



Universidad del Azuay

Facultad de Ciencia y Tecnología

Escuela de Ingeniería Electrónica

**DISEÑO Y CONSTRUCCIÓN DE UN RUGOSÍMETRO DIGITAL EN
3D CON INTERFACE A LA PC BASADO EN LABVIEW V8.1 CON
APLICACIÓN EN PULIDO DE PIEZAS METALICAS**

**Tesis Previa a la Obtención del Título de
Ingeniero Electrónico**

Autores: HERNÁN SANTIAGO MALDONADO FAJARDO

MAURICIO EFRAÍN VILLOTA VERDUGO

Director: Ing. Hernán Viteri C

**Cuenca – Ecuador
2007**

DEDICATORIA

Con todo el cariño dedico esta tesis en primer lugar a Dios por colmarme de bendiciones, a mis abuelitas Luli y Blanquita, que con sus palabras me alientan a seguir adelante. A Hernán, Anita, Ceci y Malí que han estado siempre a mi lado y me brindan su amor incondicional, todo lo que he logrado ser ha sido gracias a ellos y su esfuerzo. Sin lugar a duda a María Fernanda, por su constante apoyo, cariño y sonrisa.

Tuero

Este trabajo lo dedico a mis padres, quienes me han apoyado incondicionalmente durante toda mi carrera, a mi Papá por su sabiduría y a mi Mamá por su paciencia, ya que sin el respaldo de ellos no hubiera podido culminar una meta más de mi vida; y especialmente a mi pequeño hijo Santiago, que es la inspiración que me impulsa a superarme y seguir adelante cada día, siempre estaré a tu lado.....

Epacho

AGRADECIMIENTO

Nuestro sincero agradecimiento al Ing. Hernán Viteri por su valioso apoyo y guía durante el desarrollo de la presente tesis.

A los señores profesores de la escuela de electrónica por habernos brindado una solvente formación.

Índice de Contenidos

Dedicatoria	ii
Agradecimiento	iii
Índice de Contenidos	iv
Resumen	xii
Abstract	xiii
INTRODUCCION	xiv
 CAPITULO I: RUGOSIDAD	
1.1 Generalidades.....	1
1.2 Rugosidad.....	1
1.3 Importancia de cuantificar la rugosidad.....	2
1.4 Parámetros de la superficie de un pieza maquinada.....	3
1.4.1 Superficie real.....	3
1.4.2 Superficie geométrica.....	3
1.4.3 Superficie de referencia.....	3
1.4.4 Perfil real.....	4
1.4.5 Longitud básica.....	4
1.4.6 Longitud de evaluación (l_n).....	4
1.4.7 Línea media de los mínimos cuadrados	5
1.4.8 Línea media aritmética (o línea central).....	5
1.4.9 Cresta local del perfil.....	6
1.4.10 Valle local del perfil.....	6
1.4.11 Cresta del perfil.....	6
1.4.12 Valle del perfil.....	6
1.5 Parámetros de rugosidad.....	6
1.5.1 Parámetros de rugosidad con respecto a la dirección de las alturas.....	8
1.5.1.1 Altura de una irregularidad del perfil.....	8
1.5.1.2 Altura máxima de una cresta	8
1.5.1.3 Profundidad máxima de un valle (R_m).....	8
1.5.1.4 Altura máxima del perfil (R_y).....	8
1.5.1.5 Altura de las irregularidades en diez puntos (R_z).....	9
1.5.1.6 Altura media de las irregularidades del perfil (R_c).....	9
1.5.1.7 Desviación media aritmética del perfil (R_a).....	10
1.5.2 Parámetros de rugosidad respecto a la dirección transversal.....	10
1.5.2.1 Paso de las irregularidades del perfil.....	10

1.5.2.2 Paso medio de las irregularidades del perfil.....	11
1.5.2.3 Paso de las crestas locales del perfil.....	11
1.5.2.4 Paso medio de las crestas del perfil.....	11
1.5.2.5 Longitud desarrollada del perfil (L_o).....	11
1.5.2.6 Relación de longitud del perfil (lr).....	12
1.5.2.7 Densidad de las crestas del perfil (D).....	12
1.5.3 Parámetros de rugosidad respecto a la forma de las irregularidades.....	12
1.5.3.1 Longitud portante del perfil (μ_p).....	12
1.5.3.2 Curva de la tasa de longitud portante del perfil.....	13
1.6 Especificación de la rugosidad.....	13
1.6.1 Representación.....	13
1.6.2 Simbología.....	15
1.6.3 Clases de rugosidad.....	16
1.6.4 Aplicaciones de las distintas clases de rugosidad.....	16
1.6.5 Proceso de fabricación.....	17
1.7 Métodos para medir la rugosidad.....	18
1.7.1 Rugosímetros.....	18
1.7.1.1 Principio de funcionamiento.....	19
1.7.1.2 Tipos de Rugosímetros.....	20
1.7.1.3 Rugosímetro de palpador mecánico.....	20
1.7.1.4 Rugosímetro de palpador inductivo.....	20
1.7.1.5 Rugosímetro de palpador capacitivo.....	21
1.7.1.6 Rugosímetro de palpador piezoeléctrico.....	22
1.7.1.7 Rugosímetro con patín mecánico.....	22
1.8 Selección del Rugosímetro.....	23

CAPITULO II: DISEÑO DEL HARDWARE

2.1 Diseño del circuito general.....	24
2.1.1 Introducción teórica al programa Protel 99SE.....	24
2.1.2 Conceptos básicos.....	24
2.1.2.1 Combinaciones de teclas útiles.....	28
2.1.2.2 Los cursores de edición.....	28
2.1.2.3 Obtención de ayuda.....	29
2.1.3 Creación de un Printed Circuit Board (PCB) con Protel.....	29
2.1.3.1 Creación del esquemático.....	30
2.1.3.2 Creación de una PCB.....	33
2.1.3.3 Comprobación del esquemático.....	36
2.1.3.4 Colocación de componentes.....	36
2.1.3.5 Rutado.....	37
2.2 Diseño del circuito de mando.....	38
2.2.1 Sensor óptico (OPTOACOPLADOR).....	38
2.2.2 Encoders (Opcional).....	40
2.2.3 Reset general.....	42

2.3 Comunicación serial.....	43
2.3.1 Comunicaciones serie asíncronas.....	43
2.3.1.1 Norma RS232.....	43
2.3.1.2 Conexión del microcontrolador al puerto serie de PC.....	44
2.3.1.3 El conector DB9 del PC.....	44
2.3.1.4 El chip MAX 232.....	45
2.4 Selector manual / normal.....	47
2.4.1 Modo Normal.....	47
2.4.2 Modo Manual.....	48
2.5 Conector para joystick.....	48
2.6 Condensadores by-pass.....	49
2.7 Leds de señalización.....	50
2.8 Conexión del microcontrolador.....	51
2.9 Diseño del circuito de fuerza.....	52
2.9.1 Conector de Datos.....	52
2.9.2 Leds de Señalización.....	52
2.9.3 Fin Carrera.....	53
2.10 Comando de los motores.....	54
2.10.1 Motores de Corriente Continua.....	54
2.10.2 Velocidad.....	54
2.10.3 Sentido de giro y alimentación.....	55
2.10.4 La etapa de potencia.....	56
2.11 Diseño de la fuente de alimentación.....	58
2.11.1 Protección y rectificación.....	58
2.11.2 Estabilizadores de Tensión.....	58

CAPITULO 3: DISEÑO DEL SOFTWARE

3.1 Diseño del firmware para microcontrolador.....	60
3.1.1 Introducción teórica sobre los pic16f877.....	60
3.1.1.1 Introducción a los microcontroladores.....	60
3.1.1.2 Controlador y Microcontrolador.....	60
3.1.1.3 Diferencia entre un Microprocesador y un Microcontrolador.....	61
3.1.1.4 Aplicaciones de Microcontroladores.....	63
3.1.1.5 El mercado de los microcontroladores.....	64
3.1.1.6 ¿Qué microcontrolador emplear.....	65
3.1.1.7 Arquitectura básica.....	67
3.1.1.8 El procesador o UCP.....	68
3.1.1.9 Memoria.....	68
3.1.1.10 Puertas de Entrada y Salida.....	71
3.1.1.11 Reloj principal.....	71
3.2.1 Recursos Especiales.....	72
3.2.1.1 Temporizadores o “timers”.....	72
3.2.1.2 Perro guardián o “watchdog”.....	73

3.2.1.3 Protección ante fallo de alimentación o “brownout”	73
3.2.1.4 Estado de reposo ó de bajo consumo.....	73
3.2.1.5 Conversor a/d (cad).....	73
3.2.1.6 Conversor d/a (cda).....	74
3.2.1.7 Comparador analógico.....	74
3.2.1.8 Modulador de anchura de impulsos o pwm.....	74
3.2.1.9 Puertas de e/s digitales.....	74
3.2.1.10 Puertas de comunicación.....	74
3.3 La familia de los pic como elección.....	75
3.3.1 Características relevantes.....	76
3.3.1.1	76
Arquitectura.....	76
3.3.1.2 Segmentación.....	76
3.3.1.3 Formato de las instrucciones.....	77
3.3.1.4 Juego de instrucciones.....	77
3.3.1.5 Todas las instrucciones son ortogonales.....	77
3.3.1.6 Arquitectura basada en un “banco de registros.....	77
3.3.1.7 Diversidad de modelos de microcontroladores con prestaciones y recursos diferentes.....	77
3.3.1.8 Herramientas de soporte potentes y económicas.....	77
3.4 Introducción teórica al programa picbasic pro v2.44.....	77
3.4.1 Introducción.....	77
3.4.2 Los Micros.....	78
3.4.3 Programación.....	78
3.4.4 Bases del PBP.....	79
3.4.4.1 Identificadores.....	79
3.4.4.2 Etiquetas de línea (LABELS)	80
3.4.4.3 Variables.....	80
3.4.4.4 Alias.....	81
3.4.4.5 Arrays (Arreglos).....	81
3.4.4.6 Constantes.....	82
3.4.4.7 Símbolos.....	83
3.4.4.8 Constantes Numéricas.....	83
3.4.4.9 Cadena de Constantes.....	83
3.4.4.10 Puertos y otros registros.....	83
3.4.4.11 Pines.....	84
3.4.4.12 Comentarios.....	85
3.4.4.13 Declaraciones múltiples.....	85
3.4.4.14 Include.....	85
3.4.4.15 Define.....	85
3.4.4.16 Operadores matemáticos.....	87
3.4.4.17 Operadores de Comparación.....	89
3.4.4.18 Operadores Lógicos.....	89
3.4.4.19 Referencia de declaraciones.....	90
3.4.4.20 ADCIN.....	93

3.4.4.21 ASM ENDASM.....	93
3.4.4.22 CALL.....	94
3.4.4.23 CLEAR.....	94
3.4.4.24 EEPROM.....	94
3.4.4.25 END.....	95
3.4.4.26 FOR.NEXT.....	95
3.4.4.27 GOTO.....	96
3.4.4.28 INPUT.....	96
3.4.4.29 OUTPUT.....	96
3.4.4.30 ON INTERRUPT.....	97
3.4.4.31 PWM.....	98
3.4.4.32 RETURN.....	98
3.4.4.33 SERIN.....	99
3.4.4.34 SEROUT.....	100
3.4.4.35 STOP.....	100
3.4.4.36 SWAP.....	100
3.4.4.37 WHILE...WEND.....	101
3.5 Programación del pic.....	101
3.5.1 Aspectos Generales.....	101
3.5.2 Diagrama de bloques del programa.....	102
3.5.2.1 Configuraciones generales.....	102
3.5.2.1.1 Configuración del oscilador.....	102
3.5.2.1.2 Configuración del USART.....	103
3.5.2.1.3 Configuración del convertor.....	105
3.5.2.1.4 Declaración de variables.....	105
3.5.2.2 Transmisión y recepción de datos.....	106
3.5.2.2.1 Recepción del dato de superficie.....	106
3.5.2.2.2 Transmisión de datos y conversión.....	108
3.5.2.3 Mando de motores.....	109
3.6 Diseño del software de control en la PC.....	111
3.6.1 Introducción teórica sobre el programa Labview v8.1.....	111
3.6.2 ¿Qué es LabVIEW?.....	111
3.6.2.1 Ventajas del programa LabView.....	112
3.6.2.2 Aplicaciones de Labview.....	112
3.6.3 Programación gráfica con Labview.....	112
3.6.3.1 Instrumentos virtuales.....	113
3.6.4 Tipos de variables y datos numéricos.....	115
3.6.4.1 Número de bits de un número.....	115
3.6.4.2 Signo en el número.....	115
3.6.4.3 Números fraccionarios.....	115
3.6.4.4 Datos booleanos y alfanuméricos.....	116
3.6.4.5 Matrices.....	116
3.6.5 Paletas o menús de trabajo.....	117
3.6.5.1 Paleta de controles.....	117

3.6.5.1.1 Numéricos.....	118
3.6.5.1.2 Booleanos.....	118
3.6.5.1.3 String & Path.....	119
3.6.5.1.4 List & Table.....	119
3.6.5.1.5 Array & Clusters.....	119
3.6.5.1.6 Graph.....	120
3.6.5.1.7 Ring & Enum.....	120
3.6.5.1.8 Classic Controls.....	121
3.6.5.1.9 Decorations.....	121
3.6.5.1.10 Error Cluster.....	121
3.6.5.1.11 i / o.....	122
3.6.5.2 Paleta de funciones.....	122
3.6.5.2.1 Structures.....	123
3.6.5.2.2 Numeric.....	123
3.6.5.2.3 Trig & Log.....	124
3.6.5.2.4 Boolean.....	124
3.6.5.2.5 Comparison.....	124
3.6.5.2.6 Conversión.....	125
3.6.5.2.7 String.....	125
3.6.5.2.8 Array & Cluster.....	125
3.6.5.2.9 File I/O.....	126
3.6.5.2.10 Time & Dialog.....	126
3.6.5.2.11 Vi.....	127
3.6.5.2.12 Analyze.....	127
3.6.5.2.13 Ni Measurements.....	127
3.6.5.2.14 Instrument I/O.....	128
3.6.5.2.15 Network.....	128
3.6.6 Estructuras y elementos de programación.....	129
3.6.6.1 Los Ciclos While.....	129
3.6.6.2 Ciclo For Loop.....	130
3.6.6.3 Cuadros de casos ‘CASE’.....	131
3.6.6.4 Las Secuencias.....	131
3.6.7 Variables.....	132
3.6.7.1 Variables Locales.....	132
3.6.7.2 Variables Globales.....	133
3.6.8 Cuadros de fórmula.....	134
3.6.9 Manejo de puerto serial.....	135
3.6.9.1 Inicialización Del Puerto.....	136
3.6.9.2 Escritura Del Puerto.....	136
3.6.9.3 Contador De Bytes.....	137
3.6.9.4 Lectura Del Puerto.....	137
3.6.9.5 Fin De Puerto.....	137
3.7 Programación de la PC.....	138
3.7.1 Diagrama de bloques.....	138

3.7.1.1 Adquisición de datos y protocolo PC – PIC.....	139
3.7.1.1.1 Panel frontal de adquisición de datos.....	139
3.7.1.1.2 Diagrama de programación de adquisición de datos.....	140
3.7.1.2 Visualización de curvas individuales en 2D.....	145
3.7.1.2.1 Panel Frontal De Visualización En 2D.....	145
3.7.1.2.2 Diagrama de programación para visualización de curvas individuales en 2d.....	148
3.7.1.3 Visualización de curvas individuales en 3d.....	150
3.7.1.3.1 Panel Frontal De Visualización En 3d.....	150
3.7.1.3.2 Diagrama De Programación De Visualización De Curvas Individuales....	151
3.7.1.4 Cálculo de parámetros de rugosidad.....	153
3.7.1.4.1 Panel Frontal Del Cálculo De Parámetros De Rugosidad.....	153
3.7.1.4.2 Diagrama De Programación Del Cálculo De Parámetros De Rugosidad...	154
3.7.1.5 Superficie total en 3d.....	155
3.7.1.5.1 Panel Frontal De La Superficie Total En 3d.....	155
3.7.1.5.2 Diagrama De Programación.....	156

CAPITULO 4: ASPECTOS CONSTRUCTIVOS Y VALIDACIÓN DEL EQUIPO

4.1 Aspectos generales del PCB.....	157
4.1.1 Placa impresa circuito de fuerza.....	157
4.1.2 Placa impresa circuito de mando.....	158
4.1.3 Placa impresa fuente de alimentación.....	160
4.2 Construcción de la parte mecánica.....	160
4.2.1 Eje X.-	161
4.2.2 Eje Y.-	164
4.2.3 Eje Z.-	165
4.2.4 Micrómetro.-	166
4.3 Rugosímetro construido.....	167
4.3.1 Caja de Control.....	168
4.3.2 Control remoto.....	169
4.3.3 Punta del Sensor.....	170
4.4 Rugosímetro Patrón.....	170
4.5 Mediciones de una pieza maquinada.....	173
4.6 Pruebas Realizadas.....	173
4.6.1 Pruebas obtenidas por el Rugosímetro Patrón.....	173
4.6.2 Pruebas obtenidas por el Rugosímetro construido.....	175
4.7 Conclusiones de resultados.....	178
CONCLUSIONES Y RECOMENDACIONES.....	179
Bibliografía.....	182
ANEXO A.....	185
Esquemas eléctricos.....	186
ANEXO B.....	189

Plano mecánico del Rugosímetro.....	190
ANEXO C.....	191
Hoja de datos de los componentes microcontrolador 16f877a y MAX232.....	192
ANEXO D.....	195
Recomendaciones para construcción de punta de diamante.....	196

Resumen

La presente tesis de Ingeniería trata sobre el diseño y construcción de un Rugosímetro Digital con visualización en tres dimensiones comandada con microcontrolador e interfaz a la PC basada en el programa LabView mediante el puerto serial.

El equipo está diseñado para calcular los valores principales de rugosidad media de piezas maquinadas, que forman parte de maquinarias industriales en general, estando en capacidad de competir con equipos similares existentes en el mercado, considerando por este motivo, que el presente trabajo de tesis es un importante aporte al desarrollo tecnológico del medio.

Cuenta además con la ventaja de poseer un sistema de instrumentación virtual, lo que reduce su costo y su tamaño ya que puede conectarse a un ordenador normal, facilitando de esta manera su operación al usuario.

Abstract

In this work, the design and construction of a digital rugosimeter is presented. This device includes 3-D-Vision commanded by a microcontroller, interfaced to a PC by a serial port. For this purpose the Software Labview was used.

This equipment calculates the average roughness parameters of parts of industrial machines. The apparatus developed has the same features of similar equipments used for similar purposes in the industrial field.

This fact, joined with the advantage of a virtual instrumentation system, present an important contribution to the technological development of electronical devices.

Besides, this is a low- cost equipment, able to be connected to a normal PC, easy to operate for the user.

Maldonado Fajardo Hernán Santiago

Villota Verdugo Mauricio Efraín

Ing Hernán Viteri Cerda

Junio 2007

DISEÑO Y CONSTRUCCIÓN DE UN RUGOSÍMETRO DIGITAL EN 3D CON INTERFACE A LA PC BASADO EN LABVIEW V8.1 CON APLICACIÓN EN PULIDO DE PIEZAS METÁLICAS

INTRODUCCION

Debido al gran crecimiento industrial y tecnológico, la maquinaria también ha tenido que evolucionar y mejorar, esto obliga a construir piezas de mejor calidad en su acabado final, ya sean rodamientos, engranajes, herramientas, etc. Las piezas existentes en una máquina pueden estar en contacto unas con otras, ya sea en movimiento o fijas, estas producirán desgaste, por esta razón, se debe prestar especial atención a una característica que todos los materiales poseen, y es la Rugosidad. La rugosidad es el conjunto de irregularidades de la superficie real de una pieza, de esta depende su funcionalidad, en la industria, existe una amplia gama de niveles de rugosidad utilizadas, como ejemplo se puede mencionar la rugosidad ocupada como punto importante en llantas de autos, pistones de motores, baldosas de pisos, dentaduras, dermis y epidermis, maquinaria industrial, etc.

El presente trabajo de tesis final detalla el diseño y construcción de un Rugosímetro digital comandado mediante microcontrolador y con interfaz a la PC via comunicación serial RS232 con el programa LabView, para medir los parámetros principales de rugosidad tales como la desviación media aritmética del perfil, la altura de las irregularidades en diez puntos y la altura máxima del perfil, así como también, graficar su perfil tanto en dos como en tres dimensiones para su mejor observación.

Para el desarrollo del equipo se dividió a la tesis en cuatro capítulos principales: el primer capítulo trata todo lo referente a Rugosidad en piezas metálicas, esto incluye, cuantificación de la rugosidad, parámetros de piezas maquinadas, parámetros de rugosidad, etc. Además se tratará la forma en que se mide rugosidad, en otras

palabras, los tipos de sensores que existen y la forma de escoger cada uno de acuerdo a nuestras exigencias y la correcta selección de un rugosímetro.

El capítulo dos trata sobre el diseño del circuito electrónico completo, esto es un circuito de control, un circuito de fuerza, y la fuente de alimentación, el diseño implica, el diseño esquemático de los circuitos, y los circuitos impresos de los mismos. También sobre el tipo de transmisión que se ocupa para la transferencia de datos PC-PIC, en este caso transmisión serial RS-232; así como motores de corriente continua y como se manejan.

El capítulo tres trata sobre el diseño del software, tanto del PC el cuál está realizado en LabView, como el del microcontrolador el cuál está creado en PICBasic Pro 2.44. Para la elaboración de los dos programas es necesario tener conceptos básicos de LabView, PicBasic Pro y el microcontrolador PIC16F877, por esta razón, se hizo una breve introducción a los dos lenguajes de programación indicando la forma en que se estructuran, indicando también los registros que se usan del microcontrolador.

El capítulo cuatro habla en primera instancia sobre los aspectos constructivos de las tarjetas de circuito impreso, y la forma de construcción del Rugosímetro, teniendo en cuenta que como es un Rugosímetro en 3D, este tiene que tener movimiento en tres ejes, X, Y y Z. En segunda instancia, trata sobre la validación del equipo, esto se logra al realizar pruebas con un Rugosímetro patrón, y compararlas con las pruebas del equipo construido. Se incluye además las conclusiones obtenidas a lo largo del desarrollo de la presente tesis, así como también las recomendaciones de los autores.

Por último los autores consideramos que el desarrollo de una tesis de graduación implica no solamente la aplicación de los conocimientos adquiridos en las aulas universitarias sino la oportunidad de demostrar que podemos ser constructores de conocimiento, lo que queda demostrado con el presente trabajo.

CAPITULO I

RUGOSIDAD

1.1 Generalidades

Dentro del mundo de la tecnología, se observan gran cantidad de piezas que han de ponerse en contacto con otras y rozarse a altas velocidades. El acabado final y la textura de una superficie son de gran importancia e influencia para definir la capacidad de desgaste, lubricación, resistencia a la fatiga y aspecto externo de una pieza o material, por lo que la rugosidad es un factor importante a tener en cuenta.

La importancia que tiene el conocimiento de la rugosidad a nivel atómico de capas finas de metales es muy grande. El rendimiento de las superficies reflectantes depende estrechamente de la rugosidad de esas superficies. La calidad superficial obtenida en un proceso de mecanizado viene afectada por diversos factores propios del proceso. Con la aparición de los procesos a alta velocidad y por las exigencias del mercado actual esta temática ha cobrado gran relevancia.

1.2 Rugosidad

El acabado superficial de los cuerpos puede presentar errores de forma macrogeométricos y microgeométricos. La rugosidad superficial es el conjunto de irregularidades de la superficie real (fig 1.1), definidas convencionalmente en una sección donde los errores de forma y las ondulaciones han sido eliminados.

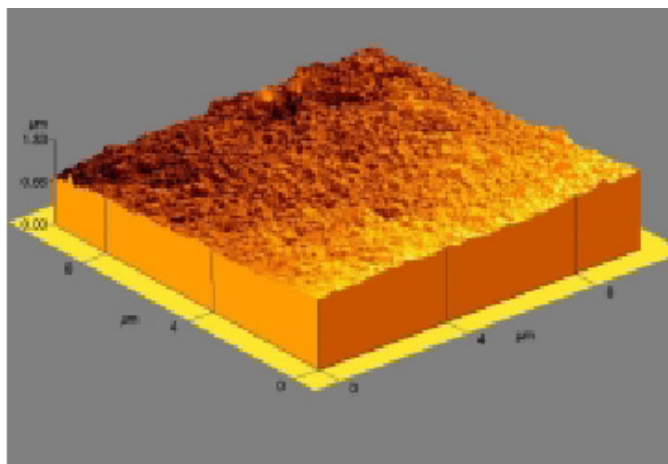


Fig1.1 Gráfico de una superficie rugosa¹

¹ Fuente: Moises Hinojosa, Martín Melo, La Rugosidad de las Superficies: Topografía, Pag. 25

En los procesos de maquinado, las características superficiales del producto influyen en su funcionalidad. La figura dominante en una superficie está influenciada por el método de maquinado, ya que cada tipo de herramienta de corte deja marcas distintivas en la superficie. Se pueden distinguir tres aspectos que influyen en la calidad de la superficie de los productos maquinados:

1. Condiciones y características de la herramienta.
2. Condiciones de operación de la máquina-herramienta.
3. Propiedades mecánicas de la pieza de trabajo.

El identificar la influencia que estos aspectos tienen en las superficies maquinadas permite mejorar los parámetros de corte, detectar eventuales fallas de maquinado (tales como vibraciones, malas sujeciones, etc) y encontrar situaciones de trabajo que den como resultado una mayor calidad en el producto. Una metodología para poder encontrar estos parámetros es identificando las componentes de la huella de la herramienta y las características de rugosidad asociadas a las propiedades del material al ser maquinado.

1.3 Importancia de cuantificar la rugosidad

Tanto en aplicaciones industriales como en la vida cotidiana, el grado de rugosidad en las superficies es importante, en ocasiones es deseable tener una rugosidad “alta” y en otras ocasiones esta condición es indeseable. En algunos casos se busca que la superficie del producto terminado presente un mínimo de rugosidad, ya que esto le da brillo, mejor apariencia y disminuye la fricción de la superficie al estar en contacto con otra, reduciendo el fenómeno del desgaste y la corrosión de dichos materiales.

Por otra parte, la fricción entre dos superficies es lo que permite sujetar un objeto sin que este se resbale. Es la rugosidad de los neumáticos de los automóviles lo que favorece la fricción entre ellos y el suelo, permitiendo de esta manera el agarre y el avance controlado y seguro. En ocasiones se busca maximizar el área superficial, lo que se consigue incrementando la rugosidad, como en el caso de los catalizadores, cuya eficiencia es mejor mientras mayor sea la superficie de contacto con los

reactivos. Es la rugosidad de los “acetatos”, principalmente lo que determina si puede utilizarse en una impresora láser o en una impresora de inyección de tinta.

1.4 Parámetros de la superficie de una pieza maquinada

Antes de revisar la rugosidad, se debe tomar en cuenta los diferentes parámetros existentes en una superficie, ya que estos ayudarán a comprender de mejor manera la utilidad e importancia que tiene la medición de la rugosidad.

1.4.1 Superficie real (fig1.2).- Superficie que limita el cuerpo del medio que lo separa. Esta superficie es obtenida a ser una pieza sometida a un corte o a un proceso de producción, arranque y fatiga superficial.

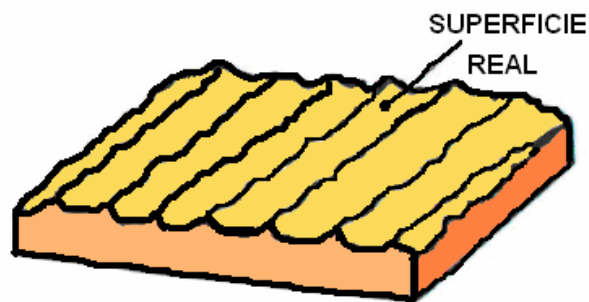


Fig1.2 Superficie real de una pieza

1.4.2 Superficie geométrica (fig1.3).- Superficie ideal cuya forma está especificada por el dibujo y/o todo documento técnico. Este tipo de superficie muestra la manera específica en la que debe quedar la pieza maquinada en teoría.

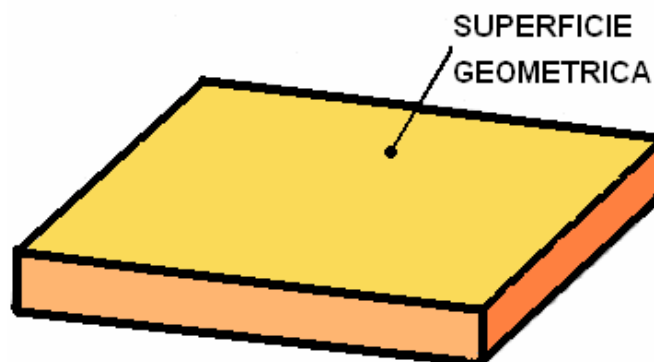


Fig1.3. Superficie Geométrica

1.4.3 Superficie de referencia.- Superficie a partir de la cual se determinan los parámetros de rugosidad. Tiene la forma de la superficie geométrica. Se puede calcular por el método de mínimos cuadrados.

1.4.4 Perfil real (fig1.4).- Es la intersección de la superficie real con un plano normal. Este perfil da una idea clara de cuan perfecta es la superficie de la pieza que estamos analizando.

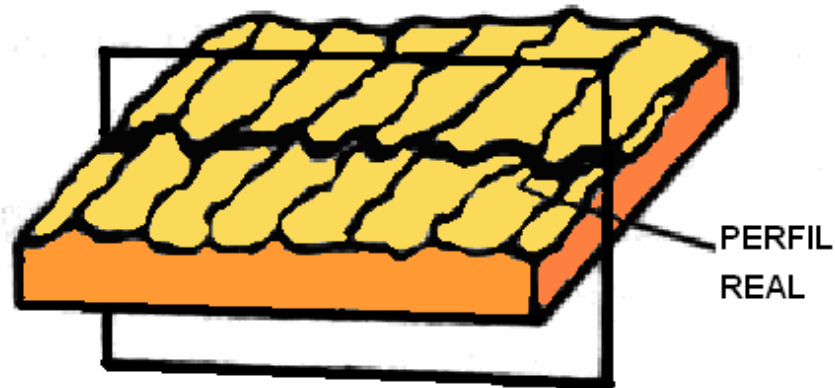


Fig1.4 Perfil real

1.4.5 Longitud básica (l) (fig 1.5 (a)).- Longitud de la línea de referencia utilizada para separar las irregularidades que forman la rugosidad superficial. Esta longitud específica y separa cada irregularidad existente en el perfil real de una pieza, suelen ser de separación simétrica entre ellas.

1.4.6 Longitud de evaluación (l_n) (fig 1.5 (b)).- Longitud utilizada para determinar los valores de los parámetros de rugosidad superficial. Puede comprender una o más longitudes básicas.

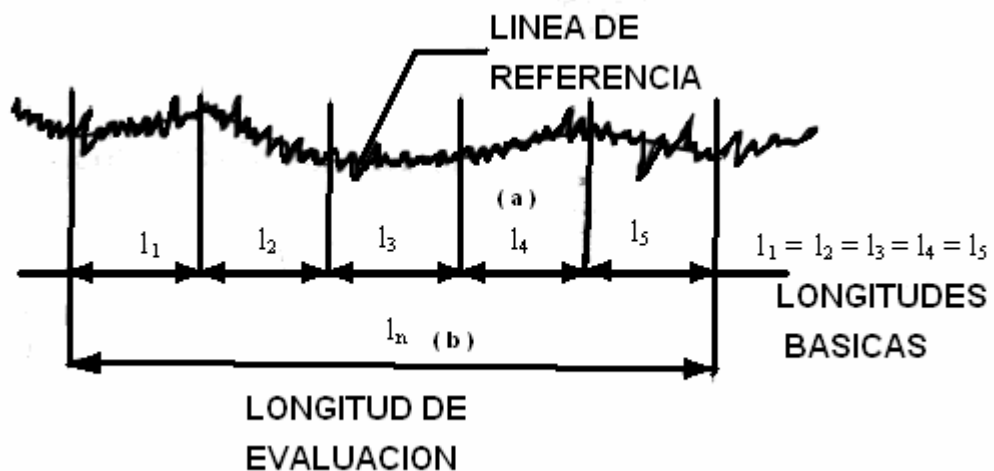


Fig1.5 Longitud básica (a) y de evaluación (b) ²

² Fuente: Manrique, E., Casanova, A., Metrología básica, Pag. 158

1.4.7 Línea media de los mínimos cuadrados (fig1.6).- Línea de referencia cuya forma es la del perfil geométrico. Divide el perfil de modo que, en el interior de la longitud básica, la suma de los cuadrados de las desviaciones a partir de esta línea es mínima.

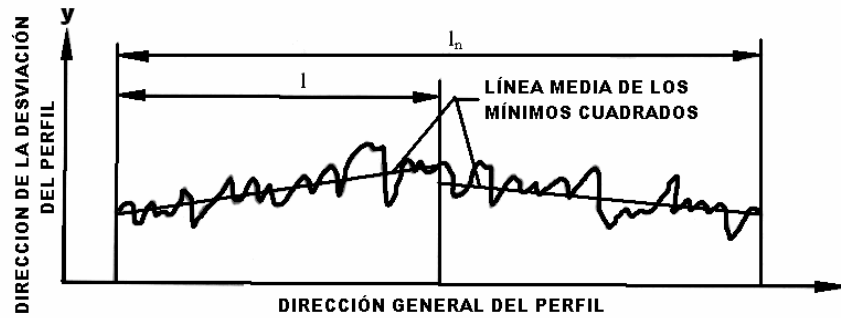


Fig1.6 Línea media de los mínimos cuadrados³

1.4.8 Línea media aritmética (o línea central).- Línea de referencia con la forma del perfil geométrico, paralela a la dirección general del perfil en el interior de la longitud básica. Divide el perfil de modo que la suma de áreas comprendidas entre ella y el perfil es igual en la parte superior e inferior (Fig 1.7).

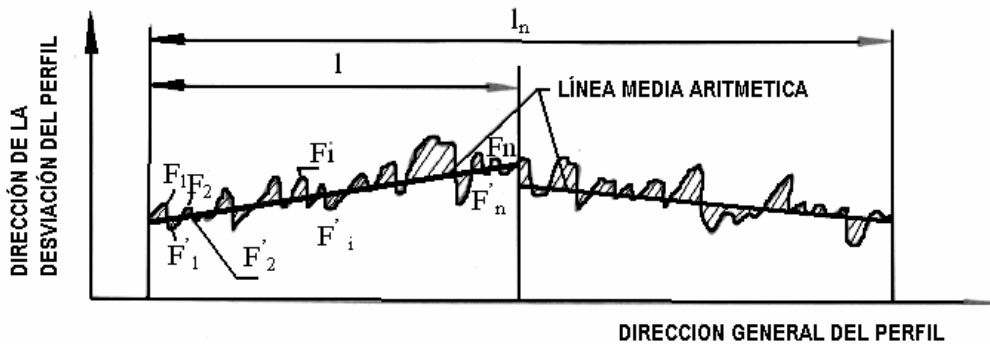


Fig1.7 Línea media aritmética⁴

La fórmula para encontrar la línea aritmética es:

$$\sum_{i=1}^n F_i = \sum_{i=1}^n F'_i \tag{1.1}$$

Donde:

F_i = áreas de la parte superior
 F'_i = áreas de la parte inferior

³ Fuente: Manrique, E., Casanova, A., Metrología básica, Pag. 162

⁴ Fuente: Manrique, E., Casanova, A., Metrología básica, Pag. 166

1.4.9 Cresta local del perfil.- Es la parte del perfil comprendida entre dos mínimos adyacentes del perfil (Fig 1.8 (a)).

1.4.10 Valle local del perfil.- Parte del perfil comprendida entre dos máximos adyacentes del perfil (Fig 1.8 (b)).

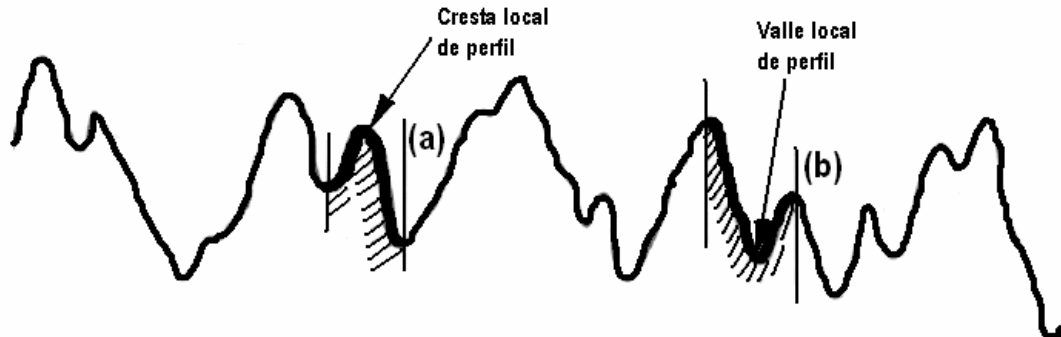


Fig1.8 Cresta local de perfil (a) y Valle local de perfil (b)⁵

1.4.11 Cresta del perfil.- Parte del perfil dirigida hacia el exterior del cuerpo uniendo dos intersecciones consecutivas del perfil con la línea media (Fig 1.9 (a)).

1.4.12 Valle del perfil.- Parte del perfil dirigida hacia el interior del cuerpo uniendo dos intersecciones consecutivas del perfil con la línea media.

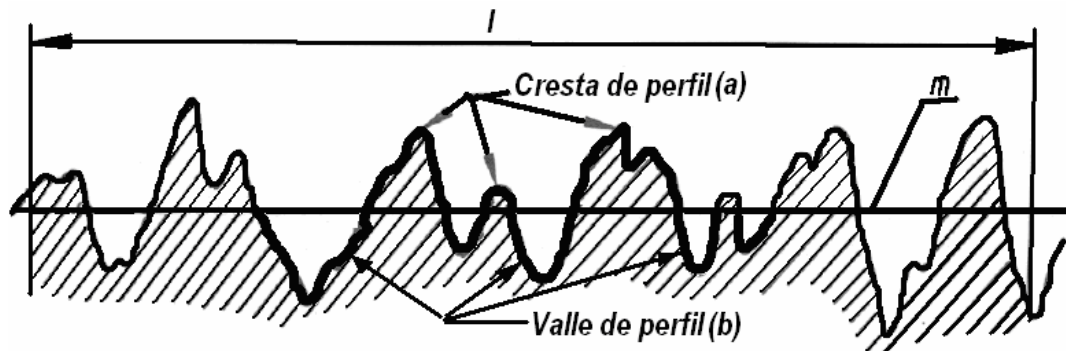


fig1.9 Cresta de perfil (a) y Valle de perfil (b)⁶

1.5 Parámetros de rugosidad.

En general, los parámetros utilizados para cuantificar la rugosidad pueden interpretarse como parámetros propios de la distribución estadística de alturas de perfil o superficie bajo análisis. Antes de analizar los parámetros de rugosidad es importante distinguir entre el análisis de rugosidad propiamente dicha y otros

⁵ Fuente: Manrique, E., Casanova, A., Metrología básica, Pag. 168

⁶ Fuente: Manrique, E., Casanova, A., Metrología básica, Pag. 171

componentes la textura o morfología como la ondulación, curvatura y la inclinación o tendencia.

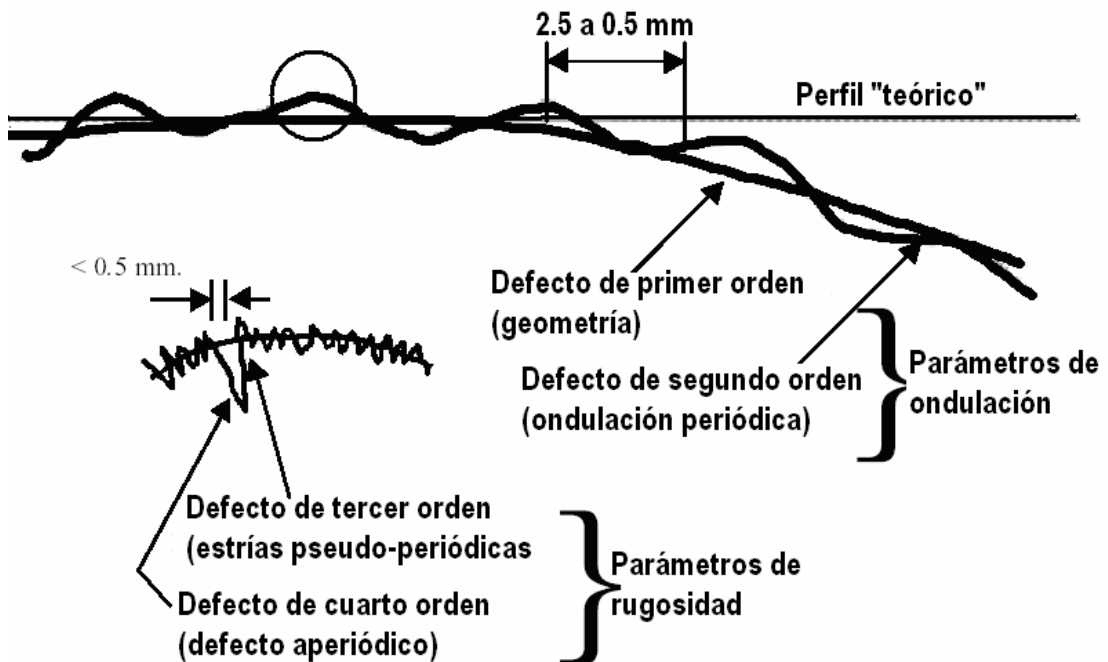


fig1.10. Los cuatro primeros órdenes de defectos topográficos en una superficie con curvatura e inclinación⁷

En la figura 1.10, se puede observar que una superficie puede poseer curvatura y/o ondulación periódica y aperiódica, dichos componentes deben ser eliminados o extraídos antes de cuantificar la rugosidad.

Dicha figura, ilustra los órdenes de defectos topográficos que distinguen algunos autores. La desviación del perfil respecto a la figura esperada (en el maquinado por ejemplo) se considera el defecto de primer orden. La ondulación periódica a nivel de microscopio se considera defecto de segundo orden mientras que las estrías pseudo periódicas pertenecen al tercer orden. En cuarto orden se tienen los defectos aperiódicos.

Los parámetros de rugosidad podemos subdividir en dos partes, con respecto a la dirección de las alturas, a la dirección transversal y a la forma de las irregularidades.

⁷ Fuente: Vicente Portela, Metrología Dimensional, Pag. 169

1.5.1 Parámetros de rugosidad con respecto a la dirección de las alturas.-

1.5.1.1 Altura de una irregularidad del perfil.- Suma de la altura de una cresta y de la profundidad de un valle adyacente (Fig1.11).

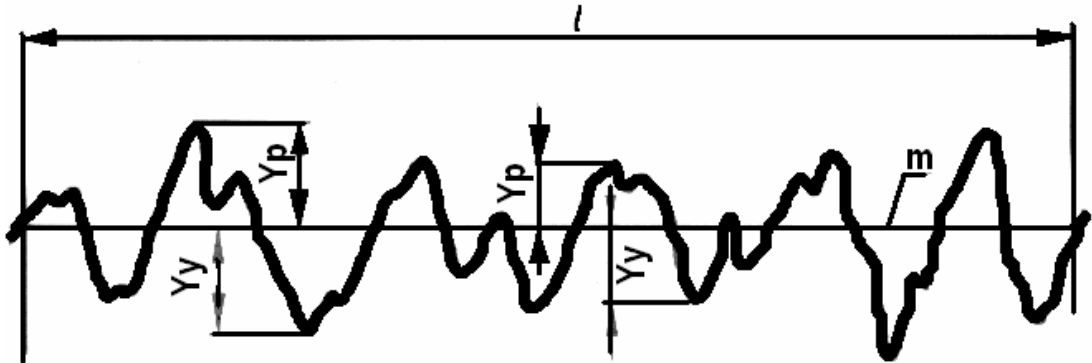


fig1.11 Altura de una irregularidad⁸

1.5.1.2 Altura máxima de una cresta (R_p).- Distancia del punto más alto del perfil a la línea media, dentro de la longitud con abreviatura l (Fig1.12).

1.5.1.3 Profundidad máxima de un valle (R_m).- Distancia del punto más bajo del perfil a la línea media, dentro de la longitud l (Fig1.12).

1.5.1.4 Altura máxima del perfil (R_y).- Máxima distancia entre la cresta más alta (R_p) y el valle más bajo (R_m) (Fig1.12).

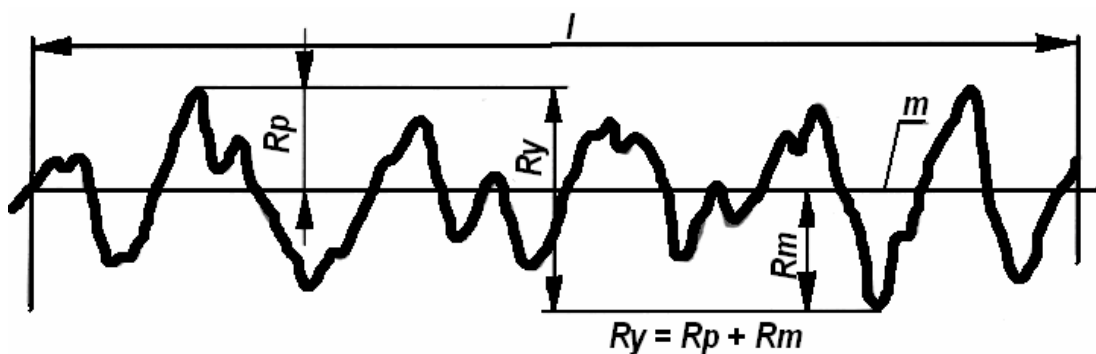


fig1.12 Altura y profundidad máxima de perfil⁹

⁸ Fuente: Vicente Portela, Metrología Dimensional, Pag. 172

⁹ Fuente: Vicente Portela, Metrología Dimensional, Pag. 175

1.5.1.5 Altura de las irregularidades en diez puntos (Rz).- Media de los valores absolutos de las alturas de las cinco crestas del perfil más altas y de las profundidades de los cinco valles del perfil más bajos, dentro de la longitud básica (Fig1.13).

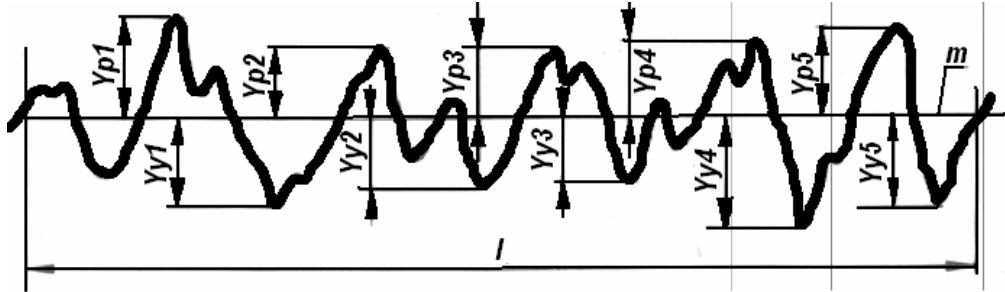


fig1.13 Parámetro de rugosidad Rz¹⁰

Matemáticamente el parámetro de rugosidad Rz se especifica de la siguiente manera:

$$Rz = \frac{\sum_{i=1}^5 |Y_{pi}| + \sum_{i=1}^5 |Y_{yi}|}{5} \quad (1.2)$$

donde:

Y_{pi} es la altura de la *i*ésima cresta más alta;

Y_{yi} es la profundidad del *i*ésimo valle más bajo.

5 es el número de crestas y de valles del perfil.

1.5.1.6 Altura media de las irregularidades del perfil (Rc).- Es la suma de los valores medios de las alturas de las crestas y de las profundidades de los valles, dentro de la longitud básica. Matemáticamente se tiene:

$$Rc = \frac{\sum_{i=1}^n |Y_{pi}| + \sum_{i=1}^n |Y_{yi}|}{n} \quad (1.3)$$

donde:

Y_{pi} es la altura de la *i*ésima cresta más alta;

Y_{yi} es la profundidad del *i*ésimo valle más bajo.

n es el n° de crestas y de valles del perfil.

¹⁰ Fuente: Vicente Portela, Metrología Dimensional, Pag. 177

1.5.1.7 Desviación media aritmética del perfil (Ra).- Es la media aritmética de los valores absolutos de las desviaciones del perfil, en los límites de la longitud básica l (Fig 1.14).

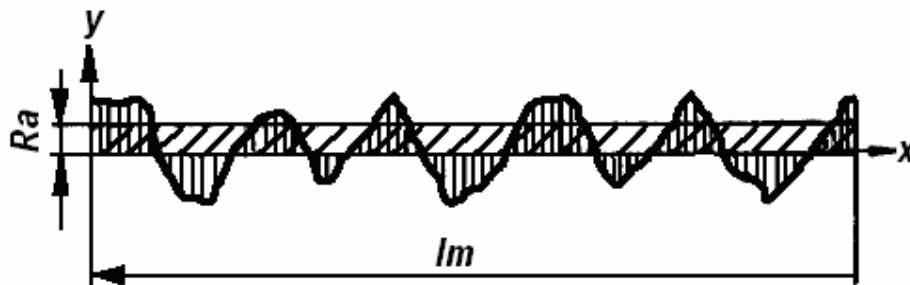


fig1.14 Desviación media aritmética¹¹

El valor promedio aritmético de todas las distancias absolutas del perfil de rugosidad R desde la línea central dentro de la longitud de medición se puede expresar matemáticamente de la siguiente manera:

$$Ra = \frac{1}{l_m} \int_{x=0}^{x=l_m} |y| dx \quad (1.4)$$

Donde:

Ra = Desviación media aritmética del perfil

l_m = Longitud básica

Y = Valores de desviación de perfil

1.5.2 Parámetros de rugosidad respecto a la dirección transversal.

1.5.2.1 Paso de las irregularidades del perfil.- Longitud de la línea media que contiene una cresta y un valle consecutivo (Fig1.15).

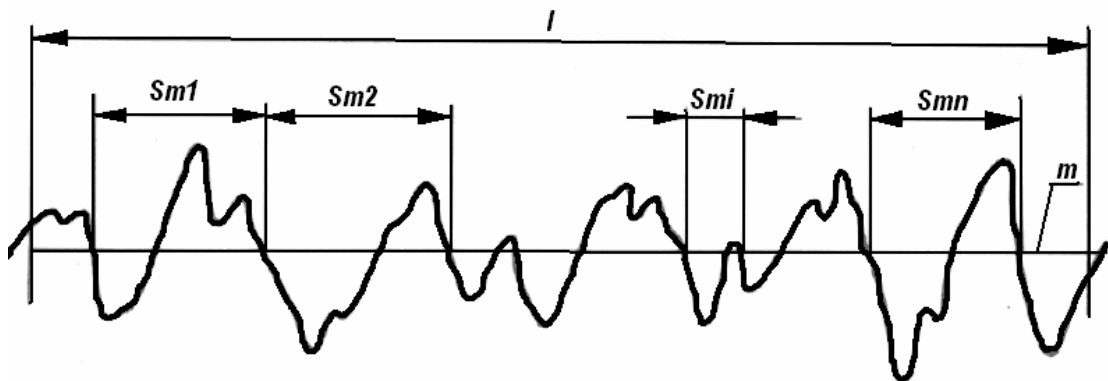


Fig1.15 Paso de las irregularidades¹²

¹¹ Fuente: Vicente Portela, Metrología Dimensional, Pag. 178

¹² Fuente: Vicente Portela, Metrología Dimensional, Pag. 179

1.5.2.2 Paso medio de las irregularidades del perfil.- Para definir el paso medio de las irregularidades de perfil utilizamos la siguiente ecuación:

$$S_m = \frac{1}{n} \sum S_{mi} \quad (1.5)$$

Donde:

S_m = paso medio de las irregularidades de perfil

n = número de pasos

S_{mi} = valor de paso de perfil

1.5.2.3 Paso de las crestas locales del perfil.- Longitud de una sección de la línea media delimitada por la proyección sobre esta línea de los dos puntos más altos de las crestas locales adyacentes del perfil (Fig1.16).

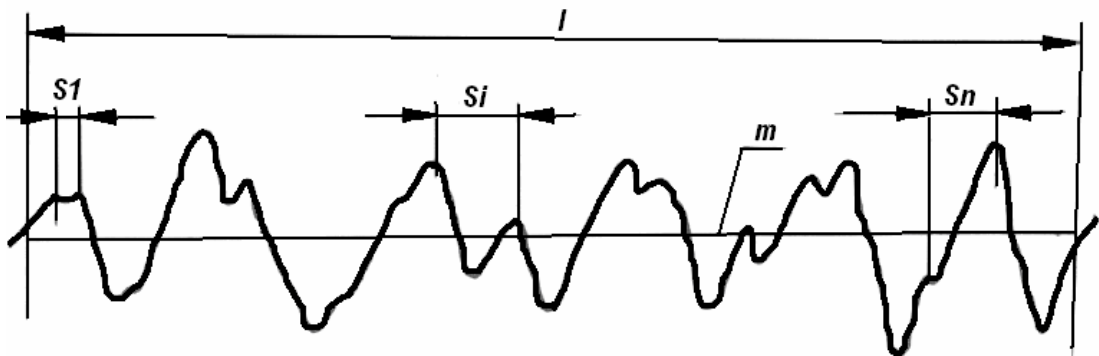


Fig1.16 Paso de las crestas locales del perfil¹³

1.5.2.4 Paso medio de las crestas del perfil.- Para definir el paso medio de las crestas de perfil se utiliza la siguiente ecuación:

$$S = \frac{1}{n} \sum S_i \quad (1.6)$$

Donde:

S = paso medio de las crestas de perfil

n = número de pasos

S_i = valor de paso de las crestas de perfil

1.5.2.5 Longitud desarrollada del perfil (L_o).- Longitud que se obtendría al desarrollar el perfil en línea recta(Fig1.17).

¹³ Fuente: Vicente Portela, Metrología Dimensional, Pag. 182

1.5.2.6 Relación de longitud del perfil (l_r).- Relación entre la longitud desarrollada y la longitud básica(Fig1.17).

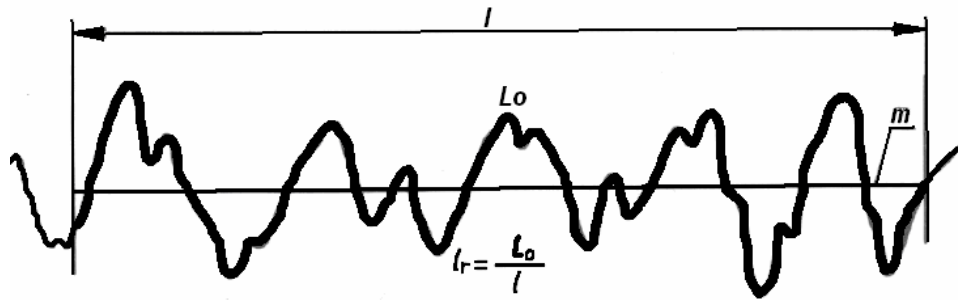


Fig1.17 Relación de longitud l ¹⁴

1.5.2.7 Densidad de las crestas del perfil (D).- Número de crestas por unidad de longitud, que se la puede expresar mediante la siguiente ecuación:

$$D = \frac{1}{S_m} \tag{1.7}$$

Donde:

- D = densidad de las crestas de perfil
- S_m = Valor del paso de perfil

1.5.3 Parámetros de rugosidad respecto a la forma de las irregularidades.

1.5.3.1 Longitud portante del perfil (μ_p).- Suma de las longitudes de segmentos obtenidos cortando las crestas por una línea paralela a la línea media, dentro de la longitud básica, por un nivel de corte dado (Fig1.18).

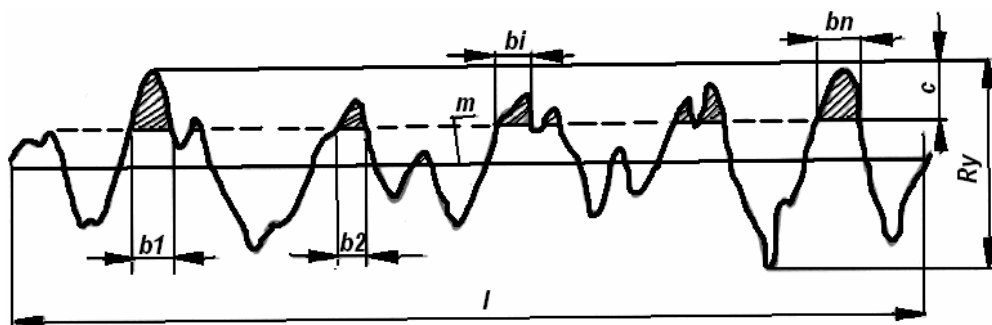


Fig1.18 Longitud portante del perfil ¹⁵

¹⁴ Fuente: Vicente Portela, Metrología Dimensional, Pag. 185

¹⁵ Fuente: Vicente Portela, Metrología Dimensional, Pag. 188

Matemáticamente se tiene:

$$\mu_p = b_1 + b_2 + b_3 + \dots + b_n \quad (1.8)$$

Donde:

μ_p = longitud portante del perfil

b_n = longitudes de los segmentos

1.5.3.2 Curva de la tasa de longitud portante del perfil.- Es el gráfico que representa la relación entre los valores de la tasa de longitud portante del perfil y el nivel de corte del perfil (Fig1.19).

