ESCUELA POLITÉCNICA DEL EJÉRCITO
SEDE – LATACUNGA

CARRERA DE INGENIERÍA
ELECTRÓNICA E INSTRUMENTACIÓN

PROYECTO DE GRADO PARA LA OBTENCIÓN DEL TÍTULO EN INGENIERÍA ELECTRÓNICA E INSTRUMENTACIÓN

"DISEÑO Y CONSTRUCCIÓN DE CONTADORES DE PRODUCCIÓN PARA LAS CARDAS C4/0 DE LA SECCIÓN DE APERTURA Y MONITOREO DE LOS CONTADORES MEDIANTE UNA RED DE MICROCONTROLADORES EN LA PLANTA TEXTILERA "RÍO BLANCO"

EDWIN PATRICIO PRUNA PANCHI

CRISTIAN SEBASTIÁN VIERA ZAMBRANO

LATACUNGA, NOVIEMBRE DEL 2007

CERTIFICACIÓN

Certificamos que el presente proyecto de grado fue elaborado en su totalidad por los señores:

Edwin Patricio Pruna Panchi.
Cristian Sebastián Viera Zambrano.

Bajo nuestra dirección, como requisito para la obtención del título de Ingeniero Electrónico en Instrumentación.

_____

Ing. Julio Acosta.
DIRECTOR DEL PROYECTO

_____

Ing. Galo Ávila.
CODIRECTOR DEL PROYECTO

LATACUNGA, NOVIEMBRE DEL 2007

ELABORADO POR:

Edwin Patricio Pruna Panchi

Cristian Sebastián Viera Zambrano

ING. ARMANDO ÁLVAREZ
COORDINADOR DE CARRERA

DR. EDUARDO VÁSQUEZ ALCÁZAR
SECRETARIO ACADÉMICO

AGRADECIMIENTO

Este trabajo no hubiera sido posible sin la colaboración y el apoyo de varias personas e instituciones. Agradecemos a todos aquellos, que voluntaria e incondicionalmente han sido nuestros maestros tanto en lo académico como en lo personal.

Por otro lado queremos agradecer al Departamento de Ingeniería Electrónica e Instrumentación, de la Escuela Politécnica del Ejército, que de forma directa o indirecta, han aportado para que se finalice de manera satisfactoria el proyecto.

Pues también queremos agradecer a las personas que nos brindaron su apoyo a: nuestras familias y amigos más cercanos, ya que permanecieron en todo momento para que se pueda culminar el proyecto sin ningún contratiempo. Sin aportar ningún nombre en especial, queremos agradecer a todos los buenos consejos, la paciencia demostrada, así como la convivencia y el intercambio de ideas, lo cual siempre es reconfortante y por otro lado beneficioso para la realización de un trabajo de estas características.

Un agradecimiento especial para todas las personas que han contribuido en la tesis, aportando su experiencia para la evaluación del trabajo, así como contribuyendo desinteresadamente mediante su opinión crítica y constructiva. Gracias por los consejos y por vuestro tiempo.

Edwin, Sebastián

DEDICATORIA

Este trabajo está dedicado a Dios, a mis padres (Raúl y Beatriz) por el apoyo incondicional que me dieron en el transcurso de mi vida estudiantil, también una dedicatoria especial a mi hijo Esteban y a Ivón por todo el cariño y amor que supieron darme durante mi trayectoria académica. Y a todas las personas buenas que estuvieron junto a mí en todo momento.

Gracias.

Que Dios los bendiga.

Edwin Patricio Pruna Panchi.

Mi tesis se la dedico con todo amor, cariño y respeto:

A dios por darme la oportunidad de vivir y darme una familia maravillosa.

A mis padres que me dieron la vida por su comprensión y ayuda en momentos buenos y malos. Me han enseñado a encarar las adversidades sin perder nunca la dignidad ni desfallecer en el intento. Me han dado todo lo que soy como persona, mis valores, mis principios, mi perseverancia y mi empeño, y todo ello con una gran dosis de amor y sin pedir nunca nada a cambio.

A la memoria de mí querida y siempre recordada abuelita Esther, que aunque no se encuentre a mi lado, siempre la siento y la llevo dentro de mi corazón.

A mis hermanas y hermanos por sus sabios consejos y ejemplo de superación.

A todas aquellas personas quienes de una u otra forma hicieron posible que una de las metas de mi vida hoy se encuentre plasmada.

A todos ellos,
Muchas gracias de todo corazón.

Cristian Sebastián Viera. Z

CONTENIDO

INTRODUCCIÓN **X**

CAPITULO I

FUNDAMENTOS.....1

1.1 Descripción del problema…….…1
1.2 Fundamentos Físicos del proceso de hilatura…….....2
1.2.1 Introducción……2
1.2.2 Tipos de fibras utilizadas para el proceso de hilatura……..2
1.2.3 Estudio de la Máquina RIETER serie EZA1……..6
1.2.4 Pasos del proceso de hilatura……..8
1.3 Estudio de las redes de comunicación……..…...13
1.3.1 Introducción al protocolo RS232……13
1.3.2 Tipos de comunicaciones digitales……14
1.3.3 El puerto serial de una PC……...16
1.3.4 La interfase RS485……...20
1.3.5 Acondicionamiento de señales……23
1.3.6 Sensores inductivos……..24
1.4 Estudio de los microcontroladores……26
1.4.1 Características generales de los microcontroladores……26
1.4.2 Características……...29
1.4.3 Lenguajes de programación……30

CAPITULO II

ANÁLISIS Y DISEÑO DE LOS CONTADORES
Y MONITOREO A TRAVÉS DE LA RED INDUSTRIAL……...32

2.1 Descripción del software de programación de los µC…….32
2.1.1 Introducción al Protón IDE……...32
2.1.2 Síntesis del manejo del software IC – PROG……...74
2.2 Diseño del sistema de monitoreo y control del proceso……...77
2.2.1 Diseño e implementación del hardware…….77
2.2.2 Descripción del hardware del sistema……79
2.2.3 Descripción de la plataforma utilizada para la Implementación del software……...83
2.3 Implementación del software…….101
2.3.1 Programación del microcontrolador PIC 16F877A……101
2.3.2 Interfase gráfica……...101

CAPITULO III

ANALISIS DE RESULTADOS…….……102

3.1 Descripción funcional…….102
3.1.1 Módulo de monitoreo……..103
3.1.2 Cable de comunicación……..106
3.2 Pruebas del sistema……...107
3.2.1 Prueba funcional del software……...108
3.2.2 Prueba funcional del hardware…….110
3.2.3 Prueba en modo local……111
3.3 Alcances y limitaciones……..112
3.3.1 Alcances……112
3.3.2 Limitaciones……..113

CAPITULO IV

CONCLUSIONES Y RECOMENDACIONES…….114

4.1 Conclusiones…….…….114
4.2 Recomendaciones……..116

BIBLIOGRAFÍA.....……119

GLOSARIO. ……120

ANEXOS…..……122

ANEXO A: DESCRIPCIÓN DE LOS CIRCUITOS INTEGRADOS
ANEXO B: DIAGRAMAS DE CONEXIONES
ANEXO C: CIRCUITOS IMPRESOS
ANEXO D: DIAGRAMAS DE FLUJO
ANEXO E: PROGRAMACIÓN REALIZADA EN LABVIEW
ANEXO F: MANUAL DE USUARIO

INTRODUCCIÓN

En la actualidad los microcontroladores son la herramienta fundamental para el desarrollo científico y tecnológico, saber sobre estas mini computadoras es muy importante ya que se encuentran en todas partes.

El Capitulo I se trata sobre los fundamentos físicos del proceso de hilatura, los tipos de fibras utilizadas en el mismo, el estudio de las máquinas en las que se implementaron los contadores; así como los pasos del proceso de hilatura. El estudio de las redes de comunicación, también se realizó el estudio de los sensores, microcontroladores y se detallan los lenguajes de programación.

El Capitulo II se indica la descripción del software de programación de los microcontroladores, además se trata sobre el diseño e implementación del hardware y software del proyecto.

El Capitulo III se indica sobre la descripción funcional del proyecto, las pruebas realizadas al sistema y los alcances y limitaciones del mismo.

En el capítulo IV se encuentran las conclusiones y recomendaciones obtenidas durante la ejecución de éste proyecto.

CAPITULO I

FUNDAMENTOS

En el presente capítulo se describe el proceso de hilatura, así como las características de las máquinas de producción de hilo, los diferentes parámetros de programación y sensores que se utilizan dentro de la misma.

1.1 DESCRIPCIÓN DEL PROBLEMA

La empresa Textiles Río Blanco en sus diferentes procesos de producción, utiliza contadores electrónicos originales de la firma textil RIETER, los cuales dan la posibilidad de apreciar diferentes parámetros de producción en todo el proceso de hilatura tales como velocidad de entrada, velocidad de salida, torsión, metros producidos por turnos, sean estos de 1 a 4 turnos diarios.

La tecnología electrónica con la que fueron construidos está discontinuada y en poco tiempo no existirían repuestos, de existir el costo de mano de obra por reparación y repuestos serían excesivamente costosos, además la adquisición de datos de producción diaria es tomada por un operador, habiendo así la posibilidad de que existan fallas y errores en los reportes diarios.

El diseño y construcción de un sistema de conteo de producción para las 5 cardas C4/0 mediante microcontroladores, conjuntamente con una red industrial que pueda monitorear la producción diaria que cada una de estas máquinas entrega, es una importante necesidad para la fábrica. Además este sistema de conteo y monitoreo de procesos se podrá adaptar a cualquier tipo de maquinaria donde se requiera tener datos de producción, sea en unidades m/min, Km/h y de estas unidades tener Kg/día, Kg/h, mediante la aplicación de factores textiles de cada máquina.

