recibir es de veinte y dos correspondiendo al contenido de once variables, estos datos son separados en longitudes de dos, restados diez y multiplicados por cien; correspondientes a los regímenes de marcha en RPM: primera UP, segunda UP, segunda DW, tercera UP, tercera DW, cuarta UP, cuarta DW, quinta UP, quinta DW, retro UP, máxima RPM. Para comprobar su veracidad, deberán observarse en Excel desde la sexta fila y columnas correspondientes de la siguiente manera: Primera UP (=3000), Segunda DW (=1000), Segunda UP (=2800), Tercera DW (=1700), Tercera UP (=2700), Cuarta DW (=1900), Cuarta UP (=3100), Quinta DW (=2300), Quinta UP (=3400), Retro UP (=3000), Máxima RPM (=2600).

En el tercer cuadro de texto recibe cuatro caracteres denominado Datos3 (1414) de la figura 4.7, estos datos son separados en longitudes de dos y restados diez; caracteres correspondiendo al contenido de dos variables que se requiere para el número de cilindros y el número de marchas. Para comprobar su veracidad se observa en Excel desde la octava fila y columnas correspondientes de la siguiente manera: N. Cilindro (=4), N. Marcha (=4).

En el cuarto cuadro de texto de la figura 4.7 llamado Cadena (33113011221122112211231122112411261126112611271129332833193319122312251229122912301230123012301230123112301231123233303325332513231332133313331334133413361334133513351336133613333329332233271433143414351435143414351435143514351435143414), el número de caracteres a recibir es indeterminado; esto depende del tiempo total de grabación que desea el conductor, debido a que cada RPM con su respectiva marcha se graba cada 0,5 segundo. Estos datos se separan en grupos de cuatro y a cada grupo nuevamente se separa en longitudes de dos para ser restados diez; finalmente la primera longitud de dos es multiplicada por cien. Para comprobar que el dato guardado de marcha y RPM son los correctos se tomó de la figura 4.7 los caracteres del primer grupo de cuatro (3311), este valor se separa en longitudes de dos correspondiendo a los números 33 y 11, luego se resta diez obteniendo 23 y 1, por último, la primera longitud es multiplicada por cien teniendo el valor de 2300. Adquiriendo así los datos reales de marcha y su RPM; en este caso primera marcha (1) con su respectiva RPM (2300).

De esta forma queda demostrado que los datos almacenados en el documento en Excel son correctos, ya que el valor de 1 corresponde a la primera marcha que fue almacenado en la hoja de Excel en la posición B10; y el valor de 2300 que es la RPM de dicha marcha almacenada en la posición C10.

ESPE-L																	
ALMACENAMIENTO Y ANÁLISIS DE RPM																	
Segundos	Minutos	Hora	Día Semana	Día Mes	Mes	Año											
59	26	22	6	1	2	8	Primera UP.	Segunda DW	Segunda UP.	Tercera DW.	Tercera UP.	Cuarta DW.	Cuarta UP.	Quinta DW.	Quinta UP.	Retro UP.	Max.RPM
3000	1000	2800	1700	2700	1900	3100	2300	3400	3000	2600							
N. Cilindro.	N. Marcha																
4	4																
Tiempo (s)	Marcha	RPM	Torque lbf.ft	Potencia HP													
0	1	2300	141,801	62,207													
0,5	1	2000	148,675	56,666													
1	1	1200	141,78	32,415													
1,5	1	1200	141,78	32,415													
2	1	1200	141,78	32,415													
2,5	1	1300	145,164	36,008													
3	1	1200	141,78	32,415													
3,5	1	1400	146,808	39,162													
4	1	1600	149,295	45,518													
4,5	1	1600	149,295	45,518													
5	1	1600	149,295	45,518													
5,5	1	1700	150,442	48,79													
6	23	1900	149,878	54,322													
6,5	23	1800	150,417	51,595													
7	23	900	120,41	20,686													
7,5	2	900	120,41	20,686													
8	2	1300	145,164	36,008													
8,5	2	1500	148,1	42,383													
9	2	1900	149,878	54,322													
9,5	2	1900	149,878	54,322													
10	2	2000	148,675	56,666													
10,5	2	2000	148,675	56,666													
11	2	2000	148,675	56,666													
11,5	2	2000	148,675	56,666													
12	2	2000	148,675	56,666													
12,5	2	2100	146,835	58,817													

Figura 4.8. Página de datos archivo “reporte”.

Una vez realizadas todas las pruebas se puede concluir que el módulo de descarga en su totalidad funciona correctamente.

#### 4.6.2. Prueba de la recolección de datos (Excel).

Para realizar esta prueba, el módulo almacenó datos dos veces con diferentes parámetros para adquirir en el sistema la recolección de datos totales. El software de monitoreo de forma automática debe crear un archivo denominado “consulta”, en la siguiente dirección de memoria del disco duro “C:\consulta.xls” el mismo debe presentar un formato adecuado para ser abierto en Excel, en la figura 4.9 se muestra el resultado obtenido.

	A	B	C	D	E	F	G	H	I	J	K	L
62	26	4	2400	139,571	63,84							
63	26,5	4	2500	137,244	65,441							
64	27	4	2500	137,244	65,441							
65	27,5	4	2500	137,244	65,441							
66	28	4	2500	137,244	65,441							
67	28,5	4	2500	137,244	65,441							
68	29	4	2400	139,571	63,84							
69	ESPE-L											
70	ALMACENAMIENTO Y ANÁLISIS DE RPM											
71	Segundos	Minutos	Hora	Día Semana	Día Mes	Mes	Año					
72	59	26	22	6	1		8					
73	Primera UP.	Segunda DW	Segunda UP.	Tercera DW.	Tercera UP.	Cuarta DW.	Cuarta UP.	Quinta DW.	Quinta UP.	Retro UP.	Max.RPM	
74	3000	1000	2800	1700	2700	1900	3100	2300	3400	3000	2600	
75	N. Cilindro.	N. Marcha										
76	4	4										
77	Tiempo (s)	Marcha	RPM	Torque lbf.ft	Potencia HP							
78	0	1	2300	141,801	62,207							
79	0,5	1	2000	148,675	56,666							
80	1	1	1200	141,78	32,415							
81	1,5	1	1200	141,78	32,415							
82	2	1	1200	141,78	32,415							
83	2,5	1	1300	145,164	36,008							
84	3	1	1200	141,78	32,415							
85	3,5	1	1400	146,808	39,162							
86	4	1	1600	149,295	45,518							
87	4,5	1	1600	149,295	45,518							
88	5	1	1600	149,295	45,518							
89	5,5	1	1700	150,442	48,79							
90	6	23	1900	149,878	54,322							
91	6,5	23	1800	150,417	51,595							
92	7	23	900	120,41	20,686							
93	7,5	2	900	120,41	20,686							
94	8	2	1300	145,164	36,008							
95	8,5	2	1500	148,1	42,383							
96	9	2	1900	149,878	54,322							

Figura 4.9. Datos recolectados en Excel (archivo de reportes generales).

#### 4.6.3. Descripción y comprobación de funcionamiento del software de visualización en LabView.

La opción “PORTADA” muestra la presentación del proyecto y el botón finalizar para salir de manera automática de la ejecución del mismo.

En la parte superior diez opciones denominadas “PORTADA”, “INICIO”, “OBTENCIÓN DATOS”, “CURVAS RPM”, “CURVAS RPM2”, “EVALUAR SHIFT LITE”, “VALORES MAX.MIN.”, “VALORES PROMEDIO”, “SIMULACIÓN”, “CONSULTA”.

La figura 4.10 corresponde a la pestaña “INICIO”; muestra la descarga de datos (recepción de cadenas vía RS232) al instante que se ingrese el código que indique la LCD (opción 6. Download PC), este botón se debe presionar una sola vez y antes de ingresar a cada una de las opciones que se presenta.

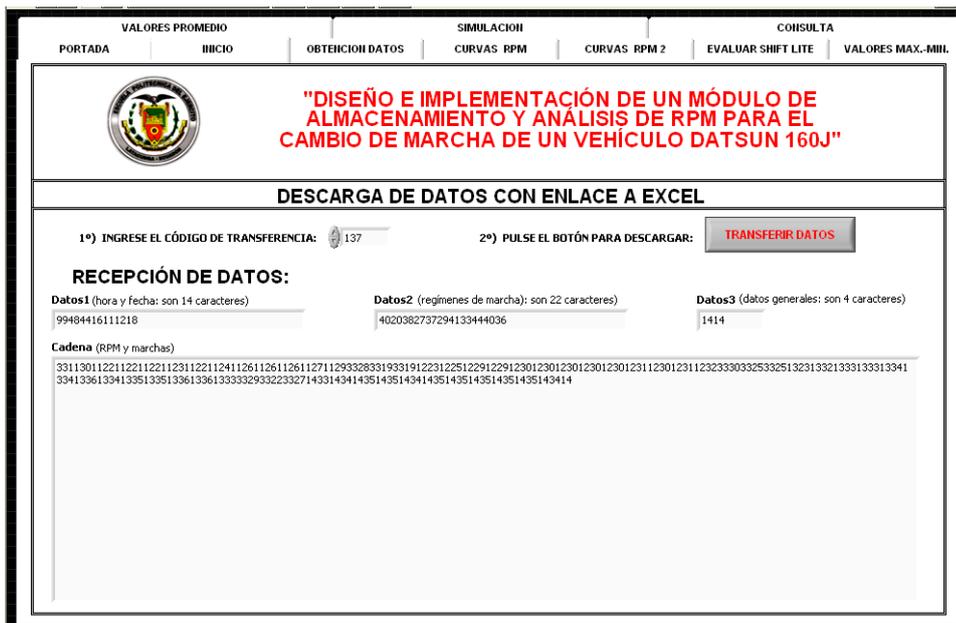


Figura 4.10. Funcionamiento de la ventana “INICIO”.

Una vez que hemos transferido los datos podemos empezar con el análisis de de los datos obtenidos.

En la figura 4.11 presenta la página “OBTENCIÓN DATOS”, luego de dar un click en el botón leer memoria, podemos observar que los datos de hora, fecha, regímenes de cambio de marcha, número de cilindros, número de marchas y RPM están separados y organizados correctamente.



Figura 4.11. Funcionamiento de la ventana “OBTENCIÓN DATOS”.

La figura 4.12 presenta el panel frontal de la página “CURVAS RPM”, luego de dar un click en el botón graficar, observamos y verificamos que las curvas RPM vs. Tiempo y Marcha vs. Tiempo se han graficado correctamente.

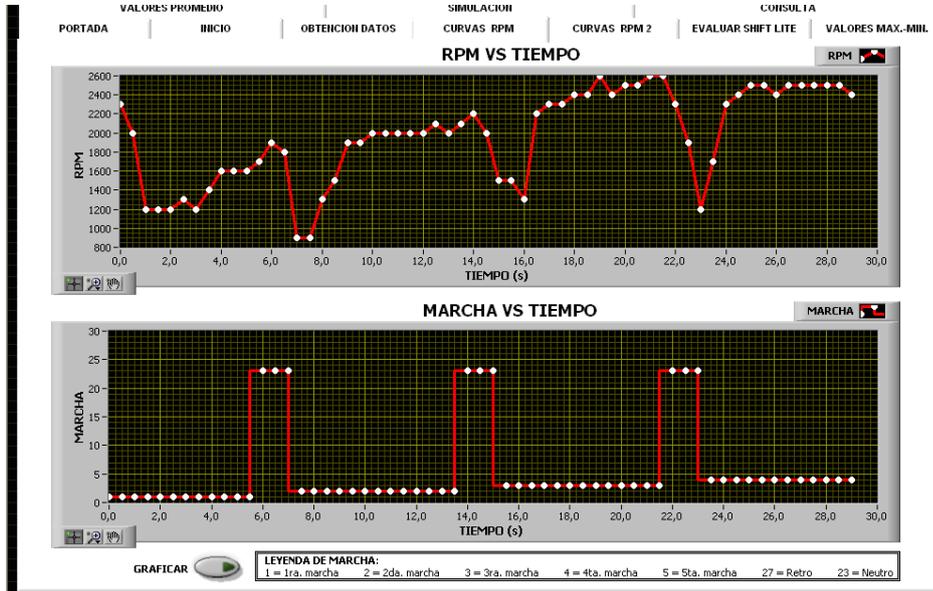


Figura 4.12. Funcionamiento de la ventana “CURVAS RPM”.