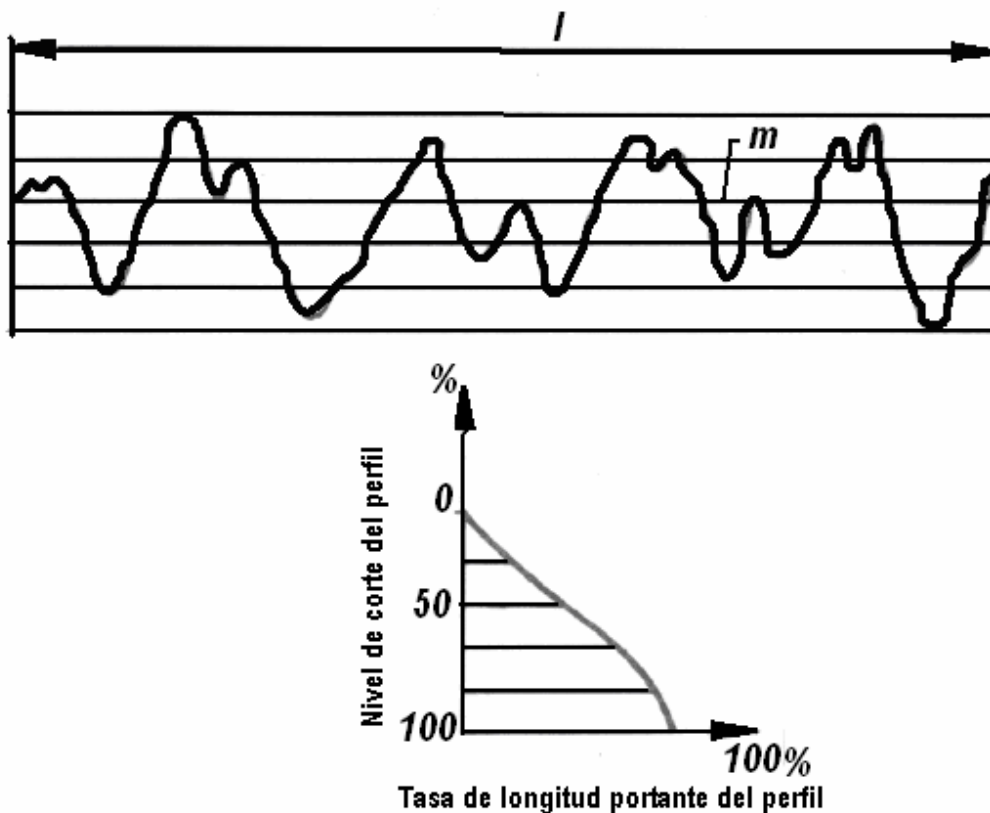


Fig1.19 Curva de la tasa de longitud portante¹⁶

1.6 Especificación de la rugosidad.

1.6.1 Representación.- El valor que define la rugosidad, normalmente expresado en micras (μ), se colocará sobre los símbolos como se refleja en la figura 1.20

¹⁶ Fuente: Vicente Portela, Metrología Dimensional, Pag. 193

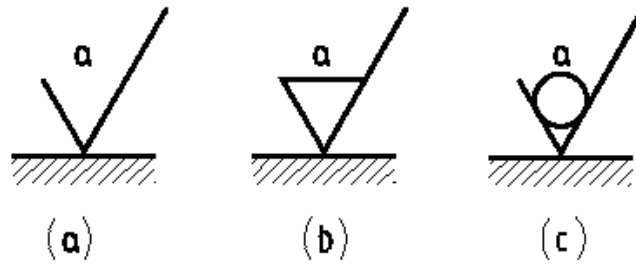


Fig1.20 Representación de rugosidad¹⁷

- (a) La superficie obtenida en la figura puede obtenerse por medio de cualquier proceso de fabricación.
- (b) La superficie debe obtenerse por medio de procedimientos con arranque de viruta.
- (c) La superficie debe realizarse sin arranque de viruta.

Cuando se especifica un solo valor, éste se refiere al máximo valor permitido de rugosidad superficial. Si fuera necesario establecer criterios de valor máximo y mínimo para la rugosidad, deben indicarse como en la figura situando el valor máximo encima del mínimo(Fig1.21a).

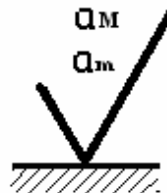


Fig1.21a Indicación de valor de rugosidad¹⁸

En la figura 1.21b expresamos un ejercicio con lo antes mencionado:

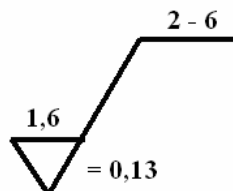


Fig1.21b Ejemplo de representación¹⁹

¹⁷ Fuente: Comité de Metrología de Madrid, Consejos de metrología, Pag. 75

¹⁸ Fuente: Comité de Metrología de Madrid, Consejos de metrología, Pag. 78

¹⁹ Fuente: Comité de Metrología de Madrid, Consejos de metrología, Pag. 81

El significado de cada número y símbolo es el siguiente:

1, 6 es el valor Ra de la rugosidad en μm .

2 es el valor de la altura de la ondulación (no necesario).

= es la orientación de la rugosidad (en este caso paralela a la línea).

0, 13 es el paso de la rugosidad en μm (no necesario)

6 es el valor del paso de la ondulación en mm (no necesario).

1.6.2 Simbología.- A continuación se presenta una tabla 1.1 de la simbología más utilizada en cuanto a representación de parámetros de rugosidad:

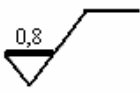
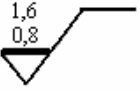
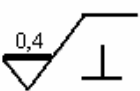
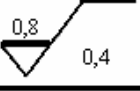
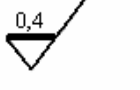
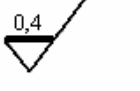
SÍMBOLO	ORIENTACIÓN DE LA RUGOSIDAD	EJEMPLO	RUGOSIDAD
=	Paralela a la línea representativa de la superficie sobre la que el símbolo va indicado		Rugosidad máxima Ra = 0,8 μm .
⊥	Perpendicular a la línea representativa de la superficie sobre la que el símbolo va indicado		Rugosidad comprendida entre Ra = 0,8 μm y Ra = 1.6 μm
×	Cruzado respecto a la línea representativa de la superficie sobre la que el símbolo va indicado		Rugosidad máxima Ra = 0,4 μm en la orientación
M	Multidireccional		Rugosidad máxima Ra = 0,8 μm con paso de la rugosidad de 0,4mm
C	Aproximadamente circular respecto al centro de la superficie sobre la que el símbolo va indicado		Rugosidad máxima Ra = 0,4 μm con altura de 1mm
R	Aproximadamente radial respecto al centro de la superficie sobre la que el símbolo va indicado		Rugosidad máxima Ra = 0,4 μm con altura de 1mm y paso de la ondulación de mm

Tabla1.1 Simbología de Rugosidad²⁰

²⁰ Fuente: Comité de Metrología de Madrid, Consejos de metrología, Pag. 92

1.6.3 Clases de rugosidad.- Los valores de rugosidad pueden indicarse bien mediante los números de las clases de rugosidad correspondientes, que aparecen en la tabla siguiente, bien mediante el valor numérico de la rugosidad, expresado en micras. Se recomienda, aunque no es obligatorio, usar preferentemente las clases de rugosidad en lugar de los valores reales de ésta para evitar errores de interpretación del valor numérico (tabla 1.2).

Rugosidad R_a (μm)	Clase de rugosidad
50	N 12
25	N 11
12.5	N 10
6.3	N 9
3.2	N 8
1.6	N 7
0.8	N 6
0.4	N 5
0.2	N 4
0.1	N 3
0.05	N 2
0.025	N 1

Tabla 1.2 Clases de rugosidad.

1.6.4 Aplicaciones de las distintas clases de rugosidad.-Cada una de las distintas clases de rugosidad tiene aplicaciones diferentes, a continuación se presenta algunas de ellas:

N1 Espejos. Bloques patrón

N2 Planos de apoyo de relojes comparadores.

N3 Herramientas de precisión. Cojinetes superacabados. Acoplamientos estancos de alta presión en movimiento alternativo. Superficies bruñidas de retención sin retén.

N4 Soportes de cigüeñales y árboles de levas. Pies de válvulas. Superficies de cilindros de bombas hidráulicas. Cojinetes lapeados. Pernos de árboles para rotores de turbinas, reductores.

N5 Árboles acanalados. Superficie exterior de pistones. Acoplamientos efectuados a presión. Asientos de válvulas..

N6 Tambores de freno. Agujeros brochados. Cojinetes de bronce. Dientes de engranaje. Superficies de piezas deslizantes, como patines y sus guías.

N7 Caras de engranajes. Árboles y orificios de engranajes. Cara de émbolo.

N8 Pernos y cojinetes para transmisión (montaje a mano). Superficies de acoplamiento de partes fijas desmontables.

N9 Superficies laterales de retención con retenes normales.

1.6.5 Proceso de fabricación.-En el proceso de fabricación de una pieza se toma muy en cuenta la desviación media aritmética del perfil Ra, ya que esta debe estar en los límites establecidos dependiendo del trabajo que se ha realizado a la pieza. Dicho esto, se presenta la tabla 1.3 de los procesos de fabricación y su rugosidad frecuente y menos frecuente:

PROCESO DE FABRICACION	RUGOSIDAD													
	Ra μm	1	2	4	8	16	32	64	120	250	500	800	1000	
	Ra μm	0,025	0,05	0,1	0,2	0,4	0,8	1,6	3	6	12	16	20	25
Desbarbado. Oxicrote														
Torneado. Fresado. Limado														
Taladrado														
Cepillado														
Mandrinado														
Brochado														
Rectificado														
Bruñido														
Superacabado (lapeado)														
Pulido														
Granallado														
Laminación en caliente														
Moldeo en arena														
Forja														
Moldeo en coquilla														
Moldeo a presión														
Laminado, estirado, trefilado														
Extrusión														

Tabla 1.3 Procesos de fabricación²¹

Siendo la parte cuadrículada la rugosidad más frecuente y la rayada menos frecuente.

²¹ Fuente: Comité de Metrología de Madrid, Consejos de metrología, Pag. 139

1.7 Métodos para medir la rugosidad.

El método más utilizado en la industria para cuantificar la rugosidad se basa en el registro de perfiles de alturas mediante un rugosímetro o perfilómetro. El tratamiento de estos perfiles ayudan a determinar los parámetros de rugosidad como la Desviación media aritmética del perfil (R_a) y Altura de las irregularidades en diez puntos (R_z).

Existen otros tipos de instrumentos para medir la rugosidad tales como el Microscopio de Fuerza Atómica (MFA), que aplicado a observaciones topográficas puede considerarse como un perfilómetro de resolución sub-anométrica, Típicamente el MFA permite longitudes de muestreo máximas de $100\mu\text{m}$ con resolución nanométrica.

1.7.1 Rugosímetros.- El rugosímetro o perfilómetro es el equipo más utilizado en la industria en general para medir la rugosidad y sus componentes antes mencionados. En la fig1.22 se puede observar un rugosímetro con todos sus componentes.



Fig1.22 Rugosímetro o Perfilómetro

1.7.1.1 Principio de funcionamiento.- El principio de operación de un rugosímetro es muy simple: una fina punta en contacto con la superficie a ser analizada realiza un barrido en línea recta y las variaciones de alturas se convierten en señales eléctricas y estas a su vez se registra y grafican mediante un interfaz a un computador (Fig1.23).

Es posible construir imágenes de la topografía en 3D mediante barridos sucesivos, esta técnica es denominada “Microscopía Mecánica de Barrido”.

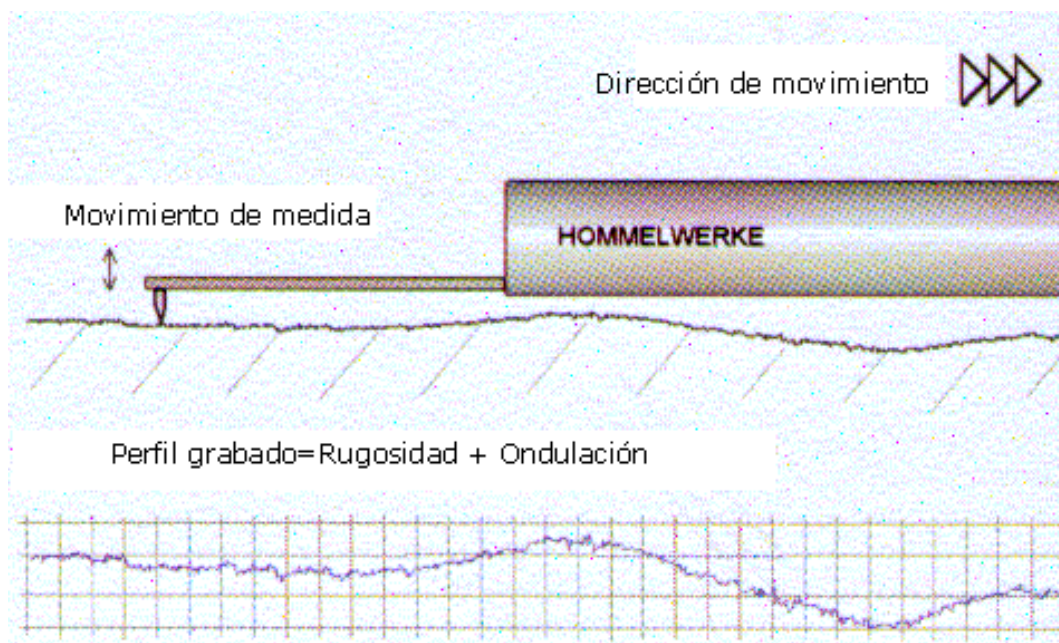


Fig1.23 Principio de funcionamiento de un Rugosímetro²²

Un parámetro muy importante de estos equipos es la forma y el radio de la punta que se utiliza en el barrido, ya que esta influye de forma importante en la resolución lateral de las mediciones. Típicamente se utilizan puntas con radios de algunos micrómetros, $2\ \mu\text{m}$ es un valor común.

Aunque la resolución vertical es generalmente menor que el radio de la punta, no pueden detectarse con mayor precisión valores de rugosidad menores al radio de la punta. En general, una punta burda o desgastada resulta en valores de rugosidad más bajos que los obtenidos por puntas finas.

²² Fuente: Moises Hinojosa, Martín Melo, La Rugosidad de las Superficies: Topografía, Pag. 68

1.7.1.2 Tipos de Rugosímetros.- Existen diferentes tipos de rugosímetros dependiendo del palpador o sensor que utilizan para la medición de la rugosidad. A breves rasgos se analizarán algunos de estos tipos de rugosímetros y sus palpadores.

1.7.1.3 Rugosímetro de palpador mecánico.- Instrumento para la medida de la calidad superficial basado en la amplificación eléctrica de la señal generada por un palpador que traduce las irregularidades del perfil de la sección de la pieza (fig 1.24)

Sus elementos principales son el palpador, el mecanismo de soporte y arrastre de éste, el amplificador electrónico, un calculador y un registrador.

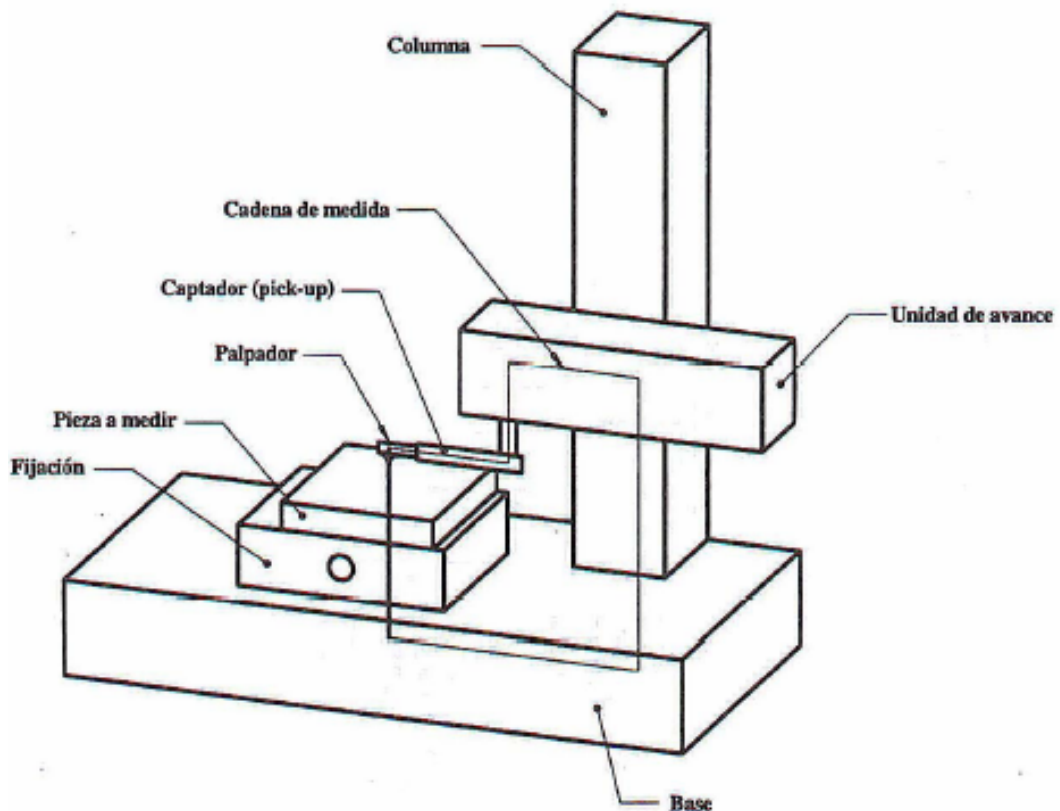


Fig1.24 Configuración mecánica de un Rugosímetro con palpador mecánico.

1.7.1.4 Rugosímetro de palpador inductivo.- En los rugosímetros de palpador inductivo, el principio de funcionamiento es el siguiente: El desplazamiento de la aguja al describir las irregularidades del perfil modifica la longitud del entrehierro el

circuito magnético, y con ello el flujo de campo magnético que lo atraviesa, generando una señal eléctrica (Fig1.25).

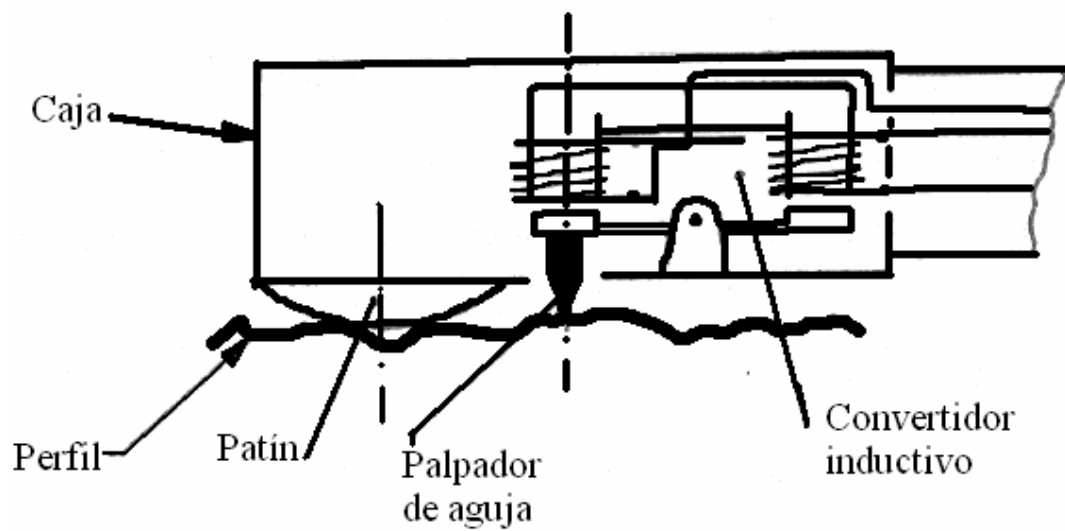


Fig1.25 Rugosímetro con palpador inductivo

1.7.1.5 Rugosímetro de palpador capacitivo.- El desplazamiento vertical del palpador aproxima las dos láminas de un condensador, modificando su capacidad y con ella la señal eléctrica (Fig1.26)

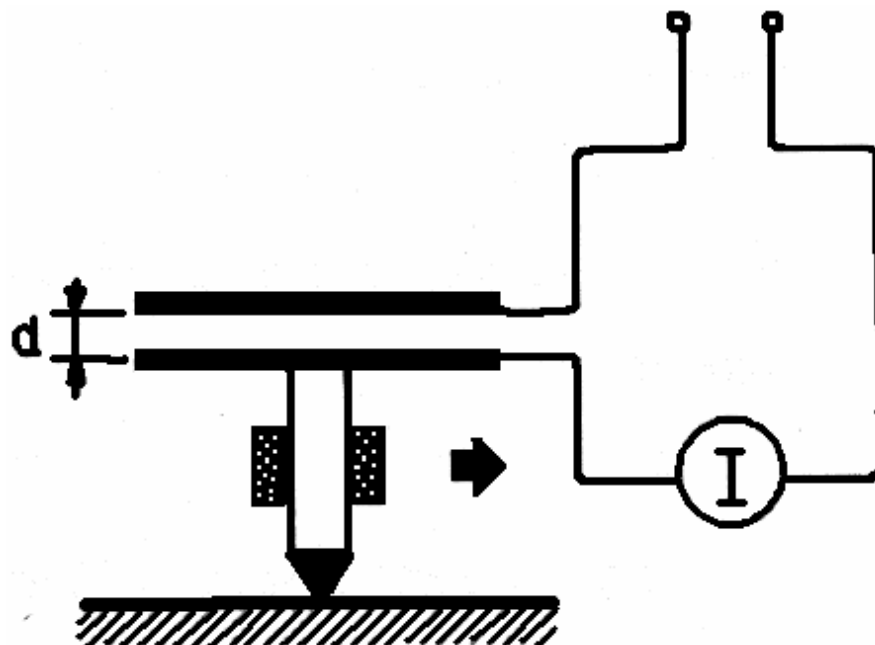


Fig1.26 Rugosímetro con palpador capacitivo

1.7.1.6 Rugosímetro de palpador piezoeléctrico.- El desplazamiento de la aguja del palpador deforma elásticamente un material piezoeléctrico, que responde a dicha deformación generando una señal eléctrica (Fig1.27).

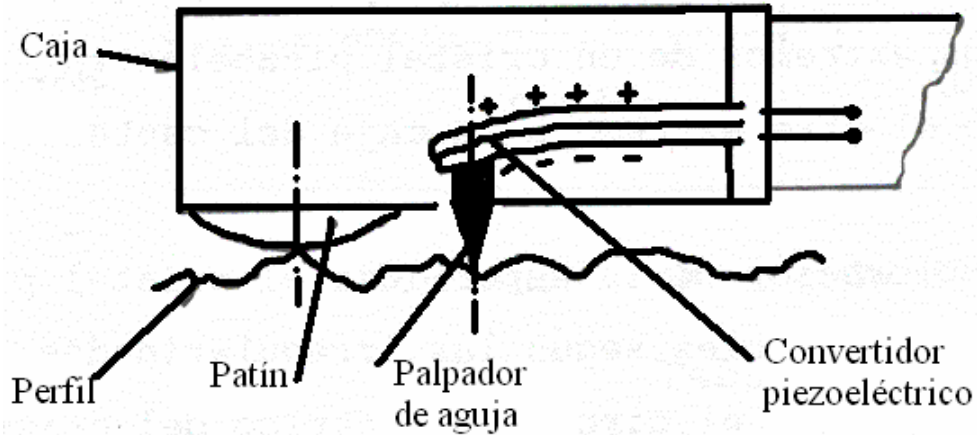


Fig1.27 Rugosímetro con palpador piezoeléctrico.

1.7.1.7 Rugosímetro con patín mecánico.- El patín describirá las ondulaciones de la superficie mientras la aguja recorre los picos y valles del perfil. Así se separan mecánicamente ondulación y rugosidad que son simplemente desviaciones respecto a la superficie geométrica con distinta longitud de onda. En la figura 38, la flecha indica el patín mecánico utilizado en estos rugosímetros (Fig1.28).

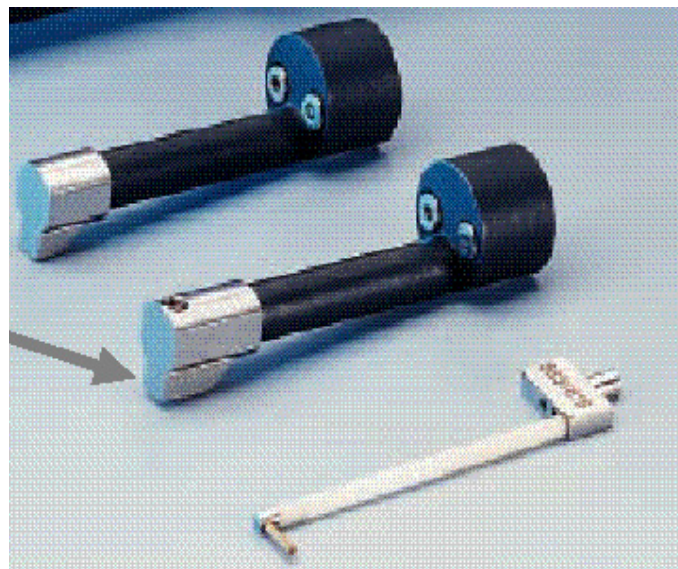


Fig1.28 Rugosímetro con patín mecánico²³.

²³ Fuente: Moises Hinojosa, Martín Melo, La Rugosidad de las Superficies: Topografía, Pag. 73

1.8 Selección del Rugosímetro

En el desarrollo de los siguientes capítulos estará enfocado al diseño y construcción tanto de la parte física como en lo que se refiere a software de control, de un rugosímetro digital utilizando un sensor óptico el cuál está adaptado a un micrómetro con palpador mecánico de aguja el cual permitirá obtener los datos de rugosidad de las piezas que se requieran.

Este Rugosímetro se asemeja a los anteriores en el palpador mecánico de aguja ya que en la mayoría de estos equipos se utiliza esta técnica en la toma de datos, lo que varía es la conversión del mismo. En este caso, se utilizó la ventaja del movimiento de los piñones internos de un micrómetro y se adaptó en él una placa para que tape al sensor óptico H21A al subir y bajar la punta del mismo, obteniendo una variación de voltaje lineal a la salida.

El acoplamiento del sensor óptico al micrómetro redujeron los costos del Rugosímetro, ya que un sensor de estas características en el mercado tiene un precio elevado.

El Rugosímetro presenta las siguientes ventajas:

- Visualización en tres dimensiones
- Palpador acoplado a un sensor óptico
- Tarjeta de fuente de poder
- Tarjeta de control del microcontrolador e interface con RS 232
- Tarjeta de fuerza para comando de motores.
- Cálculo rugosidad promedio Ra, y la profundidad de rugosidad Rz.
- Almacenar las gráficas obtenidas
- Impresión de datos obtenidos
- Selección de curvas individuales para su cálculo

CAPITULO II

DISEÑO DEL HARDWARE

2.1 Diseño del circuito general

2.1.1 Introducción teórica al programa Protel 99SE.- Protel junto con Orcad constituyen las dos empresas de software de CAD electrónico más importantes en entorno PC.

Protel es un sistema de diseño completo para entornos Windows y proporciona un conjunto de herramientas integradas formadas por un gestor de documentos integrado, un capturador de diseños esquemáticos, un simulador analógico basado en SPICE, un simulador digital basado en CUPL (lenguaje similar a ORCAD/PLD), una herramienta de diseño de placas de circuito impreso (PCBs) y soporte para trabajo en redes de ordenadores..

2.1.2 Conceptos Básicos.- Al utilizar el programa se puede distinguir las siguientes partes de la pantalla, las cuales se detallan a continuación:

a. Gestor de Diseño (fig 2.29).- Este panel muestra el árbol de navegación similar al Explorador de Windows, ofrece una vista de las bases de datos de diseño que se tiene abiertas y los documentos que contienen. Haciendo clic en los iconos + y – se puede expandir y contraer las carpetas.

Haciendo clic en un documento o carpeta se lo puede abrir o ver en la ventana de diseño principal. Si se hace clic con el botón derecho en un documento o carpeta aparece un menú de opciones. También se muestran en el panel del gestor de diseño paneles de edición específicos que aparecen como pestañas cuando se editan varios tipos de documentos.

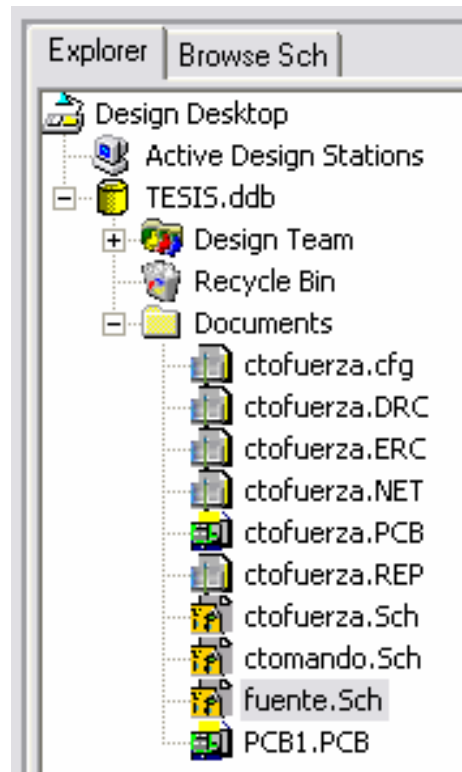


Fig2. 29 Gestor de diseño

b. Estaciones de diseño activas (fig2.30).- Haciendo clic en el símbolo + se puede ver una lista de ordenadores de la red que están ejecutando Protel.

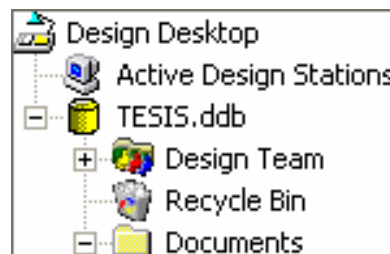


Fig2. 30 Estaciones de diseño activas

c. Equipo de diseño (fig2.31).- Esta es una carpeta especial dentro de cada base de datos de diseño que permite configurar y mostrar información sobre las propiedades de compartición del diseño.



Fig2. 31 Equipo de diseño

d. Ventana de diseño (fig2.32).- Cada base de datos de diseño abierta dispone de una ventana de diseño asociada empleada para mostrar documentos de diseño abiertos y carpetas.

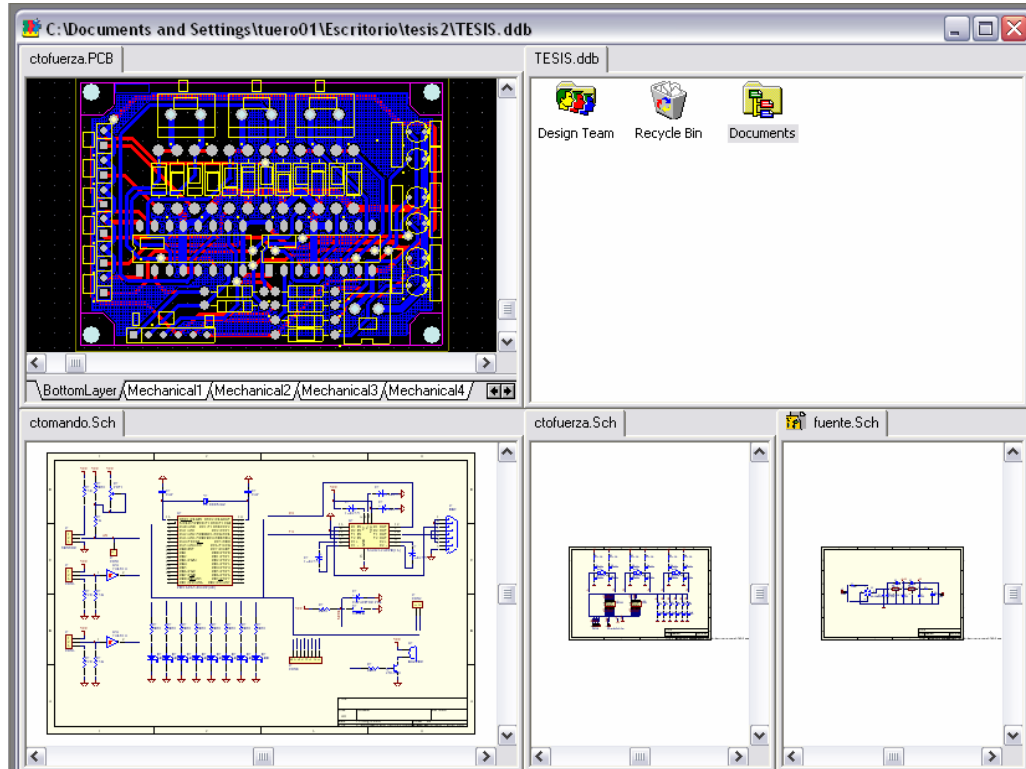


Fig2. 32 Ventana de diseño

e. Pestañas de diseño (fig2.33).- Los documentos abiertos y las carpetas aparecen como pestañas en la ventana de diseño.

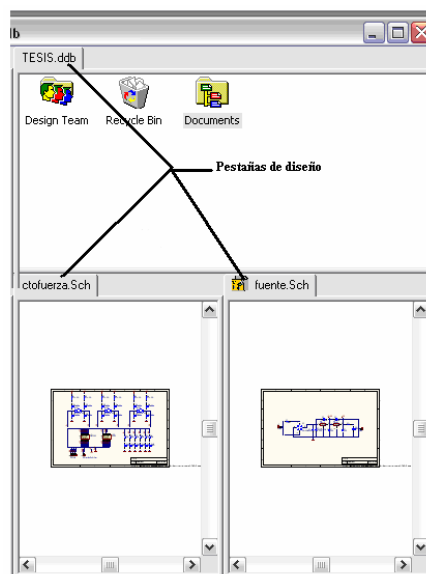


Fig2.33 Pestañas de diseño

Con un clic en una pestaña hace a ese documento o carpeta activa. Al hacer clic con el botón derecho sobre la pestaña aparece un menú de opciones, incluyendo la posibilidad de dividir la ventana para mostrar varias pestañas al mismo tiempo.

f. Menú del Explorador de diseño (fig2.34).- Dando clic en la flecha de la esquina izquierda de la barra de menús activa el menú del Explorador, que ofrece varias opciones del sistema.

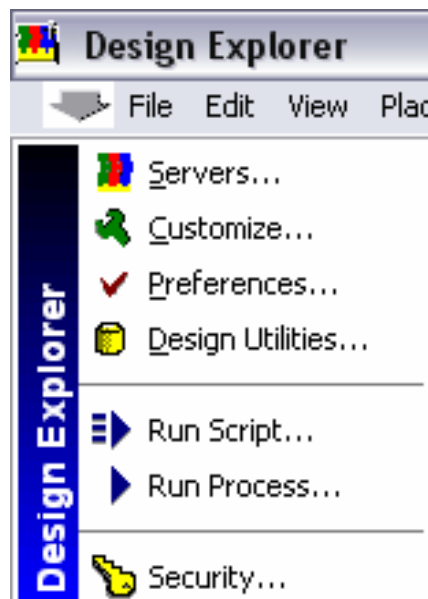


Fig2. 34 Menú del explorador de diseño

g. Asistente de ayuda (fig2.35).- Haciendo clic en este icono nos ofrece ayuda simplemente tecleando la pregunta. Existe un sistema de análisis de preguntas en lenguaje natural (inglés) que presenta una lista de temas de ayuda relacionados.

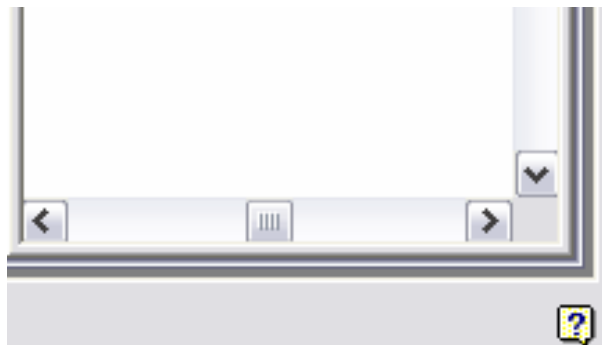


Fig2. 35 Asistente de ayuda

2.1.2.1 Combinaciones de teclas útiles.- Las siguientes combinaciones de teclas se pueden usar para navegar por el Explorador de diseño:

CTRL-TAB.- Circula a través de los documentos abiertos en la ventana de diseño.

CTRL-F4.- Cierra el documento activo en la ventana de diseño. Cuando se trabaja en un esquemático o en una PCB en Protel, se pueden emplear las siguientes teclas para cambiar la vista:

V, D.- Para ver el documento entero.

V, F.- Zoom para encajar todos los objetos del documento.

RePág.- Zoom para acercar centrado en la posición del cursor.

AvPág.- Zoom para alejar centrado en la posición del cursor.

Fin.- Redibuja la pantalla

Esc.- Termina la acción en proceso

Clic-sencillo.- Enfoca un objeto. Aparece el cursor de edición, permitiendo cambiar la forma del objeto.

Clic-mayúsculas.- Selecciona o deselecciona un objeto. Los objetos seleccionados pueden cortarse, copiarse, borrarse, etc. Clic-mayúsculas es acumulativo, se pueden seleccionar varios objetos, para deseleccionarlos todos pulsar X, A.

Doble-Clic.- Abre el dialogo de propiedades del objeto para su modificación.

Clic-botón derecho.- Termina la acción en proceso.

2.1.2.2 Los cursores de edición.- Cuando se activa cualquier proceso en Protel sobre un objeto o documento, el cursor cambia a una cruz para permitir un posicionado

preciso. En general, una vez que se ha entrado en modo edición permanece en este modo hasta que se haga un clic con el botón derecho o se pulse Escape para volver al cursor normal de Windows.

Esto permite repetir acciones como la colocación de múltiples objetos, sin tener que volver a lanzar el proceso. Además, con Protel se puede mientras se está realizando una determinada tarea realizar otra tarea, para luego volver a la primera, como por ejemplo durante una colocación de componentes se puede cambiar el zoom para luego seguir colocándolo. Un ejemplo de esto podemos observarlo en la fig2.36

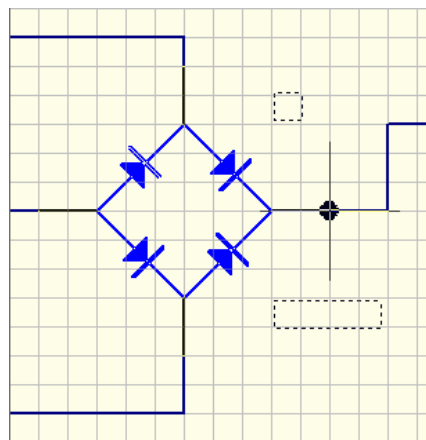


Fig2. 36 Cursor en forma de cruz para posicionado correcto

2.1.2.3 Obtención de ayuda.- El sistema de ayuda en lenguaje natural permite realizar preguntas en Inglés sencillo obteniendo una lista de temas de ayuda apropiados.

También se puede pulsar F1 en cualquier momento para obtener ayuda de una determinada opción o menú en el que se encuentre. Además se dispone de ayuda del tipo “Qué es esto”, que consiste en una pequeña marca de interrogación colocada en el borde superior derecho que al pulsar ofrece una explicación somera del control donde este situado.

2.1.3 Creación de un printed circuit board (pcb) con Protel.- Para crear una PCB con Protel, se debe seguir los siguientes pasos:

- Crear un esquemático

- Crear una PCB
- Comprobar el esquemático
- Colocar los componentes
- Rutado
- Impresión

2.1.3.1 Creación del esquemático.- Cuando se requiere realizar un circuito en PCB, existen dos caminos: o bien se crea primero un esquemático o se coloca directamente los componentes sobre la placa PCB. Este último camino es bastante dificultoso cuando las dimensiones de la placa empiezan a ser grandes, por lo que es más cómodo crear un esquemático.

Para ello, en el menú **File**, se selecciona **New**. Entonces se obtiene una lista de tipos de archivos que se pueden crear. Se pincha en **Schematic Document (Fig2.37)**, que genera un fichero de extensión *.sch. En este fichero, se dibuja el circuito que se quiere pasar a PCB.

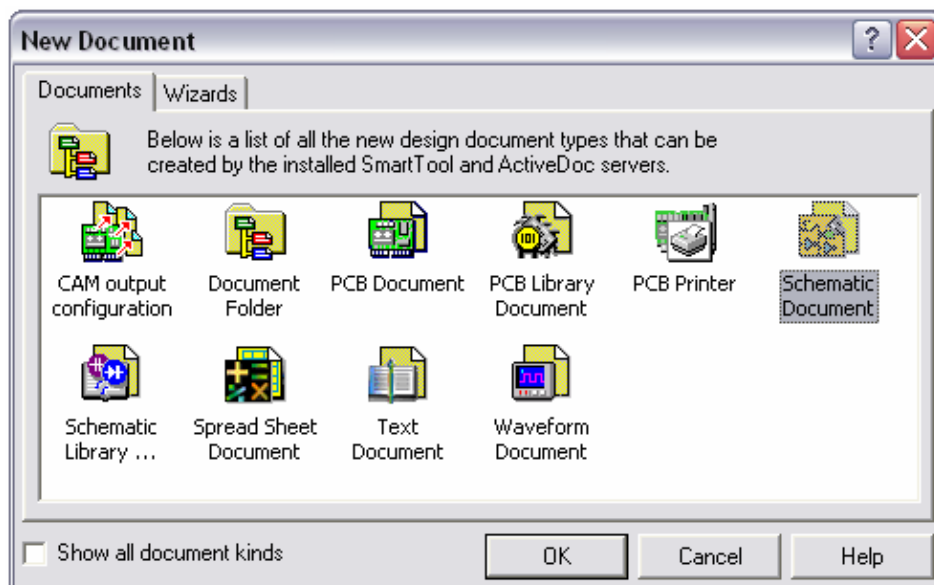


Fig2. 37 Creación de un esquemático

Lo primero, antes de nada, es cargar las librerías que se vaya a utilizar.

Cuando se abre un esquemático, se obtiene dos partes en la pantalla: una a la derecha donde se encuentra el circuito, y otra a la izquierda, donde se tiene dos pestañas:

Explorer y Browse Sch. En esta última pestaña, se selecciona Libraries en Browse y se pulsa en Add/Remove para poner o quitar librerías.

Si no se encuentra algún componente requerido, se utiliza el comando Find que está abajo a la izquierda. Con esta función, Protel busca todos los componentes con ese nombre en cualquier librería.

Hay ciertos componentes, como las resistencias o los condensadores, que no se encuentran en ninguna librería específica, por lo que se los puede obtener de la librería Miscellaneous, que es la librería que se carga por defecto.

Una vez que se tiene cargadas las librerías que se vayan a utilizar, se pasa a la creación del esquemático en sí. En la parte izquierda de la pantalla, aparece una lista de los componentes que contiene la librería que se elija.

Para crear el circuito, sólo es necesario seleccionar uno de los componentes y colocarlo sobre la cuadrícula del esquemático. En cuanto se haya colocado el componente deseado, hay que pinchar dos veces sobre él. Entonces, aparecerá un cuadro llamado Part como muestra la fig2.38 donde hay cuatro pestañas:

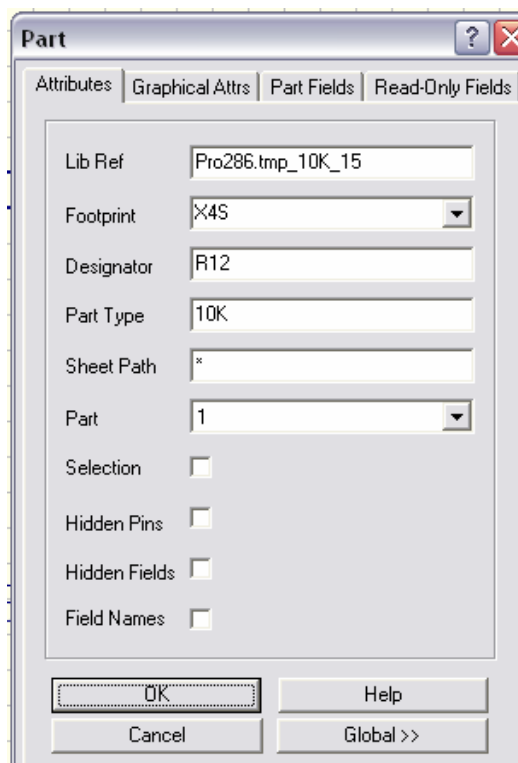


Fig2. 38 Configuración de elemento

- Attributes
- Graphical Attrs
- Part Fields
- Read-Only Fields

Sólo se utiliza la primera pestaña que es la que aparece por defecto, modificando las siguientes opciones:

- **Lib Ref:** se coloca qué tipo de componente se ha colocado en el circuito. Esta opción no se debe cambiar.
- **Footprint:** se define el número de pines, su grosor, la distancia entre pines, etc., que tiene un componente, para que luego Protel pueda crear correctamente la PCB.

Por defecto, pueden ocurrir dos cosas: o que no reconozca el componente por lo que esta casilla esté vacía o que lo reconozca (sobre todo cuando son pastillas) y ponga **DIP-x**. Cuando viene el footprint, hay que quitarle el guión (**DIPx**), caso contrario, no reconoce este footprint.

También existen otras funciones (**DIPxBOTSOL**) que, a diferencia de las establecidas por Protel, permiten soldar sólo por la cara de abajo.

- **Designator:** es el nombre que el diseñador da a la pastilla. Es importante que puertas de una misma pastilla tengan el mismo Designator. Si no, el programa lo considerará como dos pastillas diferentes.
- **Part:** aquí se indica qué elemento (o qué puerta, si es una pastilla de puertas lógicas) de una pastilla se utiliza. De esta forma, se puede utilizar los diferentes elementos que tenga una pastilla sin confundirlos.

Puede ocurrir, que se necesite conectar dos puntos que se encuentren alejados físicamente en el "dibujo". En este caso, Protel permite, mediante la función **NetList** (fig2.39), poner "banderas" en puntos determinados que indican que dichos puntos están eléctricamente conectados, es decir, que tendrá que haber una pista que los conecte cuando se haga el rutado de la PCB. Estos **Netlist** están en el menú **Wiring Tools**.



Fig2. 39 Netlist

Otro punto interesante que se debe poner, son los **Power Object**. Estos objetos son los que conectan las alimentaciones y masas a los conectores banana. Estos objetos son Vcc y Gnd que se sacan del menú **Power Object** (fig2.40).



Fig2. 40 Power Objets

2.1.3.2 Creación de una PCB.- Antes de comprobar si el esquemático realizado es correcto, se debe crear la placa donde se van a insertar los componentes. Para este caso, se utiliza el "wizard" de PCB. Para ello se ejecuta Printed Circuit Board Wizard que se encuentra en la pestaña Wizards como se observa en la figura 2.41.

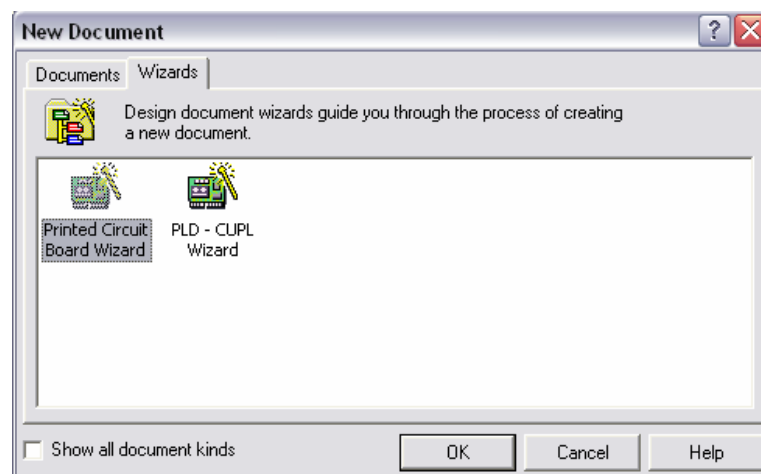


Fig2.41 PCB Wizard

Se Utiliza Custom Made Board cuyas unidades son métricas como muestra la fig2.42.

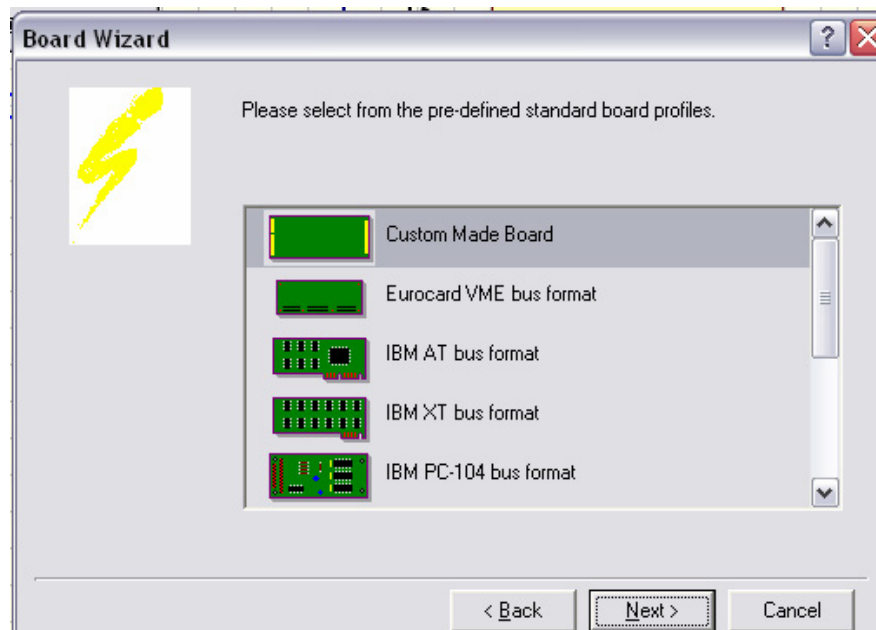


Fig2. 42 Custom Made Board

En la segunda pantalla (fig2.43), aparecen una serie de opciones para definir las diferentes longitudes: las dimensiones del rectángulo (también se puede definir una placa circular u ovalada), el grosor de la pista, etc. Además en la parte baja de la pantalla existen una serie de opciones que no se utilizan, por lo que se desactivan.

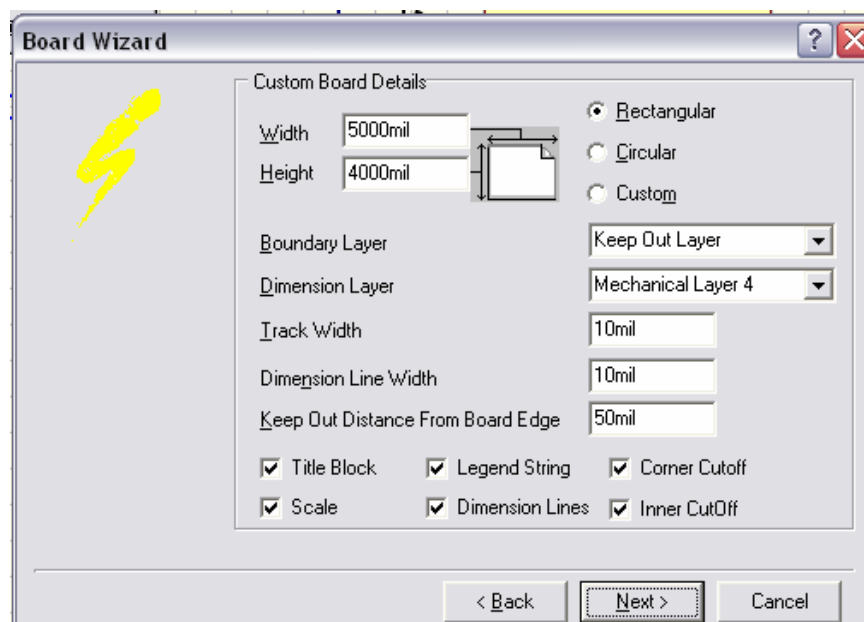


Fig2. 43 Definición de longitudes

En la tercera pantalla (fig.2.44), aparece un dibujo de cómo será la forma de la placa y de cuál será su tamaño.

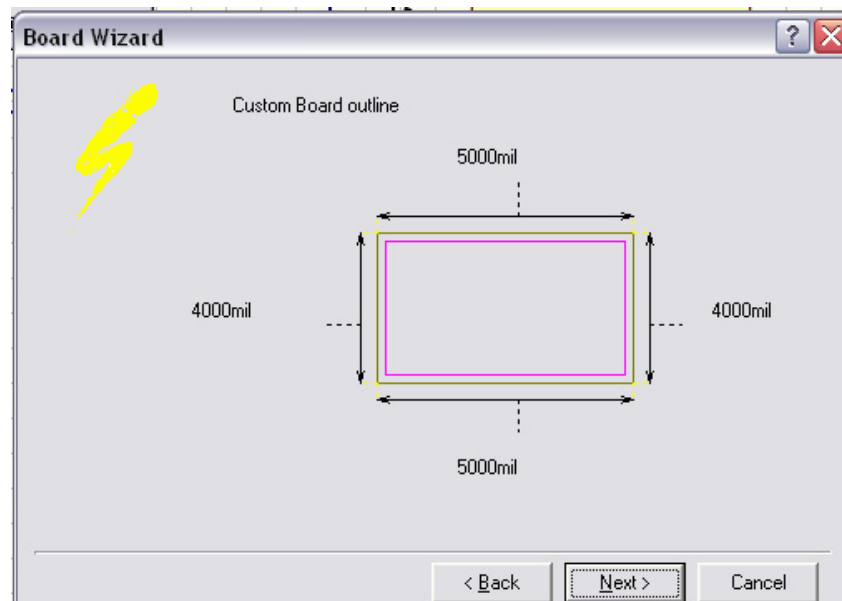


Fig2.44 Forma y tamaño del PCB

Las dos siguientes pantallas, se dejan como están. Sin embargo, en la séptima pantalla se pregunta el tipo de soldaduras que se van a hacer: superficiales o insertadas. También pregunta, cuántas vías se quiere que haya entre pines (fig2.45).

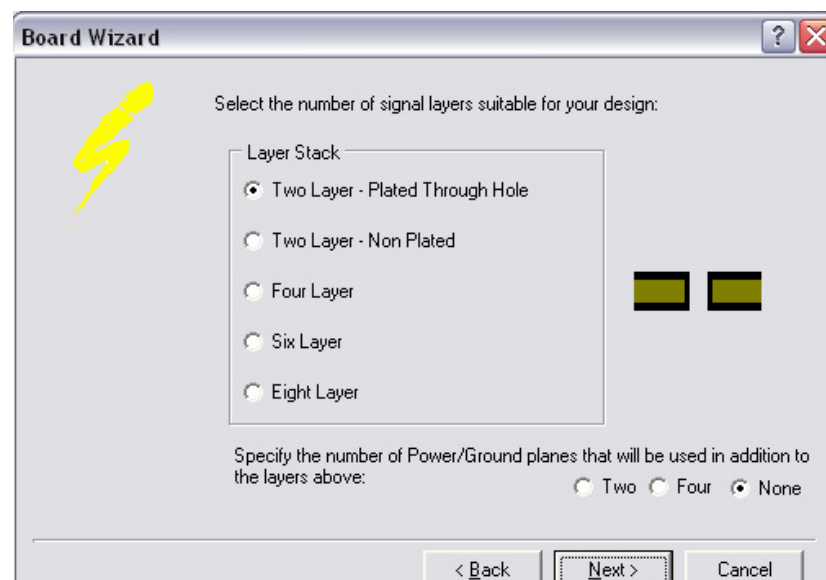


Fig2. 45 Opciones de soldadura

En la siguiente pantalla, pregunta las dimensiones de las pistas, de los agujeros de los pines, y la distancia entre pistas (fig2.46). En un principio, se deja como está. Lo más

probable es que luego, una vez se coloque los componentes y rutado, se deba cambiar estas longitudes.

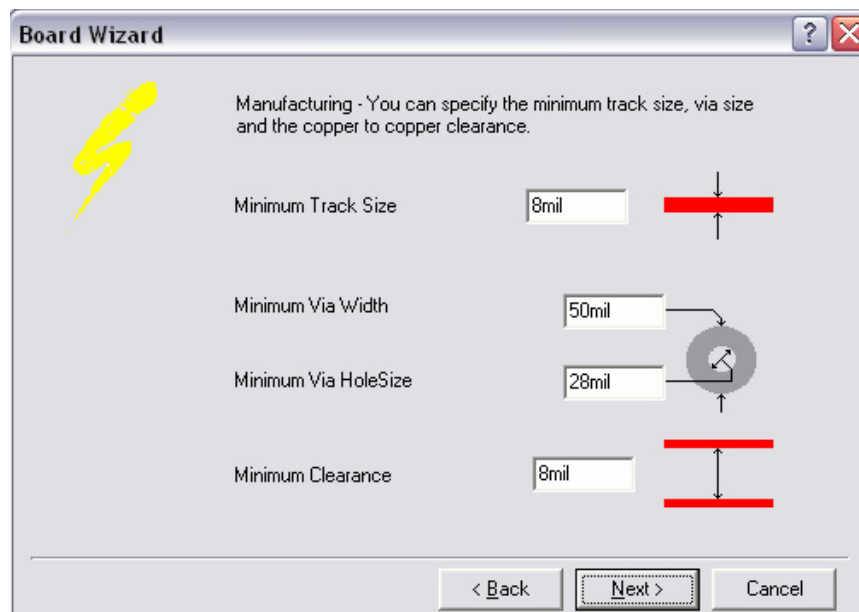


Fig2. 46 Dimensiones de pistas y agujeros

Ahora se crea y se abre un nuevo fichero cuyo nombre es PCB1.PCB. Este fichero está vacío, pero ya contiene la placa donde se van a colocar los componentes.

2.1.3.3 Comprobación del esquemático.- Una vez realizado todo el esquemático y creado la placa donde se van a insertar los componentes, se comprueba que está bien realizado el circuito para que Protel sea capaz de construir correctamente la PCB.

Esta comprobación se realiza en el menú Design, y aquí en Update PCB. Sin tocar nada, se da a Preview Changes. Entonces sale una lista con todos los movimientos que se han ido haciendo a lo largo de la construcción del esquemático. Si no hay ningún error, al final de la lista se escribe "All macros validated".

Sin embargo, si se ha producido algún error, se escribe "X errors found" y, al lado del movimiento donde se ha producido el error, se escribe qué tipo de error ha sido. Si todo es correcto, damos a Execute y se pasa entonces al siguiente punto.

2.1.3.4 Colocación de componentes.- Cuando se haya comprobado y ejecutado el Update PCB, aparecerá en el fichero.pcb que se ha creado con el "wizard" todos los

componentes del esquemático, pero ordenados en línea al lado de la placa que se haya definido.

Una vez hecho lo anterior se pueden realizar varias acciones. Una de ellas, es decir a Protel que coloque los componentes sobre la PCB según el tamaño de los componentes. Para ello, se usa en el menú Tools la opción Auto Placement. Aquí se tiene un nuevo menú, donde se selecciona Auto Placer. De nuevo aquí, se tiene dos opciones, así que seleccionamos Cluster Placer y ya se tiene una colocación posible.

Otra opción es la de ordenar a Protel que coloque los componentes según se utilicen, es decir, que ordene los componentes según sus conexiones, de tal forma que las pistas sean lo más pequeñas posibles. Para ello, se sigue la secuencia explicada anteriormente, y en vez de seleccionar Cluster Placer, se opta por Statistical Placer.

El problema que tiene cualquiera de las dos opciones anteriores es que Protel junta demasiado los componentes y, por tanto, también juntará demasiado las pistas. Por esto, es recomendable colocar "a mano" todos los componentes. De esta forma, se pueden reagrupar los componentes en conjuntos que desempeñen cada uno de ellos una función específica.

2.1.3.5 Rutado.- Una vez colocados todos los componentes, se hace de forma automática. Para ello, en el menú Auto Route, se utiliza la opción All. Sin cambiar nada de lo que pone, se ejecuta Route All.

Puede ocurrir que se deba enrutar más de una vez debido a problemas con el grosor de las pistas, la separación entre ambas, etc. Por lo general, la primera vez que se enruta, las pistas son demasiado finas, y se juntan demasiado unas a otras. Se debe, por tanto, cambiar estas longitudes. Para ello, en el menú Design, se selecciona Rules. En esta opción, se tiene todas las reglas que utiliza Protel para enrutar. Por ejemplo:

- En la regla Clearance Constraint (pestaña Routing), se cambia la distancia que debía haber entre las pistas y el agujero de un pin.
- En la regla Width Constraint (pestaña Routing), se indica el grosor mínimo, el máximo y el preferido de las pistas.

- En la regla Paralel Segment Constraint (pestaña High Speed), se cambia la separación entre pistas.

Con estas nuevas reglas, se vuelve a enrutar. Si la placa que se obtiene no es adecuada, se cambian de nuevo las propiedades. Y así hasta que se obtenga la placa deseada.