1.2 FUNDAMENTOS FÍSICOS DEL PROCESO DE HILATURA

1.2.1 Introducción.

La empresa Textiles Río Blanco produce hilo de fibra de algodón, para obtener este producto final, el algodón pasa por diferentes procesos, empezando por la apertura, en donde se libera el algodón de toda clase de impurezas, después pasa por las cardas en donde se obtiene mechas de algodón, para obtener una mejor calidad de las fibras[1] se las somete a un proceso de peinado, regresa a las cardas, pasa a las mecheras en donde se realiza la torsión y templado de las mechas para obtener unas de menor diámetro, seguido a la fase de hilatura[2] en donde se aplica otro proceso de torsión y templado de las mechas de menor diámetro, pasa a la fase de enconado, para obtener un producto final listo para ser comercializado.

1.2.2 Tipos de fibras utilizadas para el proceso de hilatura.

Fibras textiles

Están formadas por elementos de longitud muy superior a su diámetro, que se emplean para la fabricación de hilados. La unión de muchas fibras textiles da origen a un hilo. A su vez, mediante hilos podemos formar los hilados o tejidos. En el mercado existe una gran variedad de fibras textiles que se pueden clasificar en tres grandes grupos: naturales (que a su vez pueden ser de origen mineral, vegetal y animal), artificiales y sintéticas.

[1] Se refiere a las que se pueden hilar o utilizar para fabricar telas mediante operaciones como tejido, trenzado.

Fibras naturales de origen mineral.

Se obtienen de minerales de estructura fibrosa. Solamente el amianto posee esta cualidad. Recientemente se ha comprobado que su manipulación provoca leucemia y cánceres, por lo que su uso está prohibido.

Fibras naturales de origen vegetal

Fibras vegetales.- Formadas por diferentes flores, frutos o tallos vegetales. En nuestra cultura occidental las fibras naturales de origen vegetal más importantes son las de algodón, lino y esparto.

Algodón.- Está formado por el 91 % de celulosa. Esta celulosa se emplea para fabricar fibras naturales y como materia prima para la fabricación de fibras artificiales (celulósicas). Tiene la ventaja de ser muy agradable al tacto y no producir alergias.

Lino.- Es una fibra más resistente que el algodón, pero menos elástica y flexible. Conduce el calor con facilidad, lo que origina que las prendas sean frescas en verano. Se usa para fabricar ropa de cama, vestidos, camisas, chaquetas, etc.

Esparto.- Su aplicación se centra en la fabricación de artículos de artesanía popular.

Fibras naturales de origen animal

Además de los pelos de diversas especies animales (oveja, camello, llama, conejo, etc.), también se emplean la seda y el cuero.

Lana.- Se obtiene del pelo que recubre el cuerpo de las ovejas. Tiene las siguientes propiedades: Es muy elástica, por lo que no se arruga con facilidad.

[2] proceso final en la transformación de las fibras en hilo

Tiene el inconveniente de que el calor húmedo afecta a la fibra; por eso no se debe planchar con vapor.

Seda.- Es un filamento continuo producido por el gusano de seda. Es una fibra muy apreciada para la fabricación de prendas de vestir, cortinas, tapices, etc. Es cara, muy elástica, pero tiene el inconveniente de ser mala conductora del calor.

Cuero.- Es el pellejo (piel) de un animal que se ha sometido a un proceso de curtido. El curtido consiste en eliminar el pelo y la epidermis, dejando solamente la dermis. Luego se le añaden sustancias curtientes para darle elasticidad y evitar que sea atacada por hongos o insectos.

Fibras artificiales

Estas fibras se inventaron a finales del siglo XIX, para diferenciarlas de las fibras sintéticas, cuya materia prima procede de síntesis química.

Las fibras artificiales más importantes son las celulósicas (también conocidas con el nombre de rayones) y las proteínicas.

Fibras celulósicas

Se las conoce, también, con el nombre de rayones. Sus características son:

La materia prima es la celulosa (materia vegetal de árboles o plantas), a la que se añaden distintos disolventes para dar origen a los distintos tipos de rayones.

Tienen una gran resistencia en seco, pero no tanto cuando están húmedas.

Son agradables al tacto. Normalmente suelen ir mezcladas con fibras naturales y/o fibras sintéticas, para darles mayor resistencia mecánica.

Fibras proteínicas

Se utilizan menos que los rayones, aunque cuando se inventaron se les auguraba un gran futuro. El proceso de fabricación consiste en disolver una proteína de origen animal o vegetal y luego formar filamentos.

Fibras sintéticas

Son las más utilizadas en la actualidad. Tienen la ventaja de ser muy resistentes a todo tipo de agentes externos, no necesitan casi planchado y la suciedad desaparece de ellas con facilidad. Su mayor inconveniente es que son poco higroscópicas (no absorben el sudor), por lo que en verano son muy calientes y en invierno muy frías. Ver Tabla 1.1.

Para solucionar estos problemas, normalmente se suelen mezclar con fibras naturales.

Tipo de fibra	Fibra más conocida	Características
Poliamidas	Nailon	Son muy resistentes y elásticas; no son atacadas por insectos o putrefacción. Se deforman con el calor (ya que son termo-plásticas).
Poliéster	Tergal	Se obtiene a partir de un diácido. Normalmente este tipo de fibra se mezcla con la lana.
Acrílicas	Leacril	Son muy resistentes a la acción de la intemperie y de la luz. Generalmente se emplean en géneros de punto.
Polivinílicas	Rhovil	Son fibras muy suaves, por lo que se suelen utilizar para la fabricación de prendas de recién nacidos.
Polietilénicas	Sarán	Tienen una gran resistencia a la fricción. Por ello se usan mucho en artículos de tapicería, alfombras y moquetas.

Polipropilénicas	Merklón	Resisten muy bien la fricción, así como toda clase de tratamientos y agentes químicos. Se emplean mucho en la fabricación de tapicerías, artículos de uso industrial y prendas de trabajo.
Poliuretano	Lycra	Tienen una enorme elasticidad. Se emplean en la fabricación de prendas de corsetería, bañadores, vestuario deportivo, etcétera.

Tabla 1.1. *Características más importantes de las principales fibras sintéticas*

1.2.3 Estudio de la Máquina RIETER serie EZA1.

El cardado es un proceso en el cual se endereza parcialmente las fibras y forma con ellas una trama delgada que se unen en una cuerda suave conocida como cinta de cardado. La máquina para cardar se compone de cilindros cubiertos con una guarnición gruesa y pesada, de elementos llamados chapones y otro elementos como peinador, otros cilindros de salida y de una apiladora donde se recoge la cinta en botes que es el resultado del proceso de esta máquina, en la figura 1.1 se observa una carda.

El material previamente procesado ingresa a la tolva de alimentación, la cual guía y regula la cantidad de material que va ingresar al cilindro alimentador que es controlado electrónicamente para obtener una alimentación uniforme dependiendo tanto de los dispositivos de la entrada que es la tolva en la cual se tiene un sensor de desplazamiento, en la salida los cilindros calandradores que es por donde se obtiene la cinta de carda se mide la variación de la cinta, en base a un set point, un motor controla el sistema de regulación mediante la variación de velocidad para tener una cinta de material uniforme a la salida.

Figura 1.1 *Carda Rieter.*

Los cilindros de alimentación son los que lleva el material al likerin el cual es el encargado de abrir y limpiar el material, mediante estiramientos se logra obtener una fina capa de material en forma de velo el cual es llevado por el gran tambor, el mismo es forrado por una guarnición en forma de dientes de sierra, los chapones son los encargados de limpiar, sacar fibra corta y da uniformidad al velo que sigue la trayectoria hacia el peinador, el mismo dispone de unas guarniciones que arrastra el material y mediante rodillos se lleva hacia los cilindros calandradores proceso en el cual el velo se transforma en cinta que es recogida en botes, función que realiza la apiladora.

En el mismo eje de estos cilindros calandradores se encuentra acoplado un disco con agujeros que da la señal al sensor de proximidad, este sensor es el que da la información de la cantidad de material que se está entregando.

En la figura 1.2 se muestra el lugar de donde se capta la señal que es enviada para el procesamiento del contador.

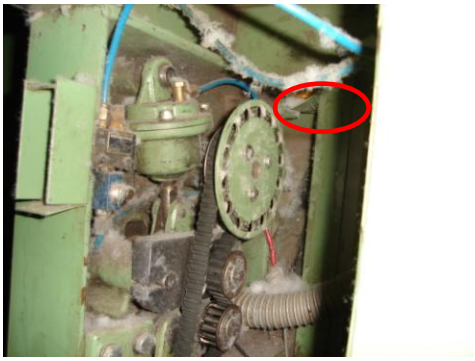


Figura 1.2 *Ubicación de sensor de proximidad en la Carda.*

El sensor utilizado es uno de proximidad inductivo tipo NPN (figura 1.3), este sensor tiene tres terminales, necesita de una fuente de alimentación de 24Vdc, un terminal que envía la señal y el terminal de tierra.

Figura 1.3 *Sensor de proximidad utilizado por la Carda.*

1.2.4 Pasos del Proceso de hilatura

Proceso de hilatura convencional

El proceso comienza desde que la materia prima llega al almacén. En este almacén se encuentra material natural, sintético y artificial así como desperdicio recuperado. Una vez que la materia prima esta dentro del sistema pasa a los siguientes procesos:

Apertura.