La figura 4.13 muestra la página “CURVAS RPM2”, luego de dar un click en el botón graficar, obtenemos las curvas Torque vs. Tiempo y Potencia vs. Tiempo. Las gráficas y los valores obtenidos son correctos.

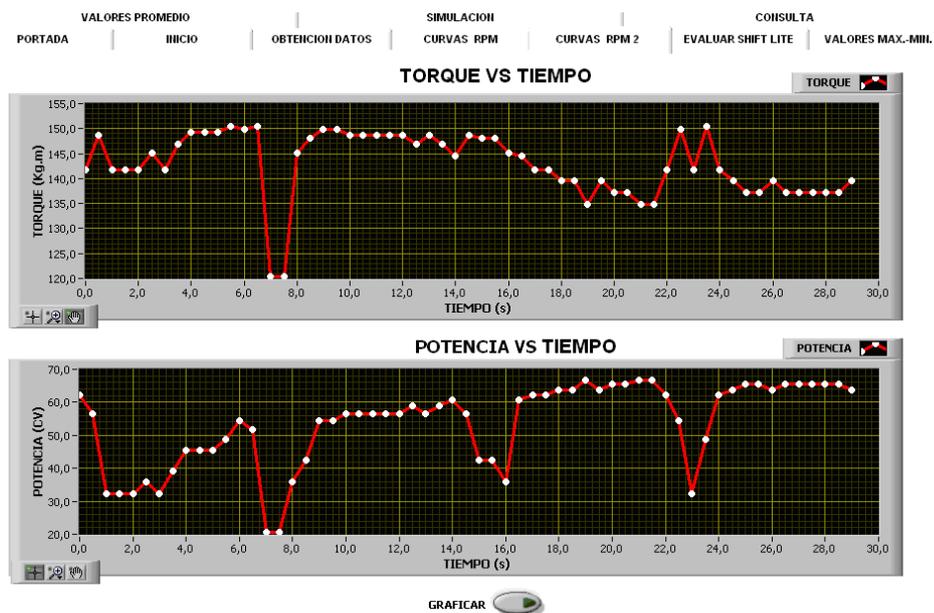


Figura 4.13. Funcionamiento de la ventana “CURVAS RPM2”.

La figura 4.14 presenta la página “EVALUAR SHIFT LITE”, al dar click en el botón evaluar, obtenemos las tablas de evaluación del régimen del giro del motor de cada una de las marchas, la evaluación general de las marchas y total. Los valores obtenidos son correctos.

**EVALUAR USO DE SHIFT LITE**

VALORES PROMEDIO
SIMULACION
CONSULTA

PORTADA
INICIO
OBTENCION DATOS
CURVAS RPM
CURVAS RPM 2
EVALUAR SHIFT LITE
VALORES MAX.-MIN.

**REGIMENES DE MARCHA**

1 UP.	2 DW.	2 UP.	3 DW.	3 UP.	4 DW.	4 UP.	5 DW.	5 UP.	R. max.
3000	1000	2800	1700	2700	1900	3100	2300	3400	3000

(RPM)

**PRIMERA MARCHA**

RPM	RÉGIMEN
2300	1
2000	1
1200	1
1200	1
1200	1
1300	1
1200	1
1400	1

**SEGUNDA MARCHA**

RPM	RÉGIMEN
900	1
1300	2
1500	2
1900	2
1900	2
2000	2
2000	2
2000	2

**TERCERA MARCHA**

RPM	RÉGIMEN
1500	1
1300	1
2200	2
2300	2
2300	2
2400	2
2400	2
2400	2
2400	2

**CUARTA MARCHA**

RPM	RÉGIMEN
1700	1
2300	2
2400	2
2500	2
2500	2
2400	2
2500	2
2500	2

**QUINTA MARCHA**

RPM	RÉGIMEN

**RETRO**

RPM	RÉGIMEN

**EVALUACIÓN GENERAL**

1RA. MARCHA: 1

2DA. MARCHA: 1,9

3RA. MARCHA: 1,8

4TA. MARCHA: 1,9

5TA. MARCHA: 0

**EVALUACIÓN TOTAL DE RÉGIMENES:** 1,7

EVALUAR

**LEYENDA DE RÉGIMEN DE GIRO DEL MOTOR**

3 = Motor sobrerrevolucionado.  
2 = Motor a régimen de giro normal.  
1 = Motor bajo revoluciones.

Figura 4.14. Funcionamiento de la ventana “EVALUAR SHIFT LITE”.

La página “VALORES MAX. MIN.” se indica en la figura 4.15; luego de dar un click en el botón leer se adquiere los valores máximos y mínimos de RPM, torque y potencia correspondiente a cada marcha, también indica los valores generales de los mismos. Los datos obtenidos son correctos.



Figura 4.15. Funcionamiento de la ventana “VALORES MAX.MIN.”.

En la figura 4.16 se indica la página “VALORES PROMEDIO”, en el momento que se presiona el botón obtener se consigue los valores promedio de RPM, torque y potencia correspondientes a cada marcha; también se adquiere el promedio general de estas variables; los datos obtenidos son correctos.

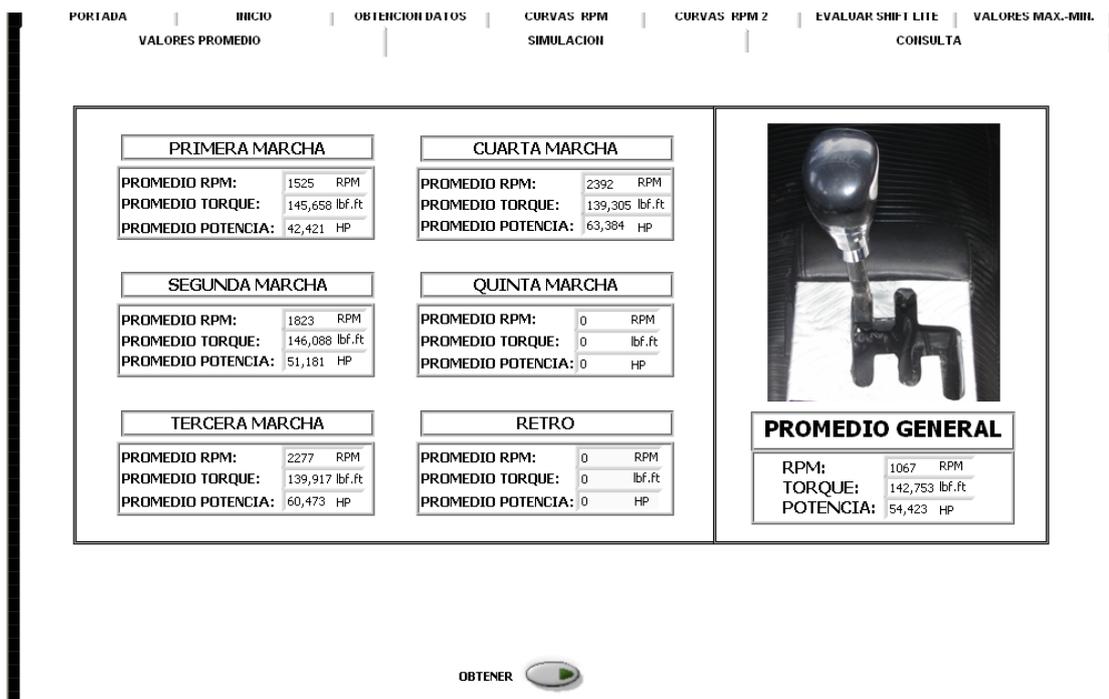


Figura 4.16. Funcionamiento de la ventana “VALORES PROMEDIO”.

La ventana de simulación se indica en la figura 4.17, esta página realiza una simulación en el instante que se ingrese el tiempo en segundos y se presione el botón inicio según los datos grabados, mostrando la marcha en que se encuentra con su respectiva revolución, torque y potencia.

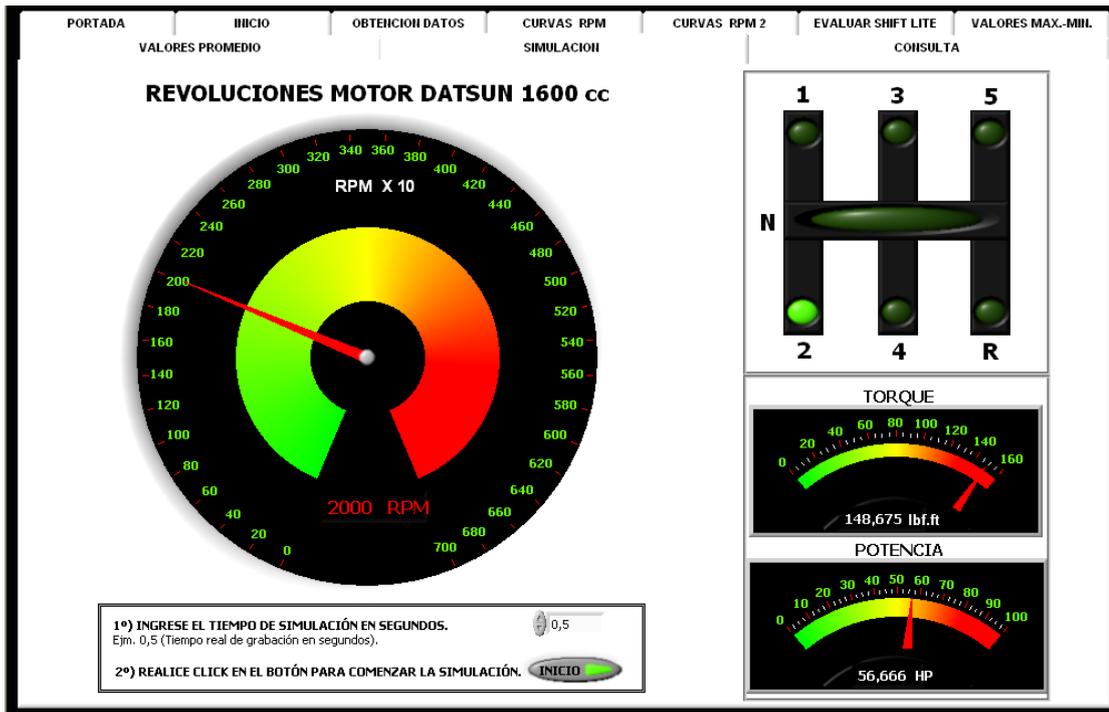


Figura 4.17. Funcionamiento de la ventana “SIMULACIÓN”.

La figura 4.18 se presenta la página “CONSULTA”, al dar un click en la primera opción (botón adquirir reporte de datos) nos muestra los datos de tiempo, marcha, RPM, torque y potencia en una tabla. La segunda opción guarda los datos generales y los datos de la tabla en los reportes generales (C:\consulta.xls); los datos obtenidos son correctos. La tercera opción se ingresa la dirección y el nombre del archivo, en este caso C:\reporte.xls para que guarde los datos de la tabla y los datos generales en un reporte individual. La figura 4.19 y 4.20 corresponde a los archivos de: consulta.xls y reporte.xls respectivamente.

PORTADA INICIO OBTENCIÓN DATOS CURVAS RPM CURVAS RPM 2 EVALUAR SHIFT LITE VALORES MAX.-MIL.