2.2 Diseño del circuito de mando

Una vez que se ha realizado una introducción al programa PROTEL 99SE, se procede a formalizar los circuitos que se utilizarán para el manejo del Rugosímetro Digital propuesto en esta tesis.

Al circuito de mando se ha dividido en secciones específicas para poder explicar de mejor manera el funcionamiento del mismo.

2.2.1 Sensor óptico (OPTOACOPLADOR).- Esta sección del circuito se encarga de acoplar el sensor óptico con el palpador que se utiliza en el rugosímetro, para que de esta manera los datos de rugosidad obtenidos puedan ser enviados al microcontrolador.

En este caso se usa el sensor H21A (fig2.47) que es un dispositivo que contiene un emisor de rayos infrarrojos (diodo LED) y un receptor de los mismos (fototransistor). Ambos elementos están situados a la misma altura y a una distancia de 3 milímetros enfrentados. Su finalidad es detectar objetos que se interponen entre el emisor y el receptor que cortan el haz de luz.

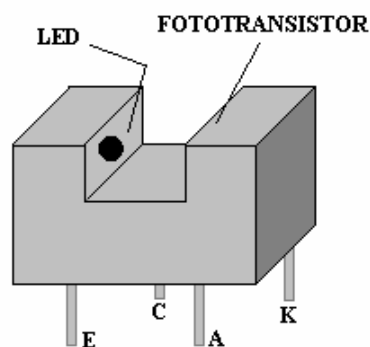


Fig2. 47 Dispositivo H21A

Internamente el dispositivo H21A viene dado como muestra la figura 2.48

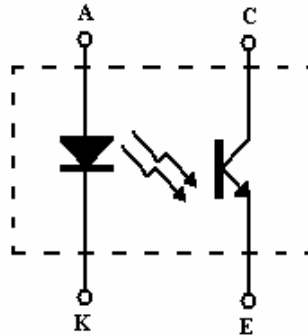


Fig2. 48 Diagrama interno del H21A

En el circuito de la placa el dispositivo mencionado con anterioridad se coloca al conector J2 como se especifica en la figura 2.49

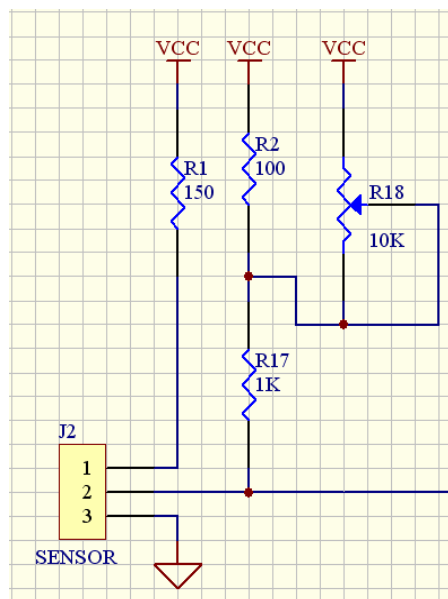


Fig2.49 Conexión del sensor óptico al circuito de la placa

En el conector J2 se conectarán los pines de la siguiente manera:

Pin 1.- Al emisor del dispositivo y este a una resistencia de 150Ω requerido.

Pin 2.- Al colector del dispositivo y este a un partidor con una resistencia de $1K\Omega$ y 100Ω . Este a su vez cuenta con un potenciómetro de $1K\Omega$ que sirve para calibrar el nivel de 0v a 5v para el rango de la escala de conversión.

Pin 3.- En este pin van conectados tanto el cátodo como el emisor y este a su vez va a tierra.

2.2.2 Encoders (Opcional).- Esta parte del circuito se enfoca a la medición de las distancias que recorren los ejes X y Y del Rugosímetro por medio del conteo de pulsos. Para realizar esto se utiliza el dispositivo CNY70 (fig2.50) que se trata de un sensor que contiene un emisor (diodo LED) y un receptor de rayos infrarrojos (fototransmisor) en la cápsula.

Es un dispositivo de reflexión ideal para detectar cambios de tono sobre una superficie a la que debe situarse a pocos milímetros de distancia. La luz que emite por el diodo LED es reflejada por la superficie y detectada por el fototransistor. Funciona a frecuencias elevadas y aísla el circuito eléctrico y del microprocesador de control.

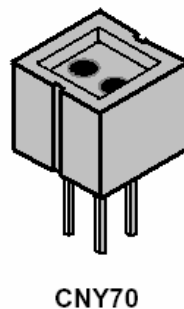


Fig2. 50 Dispositivo CNY70

Internamente es muy similar al dispositivo antes mencionado, como se puede observar en la figura 2.51

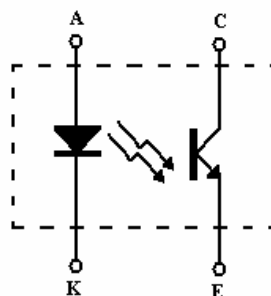


Fig2.51 Diagrama interno del CNY70

Para el correcto funcionamiento de este dispositivo, se maneja discos con franjas negras y blancas para que el CNY70 pueda ir detectando y contando los cambios de tonalidad en el disco y por ende el movimiento que realiza los ejes como muestra la figura 2.52

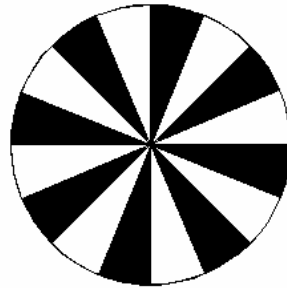


Fig2.52 Disco del encoder

En el circuito de la placa los encoders se colocan en los conectores J3 y J5 como se especifica en la figura 2.53

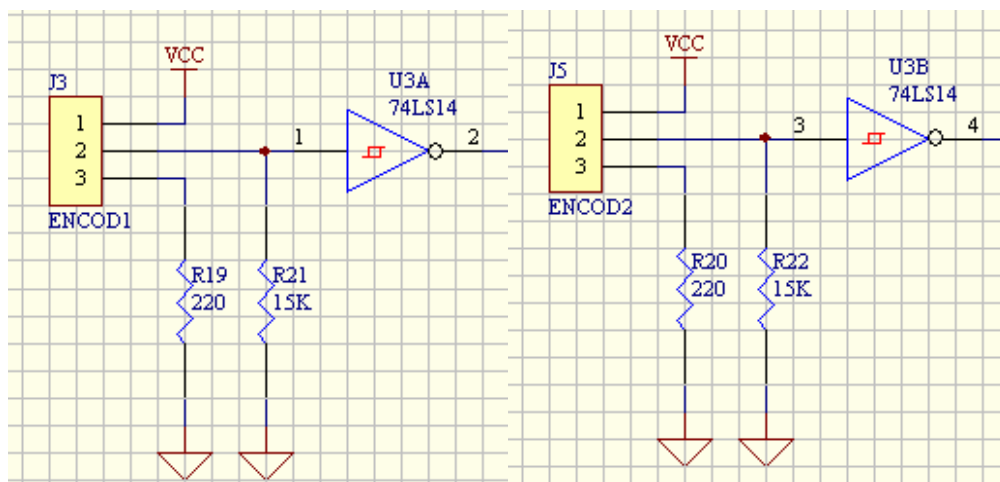


Fig2. 53 Conexión de los encoders al circuito de la placa

Los conectores J3 y J5 serán colocados de la siguiente manera:

Pin 1.- el ánodo y el colector del CNY70 y este a su vez a VCC.

Pin 2.- el cátodo del dispositivo de este a su vez a una resistencia de 220Ω requerida.

Pin 3.- al emisor del dispositivo y este a su vez a una resistencia de $15K\Omega$ y una puerta inversora de tipo trigger Smith.

La puerta inversora tipo Trigger Smith se encarga de reconstruir la señal de entrada para poder enviarla sin errores a los puertos RB0 y RB1 del microcontrolador. En la figura 2.54 se puede observar el funcionamiento de la misma.

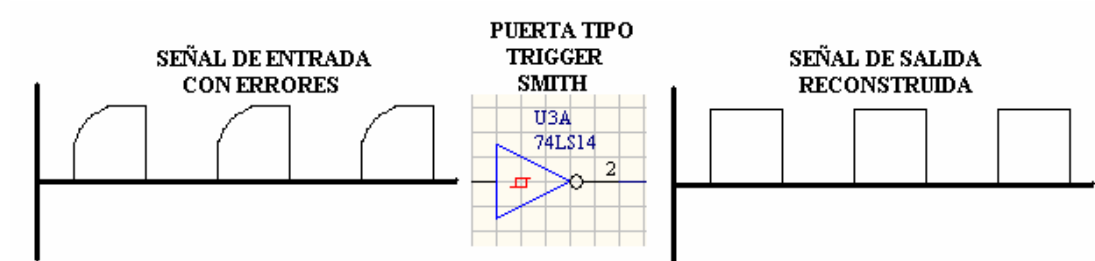


Fig2. 54 Funcionamiento de la puerta inversora tipo Trigger Smith

2.2.3 Reset general.- Esta parte del circuito se encarga de dar un reset al microcontrolador en caso de alguna falla o cuando requiera que el sistema deje de hacer el proceso que esta realizando.

Tiene la particularidad de contar con un conector J10 en el cual se puede colocar un pulsante fuera de la placa para comodidad del usuario al momento de resetear el sistema.

La conexión queda establecida como muestra la figura 2.55, el pulsante S1 manda una señal al MCLR del microcontrolador que es activo a cero y de esta manera el mismo es reseteado.

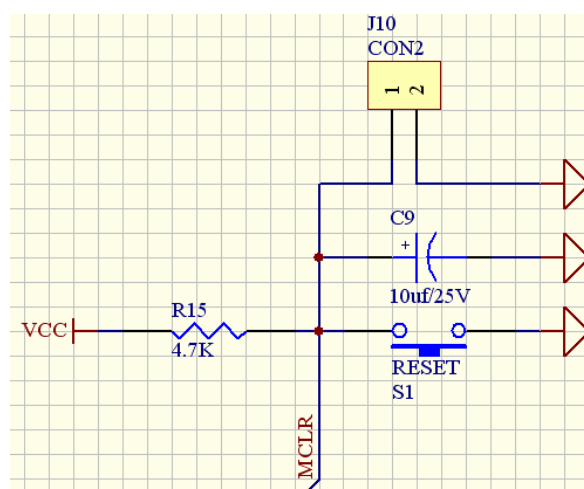


Fig2. 55 Conexión del reset del microcontrolador

2.3 Comunicación serial

2.3.1 Comunicaciones serie asíncronas.- Los datos serie se encuentran encapsulados en tramas de la forma como indica la figura 2.56

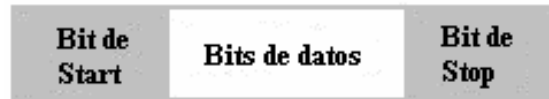


Fig2.56 Datos serie

Primero se envía un bit de start, a continuación los bits de datos (primero el bit de mayor peso) y finalmente los bits de STOP.

El número de bits de datos y de bits de Stop es uno de los parámetros configurables, así como el criterio de paridad par o impar para la detección de errores.

En la figura 2.57 se puede ver un ejemplo de la transmisión del dato binario 10011010. La línea en reposo está a nivel alto:

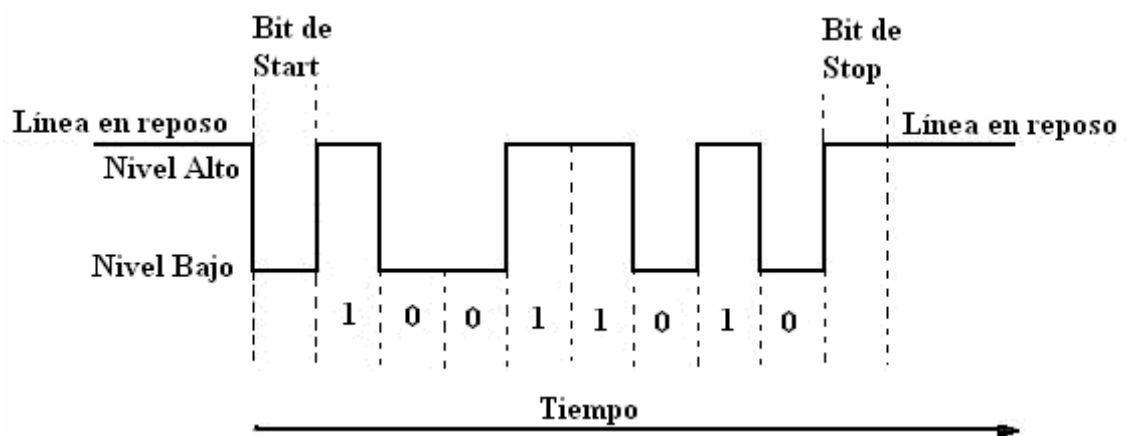


Fig2. 57 Transmisión de dato binario 10011010

2.3.1.1 Norma RS232.- La Norma RS-232 fue definida para conectar un ordenador a un modem. Además de transmitirse los datos de una forma serie asíncrona son necesarias una serie de señales adicionales, que se definen en la norma (fig2.58). Las tensiones empleadas están comprendidas entre +15/-15 voltios.

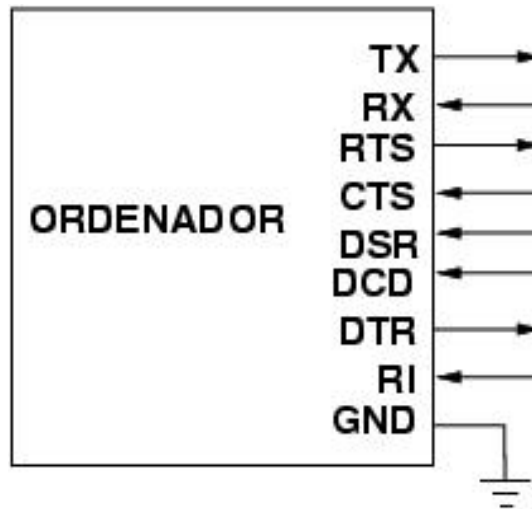


Fig2. 58 Norma RS232

2.3.1.2 Conexión del microcontrolador al puerto serie de la PC.- Para conectar la PC a un microcontrolador por el puerto serie se utilizan las señales Tx, Rx y GND. El PC utiliza la norma RS232, por lo que los niveles de tensión de los pines están comprendidos entre +15 y -15 voltios.

Los microcontroladores normalmente trabajan con niveles TTL (0-5v). Es necesario por tanto intercalar un circuito que adapte los niveles como muestra la figura 2.59

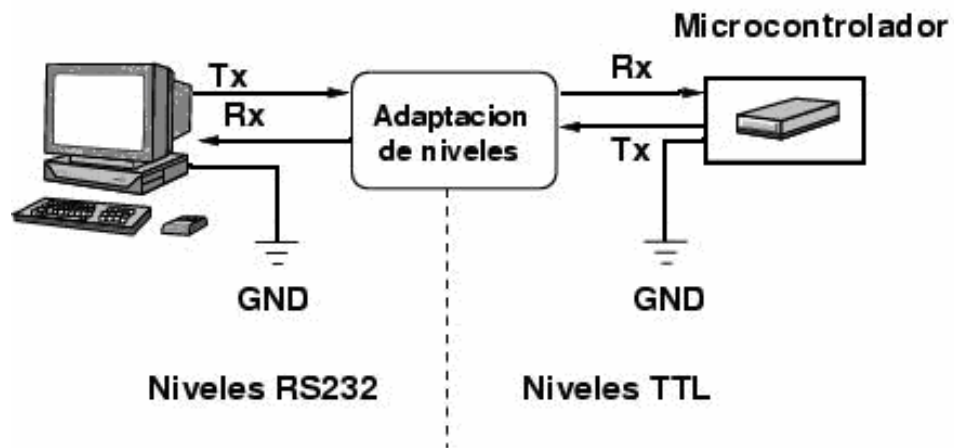


Fig2. 59 Conexión del microcontrolador con la PC

2.3.1.3 El conector DB9 del PC.- En la PC 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... (fig2.60).

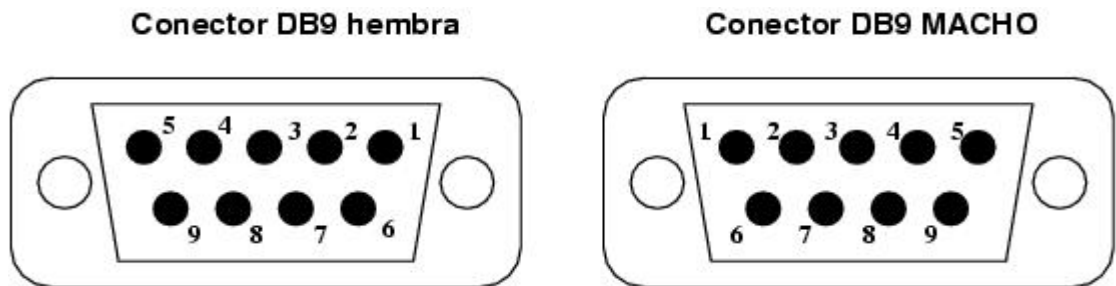


Fig2.60 Conectores DB9 macho y hembra

La información asociada a cada uno de los pines esta indicada en la tabla 2.1:

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)

Tabla2.1 Información de pines del DB9

2.3.1.4 El chip MAX 232.- Debido a los voltajes que se hacen necesarios se ha utilizado como interfaz el MAX232, que permite asegurar la correcta alimentación de la señal y activar de modo adecuado la comunicación vía serie.

- 12V equivalen a un ``0" lógico.
- -12V equivalen a un ``1" lógico.

Como los niveles lógicos que salen del micro no son compatibles con los lógicos del puerto, se necesita usar el dispositivo antes mencionado para que adecue estos valores.

La conexión es sencilla, sólo es necesario comunicarlo con dos pines del puerto serie además de la tierra. Debido a ello, se eligió por su funcionalidad y sencillez en la comunicación.

La entrada a este chip, proviene de las pines 25 y 26 del PIC, que son respectivamente TX y RX. Éstas, van conectadas directamente al MAX-232 a partir de las pines 11 y 12, que son de entrada. Y los pines de salida serán las que se conectarán con el puerto serie.

Posteriormente, se ha añadido un conector de puerto serie macho en la placa, que se conectará a través de un cable serie hembra con el puerto serie del PC.

El MAX 232 presenta las siguientes ventajas (fig2.61)

- Voltaje de funcionamiento 5v
- Velocidad de 120 kbit/s
- 2 transmisores y 2 receptores
- Entradas de nivel de $\pm 30v$
- Baja corriente de trabajo 8Ma
- Distancia de trabajo máx. 15m

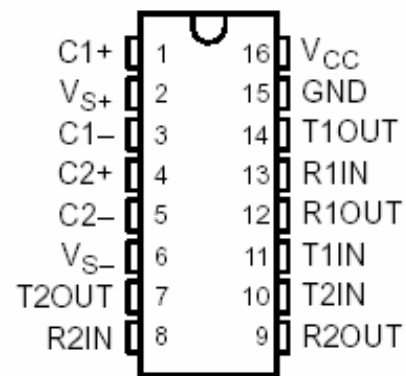


Fig2.61 MAX232

Lleva internamente 2 convertidores de nivel de TTL a rs232 y otros 2 de rs232 a TTL con lo que en total se puede manejar 4 señales del puerto serie del PC, por lo general las mas usadas son; TX, RX. Para que el max232 funcione correctamente se debe colocar unos condensadores externos de 1 microfaradio en los pines especificados en la figura 2.62

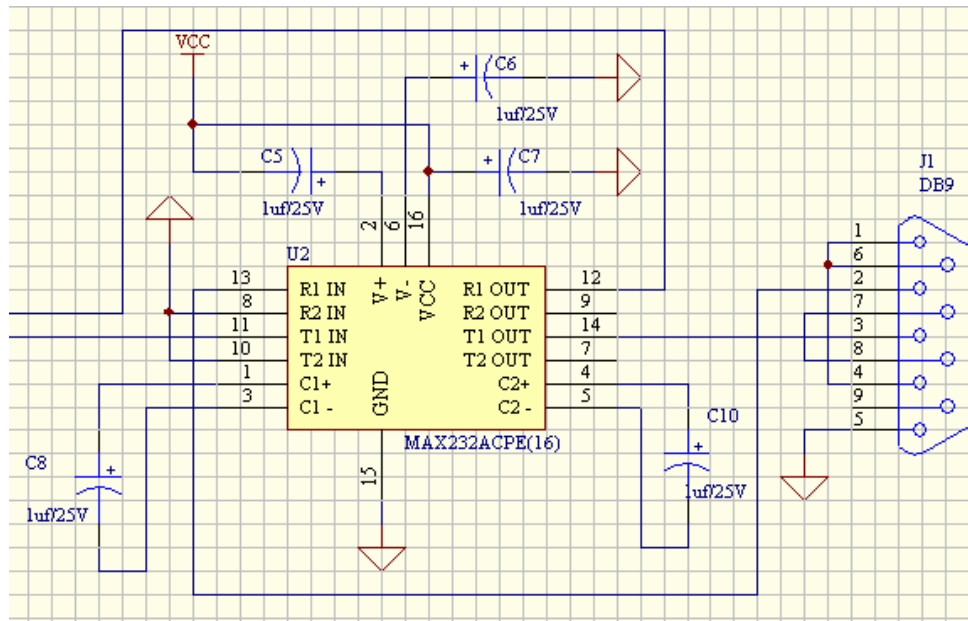


Fig2. 62 Conexión del MAX232 y del DB9

Los pines más importantes del MAX232 van conectados de la siguiente manera:

Pin 11.- Se conectará al pin 25 del microcontrolador equivalente a TX

Pin 12.- Se conectará el pin 26 del microcontrolador equivalente a RX

Pin 13.- Se conectará el pin 2 del DB9 equivalente a RX

Pin 14.- Se conectará el pin 3 del DB9 equivalente a TX.

La conexión de los condensadores de 1 microfaradio viene establecida para que funcione de manera correcta.

2.4 Selector manual / normal

En esta sección del circuito, se tiene la posibilidad de seleccionar el modo de funcionamiento de la placa, se tiene 2 posibilidades:

2.4.1 Modo Normal.- en este modo la placa recibirá los datos por el canal analógico RA0 del microcontrolador y los motores serán manejados por software de manera automática como ya se verá en el desarrollo de los siguientes capítulos.

2.4.2 Modo Manual.- en esta opción se puede controlar los motores a libertad del usuario, tanto en su dirección, como en la velocidad ya que el PWM controlado con la variación de la entrada analógica RA1.

La conexión de esta opción la especificamos en la figura 2.63 a continuación:

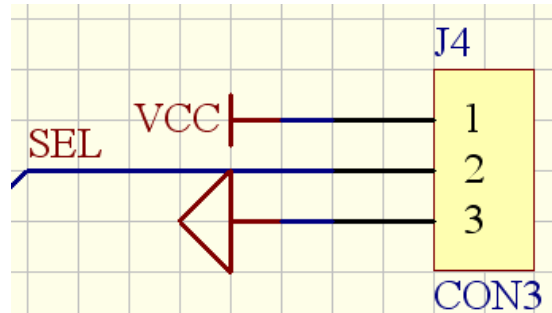


Fig2. 63 Selector Manual / Normal

En el conector J4 se colocará un Jumper, que dará un nivel lógico 0 o 1 a la entrada del pin RE0 y este a su vez indicará al microcontrolador en que modo se encuentra, de manera que pueda funcionar solo uno a la vez.

2.5 Conector para joystick

Cuando se ha elegido el modo manual ya antes mencionado, se necesita un joystick con el cual se puede controlar los motores y la velocidad por medio de un potenciómetro. La figura 2.64 muestra un ejemplo de dicho joystick:

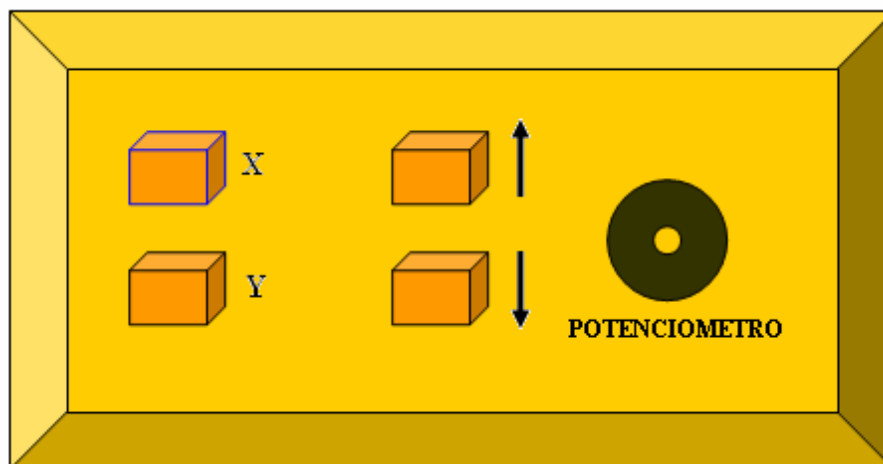


Fig2.64 Ejemplo de joystick para mando manual del circuito

En la figura 2.65 se presenta la conexión del joystick en el circuito:

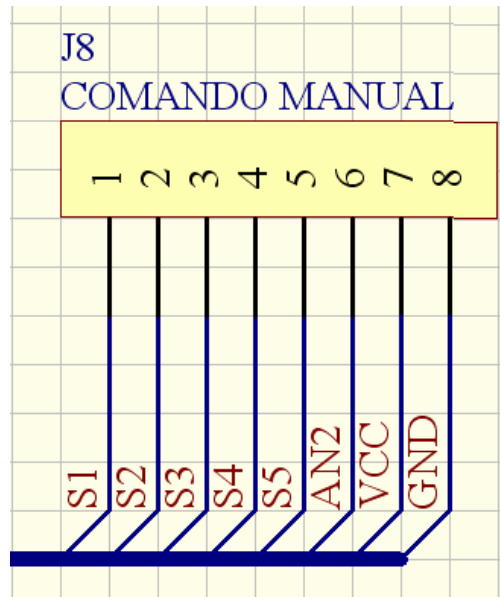


Fig2. 65 Conexión del joystick en el circuito de mando

El conector J8 presenta los siguientes pines de conexión:

Pines 1-3.- Conectores para indicar el motor que se quiere mover

Pines 4-5.- Conectores para indicar la dirección a la que se desea que se mueva

Pin 6.- En este pin se conecta el potenciómetro del joystick que le indica al RA1 a que velocidad debe colocar el PWM para la velocidad del motor

Pin 7.- Alimentación del joystick

Pin 8.- Tierra del joystick.

Cabe recalcar que para que el motor seleccionado se mueva a la dirección que se quiera, se debe mantener presionado el botón de dicho motor, caso contrario si solo se pulsa el de la dirección, el programa del microcontrolador no entenderá la orden y por ende no realizará ninguna acción.

2.6 Condensadores by-pass

Estos condensadores se encargan de eliminar los ruidos por acoplamiento inductivo (cables) y capacitivo (60hz) existentes en el medio, de manera que el sistema este libre de ruido para que no exista error en la adquisición de datos y las medidas obtenidas tengan la mayor precisión posible. La figura 2.66 indica la conexión de dichos condensadores en el circuito de mando.

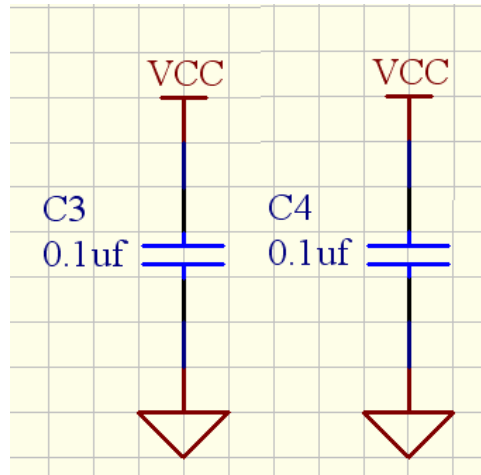


Fig2. 66 Condensadores By-Pass

2.7 Leds de señalización

El circuito de mando se tienen LED's de señalización que permiten visualizar que motor esta activo así también como la dirección que tiene el mismo, de la misma manera, esta señalización esta colocada en los encoders y cuando uno de estos esta en funcionamiento se puede visualizar. Los LED's están conectados como indica la figura 2.67

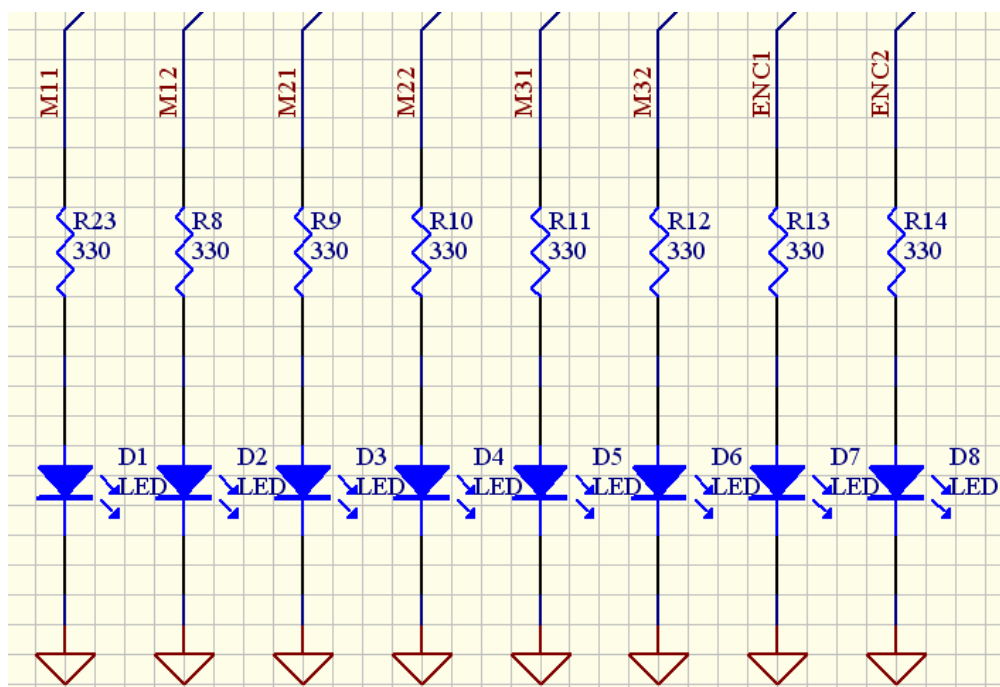


Fig2. 67 Conexión de los LED's de señalización

Cada LED cuenta con una resistencia limitadora de 330Ω.

2.8 Conexión del microcontrolador

Una vez que se ha visto las partes del circuito de mando, como último punto colocamos la conexión del microcontrolador, en él estarán conectados todos los componentes detallados anteriormente para que puedan funcionar. La conexión del microcontrolador muestra la figura 2.68 a continuación:

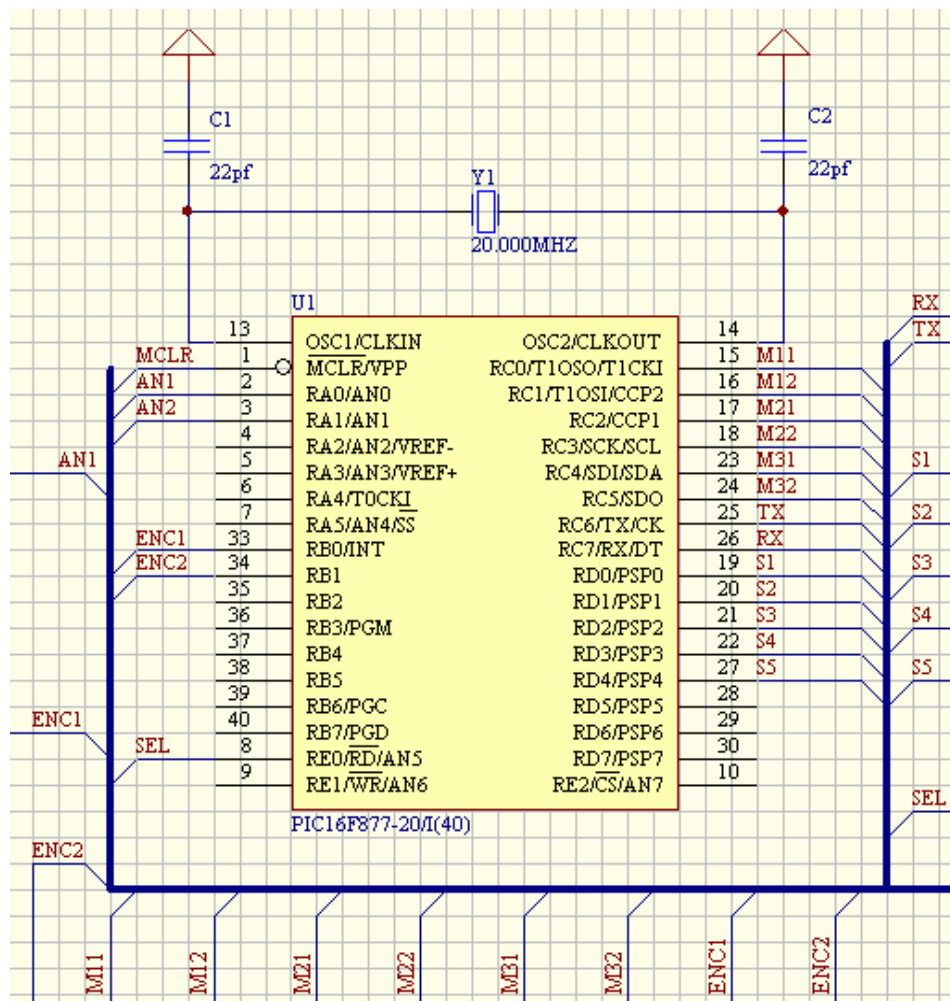


Fig2.68 Conexión del Microcontrolador

En esta figura se puede visualizar que todas las partes antes mencionadas, se conectan a los pines respectivos del microcontrolador mediante un bus de datos (Línea azul gruesa) para facilitar la conexión sin muchos cables.

Es imprescindible la conexión de un oscilador en un microcontrolador, en este caso se coloca un oscilador de 20Mhz con el fin de tener una buena comunicación con la PC.

2.9 Diseño del circuito de fuerza

Para un mejor entendimiento del funcionamiento del circuito de fuerza, se ha dividido al circuito total en varias partes, de igual forma que se hizo con el circuito de mando.

2.9.1 Conector de Datos.- Esta sección del circuito se encarga de comunicar las tarjetas de mando y fuerza, la conexión queda establecida como indica la figura 2.69

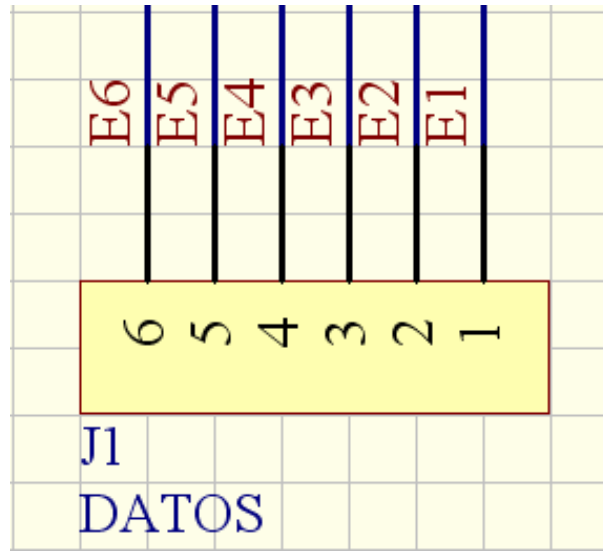


fig2. 69 Conector del bus de datos

En este conector J1 estarán conectados los datos de cada motor, es decir, la información necesaria para la dirección y velocidad que se necesite.

2.9.2 Leds de señalización.- El circuito de fuerza se cuenta también con LED's de señalización que permite visualizar que motor esta activo así también como la dirección que tiene el mismo.

La razón por la cual tanto el circuito de fuerza como el de mando tiene LED's de señalización es debido a que se puede comprobar el funcionamiento de los mismos en forma separada, ya que en caso de tener algún problema lo se podrá identificar de una manera mas sencilla

Los LED's están conectados como indica la figura 2.70

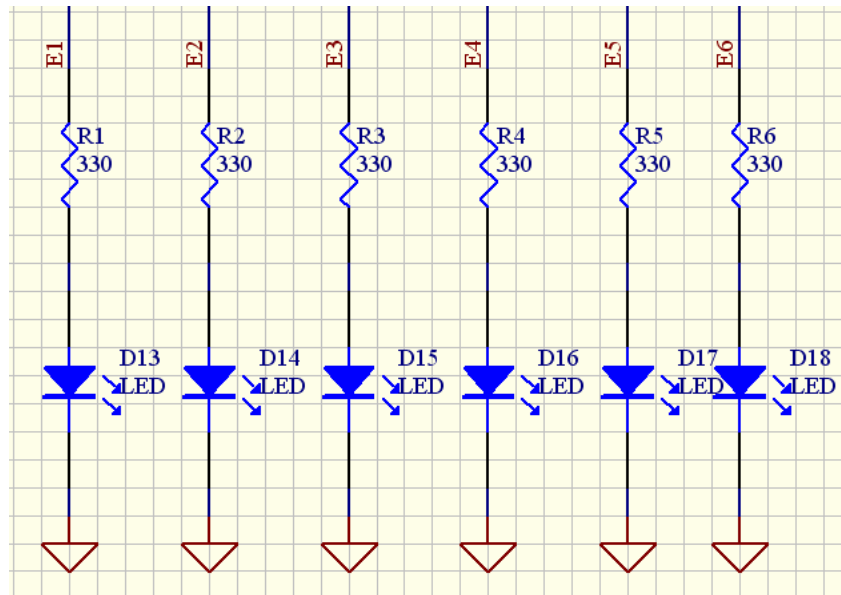


fig2. 70 Conexión de los LED's de señalización en el circuito de fuerza

2.9.3 Fin Carrera.- Para proteger a los motores del rugosímetro, se ha visto conveniente colocar un fin carrera a cada extremo del recorrido de los motores. El funcionamiento es sencillo, en caso de que un motor pase el límite de movimiento, se activa el fin carrera y quita la línea de alimentación (en este caso 12v) y el motor se apaga.

En la figura 2.71 se observa como quedan colocados estos dispositivos a cada extremo de los motores.

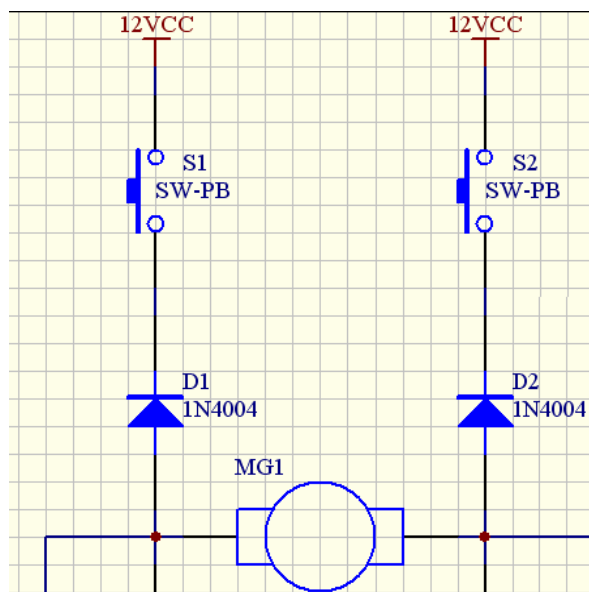


fig2. 71 Colocación de fin carrera a las líneas de un motor

2.10 Comando de los motores

2.10.1 Motores de Corriente Continua.- Los motores de corriente continua son muy utilizados en minirobots. Se puede encontrar motores de este tipo en prácticamente todos los coches teledirigidos y/o coches de juguete, los ventiladores de mano y accionamientos varios. Las partes más importantes del control de un motor son:

- Cómo controlar su velocidad.
- Controlar el sentido de giro.
- Alimentación.

Aparentemente, si se conecta un motor de continua a una batería, siempre debería moverse a la misma velocidad, bajo tensión constante. Si no se cambia la polaridad, parece imposible cambiarlos de sentido y en tercer lugar, si se quiere alimentar los motores a nuestro antojo, ¿cómo hacerlo si el PIC envía señales de 0 a 5V?

2.10.2 Velocidad.- Para poder controlar este aspecto se enviará al motor una señal desde el PIC de la forma PWM o tren de pulsos, que será del tipo como muestra la figura 2.72:

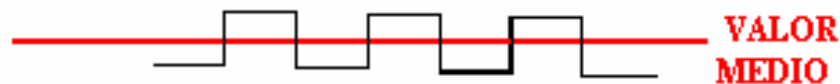


fig2. 72 Forma de onda para PWM

Esta gráfica es una forma de onda en tensión. Para entenderlo mejor, realmente se está alimentando el motor y apagándolo en períodos de tiempo del orden de milisegundos.

Con motivo de la inercia del propio motor estos cambios provocan que se mueva a una velocidad constante. Así pues, cuanto mayor sea el tiempo del pulso, mayor velocidad obtendremos.

En la siguiente figura (fig2.73) se observa como el tren de pulsos al ser más extenso, la velocidad del motor también es mayor:

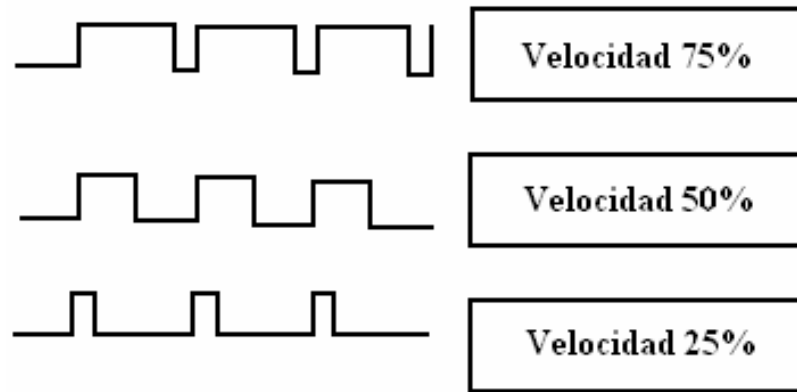


fig2. 73 Diferentes velocidades obtenidas con distintos PWM

2.10.3 Sentido de giro y alimentación.- La única manera de controlar este aspecto del motor, es rodeándolo de un circuito, de tal forma que se pueda, al enviar señales, cambiar la polaridad de la alimentación del motor. Este circuito es el puente en H y a continuación se va a ver su funcionamiento en la figura 2.74:

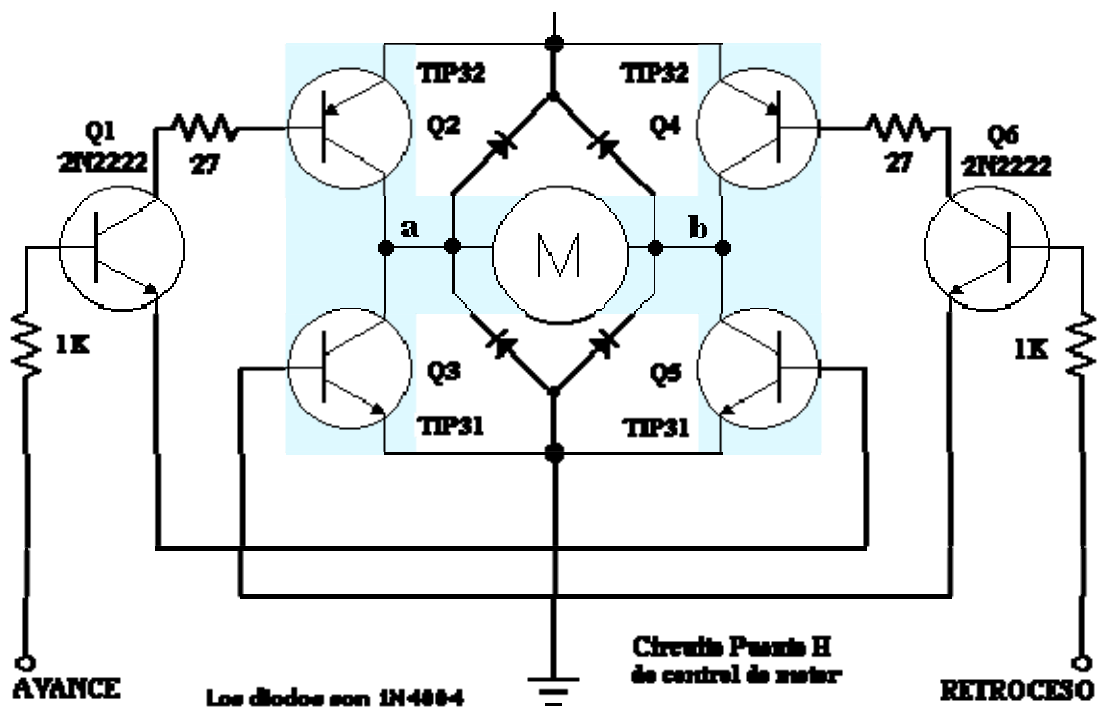


fig2. 74 Configuración del PUENTE H²⁴

Como muestra la figura, si se activa los transistores Q2 y Q5 (1 lógico) por medio de transistor Q1, el motor girará en un sentido y si activamos Q3 y Q4 por medio de Q6,

²⁴ Fuente: F. R. Bahía Blanca, Electrónica Aplicada II, Pag 70

la corriente circulará en sentido opuesto y por tanto el motor también. Así, se va a analizar los cuatro casos posibles, con los pares

Q2Q5/Q3Q4:

- 00 Los motores no reciben alimentación, quedan libres.
- 01 El motor gira en un sentido.
- 10 El motor gira en sentido contrario al anterior.
- 11 Frenado dinámico: Convine **no hacer**.

Se tiene que diferenciar los casos extremos: si el motor no se alimenta y tiene una velocidad dada por algún valor anterior, éste solo frenará por rozamiento y la inercia del robot. Si en cambio se utiliza el último caso, se le ordena al motor girar en los dos sentidos a la vez. Como consecuencia de esto, el motor se para de forma inmediata y eficaz, pero parece claro que consumirá mucha más corriente, existiendo la posibilidad de quemar el circuito o el propio motor en casos extremos.

2.10.4 La etapa de potencia.- En la placa está el circuito integrado L293. Este integrado contiene dos puentes en H y por tanto con él se podrá manejar hasta dos motores. El motor, gracias a este integrado, podrá ser alimentado a una tensión distinta de los 5V que emite el microcontrolador. Además, este integrado soporta hasta una un Amperio de corriente lo cual nos es de gran utilidad. La conexión del L293 queda establecida de la forma como indica la figura 2.75

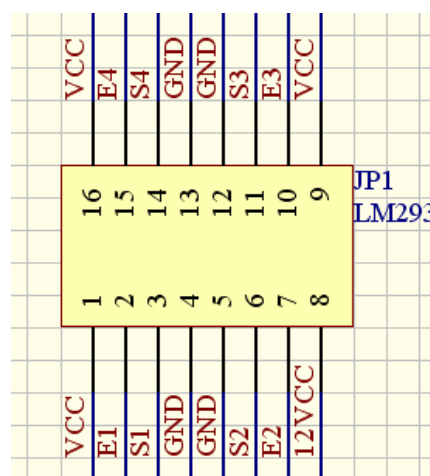


fig2. 75 Conexión del L293

Como se observa en la figura, el L293 tiene cuatro entradas (E1, E2, E3, E4), estas comandarán la dirección de los motores que se envía desde el microcontrolador, y a su vez también tiene cuatro salidas (S1, S2, S3, S4) que van conectados a cada extremo de los motores para que activen la dirección y velocidad que quiera.

La conexión del L293 junto con un motor queda establecida como muestra la figura 2.76 a continuación:

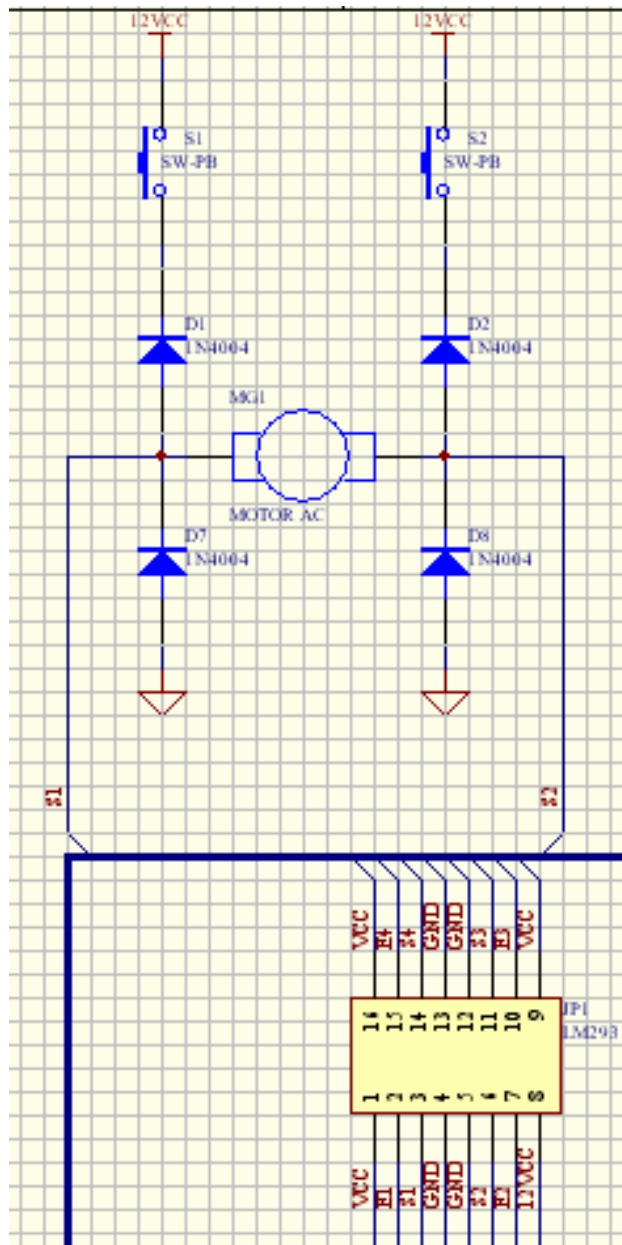


fig2. 76 Conexión del motor y del L293

Los diodos colocados en antiparalelo al motor sirven para proteger al integrado de las corrientes de redireccionamiento, es decir, cuando cambia de giro el motor.

2.11 Diseño de la fuente de alimentación

Una vez explicado los circuitos de mando y fuerza del circuito general para el mando y control del rugosímetro, ahora se explicará el funcionamiento de la fuente de alimentación que será conectada a las 2 placas anteriores.

2.11.1 Protección y rectificación.- En esta etapa, se alimentará al circuito con corriente alterna al conector J2, en el cual estará conectado un fusible de 2 Amperios para proteger a los elementos siguientes en caso de sobrevoltaje de la línea, luego se conecta un puente de Greatz para rectificar la señal y transformarla en corriente continua. El esquema de esta primera parte se muestra en la figura 2.77 a continuación:

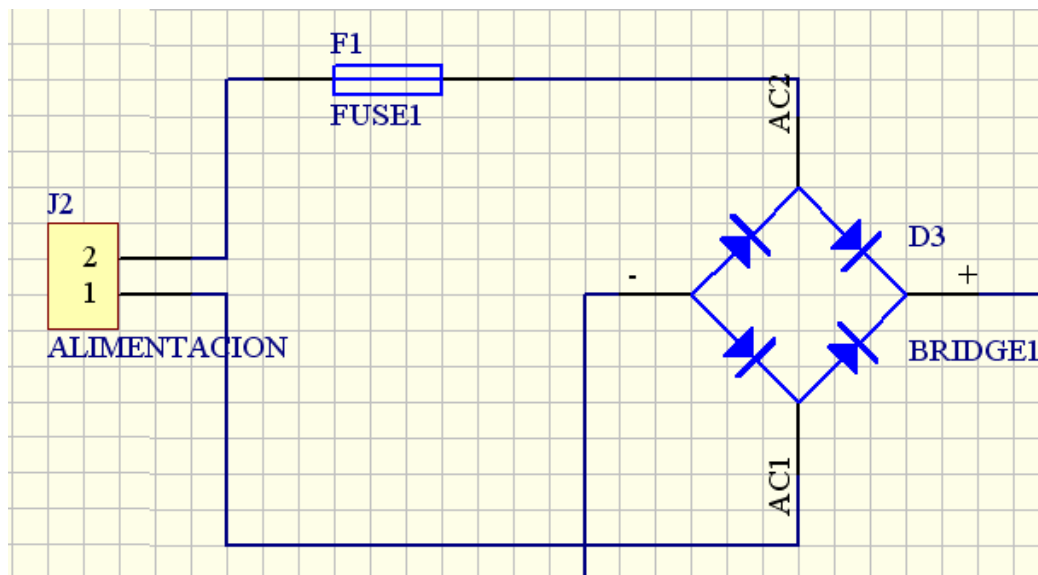


fig2. 77 Conexión del Puente de Greatz y del fusible de protección de línea

2.11.2 Estabilizadores de Tensión.- Una vez que ya se tiene la señal rectificada, se procede a estabilizar el voltaje a los valores con los que se va a trabajar en las tarjetas, tanto de mando como de fuerza.

Para esto se utiliza los integrados 78L12 para tener una salida estable de 12v y un 78L05 para tener una salida estable de 5v de corriente continua.

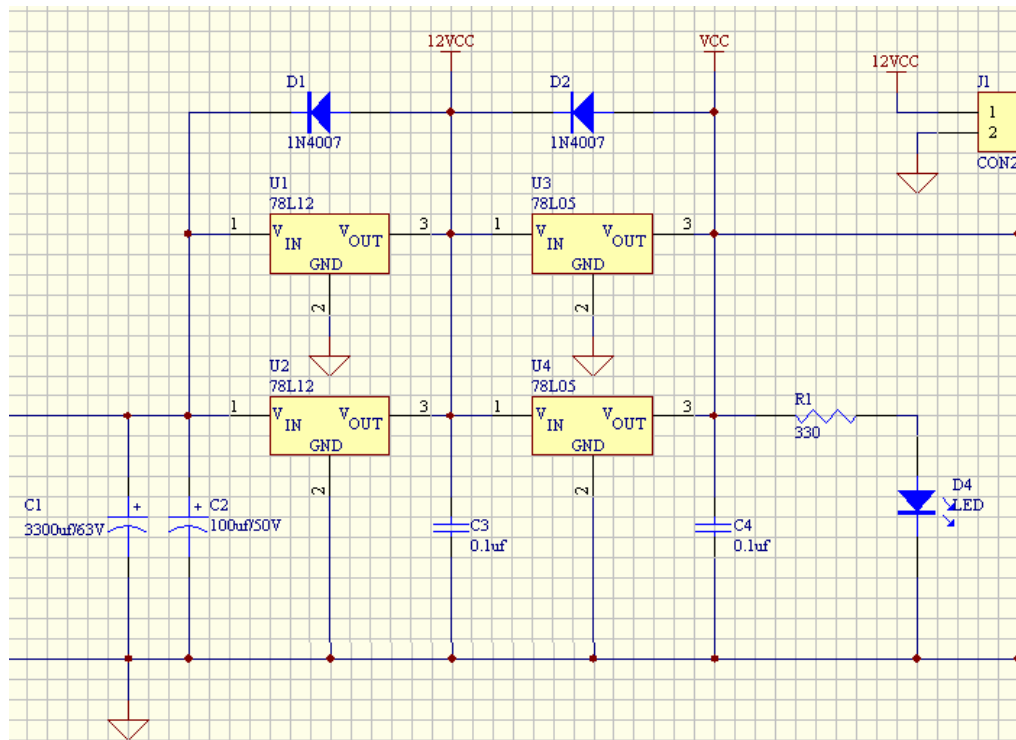


fig2. 78 Conexión de la fuente de alimentación

Como se observa en la figura 2.78, se ha colocado primero dos 78L12 en paralelo para elevar la corriente de la fuente y de la misma manera dos 78L05 en paralelo para obtener el mismo resultado.

El condensador C1 de 3300 microfaradios, nos sirve para eliminar el efecto ripple que se obtiene de la señal rectificada del puente de Greatz. Los condensadores C2, C3 y C4 nos sirven para purificar más la señal continua.

Los diodos colocados en antiparalelo a los integrados, sirven para protección de los mismos en caso de tener corrientes de redireccionamiento producidas por algún cortocircuito o defecto externo de la placa. Se tiene además, un LED de señalización que nos indicará cuando la fuente este en funcionamiento.

CAPITULO 3

DISEÑO DEL SOFTWARE

3.1 Diseño del firmware para microcontrolador.

3.1.1 Introducción teórica sobre los pic16f877.-

3.1.1.1 Introducción a los microcontroladores.- Los microcontroladores están conquistando el mundo. Están presentes en los trabajos, en casa y en la vida en general. Se pueden encontrar controlando el funcionamiento de los ratones y teclados de los computadores, en los teléfonos, en los hornos microondas y los televisores de nuestro hogar. Pero la invasión acaba de comenzar y el nacimiento del siglo **XXI** será testigo de la conquista masiva de estos diminutos computadores, que gobernarán la mayor parte de los aparatos que fabricarán y usarán los humanos.

3.1.1.2 Controlador y Microcontrolador.- Recibe el nombre de controlador el dispositivo que se emplea para el gobierno de uno o varios procesos. Por ejemplo, el controlador que regula el funcionamiento de un horno dispone de un sensor que mide constantemente su temperatura interna y, cuando traspasa los límites prefijados, genera las señales adecuadas que accionan los efectores que intentan llevar el valor de la temperatura dentro del rango estipulado.

Aunque el concepto de controlador ha permanecido invariable a través del tiempo, su implementación física ha variado frecuentemente. Hace tres décadas, los controladores se construían exclusivamente con componentes de lógica discreta, posteriormente se emplearon los microprocesadores, que se rodeaban con chips de memoria y E/S sobre una tarjeta de circuito impreso. En la actualidad, todos los elementos del controlador se han podido incluir en un chip, el cual recibe el nombre de **microcontrolador**. Realmente consiste en un sencillo pero completo computador contenido en el corazón (chip) de un circuito integrado.

Un microcontrolador es un circuito integrado de alta escala de integración que incorpora la mayor parte de los elementos que configuran un controlador.

Un microcontrolador dispone normalmente de los siguientes componentes:

- Procesador o UCP (Unidad Central de Proceso).
- Memoria RAM para Contener los datos.
- Memoria para el programa tipo ROM/PROM/EPROM.
- Líneas de E/S para comunicarse con el exterior.
- Diversos módulos para el control de periféricos (temporizadores, Puertas Serie y Paralelo, CAD: Conversores Analógico/Digital, CDA: Conversores Digital/Analógico, etc.).
- Generador de impulsos de reloj que sincronizan el funcionamiento de todo el sistema.

Los productos que para su regulación incorporan un microcontrolador disponen de las siguientes ventajas:

- Aumento de prestaciones: un mayor control sobre un determinado elemento representa una mejora considerable en el mismo.
- Aumento de la fiabilidad: al reemplazar el microcontrolador por un elevado número de elementos disminuye el riesgo de averías y se precisan menos ajustes.
- Reducción del tamaño en el producto acabado: La integración del microcontrolador en un chip disminuye el volumen, la mano de obra y los stocks.
- Mayor flexibilidad: las características de control están programadas por lo que su modificación sólo necesita cambios en el programa de instrucciones.