Proceso que consiste en abrir, mezclar, limpiar y formar el batido Aquí las pacas viajan a través de los ductos que jalan pequeños mechones de fibra de la parte inferior de la paca y los deja caer sobre una malla. La suciedad y basura se elimina con aire a alta velocidad.

De aquí las fibras limpias y sueltas se introducen a las máquinas de cardado (figura 1.4). La empresa cuenta con 6 máquinas conectadas entre sí.

Figura 1.4 *Máquina de apertura.*

Cardado.

El cardado endereza parcialmente las fibras y forma con ellas una trama delgada que se une en una cuerda suave conocida como mecha o cinta cardada. Este proceso es automatizado, y las máquinas cuentan con unos rodillos cubiertos de una capa gruesa y pesada llamados chapones, del cual se obtiene una cinta que puede alimentar al proceso de open end, peinado o estirado dependiendo de las características del hilo como se muestra en la (figura 1.1).

Estirado.

El estirado aumenta el paralelismo de las fibras y combina varias mechas de carda en una cinta de manuar. Esta es una operación de mezclado que contribuye a dar mayor uniformidad al hilo. El estirado se lleva a cabo por medio de conjuntos de rodillos, cada uno de los cuales gira a mayor velocidad que el conjunto anterior, ver (figura 1.5).

Figura 1.5 *Máquina de estirado.*

Veloz.

Llamado también trenzado, es un proceso que reduce el diámetro de la cinta o mecha de manuar, aumenta el paralelismo de las fibras y proporciona torsión. El producto se denomina pabilo, el cual es un cabo suave de fibras torcidas. Las máquinas cuentan con 3 rodillos por los que pasa la cinta, el primero es más lento que el último y de la velocidad de estos depende el calibre del hilo, después el material pasa por los husos que giran y enrollan al pabilo en los carretes dándole cierta torsión, ver (figura 1.6).

Figura 16 *Máquina veloz.*

Continua Trozil.

Este proceso proporciona la torsión que hace del hilo simple un hilado de fibra discontinua, la hilatura en anillo estira, tuerce y enrolla en una sola operación continua. La función primordial es producir hilo transformando los pabilos a través de estirar, torcer y enrollar el material en una sola operación. Se producen hilos de diferentes calibres o gruesos llamados títulos.

Las máquinas se alimentan manualmente con los carretes de pabilo y a través del estiramiento y torsión le dan al hilo el calibre requerido, por último el hilo se va enrollando en las canillas, ver (figura 1.7).

Figura 1.7 *Máquina continua.*

Devanado.

El hilado que llega de marco de hilatura es un hilo simple que debe embobinarse en carretes. A medida que se embobina, puede dársele más torsión o combinarse con otros hilos sencillos para hacer un hilo de varios cabos. Su función es reducir el hilo en crudo que viene del bobinado a carretes o conos dándole la presentación deforma de madejas. Este es un proceso alternativo y el hilo obtenido va directo al almacén de venta.

Figura 1.8 *Máquina de Devanado*

Proceso de hilatura peinado

El objetivo fundamental del peinado es colocar las fibras en posición paralela y eliminar cualquier fibra corta del resto, de manera que las fibras peinadas tendrán una longitud más uniforme. Para llegar a este proceso se pasó por apertura y cardado. Ver (fig1.9).

Figura 1.9 *Máquina de peinado.*

<u>Almacén de hilo</u>.

Aquí se almacena el hilo producido, el hilo que sobra después de tejer las telas y el hilo teñido para volverlo a usar o para vender.

Figura 1.10 *Almacén de hilo.*

Determinación de Parámetros de las Máquinas.

Todos los parámetros que se pueden considerar están basados a la señal que entrega el sensor de proximidad, esta señal consiste en pulsos de onda cuadrada de frecuencia dependiente de la velocidad de cada máquina.

En base a los pulsos que el sensor envía, la primera acción que el contador realiza es los pulsos transformarla en frecuencia, esto se lo realiza con un contador que sea capaz de contabilizar los pulsos en un periodo de 1Seg. este número de pulsos contados en un segundo equivale a un valor de frecuencia, el cálculo de revoluciones (RPM) se lo realiza mediante la fórmula 1.1.

$$RPM = \frac{Frecuencia \times 60}{ND} \qquad (1.1)$$

Donde:

Frecuencia= Pulsos/Segundo
1 minuto tiene 60 Seg.
ND= Número de dientes ó agujeros (14).

Para la determinación de los parámetros debemos tomar en cuenta solo el número de agujeros o el número de dientes que tiene un piñón:

En caso de la carda se tienen 14 pulsos por revolución.

1.3 ESTUDIO DE LAS REDES DE COMUNICACIÓN

1.3.1 Introducción al protocolo RS232.

Protocolo de comunicación.- Conjunto formal de reglas convencionales que rigen el formato y la sincronización relativa al intercambio de mensajes entre dos o más dispositivos en una red de comunicaciones.

Interfase de comunicación. Es el hardware que permite a un equipo de control o de cómputo la conexión a un sistema de comunicación.

La interfase serial RS-232

RS-232C.- Norma internacional de transmisión serial entre dos dispositivos, que utiliza señales de voltaje para comunicarse. Debido a que se emplea una señal de voltaje, el RS-232C no puede transmitir señales a largas distancias (máximo 15 metros).

1.3.2 Tipos de comunicaciones digitales

Sincrónica.

En una transmisión síncrona los datos son enviados un bit a continuación de otro por una línea que une la salida del transmisor, TXD, del un lado con la línea de recepción, RXD, del otro lado. El transmisor y el receptor son sincronizados con una línea extra que trasmite pulsos de reloj que básicamente le indican al receptor cuando leer un pulso. La duración del bit es determinada por la duración de los pulsos de sincronismo. Como se puede entender, el uso de esta técnica implica la existencia de un cable extra para llevar la señal de reloj, lo cual resulta en un costo extra.

Asíncrona.

En la transmisión asíncrona no se emplea una señal de reloj, más bien se utiliza una técnica que recurre a “encapsular” los datos con un bit de inicio y uno o dos bits de parada, y así no es necesaria la línea extra de sincronismo.

Pero encapsular con un bit de inicio y otro de parada los datos no es suficiente, hay otras reglas de transmisión sobre las que se deben poner de acuerdo el transmisor y receptor. Por ejemplo, mientras el estado de la línea está en alto, el receptor deberá interpretar como que no existe transmisión y, por lo mismo, el canal está en modo de espera (idle). Cuando la línea de comunicación cambia de estado (se recibe el bit de inicio), el receptor debe interpretar ese cambio como el comienzo de la transmisión. Pero, ¿que sucede si ese cambio de estado fue producido por un transitorio? En situaciones así, el algoritmo de comunicación debe instruir al receptor esperar cierto tiempo para volver a leer la línea y así asegurarse que el cambio de estado corresponde efectivamente al bit de inicio.

Pero, volver a leer la línea ¿después de qué tiempo? Igualmente, ¿cada qué tiempo el receptor debe leer la línea para leer un bit de la trama? En ambas situaciones se comprenderá que se necesita saber el tiempo de duración de un bit. Por todo lo indicado, antes de iniciar cualquier comunicación con el puerto

RS-232 se debe determinar el protocolo a seguir. Esto debe ser hecho por el usuario quien debe decidir sobre:

El Protocolo serial: esto es, el número bits de datos, la paridad, el número de bits de parada)

- La velocidad de transmisión.
- El protocolo de control de flujo (RTS/CTS o XON/XOFF).

Con RS232C se puede transmitir los datos en grupos de 5, 6, 7, u 8 bits aunque los más usados son 7 y 8 bits.

La velocidad de transmisión (normalmente 9600 bits por segundo para aplicaciones industriales) debe ser constante durante la trasmisión de una trama para garantizar que los bits lleguen uno tras de otro en el momento correcto. Cualquier retardo provocaría una lectura incorrecta.

Los bits de datos son enviados al receptor después del bit de inicio, el bit menos significativo es transmitido primero. Dependiendo de la configuración de la transmisión, un bit de paridad es enviado después de los bits de datos. El propósito de cada unos de estos bits especiales se indica a continuación:

- **Bit de inicio.-** cuando el receptor detecta el bit de inicio sabe que la transmisión ha comenzado y es a partir de entonces que debe leer las señales de la línea a intervalos concretos de tiempo, en función de la velocidad de transmisión.

- **Bit de paridad.-** con este bit se pueden descubrir errores en la transmisión. Se puede dar paridad par o impar. En la paridad par, por ejemplo, la palabra de datos a transmitir se completa con el bit de paridad de manera que el número de bits 1 enviados sea par.

- **Bit de parada.-** indica la finalización de la transmisión de una palabra de datos. El protocolo de transmisión de datos permite 1, 1.5 y 2 bits de parada.

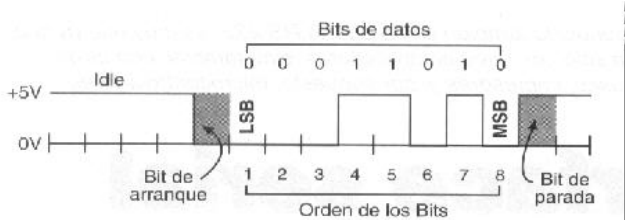


Figura 1.11. *Trama TTL*.

En la (Figura1.11) se muestra la trama con los voltajes correspondientes a la lógica TTL, la cual no se emplea en la norma RS-232.