VALORES PROMEDIO SIMULACION CONSULTA

### REPORTE DE DATOS

TIEMPO (s)	MARCHA	RPM	TORQUE (lb.ft)	POTENCIA (HP)
0,0	1,0	2300,0	141,8	62,2
0,5	1,0	2000,0	148,7	56,7
1,0	1,0	1200,0	141,8	32,4
1,5	1,0	1200,0	141,8	32,4
2,0	1,0	1200,0	141,8	32,4
2,5	1,0	1300,0	145,2	36,0
3,0	1,0	1200,0	141,8	32,4
3,5	1,0	1400,0	146,8	39,2
4,0	1,0	1600,0	149,3	45,5
4,5	1,0	1600,0	149,3	45,5
5,0	1,0	1600,0	149,3	45,5
5,5	1,0	1700,0	150,4	48,8
6,0	23,0	1900,0	149,9	54,3
6,5	23,0	1800,0	150,4	51,6
7,0	23,0	900,0	120,4	20,7
7,5	2,0	900,0	120,4	20,7
8,0	2,0	1300,0	145,2	36,0
8,5	2,0	1500,0	148,1	42,4
9,0	2,0	1900,0	149,9	54,3
9,5	2,0	1900,0	149,9	54,3
10,0	2,0	2000,0	148,7	56,7
10,5	2,0	2000,0	148,7	56,7
11,0	2,0	2000,0	148,7	56,7
11,5	2,0	2000,0	148,7	56,7
12,0	2,0	2000,0	148,7	56,7
12,5	2,0	2100,0	146,8	58,8

1) ADQUIRIR REPORTE DE DATOS

2) GUARDAR REPORTE GENERALES EN EXCEL

INGRESE LA DIRECCIÓN Y EL NOMBRE DEL ARCHIVO DEL REPORTE INDIVIDUAL (Ejm. C:\reporte.xls)

**GUARDAR REPORTE**

**LEYENDA DE MARCHA:**

1 = 1ra. marcha    2 = 2da. marcha    3 = 3ra. marcha  
 4 = 4ta. marcha    5 = 5ta. marcha    27 = Retro  
 23 = Neutro

Figura 4.18. Funcionamiento de la ventana "CONSULTA".

Microsoft Excel - consulta.xls

Archivo Edición Ver Insertar Formato Herramientas Datos Ventana ?

ESPE-L

Segundos	Minutos	Hora	Día Semana	Día Mes	Mes	Año					
59	26	22	6	1	2	8					
Primera UP.	Segunda DW	Segunda UP.	Tercera DW.	Tercera UP.	Cuarta DW.	Cuarta UP.	Quinta DW.	Quinta UP.	Retro UP.	Max.RPM	
3000	1000	2800	1700	2700	1900	3100	2300	3400	3000	2600	
N. Cilindro.	N. Marcha										
4	4										
Tiempo (s)	Marcha	RPM	Torque lb.ft	Potencia HP							
0	1	2300	141,801	62,207							
0,5	1	2000	148,675	56,666							
1	1	1200	141,78	32,415							
1,5	1	1200	141,78	32,415							
2	1	1200	141,78	32,415							
2,5	1	1300	145,164	36,008							
3	1	1200	141,78	32,415							
3,5	1	1400	146,808	39,162							
4	1	1600	149,295	45,518							
4,5	1	1600	149,295	45,518							
5	1	1600	149,295	45,518							
5,5	1	1700	150,442	48,79							
6	23	1900	149,878	54,322							
6,5	23	1800	150,417	51,595							
7	23	900	120,41	20,686							
7,5	2	900	120,41	20,686							
8	2	1300	145,164	36,008							
8,5	2	1500	148,1	42,383							
9	2	1900	149,878	54,322							

consulta

Figura 4.19. Reportes generales C:\consulta.xls.

Microsoft Excel - reporte.xls

Archivo Edición Ver Insertar Formato Herramientas Datos Ventana ?

ESPE-L

1	A	B	C	D	E	F	G	H	I	J	K	L
2	ALMACENAMIENTO Y ANÁLISIS DE RPM											
3	Segundos	Minutos	Hora	Día Semana	Día Mes	Mes	Año					
4	59	26	22	6	1	2	8					
5	Primera UP.	Segunda DW	Segunda UP.	Tercera DW.	Tercera UP.	Cuarta DW.	Cuarta UP.	Quinta DW.	Quinta UP.	Retro UP.	Max.RPM	
6	3000	1000	2800	1700	2700	1900	3100	2300	3400	3000	2600	
7	N. Cilindro.	N. Marcha										
8	4	4										
9	Tiempo (s)	Marcha	RPM	Torque lbf.ft	Potencia HP							
10	0	1	2300	141,801	62,207							
11	0,5	1	2000	148,675	56,666							
12	1	1	1200	141,78	32,415							
13	1,5	1	1200	141,78	32,415							
14	2	1	1200	141,78	32,415							
15	2,5	1	1300	145,164	36,008							
16	3	1	1200	141,78	32,415							
17	3,5	1	1400	146,808	39,162							
18	4	1	1600	149,295	45,518							
19	4,5	1	1600	149,295	45,518							
20	5	1	1600	149,295	45,518							
21	5,5	1	1700	150,442	48,79							
22	6	23	1900	149,878	54,322							
23	6,5	23	1800	150,417	51,595							
24	7	23	900	120,41	20,686							
25	7,5	2	900	120,41	20,686							
26	8	2	1300	145,164	36,008							
27	8,5	2	1500	148,1	42,383							
28	9	2	1900	149,878	54,322							
29	9,5	2	1900	149,878	54,322							
30	10	2	2000	148,675	56,666							
31	10,5	2	2000	148,675	56,666							
32	11	2	2000	148,675	56,666							
33	11,5	2	2000	148,675	56,666							
34	12	2	2000	148,675	56,666							
35	12,5	2	2100	146,835	58,817							

reporte /

Figura 4.20. Reporte individual C:\reporte.xls.

## **V. CAPITULO**

### **CONCLUSIONES Y RECOMENDACIONES.**

#### **5.1. CONCLUSIONES.**

- Con la selección del Pic 16F877A, los trompos de retro, la memoria EEPROM 24LC512, el MAX 232, el RTC DS1307 se logró desarrollar satisfactoriamente el diseño electrónico del presente proyecto.
- El tacómetro de este módulo, es una herramienta muy útil para los pilotos de competición para lograr un cajeado óptimo, pues su shift lite se activa a distintos regímenes de giro del motor, evaluando la marcha en la que se halla el vehículo, logrando de esta manera aprovechar al máximo la potencia y torque del vehículo.
- Obteniendo las Curvas de Torque y Potencia en el Banco de Pruebas Dinamométricas, se logró obtener valores reales que nos ayudan a conocer cuando el vehículo tienen un mejor desempeño, además estas curvas se usan para programar las shift lite y simular dichos valores en el software de LabView.
- Usando la comunicación serial asincrónica y el integrado MAX 232 se logró tener un protocolo de comunicación fiable y libre de errores entre el módulo montado en el vehículo y una PC externa.

- Se implementó el software en LabView para analizar los datos recibidos, así como también se generó una base de datos enlazando con Excel, con el fin de llevar un registro de datos que sirva de ayuda para mejorar el desempeño del conductor en un vehículo.
- Se logró determinar que para un conductor es mucho más fácil visualizar las RPM en un tacómetro análogo antes que un tacómetro digital, pues dependiendo del período de muestreo del tacómetro digital su lectura cambia continua y bruscamente, situación que distrae mucho al conductor del vehículo.

## 5.2. RECOMENDACIONES.

- El módulo que se construyó en el presente proyecto, sería una herramienta muy útil para los pilotos de competición, ya que les ayudaría a obtener valores de torque y potencia máximos sin necesidad de sobre revolucionar sus vehículos.
- Es recomendable el uso de memoria externa EEPROM ya que a bajo costo, nos permiten almacenar una gran cantidad de datos con gran fiabilidad y sus ciclos de lectura y escritura no son ninguna limitación para el usuario.
- Se recomienda al Ingeniero Automotriz, el uso del software de programación "Pic Basic", ya que este brinda muchas facilidades en su manejo, con excelentes resultados.
- Una vez que el presente proyecto ha sido sometido a pruebas con resultado satisfactorio, recomendamos usar el presente proyecto, para evaluar el estilo de manejo de un conductor y así evitar que el motor del vehículo sufra un temprano deterioro.
- La distancia alcanzable entre dos aparatos RS232 depende como en todos los métodos de transmisión en serie del cable usado y de la gama de Baudios. Con respecto a longitud del cable se propone que para este proyecto no sean superiores a 2 metros, para que no se pierda datos al momento que se descargue datos.

## **BIBLIOGRAFÍA:**

- Reyes Carlos, Microcontroladores PIC, 1ra Edición, Ayerve C.A, Ecuador, 2005.
- Manual CEAC del Automóvil, ED CEAC S.A., España, 2003.
- Manual del Automóvil Reparación y Mantenimiento Tomo 3, ED Cultural S.A., España, 2003.
- Remache C Darwin, Diseño y Construcción de un Economizador de Gasolina, Latacunga 1991.
- José Angulo, Microcontroladores PIC 2<sup>da</sup> Parte, 2<sup>da</sup> Edición, McGraw Hill, España, 2006.
- Manual del Curso LabVIEW Basic I, National Instrument Corporation, Texas, 1998.

## **Internet.**

- [www.mecanicavirtual.org](http://www.mecanicavirtual.org)
- [www.electronica2000.com](http://www.electronica2000.com)
- [www.tallermecanico.com](http://www.tallermecanico.com)
- [www.natinst.com](http://www.natinst.com)
- [www.Wikipedia.com](http://www.Wikipedia.com)

***ANEXOS***

**ANEXO A. HOJA DE DATOS DE TORQUE Y POTENCIA DEL DATSUN  
160J OBTENIDAS EN BANCO DE PRUEBAS.**

**MAHA LPS 3000 Diagrama de Potencias**

Fecha 23.01.2008  
Hora 10:45

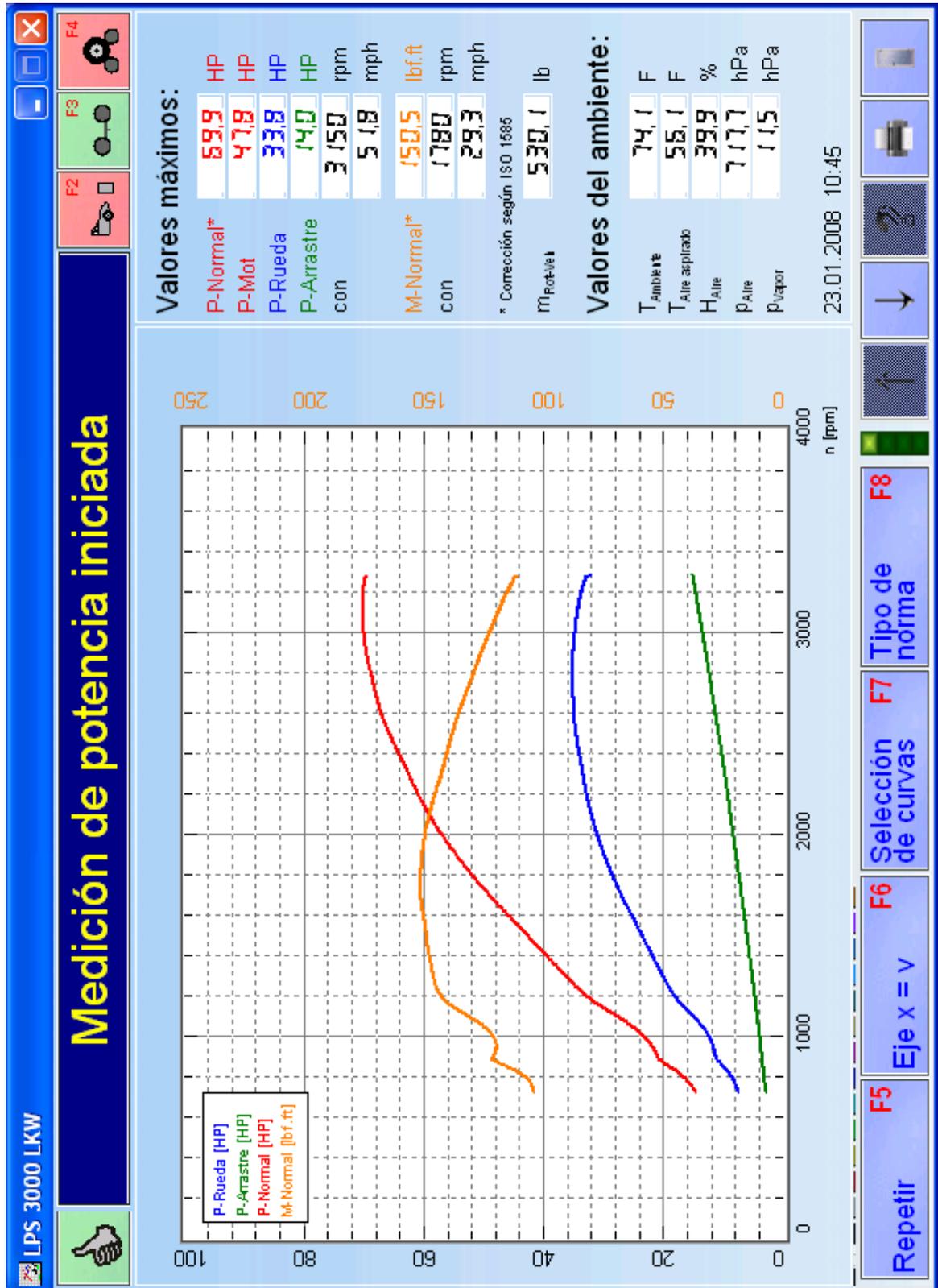
Matrícula IBH0630  
Inspector PRUEBA DE POTENCIA-TORQUE  
Comentarios1 EVALUACION PROYECTO ESPE 23-01-08  
Comentarios2 PRUEBA HECHA EN 3RA MARCHA DE RELACION  
1:1  
Comentarios3  
Comentarios4  
Comentarios5