El microcontrolador es en definitiva un circuito integrado que incluye todos los componentes de un computador. Debido a su reducido tamaño es posible montar el controlador en el propio dispositivo al que gobierna. En este caso el controlador recibe el nombre de controlador empotrado.

3.1.1.3 Diferencia entre un Microprocesador y un Microcontrolador.- El microprocesador es un circuito integrado que contiene la Unidad Central de Proceso (UCP), también llamada procesador, de un computador. La UCP está formada por la

Unidad de Control, que interpreta las instrucciones, y el Camino de Datos, que las ejecuta.

Las patitas de un microprocesador sacan al exterior las líneas de sus buses de direcciones, datos y control, para permitir conectarle con la Memoria y los Módulos de E/S y configurar un computador implementado por varios circuitos integrados. Se dice que un microprocesador es un sistema abierto porque su configuración es variable de acuerdo con la aplicación a la que se destine. (Fig3 .79)

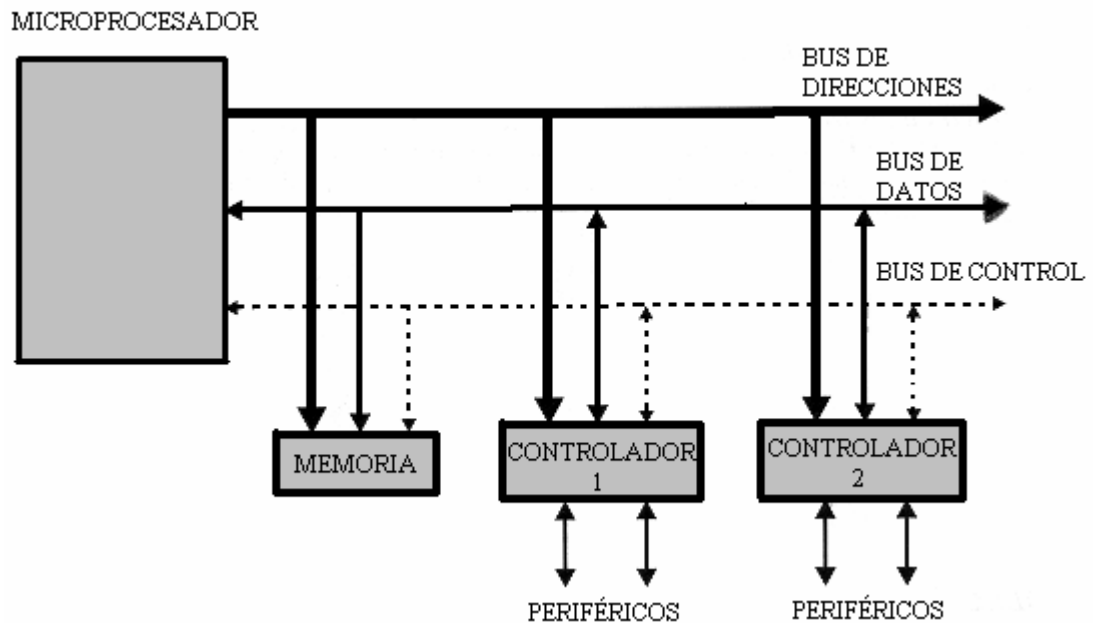


Fig3.79 Estructura de un sistema abierto basado en un microprocesador. La disponibilidad de los buses en el exterior permite que se configure a la medida de la aplicación²⁵.

Si sólo se dispusiese de un modelo de microcontrolador, éste debería tener muy potenciados todos sus recursos para poderse adaptar a las exigencias de las diferentes aplicaciones. Esta potenciación supondría en muchos casos un despilfarro. En la práctica cada fabricante de microcontroladores oferta un elevado número de modelos diferentes, desde los más sencillos hasta los más poderosos. Es posible seleccionar la capacidad de las memorias, el número de líneas de E/S, la cantidad y potencia de los elementos auxiliares, la velocidad de funcionamiento, etc. Por todo ello, un aspecto muy destacado del diseño es la selección del microcontrolador a utilizar.

²⁵ Fuente: <http://www.todopic.com.ar/apuntes/microcontroladores.htm>

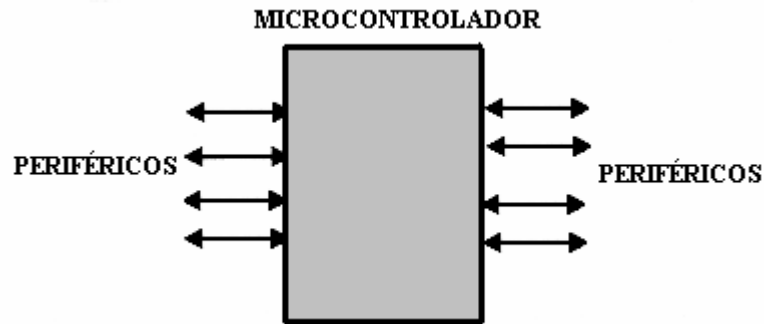


Fig 3.80 Microcontrolador

El microcontrolador es un sistema cerrado. Todas las partes del computador están contenidas en su interior y sólo salen al exterior las líneas que gobiernan los periféricos (fig3.80).

3.1.1.4 Aplicaciones de Microcontroladores.- Cada vez existen más productos que incorporan un microcontrolador con el fin de aumentar sustancialmente sus prestaciones, reducir su tamaño y coste, mejorar su fiabilidad y disminuir el consumo.

Algunos fabricantes de microcontroladores superan el millón de unidades de un modelo determinado producidas en **una semana**. Este dato puede dar una idea de la masiva utilización de estos componentes.

Los microcontroladores están siendo empleados en multitud de sistemas presentes en nuestra vida diaria, como pueden ser juguetes, horno microondas, frigoríficos, televisores, computadoras, impresoras, módems, el sistema de arranque de nuestro coche, etc. Y otras aplicaciones con las que seguramente no se esta tan familiarizados como instrumentación electrónica, control de sistemas en una nave espacial, etc. Una aplicación típica podría emplear varios microcontroladores para controlar pequeñas partes del sistema.

Estos pequeños controladores podrían comunicarse entre ellos y con un procesador central, probablemente más potente, para compartir la información y coordinar sus acciones, como, de hecho, ocurre ya habitualmente en cualquier PC.

3.1.1.5 El mercado de los microcontroladores.- Aunque en el mercado de la microinformática la mayor atención la acaparan los desarrollos de los microprocesadores, lo cierto es que se venden cientos de microcontroladores por cada uno de aquéllos.

Existe una gran diversidad de microcontroladores. Quizá la clasificación más importante sea entre microcontroladores de 4, 8, 16 ó 32 bits. Aunque las prestaciones de los microcontroladores de 16 y 32 bits son superiores a los de 4 y 8 bits, la realidad es que los microcontroladores de 8 bits dominan el mercado y los de 4 bits se resisten a desaparecer. La razón de esta tendencia es que los microcontroladores de 4 y 8 bits son apropiados para la gran mayoría de las aplicaciones, lo que hace absurdo emplear micros más potentes y consecuentemente más caros. Uno de los sectores que más tira del mercado del microcontrolador es el mercado automovilístico. De hecho, algunas de las familias de microcontroladores actuales se desarrollaron pensando en este sector, siendo modificadas posteriormente para adaptarse a sistemas más genéricos. El mercado del automóvil es además uno de los más exigentes: los componentes electrónicos deben operar bajo condiciones extremas de vibraciones, choques, ruido, etc. y seguir siendo fiables. El fallo de cualquier componente en un automóvil puede ser el origen de un accidente.

En cuanto a las técnicas de fabricación, cabe decir que prácticamente la totalidad de los microcontroladores actuales se fabrican con tecnología CMOS 4 (Complementary Metal Oxide Semiconductor). Esta tecnología supera a las técnicas anteriores por su bajo consumo y alta inmunidad al ruido.

La distribución de las ventas según su aplicación es la siguiente:

- Una tercera parte se absorbe en las aplicaciones relacionadas con los computadores y sus periféricos.
- La cuarta parte se utiliza en las aplicaciones de consumo (electrodomésticos, juegos, TV, vídeo, etc.)
- El 16% de las ventas mundiales se destinó al área de las comunicaciones.

- Otro 16% fue empleado en aplicaciones industriales.
- El resto de los microcontroladores vendidos en el mundo, aproximadamente un 10% fueron adquiridos por las industrias de automoción.

También los modernos microcontroladores de 32 bits van afianzando sus posiciones en el mercado, siendo las áreas de más interés el procesamiento de imágenes, las comunicaciones, las aplicaciones militares, los procesos industriales y el control de los dispositivos de almacenamiento masivo de datos.

3.1.1.6 ¿Qué microcontrolador emplear?.- A la hora de escoger el microcontrolador a emplear en un diseño concreto hay que tener en cuenta multitud de factores, como la documentación y herramientas de desarrollo disponibles y su precio, la cantidad de fabricantes que lo producen y por supuesto las características del microcontrolador (tipo de memoria de programa, número de temporizadores, interrupciones, etc.):

Costes.- Como es lógico, los fabricantes de microcontroladores compiten duramente para vender sus productos. Y no les va demasiado mal ya que sin hacer demasiado ruido venden 10 veces más microcontroladores que microprocesadores.

Si el fabricante desea reducir costes debe tener en cuenta las herramientas de apoyo con que va a contar: emuladores, simuladores, ensambladores, compiladores, etc. Es habitual que muchos de ellos siempre se decanten por microcontroladores pertenecientes a una única familia.

Aplicación. Antes de seleccionar un microcontrolador es imprescindible analizar los requisitos de la aplicación:

- **Procesamiento de datos:** puede ser necesario que el microcontrolador realice cálculos críticos en un tiempo limitado. En ese caso se debe asegurarse de seleccionar un dispositivo suficientemente rápido para ello.

Por otro lado, habrá que tener en cuenta la precisión de los datos a manejar: si no es suficiente con un microcontrolador de 8 bits, puede ser necesario acudir a microcontroladores de 16 ó 32 bits, o incluso a hardware de coma flotante. Una

alternativa más barata y quizá suficiente es usar librerías para manejar los datos de alta precisión. -

- **Entrada Salida:** para determinar las necesidades de Entrada/Salida del sistema es conveniente dibujar un diagrama de bloques del mismo, de tal forma que sea sencillo identificar la cantidad y tipo de señales a controlar. Una vez realizado este análisis puede ser necesario añadir periféricos hardware externos o cambiar a otro microcontrolador más adecuado a ese sistema.
- **Consumo:** algunos productos que incorporan microcontroladores están alimentados con baterías y su funcionamiento puede ser tan vital como activar una alarma antirrobo. Lo más conveniente en un caso como éste puede ser que el microcontrolador esté en estado de bajo consumo pero que despierte ante la activación de una señal (una interrupción) y ejecute el programa adecuado para procesarla.
- **Memoria:** para detectar las necesidades de memoria de nuestra aplicación debemos separarla en memoria volátil (RAM), memoria no volátil (ROM, EPROM, etc.) y memoria no volátil modificable (EEPROM). Este último tipo de memoria puede ser útil para incluir información específica de la aplicación como un número de serie o parámetros de calibración.

El tipo de memoria a emplear vendrá determinado por el volumen de ventas previsto del producto: de menor a mayor volumen será conveniente emplear EPROM, OTP y ROM. En cuanto a la cantidad de memoria necesaria puede ser imprescindible realizar una versión preliminar, aunque sea en pseudo-código, de la aplicación y a partir de ella hacer una estimación de cuánta memoria volátil y no volátil es necesaria y si es conveniente disponer de memoria no volátil modificable.

- **Ancho de palabra:** el criterio de diseño debe ser seleccionar el microcontrolador de menor ancho de palabra que satisfaga los requerimientos de la aplicación. Usar un microcontrolador de 4 bits supondrá una reducción en los costes importante, mientras que uno de 8 bits puede ser el más adecuado si el ancho de los datos es de un byte.

Los microcontroladores de 16 y 32 bits, debido a su elevado coste, deben reservarse para aplicaciones que requieran sus altas prestaciones (Entrada/Salida potente o espacio de direccionamiento muy elevado).

- **Diseño de la placa:** la selección de un microcontrolador concreto condicionará el diseño de la placa de circuitos. Debe tenerse en cuenta que quizá usar un microcontrolador barato encarezca el resto de componentes del diseño.

3.1.1.7 Arquitectura básica.- Aunque inicialmente todos los microcontroladores adoptaron la arquitectura clásica de von Neumann, en el momento presente se impone la arquitectura Harvard.

La arquitectura de von Neumann se caracteriza por disponer de una sola memoria principal donde se almacenan datos e instrucciones de forma indistinta. A dicha memoria se accede a través de un sistema de buses único (direcciones, datos y control).

La arquitectura Harvard dispone de dos memorias independientes una, que contiene sólo instrucciones y otra, sólo datos. Ambas disponen de sus respectivos sistemas de buses de acceso y es posible realizar operaciones de acceso (lectura o escritura) simultáneamente en ambas memorias.

En la figura 3.81 se puede observar que la arquitectura Harvard dispone de dos memorias independientes para datos y para instrucciones, permitiendo accesos simultáneos. Los microcontroladores **PIC** responden a la arquitectura Harvard.

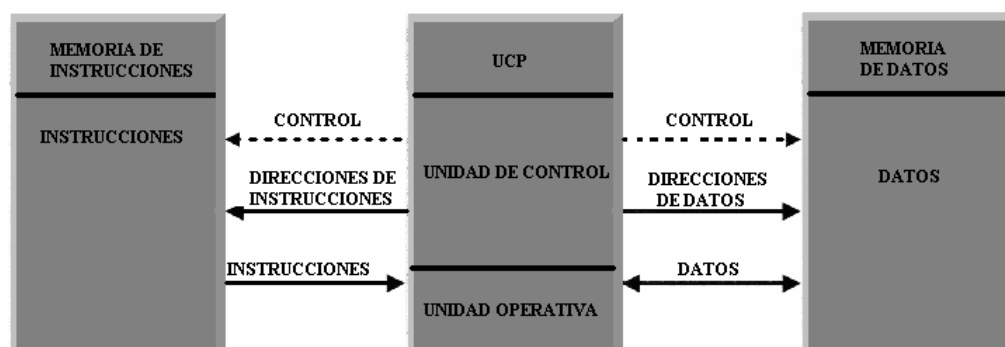


Fig 3.81 Memorias del hardware

3.1.1.8 El procesador o UCP.- Es el elemento más importante del microcontrolador y determina sus principales características, tanto a nivel hardware como software.

Se encarga de direccionar la memoria de instrucciones, recibir el código OP de la instrucción en curso, su decodificación y la ejecución de la operación que implica la instrucción, así como la búsqueda de los operandos y el almacenamiento del resultado.

Existen tres orientaciones en cuanto a la arquitectura y funcionalidad de los procesadores actuales.

CISC: Un gran número de procesadores usados en los microcontroladores están basados en la filosofía CISC (Computadores de Juego de Instrucciones Complejo). Disponen de más de 80 instrucciones máquina en su repertorio, algunas de las cuales son muy sofisticadas y potentes, requiriendo muchos ciclos para su ejecución. Una ventaja de los procesadores CISC es que ofrecen al programador instrucciones complejas que actúan como macros.

RISC: Tanto la industria de los computadores comerciales como la de los microcontroladores están decantándose hacia la filosofía RISC (Computadores de Juego de Instrucciones Reducido). En estos procesadores el repertorio de instrucciones máquina es muy reducido y las instrucciones son simples y, generalmente, se ejecutan en un ciclo. La sencillez y rapidez de las instrucciones permiten optimizar el hardware y el software del procesador. Los microcontroladores **PIC** utilizan tecnología RISC.

SISC: En los microcontroladores destinados a aplicaciones muy concretas, el juego de instrucciones, además de ser reducido, es “específico”, o sea, las instrucciones se adaptan a las necesidades de la aplicación prevista. Esta filosofía se ha bautizado con el nombre de SISC (Computadores de Juego de Instrucciones Específico).

3.1.1.9 Memoria.- En los microcontroladores la memoria de instrucciones y datos está integrada en el propio chip. Una parte debe ser no volátil, tipo ROM, y se destina a contener el programa de instrucciones que gobierna la aplicación. Otra

parte de memoria será tipo RAM, volátil, y se destina a guardar las variables y los datos.

Hay dos peculiaridades que diferencian a los microcontroladores de los computadores personales:

1. No existen sistemas de almacenamiento masivo como disco duro o disquetes.
2. Como el microcontrolador sólo se destina a una tarea en la memoria ROM, sólo hay que almacenar un único programa de trabajo.

La RAM en estos dispositivos es de poca capacidad pues sólo debe contener las variables y los cambios de información que se produzcan en el transcurso del programa. Por otra parte, como sólo existe un programa activo, no se requiere guardar una copia del mismo en la RAM pues se ejecuta directamente desde la ROM.

Los usuarios de computadores personales están habituados a manejar Megabytes de memoria, pero, los diseñadores con microcontroladores trabajan con capacidades de ROM comprendidas entre 512 bytes y 8 k bytes y de RAM comprendidas entre 20 y 512 bytes.

Según el tipo de memoria ROM que dispongan los microcontroladores, la aplicación y utilización de los mismos es diferente. Se describen las cinco versiones de memoria no volátil que se pueden encontrar en los microcontroladores del mercado.

1º. ROM con máscara.- Es una memoria no volátil de sólo lectura cuyo contenido se graba durante la fabricación del chip. El elevado coste del diseño de la máscara sólo hace aconsejable el empleo de los microcontroladores con este tipo de memoria cuando se precisan cantidades superiores a varios miles de unidades.

2ª. OTP .- El microcontrolador contiene una memoria no volátil de sólo lectura “programable una sola vez” por el usuario. OTP (One Time Programmable). Es el usuario quien puede escribir el programa en el chip mediante un sencillo grabador controlado por un programa desde un PC. La versión OTP es recomendable cuando es

muy corto el ciclo de diseño del producto, o bien, en la construcción de prototipos y series muy pequeñas.

Tanto en este tipo de memoria como en la EPROM, se suele usar la encriptación mediante fusibles para proteger el código contenido.

3ª EPROM.- Los microcontroladores que disponen de memoria EPROM (Erasable Programmable Read Only Memory) pueden borrarse y grabarse muchas veces. La grabación se realiza, como en el caso de los OTP, con un grabador gobernado desde un PC. Si, posteriormente, se desea borrar el contenido, disponen de una ventana de cristal en su superficie por la que se somete a la EPROM a rayos ultravioleta durante varios minutos. Las cápsulas son de material cerámico y son más caras que los microcontroladores con memoria OTP que están hechos con material plástico.

4ª EEPROM.- Se trata de memorias de sólo lectura, programables y borrables eléctricamente EEPROM (Electrical Erasable Programmable Read Only Memory). Tanto la programación como el borrado, se realizan eléctricamente desde el propio grabador y bajo el control programado de un PC. Es muy cómoda y rápida la operación de grabado y la de borrado. No disponen de ventana de cristal en la superficie.

Los microcontroladores dotados de memoria EEPROM una vez instalados en el circuito, pueden grabarse y borrarse cuantas veces se quiera sin ser retirados de dicho circuito. Para ello se usan “grabadores en circuito” que confieren una gran flexibilidad y rapidez a la hora de realizar modificaciones en el programa de trabajo.

El número de veces que puede grabarse y borrarse una memoria EEPROM es finito, por lo que no es recomendable una reprogramación continua. Son muy idóneos para la enseñanza y la Ingeniería de diseño.

Se va extendiendo en los fabricantes la tendencia de incluir una pequeña zona de memoria EEPROM en los circuitos programables para guardar y modificar cómodamente una serie de parámetros que adecuan el dispositivo a las condiciones del entorno. Este tipo de memoria es relativamente lenta.

5ª FLASH.-Se trata de una memoria no volátil, de bajo consumo, que se puede escribir y borrar. Funciona como una ROM y una RAM pero consume menos y es más pequeña. A diferencia de la ROM, la memoria FLASH es programable en el circuito. Es más rápida y de mayor densidad que la EEPROM. La alternativa FLASH está recomendada frente a la EEPROM cuando se precisa gran cantidad de memoria de programa no volátil. Es más veloz y tolera más ciclos de escritura/borrado.

Las memorias EEPROM y FLASH son muy útiles al permitir que los microcontroladores que las incorporan puedan ser reprogramados “en circuito”, es decir, sin tener que sacar el circuito integrado de la tarjeta.

Así, un dispositivo con este tipo de memoria incorporado al control del motor de un automóvil permite que pueda modificarse el programa durante la rutina de mantenimiento periódico, compensando los desgastes y otros factores tales como la compresión, la instalación de nuevas piezas, etc. La reprogramación del microcontrolador puede convertirse en una labor rutinaria dentro de la puesta a punto.

3.1.1.10 Puertas de Entrada y Salida.- La principal utilidad de las patitas que posee la cápsula que contiene un microcontrolador es soportar las líneas de E/S que comunican al computador interno con los periféricos exteriores.

Según los controladores de periféricos que posea cada modelo de microcontrolador, las líneas de E/S se destinan a proporcionar el soporte a las señales de entrada, salida y control.

3.1.1.11 Reloj principal.- Todos los microcontroladores disponen de un circuito oscilador que genera una onda cuadrada de alta frecuencia, que configura los impulsos de reloj usados en la sincronización de todas las operaciones del sistema.

Generalmente, el circuito de reloj está incorporado en el microcontrolador y sólo se necesitan unos pocos componentes exteriores para seleccionar y estabilizar la

frecuencia de trabajo. Dichos componentes suelen consistir en un cristal de cuarzo junto a elementos pasivos o bien un resonador cerámico o una red R-C.

Aumentar la frecuencia de reloj supone disminuir el tiempo en que se ejecutan las instrucciones pero lleva aparejado un incremento del consumo de energía.

3.2.1 Recursos especiales

Cada fabricante oferta numerosas versiones de una arquitectura básica de microcontrolador. En algunas amplía las capacidades de las memorias, en otras incorpora nuevos recursos, en otras reduce las prestaciones al mínimo para aplicaciones muy simples, etc. La labor del diseñador es encontrar el modelo mínimo que satisfaga todos los requerimientos de su aplicación. De esta forma, minimizará el coste, el hardware y el software.

Los principales recursos específicos que incorporan los microcontroladores son:

- Temporizadores o “Timers”.
- Perro guardián o “Watchdog”.
- Protección ante fallo de alimentación o “Brownout”.
- Estado de reposo o de bajo consumo.
- Conversor A/D.
- Conversor D/A.
- Comparador analógico.
- Modulador de anchura de impulsos o PWM.
- Puertas de E/S digitales.
- Puertas de comunicación.

3.2.1.1 Temporizadores o “Timers”.- Se emplean para controlar periodos de tiempo (temporizadores) y para llevar la cuenta de acontecimientos que suceden en el exterior (contadores). Para la medida de tiempos se carga un registro con el valor adecuado y a continuación dicho valor se va incrementando o decrementando al ritmo de los impulsos de reloj o algún múltiplo hasta que se desborde y llegue a 0, momento en el que se produce un aviso.

Cuando se desean contar acontecimientos que se materializan por cambios de nivel o flancos en alguna de las patitas del microcontrolador, el mencionado registro se va incrementando o decrementando al ritmo de dichos impulsos.

3.2.1.2 Perro guardián o “Watchdog”.- Cuando el computador personal se bloquea por un fallo del software u otra causa, se pulsa el botón del reset y se reinicializa el sistema. Pero un microcontrolador funciona sin el control de un supervisor y de forma continuada las 24 horas del día. El Perro guardián consiste en un temporizador que, cuando se desborda y pasa por 0, provoca un reset automáticamente en el sistema.

Se debe diseñar el programa de trabajo que controla la tarea de forma que refresque o inicialice al Perro guardián antes de que provoque el reset. Si falla el programa o se bloquea, no se refrescará al Perro guardián y, al completar su temporización, “ladrará y ladrará” hasta provocar el reset.

3.2.1.3 Protección ante fallo de alimentación o “Brownout”.- Se trata de un circuito que resetea al microcontrolador cuando el voltaje de alimentación (VDD) es inferior a un voltaje mínimo (“brownout”). Mientras el voltaje de alimentación sea inferior al de brownout el dispositivo se mantiene reseteado, comenzando a funcionar normalmente cuando sobrepasa dicho valor.

3.2.1.4 Estado de reposo ó de bajo consumo.- Son abundantes las situaciones reales de trabajo en que el microcontrolador debe esperar, sin hacer nada, a que se produzca algún acontecimiento externo que le ponga de nuevo en funcionamiento. Para ahorrar energía, (factor clave en los aparatos portátiles), los microcontroladores disponen de una instrucción especial (SLEEP en los PIC), que les pasa al estado de reposo o de bajo consumo, en el cual los requerimientos de potencia son mínimos. En dicho estado se detiene el reloj principal y se “congelan” sus circuitos asociados, quedando sumido en un profundo “sueño” el microcontrolador. Al activarse una interrupción ocasionada por el acontecimiento esperado, el microcontrolador se despierta y reanuda su trabajo.

3.2.1.5 Conversor A/D (CAD).- Los microcontroladores que incorporan un Conversor A/D (Analógico/Digital) pueden procesar señales analógicas, tan

abundantes en las aplicaciones. Suelen disponer de un multiplexor que permite aplicar a la entrada del CAD diversas señales analógicas desde las patitas del circuito integrado.

3.2.1.6 Conversor D/A (CDA).- Transforma los datos digitales obtenidos del procesamiento del computador en su correspondiente señal analógica que saca al exterior por una de las patitas de la cápsula. Existen muchos efectores que trabajan con señales analógicas.

3.2.1.7 Comparador analógico.- Algunos modelos de microcontroladores disponen internamente de un Amplificador Operacional que actúa como comparador entre una señal fija de referencia y otra variable que se aplica por una de las patitas de la cápsula. La salida del comparador proporciona un nivel lógico 1 ó 0 según una señal sea mayor o menor que la otra.

También hay modelos de microcontroladores con un módulo de tensión de referencia que proporciona diversas tensiones de referencia que se pueden aplicar en los comparadores.

3.2.1.8 Modulador de anchura de impulsos o PWM.- Son circuitos que proporcionan en su salida impulsos de anchura variable, que se ofrecen al exterior a través de las patitas del encapsulado.

3.2.1.9 Puertas de E/S digitales.- Todos los microcontroladores destinan algunas de sus patitas a soportar líneas de E/S digitales. Por lo general, estas líneas se agrupan de ocho en ocho formando Puertas. Las líneas digitales de las Puertas pueden configurarse como Entrada o como Salida cargando un 1 ó un 0 en el bit correspondiente de un registro destinado a su configuración.

3.2.1.10 Puertas de comunicación.- Con objeto de dotar al microcontrolador de la posibilidad de comunicarse con otros dispositivos externos, otros buses de microprocesadores, buses de sistemas, buses de redes y poder adaptarlos con otros elementos bajo otras normas y protocolos. Algunos modelos disponen de recursos que permiten directamente esta tarea, entre los que destacan:

- UART, adaptador de comunicación serie asíncrona.
- USART, adaptador de comunicación serie síncrona y asíncrona
- Puerta paralela esclava para poder conectarse con los buses de otros microprocesadores.
- USB (Universal Serial Bus), que es un moderno bus serie para los PC.
- Bus I²C, que es un interfaz serie de dos hilos desarrollado por Philips.
- CAN (Controller Area Network), para permitir la adaptación con redes de conexión multiplexado desarrollado conjuntamente por Bosch e Intel para el cableado de dispositivos en automóviles. En EE.UU. se usa el J1850.

3.3 La familia de los pic como elección.

Para las aplicaciones más habituales (casi un 90%) la elección de una versión adecuada de PIC es la mejor solución; sin embargo, dado su carácter general, otras familias de microcontroladores son más eficaces en aplicaciones específicas, especialmente si en ellas predomina una característica concreta, que puede estar muy desarrollada en otra familia.

Los detalles más importantes que vuelven “locos” a los profesionales de la microelectrónica y microinformática y las razones de la excelente acogida que tienen los PIC son los siguientes:

- Sencillez de manejo: Tienen un juego de instrucciones reducido; 35 en la gama media.
- Buena información, fácil de conseguir y económica.
- Precio: Su coste es comparativamente inferior al de sus competidores.
- Poseen una elevada velocidad de funcionamiento. Buen promedio de parámetros: velocidad, consumo, tamaño, alimentación, código compacto, etc.
- Herramientas de desarrollo fáciles y baratas. Muchas herramientas software se pueden recoger libremente a través de Internet desde Microchip <http://www.microchip.com>.

- Existe una gran variedad de herramientas hardware que permiten grabar, depurar, borrar y comprobar el comportamiento de los PIC.
- Diseño rápido.
- La gran variedad de modelos de PIC permite elegir el que mejor responde a los requerimientos de la aplicación.

Una de las razones del éxito de los PIC se basa en su utilización. Cuando se aprende a manejar uno de ellos, conociendo su arquitectura y su repertorio de instrucciones, es muy fácil emplear otro modelo.

3.3.1 Características Relevantes.-

3.3.1.1 Arquitectura.- La arquitectura del procesador sigue el modelo Harvard. En esta arquitectura, la CPU se conecta de forma independiente y con buses distintos con la memoria de instrucciones y con la de datos.

La arquitectura Harvard permite a la CPU acceder simultáneamente a las dos memorias. Además, propicia numerosas ventajas al funcionamiento del sistema como se irán describiendo.

3.3.1.2 Segmentación.- Se aplica la técnica de segmentación (“pipe-line”) en la ejecución de las instrucciones. La segmentación permite al procesador realizar al mismo tiempo la ejecución de una instrucción y la búsqueda del código de la siguiente.

De esta forma se puede ejecutar cada instrucción en un ciclo (un ciclo de instrucción equivale a cuatro ciclos de reloj).

Las instrucciones de salto ocupan dos ciclos al no conocer la dirección de la siguiente instrucción hasta que no se haya completado la de bifurcación.

3.3.1.3 Formato de las instrucciones.- El formato de todas las instrucciones es de la misma longitud. Todas las instrucciones de los microcontroladores de la gama baja tienen una longitud de 12 bits. Las de la gama media tienen 14 bits y más las de la gama alta. Esta característica es muy ventajosa en la optimización de la memoria de

instrucciones y facilita enormemente la construcción de ensambladores y compiladores.

3.3.1.4 Juego de instrucciones.- Procesador RISC (Computador de Juego de Instrucciones Reducido). Los modelos de la gama baja disponen de un repertorio de 33 instrucciones, 35 los de la gama media y casi 60 los de la alta.

3.3.1.5 Todas las instrucciones son ortogonales.- Cualquier instrucción puede manejar cualquier elemento de la arquitectura como fuente o como destino.

3.3.1.6 Arquitectura basada en un “banco de registros”.- Esto significa que todos los objetos del sistema (puertas de E/S, temporizadores, posiciones de memoria, etc.) están implementados físicamente como registros.

3.3.1.7 Diversidad de modelos de microcontroladores con prestaciones y recursos diferentes.- La gran variedad de modelos de microcontroladores PIC permite que el usuario pueda seleccionar el más conveniente para su proyecto.

3.3.1.8 Herramientas de soporte potentes y económicas.- La empresa Microchip y otras que utilizan los PIC ponen a disposición de los usuarios numerosas herramientas para desarrollar hardware y software. Son muy abundantes los programadores, los simuladores software, los emuladores en tiempo real, Ensambladores, Compiladores C, Intérpretes y Compiladores BASIC, etc.

La arquitectura Harvard y la técnica de segmentación son los principales recursos en los que se apoya el elevado rendimiento que caracteriza estos dispositivos programables, mejorando dos características esenciales:

1. Velocidad de ejecución.
2. Eficiencia en la compactación del código.

3.4 Introducción teórica al programa picbasic pro v2.44

3.4.1 Introduccion.- El compilador PicBasic Pro (PBP) es un lenguaje de programación de nueva generación que hace más fácil y rápido para programar microcontroladores Pic micro de Microchip Technology . El lenguaje Basic es mucho más fácil de leer y escribir que el lenguaje ensamblador Microchip.

3.4.2 Los Micros.- El PBP produce código que puede ser programado para una variedad de micro controladores PIC que tengan de 8 a 68 pins y varias opciones en el chip incluyendo convertidores A/D, temporizadores y puertos seriales.

Hay algunos micros PIC que no trabajaran con el PBP, por ejemplo las series PIC 16C5X incluyendo el PIC 16C54 Y PIC 15C58. Estos micro PIC están basados en el viejo núcleo de 12 bit en lugar del núcleo más corriente de 14 bit.

El PBP necesita alguna de las opciones que solamente están disponibles con el núcleo de 14 bit como el stack (pila) de 8 niveles.

3.4.3 Programación.- Para operar el PBP, necesitará un editor ó procesador de texto para crear su programa fuente, algún tipo de programador de micros PIC como el MICROCODE STUDIO PLUS 2.2.1.1 y el propio PIC BASIC PRO 2.44.

La secuencia de eventos es similar a la siguiente:

Primero cree el archivo fuente BASIC para el programa, usando su editor o procesador de texto preferido. Si lo desea, EDIT (incluído en DOS) ó NOTEPAD (incluído en WINDOWS), pueden ser utilizados.El nombre del archivo fuente debe terminar con la extensión .BAS (pero no es excluyente).

El archivo de texto creado debe ser texto ASCII puro. No debe contener códigos especiales insertados por procesadores de texto para sus propósitos específicos. Normalmente se tiene la opción de grabar el archivo como texto ASCII puro en la mayoría de los procesadores de texto.

Ejemplo de programa para hacer parpadear un LED conectado al puerto PORTB.0 , aproximadamente una vez por segundo:

```

loop: high PORTB.0      ´ enciende el LED.
                        ´
pause 500              ´ demora de .5 segundos
                        ´
low PORTB.0           ´ apaga el LED
                        ´
pause 500              ´ demora de .5 segundos

```

goto loop ' vuelve a loop y hace parpadear el LED indefinidamente

end

Una vez que el programa escrito funcionará sin errores puede compilarlo de la siguiente manera:

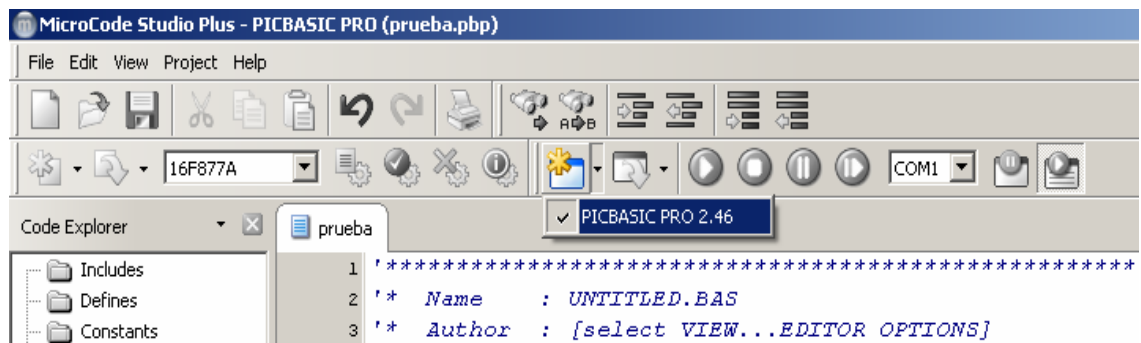


fig3.82 Compilación del programa

Como muestra la figura 3.82, se selecciona la opción de compilar en la barra de herramientas y se genera un cartel en la parte inferior izquierda, que indica que se compilo correctamente.

Si todo funciona bien, se crea un archivo de código microPIC (en este caso prueba.HEX). Si existen errores, se emitirá un listado de los mismos, que deberán ser corregidos en su archivo fuente BASIC antes de ser compilados nuevamente.

Para seleccionar diferentes microcontroladores debemos seleccionar en la pestaña que se encuentra en la pestaña superior izquierda.

3.4.4 Bases del PBP

3.4.4.1 Identificadores.- Un identificador es simplemente un nombre. Son usados en PBP como etiquetas de líneas y nombres de variables. Un identificador es cualquier secuencia de letras, dígitos y símbolos, aunque no deben comenzar con un dígito.

Los identificadores no distinguen las letras mayúsculas de las minúsculas, por lo que **etiqueta**, **ETIQUETA**, **EtiquEtA**, son todas tratadas como equivalentes. Aunque las etiquetas pueden tener cualquier número de caracteres de longitud PBP solamente reconoce los primeros 32.

3.4.4.2 Etiquetas de línea (LABELS).- Para marcar líneas que el programa puede desear referenciar con comandos GOTO ó GOSUB, PBP usa etiquetas de línea. PBP no permite número de línea y no requiere que cada línea sea etiquetada. Cualquier línea puede comenzar con una etiqueta de línea que es simplemente un identificador seguido por un punto y coma (;)

A continuación un ejemplo de lo antes mencionado:

mostrar: Serout 0, N2400, ["UNIVERSIDAD, AZUAY!"], 13, 10]

Goto mostrar

3.4.4.3 Variables.- Variables es donde se guardan datos en forma temporaria. Son creadas usando la palabra clave VAR y pueden ser bits, bytes ó word. Espacio para cada variable es automáticamente destinado en la memoria del micro controlador por PBP. El formato para crear una variable es el siguiente:

Etiqueta VAR tamaño (.modificadores)

Etiqueta es cualquier identificador excluyendo palabras claves como se describe anteriormente. Tamaño es bit, byte ó word. Modificadores opcionales agregan control adicional acerca de cómo se crea la variable. Algunos ejemplos de creación de variables son:

perro var byte

gato var bit

W0 var word

No hay variables predefinidas de usuarios de PBP. Por razones de compatibilidad existen dos archivos que crean las variables standard usadas con BASIC stamps: "bs1defs.bas" y "bs2defs.bas". Para usar uno de estos archivos agregue la línea:

Include "bs1defs.bas" ò Include "bs2defs.bas"

Estos archivos contienen numerosas declaraciones VAR que crean todas las variables de BASIC Stamps y definiciones de pin. El número de variables

disponibles depende de la cantidad de RAM en un dispositivo en particular y el tamaño de las variables y los arrays. PBP reserva aproximadamente 24 posiciones RAM para su propio uso. También puede crear variables temporarias adicionales para usar en ordenamiento de ecuaciones complejas.

3.4.4.4 Alias.- VAR también puede ser usado para crear un alias para una variable. Esto es muy útil para acceder al interior de una variable. A continuación presentamos la tabla 3.1 de ejemplos de creación de alias a variables:

Modificador	Descripción
BIT0 O 0	Crea alias al bit 0 de byte o word
BIT1 O 1	Crea alias al bit 1 de byte o word
BIT3 O 3	Crea alias al bit 3 de byte o word
BIT5 O 5	Crea alias al bit 5 de byte o word
BIT7 O 7	Crea alias al bit 7 de byte o word
BIT9 O 9	Crea alias al bit 9 de word
BIT11 O 11	Crea alias al bit 11 de word
BIT13 O 13	Crea alias al bit 13 de word
BIT15 O 15	Crea alias al bit 15 de word
BYTE0 O LOWBYTE	Crea alias al lowbyte de word
BYTE1 O HIGHBYTE	Crea alias al highbyte de word

Tabla3.1 Creación de Alias a una variable²⁶

3.4.4.5 Arrays (Arreglos).- Los arreglos de variables pueden ser creados en una manera similar a las variables.

Etiqueta VAR tamaño (número de elementos)

²⁶ <http://www.micros1.com.ar/picbasic/tablas1.htm>

Etiqueta es cualquier identificador, excluyendo palabras claves, como se describió anteriormente. Tamaño es BIT, BYTE ó WORD. Número de elementos es cuantos lugares en el arreglo se desean. Algunos ejemplos de creación de arreglo son los siguientes:

sharks var byte[10]

fish var bit [8]

La primera ubicación dentro del arreglo es el elemento cero. En el arreglo fish anterior los elementos están numerados fish (0) a fish (7) conteniendo 8 elementos en total. Dada la forma en que los arreglos están localizados en memoria hay límites de tamaño para cada tipo como muestra la tabla 3.2.

Tamaño	Número máximo de elementos
BIT	128
BYTE	64
WORD	32

Tabla3.2 Límites de tamaño

3.4.4.6 Constantes.- Las llamadas constantes pueden ser creadas de manera similar a las variables. Puede ser más conveniente usar un nombre de constante en lugar de un número constante. Si el número necesita ser cambiado, únicamente puede ser cambiando en un lugar del programa donde se define la constante. No pueden guardarse datos variables dentro de una constante.

Etiqueta CON expresión constante

Algunos ejemplos son:

Mice con 3

Traps con mice *1000

3.4.4.7 Símbolos.- SYMBOL provee otro método para renombrar (darle alias) a variables y constantes, pero puede ser usado para crear una variable.

SYMBOL lion = cat ´ cat fue previamente creada usando VAR

SYMBOL mouse = 1 ´ igual que mouse con 1

3.4.4.8 Constantes Numéricas.- PBP permite definir constantes numéricas en tres bases: decimal, binario y hexadecimal. Valores binarios son definidos usando el prefijo “%” y valores hexadecimales usando el prefijo “\$”. Los valores decimales se toman por defecto y no requieren prefijo.

%100 ´ valor binario para el decimal 4.

\$100 ´ valor hexadecimal para el decimal 256.

Para facilitar la programación, los caracteres son convertidos en sus equivalentes ASCII. La constante debe ser puesta entre comillas y contener sólo un caracter (de lo contrario, ellas son una cadena de constantes).

“A” ´ ASCII valor para el decimal 65

“d” ´ ASCII valor para el decimal 100

3.4.4.9 Cadena de Constantes.- PBP no provee capacidad de manejo de cadenas, pero las cadenas pueden ser usadas con algunos comandos. Una cadena contiene uno o más caracteres y es delimitado entre comillas. No se soportan secuencias de escape para caracteres no-ASCII (aunque, la mayoría de los comandos PBP tienen este manejo incorporado)

“UDA” ´ String (forma abreviada de “U”, “D”, “A)

Las cadenas son usualmente tratadas como una lista de valores de caracteres individuales.

3.4.4.10 Puertos y otros registros.- Todos los registros inclusive los puertos del PICmicro MCU, pueden ser accedidos como cualquier otra variable en PicBasic.

Esto significa que pueden ser leídos, escritos o utilizados en ecuaciones directamente:

PORTA = %01010101. 'Escribe el valor en el PUERTO A

3.4.4.11 Pines.- A los pines se puede acceder de diferentes modos. El mejor camino para especificar un pin para una operación, es simplemente usar sus nombres **PORT** y un número de bit:

PORTB.1= 'Colocar PORTB, bit 1 a 1

Para recordar fácilmente para qué puede ser usado un pin, debe asignarse un nombre usando el comando VAR. De esta manera, el nombre puede ser utilizado luego en cualquier operación:

Led var PORTA.O 'Renombra PORTA.O como led

High led 'Coloca led (PORTA:O) en valor alto

Para compatibilidad con el BASIC Stamp, los pines usados en los comandos del PBP pueden, además, ser referidos por un número, 0-15. Estos pins están físicamente distribuidos sobre diferentes puertos del hardware del micro PIC, dependiendo de cuántos pins tiene el microcontroladores.

Para colocar un pin o port como salida (ó entrada) debe dar valores al registro **TRIS**. Colocando el bit de **TRIS** como **0**, hace su pin una **salida**, y colocandolo en **1** lo hace una **entrada**. Por ejemplo:

TRISA = %00000000 'O TRISA = 0 'Coloca todos los pins PORTA como salidas.

TRISB =% 11111111 'O TRISB = 1 'Coloca todos los pins PORTB como entradas.

TRISC =% 10101010 'Coloca todos los pins pares como salidas y los impares como entradas.

Cada bit individual puede ser manejado de la misma manera

TRISA.0 = 0 ‘Coloca el PORTA, pin 0 como salida. Todos los demás pin permanecen sin cambio.

3.4.4.12 Comentarios.- Un comentario de PBP comienza con la palabra clave REM o el apóstrofe ('). Todos los demás caracteres de esa línea se ignoran.

3.4.4.13 Declaraciones múltiples.- Para permitir programas más compactos y agrupamientos lógicos de comandos relacionados, PBP soporta el uso de (:) para separar comandos ubicados en la misma línea. Los siguientes dos ejemplos son equivalentes.

W2 = W0

W0 = W1

W1 = W2

Es lo mismo que:

W2 = w0 : W0 = W1 : W1 = W2

En los dos casos, el tamaño del código generado es el mismo.

3.4.4.14 Include.- Se puede agregar archivos fuente BASIC a un programa PBP usando **INCLUDE**. Usted puede tener su rutina standard, definiciones u otros archivos que desee guardar en forma separada. Los archivos de definición de modo serial y de stamp son ejemplo de esta. Estos archivos pueden ser incluidos en programas donde ser necesario, pero no en programas donde no se los necesita.

Las líneas de código fuente del archivo incluido son insertadas dentro del programa exactamente donde se coloca el **INCLUDE**.

INCLUDE “modedefs.bas”

3.4.4.15 Define.- Algunos elementos, como el oscilador y las ubicaciones de los pin LCD, están predefinidos en PBP. **DEFINE** le permite a un programa PBP cambiar estas definiciones si así lo desea.

Define puede ser usado para cambiar el valor predefinido del oscilador, los pins de DEBUG y el baud rate y las ubicaciones de los pin LCD además de otras cosas. Estas definiciones deben estar en mayúsculas, la tabla 3.3 muestra algunas de las definiciones que PBP tiene establecidas:

DEFINE BUTTON_PAUSE 50	demora en el anti-rebote del botón en ms
DEFINE CHAR_PACING 1000	paso de la salida serial en us
DEFINE DEBUG_REG _PORTL	Standard del pin port
DEFINE DEBUG_BIT 0	standard n del pin bit
DEFINE DEBUG_BAUD 2400	standardn del baud rate
DEFINE DEBUG_MODE 1	modo Standard n: 0=CIERTO,1=INVERTIDO
DEFINE DEBUG_PACING 1000	paso de Standard n en us
DEFINE HSER_RCSTA 90 h	setear registro receive
DEFINE HSER_TXSTA 20 h	setear registro transmit
DEFINE HSER_BAUD 2400	setear baud rate
DEFINE HSER_EVEN 1	usar solo si se desea paridad par
DEFINE HSER_ODD 1	usar solo si se desea paridad impar
DEFINE I2C_INTERNAL 1	usar para EEPROM interno en 16CEXX y 12CEXX
DEFINE I2C_SLOW 1	usar para OSC > 8 Mhz con dispositivos de velocidad standard
DEFINE LCD_DREG PORTB	port de data LCD
DEFINE LCD_DBIT 0	datos LCD comenzando en bit 0 o 4
DEFINE LCD_RSREG PORTB	port de selección de registro LCD

DEFINE LCD_RSBIT 4	bit de selección de registro LCD
DEFINE LCD_EREG PORTB	port de habilitacion LCD
DEFINE LCD_EBIT 5	bit de habilitacion LCD
DEFINE LCD_BITS 4	bus del LCD de 4 u 8 bits
DEFINE LCD_LINES 2	numero de lineas en LCD
DEFINE OSC 4	3 (3.58) 4 8 10 12 16 20 Mhz.
DEFINE OSCCAL_1K 1	setea OSCCAL para PIC12C671
DEFINE OSCCAL_2K 1	setea OSCCAL para PIC12C672

Tabla3.3 Definiciones establecidas en PBP²⁷

3.4.4.16 Operadores matemáticos.- PBP efectua todas las operaciones matemáticas en urden jerárquico. Esto significa que existe precedencia para los operadores. Multiplicación y división son efectuados antes que suma y resta. Para asegurarse que las operaciones son efectuadas en el orden que se desea, use paréntesis para agrupar las operaciones.

$$A = (B + C) * (D - E)$$

Todas las operaciones matemáticas se realizan sin signo y con una precisión de 16 bit.

Los operadores soportados son las que muestra la tabla 3.4 :

Operador matemático	Descripción
+	Suma
-	Resta
*	Multiplicación
**	16 bits superiores de la Multiplicación

²⁷ <http://www.micros1.com.ar/picbasic/tablasdefine.htm>

*/	16 nits medios de la Multiplicación
/	Division
//	Resto (módulo)
<<	Desplazamiento izquierdo
>>	Desplazamiento derecho
ABS	Valor absoluto
COS	Coseno
DCD	2m decodificador
DIG	Digito
MAX	Maximo *
MIN	Minimo *
NCD	Codificar
REV	Invertir bits
SIN	Seno
SQR	Raiz cuadrada
&	Bit inteligente AND
÷	Bit inteligente OR
^	Bit inteligente EXCLUSIVE OR
~	Bit inteligente NOT
& /	Bit inteligente NOT AND
÷ /	Bit inteligente NOT OR
^ /	Bit inteligente NOT EXCLUSIVE OR

Tabla3.4 Operaciones matemáticas soportadas²⁸

²⁸ <http://www.micros1.com.ar/picbasic/tablasopemate.htm>

3.4.4.17 Operadores de Comparación.- Se usan en declaraciones **IF ... THEN** para comparar una expresión con otra .Los operadores soportados son los indicados en la tabla 3.5 :

Operador	Descripción
= o ==	Igual
<> o !=	No igual
<	Menor
>	Mayor
<=	Menor o igual
>=	Mayor o igual

Tabla3.5 Operadores de comparación²⁹

3.4.4.18 Operadores Lógicos.- Los operadores lógicos difieren de las operaciones de bit inteligente. Entregan un resultado CIERTO / FALSO de su operación. Valores 0 son tratados como falso. Cualquier otro valor es ciert³⁰o. Se usan junto a operadores de comparación en una declaración IF .. THEN. En la tabla 3.6 se presenta los operadores soportados:

Operador	Descripción
AND o &&	AND logico
OR o	OR logico
XOR o ^^	OR exclusivo logico
NOT AND	NAND logico
NOT OR	NOR logico
NOT XOR	NXOR logico

Tabla3.6 Operadores lógicos

²⁹ <http://www.micros1.com.ar/picbasic/tablasopcomp.htm>

³⁰ <http://www.micros1.com.ar/picbasic/tablasoplog.htm>

Ejemplo:

If (A == big) AND (B > mean) then run

Asegurese de usar parentesis para indicarle a PBP el orden en que quiere que se realicen las operaciones.

3.4.4.19 Referencia de declaraciones.- La tabla 3.7 muestra las referencias de declaraciones que tiene el programa PBP para el distinto manejo en los programas que se desee realizar:

<u>@</u>	Inserta una linea de codigo ensamblador
<u>ASM...ENDASM</u>	Inserta una seccion de codigo ensamblador
<u>BRANCH</u>	GOTO computado(equiv. a ON..GOTO)
<u>BRANCHL</u>	BRANCH fuera de pagina(BRANCH largo)
<u>BUTTON</u>	Anti-rebote y auto-repeticion de entrada en pin especificado
<u>CALL</u>	Llamada a subrutina de ensamblador
<u>CLEAR</u>	Hace cero todas las variables
<u>COUNT</u>	Cuenta el numero de pulsos en un pin
<u>DATA</u>	Define el contenido inicial en un chip EEPROM
<u>DEBUG</u>	Señal asincronica de salida en un pin fijo y baud
<u>DISABLE</u>	Deshabilita el procesamiento de ON INTERRUPT
<u>DTMFOUT</u>	Produce tonos en un pin
<u>EEPROM</u>	Define el contenido inicial en un chip EEPROM
<u>ENABLE</u>	Habilita el procesamiento de ON INTERRUPT
<u>END</u>	Detiene la ejecucion e ingresa en modo de baja potencia
<u>FOR...NEXT</u>	Ejecuta declaraciones en forma repetitiva
<u>FREQOUT</u>	Produce hasta 2 frecuencias en un pin

<u>GOSUB</u>	Llama a una subrutina BASIC en la etiqueta especificada
<u>GOTO</u>	Continúa la ejecución en la etiqueta especificada
<u>HIGH</u>	Hace alto la salida del pin
<u>HSERIN</u>	Entrada serial asincronica(hardware)
<u>HSEROUT</u>	Salida serial asincronica(hardware)
<u>I2CREAD</u>	Lee bytes de dispositivo I2C
<u>I2CWRITE</u>	Graba bytes en dispositivo I2C
<u>IF..THEN..ELSE..ENDIF</u>	Ejecuta declaraciones en forma condicional
<u>INPUT</u>	Convierte un pin en entrada
(<u>LET</u>)	Asigna el resultado de una expresión a una variable
<u>LCDOUT</u>	Muestra caracteres en LCD
<u>LOOKDOWN</u>	Busca un valor en una tabla de constantes
<u>LOOKDOWN2</u>	Busca un valor en una tabla de constantes o variables
<u>LOOKUP</u>	Obtiene un valor constante de una tabla
<u>LOOKUP2</u>	Obtiene un valor constante o variable de una tabla
<u>LOW</u>	Hace bajo la salida de un pin
<u>NAP</u>	Apaga el procesador por un corto periodo de tiempo
<u>ON INTERRUPT</u>	Ejecuta una subrutina BASIC en un interrupt
<u>OUTPUT</u>	Convierte un pin en salida
<u>PAUSE</u>	Demora (resolución 1mseg.)
<u>PAUSEUS</u>	Demora (resolución 1 useg.)
<u>PEEK</u>	Lee un byte del registro
<u>POKE</u>	Graba un byte en el registro

<u>POT</u>	Lee el potenciómetro en el pin especificado
<u>PULSIN</u>	Mide el ancho de pulso en un pin
<u>PULSOUT</u>	Genera pulso hacia un pin
<u>PWM</u>	Salida modulada en ancho de pulso a un pin
<u>RANDOM</u>	Genera número pseudo-aleatorio
<u>RCTIME</u>	Mide el ancho de pulso en un pin
<u>READ</u>	Lee byte de un chip EEPROM
<u>RESUME</u>	Continúa la ejecución después de una interrupción
<u>RETURN</u>	Continúa en la declaración que sigue al último GOSUB
<u>REVERSE</u>	Convierte un pin de salida en entrada o viceversa
<u>SERIN</u>	Entrada serial asincrónica (tipo BS!)
<u>SERIN2</u>	Entrada serial asincrónica (tipo BS2)
<u>SEROUT</u>	Salida serial asincrónica (tipo BS1)
<u>SEROUT2</u>	Salida serial asincrónica (tipo BS2)
<u>SHIFTIN</u>	Entrada serial síncrona
<u>SHIFTOUT</u>	Salida serial síncrona
<u>SLEEP</u>	Apaga el procesador por un período de tiempo
<u>SOUND</u>	Genera un tono o ruido blanco en un pin
<u>STOP</u>	Detiene la ejecución del programa
<u>SWAP</u>	Intercambia los valores de dos variables
<u>TOGGLE</u>	Hace salida a un pin y cambia su estado
<u>WHILE..WEND</u>	Ejecuta declaraciones mientras la condición sea cierta
<u>WRITE</u>	Graba bytes a un chip EEPROM
<u>XIN</u>	Entrada X - 10

<u>XOUT</u>	Salida X - 10
-------------	---------------

Tabla3.7 Referencia de declaraciones³¹

3.4.4.20 ADCIN.- Lee el conversor analogico del micro y guarda el resultado en el Var. Mientras que los registros del ADC se pueden alcanzar directamente, ADCIN hace el proceso un poco más fácil.

ADCIN Channel , Var

Antes de que ADCIN pueda ser utilizado, el registro de TRIS se debe fijar como entradas. ADCON1 también necesita ser asignado como entradas de información analogicas y en algunos casos para fijar el formato del resultado y la fuente del reloj.

Varios DEFINES pueden también ser utilizados. Los valores por defecto se muestran abajo:

DEFINE ADC_BITS 8 ' Fije el número de BITS en el resultado (8, 10 o 12).

DEFINE ADC_CLOCK 3 'Fije EL CLOCK (rc = 3)

DEFINE ADC_SAMPLEUS 50 ' Fije el tiempo de muestreo en microsegundos

ADC_SAMPLEUS es el número demicrosegundos que el programa espera entre fijar el canal y comenzar la conversión analogica/digital.

TRISA = 255 ' Fije PORTA todas entradas

ADCON1 = 0 ' PORTA es analogico

ADCIN 0, B0 ' Lea el canal 0 a B0

3.4.4.21 ASM ENDASM.- Estas instrucciones le dicen a PBP que el codigo entre estas dos líneas esta en lenguaje ensamblador y no debe ser interpretado como declaraciones PBP .Se puede usar estas dos instrucciones libremente para mezclar código ensamblador con declaraciones PBP.

³¹ <http://www.micros1.com.ar/picbasic/tablasrefdec.htm>

El tamaño máximo para una sección de texto ensamblador es 8 K. Este es el tamaño máximo para el fuente actual, incluyendo comentarios, no el código generado.

A continuación un ejemplo de esta herramienta:

ASM

Bsf PORTA,0 ;setea bit 0 en PORTA

Bcf PORTB.0 ;setea bit 0 en PORTB

ENDASM

3.4.4.22 CALL.- Ejecuta la subrutina ensamblador llamada etiqueta. Normalmente se usa GOSUB para ejecutar una subrutina PBP.

CALL etiqueta

La principal diferencia entre GOSUB y CALL, es que con ésta última no se chequea la existencia de etiquetas hasta el momento de ensamblar.

Usando CALL se puede acceder a una etiqueta en una sección de lenguaje ensamblador , lo que es inaccesible mediante PBP.

CALL pass ´ ejecuta la subrutina ensamblada, denominada pass

3.4.4.23 CLEAR.- Coloca en cero todos los registros en cada banco. Coloca en cero todas las variables, incluyendo las del sistema .Esto no se hace automáticamente al comenzar un programa en PBP , como sucede en BASIC Stamps .Por lo general ,las variables deben ser colocadas en un estado inicial apropiado por el programa , y no usando CLEAR .

CLEAR ´ Coloca todas las variables en cero

3.4.4.24 EEPROM.- _ Guarda constantes en un chip EEPROM. Si se omite el valor opcional Location, la primera declaración se guarda en la dirección 0 del EEPROM y las subsiguientes en las siguientes direcciones del mismo .Si se indica un valor Location, éste indica la dirección de comienzo para guardar los datos.

EEPROM {Location,} [constante {,constante ...}]

Constante puede ser una constante numérica ó una cadena de constantes. Solo se guardan los bytes menos significativos de los valores numéricos. Las cadenas son guardadas como bytes consecutivos d valores ASCII .No se agregan automáticamente terminadores, ni se completa el largo.

EEPROM solo trabaja con microcontroladores con EEPROM incorporado. Dado que el EEPROM es una memoria no volátil, los datos permanecerán intactos aún sin alimentación.

Los datos son guardados en el EEPROM solo una vez, cuando el micro controlador es programado, no cada vez que se ejecuta el programa. Se puede usar WRITE para colocar valores en el EEPROM en el momento de la ejecución.

3.4.4.25 END.- Detiene la ejecución del proceso y entra en modo de baja potencia .Todos los pins de I/O permanecen en el estado en que se encuentran, END trabaja ejecutando una instrucción SLEEP continua dentro de un loop.

3.4.4.26 FOR .. NEXT.- El loop FOR .. NEXT permite a los programas ejecutar un número de declaraciones (Body) un numero de veces, usando una variable como contador. Debido a su complejidad y versatilidad, es mejor describirla paso a paso.

FOR Count = Start **TO** End {**STEP** {-} Inc}

{Body}

NEXT {Count}

El valor de Start se asigna a la variable índice, Count, que puede ser una variable de cualquier tipo.