El estándar RS-232 establece que un 1 lógico se represente con un voltaje entre -3V y -15V, mientras que un 0 lógico se represente con un voltaje entre +3V y +15V.

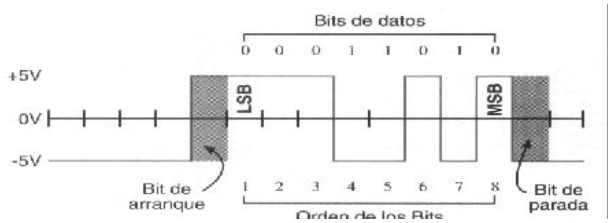


Figura 1.12. *Trama RS-232 típica*.

Los voltajes más usados son +12V y -12V. El estado de reposo (idle) se representa con un 1 lógico; es decir, -12V. Dependiendo de la velocidad de transmisión empleada, es posible tener cables de hasta 15 metros. En la (Figura 1.12) se muestran los voltajes RS-232 para voltajes de +/- 5V.

1.3.3. El puerto serial de una PC.

El puerto serial de una PC se rige por el estándar RS-232C. Este estándar fue empleado en los 60s para comunicar un Equipo Terminal de Datos, DTE (Data Terminal Equipment, el PC en este caso) y un equipo de comunicación de datos, DCE (Data Communication Equipment, habitualmente un módem).

Para conseguir los niveles de voltaje correctos hay circuitos que dan soporte para convertir los niveles de voltaje TTL a RS-232 y viceversa. En el ejemplo de la

(Figura 1.13) se muestran los circuitos que se emplean en una computadora para manejar la interfase serial RS-232 de la misma.

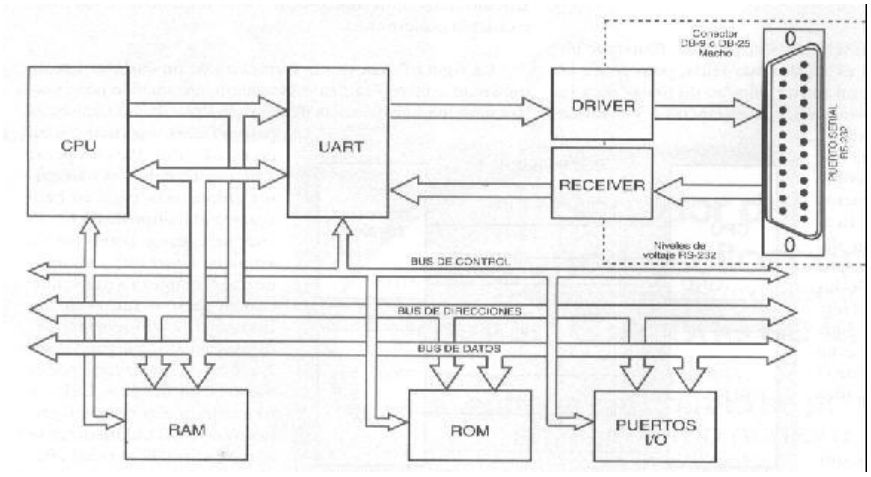


Figura 1.13. *Trabajo del UART.*

El driver convierte de TTL a RS-232 para la transmisión y el receptor (receiver) de RS-232 a TTL.

El estándar especifica un conector DB-25 de 25 pines y que el conector DTE debe ser macho y el conector DCE hembra. Muchos de los 25 pines no son necesarios y solo se emplean las siguientes:

- **Línea de transmisión de datos (TxD).-** línea por la que el DTE (PC) envía los datos.

- **Línea de recepción de datos (RxD).-** línea por la que el DTE (PC) recibe los datos.

- **DTE preparado (DTR).-** línea por la que el DTE (PC) indica al DCE (módem) que está activo para comunicarse con el módem.

- **DCE preparado (DSR).-** línea por la que el DCE (módem) indica al DTE (PC) que está activo para establecer la comunicación.

- **Petición de envío (RTS).-** con esta línea, el DTE (PC) indica al DCE (módem) que está preparado para transmitir datos.

- **Preparado para enviar (CTS).-** tras un RTS, el DCE (módem) pone esta línea en 1 lógico, tan pronto como está preparado para recibir datos.

- **Tierra.-** necesaria para que tenga lugar la transmisión.

Por esta razón en muchas PC modernas se utiliza el conector DB-9 macho.

En la (Tabla 1.2) se listan los pines con su respectiva numeración:

Numero	de Pin	Señal	Descripción	E/S
En DB-25	En DB-9			
1	1	-	Masa chasis	-
2	3	TxD	Transmit Data	S
3	2	RxD	Receive Data	E
4	7	RTS	Request To Send	S
5	8	CTS	Clear To Send	E
6	6	DSR	Data Set Ready	E
7	5	SG	Signal Ground	-
8	1	CD/DCD	(Data) Carrier Detect	E
15	-	TxC(*)	Transmit Clock	S
17	-	RxC(*)	Receive Clock	E
20	4	DTR	Data Terminal Ready	S
22	9	RI	Ring Indicator	E
24	-	RTxC(*)	Transmit/Receive Clock	S

Tabla 1.2. *Descripción de pines en RS-232.*

Conector DB 9 Conector DB 25

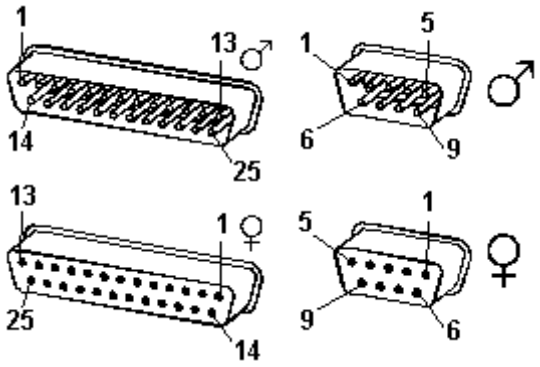


Figura 1.14*. Formato y distribución de pines en RS-232.*

Conexión de las líneas.

Para hacer posible la comunicación entre dos equipos PC se interconectan los pines de la siguiente manera:

(PC1)	**RxD**	**<==========**	**TxD**	**(PC2)**
(PC1)	**TxD**	**==========>**	**RxD**	**(PC2)**
(PC1)	**DTR**	**==========>**	**DSR**	**(PC2)**
(PC1)	**DSR**	**<==========**	**DTR**	**(PC2)**
(PC1)	**RTS**	**==========>**	**CTS**	**(PC2)**
(PC1)	**CTS**	**<==========**	**RTS**	**(PC2)**
(PC1)	**TIERRA**	**==========**	**TIERRA**	**(PC2)**

Los pines que portan los datos son RxD y TxD, los demás se encargan de otros trabajos tal como se indica más arriba.

Para controlar el puerto serie, la computadora usa direcciones de E/S e interrupciones (IRQ): para COM1 la dirección 3F8h y el IRQ4 y para COM2 la 2F8h e IRQ3. Luego el estándar añadió nuevos puertos serie siendo las direcciones de E/S, 3E8 para COM3 y 2E8 para COM4, sin especificar las IRQ, debiendo el usuario definirlas en función de las que tenga libres o el uso que vaya a hacer de los puertos serie. Se puede tener en diferentes puertos las mismas IRQ siempre que no se utilicen los dos al mismo tiempo.

1.3.4. La interfase RS-485

RS-485.- Norma internacional de comunicación serial que permite entablar la comunicación entre 32 dispositivos sobre un mismo canal. Esta norma tiene la característica de transmisión diferencial y permite la comunicación entre dos dispositivos separados a una distancia máxima de 1200 metros

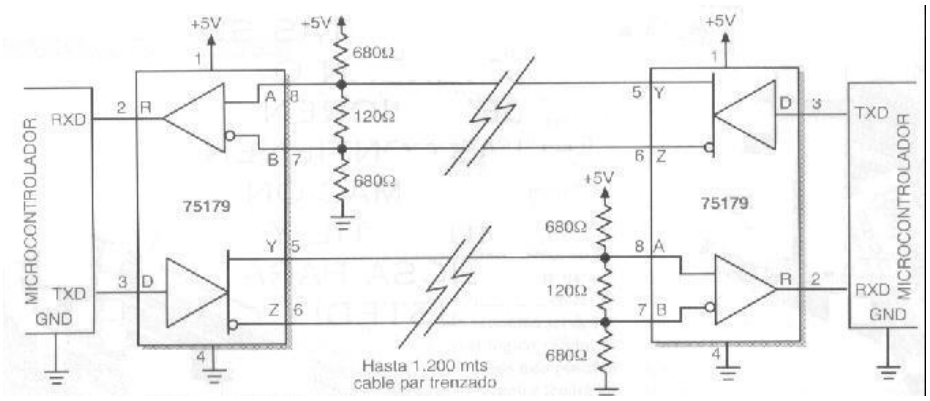


Figura 1.15. *RS-485 y su modo diferencial.*

La característica más relevante de la comunicación RS-485 es que puede trabajar en modo diferencial tal como se muestra en la (Figura 1.15).

Es bastante simple convertir una RS-232 a RS-485 o viceversa. En la (Figura 1.16) se ve uno de estos circuitos de conversión basado en el MAX-232.