P-Normal 69,9 HP Corrección según ISO 1585  
P-Mot 47,8 HP  
P-Rueda 33,8 HP  
P-Arrastre 14 HP  
en 3150 rpm  
83,4 km/h  
51,82 mph  
M-Normal 150,5 lbf.ft Corrección según ISO 1585  
en 1780 rpm  
29,3 mph  
47,2 km/h  
T-ambiente 74,1 °F  
23,4 °C  
T-aspiración 56,1 °F  
13,4 °C  
H-aire 39,9 %  
p-aire 717,7 hPa  
p-vapor 11,5 hPa

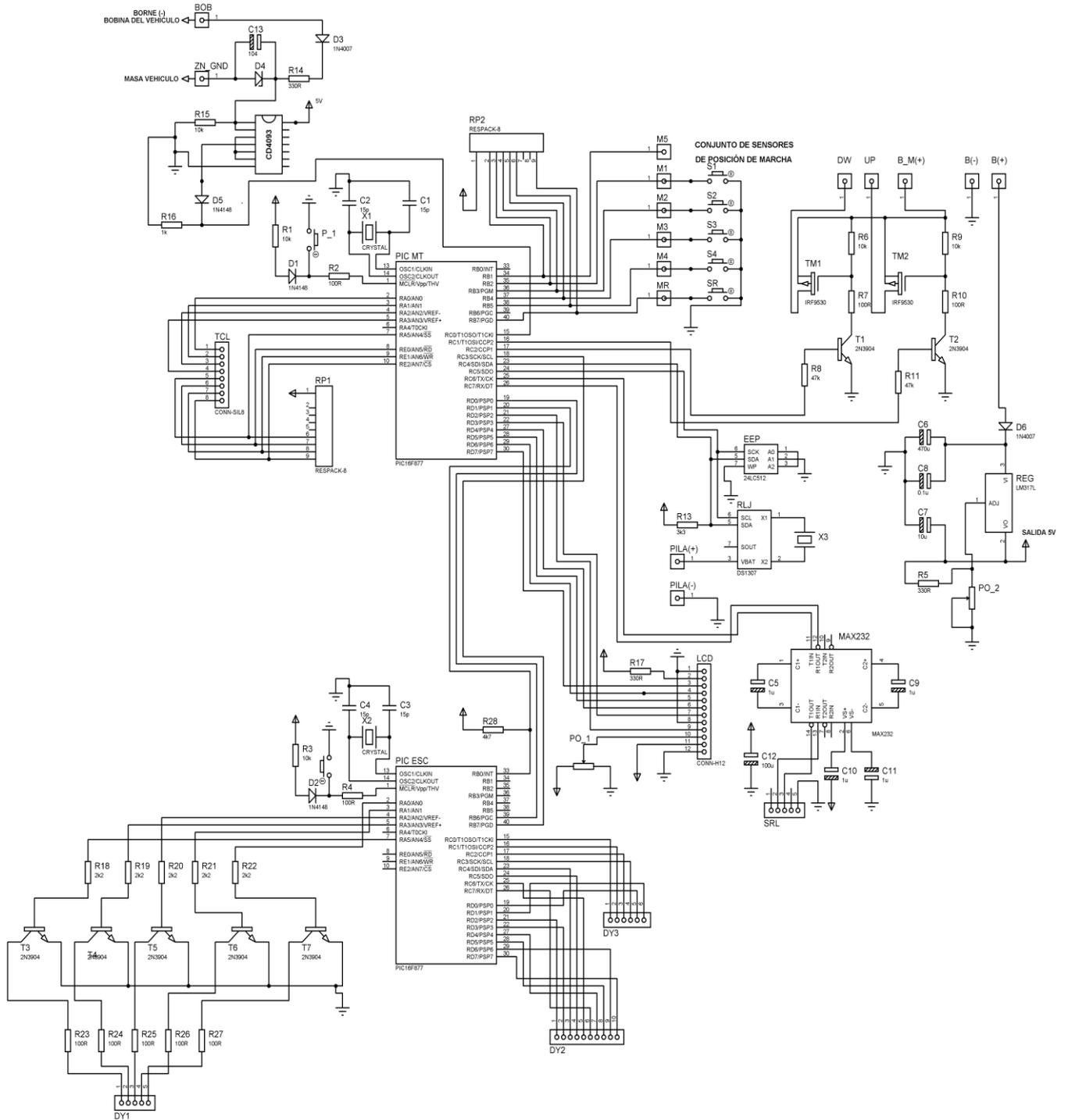
w [rpm]	v [km/h]	v [mph]	P-Normal [HP]	M-Normal [lbf.ft]	P-Rueda [HP]	P-Arrastre [HP]
723,85	19,18	11,92	14,248	103,258	7,161	2,57
749,96	19,88	12,35	14,834	103,722	7,468	2,664
799,15	21,18	13,16	16,248	106,635	8,26	2,838
851,98	22,58	14,03	18,535	114,099	9,634	3,026
901,16	23,88	14,84	20,686	120,41	10,929	3,2
950,35	25,19	15,65	21,521	118,804	11,325	3,374
998,93	26,47	16,45	22,875	120,078	12,064	3,56
1051,76	27,87	17,32	24,923	124,262	13,244	3,779
1100,95	29,18	18,13	27,396	130,505	14,726	3,987
1150,14	30,48	18,94	30,108	137,308	16,372	4,192
1199,33	31,78	19,75	32,415	141,78	17,752	4,388
1247,91	33,07	20,55	34,255	143,941	18,812	4,585

1300,74	34,47	21,42	36,008	145,164	19,778	4,816
1349,93	35,78	22,23	37,593	146,045	20,647	5,029
1399,11	37,08	23,04	39,162	146,808	21,508	5,24
1451,94	38,48	23,91	40,874	147,65	22,439	5,477
1501,13	39,78	24,72	42,383	148,1	23,261	5,687
1550,32	41,09	25,53	43,936	148,666	24,101	5,908
1598,90	42,37	26,33	45,518	149,295	24,959	6,129
1651,73	43,77	27,2	47,274	150,096	25,902	6,385
1700,92	45,08	28,01	48,79	150,442	26,721	6,602
1750,11	46,38	28,82	50,206	150,469	27,474	6,817
1799,29	47,68	29,63	51,595	150,417	28,186	7,054
1847,87	48,97	30,43	52,919	150,182	28,872	7,27
1900,70	50,37	31,3	54,322	149,878	29,576	7,524
1949,89	51,68	32,11	55,499	149,275	30,149	7,755
1999,08	52,98	32,92	56,666	148,675	30,708	7,993
2051,91	54,38	33,79	57,862	147,904	31,27	8,249
2101,10	55,68	34,6	58,817	146,835	31,701	8,47
2150,29	56,99	35,41	59,724	145,697	32,114	8,676
2198,87	58,27	36,21	60,628	144,604	32,5	8,907
2251,70	59,67	37,08	61,477	143,188	32,832	9,155
2300,88	60,98	37,89	62,207	141,801	33,089	9,397
2350,07	62,28	38,7	63,011	140,634	33,398	9,637
2399,26	63,59	39,51	63,84	139,571	33,713	9,888
2447,84	64,87	40,31	64,623	138,453	33,983	10,153
2500,67	66,27	41,18	65,441	137,244	34,248	10,446
2549,86	67,58	41,99	66,144	136,05	34,457	10,717
2599,05	68,88	42,8	66,772	134,75	34,611	10,993
2651,88	70,28	43,67	67,289	133,086	34,666	11,29
2701,06	71,58	44,48	67,753	131,572	34,735	11,539
2750,25	72,89	45,29	68,108	129,902	34,768	11,748
2798,83	74,17	46,09	68,498	128,356	34,764	12,016
2851,66	75,57	46,96	68,934	126,78	34,771	12,307
2900,85	76,88	47,77	69,26	125,226	34,723	12,58
2950,04	78,18	48,58	69,475	123,525	34,595	12,854
2999,23	79,49	49,39	69,692	121,884	34,456	13,132
3047,81	80,77	50,19	69,785	120,083	34,24	13,411
3100,64	82,17	51,06	69,879	118,197	34,001	13,714
3149,83	83,48	51,87	69,938	116,454	33,757	13,999
3199,01	84,78	52,68	69,85	114,525	33,409	14,285
3251,84	86,18	53,55	69,513	112,121	32,867	14,596
3277,96	86,87	53,98	69,312	110,896	32,573	14,753

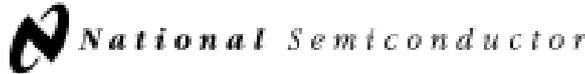
**ANEXO B. GRÁFICAS DE CURVA DE TORQUE Y POTENCIA DEL DATSUN 160J OBTENIDAS EN BANCO DE PRUEBAS.**



# ANEXO C. CIRCUITO COMPLETO DEL MÓDULO DE ANÁLISIS Y ALMACENAMIENTO.



ANEXO D. DATASHEET CD4093 ( Compuerta NAND Schmitt -Trigger).



February 1993

## CD4093BM/CD4093BC Quad 2-Input NAND Schmitt Trigger

### General Description

The CD4093B consists of four Schmitt-trigger circuits. Each circuit functions as a 2-input NAND gate with Schmitt-trigger action on both inputs. The gate switches at different points for positive and negative-going signals. The difference between the positive ( $V_{T+}$ ) and the negative voltage ( $V_{T-}$ ) is defined as hysteresis voltage ( $V_H$ ).

All outputs have equal source and sink currents and conform to standard B-series output drive (see Static Electrical Characteristics).