Se ejecuta el Body, este es opcional y puede ser omitido (quizás por un loop de demora).

El valor de Inc es sumado a (ó restado si se especifica “-“) Count. Si no se define una cláusula STEP , se incrementa Count en uno.

Si Count no pasó End ó desbordó el tipo de variable, la ejecución vuelve al paso 2.

Si el loop necesita contar más de 255 (Count > 255), se debe usar una variable de tamaño word.

3.4.4.27 GOTO.- La ejecución del programa continúa en la declaración de la etiqueta, ejemplo:

GOTO etiqueta

GOTO send ´ salta a la declaración etiquetada send.

3.4.4.28 INPUT.- Convierte el Pin especificado en una entrada. Pin debe ser una constante, 0-15, o una variable que contenga un número 0-15 (p. ej., B0) o el nombre de un pin (p. ej., PORTA.0).

INPUT Pin

INPUT PORTA.0 ´ convierte el PORTA, pin 0 en entrada

En forma alternativa, el pin puede ser colocado como entrada de una forma más rápida y corta:

TRISB.0 =1 ´ Setea el PORTB, pin 0 como entrada

Todos los pins en un port pueden ser colocados como entradas seteando el registro TRIS completo de una sola vez:

TRISB = %11111111 ´ Setea todo el PORTB como entrada

3.4.4.29 OUTPUT.- Convierte el pin especificado en salida. Pin puede ser una constante, 0 - 15, ó una variable que contenga un número de 0-15 (p.ej. B0) ó un número de Pin (p.ej. PORTA.0)

OUTPUT Pin

OUTPUT PORTA.0 ´ convierte PORTA pin 0 en salida

En forma alternativa, el pin puede ser convertido en salida de una manera más rápida y corta:

`TRISB.0 = 0` ´ setea PORTB pin 0 como salida

Todos los pins de un port pueden ser seteados simultaneamente como salida usando el registro TRIS completo:

`TRISB = %00000000` ´ setea todos los pins de PORTB como salidas

3.4.4.30 ON INTERRUPT.- Permite el manejo de las interrupciones del micro controlador por medio de una subrutina PBP. Existen dos formas de manejar interrupciones usando PBP. La primera es escribir una subrutina de interrupción en lenguaje ensamblador. El segundo método es escribir un handler (manejador) de interrupciones PBP.

ON INTERRUPT GOTO Label

Cuando ocurre una interrupción, se marca con una bandera. Cuando la ejecución de la declaración PBP que se estaba ejecutando termina, el programa salta al handler de interrupciones indicado en Label. Una vez que termina el trabajo del handler, una declaración RESUME envía el programa de vuelta a donde estaba cuando ocurrió la interrupción, tomando todo como lo dejó.

DISABLE y ENABLE permiten que distintas secciones de un programa PBP se ejecuten sin la posibilidad de ser interrumpidas. El lugar más notorio para usar DISABLE es justo antes del actual handler de interrupciones.

Latencia es el tiempo entre el pedido de interrupción y el momento en que se ingresa en el handler de interrupciones. Como las declaraciones de PBP no son reentrantes (p.ej. no se puede ejecutar una declaración mientras se está ejecutando una anterior), puede existir una latencia considerable hasta que se ingrese a la rutina de interrupciones.

PBP no ingresará al handler BASIC de interrupciones hasta que haya terminado de ejecutar la declaración en curso. Si la declaración es PAUSE ó SERIN, puede demorarse bastante hasta que sea reconocida la interrupción.

Se debe diseñar el programa tomando en cuenta esta latencia. Si esta es inaceptable y las interrupciones deben ser manejadas más rápidamente, se debe usar una rutina en lenguaje ensamblador.

ON INTERRUPT agregará una instrucción después de cada declaración, para chequear si ocurre ó no una interrupción DISABLE elimina la posibilidad de usar esta instrucción y ENABLE la habilita. Normalmente, las instrucciones adicionales no son un problema, pero programas largos en pequeñas computadoras si lo son.

3.4.4.31 PWM.- Envía un tren de pulsos modulados en ancho a Pin .Cada ciclo de PWM está compuesto de 256 pasos. El ciclo útil Duty para cada ciclo varía de 0 (0%) a 255 (100%= .El ciclo PWM es repetido Cycle veces. Pin puede ser una constante, 0 - 15, ó una variable que contenga un número de 0-15 (p.ej. B0) ó un número de Pin (p.ej. PORTA.0)

PWM Pin,Duty,Cycle

Cycle depende de la frecuencia del oscilador .Con un oscilador de 4 Mhz , cada Cycle será de aproximadamente 5 mseg. de largo .Con un oscilador de 20 Mhz el largo aproximado será de 1 mseg. Definir un valor de OSC no tiene efecto sobre PWM . El tiempo de Cycle siempre cambia con la velocidad del oscilador en uso .

Pin se convierte en salida justo antes de la generación del pulso y vuelve a ser entrada , cuando cesa .La salida de PWM en un pin tiene mucho ruido , y no tiene forma de onda cuadrada .Es necesario usar algún tipo de filtro para convertirla en algo útil . Un circuito R/C se puede usar como un simple convertidor D/A .

A continuación un ejemplo del manejo de PWM:

PWM PORTB.7,127,100 ´ envía una señal PWM con un ciclo útil del 50% al pin 7 , durante 100 ciclos

3.4.4.32 RETURN.- Vuelve desde una subrutina. Retoma la ejecución en la declaración que sigue al GOSUB que llamó la subrutina, ejemplo:

Gosub sub1 ´ va a la subrutina denominada sub1

sub1: serout 0,N2400,["Lunch"] ´ envia "Lunch" al pin 0 en forma serial

RETURN ´ vuelve al programa principal despues del gosub

3.4.4.33 SERIN.- Recibe uno ó más Items en Pin, en formato standard asincrónico, usando 8 bit de datos, sin paridad y un stop bit (8N1) . Pin puede ser una constante , 0 - 15 , ó una variable que contenga un número de 0-15 (p.ej. B0) ó un número de Pin (p.ej. PORTA.0)

SERIN Pin,Mode, {Timeout,Label,} {[Qual...],} {Item...}

Timeout y Label son opciones que pueden ser incluídas para permitir al programa continuar si no se recibe un carácter durante un cierto tiempo. Timeout está especificado en unidades de 1 milisegundo.

La lista de Items de datos a ser recibida puede estar precedida por uno ó más calificadores encerrados entre corchetes. SERIN debe recibir estos bytes en un orden exacto, antes de recibir los datos. Si algún byte recibido no concuerda con el byte siguiente de la secuencia de calificación, el proceso de calificación comienza nuevamente (p.ej. el próximo hYTE recibido es comparado con el primer Item de la lista de calificación). Un Qualifier puede ser constante, variable ó una cadena de constantes. Cada carácter de una cadena es tratado como un calificador individual.

Una vez que se completan los calificadores, SERIN comienza a guardar datos en la variable asociada con cada Item. Si el nombre de variable es único, el valor del carácter ASCII recibido es guardado en la variable. Si la variable es precedida por el signo # . SERIN convierte un valor decimal en ASCII y guarda el resultado en esa variable. Todos los no-dígitos recibidos antes del primer dígito del valor decimal son ignorados y descartados. El carácter no-dígito que termina el valor decimal también se descarta.

Aunque los chips convertidores de nivel RS-232 son comunes y baratos, las excelentes especificaciones de I/O de los microPIC permiten ejecutar muchas aplicaciones sin usar convertidores de nivel. Más aún, se pueden usar entradas invertidas (N300...N9600) junto con un resistor limitador de corriente de 22k.

SERIN 1,N2400, [“A”],B0 ´ espera hasta que el carácter “A” sea recibido en forma serial en el pin 1 y coloca el próximocaracter en B0

3.4.4.34 SEROUT.- Envía uno ó más Items a Pin, en formato standard asincrónico usando 8 bits de datos, sin paridad y 1 stop bit (8N1). SEROUT es similar al comando Serout de BS1. Pin es automáticamente colocado como salida. Pin puede ser una constante, 0 - 15, ó una variable que contenga un número de 0-15 (p.ej. B0) ó un número de Pin (p.ej. PORTA.0)

SEROUT Pin,Mode,[Item[,Item...}]

SEROUT soporta 3 tipos distintos de datos, que pueden ser combinados libremente dentro de una declaración SEROUT.

Una cadena de constantes es enviada como una cadena de caracteres literales.

En algunos casos, los rangos de transmisión de SEROUT pueden presentar los caracteres demasiado rápidamente en el dispositivo receptor. Un DEFINE agrega tiempo entre caracteres en la transmisión de salida. Esto permite un tiempo adicional entre caracteres a medida que son transmitidos. Se puede lograr una demora entre cada carácter transmitido de 1 a 65535 microsegundos (.001 a 65,535 milisegundos).

Por ejemplo, para pausar 1 milisegundo entre cada carácter transmitido:

```
DEFINE CHAR_PACING 1000
```

3.4.4.35 STOP.- Detiene la ejecución del programa , ejecutando un loop sin fin ,No coloca al micro controlador en modo de baja potencia .El micro controlador trabaja igual que siempre.

STOP ´ envía el programa a un loop - detiene

3.4.4.36 SWAP.-Intercambia los valores de dos variables. Normalmente intercambiar los valores de dos variables es un proceso tedioso. SWAP lo hace con una sola declaración, sin variables intermedias. Puede ser usado con variables de bit, byte y

word. Variables de array con índice variable no son permitidas, pero sí variables de array con índice constante.

SWAP Variable1,Variable2

Temp = B0 ' anteriormente

B0=B1

B1=Temp

SWAP B0 ,B1 ' ahora

3.4.4.37 WHILE...WEND.- Ejecuta las declaraciones Statement en forma repetida, mientras la condición Condition sea cierta. Cuando Condition deja de ser cierta, la ejecución continúa con la declaración siguiente al WEND. Condition puede ser cualquier expresión de comparación.

WHILE Condition

WEND

A continuación ponemos un ejemplo de la utilización de esta herramienta:

I=1

WHILE i < = 10

Serout 0,N2400,["No:",#i,13,10]

WEND

3.5 Programacion del pic

3.5.1 Aspectos Generales.- El microcontrolador desempeñara las funciones de enviar el dato de la conversión del palpador mecánico a la PC y a su vez recibirá las órdenes de inicio y paro del sistema desde la misma.

Cabe recalcar que el sistema establecido para el comando del rugosímetro tiene su parte manual y automática. Dicha opción se podrá elegir desde la PC al inicio del programa.

3.5.2 Diagrama de bloques del programa.- Como se observa en la figura 3.83, se ha dividido al programa del microcontrolador en tres bloques principales:



fig3.83 Diagrama de bloques del programa del microcontrolador

En el primer bloque se configurará todos los parámetros requeridos en el sistema para que funcione correctamente, tales como el conversor, la configuración del USART, variables, etc. En el segundo bloque se establecerá el protocolo de comunicación que se utiliza para poder enviar y recibir datos a la PC y en el bloque 3 como comandar los motores de la parte mecánica. A continuación se detalla cada uno de los bloques.

3.5.2.1 Configuraciones generales.-

3.5.2.1.1 Configuración del oscilador.- Es muy importante que al inicio del programa se deba establecer el oscilador con el que se está trabajando, debido a que es necesario tener una sincronización correcta entre el microcontrolador y la PC, caso contrario, las demoras, datos de conversión y comunicación serán erróneas. Para establecer el oscilador se utiliza el siguiente comando:

DEFINE OSC 16

Este comando automáticamente dejará establecido un oscilador de 16 Mhz como muestra la figura 3.84

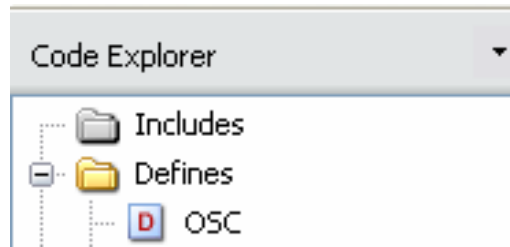


fig3.84 Declaración del oscilador

3.5.2.1.2 Configuración del USART.- Para la configuración del USART (Receptor y Transmisor Universal Síncrono y Asíncrono), se utiliza los siguientes comandos:

DEFINE HSER_RCSTA 90h

Se asigna al registro RCSTA el número en hexadecimal 90 siendo este su representación binaria el número 10010000 con los cuales se establece la configuración que presenta la tabla 3.8:

BIT	NUM	NOMBRE	DESCRIPCION
Bit 7	1	SPEN	Puerto serial habilitado
Bit 6	0	RX9	8 bits de recepción
Bit 5	0	SREN	No utilizado
Bit 4	1	CREN	Recepción continua habilitada
Bit 3	0	ADDEN	Detección de dirección deshabilitada
Bit 2	0	FERR	Error de trama deshabilitado
Bit 1	0	OERR	Error de desborde deshabilitado
Bit 0	0	RX9D	Bit de paridad deshabilitado

Tabla 3.8 Configuración del registro RCSTA

DEFINE HSER_TXSTA 20h

Al registro de transmisión se asigna el valor hexadecimal 20, siendo este en binario 00100000, quedando configurada la transmisión de la siguiente manera tal como lo indica la tabla 3.9:

BIT	NUM	NOMBRE	DESCRIPCION
Bit 7	0	CSRC	No utilizado
Bit 6	0	TX9	8 bits de transmisión
Bit 5	1	TXEN	Transmisión habilitada
Bit 4	0	SYNC	Modo asíncrono
Bit 3	0	UNIMPL	No utilizado
Bit 2	0	BRGH	Velocidad baja
Bit 1	0	TRMT	Full transmisión
Bit 0	0	TX9D	Bit de paridad deshabilitado

Tabla 3.9 Configuración del registro TXSTA

Luego de configurar estos dos registros, se necesita establecer la velocidad a la que el microprocesador y la PC se van a comunicar, para esto, es prioritario cargar un valor al registro SPBRG según el oscilador con el que se este trabajando, dicho valor se lo calcula con la fórmula 3.1 que se detallo en capítulos anteriores:

$$X = \left(\frac{\left(\frac{Fosc}{velocidad} \right)}{64} \right) - 1 \quad (3.1)$$

Donde:

X = valor a cargar en el SPBRG

Fosc = oscilador utilizado

Velocidad = velocidad requerida

Utilizando la fórmula con el oscilador de 16 Mhz que se estableció con anterioridad y si se quiere comunicar a una velocidad de 6900 baudios, el valor de X a cargar en el registro SPBRG es 25 mediante el siguiente comando:

DEFINE HSER_SPBRG 25

3.5.2.1.3 Configuración del conversor.- Antes de poder utilizar el conversor del microcontrolador es necesario configurarlo en el programa, para esto se necesita los siguientes comandos:

DEFINE ADC_BITS 8

Configuración de dato en 8 bits

DEFINE ADC_CLOCK 3

Dato de reloj preestablecida

DEFINE ADC_SAMPLEUS 50

Tiempo de muestreo 50 milisegundos

TRISA = %00000001

Configuración del puerto A al bit 0 como entrada

ADCON0 = %10000000

Configuración para oscilador de 16 Mhz

ADCON1 = %00000100

Establecer entrada analógica

3.5.2.1.4 Declaración de variables.- Antes de comenzar el programa, se establece las variables que se va a utilizar con nombres sencillos y concretos para su fácil manejo a lo largo del mismo.

LED1 VAR PORTB.0	variable led1 al bit 0 del Puerto B
LED2 VAR PORTB.1	variable led2 al bit 1 del Puerto B
MOT11 VAR PORTC.0	movimiento izquierda del motor del eje X
MOT12 VAR PORTC.1	movimiento derecha del motor del eje X
MOT21 VAR PORTC.2	movimiento izquierda del motor del eje Y
MOT22 VAR PORTC.3	movimiento derecha del motor del eje Y
DAT VAR WORD	variable de comunicación de recepción
DAT1 VAR WORD	variable de comunicación de transmisión
DAT2 VAR WORD	variable de comunicación de recepción
I VAR BYTE	variable para loop
J VAR BYTE	variable para loop
K VAR BYTE	variable para loop
RES VAR WORD	variable multiplicadora para pausa en loop

A1 VAR BYTE

variable para dato convertido

Una vez que ya se tiene configurado todos los parámetros iniciales, se puede programar al microcontrolador las opciones que se requieran para el funcionamiento del Rugosímetro (fig 3.85)

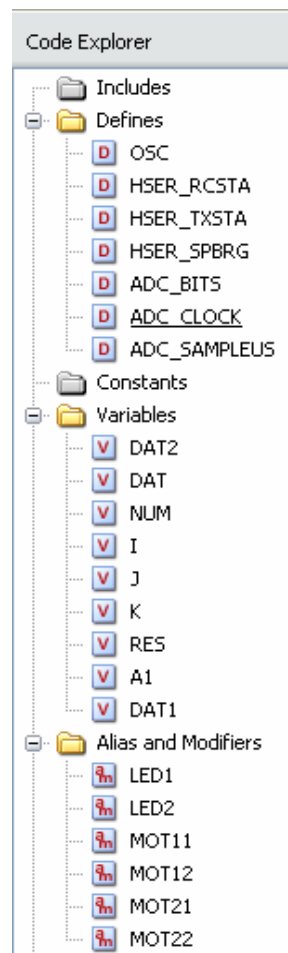


fig3.85 Configuraciones iniciales del programa

3.5.2.2 Transmisión y recepción de datos.-

3.5.2.2.1 Recepción del dato de superficie.- Una vez que el sistema está configurado, se establece ahora un protocolo de comunicación con la PC, dando como prioridad la recepción del dato de superficie que se quiere analizar mediante la siguiente secuencia de comandos:

```
HSERIN [DAT]
```

```
IF DAT = "1" THEN
```

```
DAT = 5
GOTO MOTORES
ELSE
IF DAT = "2" THEN
DAT = 10
GOTO MOTORES
ELSE
IF DAT = "3" THEN
DAT = 15
GOTO MOTORES
ELSE
IF DAT = "4" THEN
DAT = 20
GOTO MOTORES
ELSE
GOTO INICIO
ENDIF
ENDIF
ENDIF
```

HSERIN es un comando se encarga de leer uno o más datos enviados desde el puerto serial en comunicación asíncrona desde la PC. Dicho dato es guardado en la variable DAT previamente establecida.

La variable entra en varios comparadores que definirán la superficie que va a ser analizada, en caso de que el dato enviado no sea ninguno de los establecidos, se reiniciará la recepción y comparación hasta obtener una respuesta afirmativa.

Por ejemplo, si el dato enviado desde la PC es 1, la variable DAT es renombrada tomando el valor de 5 y mediante el comando GOTO se dirigirá a la secuencia de motores.

3.5.2.2.2 Transmisión de datos y conversión.- Una vez ya establecida la superficie a analizar, se pasa a la secuencia llamada motores que presenta los siguientes parámetros:

```

MOTORES:
PAUSE 100
HSEROUT ["SI"]
DAT1 = DAT * 10
PAUSE 100
LOOP:
HSERIN [DAT2]
IF DAT2 = "5" THEN GOTO SEC1
GOTO LOOP

```

En esta la secuencia, se tiene una demora de 100ms antes de enviar mediante el comando HSEROUT la palabra SI, que será recibida por la PC a manera de aviso de que el dato antes enviado fue recibido con éxito.

El siguiente paso es esperar la respuesta de la PC, esta enviará un 5, que informa al microprocesador que ya puede enviar el dato de la conversión. La variable DAT es multiplicada por 10 para luego entrar en la secuencia SEC1 que veremos mas adelante.

```

CONV:
  ADCON0.0 = 1
  ADCIN 0,A1
  PAUSEUS 50
  HSEROUT [HEX A1]
  PAUSEUS 50
  RETURN

```

En la secuencia CONV, mediante el comando ADCIN, se lee el canal 0 del puerto A que esta conectado al palpador de la placa, y el dato leído se guarda en la variable A1 para acto seguido enviarlo en forma de hexadecimal por el puerto serial a la PC.

3.5.2.3 Mando de motores.- Si se toma como ejemplo la medición de una superficie de 5mm^2 , el valor en DAT será 5, dicho dato entrará a un FOR LOOP que establecerá el movimiento de los motores y la toma de datos del palpador 5 veces. En el mismo, se tiene otro FOR LOOP que tendrá como límite la variable DAT1, esto dará un nivel alto a MOT21 que es el encargado de mover a la derecha al motor en el eje Y, dando también un nivel bajo a todas las otras salidas de los motores para asegurar que se mueva solo en la dirección que se requiere. Como DAT1 es 50, y se tiene una demora de 100ms, el motor se moverá durante 5000ms, es decir, 5 segundos:

```
FOR J=1 TO DAT
  FOR I=1 TO DAT1
    HIGH MOT21
    LOW MOT22
    LOW MOT11
    LOW MOT12
    PAUSE 100
  NEXT I
```

Luego de que el motor cumple su movimiento, se manda por el puerto serial la palabra NO, para que la PC deje de graficar y espere a que el motor del eje X se mueva durante 1 segundo hacia la derecha poniendo en nivel alto a la variable MOT11 y todas las otras salidas en nivel bajo. Luego el motor del eje Y se mueve a la izquierda en el lapso establecido por la variable RES, esta no es mas que la multiplicación de DAT (5) por 1000, lo que da una demora de 5 segundos. Cuando se cumple esta demora, se pone en nivel bajo a todas las salidas de los motores y se enviará la palabra SI, para comunicar a la PC que la secuencia esta lista y que se prepare para graficar nuevamente:

```
HSEROUT ["NO"]
LOW MOT21
LOW MOT22
HIGH MOT11
```

```

LOW MOT12
PAUSE 1000
LOW MOT21
HIGH MOT22
LOW MOT11
LOW MOT12
RES = DAT * 1000
PAUSE RES
PORTC.1 = 0
PORTC.0 = 0
PORTC.2 = 0
PORTC.3 = 0
PORTC.4 = 0
PORTC.5 = 0
HSEROUT ["SI"]
PAUSE 100
NEXT J

```

Luego de que termino el FOR LOOP mayor (con límite DAT), mediante HSEIN se espera el dato 5 enviado por la PC, para avisar que se cumplió en su totalidad la lectura y graficación de datos y que los motores pueden volver a su posición original, esto se logra mediante los siguientes comandos:

```

LOOP1:
  HSERIN [DAT2]
  IF DAT2 = "5" THEN GOTO SEC2
GOTO LOOP1
SEC2:
LOW MOT21
HIGH MOT22
LOW MOT11
HIGH MOT12
PAUSE RES
DAT = 0

```

```

PORTC.1 = 0
PORTC.0 = 0
PORTC.2 = 0
PORTC.3 = 0
PORTC.4 = 0
PORTC.5 = 0
PAUSEUS 50
GOTO INICIO

```

Como se puede observar, los motores funcionan durante la demora RES (5 segundos), y luego se borran todos los registros tanto de motores como de variables, de manera que el sistema queda listo para regresar al inicio y medir otra superficie.

3.6 Diseño del software de control en la PC.

3.6.1 Introducción teórica sobre el programa labview v8.1.-Al estudiar profundamente la configuración de los sistemas de adquisición de datos modernos, basados en equipos PC, se aprecia que una de las partes que componen dichos sistemas, es el software quien controla y administra los recursos del computador, presenta los datos, y participa en el análisis. El software es un tópico muy importante que requiere de especial cuidado.

Para los sistemas de adquisición de datos se necesita de un software de instrumentación, que sea flexible para futuros cambios, y preferiblemente que sea de fácil manejo, siendo lo mas poderoso e ilustrativo posible.

Programas y lenguajes de programación que cumplan con lo dicho existen en gran número en el mercado actual, como por ejemplo el Visual Basic, el C, el C++, el Visual C++, Pascal, LabWindows CVI, Labview, y muchos otros confeccionados específicamente para las aplicaciones que los necesiten.

3.6.2 ¿Qué es Labview?.- LabVIEW (Laboratory Virtual Instrument Engineering Workbench) es un lenguaje de programación gráfico para el diseño de sistemas de adquisición de datos, instrumentación y control. Labview permite diseñar interfaces de usuario mediante una consola interactivo basado en software. Usted puede diseñar

especificando su sistema funcional, su diagrama de bloques o una notación de diseño de ingeniería.

Labview es a la vez compatible con herramientas de desarrollo similares y puede trabajar con programas de otra área de aplicación, como por ejemplo Matlab. Tiene la ventaja de que permite una fácil integración con hardware, específicamente con tarjetas de medición, adquisición y procesamiento de datos (incluyendo adquisición de imágenes).

3.6.2.1 Ventajas del programa LabView.- Para elaborar los algoritmos de control y toma de datos en la elaboración de un rugosímetro digital con adquisición de datos, se consideró que el lenguaje más apto es el LabVIEW, y las razones son varias:

- Es muy simple de manejar, debido a que está basado en un sistema de programación gráfica, llamada lenguaje G.
- Es un programa enfocado hacia la instrumentación virtual, por lo que cuenta con numerosas herramientas de presentación, en gráficas, botones, indicadores y controles, los cuales son muy esquemáticos y de gran elegancia. Estos serían complicados de realizar en bases como c++ donde el tiempo para lograr el mismo efecto sería muchas veces mayor.
- Es un programa que permite pasar las aplicaciones entre diferentes plataformas como Macintosh y seguir funcionando.

3.6.2.2 Aplicaciones de LabVIEW.- Labview tiene su mayor aplicación en sistemas de medición, como monitoreo de procesos y aplicaciones de control, un ejemplo de esto pueden ser sistemas de monitoreo en transportación, Laboratorios para clases en universidades, procesos de control industrial. Es muy utilizado en procesamiento digital de señales (wavelets, FFT, Total Distorsion Harmonic TDH), procesamiento en tiempo real de aplicaciones biomédicas, manipulación de imágenes y audio, automatización, diseño de filtros digitales, generación de señales, entre otras, etc.

3.6.3 Programación gráfica con Labview.- Cuando se diseña con Labview se trabaja siempre bajo algo denominado VI, es decir, un instrumento virtual, se pueden crear VI a partir de especificaciones funcionales que se requieran. Este VI puede

utilizarse en cualquier otra aplicación como una subfunción dentro de un programa general. Los VI's se caracterizan por ser un cuadrado con su respectivo símbolo relacionado con su funcionalidad, tener una interfaz con el usuario, tener entradas con su color de identificación de dato, tener una o varias salidas y por su puesto ser reutilizables.

3.6.3.1 Instrumentos virtuales.- En el ambiente de trabajo de Labview existen dos paneles, el panel frontal y el panel de programación ó diagrama de bloques (Fig 3.86); en el panel frontal se diseña la interfaz con el usuario, donde estarán ubicados todos los indicadores y controles que el usuario podrá ver cuando el programa este en funcionamiento. Por ejemplo botones, perillas, gráficas, etc.

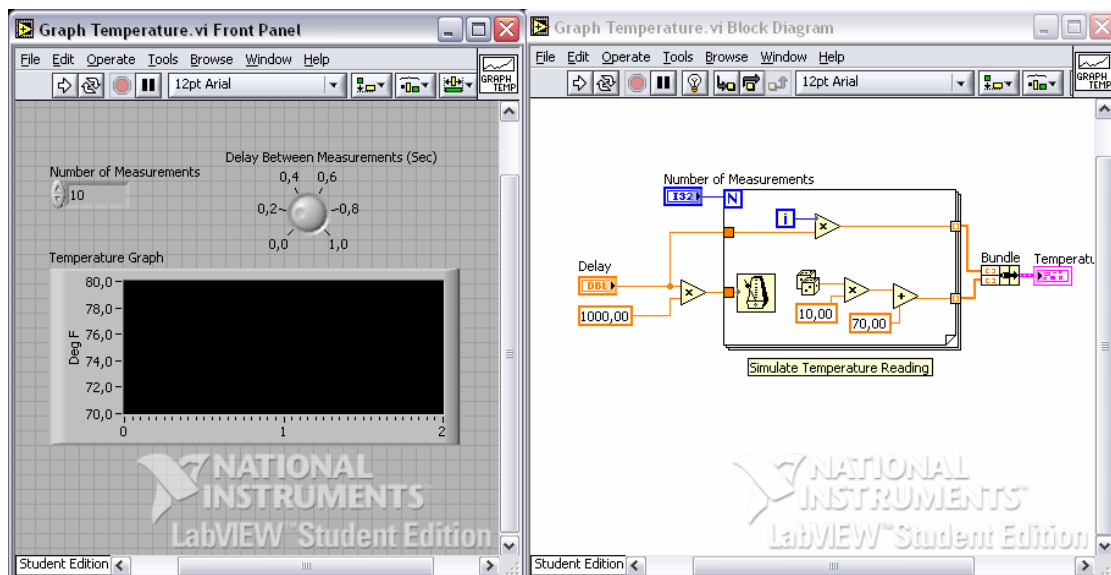


Fig86. Panel frontal y Panel de programación.

En el panel de programación se relacionan los elementos utilizados en la interfaz mediante operaciones que determinan en sí como funciona el programa o el sistema, exactamente es la parte donde se realizan las especificaciones funcionales. En él se muestra el programa en código gráfico G.

Se usan en este diagrama estructuras de programación, y flujo de datos entre las diferentes entradas y salidas, a través de líneas.

En este las subrutinas son mostradas como iconos de cajas negras, con unas entradas y unas salidas determinadas, donde en el interior se cumple una función específica. El flujo se aprecia, como se dibujaría en un bosquejo de sistemas, cuando se habla de teoría de sistemas, donde cada subsistema se representa como un cuadro con entradas y salidas.

Todos los indicadores y controles ubicados en el panel frontal están respaldados por un terminal de conexión en el diagrama de bloques tal como si se tuviera un tablero de control de una máquina o un avión, donde por el frente se ven los indicadores y por el lado posterior se aprecian todos los cables y terminales de conexión.

El icono de conexión se usa para utilizar el programa creado como subrutina en otro programa, donde el icono será la caja negra, y las entradas son las conexiones a los controles del programa subrutina, y las salidas son las conexiones a los indicadores del mismo subprograma.

Al crear el icono, se conecta a través del alambre de soldadura a los indicadores y controles en la forma que se desee que se distribuyan las entradas y salidas en la caja negra, tal como en un circuito integrado algunos pines corresponden a alguna función en él.

La idea es crear un sistema de programación modular, donde cada rutina creada llame otras rutinas, y estas a su vez otras de menor nivel, en una cadena jerárquica con cualquier límite deseado. Así cuando se use un módulo, no se requiere saber como funciona interiormente, simplemente solo basta conocer sus entradas y salidas para ser así usado.

El flujo de datos va de izquierda a derecha en el panel de programación y esta determinado por las operaciones o funciones que procesan los datos. Es fácil observar en el panel de programación como se computan los datos en cada parte del programa cuando se realiza una ejecución del programa paso a paso.

En Labview las variables se representan mediante una figura tanto en el panel frontal como en el panel de programación, de esta forma se puede observar su respuesta en

la interfaz del usuario y en el flujo de datos del código del programa. Otros objetos como gráficas y accesos directos a páginas web cumplen estas mismas condiciones.

3.6.4 Tipos de variables y datos numéricos.

3.6.4.1 Número de bits de un número.- El computador posee una memoria compuesta de una gran lista de números, los cuales son llamados bytes, que son un conjunto de unos o ceros, llamados bits. Cada byte se compone de ocho bits los cuales pueden representar un número de 0 a 255.

Para poder almacenar números mayores se requiere de más bytes, donde se tengan 16 o 32 bits. Este número se relaciona con el número de bits con los que puede trabajar el microprocesador del computador, en cuanto a la velocidad de operación. Además un número de más bits ocupa mayor espacio en memoria.

Tomando en cuenta esto, si se desea manejar el 258, y se usa un byte (8 bits), el número obtenido es 255, perdiendo toda la exactitud. Se deben usar 16bits.

3.6.4.2 Signo en el número.- Como se tiene un código binario, hay métodos para dar el carácter de positivo o negativo a un número, dejando bits que representen el signo. Cuando se opera con números con signo el método es diferente a como se hace con números sin signo.

3.6.4.3 Números fraccionarios.- Igual que con el problema del signo, se requieren de algunos métodos para representar la coma en un código binario, y las operaciones también varían. De hecho se requiere de muchos más cálculos para un microprocesador para sumar dos números de coma flotante (que posean coma, fraccionarios), que para sumar dos enteros sin signo. Para esto el microprocesador se vale del coprocesador matemático, que hace operaciones de coma flotante a gran velocidad. Los números de coma flotante dependen del número de bits, para tener una mayor exactitud.

Según lo anterior hay números de tipo entero 'I' de 8, 16 y 32 bits, de tipo sin signo (unsigned U) de 8, 16, 32, o de coma flotante de tipo simple (SGL 16), doble (DBL

32), y Extendido (EXT 64 bits). Igualmente números complejos simples, dobles y extendidos. El tipo de número se aprecia en la terminal de conexión de los controles o indicadores, pues aparece inscrito, y el color de las conexiones de punto flotante son anaranjadas o rojas, mientras que en los enteros y sin signo son azules. Se recomienda usar datos de menor número de bits, siempre y cuando no se pierda precisión, para que no se ocupe mucha memoria (fig3.87).

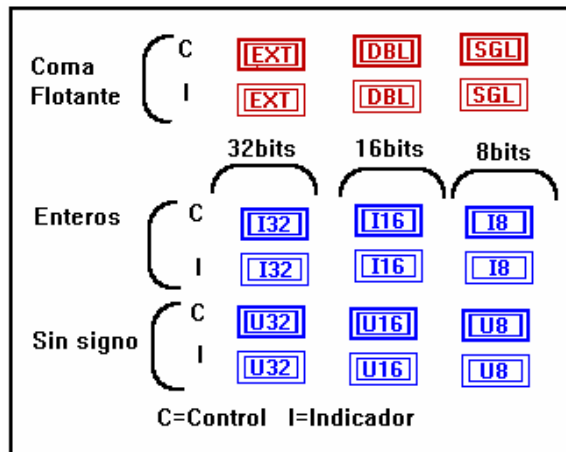


Fig3.87. Tipos de representación de los números.

3.6.4.4 Datos booleanos y alfanumericos.- Los datos booleanos también tienen su tipo de conector. Para booleanos el color de las conexiones y los cables es de color verde, y para las de tipo alfanumérico son de color rosado (fig3.88).

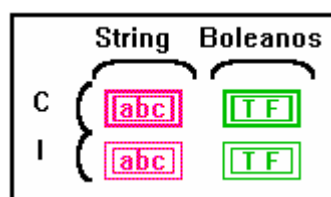


Fig3.88 Datos booleanos.

3.6.4.5 Matrices.- Las matrices son conjuntos de datos de una misma especie. Para crear una matriz se ubica en el panel frontal un cuadro de matriz (Array o arreglo) sacado del menú ARRAY & CLUSTER, y dentro se ubica el control o indicador que se mostrará. Se puede estirar el cuadro para que muestre varios datos pertenecientes a la misma matriz.

Si se estira el display lateral se aumenta el número de dimensiones. El conector será

uno solo para la matriz con todos los datos, y se diferencia de los otros conectores por tener el tipo de datos dibujado entre [], en lugar de un recuadro, así se puede poseer una matriz de cualquier clase de número, sea doble, alfanumérico, boleano, etc.

Las líneas o cables que conducen matrices son más gruesos y aumentan de espesor según sea el número de dimensiones que manejen.

3.6.5 Paletas o menús de trabajo.-En la programación se tienen dos menús importantes que se denominan paleta de controles y paleta de funciones, a continuación se detallan los componentes más utilizados de cada una de ellas.

3.6.5.1 Paleta de controles.- En la paleta de controles (fig3.89) aparecen todos los tipos de controles e indicadores que se pueden colocar en el panel frontal, como son:

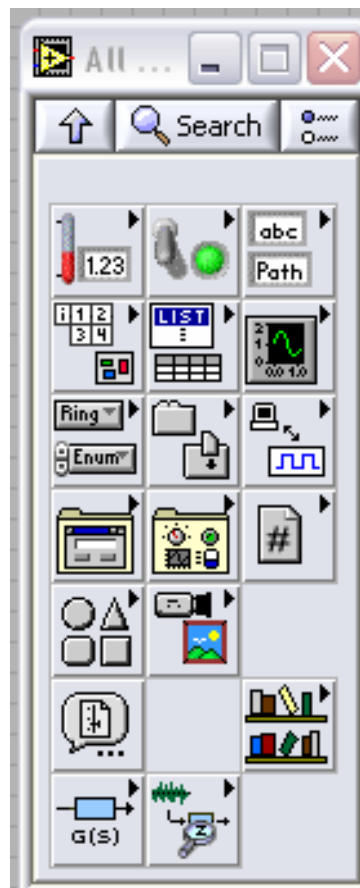


Fig3.89. Paleta de controles

3.6.5.1.1 Numéricos.- Permiten la entrada y salida de datos y valores medibles de tipo numérico, ya sea en un número real, enteros, naturales positivos como pueden ser observados en la fig3.90.

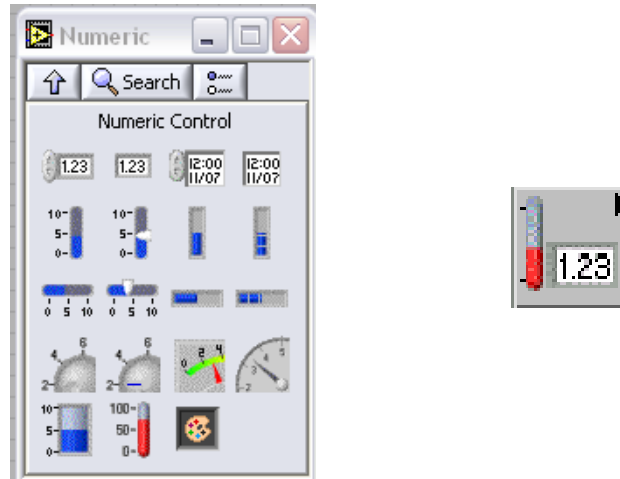


fig 3.90 Controles numéricos Numéricos

3.6.5.1.2 Booleanos.- Permiten la salida y la entrada de datos de tipo discreto, on-off, como es el caso de los pulsadores, swiches, led's indicadores como lo indica la fig3.91.

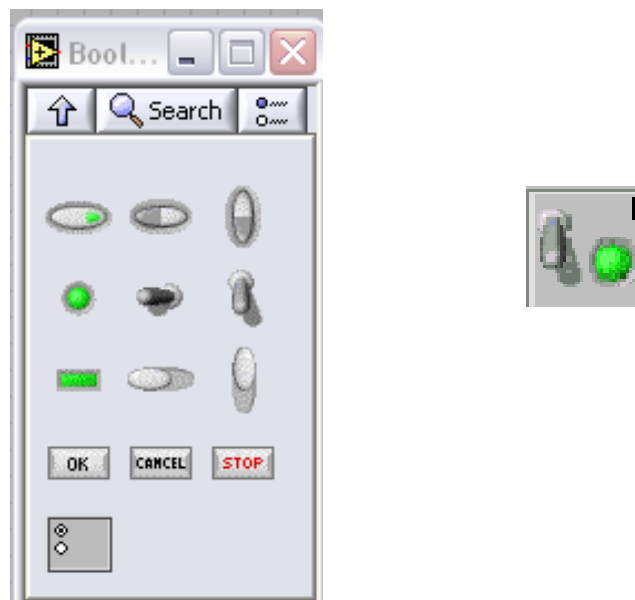


fig 3.91 Controles booleanos

3.6.5.1.3 String & Path.- Permite entrar y sacar datos de tipo alfanumérico, vistos en un indicador o control, así como también, el manejo de archivos exteriores pueden ser vistos en la fig3.92.

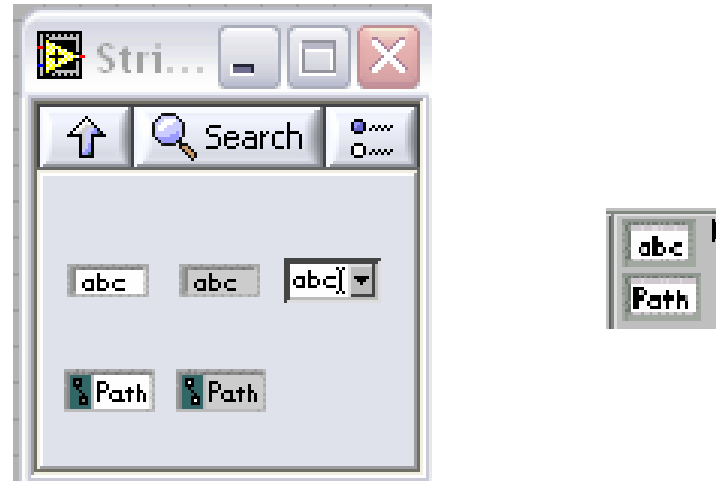


fig 3.92 Controles de String y Path

3.6.5.1.4 List & Table.- Son controles e indicadores que presentan listas de opciones donde el ítem seleccionado se entrega como un valor al programa o en una tabla que también puede cumplir las dos funciones. Su representación en el programa esta indicada como lo indica la fig3.93

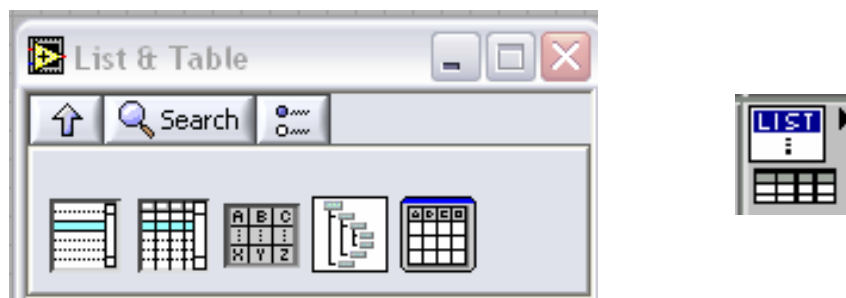


fig 3.93 Controles List y Table

3.6.5.1.5 Array & Clusters.- Permite agrupar datos para formar matrices ya sean de entrada o salida. Estas matrices pueden ser de tipo numérico, o de tipo booleano. También se pueden agrupar datos de diferentes tipos de control o de diferentes tipos de indicador, en un cluster, el cual es una agrupación que posee una sola terminal en el diagrama de bloques, semejante a un conector de un computador, el cual siendo un

solo conector lleva muchas líneas que llevan diferentes señales (fig 3.94). En las matrices todas las señales son del mismo tipo.



fig 3.94 Controles de array y clusters

3.6.5.1.6 Graph.- Controles e indicadores de gráficas. Pueden ser gráficas de barrido, graficas XY, o de tonos de colores. Se lo encuentra en el panel de control como indica la fig 3.95



fig 3.95 Controles de gráficos

3.6.5.1.7 Ring & Enum.- Controles útiles en el manejo de archivos, ya que se pueden hacer llamados tanto de arreglos como de imágenes (fig3.96).



fig3.96 Controles de llamado

3.6.5.1.8 Classic Controls.- En este icono se puede encontrar controles de versiones anteriores (fig. 3.97)

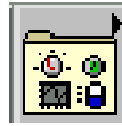


fig 3.97 Controles clásicos

3.6.5.1.9 Decorations.- Además de poderse ubicar los controles e indicadores presentados en los menús anteriores, también se pueden usar controles editados por el programador, como por ejemplo el dibujo de una bomba, o un pistón neumático, mediante el control que presenta la fig3.98

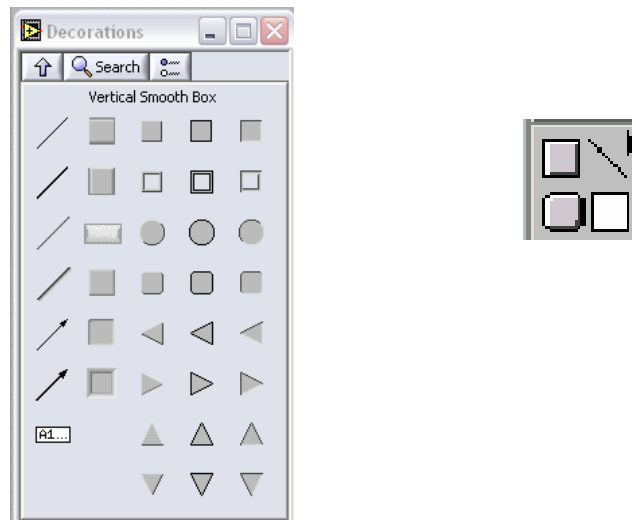


fig3.98 Control de decoraciones

3.6.5.1.10 Error Cluster.- Controles de entrada y salida, para parámetros de algoritmos manejadores de errores (fig3.99).

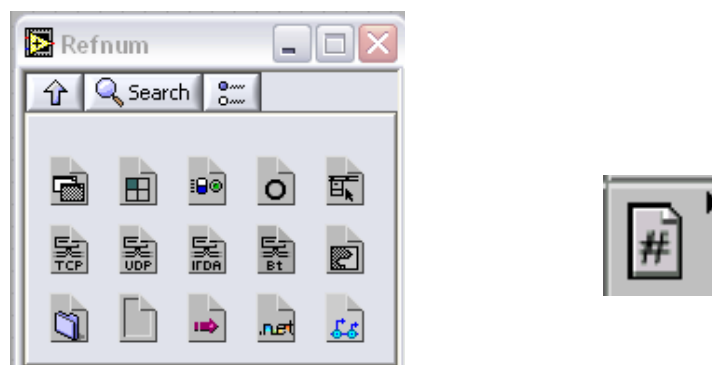


fig3.99 Controles de entrada y salida

3.6.5.1.11 I / O .-Útiles para comunicación con ni-DAQmx y todos sus controles, sus opciones las encontramos en la fig3.100

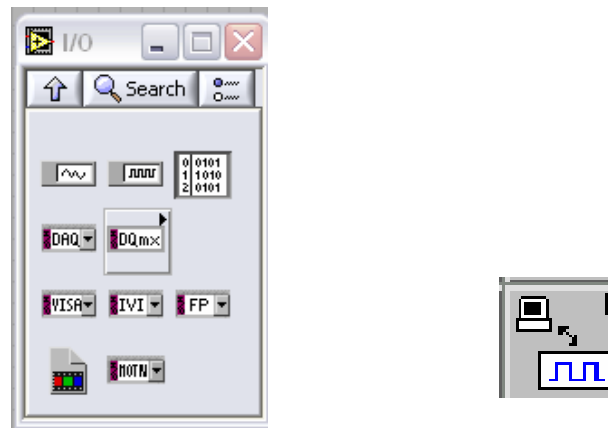


fig3.100 Controles para comunicación

3.6.5.2 Paleta de funciones.- Como se puede observar en la fig3.101, la paleta de funciones cuenta con muchas opciones con las cuales se podrá trabajar en la parte de la programación, a continuación se detalla las más importantes de estas.



Fig3.101 Paleta de funciones

3.6.5.2.1 Structures.- Contiene las estructuras básicas de programación como son las secuencias, los casos, los ciclos For-Next y Mientras, las variables de tipo global y local, y las constantes de todo tipo, como son las numéricas, las alfanuméricas, las booleanas, y algunos números especiales. A structures se puede encontrar como muestra la fig3.102

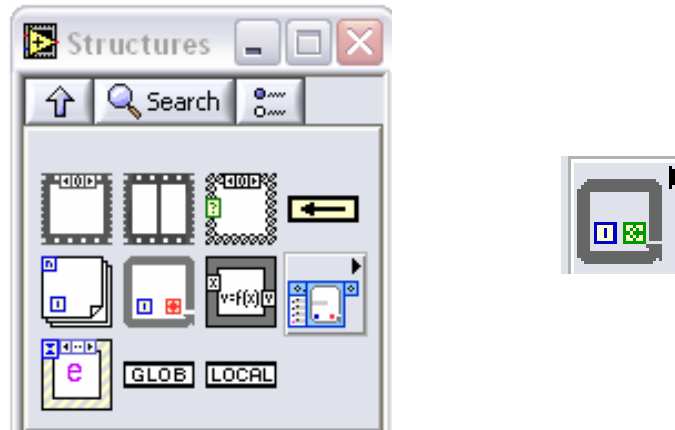


fig 3.102 Estructuras

3.6.5.2.2 Numeric.- Presenta las operaciones básicas aritméticas como son suma, resta, multiplicación, números al azar, valor absoluto, etc. como muestra la fig 3.103. Para ver la función de cada una usar la ventana de Help <ctrl-H>.

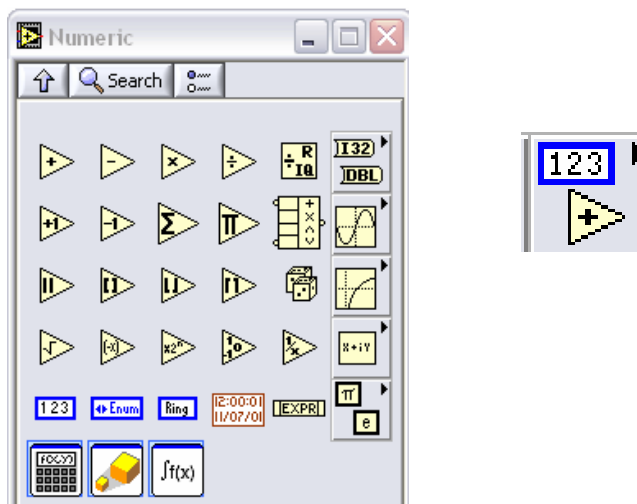


fig 3.103 Numéricos

3.6.5.2.3 Trig & Log.- Presenta funciones trigonométricas y logarítmicas como se puede observar en la fig 3.104.

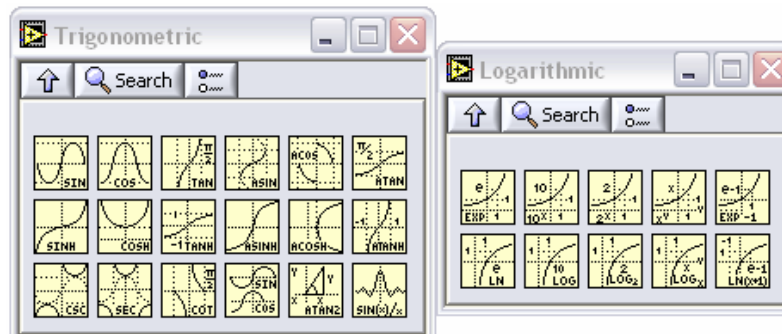


fig 3.104 Funciones trigonométricas y logarítmicas

3.6.5.2.4 Boolean.- Presenta los íconos de or, and, not, etc, además de un control en el que se puede realizar distintas operaciones, contando también con constantes booleanas, tal como se puede observar en la figura 3.105

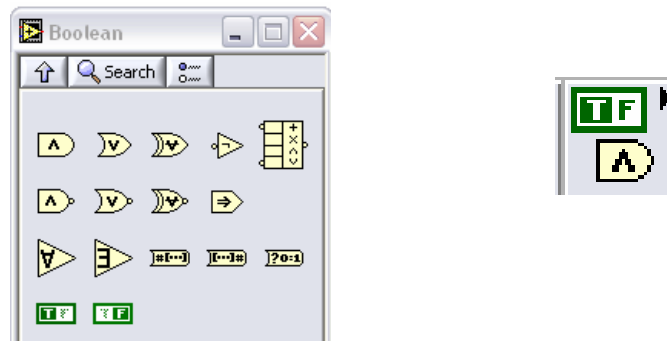


fig 3.105 Opciones booleanas

3.6.5.2.5 Comparison.- Funciones de comparación que devuelven un valor de verdadero o falso según se cumpla dicha comparación (fig 3.106).

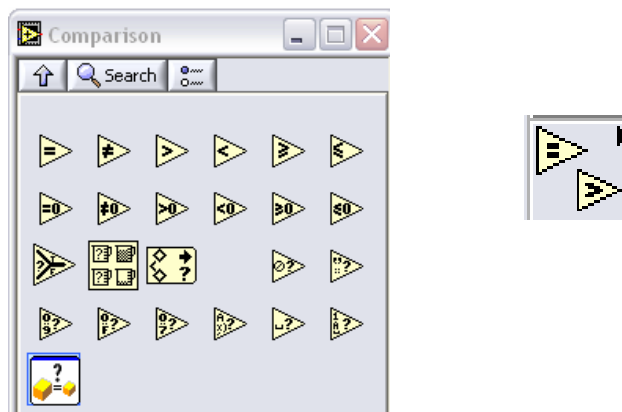


fig 3.106 Comparaciones

3.6.5.2.6 Conversion.- Conversiones de tipos de variables, de un formato a otro, por ejemplo convertir un número a otro que ocupe 32 bits en memoria, o convertir un número a una matriz de booleanos cuya representación en binario corresponda al número. Los tipos de conversiones existentes se pueden observar en la fig 3.107:

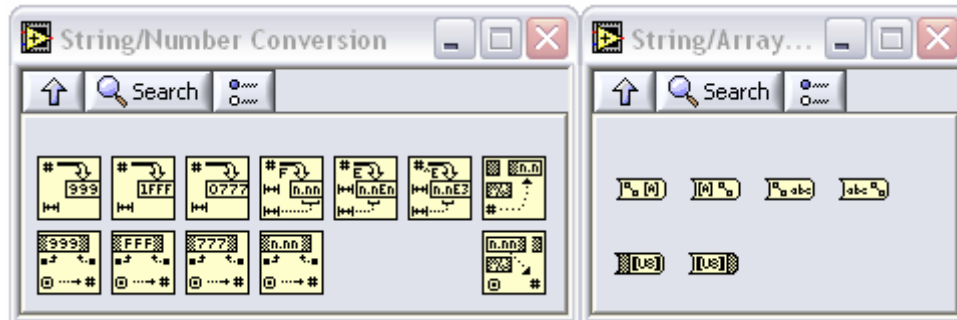


fig 3.107 Diferentes Conversiones

3.6.5.2.7 String.- Presenta herramientas para manipular cadenas de caracteres. Por ejemplo convertir todos los caracteres a mayúsculas, o reportar el valor de la longitud de la cadena. Estas opciones pueden observarse en la fig 3.108

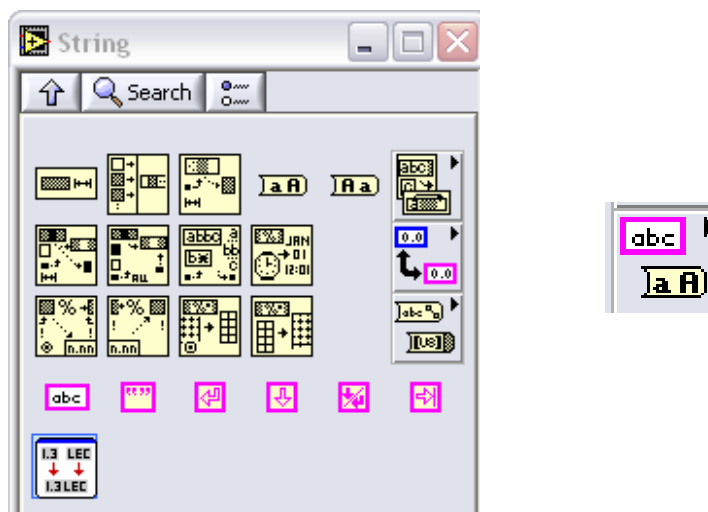


fig 3.108 Presentación de Strings

3.6.5.2.8 Array & Cluster.- Maneja las herramientas para el uso de matrices y agrupaciones. Como ejemplo puede ser agrupar un conjunto de cables en uno solo par manipular menos líneas. Las múltiples opciones que tienen los clustes se pueden ver en las fig3.109 a continuación:

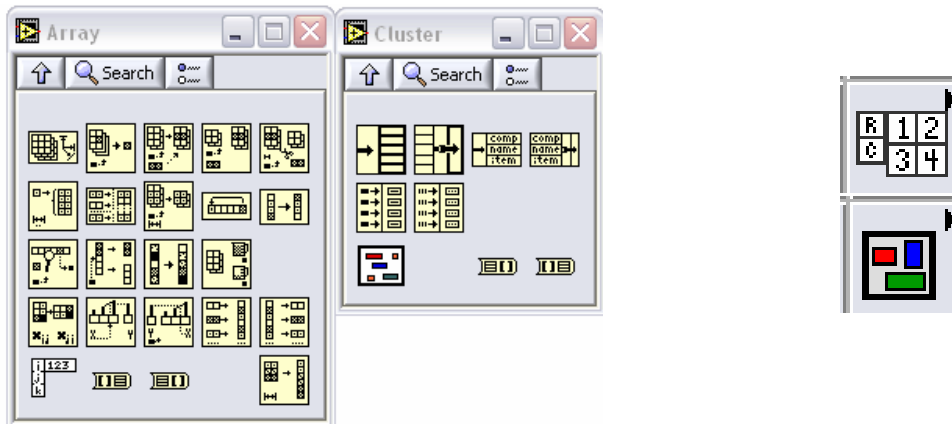


fig 3.109 Paneles de arreglos y clusters

3.6.5.2.9 File I/O.- Para el manejo de archivos y almacenamiento de información en disco. Sus opciones pueden ser vistas en la fig 3.110

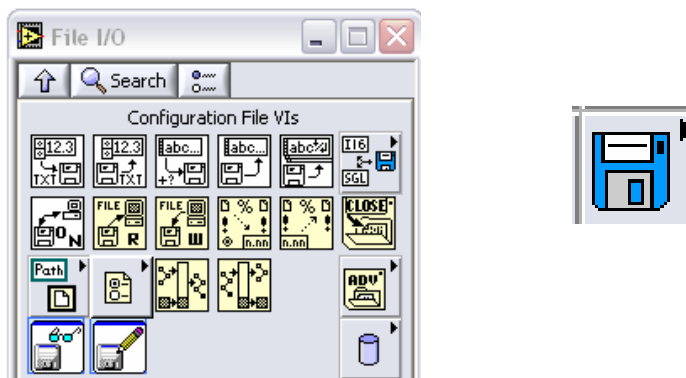


fig 3.110 Menú de archivos de entrada y salida

3.6.5.2.10 Time & Dialog.- Reportadores de tiempo, esperas, fechas, y cuadros que dan anuncios (fig 3.111)

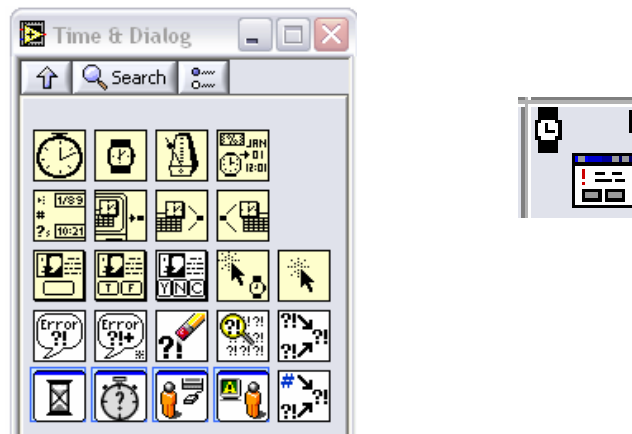


fig 3.111 Controles de tiempo y errores

3.6.5.2.11 Vi.- Para llamar bloques creados como rutinas. La representación de su icono se puede observar en la fig 3.112



Fig3.112 Llamado a subVI's

3.6.5.2.12 Analyze.- Funciones avanzadas de procesamiento de señales, estadísticas, álgebra lineal, filtros, regresión y otras que requieren de un buen entendimiento matemático. Sus múltiples opciones pueden ser observadas en la fig 3.113

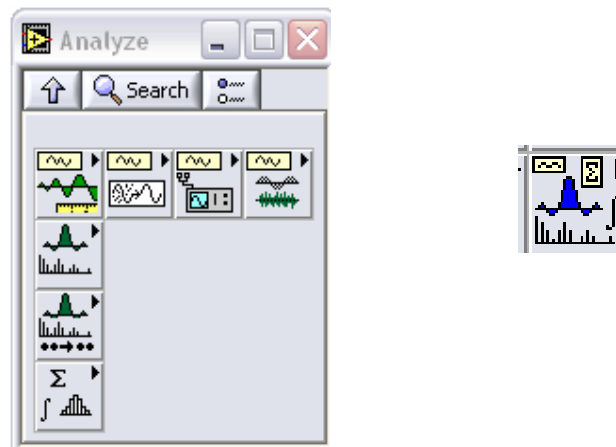


fig 3.113 Panel de análisis

3.6.5.2.13 Ni Measurements.- Para la adquisición de datos, lectura y escritura de datos a las tarjetas insertables, toma y control de señales análogas y digitales, control de los circuitos contadores que hay en algunas tarjetas y, adquisición de imágenes y video (fig 3.114).