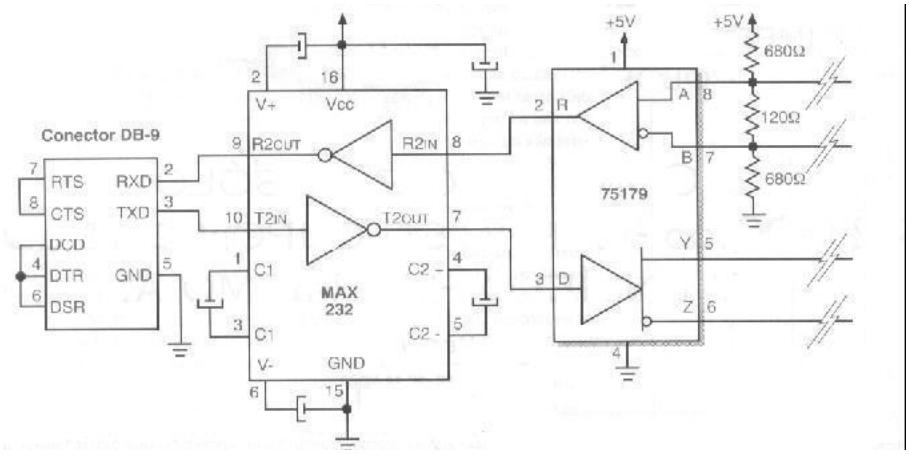


Figura 1.16. *Conversión RS-232 a RS-485.*

En la Figura (1.17) se muestra un circuito típico que permite enlazar una PC con un microcontrolador empleando RS-485.

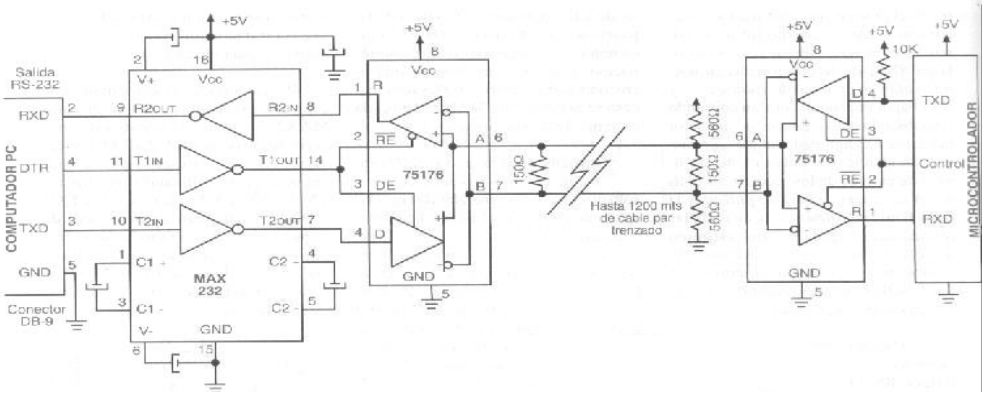


Figura 1.17. *Enlace entre una PC y un microcontrolador.*

Protección ESD

En el enlace diferencial de la interfase RS-485 proporciona una transmisión confiable ante la presencia de ruido, y las entradas diferenciales de los receptores pueden rechazar voltajes en modo-común elevados.

Consecuencias de tazas altas de transmisión de datos

Con velocidades de transmisión altas se comienza a tener problemas con las reflexiones. Para contrarrestar éstas, se instalan resistencias de terminación con un valor igual al de la impedancia del cable. Para cables RS-485 comunes (pares trenzados de 24AWG), esto significa resistencias de 120 en los dos terminales.

Para la transmisión no se emplea los niveles asociados con la lógica TTL (0 ó 5 V), pues se tiene una componente continua que es difícil de manejar. Por otro lado se tiene el problema de la sincronización entre transmisor y receptor durante la

transmisión. Para resolver estos problemas se emplean técnicas especiales de codificación: NRZ y bifásicas.

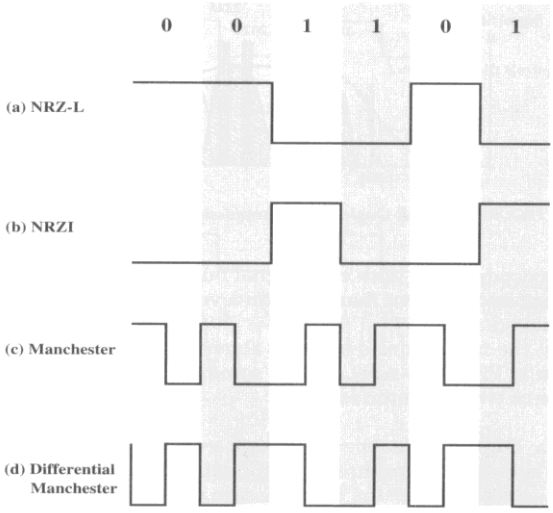


Figura 1.18.*Codificación de señales digitales*

La codificación NRZ (Nonreturn-to-zero) implica que la señal digital nunca se hace cero sino que se emplea dos voltajes, uno positivo y otro negativo, para representar los dos dígitos binarios. En la técnica NRZ-L (NRZ level) un voltaje negativo representa el dígito binario 1 y un voltaje positivo el digito binario 0.

Puesto que la señal tendrá entonces valores positivos y negativos, se consigue de esta manera reducir el efecto de la componente de continua, aunque no totalmente. Se emplea para conexiones cortas como entre un terminal y un módem o entre un terminal y una computadora cercana.

NRZ-I es una variante de la anterior y representa un ejemplo de una codificación diferencial en la que la señal es codificada comparando con la polaridad de los elementos adyacentes. Esta técnica es más confiable para detectar una transición en la presencia de ruido.

La desventaja de la transmisión NRZ es que es difícil determinar cuando termina un bit y cuando empieza otro. Si se transmite una sucesión de ceros o unos, cualquier corrimiento en el tiempo entre transmisor y receptor resultaría en una

pérdida de sincronización. También está el problema de la componente de continua.

Para resolver este problema se usan las técnicas bifásicas como la Manchester y Manchester Diferencial. En ambos casos por cada bit hay una transición obligatoria que no permite la aparición de la componente de continua y facilita la sincronización del receptor. La ausencia de una transición que se esperaba puede servir para detección de errores. El ruido debería producir una inversión antes y después de una transición esperada para que un error pase sin detectarse.

1.3.5 Acondicionamiento de señales.

Los acondicionadores de señal, adaptadores o amplificadores, en sentido amplio, son los elementos del sistema de medida que ofrecen, a partir de la señal de salida de un sensor electrónico, una señal apta para ser presentada o registrada o que simplemente permita un procesamiento posterior mediante un equipo o instrumento estándar. Normalmente, son circuitos electrónicos que ofrecen, entre otras funciones, las siguientes:

- Amplificación.
- Filtrado.
- Acoplamiento de impedancias.
- Modulación.
- Demodulación.

Si se considera, por ejemplo, el caso en que una de las etapas de tratamiento de la señal de medida es digital, y si la salida del sensor es analógica, que es lo más frecuente, hará falta un convertidor A/D. Éstos tienen una impedancia de entrada limitada, exigen que la señal aplicada sea continua o de frecuencia de variación lenta, y que su amplitud esté entre unos límites determinados, que no suelen exceder de 10 voltios. Todas estas exigencias obligan a interponer un acondicionador de señal entre el sensor, (que muchas veces ofrece señales de apenas unos milivoltios), y el convertidor A/D.

No se puede hablar de los sensores, como componentes electrónicos básicos, sin ver como se pueden adaptar a un sistema de adquisición y control. Por lo que se tendrán que ver las nuevas tecnologías de adaptación de estos sensores que como parte de una cadena de dispositivos, forman un sistema.

Estos adaptadores, como acondicionadores de señal, son los amplificadores operacionales en sus diferentes estructuras de montaje, pasando por filtros o por procesadores analógicos, convirtiendo estas señales de analógico a digital para posteriormente ser procesados con un DSP o Microcontrolador y actuando por medio de las salidas lógicas del procesador o por medio de un convertidor digital a analógico.

El procesamiento conocido como acondicionamiento de señal, incluye funciones como amplificación, filtrado, aislamiento eléctrico y multiplexado.

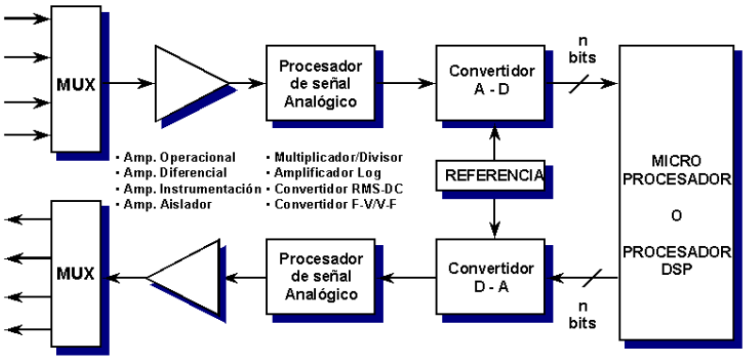


Figura 1.19 *Acondicionador de señales.*

1.3.6 Sensores Inductivos[3]

Los sensores de proximidad inductivos son detectores de posición electrónicos, que dan una señal de salida sin contacto mecánico directo, estos sensores detectan todo tipo de objetos metálicos

[3] http://www.sick.es/es/productos/sensores/inductivos/es.html

Figura 1.20 *sensores inductivos.*

Principio de funcionamiento.

Consiste en una bobina cuya frecuencia de oscilación cambia al ser aproximado un objeto metálico a su superficie axial. Esta frecuencia es empleada en un circuito electrónico para conectar o desconectar un tiristor y con ello, lo que esté conectado al mismo, de forma digital (ON-OFF) o, analógicamente. Si el objeto metálico se aparta de la bobina, la oscilación vuelve a empezar y el mecanismo recupera su estado original.