- Equal source and sink currents
- No limit on input rise and fall time
- Standard B-series output drive
- Hysteresis voltage (any input)  $T_A = 25^\circ\text{C}$

Typical	$V_{DD} = 5.0\text{V}$	$V_{HI} = 1.5\text{V}$
	$V_{DD} = 10\text{V}$	$V_{HI} = 2.2\text{V}$
	$V_{DD} = 15\text{V}$	$V_{HI} = 2.7\text{V}$
Guaranteed		$V_{HI} = 0.1 V_{DD}$

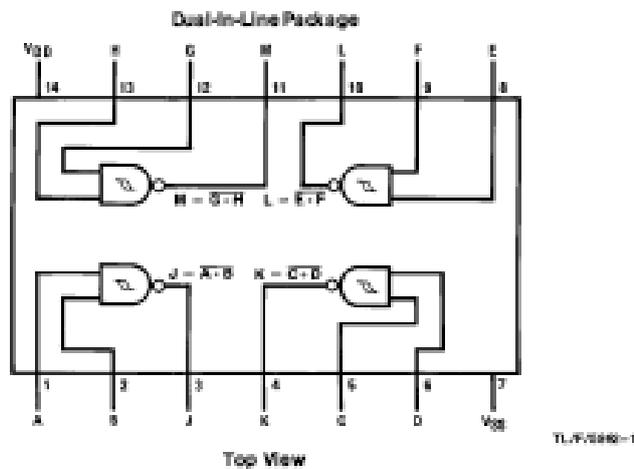
### Features

- Wide supply voltage range 3.0V to 15V
- Schmitt-trigger on each input with no external components
- Noise immunity greater than 50%

### Applications

- Wave and pulse shapers
- High-noise-environment systems
- Monostable multivibrators
- Astable multivibrators
- NAND logic

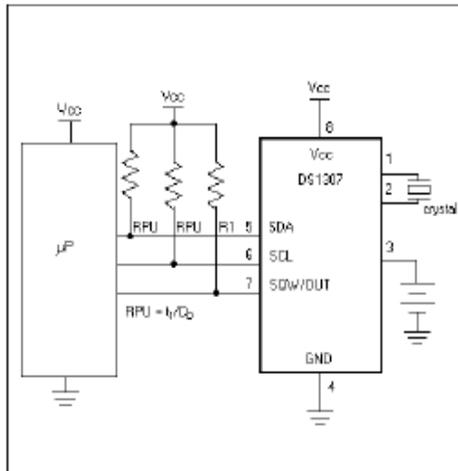
### Connection Diagram



CD4093BM/CD4093BC Quad 2-Input NAND Schmitt Trigger

## ANEXO E. DATASHEET DS1307 (Reloj en tiempo real).

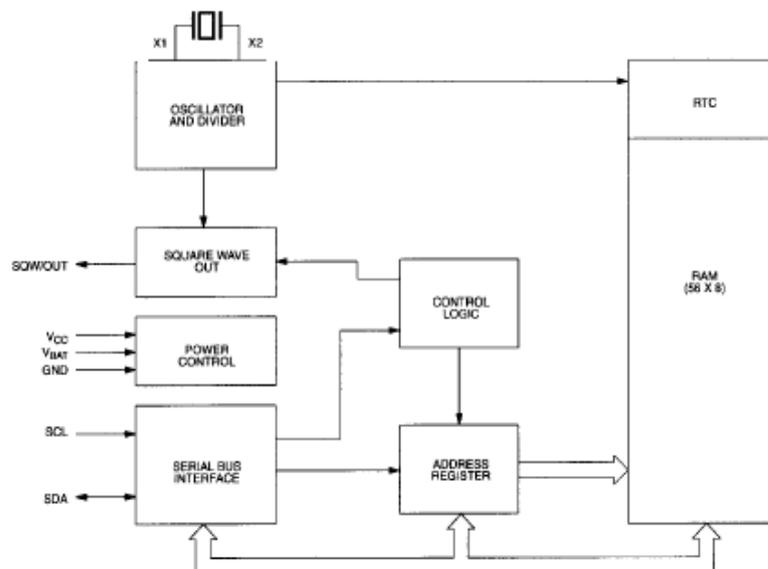
### TYPICAL OPERATING CIRCUIT



### OPERATION

The DS1307 operates as a slave device on the serial bus. Access is obtained by implementing a START condition and providing a device identification code followed by a register address. Subsequent registers can be accessed sequentially until a STOP condition is executed. When  $V_{CC}$  falls below  $1.25 \times V_{BAT}$  the device terminates an access in progress and resets the device address counter. Inputs to the device will not be recognized at this time to prevent erroneous data from being written to the device from an out of tolerance system. When  $V_{CC}$  falls below  $V_{BAT}$  the device switches into a low-current battery backup mode. Upon power-up, the device switches from battery to  $V_{CC}$  when  $V_{CC}$  is greater than  $V_{BAT} + 0.2V$  and recognizes inputs when  $V_{CC}$  is greater than  $1.25 \times V_{BAT}$ . The block diagram in Figure 1 shows the main elements of the serial RTC.

### DS1307 BLOCK DIAGRAM Figure 1



## ANEXO F. DATASHEET 24LC512 (Memoria EEPROM).



# MICROCHIP 24AA512/24LC512/24FC512

## 512K I<sup>2</sup>C™ CMOS Serial EEPROM

### Device Selection Table

Part Number	Vcc Range	Max. Clock Frequency	Temp. Ranges
24AA512	1.8-5.5V	400 kHz <sup>(1)</sup>	I
24LC512	2.5-5.5V	400 kHz	I, E
24FC512	2.5-5.5V	1 MHz	I

**Note 1:** 100 kHz for Vcc < 2.5V.

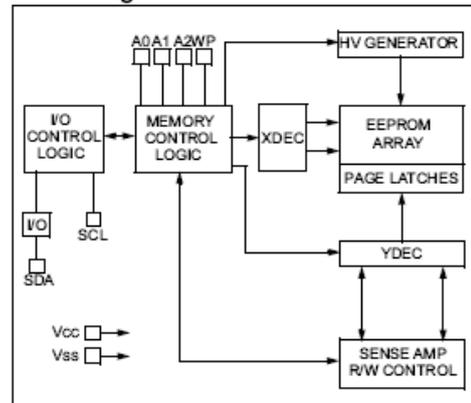
### Features

- Low power CMOS technology
  - Maximum write current 3 mA at 5.5V
  - Maximum read current 400  $\mu$ A at 5.5V
  - Standby current 100 nA typical at 5.5V
- 2-wire serial interface bus, I<sup>2</sup>C compatible
- Cascadable for up to eight devices
- Self-timed ERASE/WRITE cycle
- 128-byte Page-Write mode available
- 5 ms max write-cycle time
- Hardware write-protect for entire array
- Schmitt Trigger inputs for noise suppression
- 1,000,000 erase/write cycles
- Electrostatic discharge protection > 4000V
- Data retention > 200 years
- 8-pin PDIP, SOIC(208 mil), and DFN packages
- 14-lead TSSOP package
- Temperature ranges:
  - Industrial (I): -40°C to +85°C
  - Automotive (E): -40°C to +125°C

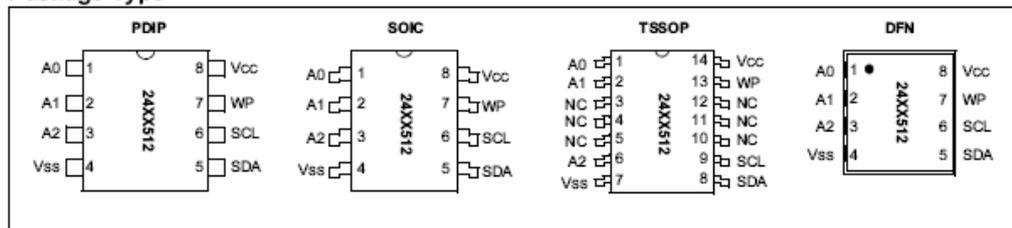
### Description

The Microchip Technology Inc. 24AA512/24LC512/24FC512 (24XX512\*) is a 64K x 8 (512 Kbit) Serial Electrically Erasable PROM, capable of operation across a broad voltage range (1.8V to 5.5V). It has been developed for advanced, low power applications such as personal communications and data acquisition. This device also has a page-write capability of up to 128 bytes of data. This device is capable of both random and sequential reads up to the 512K boundary. Functional address lines allow up to eight devices on the same bus, for up to 4 Mbit address space. This device is available in the standard 8-pin plastic DIP, SOIC, DFN and 14-lead TSSOP packages.

### Block Diagram



### Package Type



\*24XX512 is used in this document as a generic part number for the 24AA512/24LC512/24FC512 devices.

# 24AA512/24LC512/24FC512

## 2.0 PIN DESCRIPTIONS

The descriptions of the pins are listed in Table 2-1.

TABLE 2-1: PIN FUNCTION TABLE

Name	PDIP	SOIC	14-lead TSSOP	DFN	Function
A0	1	1	1	1	User Configured Chip Select
A1	2	2	2	2	User Configured Chip Select
(NC)	—	—	3, 4, 5	—	Not Connected
A2	3	3	6	3	User Configured Chip Select
Vss	4	4	7	4	Ground
SDA	5	5	8	5	Serial Data
SCL	6	6	9	6	Serial Clock
(NC)	—	—	10, 11, 12	—	Not Connected
WP	7	7	13	7	Write Protect Input
Vcc	8	8	14	8	+1.8V to 5.5V (24AA512) +2.5V to 5.5V (24LC512) +2.5V to 5.5V (24FC512)

### 2.1 A0, A1, A2 Chip Address Inputs

The A0, A1, A2 inputs are used by the 24XX512 for multiple device operations. The logic levels on these inputs are compared with the corresponding bits in the slave address. The chip is selected if the compare is true.

Up to eight devices may be connected to the same bus by using different chip select bit combinations. If these pins are left unconnected, the inputs will be pulled down internally to Vss. If they are tied to Vcc or driven high the internal pull-down circuitry is disabled.

In most applications, the chip address inputs A0, A1, and A2 are hard-wired to logic '0' or logic '1'. For applications in which these pins are controlled by a microcontroller or other programmable logic device, the chip address pins must be driven to logic '0' or logic '1' before normal device operation can proceed.

### 2.2 Serial Data (SDA)

This is a bi-directional pin used to transfer addresses and data into and data out of the device. It is an open-drain terminal, therefore, the SDA bus requires a pull-up resistor to Vcc (typical 10 k $\Omega$  for 100 kHz, 2 k $\Omega$  for 400 kHz and 1 MHz).

For normal data transfer SDA is allowed to change only during SCL low. Changes during SCL high are reserved for indicating the START and STOP conditions.

### 2.3 Serial Clock (SCL)

This input is used to synchronize the data transfer from and to the device.

### 2.4 Write Protect (WP)

This pin can be connected to either Vss, Vcc or left floating. Internal pull-down circuitry on this pin will keep the device in the unprotected state if left floating. If tied to Vss or left floating, normal memory operation is enabled (read/write the entire memory 0000-FFFF).

If tied to Vcc, WRITE operations are inhibited. Read operations are not affected.

## 3.0 FUNCTIONAL DESCRIPTION

The 24XX512 supports a bi-directional 2-wire bus and data transmission protocol. A device that sends data onto the bus is defined as a transmitter, and a device receiving data as a receiver. The bus must be controlled by a master device which generates the serial clock (SCL), controls the bus access, and generates the START and STOP conditions while the 24XX512 works as a slave. Both master and slave can operate as a transmitter or receiver, but the master device determines which mode is activated.