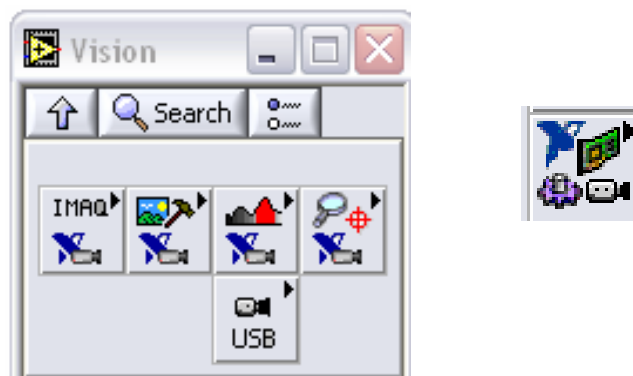


fig 3.114 Controles NI-DAQ

3.6.5.2.14 Instrument I/O.- Comunicación con instrumentos medidores a través de puertos GPIB, serial o VISA. Sus múltiples opciones se observan en la fig 3.115

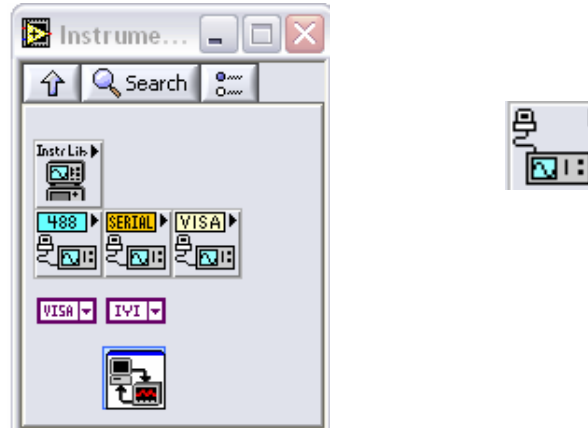


fig 3.115 Instrumentos de entrada y salida

3.6.5.2.15 Network.- Para la comunicación de computadores en red, y enlace entre diferentes aplicaciones, como es el caso del DDE, para actualizarlas simultáneamente los hechos van ocurriendo (fig 3.116). Otros parámetros son los de comunicación TCP y UDP para comunicación en red. Todo esto requiere de un aprendizaje especial.

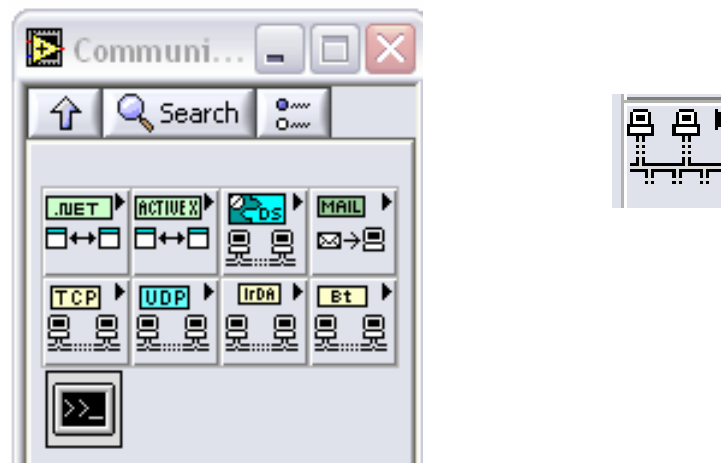


fig 3.116 Comunicación de red

3.6.6 Estructuras y elementos de programación.- Para realizar un programa dentro de cualquier lenguaje se requiere de conocer el manejo de las estructuras que gobiernan un algoritmo. En el LabView como lenguaje también cuenta con estas.

Las estructuras en LabView son:

- Los ciclos While
- Los ciclos For loop
- Los cuadros de casos
- Las secuencias

Otros elementos de programación son las variables, que pueden ser de tipo global o local, y los cuadros de fórmula.

3.6.6.1 Los Ciclos While.- Sirven para hacer que una secuencia de instrucciones se repita una cantidad de veces, siempre y cuando una afirmación sea verdadera. En LabView se ejecutarán las funciones que se encuentren dentro del cuadro de ciclo, tomando los valores que quedaron almacenados en la frontera de entrada, y sacando los resultados a la frontera de salida.

Por ejemplo si se desea contar a partir de un número 'a', durante una cantidad de veces 'b', e ir mostrando el número de conteo en un indicador 'd', y ver el último número contado en 'c', el programa sería el siguiente.

El término 'i' en el ciclo es un contador que se incrementa una unidad cada vez que se repite el ciclo. La flecha circular es el parámetro que al recibir un valor de true (verdadero), permite repetir el ciclo, y al recibir un falso, lo detiene para que el dato que haya en la frontera de salida valla al indicador c.

En la figura 3.117 se presenta un ejemplo de utilización de While tanto en la visualización del panel principal como en el panel de programación

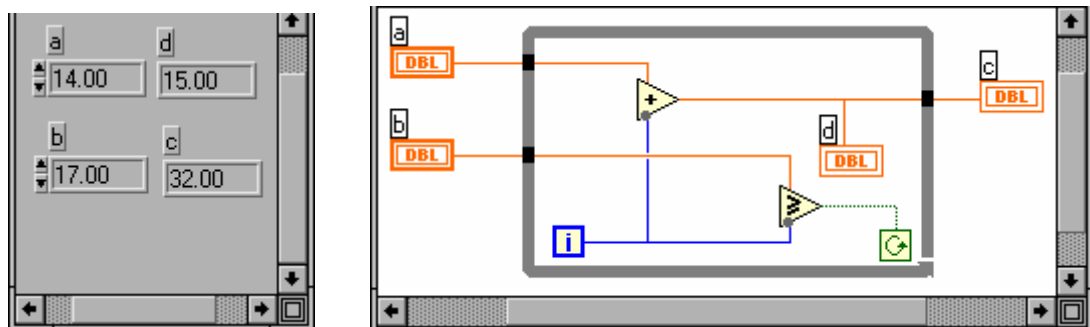


fig 3.117 Presentación de un ejemplo de While

Los datos a, y b solo llegan una vez a frontera de entrada y allí quedan almacenados en un buffer para ser usados todas las veces que el ciclo repita. Estos datos siempre serán iguales.

En el programa se sumará en cada loop el valor de 'a' con el contador que en cada iteración es mayor en uno.

El dato se mostrará en 'd', y se llevará a la frontera de salida, donde se almacena hasta que termine el ciclo. En la iteración siguiente un nuevo dato llega a la frontera borrando el anterior, así cuando el loop para, solo el último valor pasa a 'c'.

Constantemente se evalúa si el número 'b' es mayor o igual al contador. Cuando este contador alcanza a b, la comparación se vuelve falsa y el ciclo se detiene.

3.6.6.2 Ciclo For Loop.- Se comporta similar al ciclo While. Este hace un número definido de iteraciones el cual esta dado por el valor que se coloca en el parámetro 'N'. Este siempre debe ser definido, pues de no suceder así el programa no corre.

Tiene la desventaja respecto al ciclo while de tener que cumplir todas las iteraciones para terminar, mientras que en el while, se termina dependiendo de una condición, por tanto se puede crear un algoritmo que cuando detecte un error termine el ciclo. Mientras que el for-loop es un ciclo ciego, el while siempre se está chequeando. Se puede observar un ejemplo en la figura 3.118 a continuación:

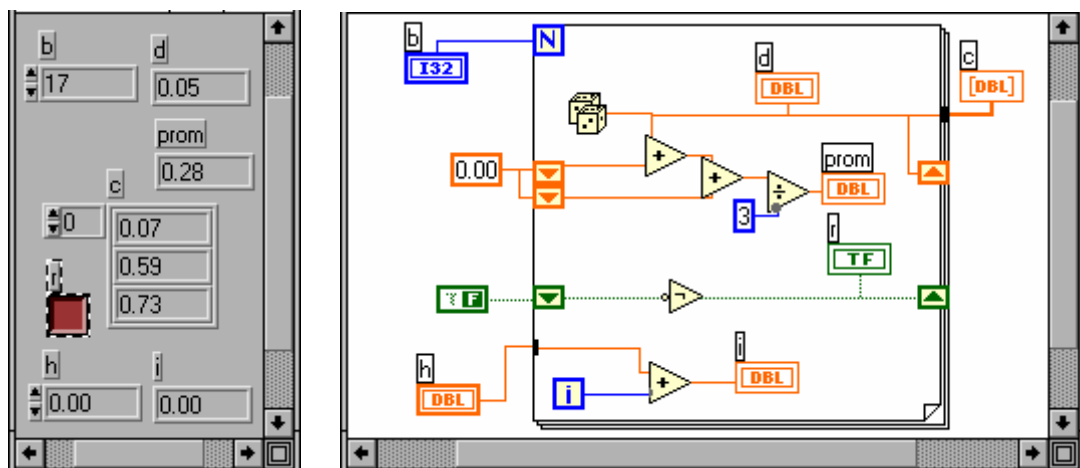


Fig3.118 Ejemplo de un ciclo For Loop

El ciclo For-loop también cuenta con un elemento ‘i’ que sirve de contador para decir en que ciclo va.

3.6.6.3 Cuadros de casos ‘CASE’.- Es una estructura de comparación y ejecución condicionada donde de acuerdo a algún parámetro se realizan las operaciones de un cuadro u otro. Si el parámetro de condición es del tipo verdadero-falso cuando éste es verdadero se ejecuta un contenido, y cuando es falso se ejecuta otro. De esta forma solo son posibles dos opciones de ejecución.

Si el parámetro es un número, se ejecuta un cuadro cuyo número de identificación corresponde al valor de entrada. En este caso puede haber tantas opciones de ejecuciones como se desee como muestra la fig 3.119.

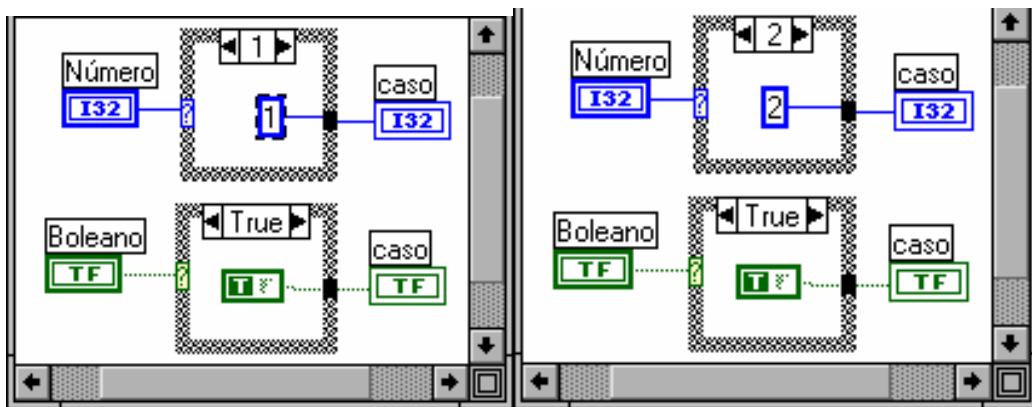


fig 3.119 Ejemplo de utilización de un Case

Para agregar un cuadro de caso cuando se usa un parámetro de selección numérico, solo basta seleccionar el pop-up menú de la estructura, dando clic con el botón derecho y seleccionando “Add Case After” para un caso de número siguiente, o “Add Case Before”. Dentro de este pop menú, se encuentran otros parámetros de control de estas estructuras. Para ver el contenido de cada caso, solo basta seleccionarlo con las flechas del indicador del caso.

3.6.6.4 Las Secuencias.- Como el LabView es un lenguaje de tipo multiproceso, puede ejecutar varias partes del programa simultáneamente. Además las funciones se van operando cuando llegan todos los parámetros de entrada de cada una lo que no

da mucha certeza de que función se realiza primero. Pero si por alguna razón se desea que un conjunto de operaciones se realice antes que otro, se puede agregar una estructura de secuencias, la cual ejecuta el contenido del primer cuadro, luego el del segundo, y así sucesivamente tal como en una cinta de fotos para cine, cada foto sigue a la otra. Para agregar un cuadro adicional tal como en las estructuras case, se logra por medio del pop-up menú en el borde del marco, Add Frame como muestra la fig 3.120 a continuación.

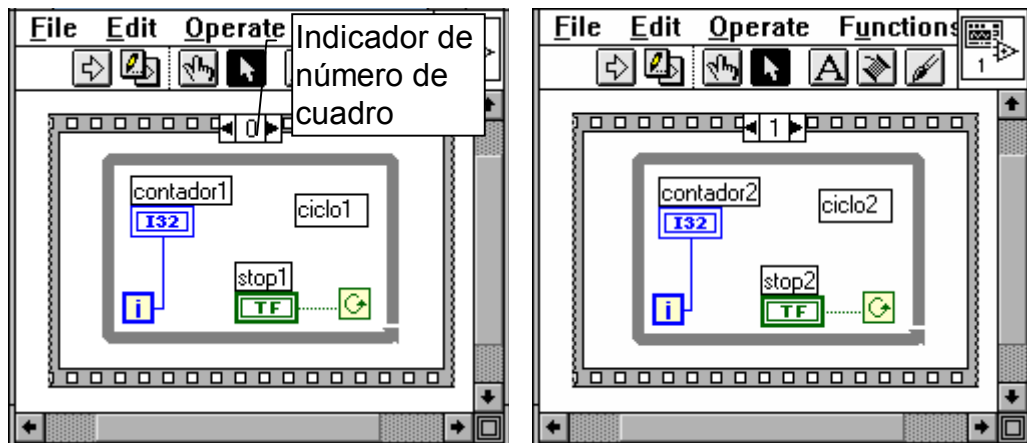


fig 3.120 Ejemplo de una secuencia

3.6.7 Variables.- Los parámetros de entrada para una función pueden lograrse a través de controles, indicadores, e incluso variables. Las salidas pueden usarse para mostrarse en indicadores o simplemente para ser almacenadas en los mismos elementos los cuales son las variables. Estas se usan para almacenar datos y pueden ser de dos tipos según su uso, de tipo Local o de tipo Global.

3.6.7.1 Variables Locales.- Son variables asociadas a algún control o indicador dentro de un programa VI, en el cual son usadas. Cuando se escribe en una de éstas, el contenido del indicador o control cambia. El uso de estas variables facilita la visualización en el diagrama cuando se va a acceder varias veces un mismo dato, puesto que evita llenar de cables conductores la pantalla, que conduzcan el valor desde el control al lugar requerido. Son una forma adecuada de escribir sobre un control, desde el algoritmo. Las variables locales solo son entendidas por el programa VI que las posee, ninguna subrutina entiende el contenido de éstas, ni ningún programa VI diferente.

Si se desea leer un valor de la variable seleccionar en el pop-up menú CHANGE TO READ. El icono de la variable mostrará las paredes laterales más gruesas que en el caso de la variable a la que se escribe. Para escribir seleccionar en el pop-up menú CHANGE TO WRITE.

En el ejemplo de la figura 3.121 se aprecia como para no tener que llevar cables desde el terminal del botón de stop al otro ciclo, para detenerlo simultáneamente, simplemente se lee el dato de una variable asociada.

En el primer cuadro se muestra un icono de una variable para ser escrita con un dato, pero como no está conectado a ninguna fuente de valores, se genera un error que no permite ejecutar el programa, lo que se puede notar por la flecha de ejecución rota.

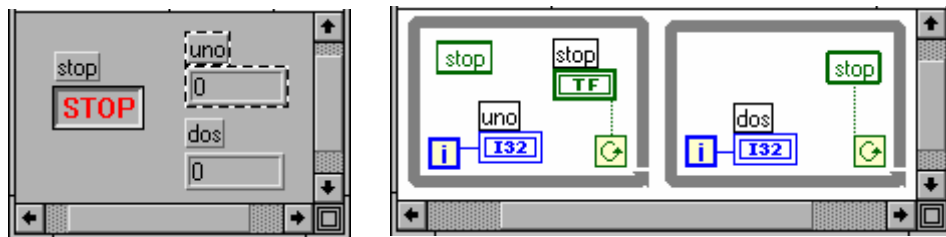


Fig3.121 Ejemplo de utilización de variables locales

3.6.7.2 Variables Globales.- La diferencia con las variables locales radica en que estas pueden ser entendidas por cualquier programa y/o subrutina VI, y pueden ser actualizadas por los mismos.

Estas se almacenan en un archivo diferente de extensión **.GLB** que consta únicamente de un panel frontal donde se encuentran todas las variables asociadas a dicho archivo, lo que quiere decir que en un archivo se pueden guardar numerosas variables.

Para crear una variable global, se selecciona del menú de estructuras y constantes. Luego con el pop-up menú se da la orden de abrir el panel frontal de esta variable, y allí se colocan todos los indicadores y controles que almacenan los datos deseados. Posteriormente se graba como cualquier otro programa VI, pero con extensión **.DBL**.

Con la variable creada solo es seleccionar en el pop-up menú del icono de la variable con SELECT ITEM, el valor al que se lee o escribe. Para colocar en el diagrama otra variable global del mismo dato, ahora se hace por medio de la opción VI, en el menú de funciones, tal como si se fuera a usar una subrutina ya creada.

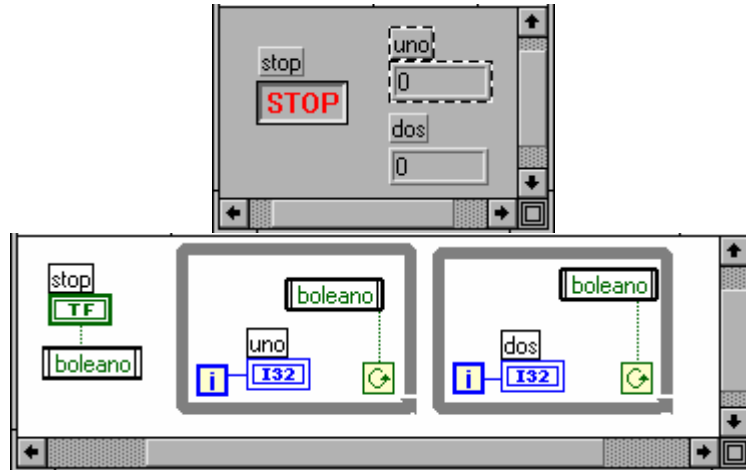


Fig3.122 Ejemplo de la utilización de una variable global

En el ejemplo de la figura 3.122 el dato del botón stop se escribe a una variable global, que se lee en otros lados del programa. También puede ser leído en otro subprograma.

Los cambios entre lectura y escritura, son idénticos al procedimiento con las variables locales. Igualmente se aprecia que el icono en estado de lectura es diferente al de escritura.

3.6.8 Cuadros de fórmula.- Cuando se realizan operaciones matemáticas complejas donde hay muchos cálculos distintos, tener un icono por cada operación puede ser complejo e implicar muchos cables y conexiones, que dificultan el entendimiento del diagrama y alargan el tiempo de programación.

Un mejor camino es tomar todas esas funciones y juntarlas en un cuadro de fórmula, donde se escribe la operación de una forma textual.

Al cuadro se le agregan unos conectores de entrada y salida de datos, con el nombre de los parámetros inscritos, y de allí se alambran los conductores. Para agregar entradas o salidas hacerlo por medio del pop-up menú **Add Input** o **Add Output**. Como regla después de cada función se debe colocar un punto y coma “;”.

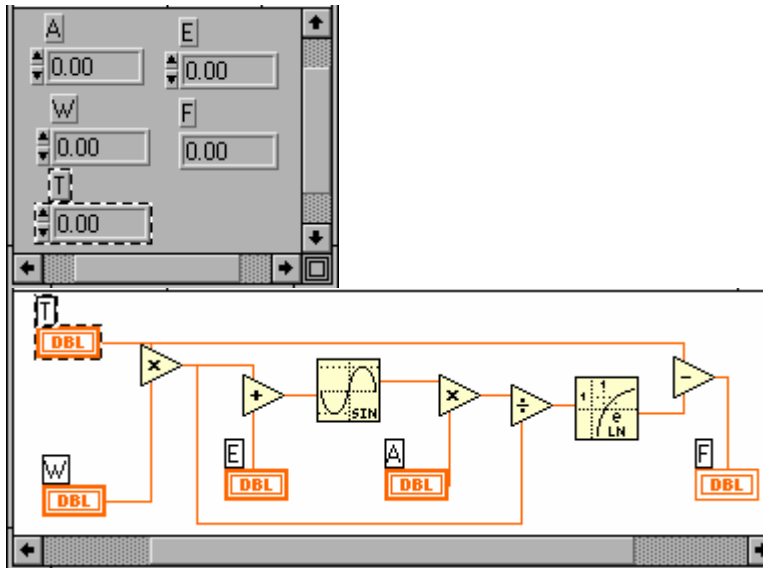


Fig3.123 Fórmula complicada

En el ejemplo de la figura 3.123 se aprecia una fórmula complicada con muchos alambres, la versión con cuadro de fórmula queda como se observa en la figura 3.124:

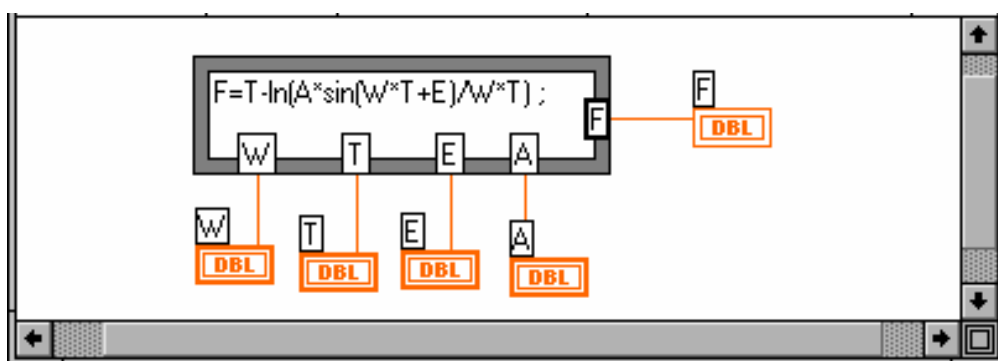


Fig3.124 Cuadro de fórmula

3.6.9 Manejo de puerto serial.- Para controlar los puertos serie, LabView presenta un menú en el que se encuentran todos los parámetros necesarios para la correcta

utilización del mismo. En la figura 3.125 se presenta el submenú y todos sus componentes.



Fig3.125 Submenú para control de puerto serial

3.6.9.1 Inicialización Del Puerto.- Para poder utilizar el puerto serie, siempre hay que inicializarlo configurando los siguientes parámetros:

- Control de flujo.- preestablecido por defecto.
- Tamaño del buffer.- indica el tamaño de entrada y salida de los datos
- Número de puerto.- Cuando se utiliza LabView en Windows los parámetros son:
 - 0 = COM1
 - 1 = COM2
- Velocidad de transmisión.- Colocamos la velocidad a la que queremos trabajar
- Bits de datos.- indica los bits que tiene los datos de entrada y salida (5-8)
- Bit de parada.- indicamos si hay o no bit de parada
- Bit de pariedad.- establecemos si vamos a tener bit de pariedad
- Error code.- nos indica si hay error en el puerto

En la figura 3.126 indicamos los conectores del inicializador del puerto

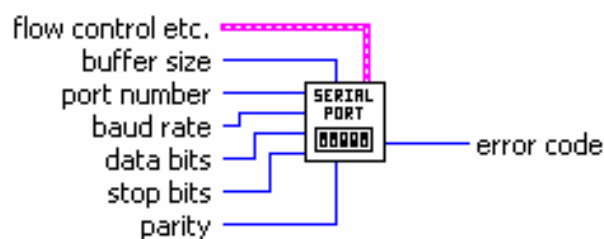


Fig3.126 Inicializador de puerto serial

3.6.9.2 Escritura Del Puerto.- Para escribir datos al puerto debemos hacerlo en forma de string, caso contrario el puerto lo no enviará, siempre se debe colocar el

mismo número de puerto que se especifica en el inicializador. La figura 3.127 muestra los conectores existentes en el icono para escribir al puerto:



Fig3.127 Subvi para escribir al puerto

3.6.9.3 Contador De Bytes.- En este subVi, el valor de retorno es el número de bytes leídos en la entrada, los conectores de este icono viene distribuidos como muestra la figura 3.128



fig3.128 Contador de bytes

3.6.9.4 Lectura Del Puerto.- Este icono es el encargado en visualizar el dato leído por el Byte Count, si el antes mencionado no presenta ningún valor, la salida del string del lector del puerto es 0, los conectores vienen distribuidos como muestra la figura 3.129

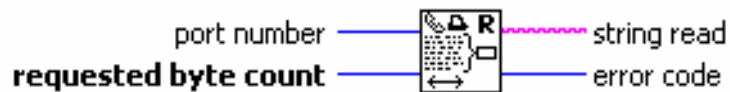


fig3.xx Lector de puerto

3.6.9.5 Fin De Puerto.- Este subVi presenta la manera de cómo cerrar el puerto para poder ser utilizado por otros programas, los conectores estan distribuidos como muestra la fig 3.130



fig3.130 Break del puerto

3.7 Programación de la PC

El programa en la PC basado en LabView, tendrá como objetivo principal visualizar a tiempo real la curva obtenida por la superficie dada por el microcontrolador desde el palpador mecánico y calcular los parámetros de rugosidad establecidos en los rugosímetros existentes en el mercado.

Como opciones adicionales, el programa podrá grabar la curva obtenida, imprimir el panel frontal y visualizar de forma individual las curvas, tanto en dos como en tres dimensiones, para que el operador tenga una mejor visión de la superficie obtenida.

3.7.1 Diagrama de bloques.- Como se puede observar en la figura 3.131, al programa de la PC se ha dividido en 5 bloques principales para su correcto funcionamiento.

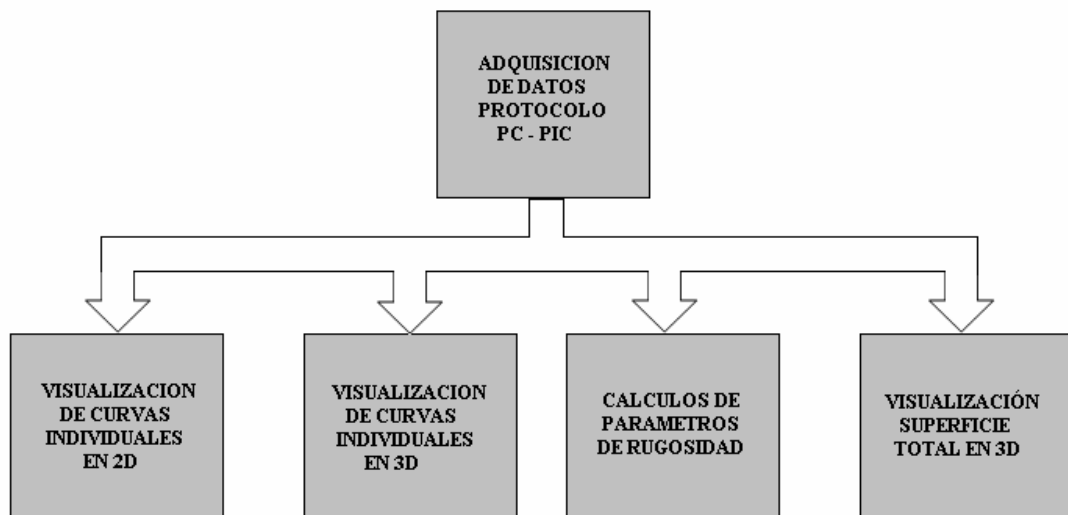


fig 3.131 Diagrama de bloques programa PC

En el primer bloque se centra en la comunicación con el microcontrolador y la recepción de los datos obtenidos por el mismo, además, el protocolo que se utilizará para que la sincronización sea correcta. En el segundo bloque, se enfatiza en la visualización de las curvas en 2 dimensiones individualmente, de la misma manera, el bloque tres se encargará de visualizar estas curvas pero en tres dimensiones, para que el operador pueda observar en nivel de pulido existente en cada una de las curvas receptadas. El bloque cuatro se encargará de realizar los cálculos de los parámetros de rugosidad y en el último bloque se podrá observar en tres dimensiones la

superficie total del material que se este analizando. A continuación se presenta con detalle cada uno de los bloques antes mencionados.

3.7.1.1 Adquisición de datos y protocolo pc – pic.- Como se explicó en la introducción teórica, el programa de LabView tiene 2 partes, el panel frontal y el diagrama de programación, en los cuales se puede configurar las opciones que el programa de rugosímetro requiere.

3.7.1.1.1 Panel Frontal De Adquisición De Datos.- Como se puede observar en la figura 3.132, el panel frontal de la adquisición de datos consta de los siguientes elementos:

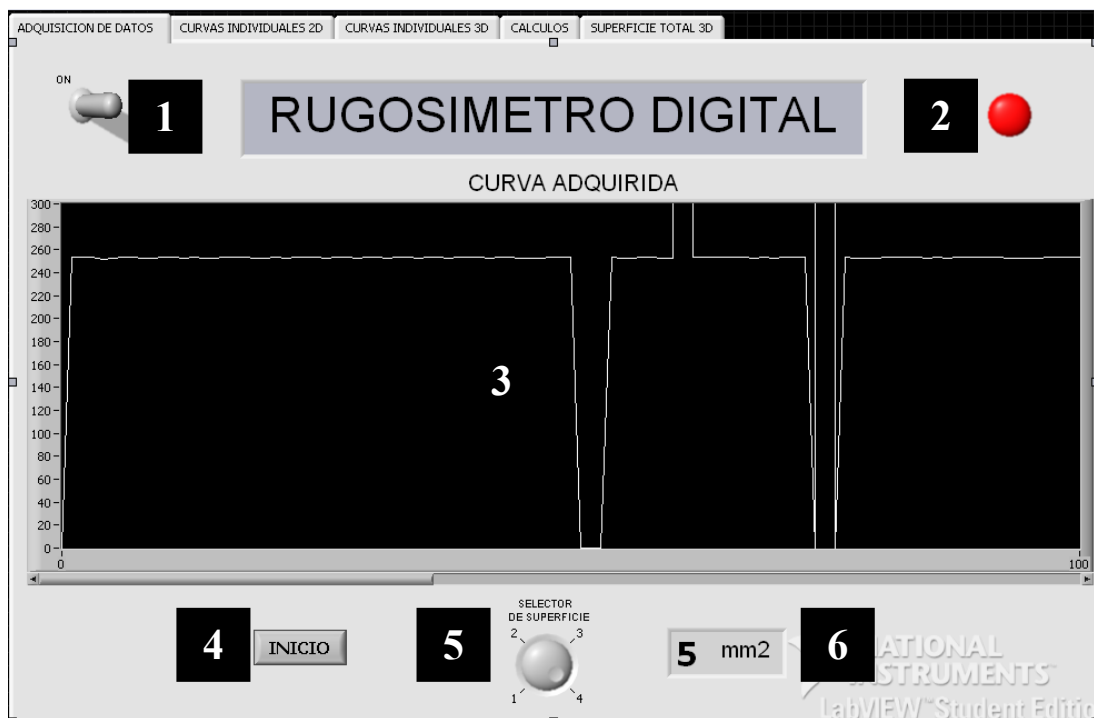


fig 3.132 Panel Frontal de Adquisición de datos

1. **Interruptor de encendido.-** Este interruptor enciende el sistema y prepara al programa a esperar la comunicación con el microcontrolador.
2. **Led indicador.-** El led indica que el sistema esta encendido o apagado.
3. **Visualizador de onda.-** Indicador a tiempo real de la curva obtenida por el puerto serial.
4. **Botón de inicio de adquisición.-** Con este botón, se comunica al microcontrolador que empiece a transmitir los datos obtenidos por el conversor.

5. **Selector de superficie.-** Con este selector, se puede escoger la superficie que se quiere visualizar, siendo estas de 5, 10, 15 y 20 milímetros cuadrados.
6. **Visualizador numérico.-** Visualiza en forma numérica la opción escogida por el selector de superficie.

3.7.1.1.2 Diagrama De Programación De Adquisición De Datos.- En el diagrama de bloques o de programación, se coloca en primer lugar un contenedor, que cumple las mismas funciones que un CASE pero con la diferencia que se puede controlar mediante las pestañas existentes en el panel frontal como indica la figura 3.133

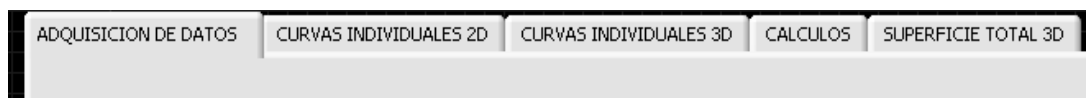


fig 3.133 Pestañas de control del contenedor

En la primera pestaña del contenedor (fig 3.134), se programa los controles existentes en el panel frontal de adquisición de datos, realizando los siguientes pasos:

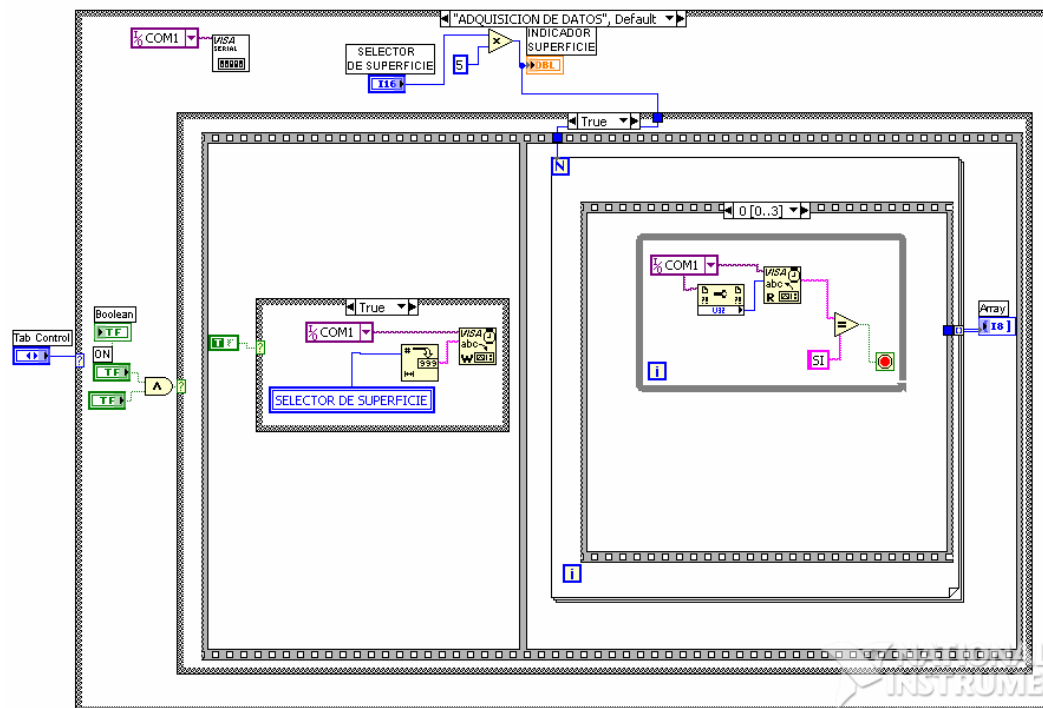


fig 3.134 Diagrama de programación de adquisición de datos

1. **Inicializador del puerto serial.-** Como primer paso, se debe inicializar al puerto serial para que este listo para comunicarse con la PC, para esto, se utiliza el protocolo VISA serial port, en el cual, se puede configurar todos los

parámetros necesarios, tales como, el número de puerto COM1-COM4, la velocidad de 9600 baudios , el número de bits, paridad, control de errores y de flujo de datos tal como muestra la figura 3.135

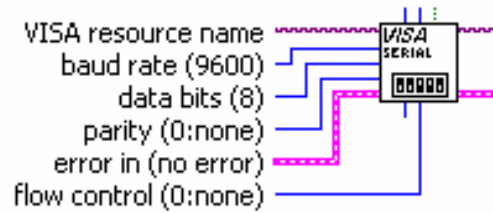


fig 3.135 Configuración del subVI VISA serial port

2. **Selector de superficie.**- Es un selector numérico al cual multiplicamos por 5 para ser enviado al indicador y luego a la secuencia de adquisición de datos, para que esta comunique al microcontrolador que superficie va a ser analizada y cuantas veces tiene que activar a los motores del rugosímetro. El selector se conecta como muestra la figura 3.136

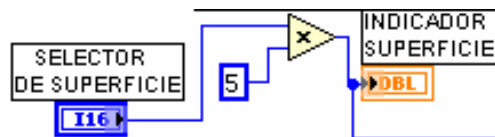


fig 3.136 Selector de superficie

3. **Inicio de adquisición de datos.**- Para que la secuencia de pueda funcionar, debe estar el interruptor encendido y haber sido presionado una vez el botón de inicio de adquisición del panel frontal. Esto se logra con una compuerta AND, que hará que el CASE pase a TRUE y se inicie la secuencia, en caso de no cumplir con lo anterior, el CASE se quedará en FALSE y la adquisición no podrá funcionar. La conexión queda como indica la figura 3.137

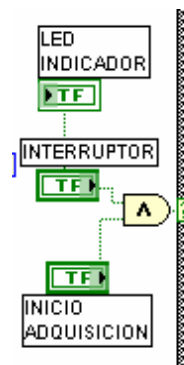


fig 3.137 Inicio de adquisición de datos

4. **Secuencia de comunicación.-** Luego de que se haya cumplido el requerimiento anterior, el CASE se coloca en TRUE y este a su vez activa la secuencia de comunicación que muestra la figura 3.138 y que se detalla a continuación:

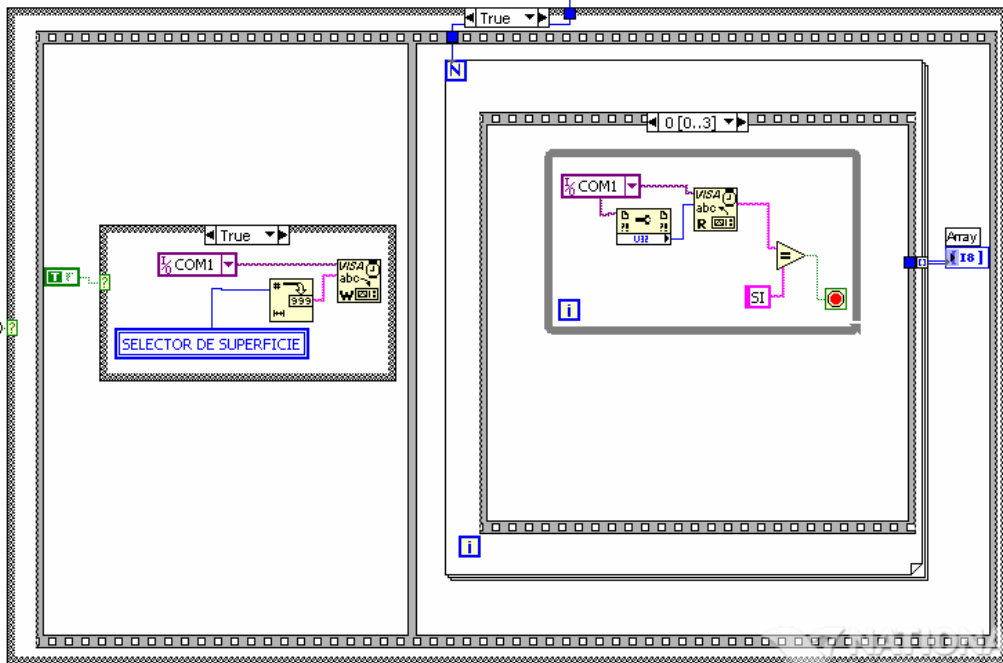


fig 3.138 Secuencia de comunicación

A) La secuencia envía en primer lugar el dato de superficie a analizar convertido de decimal a string por medio del subVI VISA write como muestra la figura 3.139

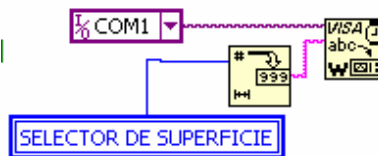


fig 3.139 Envío de superficie a analizar

B) En la segunda parte de la secuencia, se coloca un FOR LOOP que se encarga de repetir las veces que se va a adquirir los datos tomado del selector de superficie antes mencionado. En el FOR LOOP, se coloca una secuencia que tiene cuatro partes y las se detalla a continuación:

1. Mediante el subVI VISA read, se espera a que el microprocesador nos envíe la palabra SI, siendo esta la forma de comunicar que el dato de superficie fue recibido con éxito y que esta listo para enviar datos (fig 3.140)

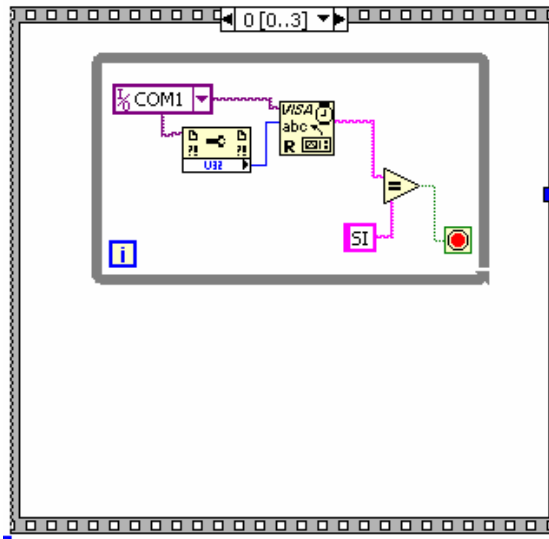


fig 3.140 Espera de dato desde el microprocesador

2. Cuando el dato es recibido, se envía un 5 para comunicar al microcontrolador que envíe los datos de conversión como muestra la figura 3.141

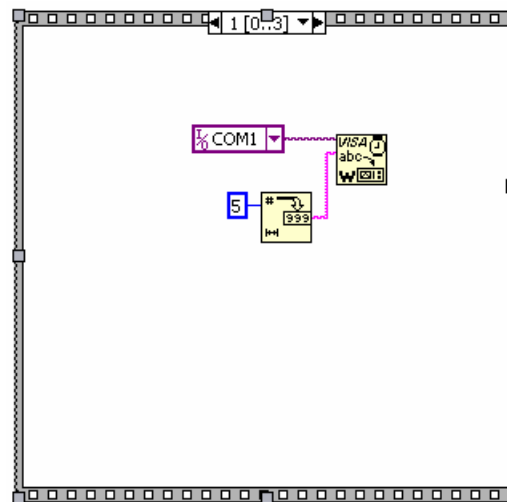


fig 3.142 Envío de dato para recepción de la conversión

3. Se coloca una demora de sincronización de 90 milisegundos como se señala en la fig. 143

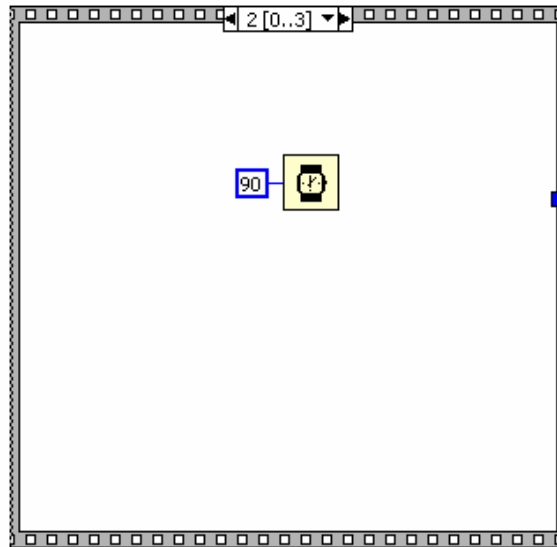


fig 3.143 Demora de sincronización

- Se dispone a recibir datos, visualizarlos y crear el arreglo de todas las curvas obtenidas, esto se logra como muestra la figura 3.144

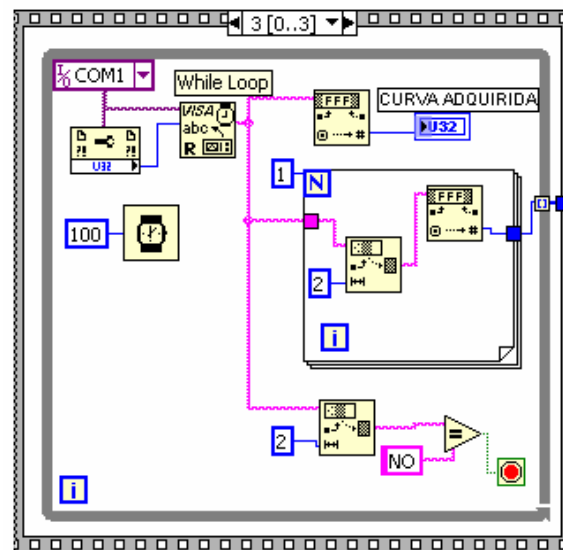


fig 3.144 Recepción de datos de conversión

Esta parte de la secuencia tiene un WHILE LOOP, el cual se encarga de recibir los datos, transformarlos a hexadecimal y visualizarlos a tiempo real, además, mediante un FOR LOOP, crear un arreglo con los datos de las curvas indexando de dos en dos los datos obtenidos, hasta que el microprocesador envíe la palabra NO, que hará que salga de dicho WHILE LOOP y se repita tantas veces como el FOR LOOP mayor este establecido.

3.7.1.2 Visualización de curvas individuales en 2D.-

3.7.1.2.1 Panel Frontal De Visualización En 2d.- Para poder visualizar las curvas obtenidas durante el proceso de trabajo del Rugosímetro, se debe seleccionar la segunda pestaña del contenedor existente en el panel frontal como muestra la figura 3.145, en el cual se pueden visualizar las diferentes opciones que se tiene para trabajar.

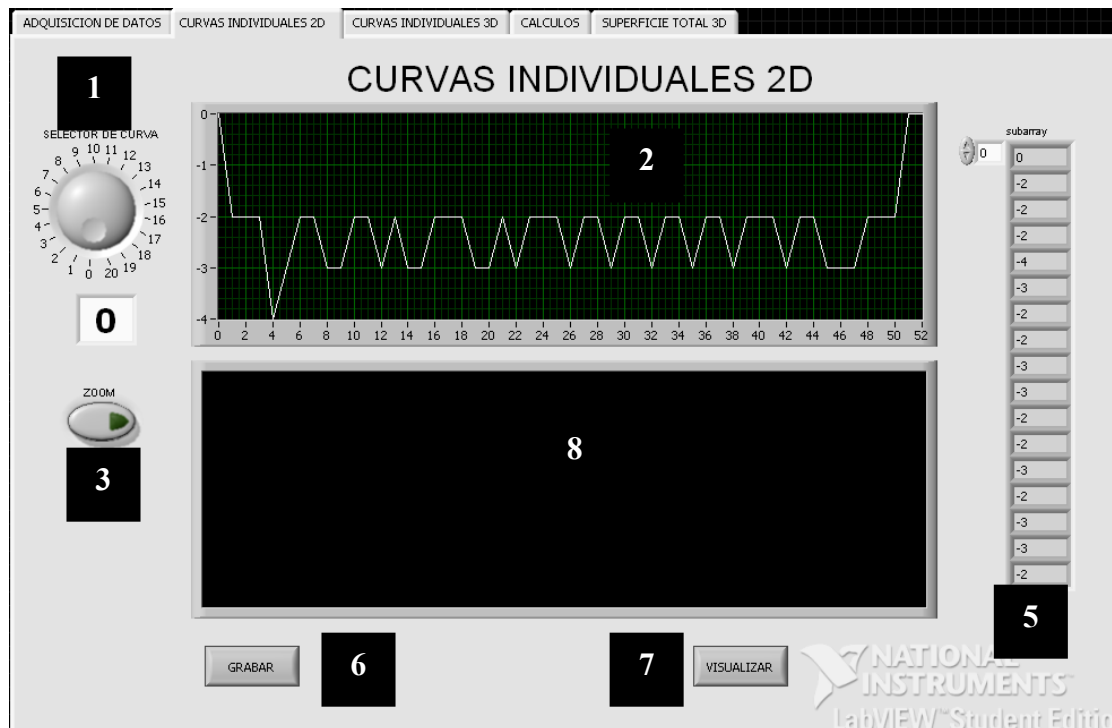


fig3.145 Visualización de curvas individuales en 2D

Como se puede observar, el este panel frontal consta de los siguientes elementos:

1. **Selector de curva.-** Mediante este selector se puede elegir la curva que se quiere visualizar del arreglo obtenido en el panel de adquisición de datos.
2. **Visualizador de onda.-** Una vez seleccionada la curva que se quiere analizar, automáticamente se graficará en el visualizador de onda.
3. **Amplificador de onda.-** El botón amplificador o zoom, es una herramienta la cual nos ayuda a visualizar de mejor forma la curva seleccionada, ya que la amplía

al tamaño de la pantalla y además activa el botón correspondiente para imprimir dicha curva como muestra la figura 3.146

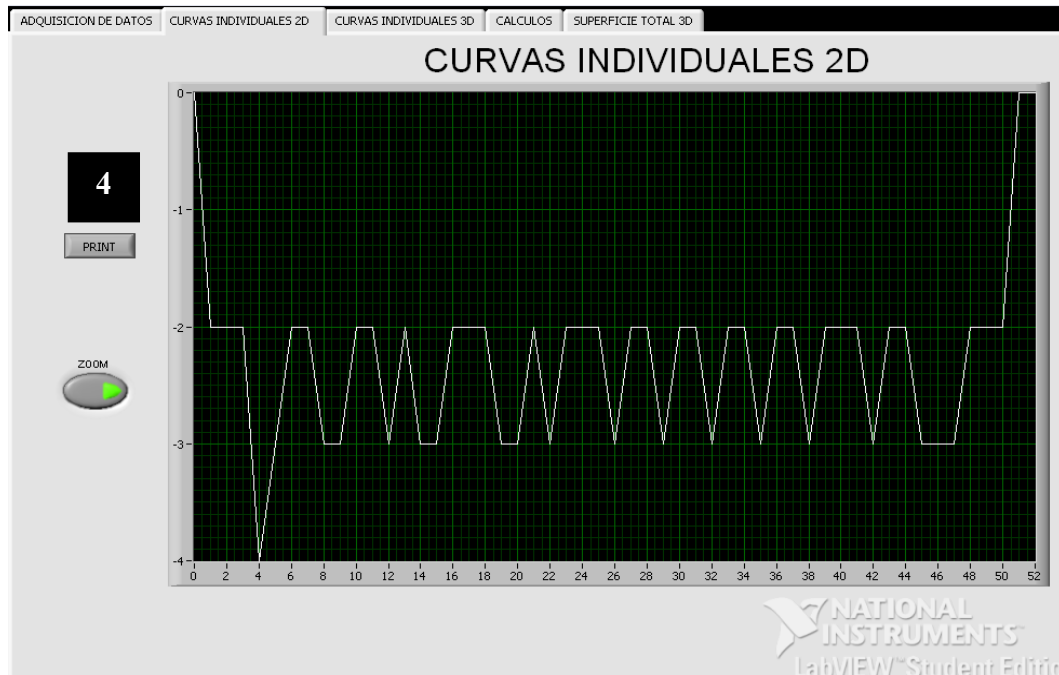


fig3.146 Zoom de curva seleccionada

4. **Imprimir.-** Con este botón, se puede imprimir directamente la señal que se este visualizando en ese momento. En la impresión saldrán también todos los elementos existentes en el panel frontal en ese momento.
5. **Arreglo de datos.-** Con este arreglo de datos, se puede identificar de una forma numérica, todos los puntos existentes en la curva.
6. **Grabar.-** Esta herramienta nos permite crear un registro de todas las curvas o señales que se necesiten, para esto, se presiona el botón y aparecerá el cuadro de diálogo en el que se pide el sitio exacto en donde se quiera guardar como muestra la fig 147.

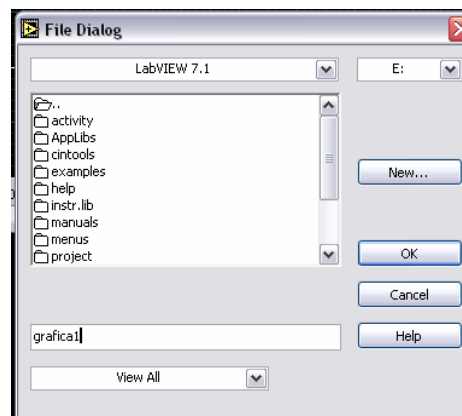


fig3.147 Cuadro de diálogo para grabar señales

7. **Visualizar.-** Cuando se quiera recuperar alguna señal ya grabada, se presiona el botón de visualizar y como en el caso anterior, se presentará el cuadro de diálogo para localizar la curva y visualizarla como muestra la figura 3.148

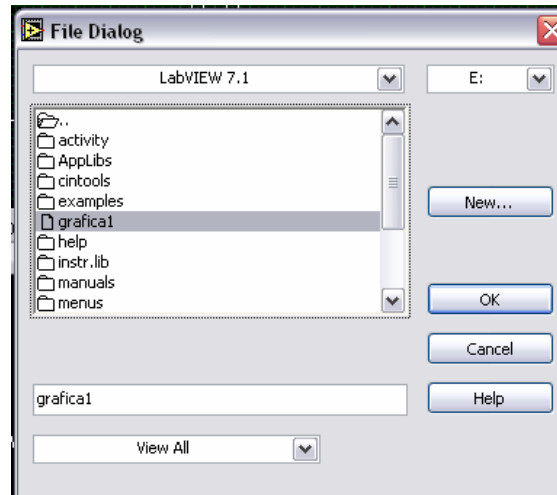


fig3.148 Cuadro de diálogo para visualizar la curva

8. **Visualizador de onda grabada.-** En este visualizador, se mostrará la onda que previamente se haya grabado como se indicó anteriormente (fig 3.149).

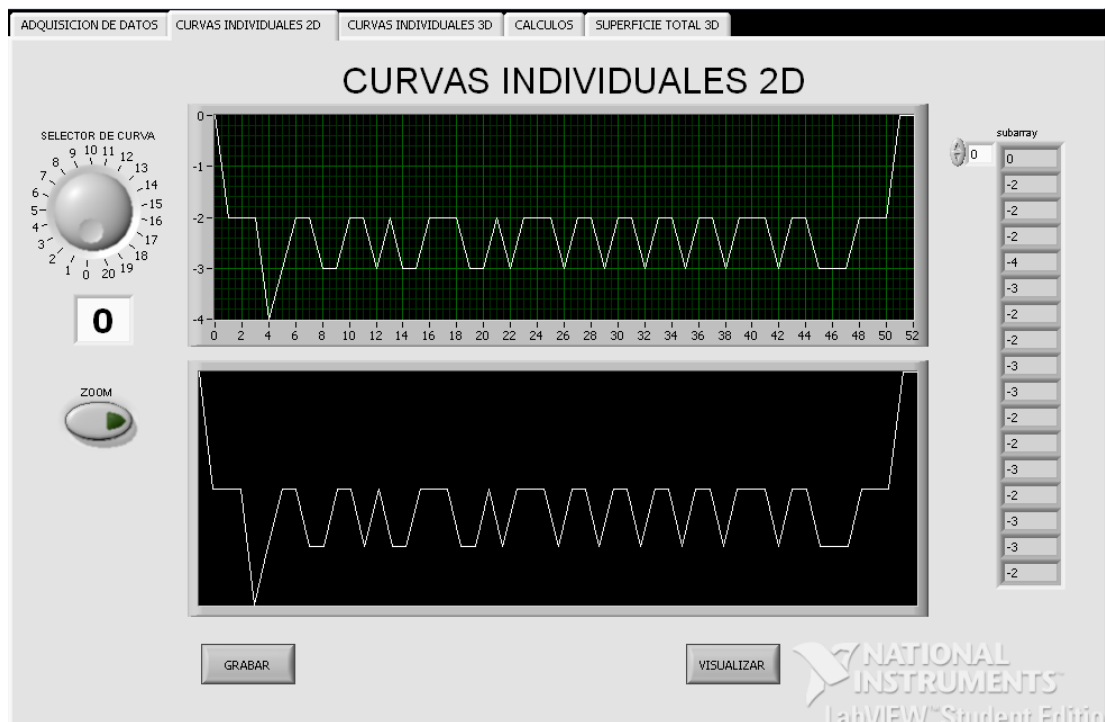


fig3.149 Curva visualizada

3.7.1.2.2 Diagrama De Programación Para Visualización De Curvas Individuales En 2d.- Para poder visualizar, grabar y ampliar las diferentes curvas previamente obtenidas, se necesita una variable local del arreglo preparado en la parte de la adquisición como muestra la figura 3.150

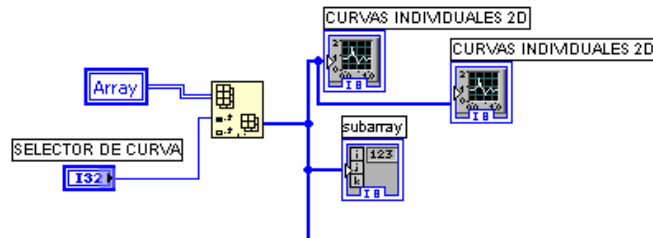


fig3.150 Obtención de curvas individuales

La variable local denominada ARRAY, posee todas las curvas enviadas por el microcontrolador a la PC, para poder elegir cualquiera de ellas, se aplica un INDEX, el cual necesita de un selector de curva que la seleccionará y automáticamente mandará la curva tanto a los visualizadores gráficos, como al subarreglo numérico.

Para la opción de ampliación o zoom, se utiliza los nodo propiedad de cada uno de los elementos existentes en el panel frontal, dicho nodo, tiene la ventaja de ocultar o aparecer lo que se necesite dando una constante falsa o verdadera.