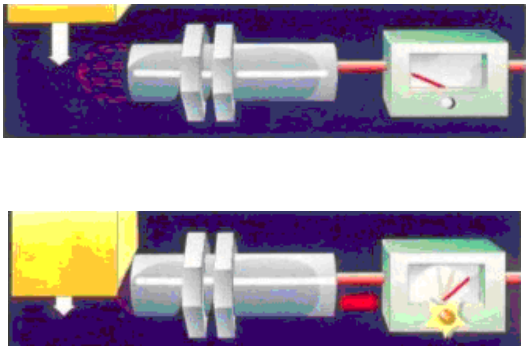


Figura 1.21 *Principio de funcionamiento del sensor inductivo*

Construcción:

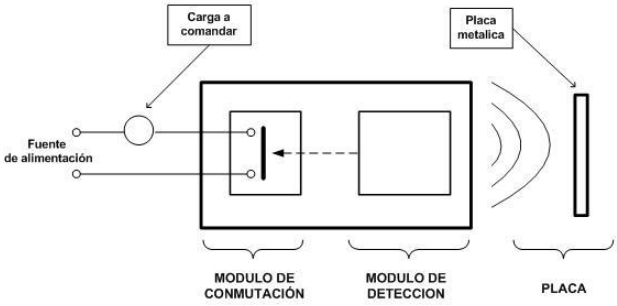


Figura 1.22 *Construcción del sensor inductivo*

VENTAJAS	DESVENTAJAS
• No entran en contacto directo con el objeto a detectar. • No se desgastan. • Tienen un tiempo de reacción muy reducido • Tiempo de vida largo e independiente del número de detecciones. • Son insensibles al polvo y a la humedad. • Incluyen indicadores LED de estado y tienen una estructura modular.	• Sólo detectan la presencia de objetos metálicos. • Pueden verse afectados por campos electromagnéticos intensos. • El margen de operación es más corto en comparación con otros sensores.

Tabla 1.3. *Ventajas y desventajas de los sensores inductivos*

Aplicaciones:

Estos sensores se desempeñan en las condiciones de trabajo más difíciles donde hay presente aceites, líquidos, polvos y vibraciones, entre algunas que se mencionan están: herramientas, máquinas textiles, líneas transportadoras, sistema de transporte, equipos de empaques, industria automotriz, etc.

1.4 ESTUDIO DE LOS MICROCONTROLADORES

1.4.1 Características generales de los Microcontroladores.

Todos los microcontroladores deben disponer de los bloques esenciales: procesador (CPU), memoria de datos y de instrucciones, líneas de E/S, oscilador de reloj y módulos controladores de periféricos.

Sin embargo, cada fabricante intenta enfatizar los recursos más idóneos para las aplicaciones a las que se destinan preferentemente.

Von Neumann.

Este tipo de arquitectura consta de los siguientes bloques:

• Memoria principal
• Unidad aritmético-lógica
• Unidad de control
• Unidad de entrada/salida

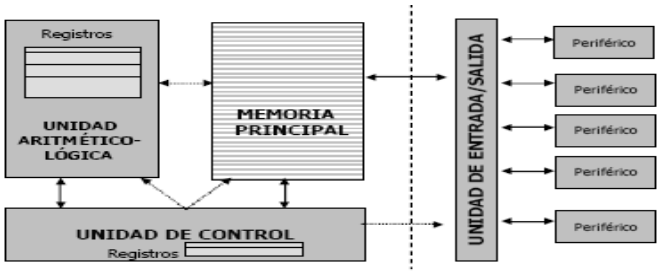


Figura 1.23 Arquitectura Von Neumann

Memoria principal: se emplea para almacenar datos o instrucciones. Se compone de una serie de celdas a cada una de las cuales se accede a través de una dirección. Cada celda está compuesta de un número de bits que nos dá el ancho de palabra de la memoria.

Unidad aritmético-lógica: permite realizar las operaciones elementales (AND, OR, NAND, NOR, suma, resta, etc.). Estas operaciones las realiza con datos procedentes normalmente de la memoria y los resultados parciales los almacena o no temporalmente en algunos registros que suele tener la propia unidad aritmética. Las operaciones básicas que puede realizar un computador se dividen en operaciones lógicas y operaciones aritméticas. Ambos tipos de operaciones se realizan dentro de esta unidad.

Unidad de control: se encarga de generar las señales necesarias para que todo el computador ejecute las instrucciones leídas de la unidad de memoria. La unidad de control es la encargada de leer de forma consecutiva las instrucciones (que se encuentran almacenadas en la unidad de memoria), y generar las señales

eléctricas necesarias para proceder a la ejecución de cada una de estas instrucciones. Es la unidad que gobierna y gestiona el comportamiento de un computador.

Unidad de entrada/salida: realiza la transferencia de información con el exterior, con las unidades que se conocen con el nombre de periféricos.

Buses: caminos que unen los distintos bloques del computador y a través de los cuales circulan los datos y las instrucciones. Dependiendo del tipo de información que se trate podemos hablar de diferentes caminos:

• **Bus de datos:** circulan los datos de las operaciones que se van a efectuar.

• **Bus de direcciones:** circulan las direcciones de memoria donde se encuentra almacenado algún tipo de información a la que se desea acceder.

• **Bus de control**: circulan las señales de control generadas por la unidad de control para la ejecución de las instrucciones.

La agrupación de algunos bloques funcionales del computador como son la unidad de control, la unidad aritmético-lógica y los registros reciben el nombre de unidad central de proceso (CPU).

Arquitecturas: Von Neumann vs 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.

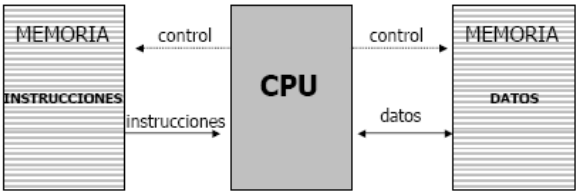


Figura 1.24 Arquitectura Harvard

1.4.2 Características

Memoria de programa: no volátil, tipo ROM (Read Only Memory)
Memoria de datos: volátil, tipo RAM (Random Access Memory)
Tipos de memoria no volátil (ROM):

• ROM con máscara
• OTP (One Time Programmable)
• EPROM (Erasable Programmable Read Only Memory)
• EEPROM (Electrical Erasable Programmable Read Only Memory)
• FLASH

Puertas de E/S: Terminales de la cápsula del microcontrolador, que lo comunican con los periféricos exteriores Puertas de E/S y reloj.

Reloj principal: 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.

Recursos especiales

- 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.
- Puertos de comunicación: UART, USART, USB, CAN.

1.4.3 Lenguajes de programación

Lenguaje máquina: representación del programa tal como la entiende el microcontrolador
0101000011111110000

Lenguaje ensamblador: representación alfanumérica del lenguaje máquina, lo que facilita su lectura.

Lenguajes de alto nivel.
Lenguaje C

Herramientas de desarrollo

Ensamblador: El programa ensamblador es un software que se encarga de traducir los nemónicos y símbolos alfanuméricos del programa escrito en ensamblador por el usuario a código máquina.

MPASM

Compilador: El programa compilador es un software que se encarga de traducir el programa escrito en C el usuario a código máquina.

MikroC

Simulador: Son capaces de ejecutar en un PC programas realizados para el microcontrolador. Los simuladores permiten tener un control absoluto sobre la ejecución de un programa, siendo ideales para la depuración de los mismos

Entornos de desarrollo:

MPLAB (incluye MPASM y MPSIM)[4]

Proteus

Placas de evaluación: Se trata de pequeños sistemas con un microcontrolador ya montado y que suelen conectarse a un PC desde el que se cargan los programas que se ejecutan en el microcontrolador. Las placas suelen incluir visualizadores LCD, teclados, LEDs, fácil acceso a los pines de E/S, etc.

[4] http://ww1.microchip.com/downloads/en/DeviceDoc/51519B.pdf

CAPÍTULO II

ANÁLISIS Y DISEÑO DE LOS CONTADORES Y MONITOREO A TRAVÉS DE LA RED INDUSTRIAL.

2.1 DESCRIPCIÓN DEL SOFTWARE DE PROGRAMACIÓN DE LOS MICROCONTROLADORES.

2.1.1 Introducción a Protón IDE.

El Protón IDE

El protón IDE es un programador y compilador profesional que cuenta con un Ambiente de Desarrollo Integrado visual poderoso (IDE).

Cuenta con un desarrollo de usuario cómodo y ambiente amistoso.

Posiblemente es el explorador del código más avanzado para PIC basado en el desarrollo del mercado. Rápidamente puede operarar con: el código del programa, los dispositivos, los Registros de la Función Especiales (SFRs).

Integración del Programador

El Protón IDE permite empezar el software de programación preferido por el usuario desde el ambiente de desarrollo. Esto permite compilar y entonces programar el microcontrolador.

Rápidamente se puede transmitir un programa en el microcontrolador sin la necesidad de un programador del hardware.

Comunicación serie

Permite transmitir y recibir los datos vía un cable de serie conectados a la PC y configurando una tabla de desarrollo. Es fácil usar la ventana de la configuración, que permite escoger el número del puerto selecto, la paridad, el tamaño del byte y número de parada. Alternativamente, la actualización en línea.

Las actualizaciones en línea permiten que se guarde correctamente actualizado con los últimos rasgos de IDE.