## ANEXO G. CÓDIGO DE PROGRAMACIÓN REALIZADO EN PIC BASIC.

```
*****
'* Name : UNTITLED.BAS *
'* Author : ROSERO FREDY—ZAMBRANO SYLVIA *
'* Notice : Copyright (c) 2007 [select VIEW...EDITOR OPTIONS] *
'* Date : 09/10/2007 *
*****

@ DEVICE WDT_OFF, LVP_OFF, BOD_OFF
PORTC.2=0;apagar shift light
PORTC.1=0;apagar shift light
PAUSE 2000; PARA QUE FUNCIONE LCD
DEFINE OSC 12
include "modedefs.bas"; libreria de comunicacion serial
adcon1=7; apagar comparadores de voltaje de puerto a
trisA=%10000;salida 1,2,3,4 teclado,entrada 1 teclado
trisB=%1111110; entrada para marchas
trisC=%001;0=chispazo,1=light_up,2=light_down,3=serial esclavc ,4=I2C_SDA,5=I2C_SCL,6=Trans
Serial,7=Recepc Serial
trisE=%111;entrada 2,3,4 teclado
trisD=%00;salida int esclavo
portb.0=1
DEFINE LCD_DREG PORTD; DEFINE PINES DEL LCD AL PUERTO d
DEFINE LCD_DBIT 4; empezando desde A0 hasta A3
DEFINE LCD_RSREG PORTD; DEFINE PIN PARA CONECTAR RS DEL LCD
DEFINE LCD_RSBIT 2; en patita A1
DEFINE LCD_EREG PORTD; DEFINE PIN PARA CONECTAR E DEL LCD
DEFINE LCD_EBIT 3; en patita A0
DEFINE LCD_BITS 4
dEFINE LCD_LINES 2
'variables para teclado
f4 var portA.0;filas
f3 var portA.1
f2 var portA.2
f1 var portA.3
c4 var portA.5;columnas
c3 var portE.0
c2 var portE.1
c1 var portE.2
tecla var byte;cual tecla aplasto

'variables para shift lite
act_rpm var word
rpm_1up var byte:rpm_2dw var byte:rpm_2up var byte
rpm_3dw var byte:rpm_3up var byte:rpm_4dw var byte
rpm_4up var byte:rpm_5dw var byte:rpm_5up var byte
rpm_Rup var byte
p_copy var byte
ind_grab var word
ini_linea var byte; dibujo digital tacometro LCD
fin_linea var byte; dibujo digital tacometro LCD

'variable para conocer el maximo de rpm que se ha recorrido
max_rpm var byte
fre var byte; BÓRRAR
'variables de numero cilndros y cmbios de marcha
```

num\_marchas var byte; aqui almaceno numero de marchas  
camb\_num\_marchas var byte; para cambiar de numero marchas  
num\_cilindro var byte ;aqui almaceno numero de cilindros  
camb\_num\_cilin var byte; para cambiar numero de cilindros  
coef\_cilindro var byte;sirve para multiplicar en las rpm

calibrar var byte; aqui almaceno calibrar tacometro  
camb\_calib var byte;para calibrar tacometro

dat var word  
dat1 var word  
dato\_enviado var word; uso para sumar +10 en transf datos  
'variable para reloj  
define i2c\_sclout 1;para no necesitar pull\_up en linea de clock  
CPIN var portc.5 ;serial sincronica reloj  
DPIN var portc.4 ; serial sincronica datos  
segu var byte  
minu var byte  
hora var byte  
diaS var byte  
diaF var byte  
mes var byte  
ano var byte  
segu\_ig var byte  
minu\_ig var byte; variables para igualar hora  
hora\_ig var byte  
diaS\_ig var byte  
diaF\_ig var byte  
mes\_ig var byte  
ano\_ig var byte  
valid\_visiesto var byte  
signo var byte; eliminar signos no deseados  
dato var byte  
actualizado var byte  
punt var byte; para salir de lazo cerrado del teclado  
'variables para tacometro  
rpm var word  
rpm\_copy var word;BORRAR  
rebote var byte; para que no se prenda shift lite enseguida de cambio de marcha  
code var word; numero d registros

RPM\_CEN\_MIL VAR BYTE:RPM\_UNID\_DEC VAR BYTE  
POSICION\_MARCHA VAR BYTE:LCD\_MARCHA VAR BYTE  
'variable de tach\_rec  
cuenta\_regresiva var byte;  
tach\_state var byte ; indicador de si grabacion o no grabacion  
'punteros de grabacion memoria  
pos\_uni\_dec var word:pos\_cen\_mil var word:pos\_cambio var word

llenado\_memory var word; saber cuanto usado de la memoria  
pos\_net var word ; posicion memoria dato a enviar  
fin\_trans var word  
dato\_enviar var byte ; contenido dato enviar PC  
repe var word  
'para igualar reloj la primera vez  
eeprom 10,[0]; validar que hora esta igualada  
read 10,actualizado  
if actualizado=0 then  
gosub grabar\_rtc

```

endif

val_predef:: ojo algun rato para iniciar hay que borrar
EEPROM 0,[30,20,31,21,32,22,33,23,34,35]; valores de iniciales de cambio de marcha
eeprom 11,[4,4,0,38]; 11 para numero de cilindros, 12 para numero de marchas, 13 maxima velocidad, 14
calib
;14 calibrar tacometro
eeprom 20,[0]
pause 20
'estas son las direcciones de grabacion de memoria externa eeprom
'pos_seg=0:pos_min=1 :pos_hor=2 :pos_diaS=3:pos_diaF=4 :pos_mes=5 :pos_ano=6
'pos_1up=7: pos_2dw=8:pos_2up=9:pos_3dw=10:pos_3up=11:pos_4dw=12: pos_4up=13
'pos_5dw=14:pos_5up=15:pos_Rmax=16:pos_maxvel=17:pos_Ncil=18:pos_Nmar=19

encerar:: cargar valores de memoria rom
read 0,rpm_1up:read 1,rpm_2dw:read 2,rpm_2up:read 3,rpm_3dw:read 4,rpm_3up
read 5,rpm_4dw :read 6,rpm_4up:read 7,rpm_5dw:read 8,rpm_5up:read 9,rpm_Rup
read 11,num_cilindro:read 12,num_marchas ; cargar valores predefinidos en memoria rom
read 13,max_rpm
read 14,calibrar
read 20,fin_trans
llenado_memory=0; variable uso para conocer cuanto use mi memoria de grabacion
punt=0

'encerar tacometro, rpm ya uso
RPM_CEN_MIL=0:RPM_UNID_DEC =0:POSICION_MARCHA =0:RPM=0000:LCD_MARCHA=0;
rpm****
rpm_copy=0000 ;BORRAR
lcdout,$0c; ojo capaz que toca quitar apagar el cursor
tecla=0
pantalla_inicio:
LCDOUT $FE,1,"digital TACH"
pause 2000
LCDOUT $FE,1,"designed by:"
LCDOUT $FE,$C0,"rOsErO_zAmbRaN0"
PORTC.1=1
PORTC.2=1
pause 2000
PORTC.1=0
PORTC.2=0
pant_1:
LCDOUT $FE,1,"1.tach mode"
LCDOUT $FE,$C0,"2.shift light"
gosub barrido_tecla
gosub antireb_tecla
if tecla=1 then pant_11
if tecla=2 then pant_121
if tecla=11 then pant_2
pause 100
goto pant_1
pant_2:
LCDOUT $FE,1,"3.rpm value"
LCDOUT $FE,$C0,"4.date-clock"
GOSUB barrido_tecla
GOSUB antireb_tecla
if tecla=3 then pant_21
if tecla=4 then pant_22
if tecla=11 then pant_3
if tecla=10 then pant_1

```

```

pause 100
goto pant_2
pant_3:
LCDOUT $FE,1,"5.settings"
LCDOUT $FE,$C0,"6.download PC"
GOSUB barrido_tecla
GOSUB antireb_tecla
if tecla=5 then pant_311
if tecla=6 then pant_32
if tecla=10 then pant_2
if tecla=11 then pant_1
pause 200
goto pant_3
pant_11:
'validar numero cilindro
if num_cilindro=3 then coef_cilindro=80
if num_cilindro=4 then coef_cilindro=60
if num_cilindro=6 then coef_cilindro=40
if num_cilindro=8 then coef_cilindro=30
if num_cilindro=1 then coef_cilindro=120

```

```

LCDOUT $FE,1,"1.only tach"
LCDOUT $FE,$C0,"2.tach rec"
GOSUB barrido_tecla
GOSUB antireb_tecla
if tecla=1 then
max_rpm=0
goto only_tach
endif
if tecla=2 then
max_rpm=0
goto tach_rec
endif
if tecla=14 then pant_1
pause 100
goto pant_11:

```

```

pant_121:: pantalla para modificar 1 up
act_rpm=rpm_1up*100
LCDOUT $FE,1,"1ra UP"
valid_keys_1up:
gosub modifi_marchas
if tecla=15 then
rpm_1up=p_copy
write 0,p_copy
LCDOUT $FE,1,"1up rpm "
LCDOUT $FE,$C0,"was modified"
pause 2000
goto pant_121;actualizo nuevos limites
endif
if tecla=11 then pant_122 ; bajar nivel
if tecla=14 then pant_1 ; salir a menu principal
'pause 100
goto valid_keys_1up

```

```

pant_122 :
act_rpm=rpm_2dw*100
LCDOUT $FE,1,"2da down:"
valid_keys_2down:

```

```
gosub modifi_marchas
if tecla=15 then
rpm_2dw=p_copy
write 1,p_copy
LCDOUT $FE,1,"2down rpm "
ICDOUT $FE,$C0,"was modified"
pause 2000
goto pant_122;actualizo nuevos limites y bajo nivel
endif
if tecla=10 then pant_121; subir nivel
if tecla=11 then pant_123 ; bajar nivel
if tecla=14 then pant_1 ; salir a menu principal
pause 100
goto valid_keys_2down
```

```
pant_123:
act_rpm=rpm_2up*100
LCDOUT $FE,1,"2da up:"
valid_keys_2up:
gosub modifi_marchas
if tecla=15 then
rpm_2up=p_copy
write 2,p_copy
LCDOUT $FE,1,"2up rpm "
ICDOUT $FE,$C0,"was modified"
pause 2000
goto pant_123;actualizo nuevos limites y bjo nivel
endif
if tecla=10 then pant_122; subir nivel
if tecla=11 then pant_124 ; bajar nivel
if tecla=14 then pant_1 ; salir a menu principal
pause 100
goto valid_keys_2up
```

```
pant_124:
act_rpm=rpm_3dw*100
LCDOUT $FE,1,"3ra down:"
valid_keys_3dw:
gosub modifi_marchas
if tecla=15 then
rpm_3dw=p_copy
write 3,p_copy
LCDOUT $FE,1,"3down rpm "
ICDOUT $FE,$C0,"was modified"
pause 2000
goto pant_124
endif
if tecla=10 then pant_123; subir nivel
if tecla=11 then pant_125 ; bajar nivel
if tecla=14 then pant_1 ; salir a menu principal
pause 100
goto valid_keys_3dw
```

```
pant_125:
act_rpm=rpm_3up*100
LCDOUT $FE,1,"3ra up:"
valid_keys_3up:
gosub modifi_marchas
if tecla=15 then
```

```
rpm_3up=p_copy
write 4,p_copy
LCDOUT $FE,1,"3up rpm "
LCDOUT $FE,$C0,"was modified"
pause 2000
goto pant_125;actualizo nuevos limites y bjo nivel
endif
if tecla=10 then pant_124; subir nivel
if tecla=11 then pant_126 ; bajar nivel
if tecla=14 then pant_1 ; salir a menu principal
pause 100
goto valid_keys_3up
```

```
pant_126:
act_rpm=rpm_4dw*100
LCDOUT $FE,1,"4ta down:"
valid_keys_4dw:
gosub modifi_marchas
if tecla=15 then
rpm_4dw=p_copy
write 5,p_copy
LCDOUT $FE,1,"4down rpm "
LCDOUT $FE,$C0,"was modified"
pause 2000
goto pant_126;actualizo nuevos limites y bjo nivel
endif
if tecla=10 then pant_125; subir nivel
if tecla=11 then pant_127 ; bajar nivel
if tecla=14 then pant_1 ; salir a menu principal
pause 100
goto valid_keys_4dw
```

```
pant_127:
act_rpm=rpm_4up*100
LCDOUT $FE,1,"4ta up:"
valid_keys_4up:
gosub modifi_marchas
if tecla=15 then
rpm_4up=p_copy
write 6,p_copy
LCDOUT $FE,1,"4up rpm "
LCDOUT $FE,$C0,"was modified"
pause 2000
goto pant_127;actualizo nuevos limites y bjo nivel
endif
if tecla=10 then pant_126; subir nivel
if tecla=11 then
if num_marchas=4 then pant_128 ; bajar nivel a retro
if num_marchas=5 then pant_129 ; bajar nivel 5dw
endif
if tecla=14 then pant_1 ; salir a menu principal
pause 100
goto valid_keys_4up
```

```
pant_128:
act_rpm=rpm_Rup*100
LCDOUT $FE,1,"Retro max:"
valid_keys_Rtro:
gosub modifi_marchas
```

```

if tecla=15 then
rpm_Rup=p_copy
write 9,p_copy
LCDOUT $FE,1,"Rmax rpm"
LCDOUT $FE,$C0,"was modified"
pause 2000
goto pant_128 ;actualizo nuevos limites y bjo nivel
endif
if tecla=10 then ; subir nivel
if num_marchas=4 then pant_127 ; subir nivel a 4up
if num_marchas=5 then pant_130 ; subir nivel a 5up
endif
if tecla=11 then pant_121 ; bajar nivel
if tecla=14 then pant_1 ; salir a menu principal
pause 100
goto valid_keys_Rtro:

```

```

pant_129:
act_rpm=rpm_5dw*100
LCDOUT $FE,1,"5ta down:"
valid_keys_5dw:
gosub modifi_marchas
if tecla=15 then
rpm_5dw=p_copy
write 7,p_copy
LCDOUT $FE,1,"5down rpm "
LCDOUT $FE,$C0,"was modified"
pause 2000
goto pant_129;actualizo nuevos limites y bjo nivel
endif
if tecla=10 then pant_127; subir nivel
if tecla=11 then pant_130 ; bajar nivel
if tecla=14 then pant_1 ; salir a menu principal
pause 100
goto valid_keys_5dw

```

```

pant_130:
act_rpm=rpm_5up*100
LCDOUT $FE,1,"5ta up:"
valid_keys_5up:
gosub modifi_marchas
if tecla=15 then
rpm_5up=p_copy
write 8,p_copy
LCDOUT $FE,1,"5up rpm "
LCDOUT $FE,$C0,"was modified"
pause 2000
goto pant_130;actualizo nuevos limites y bjo nivel
endif
if tecla=10 then pant_129; subir nivel
if tecla=11 then pant_128 ; bajar nivel
if tecla=14 then pant_1 ; salir a menu principal
pause 100
goto valid_keys_5up

```

```

pant_21:
LCDOUT $FE,1,"Maxima rpm:"
LCDOUT $FE,$C5,DEC max_rpm,"00 RPM"
GOSUB barrido_tecla

```

```
GOSUB antireb_tecla
if tecla=14 then pant_2 ; salir a menu principal
pause 100
goto pant_21:
```

```
pant_22:
punt=1
gosub reloj_fecha
GOSUB barrido_tecla
GOSUB antireb_tecla
punt=0
if tecla=15 then igualar_hora_fecha
if tecla=14 then pant_2 ; salir a menu principal
pause 100
goto pant_22
```

```
pant_311:
camb_num_cilin=num_cilindro
LCDOUT $FE,1,"No cylinders:"
valid_keys_cilindro:
LCDOUT $FE,$C8,dec camb_num_cilin
GOSUB barrido_tecla
GOSUB antireb_tecla
if tecla=1 then camb_num_cilin=1
if tecla=3 then camb_num_cilin=3
if tecla=4 then camb_num_cilin=4
if tecla=6 then camb_num_cilin=6
if tecla=8 then camb_num_cilin=8
if tecla=14 then pant_3
if tecla=11 then pant_312
if tecla=15 then
num_cilindro=camb_num_cilin
write 11,num_cilindro
LCDOUT $FE,1,"You changed to:",dec camb_num_cilin
LCDOUT $FE,$C0," cylinders"
pause 1500
goto pant_311
endif
pause 100
goto valid_keys_cilindro:
```

```
pant_312:
camb_num_marchas=num_marchas
LCDOUT $FE,1,"No Gears:"
valid_keys_marcha:
LCDOUT $FE,$C8,DEC camb_num_marchas
GOSUB barrido_tecla
GOSUB antireb_tecla
if tecla=4 then camb_num_marchas=4
if tecla=5 then camb_num_marchas=5
if tecla=14 then pant_3
if tecla=11 or tecla=10 then pant_311
if tecla=15 then
num_marchas=camb_num_marchas
write 12,num_marchas
LCDOUT $FE,1,"you changed to"
LCDOUT $FE,$C4,dec camb_num_marchas," gears "
pause 1500
goto pant_312
```

```

endif
pause 300
goto valid_keys_marcha

pant_32:
punt=1
LCDOUT $FE,1,"conecting"
LCDOUT $FE,$C0,"CODE: ",DEC fin_trans
serin portc.7,t9600,dat
if dat="C" then
LCDOUT $FE,1,"<<net ready>>"
pause 2000
dat1=0
goto rec_dat
endif
GOSUB barrido_tecla
GOSUB antireb_tecla
punt=0
IF TECLA=14 THEN PANT_3
goto pant_32

rec_dat:
punt=1
I2CREAD DPIN,CPIN,%10100000,dat1,[dato_enviar];leo memoria
pause 10
dato_enviado=dato_enviar+10
LCDOUT $FE,$C0,hex2 dato_enviar, " ",dec dat1, " ",dec fin_trans
serout portC.6, T9600,[#dato_enviado]
GOSUB barrido_tecla
GOSUB antireb_tecla
punt=0
if dat1=fin_trans then
LCDOUT $FE,1,"finish transfer"
pause 1000
goto pant_3
endif
if TECLA=14 THEN
LCDOUT $FE,1,"exit transfer"
pause 1000
goto PANT_3
endif
dat1=dat1+1
goto rec_dat

'para realizar la función de solo tacómetro
only_tach:
tach_state=50
only_tach_rec:
punt=1
rebote=rebote+1
count portc.0,500,rpm
rpm=rpm*coef_cilindro
rpm_copy=rpm
RPM_UNID_DEC=rpm//100
if rpm_unid_dec>=50 then
rpm=rpm+100
endif
RPM_CEN_MIL=rpm/100
rpm=RPM_CEN_MIL*100

```

```

fin_linea=rpm/300
max_rpm=max_rpm max RPM_CEN_MIL; para guardar la maxima revolucion
gosub UBIC_MARCHA
LCDOUT $FE,1
LCDOUT $FE,$C0,"GEAR:",LCD_MARCHA," RPM:",dec rpm_copy
LCDOUT $FE,$80
for ini_linea=1 to fin_linea;sirve linea de tacometro
LCDOUT,255
NEXT
if rebote>2 then
gosub APUNTADOR_MARCHA
endif
serout portd.0, N2400,[RPM_CEN_MIL]; enviar datos a pic esclavo
serout portc.3, N2400,[POSICION_MARCHA]
if tach_state=100 then
gosub grabar_mem_eeprom
endif
GOSUB barrido_tecla
GOSUB antireb_tecla
punt=0
if llenado_memory=31950 then
LCDOUT $FE,1,"finish REC"
LCDOUT $FE,$C0,"memory full"
pause 5000
goto fin_tach
endif

if tecla=14 then
fin_tach:
PORTC.1=0:PORTC.2=0
write 13,max_rpm
if tach_state=100 then
ind_grab=17
write 20,fin_trans ;localidad de numeros usados en la ultima grabacion
i2Cwrite DPIN,CPIN,%10100000,ind_grab,[max_rpm]
pause 10
llenado_memory=llenado_memory/320
LCDOUT $FE,1,"USED MEMORY"
LCDOUT $FE,$C0,"in ",dec llenado_memory," %"
pause 2000
endif
goto pant_11
endif

goto only_tach_rec

tach_rec:
tach_state=100
cuenta_regresiva=9
LCDOUT $FE,$1,"READY"
key_tach_rec:
punt=1
LCDOUT $FE,$c5,dec cuenta_regresiva
cuenta_regresiva=cuenta_regresiva-1
pause 990
if cuenta_regresiva=0 then
LCDOUT $FE,$1," GO...!!!!!"
llenado_memory=0

```

```

pos_cen_mil=20:pos_cambio=21
graba_fecha:
; sirve para leer memoria eeprom es de tipo word
ind_grab=0
I2CREAD DPIN,CPIN,% 11010000,0,[SEGU];leo el reloj
i2Cwrite DPIN,CPIN,% 10100000,ind_grab,[segu];escribo en la memoria
pause 10:ind_grab=ind_grab+1
I2CREAD DPIN,CPIN,% 11010000,1,[minu]
i2Cwrite DPIN,CPIN,% 10100000,ind_grab,[minu]
pause 10:ind_grab=ind_grab+1
I2CREAD DPIN,CPIN,% 11010000,2,[hora]
i2Cwrite DPIN,CPIN,% 10100000,ind_grab,[hora]
pause 10:ind_grab=ind_grab+1
I2CREAD DPIN,CPIN,% 11010000,3,[diaS]
i2Cwrite DPIN,CPIN,% 10100000,ind_grab,[diaS]
pause 10:ind_grab=ind_grab+1
I2CREAD DPIN,CPIN,% 11010000,4,[diaF]
i2Cwrite DPIN,CPIN,% 10100000,ind_grab,[diaF]
pause 10:ind_grab=ind_grab+1
I2CREAD DPIN,CPIN,% 11010000,5,[mes]
i2Cwrite DPIN,CPIN,% 10100000,ind_grab,[mes]
pause 10:ind_grab=ind_grab+1
I2CREAD DPIN,CPIN,% 11010000,6,[ano]
i2Cwrite DPIN,CPIN,% 10100000,ind_grab,[ano]
pause 10:ind_grab=ind_grab+1
i2Cwrite DPIN,CPIN,% 10100000,ind_grab,[rpm_1up]
pause 10:ind_grab=ind_grab+1
i2Cwrite DPIN,CPIN,% 10100000,ind_grab,[rpm_2dw]
pause 10:ind_grab=ind_grab+1
i2Cwrite DPIN,CPIN,% 10100000,ind_grab,[rpm_2up]
pause 10:ind_grab=ind_grab+1
i2Cwrite DPIN,CPIN,% 10100000,ind_grab,[rpm_3dw]
pause 10:ind_grab=ind_grab+1
i2Cwrite DPIN,CPIN,% 10100000,ind_grab,[rpm_3up]
pause 10:ind_grab=ind_grab+1
i2Cwrite DPIN,CPIN,% 10100000,ind_grab,[rpm_4dw]
pause 10:ind_grab=ind_grab+1
i2Cwrite DPIN,CPIN,% 10100000,ind_grab,[rpm_4up]
pause 10:ind_grab=ind_grab+1
i2Cwrite DPIN,CPIN,% 10100000,ind_grab,[rpm_5dw]
pause 10:ind_grab=ind_grab+1
i2Cwrite DPIN,CPIN,% 10100000,ind_grab,[rpm_5up]
pause 10:ind_grab=ind_grab+1
i2Cwrite DPIN,CPIN,% 10100000,ind_grab,[rpm_Rup]
pause 10:ind_grab=ind_grab+2
i2Cwrite DPIN,CPIN,% 10100000,ind_grab,[num_cilindro]
pause 10:ind_grab=ind_grab+1
i2Cwrite DPIN,CPIN,% 10100000,ind_grab,[num_marchas]
pause 10

goto only_tach_rec
endif
GOSUB barrido_tecla
GOSUB antireb_tecla
punt=0
if tecla=14 then pant_11
pause 100
goto key_tach_rec
grabar_mem_eeprom:

```

```

i2Cwrite DPIN,CPIN,%10100000,pos_cen_mil,[RPM_CEN_MIL]
pause 10
i2Cwrite DPIN,CPIN,%10100000,pos_cambio,[POSICION_MARCHA]
pause 10
pos_cen_mil=pos_cen_mil+2;pos_cambio=pos_cambio+2; para aumentar posicion
llenado_memory=llenado_memory+1
fin_trans=pos_cambio
return

```

APUNTADOR\_MARCHA::para activar o desactivar la shift lite

```

PORTC.1=0
PORTC.2=0
SELECT CASE POSICION_MARCHA
CASE 1
  IF RPM_CEN_MIL>=rpm_1up THEN
    PORTC.1=1
  ENDIF

  return
CASE 2
  IF RPM_CEN_MIL<=RPM_2dw THEN
    PORTC.2=1
  ENDIF
  IF RPM_CEN_MIL>=RPM_2up THEN
    PORTC.1=1
  ENDIF
  return
CASE 3
  IF RPM_CEN_MIL<=RPM_3dw THEN
    PORTC.2=1
  ENDIF
  IF RPM_CEN_MIL>=RPM_3up THEN
    PORTC.1=1
  ENDIF
  return
CASE 4
  IF RPM_CEN_MIL<=RPM_4dw THEN
    PORTC.2=1
  ENDIF
  IF RPM_CEN_MIL>=RPM_4up THEN
    PORTC.1=1
  ENDIF
  return
CASE 5
  IF RPM_CEN_MIL<=RPM_5dw THEN
    PORTC.2=1
  ENDIF
  IF RPM_CEN_MIL>=rpm_5up THEN
    PORTC.1=1
  ENDIF
  return
CASE 27
  IF RPM_CEN_MIL>=RPM_RUP THEN
    PORTC.1=1
  ENDIF
  return

```

case 23 ;neutro

```
return  
END SELECT
```

```
modifi_marchas:  
LCDOUT $FE,$C5,DEC act_rpm," RPM"  
GOSUB barrido_tecla  
GOSUB antireb_tecla  
if tecla=12 then act_rpm=act_RPM+100 ; subir rpm limite  
if tecla=13 then act_rpm=act_rpm-100; bajar rpm limite  
p_copy=act_rpm/100  
return
```

```
'para leer marcha  
UBIC_MARCHA:: para saber en que marcha esta el vehiculo  
IF PORTB.7=0 THEN  
POSICION_MARCHA=27;retro  
LCD_MARCHA=82  
RETURN  
ENDIF
```

```
'IF PORTB.1=1 THEN  
'POSICION_MARCHA=05  
'LCD_MARCHA=53  
'RETURN  
'ENDIF
```

```
IF PORTB.5=1 THEN  
POSICION_MARCHA=04  
LCD_MARCHA=52  
RETURN  
ENDIF
```

```
IF PORTB.4=1 THEN  
POSICION_MARCHA=03  
LCD_MARCHA=51  
RETURN  
ENDIF
```

```
IF PORTB.3=1 THEN  
POSICION_MARCHA=02  
LCD_MARCHA=50  
RETURN  
ENDIF
```

```
IF PORTB.2=1 THEN  
POSICION_MARCHA=01  
LCD_MARCHA=49  
RETURN;  
ENDIF
```

```
POSICION_MARCHA=23 ; NUM_MARCHA=N CUANDO ES NEUTRO  
LCD_MARCHA=78  
PORTC.1=0  
PORTC.2=0  
rebote=0  
RETURN
```

```
;para leer los datos del reloj  
reloj_fecha:
```

PRINCIPIO:

```
I2CREAD DPIN,CPIN,% 11010000,0,[SEGU]
I2CREAD DPIN,CPIN,% 11010000,1,[minu]
I2CREAD DPIN,CPIN,% 11010000,2,[hora]
I2CREAD DPIN,CPIN,% 11010000,3,[diaS]
I2CREAD DPIN,CPIN,% 11010000,4,[diaF]
I2CREAD DPIN,CPIN,% 11010000,5,[mes]
I2CREAD DPIN,CPIN,% 11010000,6,[ano]
```

```
LCDOUT $fe,1,hex2 hora,":",hex2 minu,":",hex2 segu
```

```
lcdout $fe,$c0
if diaS=$1 then lcdout "Dom."
if diaS=$2 then lcdout "Lun."
if diaS=$3 then lcdout "Mar."
if diaS=$4 then lcdout "Mie."
if diaS=$5 then lcdout "Jue."
if diaS=$6 then lcdout "Vie."
if diaS=$7 then lcdout "Sab."
```

```
lcdout$fe,$c5,HEX2 diaf,"/"
lcdout$fe,$cB,"/20",HEX2 ano
```

```
lcdout$fe,$c8
if mes=$1 then lcdout "ene"
if mes=$2 then lcdout "feb"
if mes=$3 then lcdout "mar"
if mes=$4 then lcdout "abr"
if mes=$5 then lcdout "may"
if mes=$6 then lcdout "jun"
if mes=$7 then lcdout "jul"
if mes=$8 then lcdout "ago"
if mes=$9 then lcdout "sep"
if mes=$10 then lcdout "oct"
if mes=$11 then lcdout "nov"
if mes=$12 then lcdout "dic"
pause 50
return
```

```
GRABAR_RTC::12:00,lunes 29 octubre 2007
```

```
minu_ig=$00:hora_ig=$12 :diaS_ig=$3 :diaF_ig=$29 :mes_ig=$01:ano_ig=$08
```

```
IGUALAR_RTC:
```

```
I2Cwrite DPIN,CPIN,% 11010000,0,[$01]
pause 10
I2Cwrite DPIN,CPIN,% 11010000,1,[minu_ig]
pause 10
I2Cwrite DPIN,CPIN,% 11010000,2,[hora_ig]
pause 10
I2Cwrite DPIN,CPIN,% 11010000,3,[diaS_ig]
pause 10
I2Cwrite DPIN,CPIN,% 11010000,4,[diaF_ig]
pause 10
I2Cwrite DPIN,CPIN,% 11010000,5,[mes_ig]
pause 10
I2Cwrite DPIN,CPIN,% 11010000,6,[ano_ig]
pause 10
write 10,1
return
```

' aqui hacer la habitacion de la memoria para obtener datos de PC

```
;filas salida  
;columnas entrada  
barrido_tecla:  
high f1:high f2:high f3:high f4; encerrar teclados  
low f1; verificar fila 1  
pause 30  
if c1=0 then tecla=1: return  
if c2=0 then tecla=2: return  
if c3=0 then tecla=3: return  
if c4=0 then tecla=10: return  
high f1
```

```
low f2; verificar fila 2  
pause 30  
if c1=0 then tecla=4: return  
if c2=0 then tecla=5: return  
if c3=0 then tecla=6: return  
if c4=0 then tecla=11: return  
high f2
```

```
low f3; verificar fila 3  
pause 30  
if c1=0 then tecla=7: return  
if c2=0 then tecla=8: return  
if c3=0 then tecla=9 : return  
if c4=0 then tecla=12 : return  
high f3
```

```
low f4; verificar fila 4  
pause 30  
if c1=0 then tecla=14 : return  
if c2=0 then tecla=16 : return  
if c3=0 then tecla=15 : return  
if c4=0 then tecla=13 : return  
high f4  
'pause 20  
if punt=1 then return  
goto barrido_tecla
```

```
antireb_tecla:  
if c1=0 then antireb_tecla  
if c2=0 then antireb_tecla  
if c3=0 then antireb_tecla  
if c4=0 then antireb_tecla  
pause 0  
return  
end
```

```
igualar_hora_fecha:  
segu_ig=$00  
minu_ig=minu  
hora_ig=hora  
diaS_ig=diaS  
diaF_ig=diaF  
mes_ig=mes
```

```

ano_ig=ano
key_igualar_hora_fecha:
gosub mostrar_igualando
GOSUB barrido_tecla
GOSUB antireb_tecla
if tecla=1 then; + horas
hora_ig=hora_ig+1
signo=hora_ig
gosub anti_sig
hora_ig=signo
  if hora_ig=$24 then
    hora_ig=$00
  endif
endif
if tecla=2 then;+minutos
minu_ig=minu_ig + 1
signo=minu_ig
gosub anti_sig
minu_ig=signo
  if minu_ig=$60 then
    minu_ig=$00
  endif
endif
if tecla=4 then
diaS_ig=diaS_ig + 1
  if diaS_ig=$8 then
    diaS_ig=$1
  endif
endif
if tecla=5 then
valid_visiesto=50;cualquier otro año no visiesto
diaF_ig=diaF_ig+1
signo=diaF_ig
gosub anti_sig
diaF_ig=signo
  if mes_ig=$4 then hasta_30
  if mes_ig=$6 then hasta_30
  if mes_ig=$9 then hasta_30
  if mes_ig=$11 then hasta_30

  if mes_ig=$2 then
    valid_visiesto=ano_ig//4
    if valid_visiesto=0 then
      if diaF_ig=$30 then
        diaF_ig=$01
      endif
    else
      if diaF_ig=$29 then
        diaF_ig=$01
      endif
    endif

    goto key_igualar_hora_fecha
  endif
endif
if tecla=6 then
mes_ig=mes_ig+1

```

```

signo=mes_ig
gosub anti_sig
mes_ig=signo
  if mes_ig=$13 then
    mes_ig=$1
  endif
endif

if tecla=11 then
ano_ig=ano_ig+1
signo=ano_ig
gosub anti_sig
ano_ig=signo
endif
if tecla=15 then
LCDOUT $FE,1,"date-clock"
LCDOUT $FE,$C0,"changed"
pause 2000
gosub IGUALAR_RTC:
pause 50
goto pant_2
endif
if tecla=14 then pant_2 ; salir a menu principal
pause 100
goto key_igualar_hora_fecha

hasta_30:
if diaF_ig=$31 then
diaF_ig=$01
endif
goto key_igualar_hora_fecha

hasta_31:
if diaF_ig=$32 then
diaF_ig=$01
endif
goto key_igualar_hora_fecha

mostrar_igualando:
LCDOUT $fe,1,hex2 hora_ig,":",hex2 minu_ig,":",hex2 segu_ig
lcdout $fe,$c0
if diaS_ig=$1 then lcdout "Dom."
if diaS_ig=$2 then lcdout "Lun."
if diaS_ig=$3 then lcdout "Mar."
if diaS_ig=$4 then lcdout "Mie."
if diaS_ig=$5 then lcdout "Jue."
if diaS_ig=$6 then lcdout "Vie."
if diaS_ig=$7 then lcdout "Sab."
lcdout$fe,$c5,HEX2 diaf_ig,/"
lcdout$fe,$cB, "/20",HEX2 ano_ig
lcdout$fe,$c8
if mes_ig=$1 then lcdout "ene"
if mes_ig=$2 then lcdout "feb"
if mes_ig=$3 then lcdout "mar"
if mes_ig=$4 then lcdout "abr"
if mes_ig=$5 then lcdout "may"
if mes_ig=$6 then lcdout "jun"
if mes_ig=$7 then lcdout "jul"
if mes_ig=$8 then lcdout "ago"

```

```
if mes_ig=$9 then lcdout "sep"
if mes_ig=$10 then lcdout "oct"
if mes_ig=$11 then lcdout "nov"
if mes_ig=$12 then lcdout "dic"
pause 50
return
anti_sig:
if signo=$0A then signo=$10
if signo=$1A then signo=$20
if signo=$2A then signo=$30
if signo=$3A then signo=$40
if signo=$4A then signo=$50
if signo=$5A then signo=$60
if signo=$6A then signo=$70
if signo=$7A then signo=$80
if signo=$8A then signo=$90
if signo=$9A then signo=$00
return
```

Latacunga, febrero 2008

Realizado por:

---

Sr. Rosero Obando Fredy Alexander

---

Srta. Zambrano Rengel Sylvia Vanessa

---

Ing. Juan Castro  
COORDINADOR DE LA CARRERA DE  
INGENIERÍA AUTOMOTRIZ

---

Dr. Eduardo Vásquez  
SECRETARIO ACADÉMICO