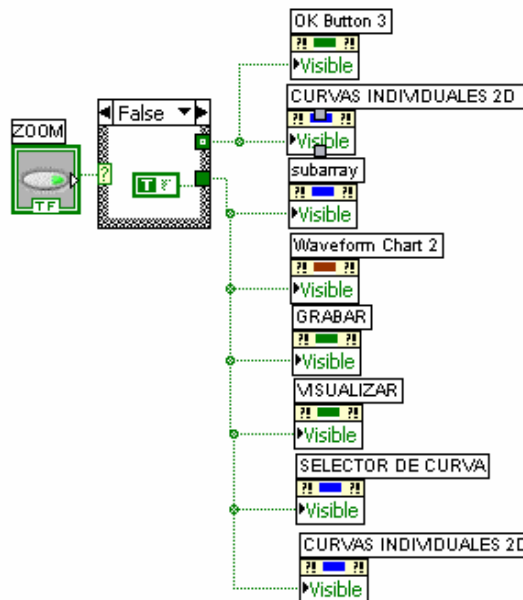


fig3.151 Opción de zoom

Como muestra la figura 3.151, mediante el botón de zoom y un CASE, se puede dar el valor de verdadero a los nodos propiedad para que aparezcan en el panel frontal.

En cuanto se pulsa, la opción cambia, ocultándose los que estaban en verdadero y apareciendo los que estaban en falso.

Una vez que se haya elegido la opción de zoom, se puede imprimir la curva seleccionada. Para esto se necesita crear un nuevo reporte, luego, introducimos la imagen en el reporte creado y se envía el mismo hacia el puerto de la impresora como muestra la figura 3.152

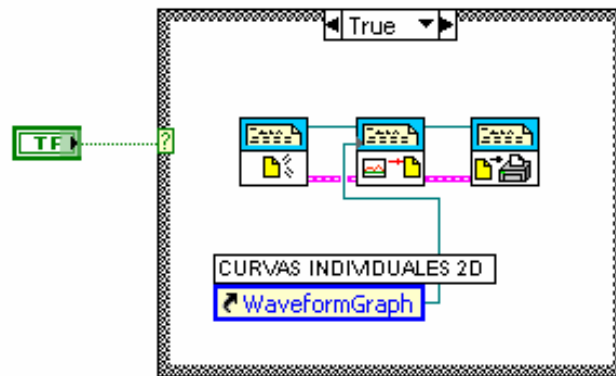


fig3.152 Impresión de forma de onda

Cuando se quiera grabar o visualizar una curva, se utiliza los botones del panel frontal y el visualizador de onda, conjuntamente con un subVI denominado REC como se puede observar en la figura 3.153

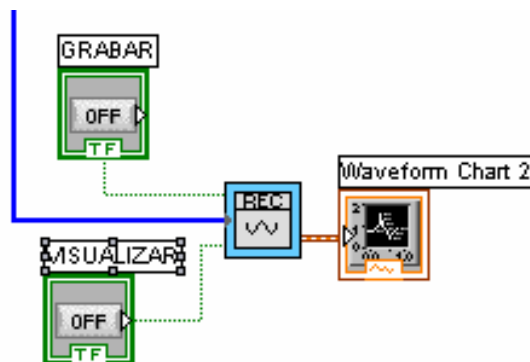


fig3.153 SubVI para grabar o visualizar una curva

Si se elige la opción GRABAR, la curva es grabada mediante el comando write waveforms to file como muestra la figura 3.154, que pedirá mediante un cuadro de diálogo, la posición y el nombre del archivo a ser grabado

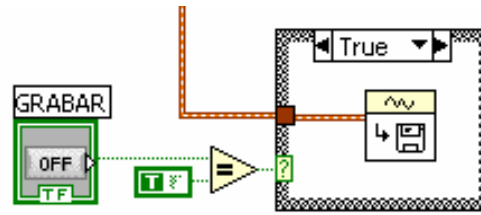


fig3.154 Grabación de curva

Ahora, cuando se presiona la opción de VISUALIZAR, con ayuda del comando read waveform from file, abrirá un cuadro de diálogo en el cual pedirá seleccionar la curva que previamente fue grabada para visualizarla de forma inmediata. La conexión de esta opción queda establecida como se muestra en la figura 3.155

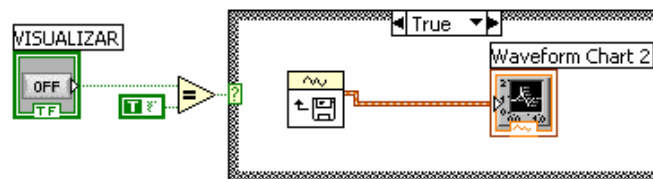


fig3.155 Visualización de curva

3.7.1.3 Visualización de curvas individuales en 3D.- En la tercera pestaña del panel frontal del Rugosímetro Digital, visualiza a las curvas guardadas en ARRAY de manera individual.

3.7.1.3.1 Panel Frontal De Visualización En 3d.- Los elementos existentes en el panel de la figura 3.156 son los siguientes:

1. **Visualizador en 3D.-** Este visualizador posee la capacidad de entregar una figura en tres dimensiones del arreglo dado en el eje Z
2. **Estilo de gráfico.-** Con esta opción, se puede elegir el estilo de cómo se quiere visualizar al gráfico, tales como, líneas, puntos, líneas y puntos, superficie normal, contorno, superficie y contorno, etc.
3. **Transparencia.-** Mediante este selector, se puede elegir el nivel de transparencia de la figura mostrada por el visualizador.
4. **Proyecciones en los ejes.-** Pulsando en los leds, se selecciona o no proyectar la figura a los ejes X, Y y Z

5. **Visualización de ejes.-** De la misma manera, con esta opción, se establece que eje estará visible en la pantalla.
6. **Auto escala.-** En cada uno de los ejes tiene la capacidad de elegir si queremos o no auto escala.
7. **Botón para visualizar.-** Pulsando el botón, automáticamente se visualizará la curva elegida en el selector.
8. **Selector de curva.-** Cumple la misma función que en el resto de las pestañas del panel frontal.

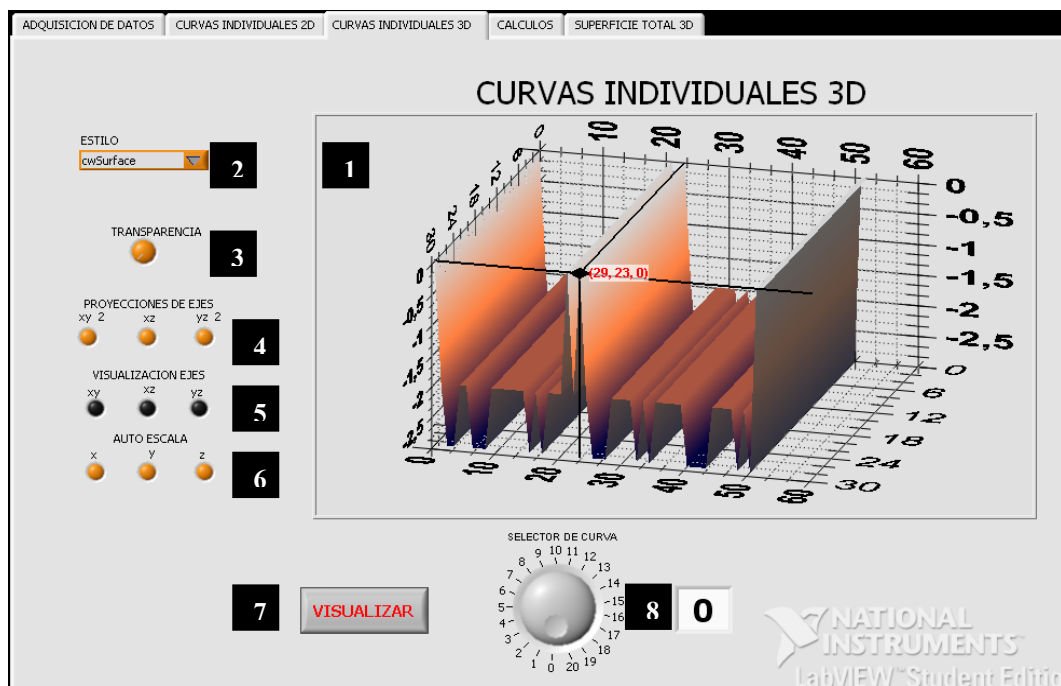


fig3.156 Panel frontal de visualización de curvas en 3D

3.7.1.3.2 Diagrama De Programación De Visualización De Curvas Individuales

En 3d.- Como en el caso de la visualización de las curvas en dos dimensiones, para hacerlo en tres dimensiones se utiliza también la variable ARRAY en la que consta todas curvas obtenidas desde el puerto serial.

Como primer paso se debe elegir que la curva a visualizar en ARRAY mediante el selector como muestra la figura 3.157

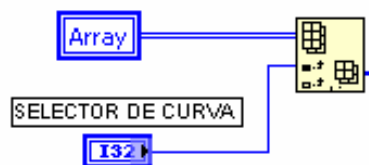


fig3.157 Selección de curva a visualizar

Una vez que se tiene la curva elegida, se introduce a la misma en un FOR LOOP que se repetirá las veces que este especificado en su contador y se visualizará siempre y cuando, el botón configurado como pulsante sea previamente accionado (fig 3.158.).

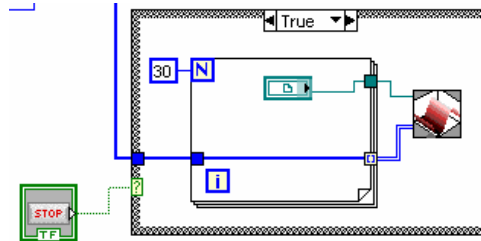


fig3.158 Visualización de la curva en 3D

Esta visualización se logra mediante el subVI llamado 3D surface.vi, en el cual se realizan automáticamente las interpolaciones entre ejes. Cuando la curva en 3D este ya graficada, a esta se puede cambiar su estilo de presentación y transparencia mediante el subVI de propiedades básicas de superficie, para proyectar la curva a los diferentes ejes se utiliza las propiedades de proyección.

De la misma manera, para poder presentar o no los ejes en la pantalla, se utiliza la ayuda del subVI Grid Properties.vi y para tener auto escala Axes Properties.vi, teniendo así, la posibilidad de manipular la grafica y visualizarla de la manera que se requiera. Todas las opciones antes mencionadas, se pueden cambiar mediante los leds conectados y configurados como controladores para que el usuario tenga facilidad en su manejo como muestra la figura 3.159

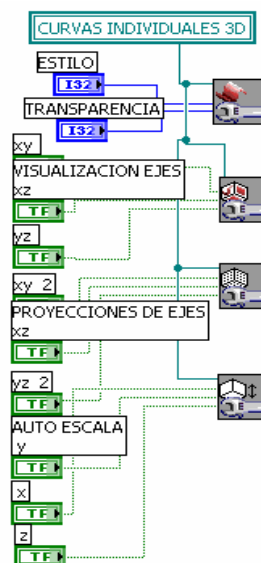


fig3.159 Configuración de curva 3D

3.7.1.4 Cálculo de parámetros de rugosidad.- Luego de haber obtenido las diferentes curvas como se ha explicado anteriormente, se prosigue a realizar los cálculos de rugosidad estándares de un rugosímetro digital, siendo estos la altura máxima de perfil RY y la altura de las irregularidades en 10 puntos RZ (1.2).

3.7.1.4.1 Panel Frontal Del Cálculo De Parámetros De Rugosidad.- Como se puede observar en la figura 3.160, hay la posibilidad de verificar los parámetros de rugosidad de cualquier curva obtenida anteriormente con un selector de señal.

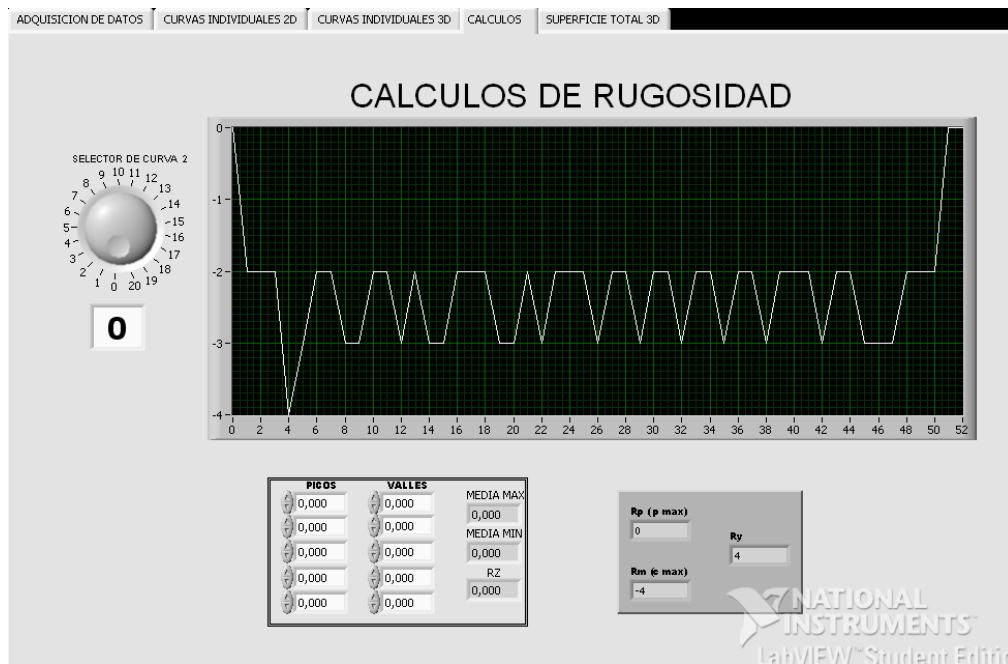


fig3.160 Cálculos de rugosidad

Los datos para calcular la altura de las irregularidades en 10 puntos RZ viene dada por la fórmula explicada en el capítulo I (1.2)

$$Rz = \frac{\sum_{i=1}^5 |Y_{pi}| + \sum_{i=1}^5 |Y_{yi}|}{5}$$

Realizando la fórmula, se introduce cinco datos de picos y cinco de valles, lo que da como resultado las medias máximas de cada uno y a su vez el valor de la altura de las irregularidades en 10 puntos Rz, como se observa en la figura 3.161

PICOS		VALLES		
<input type="text" value="0,000"/>	<input type="text" value="0,000"/>	<input type="text" value="0,000"/>	<input type="text" value="0,000"/>	MEDIA MAX
<input type="text" value="0,000"/>	<input type="text" value="0,000"/>	<input type="text" value="0,000"/>	<input type="text" value="0,000"/>	<input type="text" value="0,000"/>
<input type="text" value="0,000"/>	<input type="text" value="0,000"/>	<input type="text" value="0,000"/>	<input type="text" value="0,000"/>	MEDIA MIN
<input type="text" value="0,000"/>	<input type="text" value="0,000"/>	<input type="text" value="0,000"/>	<input type="text" value="0,000"/>	<input type="text" value="0,000"/>
<input type="text" value="0,000"/>	<input type="text" value="0,000"/>	<input type="text" value="0,000"/>	<input type="text" value="0,000"/>	RZ
<input type="text" value="0,000"/>	<input type="text" value="0,000"/>	<input type="text" value="0,000"/>	<input type="text" value="0,000"/>	<input type="text" value="0,000"/>

fig3.161 Parámetro de rugosidad Rz

Para obtener el parámetro de altura máxima de perfil RY, automáticamente el programa nos da el resultado del pico y valle máximo y realiza la suma de sus valores absolutos (fig3.162)

Rp (p max)	<input type="text" value="0"/>	Ry	<input type="text" value="4"/>
Rm (e max)	<input type="text" value="-4"/>		

fig3.162 Parámetro de rugosidad Ry

3.7.1.4.2 Diagrama De Programación Del Cálculo De Parámetros De Rugosidad.- Como muestra la figura 3.163, mediante sumadores se puede obtener los valores medios de los picos y valles ingresados por el usuario y estos a su vez van a un sumador que da como resultado el valor del parámetro buscado Rz

En el caso de Ry, se necesita un subvi llamado array max y min, este localiza de manera inmediata dichos valores y mediante operadores absolutos, realizamos la suma y obtenemos el dato requerido.

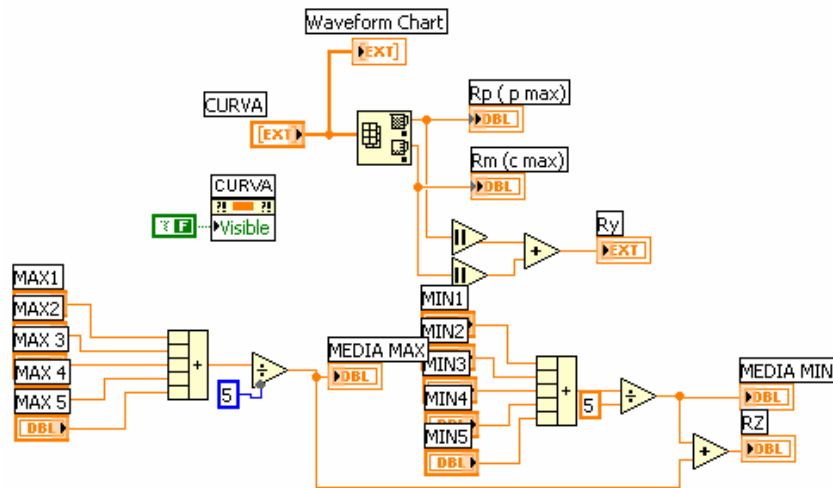


fig3.163 Diagrama de cálculos de rugosidad

3.7.1.5 Superficie total en 3D.- Mediante esta opción, se puede observar la superficie total obtenida por las diferentes curvas almacenadas, con lo cual se puede tener una idea en valores microscópicos reales de la superficie en cuestión.

3.7.1.5.1 Panel Frontal De La Superficie Total En 3d.- En este panel, tenemos como anteriormente mencionamos, la superficie total obtenida mediante la comunicación serial pc-pic, además, cuenta con un cursor el cual nos sirve para localizar un punto específico en toda la superficie (fig3.164)

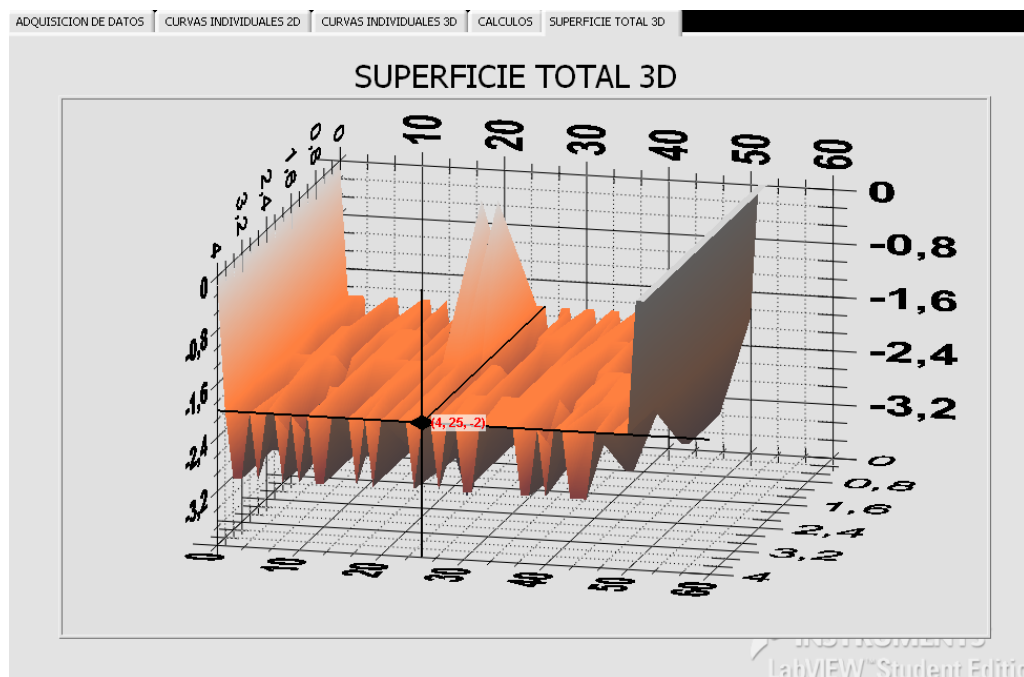


fig3.164 Superficie total en 3D

3.7.1.5.2 Diagrama De Programación De La Superficie Total En 3d.- Para poder visualizar la superficie total en tres dimensiones, simplemente se utiliza la variable local ARRAY como en otros casos se introduce estos datos en el subVI surface.vi como muestra la figura 3.165

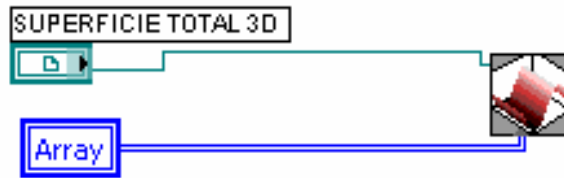


fig3.165 Superficie total en 3D

CAPITULO 4

ASPECTOS CONSTRUCTIVOS Y VALIDACIÓN DEL EQUIPO

4.1 Aspectos generales del PCB

Como ya se especificó en el capítulo II, en la elaboración del PCB (printed circuit board), hay que tomar en cuenta el ancho de las pistas, separación entre ellas, si la impresión será en una o en las dos caras, etc. En los próximos puntos se mostrará las diferentes placas que se elaboraron para el manejo tanto en la parte de control como en la de fuerza para el rugosímetro.

4.1.1 Placa impresa circuito de fuerza.- El programa Protel nos presenta la ventaja de poder presentar una visión en tres dimensiones de la placa una vez que esta este terminada, es decir, colocados todos los elementos, creado el netlist, verificado que no exista errores y la elaboración del rutado. El rutado de la placa de fuerza queda como muestra la fig.4.166.

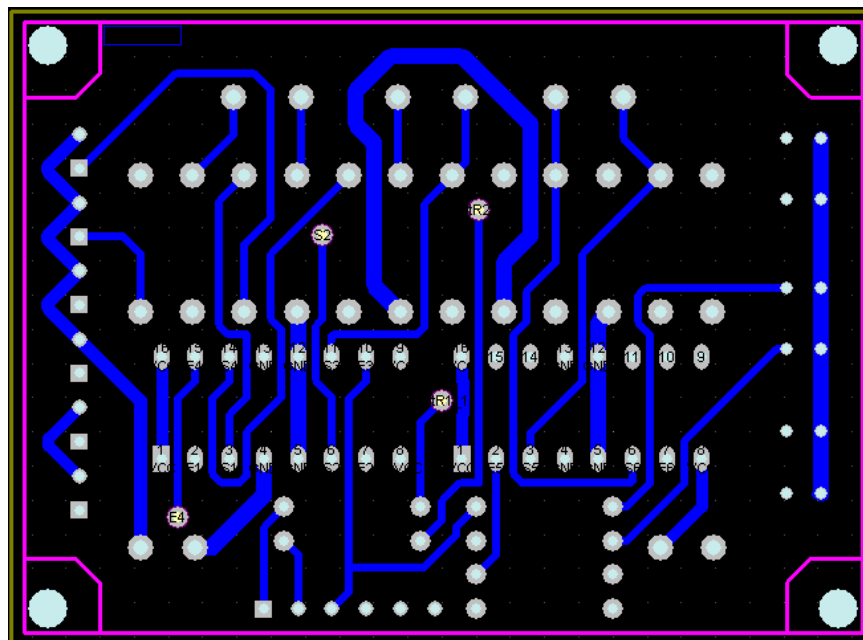


Fig4. 166 Rutado placa de fuerza

En la fig4.167 se presenta un aproximado del terminado de la placa, cabe recalcar que cuando Protel no reconoce una huella, ya sea porque esta fue creada o porque no tiene vista en tres dimensiones del objeto, simplemente coloca un rectángulo de las dimensiones necesarias para ocupar el espacio de la huella requerida.

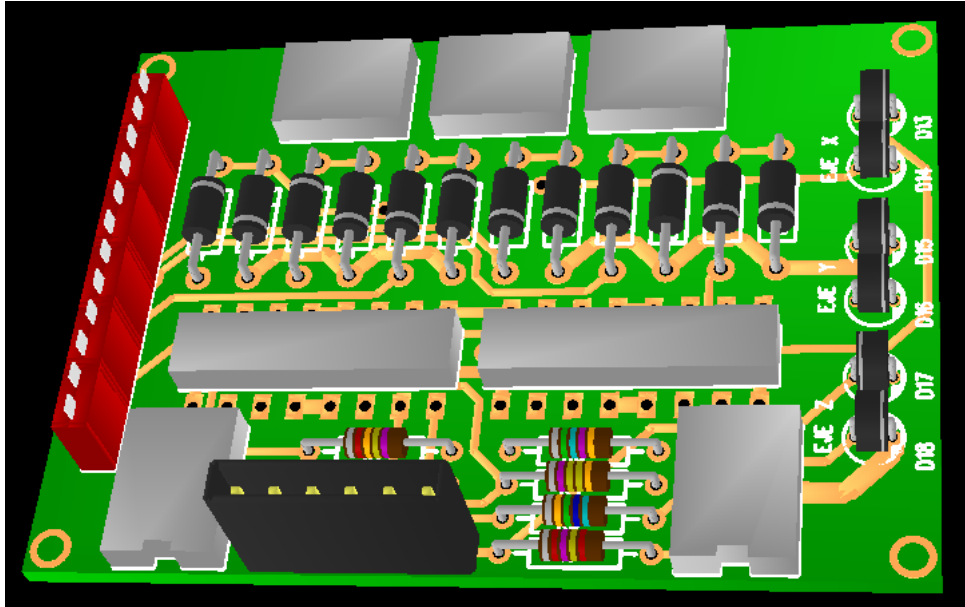


Fig4. 167 Tarjeta impresa Circuito de Fuerza

En la figura se puede observar también los diodos requeridos en la protección de los motores, los espacios en los que estarían los L293, los conectores, etc.

4.1.2 Placa impresa circuito de mando.- De manera similar a la placa de fuerza, primero se debe colocar todos los elementos como se requiera que queden establecidos, una vez hecho esto se procede al rutado, con lo que se obtiene lo que se presenta en la fig4.168.

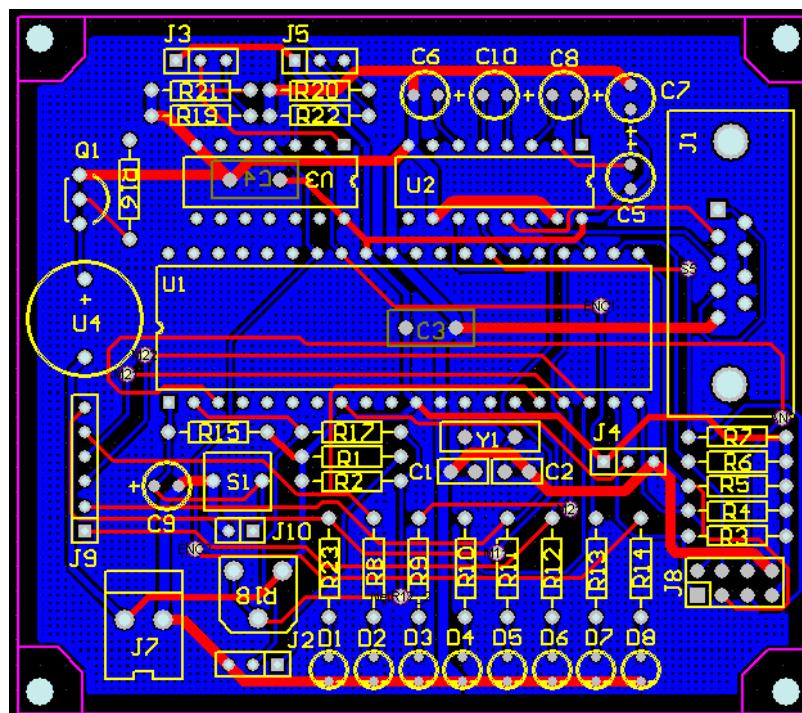


Fig4. 168 Rutado placa de mando

Como se puede observar, la placa esta elaborada a dos caras, esto se realiza de manera que se reduzca espacio y se optimice los espacios entre pistas, lo que esta con color rojo es la cara superior y lo que esta con azul es la cara inferior, de esta manera Protel nos ayuda de una manera visual a identificar las caras de las placas para que no exista errores posteriores.

Cuando ya se tiene listo y por algún motivo se desea cambiar la posición de cualquier elemento, esto si es posible, pero hay que tener mucho cuidado ya que se podría desconectar algo y por lo tanto el PCB fallaría en su totalidad.

Al igual que en la placa de fuerza, se puede realizar la simulación en tres dimensiones de la placa para tener una idea aproximada de cómo van a quedar los elementos colocados, los resultados se observan en la fig.4.169

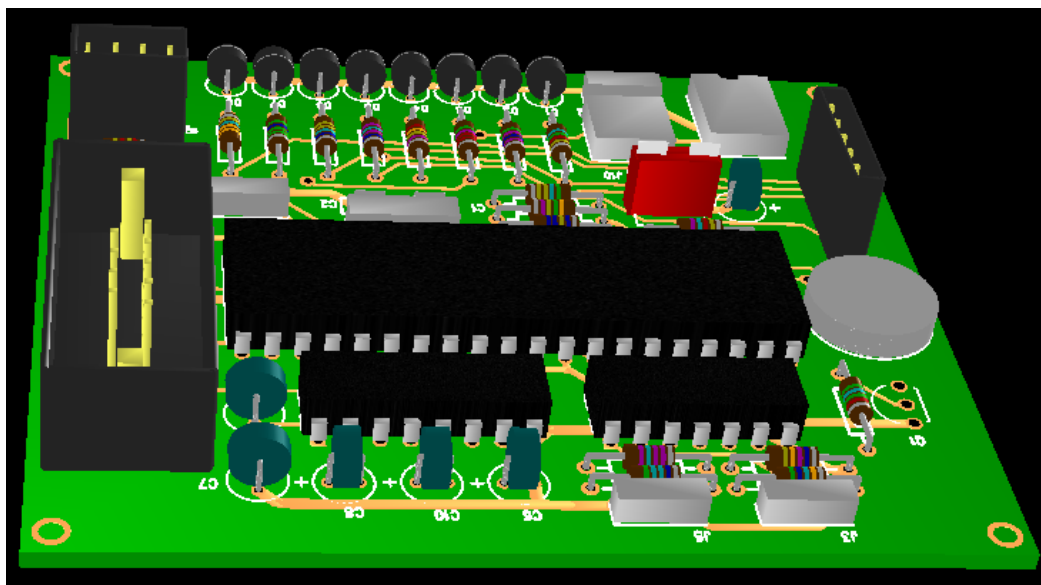


Fig4.169 Placa circuito de mando

En la figura anterior se puede observar como están colocados los elementos que en el capítulo II se trato a detalle, el microcontrolador, el Max 232, conector DB9 entre los más destacados.

En el caso de los condensadores, su presentación es diferente a la real, esto se debe a que Protel tiene solo esa huella en 3D para este tipo de elemento.

4.1.3 Placa impresa fuente de alimentación.- Como en los casos anteriores, se presenta primero el rutado de la placa de la fuente de poder, la cual alimenta tanto a la placa de fuerza como a la de mando. La fig4.170 especifica lo antes mencionado, cabe mencionar que la placa esta realizada a dos caras para que su dimensión sea mas reducida.

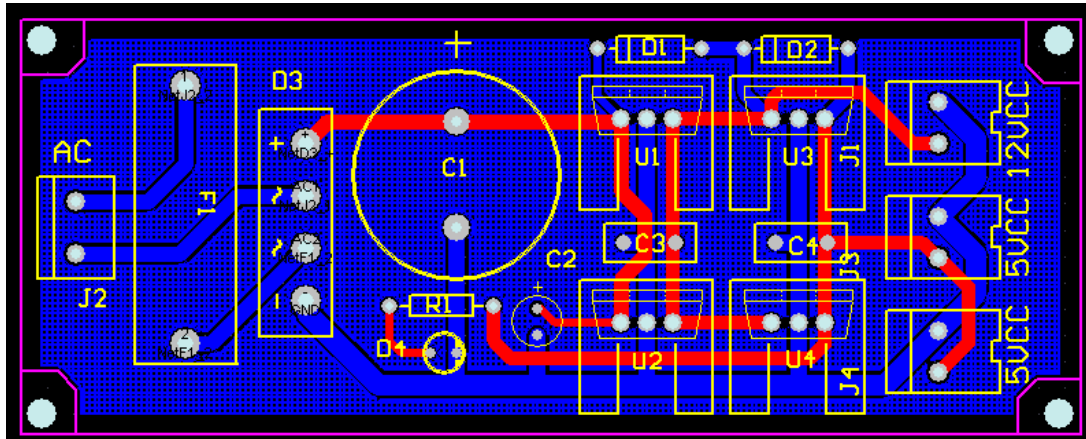


Fig4.170 Rutado de la fuente de alimentación

En la fig4.171 se muestra la vista en tres dimensiones de la placa de la fuente de alimentación.

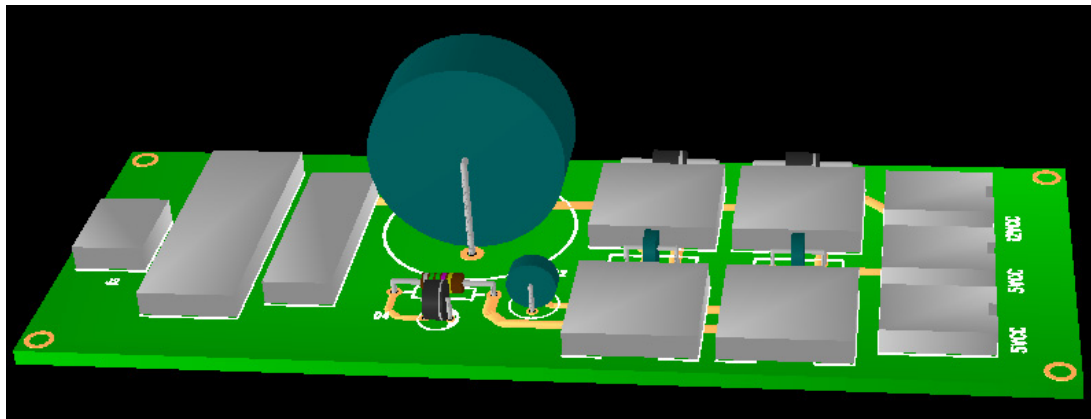


Fig4.171 Placa fuente de alimentación

4.2 Construcción de la parte mecánica

La parte mecánica tiene es una parte muy importante en la exactitud del equipo, ya que si existen fallas en la misma, el sensor no tomará medidas correctas y esto ocasionará que la PC no grafique bien y por lo tanto todo el proyecto quedaría estropeado.

En la fig.4.172 se puede observar una imagen digitalizada de la parte mecánica del Rugosímetro Digital con visualizador 3D

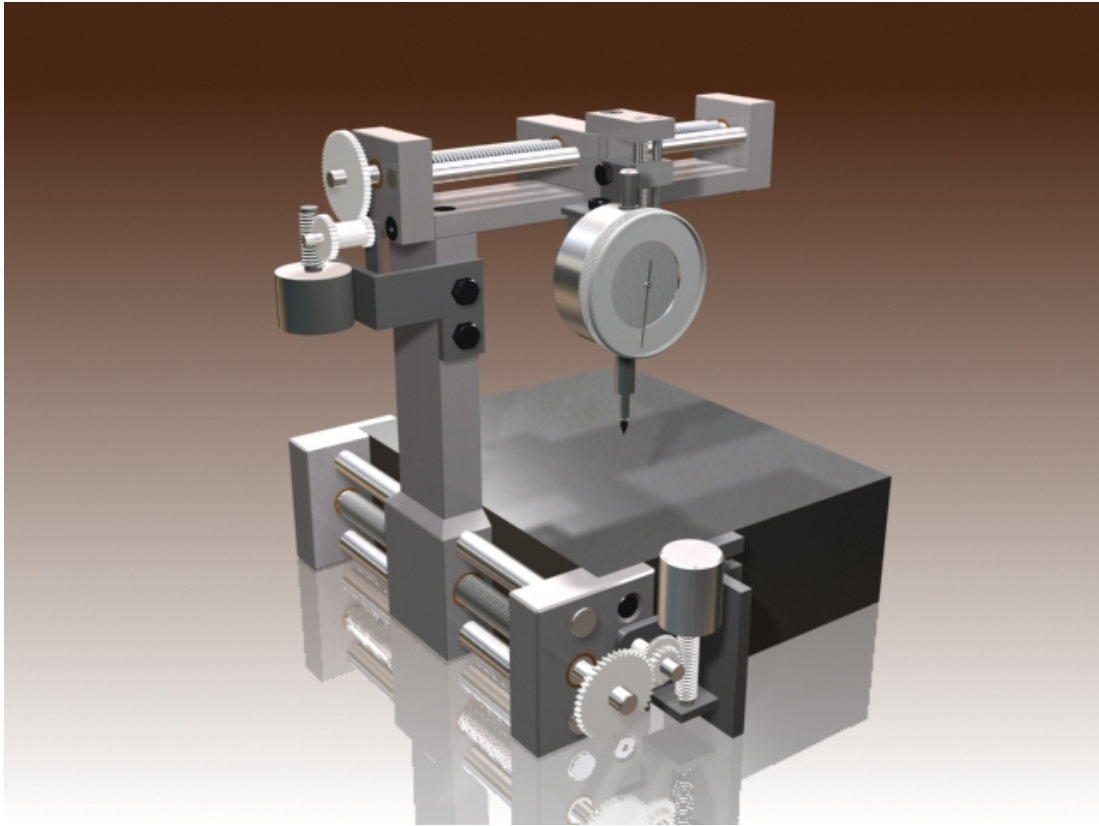


Fig4.172 Rugosímetro Digital

El rugosímetro consta de tres ejes, los cuales dos sirven para la toma de datos y el tercero para la calibración del sensor. Cuenta también con dos motores de corriente continua cuyo funcionamiento y manejo ya se especificó también. A continuación, se detalla cada parte del rugosímetro y su función.

4.2.1 Eje X.- El motor, como ya se explicó en el capítulo II, se comanda con corriente continua, este se encarga de mover un tornillo con roscado milimétrico que lleva de izquierda a derecha al brazo del rugosímetro en el eje X. En el eje del motor existe un engranaje especial que se encarga de cambiar de movimiento de rotación a lineal al movimiento del motor. El juego de engranajes sirve para dar más torque al motor y disminuir su velocidad, de manera que se tomen los datos a un tiempo constante y correcto, a continuación se presenta la fig.174 que muestra al motor del eje x:

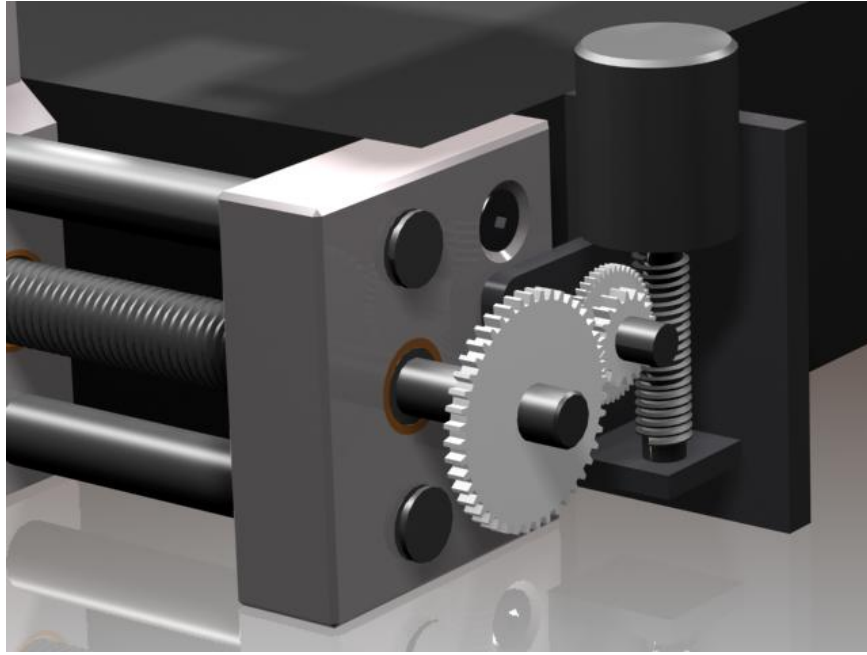


Fig4.173 Motor Eje X

Como se puede ver, consta de 3 engranajes de grilón, uno colocado en el eje del motor similar a un tornillo sin fin y este a su vez esta acoplado a un engranaje doble, el cual transmite el movimiento al mayor que esta colocado al eje principal. Tanto en los extremos de los ejes como en la parte del brazo, se colocó bujes de bronce para que no exista desgaste o movimientos laterales al momento del funcionamiento del motor, asegurando de esta manera un perfecto movimiento lineal, esto se puede observar en la fig. 4.174 a continuación:



Fig4. 174 Bujes y tornillo sinfin

En la fig.4.175 se puede visualizar dos barras laterales al tornillo que fueron colocadas para dar rigidez al sistema, ya que de no tenerla, el tornillo podría estar sujeto a mucho esfuerzo.

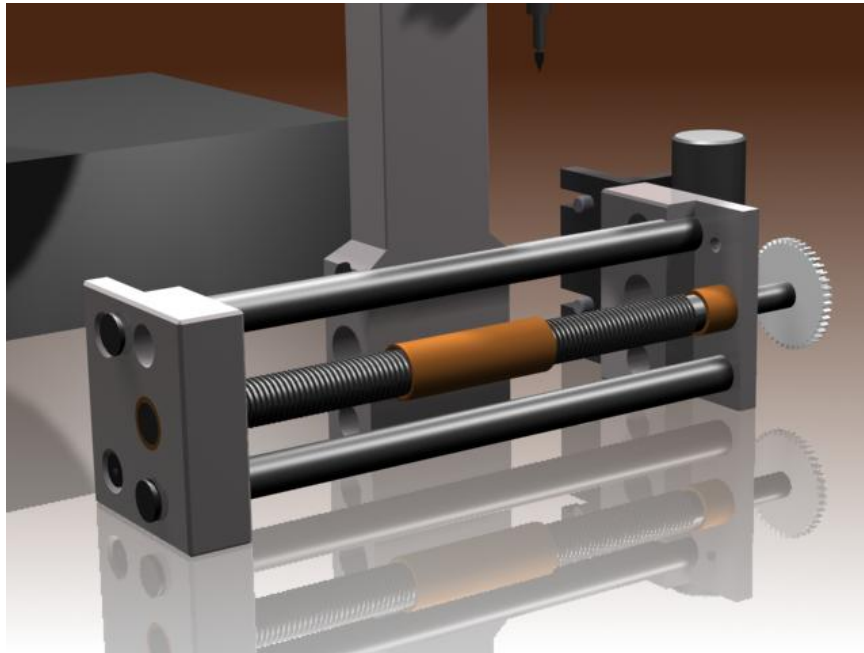


Fig4. 175 Barras laterales en el eje X

Otro punto importante en el eje X es el brazo transportador de aluminio, este se encarga de llevar al micrómetro de izquierda a derecha a lo largo del tornillo y a su vez de soportar al brazo en la dirección Y. En la fig. 4.176 se puede observar al brazo fuera del rugosímetro:

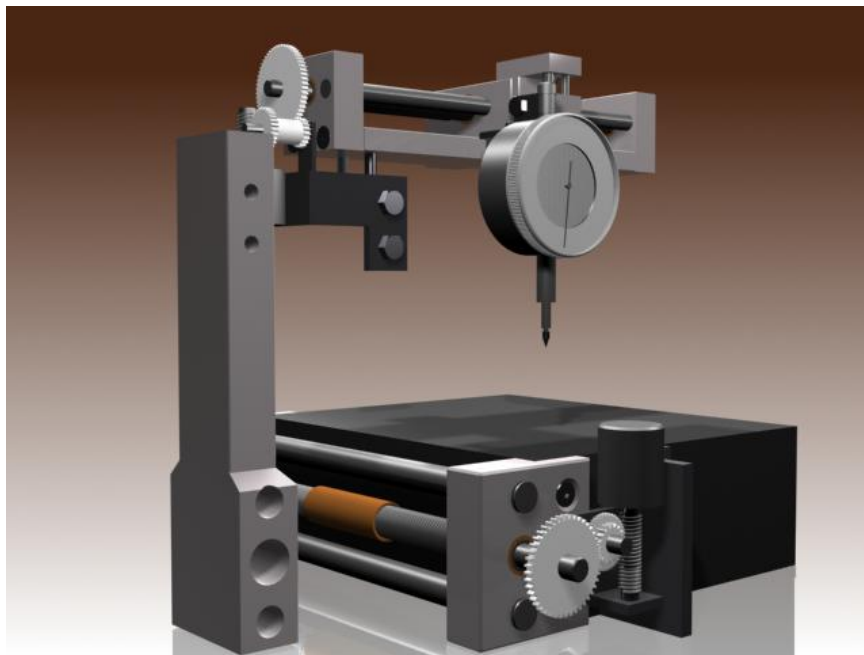


Fig4. 176 Brazo del Rugosímetro

4.2.2 Eje Y.- Como se explicó en el punto anterior, el brazo colocado en el eje X soporta al brazo ubicado en el eje Y. Este consta de un sistema de transporte muy similar al anterior. En la fig4.177 se puede observar al motor colocado al extremo del eje antes mencionado y el sistema de engranajes para cambiar el sentido de movimiento de rotación a lineal y mejorar el torque del mismo.

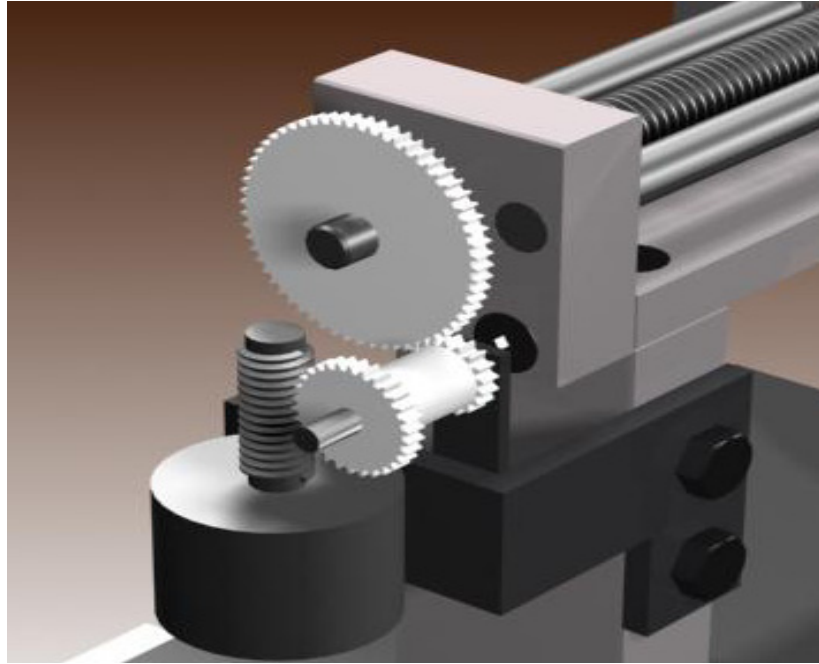


Fig4 177 Motor en el eje Y

En la figura 4.178 se puede visualizar el tornillo colocado en el eje Y, así como también las barras laterales que ayudan a soportar el peso del eje Z que contiene al micrómetro.

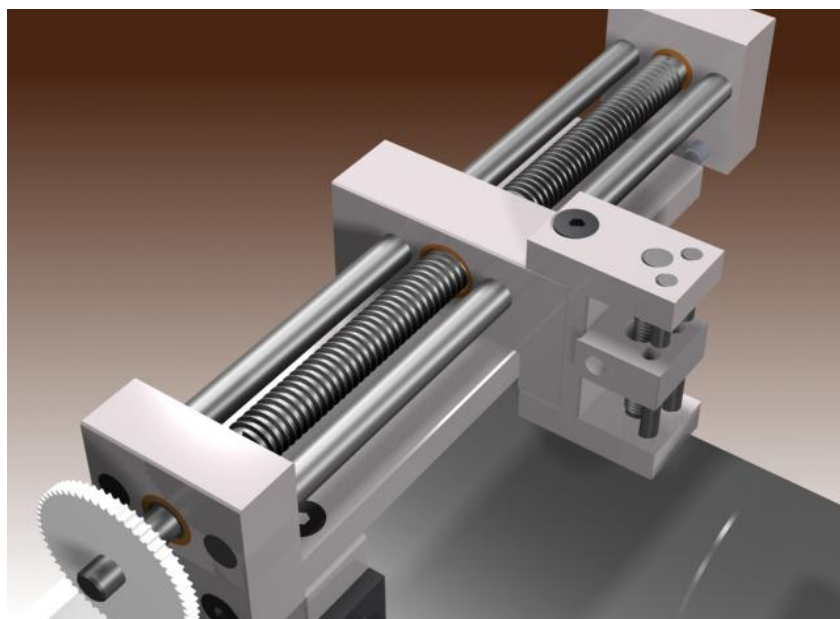


Fig4.178 Tornillo y Barras del eje Y

De igual forma que en el tornillo colocado en el eje X, en este se colocaron bujes de bronce para disminuir la fricción, obtener un movimiento continuo y de esta manera el desplazamiento del transportador del eje Z pueda circular sin ningún problema, esto se puede visualizar en la fig4.179 a continuación:

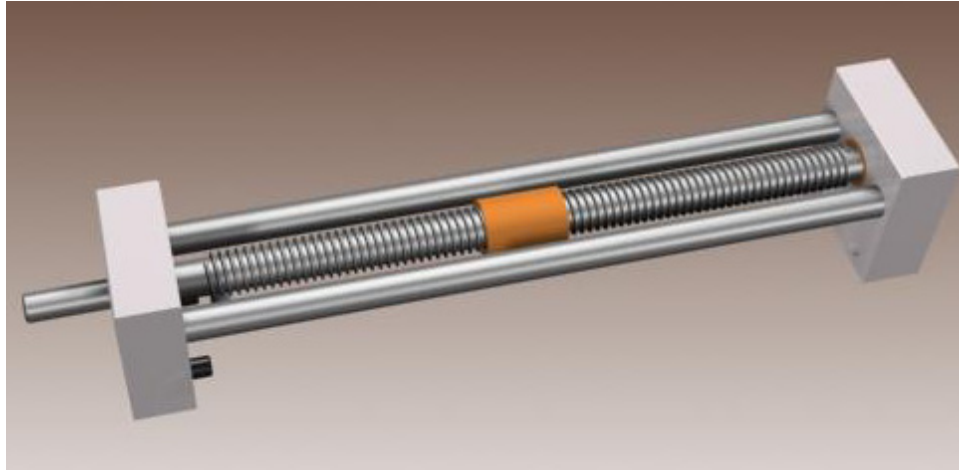


Fig4.179 Colocación de bujes en el eje Y

4.2.3 Eje Z.- A lo largo del eje Z se tiene el movimiento del micrómetro (fig4.180) y al sensor que toma los datos de rugosidad de las piezas que se requieran verificar.

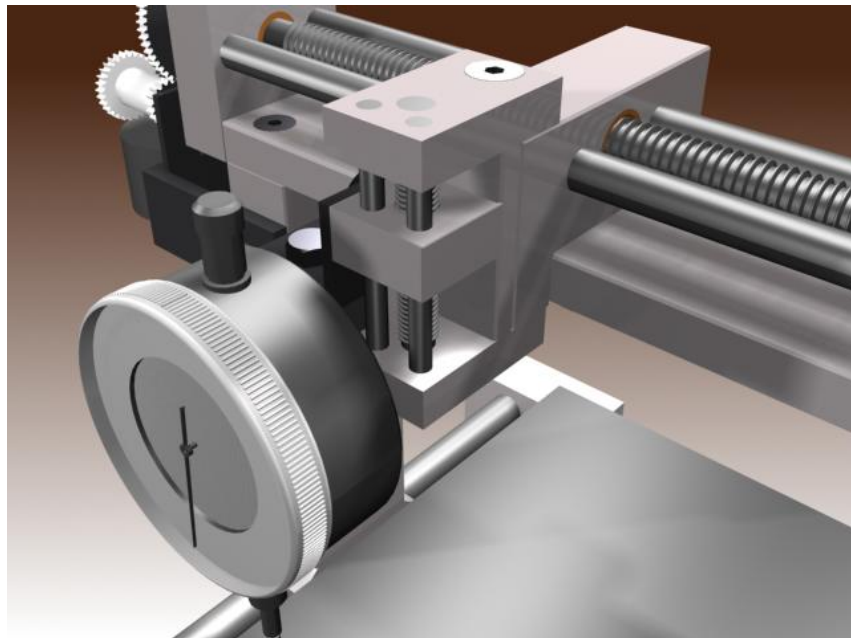


Fig4.180 Micrómetro colocado en el eje Y

Cuenta con un pequeño tornillo que sirve para subir y bajar al micrómetro, el objetivo de este sistema es poder calibrar al sensor y encerrarlo antes de tomar las

mediciones, de manera que no se tenga ningún tipo de error al momento de la toma de datos, este eje se lo puede observar en la fig4.181.

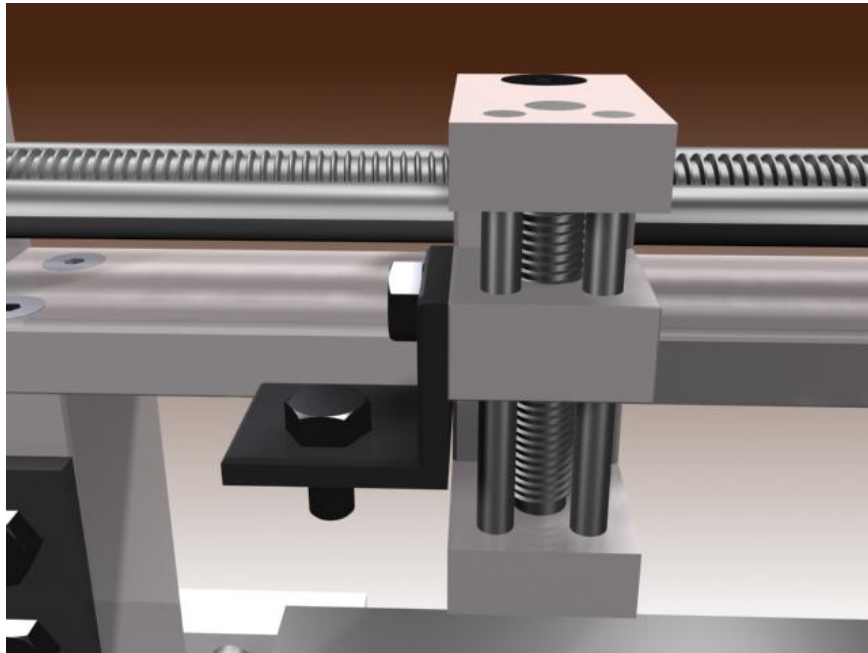


Fig4.181 Tornillo sin fin del eje Z

4.2.4 Micrómetro.- Como ya se mencionó anteriormente, el micrómetro cuenta con un sensor que se encarga de adquirir los datos de rugosidad por medio de la punta para luego enviarlos a la PC, se lo puede visualizar en la fig4.182

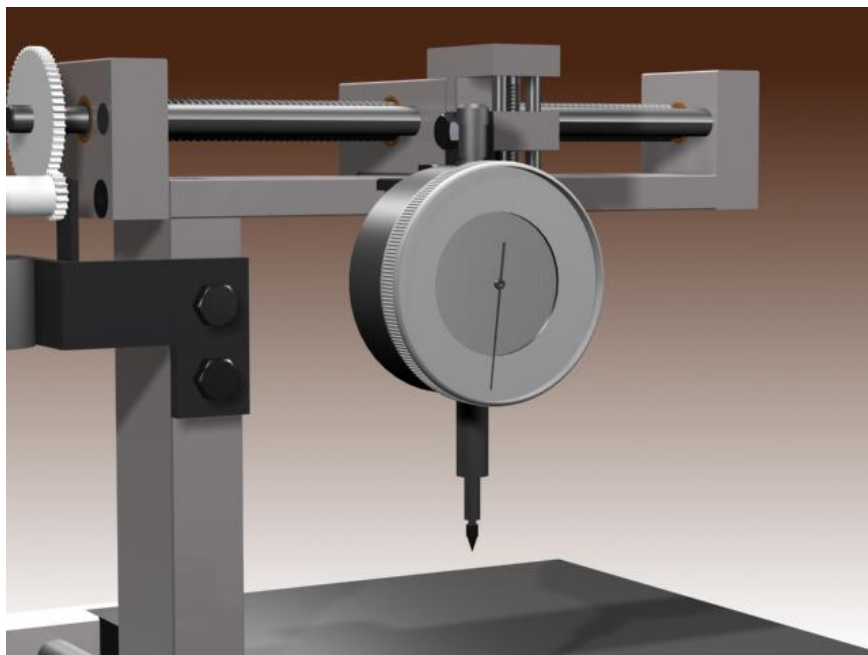


Fig4.182 Micrómetro del eje Y

El micrómetro cuenta con un soporte el cuál hace posible su desmontaje o cambio total si se necesitara cambiar de sensor por otro como se muestra en la fig4.183

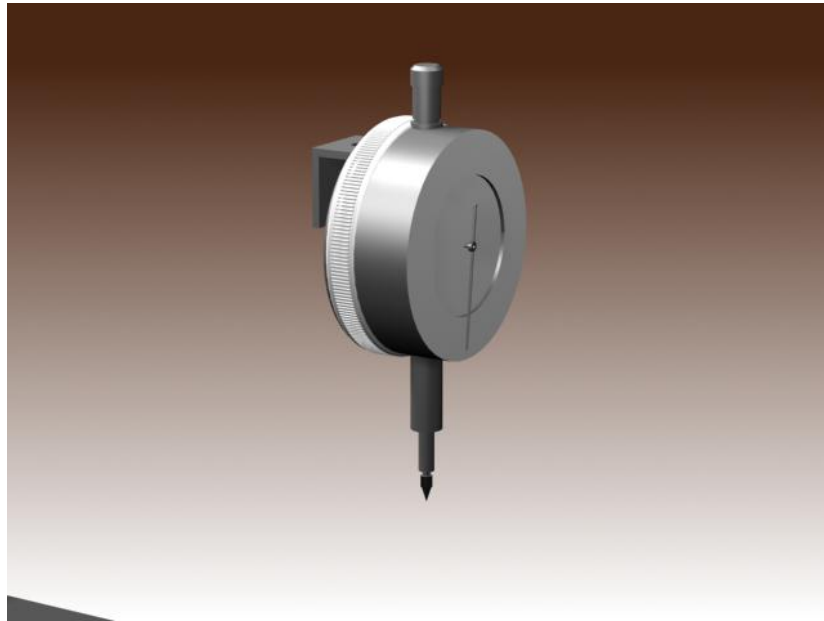


Fig4.183 Micrómetro y soporte

Una vez que se ha explicado el funcionamiento de la parte mecánica en su totalidad, se procede a realizar mediciones con un rugosímetro Standard en dos dimensiones de una empresa de la localidad, para comparar los datos obtenidos por este y por el construido en el desarrollo de esta tesis.

4.3 Rugosímetro construido

Una vez explicado el funcionamiento del rugosímetro en sus tres partes, es decir, la parte electrónica, el software y la mecánica, se presenta a continuación graficas del equipo terminado en su totalidad y sus componentes mencionados a lo largo de los capítulos anteriores.