Los Sistemas operativos apoyados

Windows 98, 98SE, yo, NT 4.0 con SP 6, 2000, XP (recomendado)
Los Requisitos del hardware
233 Procesador de MHz (500 MHz o superior)(recomendado)
64 MB RAM (128 MB o superior) (recomendado)
40 MB unidad de disco duro espacio

La Barra del menú

Menú Archivo

- **New.-** Crea un nuevo documento. Un título se genera automáticamente, mientras se muestra la información, como el autor, derechos de propiedad literaria y fecha.

- **Open.-**Despliega una caja del diálogo abierta, permitiendo que se cargue un documento en el Protón, IDE.

- **Save.-**Ahorra un documento al disco. Este botón es normalmente inválido a menos que el documento se ha cambiado.

- **Save As.-** Permite nombrar y guardar un documento en el disco.

- **Close.-** Cierra el documento actualmente activo.

- **Close All.-** Cierra todo el editor documentado y entonces crea un nuevo documento del editor.

- **Print Setup.-** Indica un diálogo de arreglo de impresión.

- **Print Preview.-** Indica una ventana de vista preliminar de impresión.

- **Print.-** Imprime la página del editor actualmente activa.

- **Exit.-** Le Permite que termine el Protón IDE.

Edit Menú

- **Undo.-** Cancela cualquier cambio hecho en la página del documento actualmente activo.

- **Redo.-** Da Marcha atrás un despliegue en orden.

- **Cut.-** Corta cualquier selección de texto de la página del documento activo y lugares en el portapapeles.

- **Copy.-** Copia cualquier selección de texto de la página del documento activo y lugares en el portapapeles.

- **Paste.-** Pega los volúmenes del portapapeles en la página del documento activo.

- **Delete.-** Anula cualquier selección del texto.

- **Select All.-** Selecciona todo el texto en la página del documento activa.

• **Change Case.-** Permite cambiar el caso de un bloque seleccionando de texto.

Main Tollbar

New

Crea un nuevo documento. Un título se genera automáticamente, mientras muestra la información como el autor, derechos de propiedad literaria y fecha.

Open

Los indicadores despliegan una caja del diálogo abierta, permitiendo cargar un documento en el Protón IDE. Si el documento ya está abierto, entonces el documento es hecho la página del editor activa.

Save

Ahorra un documento al disco. Este botón es normalmente inválido a menos que el documento se ha cambiado.

Cut

Corta cualquier selección de texto de la página del documento activo y lugares en el portapapeles. Esta opción es inválida si ningún texto se ha seleccionado.

Copy

Copia cualquier selección el texto de la página del documento activa y lugares en el portapapeles. Esta opción es inválida si ningún texto se ha seleccionado.

Paste

Pega los volúmenes del portapapeles en la página del documento activa. Esta opción es inválida si el portapapeles no contiene ningún texto conveniente.

Undo

Cancela cualquier cambio hecho a la página del documento actualmente activo.

Redo

Invierte o deshace alguna acción.

Print

Imprime la página del editor actualmente activa.

Find

Despliega un diálogo de hallazgo.

Find and Replace

Despliega un hallazgo y reemplaza el diálogo.

Indent

Realiza el cambio de las líneas seleccionadas a la próxima parada de la etiqueta. Si no se seleccionan las líneas múltiples, una sola línea se mueve de la posición del cursor actual.

Outdent

Realiza el cambio de las líneas seleccionadas a la parada de la etiqueta anterior.

Si no se seleccionan las líneas múltiples, una sola línea se mueve de la posición del cursor actual.

Block Comment

Agrega el carácter del comentario a cada línea de un bloque seleccionado de texto. Si no se seleccionan las líneas múltiples, un solo comentario se agrega a la salida de la línea que contiene el cursor.

Block Uncomment

Quita el carácter del comentario de cada línea de un bloque seleccionado de texto. Si no se seleccionan las líneas múltiples, un solo comentario está alejado de la salida de la línea que contiene el cursor.

Compile

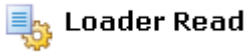
Apretando este botón, o F9, compilará la página del editor actualmente activa. El botón compile generará un archivo *.Hex .Apretando el botón compile automáticamente se guardan los archivos generados *.Hex en la dirección especificada.

Compile and Program

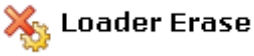
Apretando este botón, o F10, compilará la página del editor actualmente activa. Apretando el botón compile automáticamente se guardan los archivos generados *.Hex en la dirección especificada, además llama directamente al programador.

Loader Verify

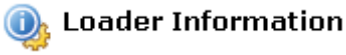
Este botón verificará un archivo hexadecimal. El cargador verifica que el botón sólo se habilita si el Cargador de MicroCode es el programador actualmente seleccionado.



Este botón manda el código y volúmenes de los datos de un microcontrolador al a Cargador de MicroCode.



Este botón borrará la memoria del programa para la serie de los 18Fxxx(x). El botón del cargador borra sólo si se habilita el Cargador de MicroCode.



Este botón desplegará la versión del microcontrolador. El botón de información de cargador sólo se habilita si el Cargador de MicroCode es el programador actualmente seleccionado.

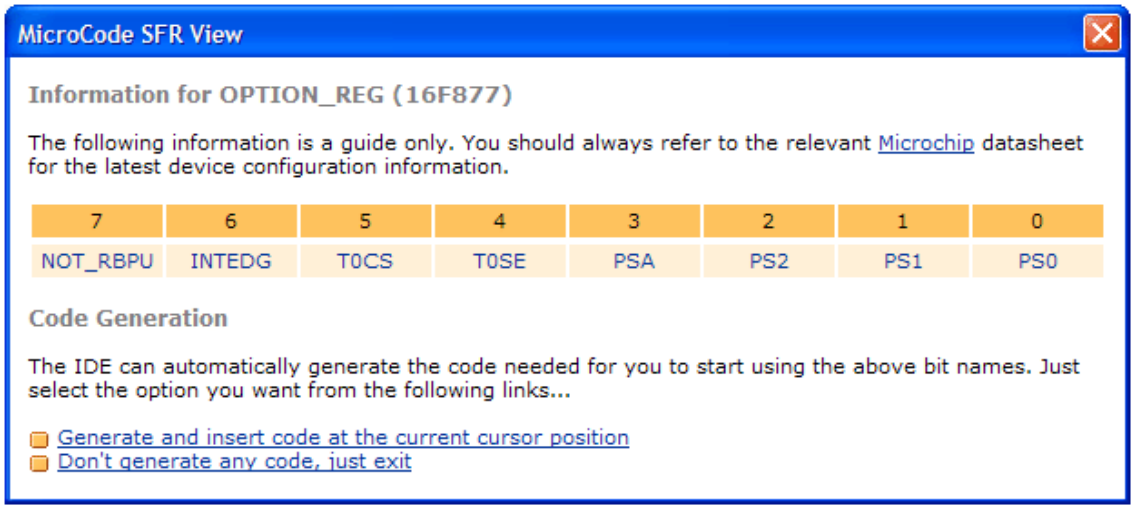
Code Explorer

El explorador del código permite navegar fácilmente el código del programa. Los códigos del explorador en árbol despliega al procesador actualmente seleccionado, incluyendo los archivos, declaraciones, constantes, variables, seudónimo y modificadores, etiquetas, macros y etiquetas de los datos.

Device Node

El nodo del dispositivo es el primer nodo en el árbol del explorador. Despliega el tipo del procesador actualmente seleccionado. Por ejemplo, si el programa tiene la declaración **DEVICE** 16F877.

Entonces el nombre del nodo del dispositivo será 16F877. No se necesita dar el nombre del dispositivo explícitamente en el programa para ser desplegado en el explorador.



Los SFR (Registros de Desplazamiento Especial) ven los despliegues de todos los nombres que pertenecen a un registro particular. Pulsando el botón un bitname desplegarán una ventana de información adicional sobre un bitname.

Include File Node

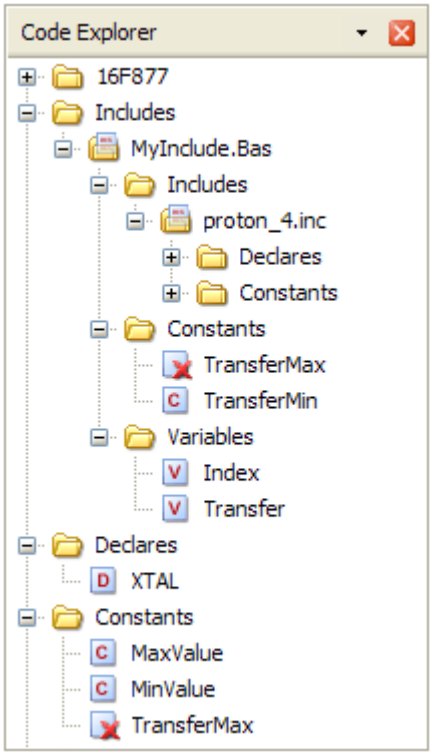
Cuando se dá un clic se incluye el archivo, el IDE abrirá ese archivo automáticamente por ver y revisar.

Include File Node

Declara, las constantes, variables, seudónimo y modificadores, etiquetas, macros y datos etiquetan los nodos del explorador trabajan en mucho la misma manera. Haciendo clic en cualquiera de estos nodos tomará la declaración necesaria.

Explorer Warnings and Errors

Cuando se dá un clic en incluya el archivo, el IDE abrirá ese archivo automáticamente por ver y revisar. Alternativamente. Para hacer esto, simplemente se da un clic en el icono y extienda el nodo.