Como se puede observar en la fig.4.184 el equipo está compuesto por cuatro partes físicas principales:

- Parte mecánica
- Caja de Control
- Control remoto
- Punta del Sensor

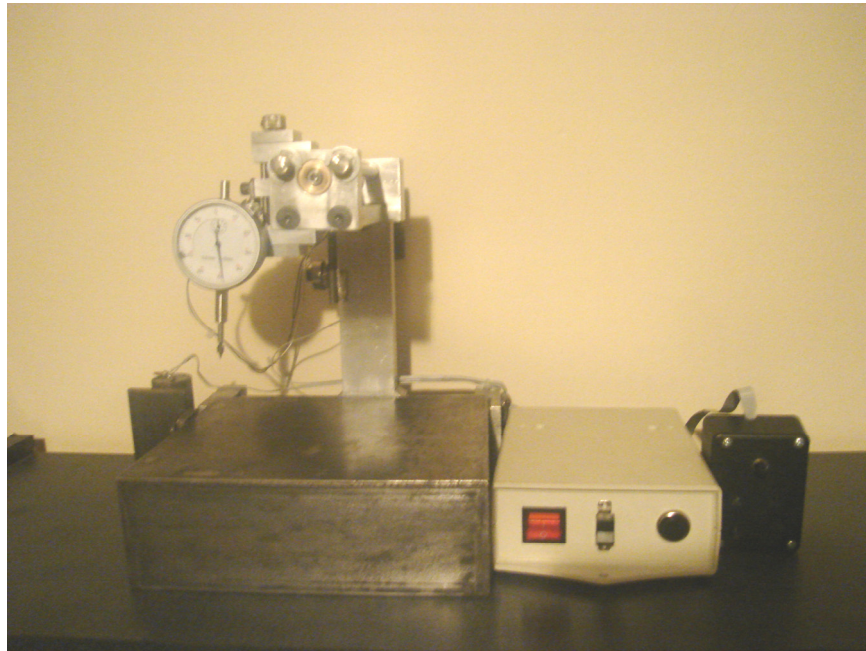


Fig.4.184 Rugosímetro y sus componentes

En puntos anteriores se trató a fondo el funcionamiento de la parte mecánica con figuras digitalizadas.

4.3.1 Caja de Control.- En la caja de control están colocadas las placas de mando y control del rugosímetro, así como también la fuente de alimentación, de manera que su manejo y transporte sea fácil y cómodo para el operador. Consta de elementos externos importantes que se detallan en la fig.4.185 a continuación:

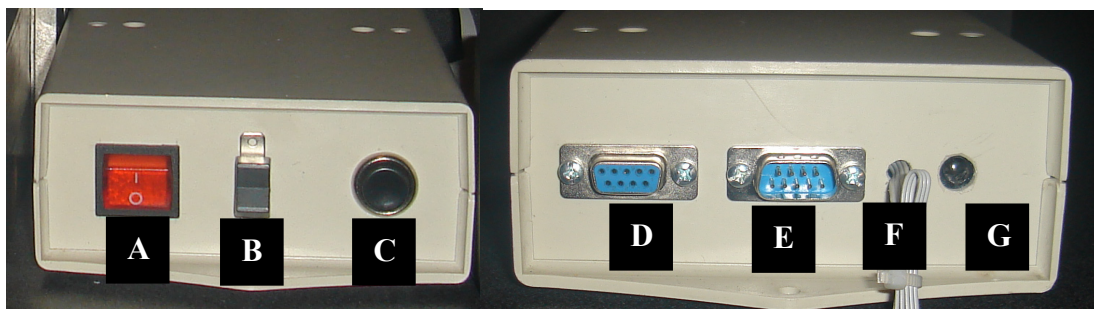


Fig.4.185 Rugosímetro y sus componentes

A) Interruptor de encendido.- Interruptor que permite el encendido y apagado de las tarjetas del equipo.

- B) Selector de opción.-** Selector con el cuál permite al operador establecer la orden al equipo de funcionar en forma manual para utilizar el control remoto o automática directamente con la PC.
 - C) Botón de Reset.-** Permite dar un reset al Microcontrolador desde el exterior sin necesidad de desarmar el equipo en caso de fallo u otro factor de emergencia.
 - D) Puerto serial.-** Conector DB9 por el cuál se comunica el microcontrolador de la tarjeta de mando con la PC
 - E) Conector para control remoto.-** Con este conector se da la posibilidad de conectar y desconectar el control remoto dependiendo del uso del mismo
 - F) Bus de datos.-** Lleva los datos rugosidad tomados por el micrómetro al microcontrolador y a su vez las órdenes de movimiento a los motores de los ejes del rugosímetro.
 - G) Alimentación.-** Conector de alimentación para transformador de 110 voltios a 12 voltios de corriente continua para alimentar a la fuente de alimentación del equipo.
- 4.3.2 Control remoto.-** Cuando se selecciona la opción manual en el Rugosímetro, se necesita un elemento externo que dirija a los motores a lo largo de los ejes y se tomen los datos de rugosidad, para esto, se construyó el control remoto que se muestra en la fig4.186

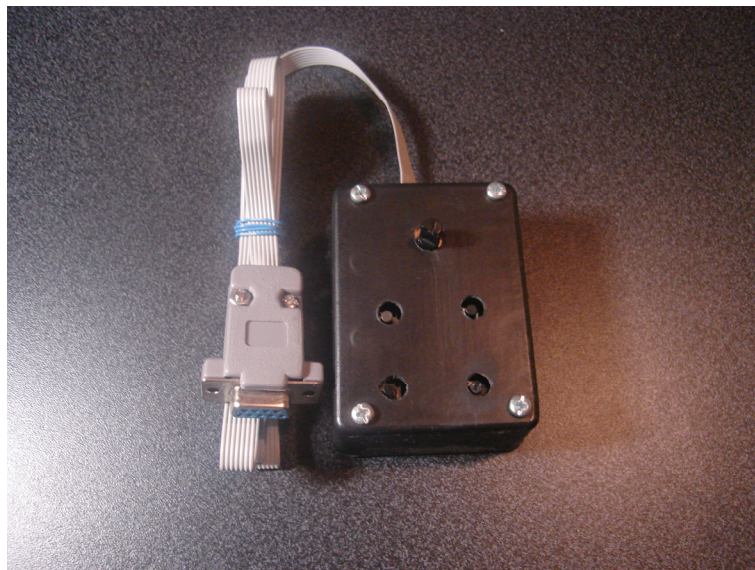


Fig4.186 Control Remoto

El control remoto consta de cuatro pulsantes que comandan de dos en dos a los motores tanto en el eje X como en el eje Y como ya se explicó en capítulos anteriores, así como también un potenciómetro que se encarga de controlar el nivel de PWM que será entregado a cada motor. La información es llevada por un bus de datos a un conector DB9 para su facilidad de conexión y desconexión de la caja de control.

4.3.3 Punta del Sensor.- La punta del sensor es muy importante en la toma de datos, por su tipo, composición, material, ancho de la punta, etc., de este dependerá la precisión del Rugosímetro y que tipo de rugosidad es capaz de medir. En la fig4.187 se puede observar la punta del sensor del micrómetro utilizada en el Rugosímetro.



Fig4.187 Punta del Sensor

Debido al tipo de la punta utilizada en el Rugosímetro, estará capacitado para medir rugosidades medianas y altas, ya que para medir rugosidades finas, se necesita una punta extremadamente fina la cual es de costo elevado. Las piezas metálicas que tienen rugosidades medias y altas son las que han sido maquinadas, debido a esto, las pruebas que se realizarán al rugosímetro se harán con piezas de este tipo.

4.4 Rugosímetro Patrón

Es importante tener un Rugosímetro Patrón con el cual comparar las mediciones que se realicen al construido en la presente tesis, debido a esto, se escogió al

Rugosímetro de la marca MITUTOYO tipo Surfest 301 que se presenta en fig4.188 a continuación:



Fig4. 188 Rugosímetro Patrón³²

Este Rugosímetro presenta las siguientes características de placa:

RANGO	eje z= 350 μm	eje x= 12.5 mm
VELOCIDAD	0.5 mm / s	
LARGO DEL CABLE	1 m	
PESO	190 g	
METODO DETECCION	inductiva	
RANGO	350 μm	
TIPO SENSOR	Diamante	

Tabla4.1 Características del Rugosímetro Patrón³³

El sensor que utiliza este rugosímetro le permite medir rugosidades altas, ya que la punta es de diamante, con radio de 5 μm y su resistencia es muy buena, lo cual lo hace excelente al momento de la toma de datos. El costo de este tipo de punto es muy elevado y de difícil acceso.

³² Fuente: Universidad Politécnica Salesiana Taller de Mecánica Industrial

³³ Fuente: <http://www.mitutoyo.com/surfest-sj-301.pdf> , Pag. 8

En la fig4.189 se puede observar al sensor de punta de diamante utilizado en el Rugosímetro Surfptest 301.

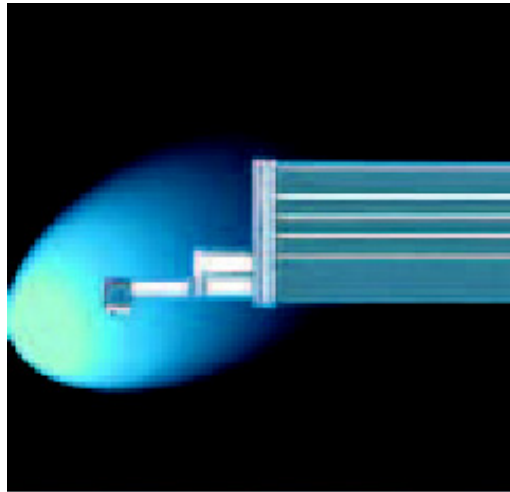


Fig4.189 Sensor Punta de Diamante³⁴

Surftest 301 tiene la capacidad de imprimir los datos que obtuvo al momento de la medición, conexión con la computadora vía RS232, pantalla LCD para menú, la fig.4.190 presenta un las dimensiones totales de este equipo a continuación:

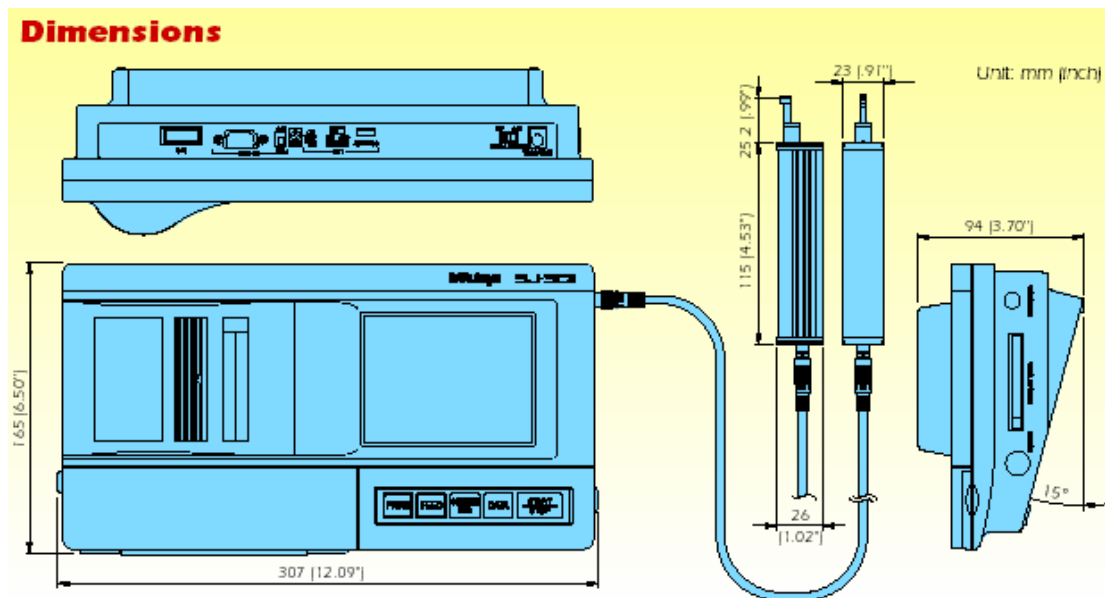


Fig4.190 Dimensiones Rugosímetro Surfptest 301³⁵

Como se puede observar, la figura nos muestra las dimensiones del equipo tanto en milímetros como en pulgadas.

³⁴ Fuente: <http://www.mitutoyo.com/surftest-sj-301.pdf>, Pag. 5

³⁵ Fuente: <http://www.mitutoyo.com/surftest-sj-301.pdf>, Pag 6

4.5 Mediciones de una pieza maquinada.

Como se mencionó con anterioridad, la punta del Rugosímetro construido sirve para medir rugosidades medias y altas, debido a esto, se realizará pruebas con una pieza maquinada como la que se presenta en la fig4.191 a continuación:

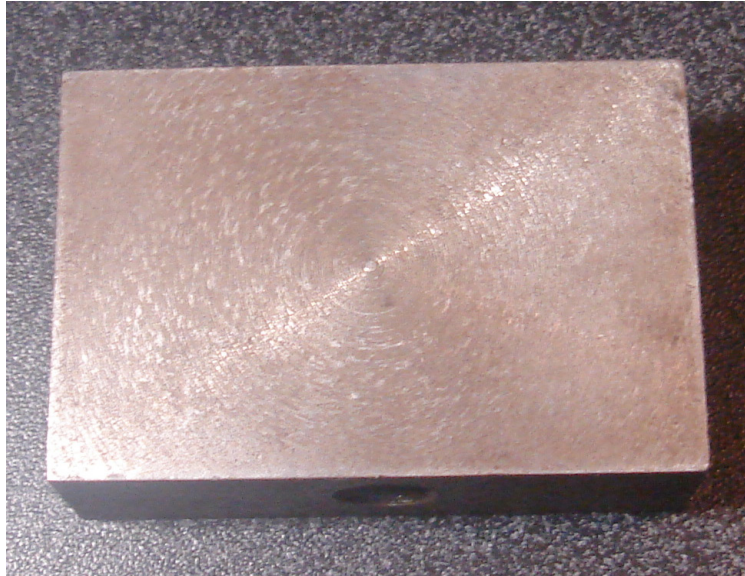


Fig4.191 Pieza maquinada

Como se puede observar en la figura, la superficie presenta un maquinado en forma circular debido al trabajo realizado para su terminado. El recorrido de la aguja del sensor tiene que hacerse perpendicularmente a las líneas que se generan al realizar el maquinado de la pieza; en caso de hacer una medición paralelamente con las líneas, la aguja puede entrar en un surco, lo cual nos daría una medición equivocada de la real rugosidad de la misma.

4.6 Pruebas Realizadas

4.6.1 Pruebas obtenidas por el Rugosímetro Patrón.- Se realizaron tres pruebas en piezas con diferentes tipos de maquinado, la cual presentó los siguientes valores:

DESCRIPCION	VALOR EN μm		
	PIEZA 1	PIEZA 2	PIEZA 3
Ra	4,72	3,82	1,04
Ry	35,9	27,3	9,1
Rz	30,3	20,8	8
Rp	16,2	13,9	3,5

Tabla4.2 Valores obtenidos por Rugosímetro Patrón

En la fig4.192 las piezas maquinadas en las cuales se realizaron las pruebas antes mencionadas:



Fig4.192 Piezas maquinadas

La gráfica obtenida en la primera prueba por el Rugosímetro Patrón se indica en la fig4.193 a continuación:

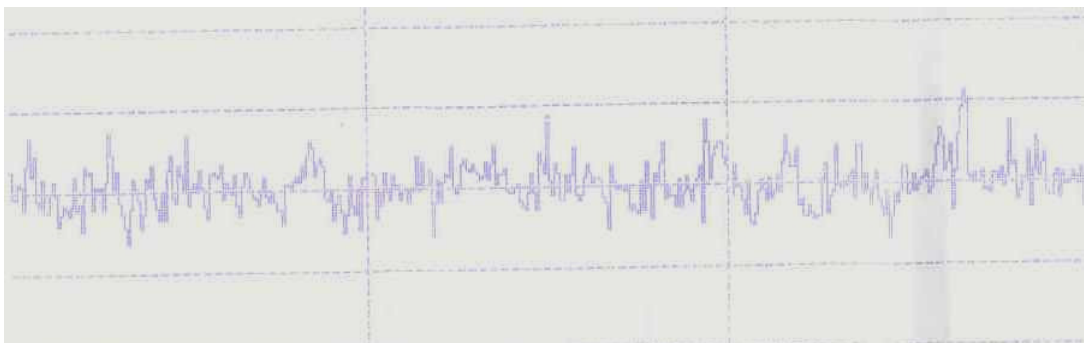


Fig4.193 Gráfica de perfil de la pieza maquinada

De igual forma, en la segunda medición se obtuvo un perfil el cual se muestra en la fig4.194 a continuación:



Fig4.194 Perfil de la pieza maquinada obtenida en la segunda medición

Por último, se presenta la gráfica del perfil obtenido en la tercera pieza como se muestra en la fig4.195

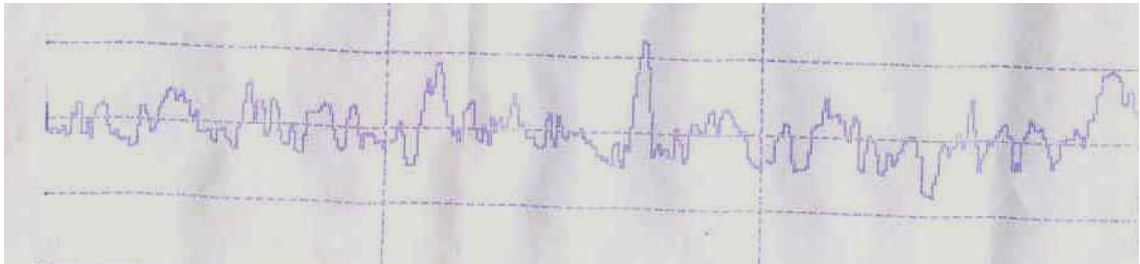


Fig4.195 Perfil de la pieza maquinada obtenida en la tercera medición

4.6.2 Pruebas obtenidas por el Rugosímetro construido.- De igual forma que en el Rugosímetro Patrón, se realizaron las pruebas respectivas con el Rugosímetro construido en la presente tesis, intentando localizar al sensor en el mismo punto en el cual partió la medida con la que se desea comparar.

Se debe tomar en cuenta, que en cada pieza se tomaron 5 mediciones, obteniendo a su vez un promedio de rugosidad para compararlo con el valor adquirido por el Rugosímetro Patrón.

Las mediciones obtenidas se presentan en la tabla 4.3 a continuación:

DESCRIPCION	PIEZA 1					
	Medición 1	Medición 2	Medición 3	Medición 4	Medición 5	Prom Total
Ra	4,3	4,1	4,5	5,3	4	4,44
Ry	32,7015	31,1805	34,2225	40,3065	30,42	33,7662
Rp	14,75846	14,07202	15,4449	18,19066	13,7288	15,2389

Tabla4.3 Valores obtenidos por Rugosímetro en la primera prueba

El perfil que se obtuvo con una distancia de 5mm fue el presentado en la fig 4.196 mostrada a continuación:

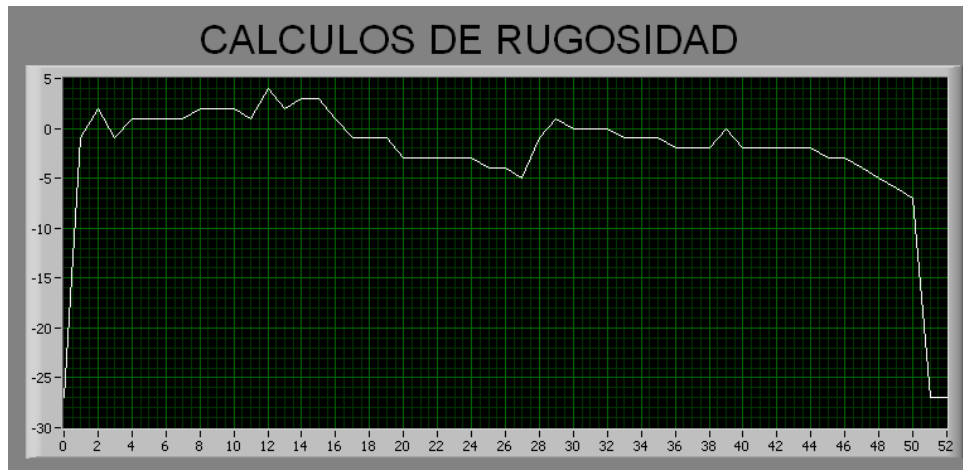


Fig4.196 Rugosidad en la primera medición

Los valores obtenidos en la segunda prueba se pueden observar en la tabla 4.4 a continuación:

DESCRIPCION	PIEZA 2					Prom Total
	Medición 1	Medición 2	Medición 3	Medición 4	Medición 5	
Ra	3,1	2,9	4,2	4	3,1	3,46
Ry	23,5755	22,0545	31,941	30,42	23,5755	26,3133
Rp	10,63982	9,95338	14,41524	13,7288	10,63982	11,87541

Tabla4.4 Valores obtenidos por Rugosímetro en la segunda prueba

De igual manera, se obtiene un perfil de superficie el cual se muestra en la fig.4.197:

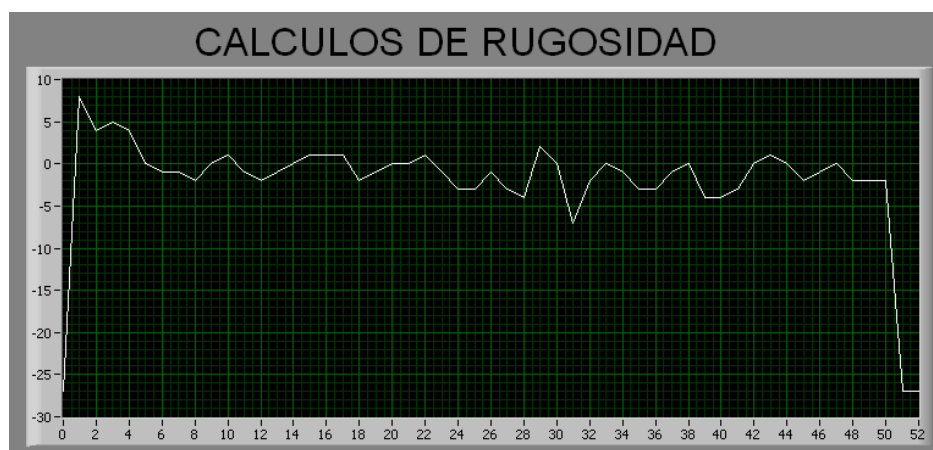


Fig4.197 Perfil de la segunda medición

Por último se presenta la tabla 4.5 en la que muestra los resultados obtenidos en la tercera pieza a continuación:

DESCRIPCION	PIEZA 3					
	Medición 1	Medición 2	Medición 3	Medición 4	Medición 5	Prom Total
Ra	0,8	2	1,7	2,2	1,9	1,72
Ry	6,084	15,21	12,9285	16,731	14,4495	13,0806
Rp	2,74576	6,8644	5,83474	7,55084	6,52118	5,903384

Tabla4.5 Valores obtenidos por Rugosímetro en la tercera prueba

El perfil obtenido se puede visualizar en la fig4.198 a continuación:

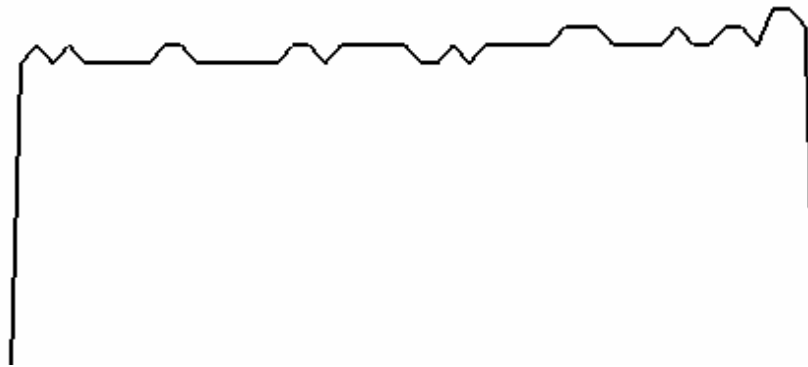


Fig4.198 Perfil de la tercera medición

Una vez obtenidas todas las mediciones, se realizó una tabla de errores con respecto a los datos obtenidos por el Rugosímetro Patrón, lo cual presentó los resultados que se muestran en la tabla 4.6 a continuación:

TABLA DE ERRORES			
PIEZA 1			
Descripción	Patrón	Medido	%Error
Ra	4,72	4,44	5,9
Ry	35,9	33,7	6,1
Rp	16,2	15,2	6,2
PIEZA 2			
Descripción	Patrón	Medido	%Error
Ra	3,82	3,46	9,4
Ry	27,3	26,31	3,6
Rp	13,9	11,87	14,6
PIEZA 3			
Descripción	Patrón	Medido	%Error
Ra	1,04	1,72	-65,4
Ry	9,1	13,08	-43,7
Rp	3,5	5,9	-68,7

Tabla 4.6 Tabla de errores en las mediciones

4.7 Conclusiones de resultados

Como conclusiones de las mediciones obtenidas en el equipo se pueden tomar las siguientes:

- En los equipos del mercado, el interés más importante con respecto a la medición es el enfoque a la Desviación media aritmética del perfil Ra, dato que el Rugosímetro construido tiene una aproximación casi exacta, con un error de porcentaje bastante bajo. Hay que recalcar que en una misma pieza, ningún equipo comercial logra obtener igual valor de Ra, aun así se los coloque en la misma superficie de muestra. Los valores de Ra siempre van a diferir en su resultado pero siempre se aproximarán.
- Fue necesario obtener varias mediciones en una misma superficie con el fin de adquirir un promedio de rugosidad para poder compararlo con el dato del Rugosímetro Patrón y verificar el margen de error que se tiene en cada tipo de rugosidad, ya que con una sola medición no sería posible, debido a que no se puede localizar con exactitud el lugar preciso en el cual el Rugosímetro Patrón fue colocado.
- Se puede verificar en la tabla 4.6 que a medida que los factores Ra, Ry y Rp en la piezas disminuyen, el error aumenta, esto se debe a que si la rugosidad tiende a bajar, la punta del sensor se hace menos sensible, lo que aumenta el error en la medición ya que dicha punta no es la correcta para rugosidades muy bajas.

CONCLUSIONES Y RECOMENDACIONES

En este punto, es formal que se presente las diferentes puntualizaciones que se han ido exponiendo durante el desarrollo de la presente Tesis de Ingeniería, las mismas que están ordenadas según es esquema presentado.

a. Conclusiones Técnicas:

- Es esencial conocer a fondo los conceptos básicos de Rugosidad y sus parámetros, ya que de estos dependerán los resultados. Al momento de obtener una curva de superficie, mediante la teoría se pueden calcular los parámetros de rugosidad y el nivel real de dicha curva.
- En el diseño del circuito, hay que prestar mucha importancia en la toma de la señal que se va a adquirir, ya que de esta depende la correcta selección de todos los esquemas y componentes que se utilizarán en el diseño del equipo.
- Para evitar el ruido por inducción electromagnética y por acoplamiento capacitivo, se separó al circuito en tres placas siendo estas: circuito de alimentación, de mando y de fuerza, alimentando las dos últimas por separado.
- La utilización del software Pic Basic Pro 2.44 simplificó en tamaño la programación del microcontrolador, teniendo en cuenta que para trabajar con el mismo, se deben tener bases sólidas del lenguaje ensamblador.
- La programación bajo el lenguaje G que posee el software Labview fue de gran ayuda en el desarrollo de la presente tesis ya que este programa esta enfocado al desarrollo de equipos de precisión.
- Debido a que el sensor está adaptado a un reloj comparador, el sistema mecánico para transportar en los ejes tiene que ser preciso, sin juego en los tornillos ni inclinaciones de ningún tipo, por lo que las piezas con las que se fabrique deben ser de primera calidad.
- Para contrastar la fiabilidad del equipo, se hizo 5 mediciones en diferentes puntos de una pieza, en tres muestras con distintas rugosidades cada una, promediando estos resultados para obtener un equivalente total el cuál se

comparó con el obtenido por un Rugosímetro Patrón, con el fin de adquirir el porcentaje de error en las medidas que el equipo presenta.

Luego de realizar estas mediciones se concluye que:

- El equipo funciona correctamente en piezas con rugosidades medias y altas con un margen de error aceptable como se puede observar en la tabla 4.6.
- En rugosidades bajas, las mediciones dejan de ser confiables, debido a que se tiene un error de más del 50%, esto se debe a que la punta que se utiliza no es la apropiada. Para este tipo de rugosidades, se debe utilizar una punta de menor diámetro y mayor dureza con lo que se asegura una mejor toma de datos.
- Se comprueba que el equipo está diseñado para medir rugosidades de piezas maquinadas por la limitante que existe en el tipo de sensor que se está utilizando
- El equipo está en condiciones de mejorar la medición, ya que tiene la capacidad de poder adaptarse a otros sensores de mayor resolución, con lo que se lograría medir rugosidades bajas existentes, por ejemplo, en piezas rectificadas.

b. Otras conclusiones importantes:

- los contenidos de las asignaturas recibidas durante la formación académica han sido de vital importancia para desarrollar la presente tesis, tanto en el aspecto técnico como humano.
- Por último, los autores consideramos que el desarrollo de una tesis de graduación implica no solamente la aplicación de los conocimientos adquiridos en las aulas universitarias sino la oportunidad de demostrar que podemos ser constructores de conocimiento, lo que queda demostrado con el presente trabajo.

En concordancia con estas conclusiones se puede recomendar lo siguiente:

a. Recomendaciones Técnicas:

- Antes de diseñar el circuito impreso, deben adquirirse todos los componentes que se van a utilizar en la placa, para evitar problemas en formas, diámetros de agujeros, tamaños de elementos, corrientes de consumo, de manera que el producto final sea una placa PCB óptima.
- Se tiene que elaborar la placa a dos caras para optimizar espacio entre componentes y reducir el tamaño del equipo en general y colocar antisolder en las mismas que las protege de la corrosión.
- En caso de encontrar algún error en el software Labview, utilizar el método paso a paso el cuál facilita sobre manera encontrar fallas en programas extensos y complicados.
- Trabajar con un ordenador de características Pentium 4 512 M de Ram mínimo, para el correcto desempeño de Labview, caso contrario, este se hará muy pesado y lento.
- Al momento de ensamblar la parte mecánica hay que tener mucho cuidado en no torcer las guías, ya que esto puede causar que los motores hagan esfuerzo innecesario y posiblemente puedan quemarse
- Los piñones de la desmultiplicación, no deben estar apretados entre sí, ya que los dientes pueden romperse.
- Para mejorar la resolución del equipo, se recomienda construir o adquirir una micropunta de las características como se muestra en la tabla 4.7 (para más especificaciones ver Anexo D), en caso de no construirla, se puede comprar todo el sensor.

Características de la punta	
material	diamante
radio de la punta	5 m
distancia recorrido	40 mm
masa	18 g
presión	4mN
Características del sensor	
método medición	sensor inductivo con salida diferencial
escala	350 m 200 m a + 150 m

Tabla 4.7 Características del sensor de diamante

b. Otras recomendaciones:

- Para un mejor conocimiento por parte de los estudiantes, es necesario que la Universidad adquiriera licencias de simulación o paquetes profesionales de programas de última tecnología, de manera que la formación profesional de sus estudiantes sea óptima.
- En el pensum de estudios de todas las facultades debería constar con obligatoriedad Investigación Científica, sus métodos y técnicas, de manera que facilite el proceso de investigación para la tesis y una correcta redacción y presentación del informe final.
- Debería propiciarse un mayor número de visitas técnicas a industrias e instituciones relacionadas con los contenidos científicos para que exista un acercamiento efectivo entre los estudiantes y el mundo del trabajo.

Bibliografía

Referencias Bibliográficas

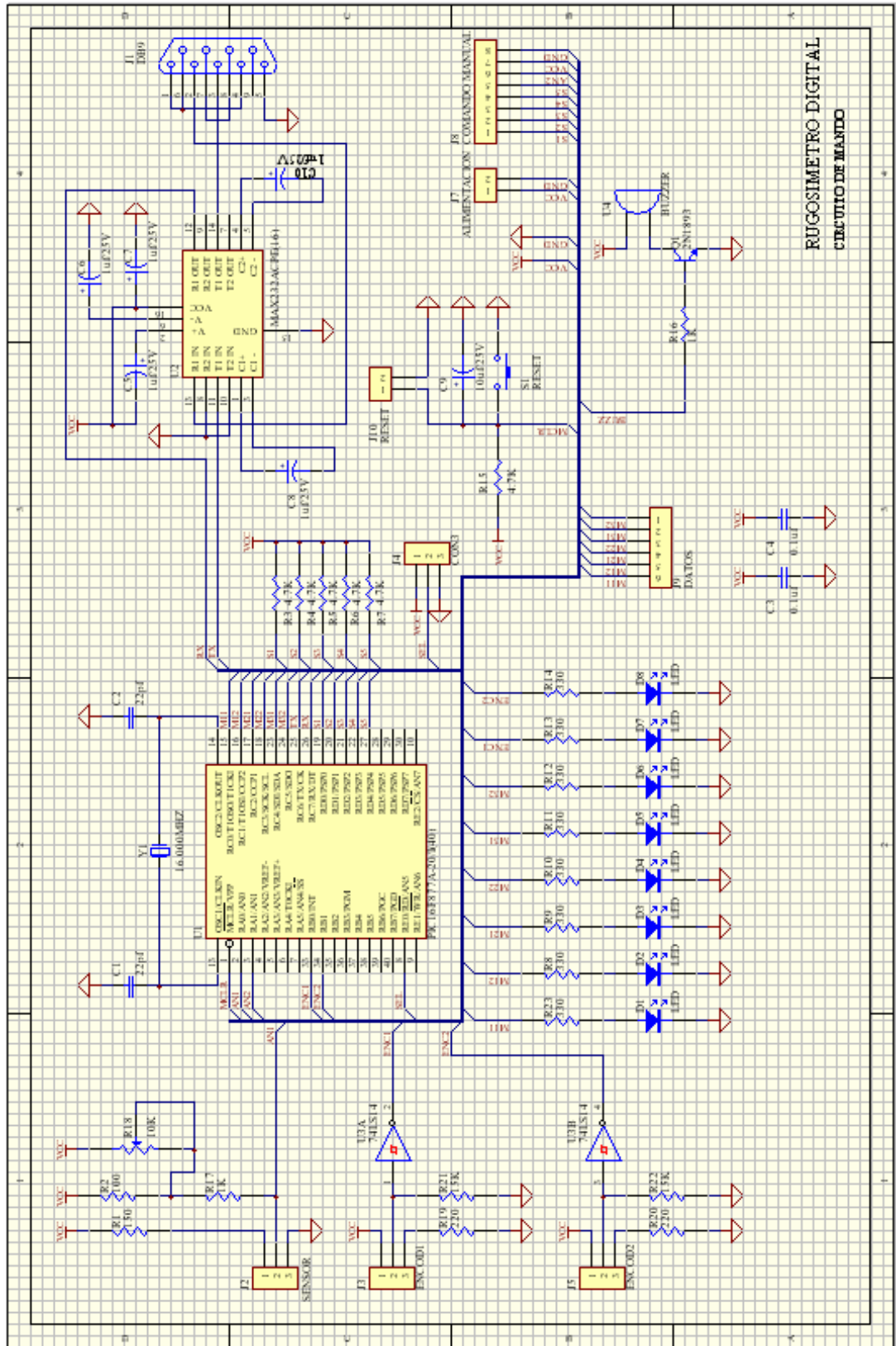
- ANTONI Manuel, DOMINGO Biel, OLIVÉ Joaquín, PRAT Jordi, Francesc J. SÁNCHEZ, "Instrumentación Virtual" Adquisición, procesado y análisis de señales, Alfaomega 2000.
- BATURONE Aníbal Ollero, "Robótica" Manipuladores y robots móviles, Alfaomega-Marcombo 2001.
- CORRALES V Santiago., "Electrónica Práctica con microcontroladores PIC".
- F.R. BAHÍA BLANCA, Electrónica Aplicada II.
- GABRIUNAS V. Apuntes de Electrónica. Universidad Distrital “Francisco José de Caldas”. 1999.
- HINOJOSA Moisés, MELO Martín, La Rugosidad de las Superficies: Topografía
- J.E. GARCÍA SÁNCHEZ; ED. TEBAR-FLORES "Circuitos y Sistemas Digitales".
- J.M. Angulo "Microprocesadores de 16 bits” interconexión de periféricos a uP". REF: No esta en biblioteca.
- LÓPEZ, F.E., CAVAZOS, R., DELGADO, M., “Caracterización de superficies maquinadas por medio de parámetros de rugosidad”, *Ingenierías*, Vol. 6, (2003).
- MANRIQUE, E., Casanova, A Metrología básica Edebé.
- MANUEL TORRES Portero, "Diseño e ingeniería electrónica asistida por ordenador en PROTEL".
- MARTIN BOLTON. ADDISON WESLEY, "Digital Systems Design with Programmable Logic". REF: No está en biblioteca.
- NIKITAS A. Alexandridis "Microprocessor Systems Design Concepts",. REF: No esta en biblioteca.
- PÉREZ García Juan C, Miguel A., ALVAREZ Antón, RODRÍGUEZ Juan C. Campo, FERRERO Fco. Javier Martín, Gustavo J. Grillo Ortega, "Instrumentación Electrónica" Thomson 2004.
- PORTELA Vicente, Curso de Metrología Dimensional. E.T.S.I.I. de Madrid 1978.

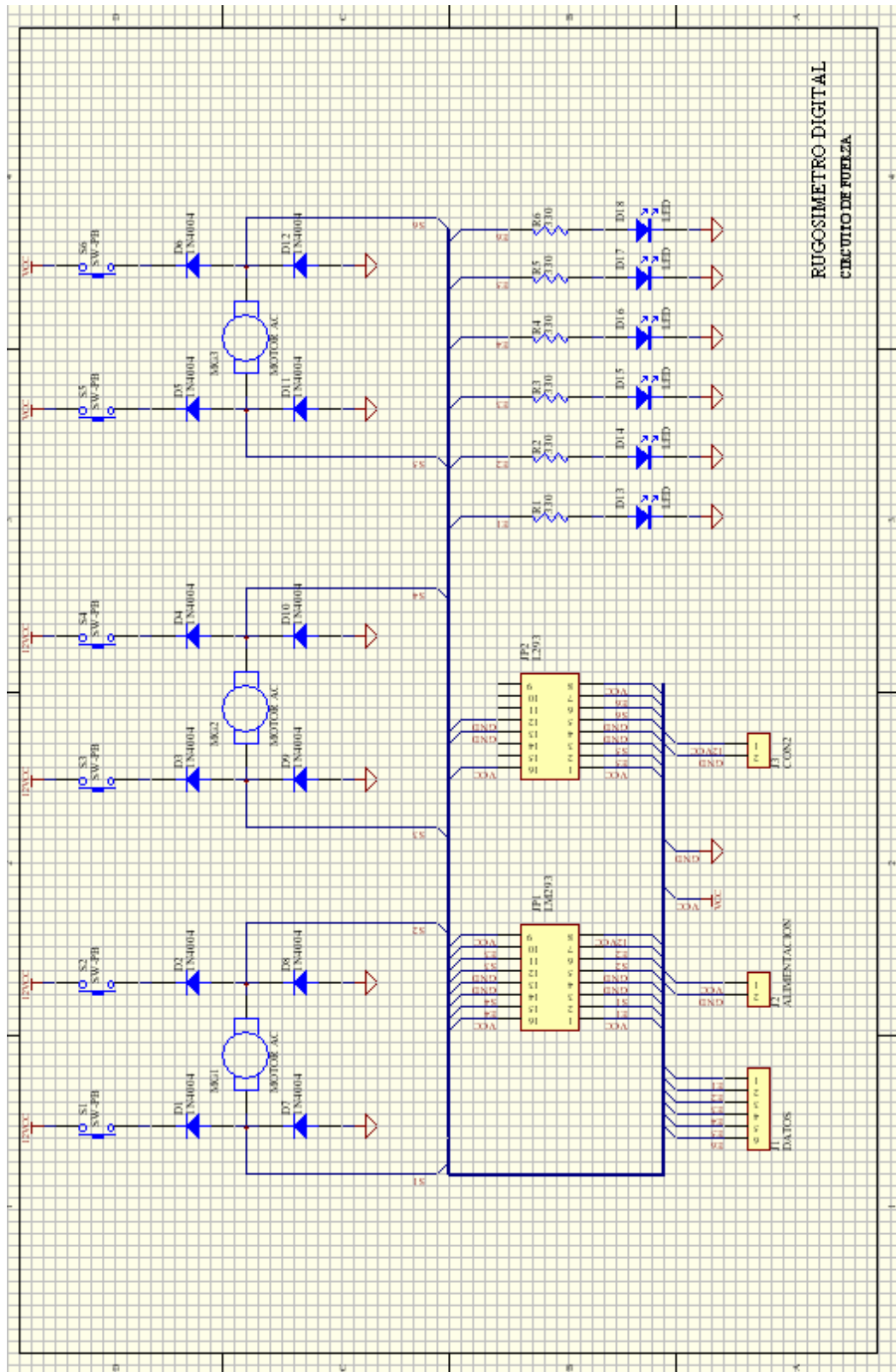
- REYES Carlos A., "Aprenda rápidamente a programar Microcontroladores PIC", Gráficas Ayerve 2004.
- RONCANCIO H., VELASCO. H. Una Introducción a Labview. Semana de Ingenio y Diseño. Universidad Distrital "Francisco José de Caldas". 2000.
- TOCCI J. Ronald, "Sistemas Digitales" Principios y Aplicaciones, Prentice Hall 1996
- TORRES Fernando, POMARES Jorge, GIL Pablo, PUENTE Santiago T., ARACIL Rafael, "Robots y Sistemas Sensoriales", Prentice Hall 2002.

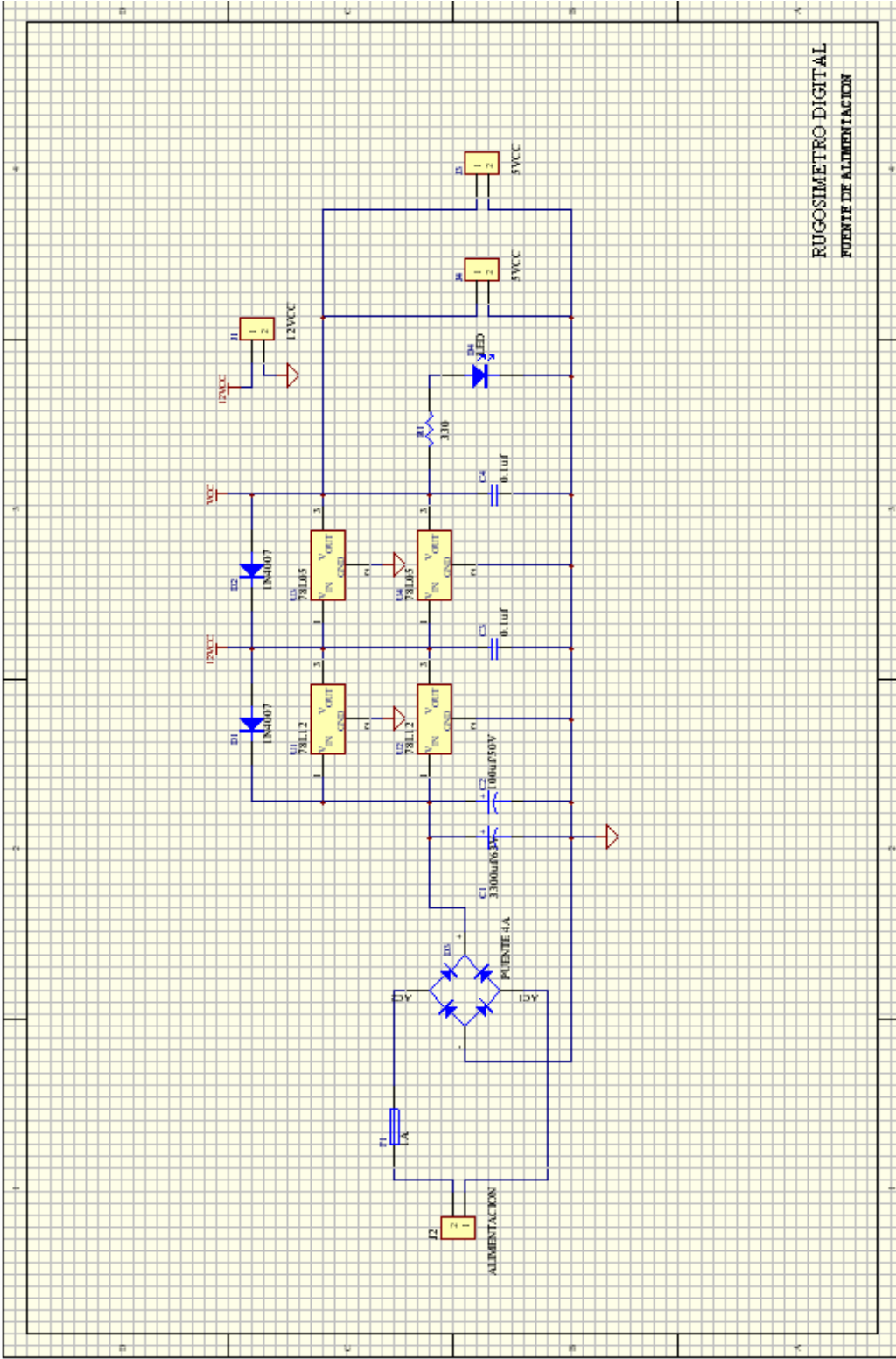
Referencias Electrónicas

- www.microchip.com
- www.todopic.com.ar
- www.maxim.com
- www.ni.com
- www.micros1.com.ar/picbasic.htm
- www.mecanique.co.uk
- www.ic-prog.com
- www.talquingelectronics.com
- www.sonsivri.com
- www.melabs.com
- www.rentron.com
- www.frino.com.ar
- www.redeya.com
- www.x-robotics.com
- www.cipres.cec.uchile.cl
- www.pablin.com.ar
- www.electronicaestudio.com
- www.mikroelektronika.com

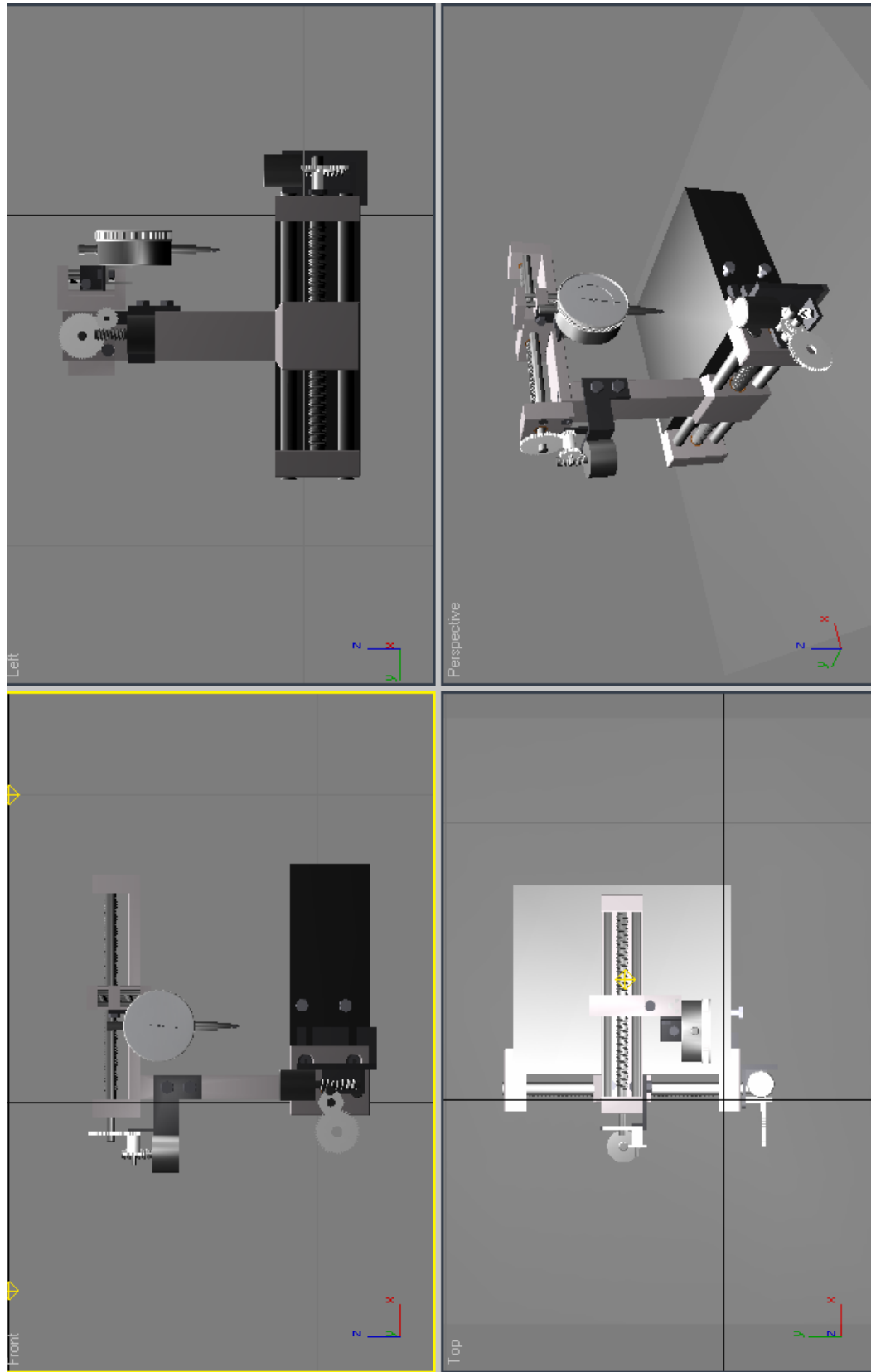
ANEXO A







ANEXO B



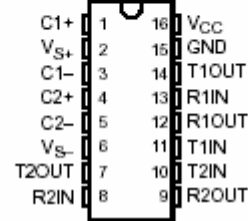
ANEXO C

MAX232, MAX232I
DUAL EIA-232 DRIVERS/RECEIVERS

SLS0471 – FEBRUARY 1999 – REVISED OCTOBER 2002

- Meet or Exceed TIA/EIA-232-F and ITU Recommendation V.28
- Operate With Single 5-V Power Supply
- Operate Up to 120 kbit/s
- Two Drivers and Two Receivers
- ± 30 -V Input Levels
- Low Supply Current . . . 8 mA Typical
- Designed to be Interchangeable With Maxim MAX232
- ESD Protection Exceeds JESD 22 – 2000-V Human-Body Model (A114-A)
- Applications
 - TIA/EIA-232-F
 - Battery-Powered Systems
 - Terminals
 - Modems
 - Computers

MAX232 . . . D, DW, N, OR NS PACKAGE
MAX232I . . . D, DW, OR N PACKAGE
(TOP VIEW)



description/ordering information

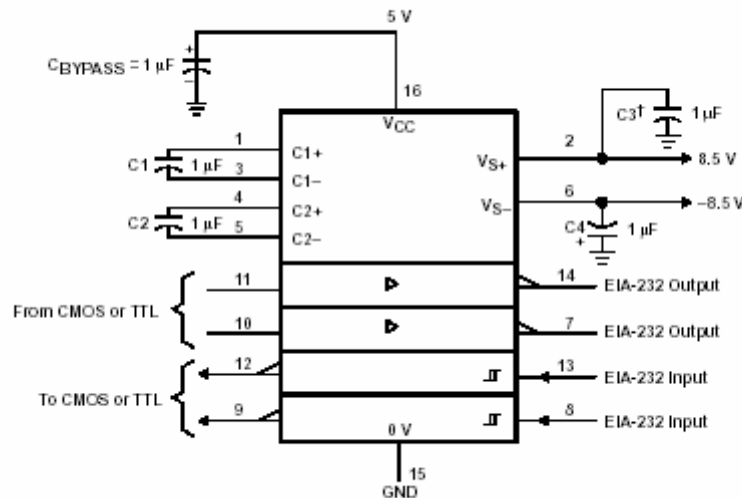
The MAX232 is a dual driver/receiver that includes a capacitive voltage generator to supply EIA-232 voltage levels from a single 5-V supply. Each receiver converts EIA-232 inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V and a typical hysteresis of 0.5 V, and can accept ± 30 -V inputs. Each driver converts TTL/CMOS input levels into EIA-232 levels. The driver, receiver, and voltage-generator functions are available as cells in the Texas Instruments LinASIC™ library.

ORDERING INFORMATION

TA	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	PDIP (N)	Tube	MAX232N	MAX232N
	SOIC (D)	Tube	MAX232D	MAX232
		Tape and reel	MAX232DR	
	SOIC (DW)	Tube	MAX232DW	MAX232
		Tape and reel	MAX232DWR	
	SOP (NS)	Tape and reel	MAX232NSR	MAX232
-40°C to 85°C	PDIP (N)	Tube	MAX232IN	MAX232IN
	SOIC (D)	Tube	MAX232ID	MAX232I
		Tape and reel	MAX232IDR	
	SOIC (DW)	Tube	MAX232IDW	MAX232I
		Tape and reel	MAX232IDWR	

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

APPLICATION INFORMATION



† C3 can be connected to VCC or GND.



PIC16F870/871

28/40-Pin, 8-Bit CMOS FLASH Microcontrollers

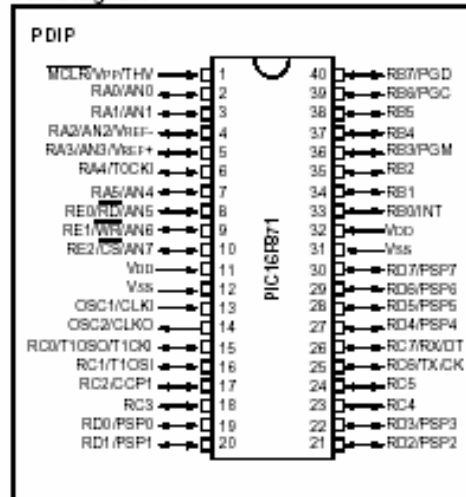
Devices Included In this Data Sheet:

- PIC16F870
- PIC16F871

Microcontroller Core Features:

- High performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two-cycle
- Operating speed: DC - 20 MHz clock input
DC - 200 ns instruction cycle
- 2K x 14 words of FLASH Program Memory
128 x 8 bytes of Data Memory (RAM)
64 x 8 bytes of EEPROM Data Memory
- Pinout compatible to the PIC16CXXX 28 and 40-pin devices
- Interrupt capability (up to 11 sources)
- Eight level deep hardware stack
- Direct, Indirect and Relative Addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low power, high speed CMOS FLASH/EEPROM technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP™) via two pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial and Industrial temperature ranges
- Low power consumption:
 - < 1.6 mA typical @ 5V, 4 MHz
 - 20 µA typical @ 3V, 32 kHz
 - < 1 µA typical standby current

Pin Diagram

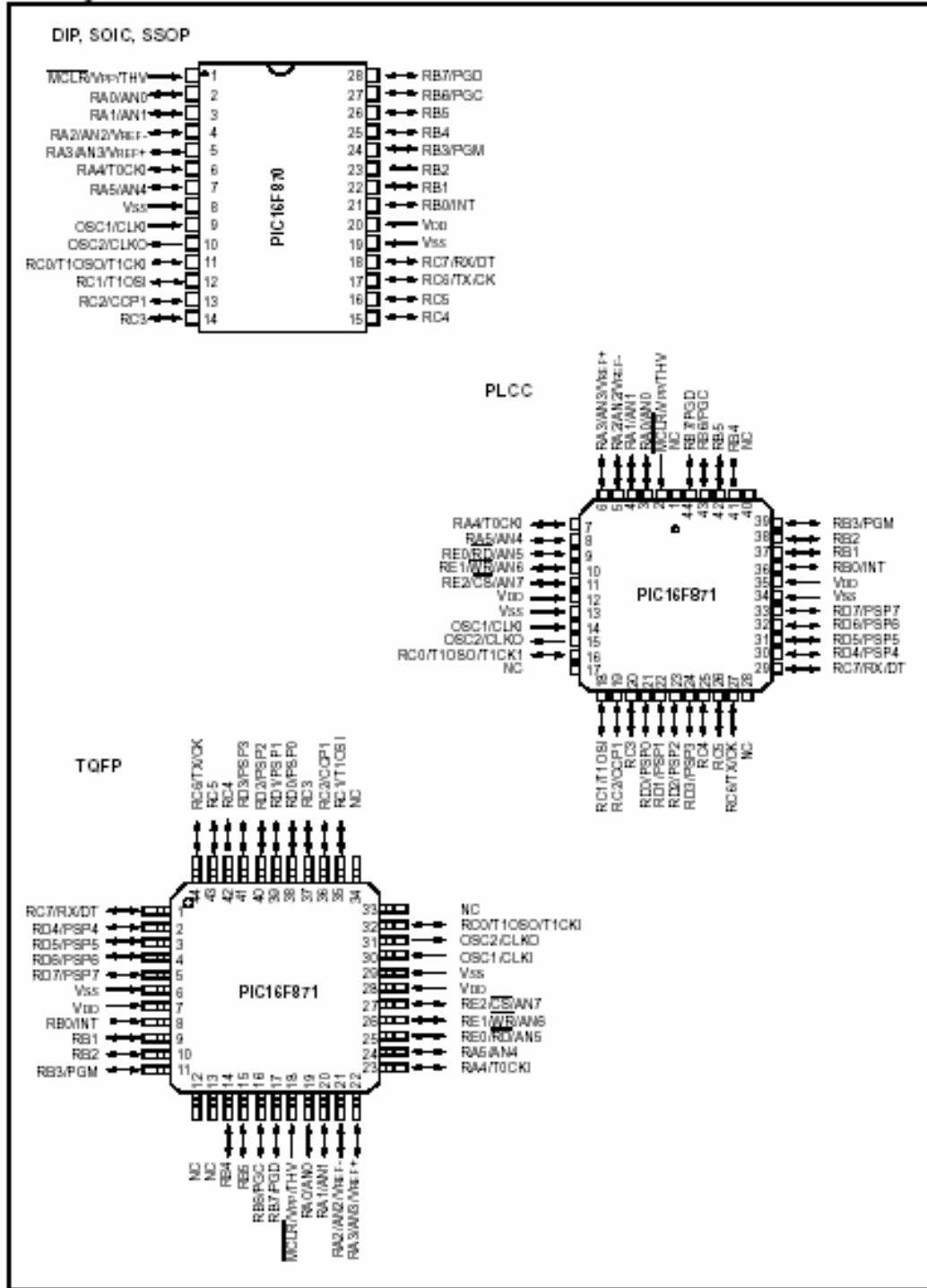


Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during SLEEP via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- One Capture, Compare, PWM module
 - Capture is 16-bit, max. resolution is 12.5 ns
 - Compare is 16-bit, max. resolution is 200 ns
 - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) 8-bits wide, with external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

PIC16F870/871

Pin Diagrams



ANEXO D

GEOMETRÍA DE LOS TIPOS DE PUNTA

	Punta lanceolada
	Punta en talla de diamante
	Punta redonda
	Punta cortante
	Micropunta
Características de la punta	
material	diamante
radio de la punta	5 m
distancia recorrido	40 mm
masa	18 g
presión	4mN

CARACTERÍSTICAS DEL SENSOR



Características del sensor	
método medición	sensor inductivo con salida diferencial
escala	350 m 200 m a + 150 m