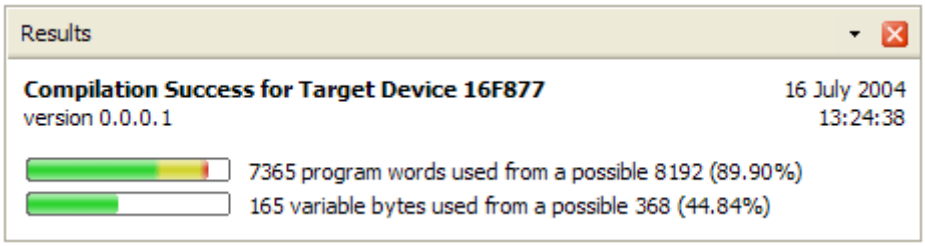


Results View

La vista de los resultados realiza dos tareas principales. Estas son: el despliegue de una lista de mensajes del error, y proporcionar un resumen en el éxito de la compilación.

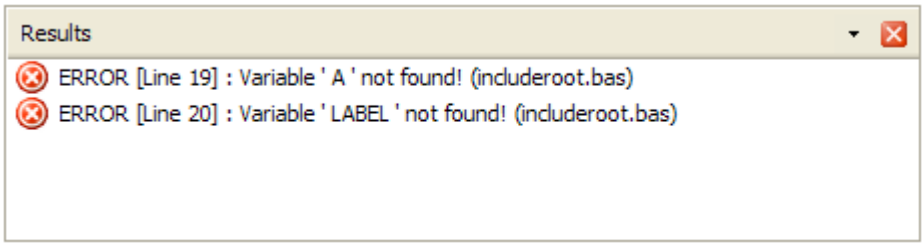
Compilation Success View

Por defecto, una exitosa compilación desplegará la vista de éxito de resultados. Esto proporciona la información sobre el dispositivo usado, la cantidad de código y datos usados, el número de la versión del proyecto y también la fecha y tiempo.



Compilation Error View

Si el programa genera advertencia o mensajes del error, la vista del error siempre se despliega.



Haciendo clic en cada error o advertencia del mensaje resaltarán la línea automáticamente en la ventana del editor principal.

De vez en cuando, el recopilador generará un archivo ASM válido pero se generan advertencias o errores. El protón IDE desplegará todas las advertencias del ensamblador o mensajes del error en el “vea error”.

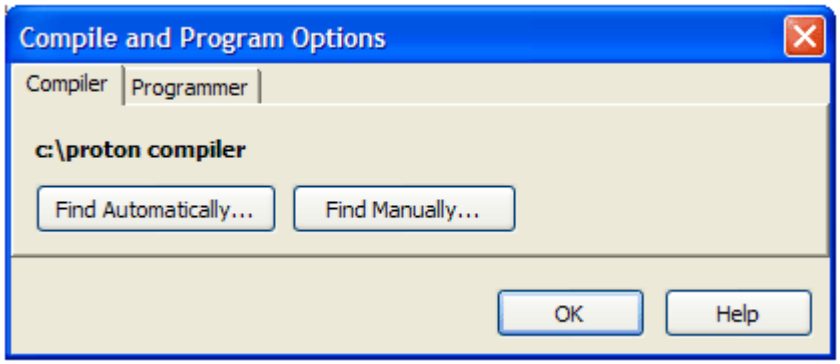
Editor Options

El diálogo de opciones de editor permite configurar y controlar los rasgos del Protón de IDE. La ventana está compuesta de cuatro áreas principales que se acceden seleccionando: General, Propiedades, Título del Programa y etiquetas de Actualización En línea.

Las Opciones generales
Las Opciones de Propiedades
Opciones de creación
Las Opciones de Actualización en línea.

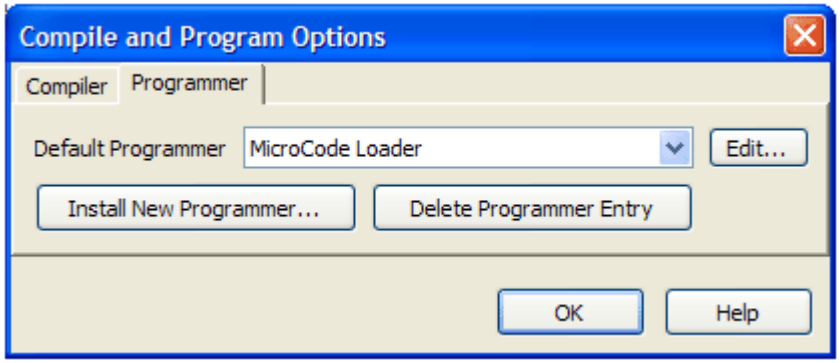
Opciones de Compilación y Programación

Compiler Tab



Se puede utilizar al Protón IDE para localizar un directorio del compilador automáticamente haciendo clic en el botón buscar. El rasgo del auto-búsqueda detendrá cuando un compilador se encuentre.

Programmer Tab



La etiqueta del programador se usa para instalar a un nuevo programador, se anula una entrada del programador o ésta revisa al programador actualmente seleccionado. Apretando Install New Programmer el Nuevo botón del Programador invocará el instalalador nuevo.

Dispositivos Soportados.

El Cargador de MicroCode apoya los dispositivos siguientes: 16F870, 16F871, 16F873(A), 16F874(A), 16F876(A), 16F877(A), 16F87, 16F88, 18F242, 18F248, 18F252, 18F258, 18F442, 18F448, 18F452, 18F458, 18F1220, 18F1320, 18F2220, 18F2320, 18F4220, 18F4320, 18F6620, 18F6720, 18F8620 y 18F8720,,,,.

Cargador Toolbar

Open Hex File

Dicho botón abre el archivo *. hexadecimal listo para programar.

Program

El botón del programa, programará el código de archivo hexadecimal cargado y datos de EEPROM en el microcontrolador designado. Al programar el dispositivo designado, una comprobación normalmente se hace para asegurar la integridad del código del usuario programado y los datos de la memoria EEPROM.

Apretando el botón del programa normalmente programarán el archivo
*. hexadecimal cargado.

Read

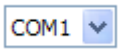
El botón leer, leerá el código actual y datos de EEPROM del microcontrolador designado.

Verify

El botón verifique, compara el archivo *.hexadecimal actualmente cargado y datos de la EEPROM localizados en el microcontrolador designado.

Erase

El botón borre, borrará toda la memoria del código en los microcontroladores PIC 16F8x y PIC18Fxxx(x)



Loader Serial Port

El cargador del puerto de serie despliega la caja que permite seleccionar el puerto de comunicación con el microcontrolador designado.

ABS	**12**	**14**	**16**

Assignment Variable = **ABS** *Variable*

Devuelva el valor absoluto de un número.

ACOS	**12**	**14**	**16**

Assignment Variable = **ACOS** *Variable*

Deduce el Arco Coseno de un valor.

ADIN	**12**	**14**	**16**

Variable = **ADIN** número de canal.

Lee el valor del Conversor Análogo Digital en una tabla.

ASIN	12	14	16

Asigna la variable = **ASIN** Variable.

Deduce el Arco Seno de un valor.

ASM...ENDASM	12	14	16

ASM

Nemónicos del ensamblador

ENDASM

@ El código nemónico del ensamblador

Los nemónicos se pasan directamente al ensamblador sin el recopilador que interfiere de forma alguna. Esto permite mucha flexibilidad que puede lograrse usando exclusivamente no siempre las órdenes BÁSICAS.

ATAN	12	14	16

Asigna la variable = **ATAN** Variable.

Deduce el Arco Tangente de un valor.

BOX	12	14	16

BOX *Set_Clear* **,** *Xpos Start* **,** *Ypos Start* **,** *Size*

Dibuja un cuadrado en un LCD gráfico.

BRANCH	12	14	16

El Índice de la RAMA, [Label1 {,... Labeln}]

Hace al programa saltar a las situaciones diferentes basada en una variable índice inconstante. En un dispositivo Microcontrolador PIC con sólo una página de memoria.

BRANCHL	12	14	16

BRANCHL Index, [Label1 {,... Labeln}]

Causa en el programa un salto para situaciones diferentes basado en una variable índice. En un dispositivo Microcontrolador PIC con más de una página de memoria.

BREAK	12	14	16

BREAK (ROMPA)

Exit a for...next, while...wend or repeat...until loop prematuramente.

BUTTON	12	14	16

BUTTON *Pin* , *DownState* , *Delay* , Rate , *Workspace* , *TargetState , Label*.

El botón en la entrada realiza una auto repetición, y envía la dirección si el botón está en el estado designado. Los circuitos del botón pueden ser en activación en alto o activación en bajo**.**

CALL	12	14	16

CALL Etiqueta

Ejecuta el subprograma nombrado la etiqueta.

CDATA	12	14	16

CDATA {datos alfanuméricos}
Se pone la información directamente en la memoria para el acceso por CRead y CWrite.

CF_INIT	12	14	16

CF_INIT

Inicializa las líneas usadas para el acceso a la memoria compacta flash por CF_Sector, CF_Read y CF_Write.

CF_SECTOR	12	14	16

CF_SECTOR *Sector Number* **,** *Operation* **,** *{Amount of Sectors}*

Instala el sector en la tarjeta flash Llamarada Compacta que será escrita o se leída por las órdenes CF_Read y CF_Write.

CF_READ	12	14	16

Variable **= CF_READ**

Lee los datos de una tarjeta compacta flash.

CF_WRITE	12	14	16

CF_WRITE {*Address Data* **,**} **[** Variable {*Variable* {**,** *Variable etc*}**]**

Escribe los datos a una tarjeta compacta flash.

CIRCLE	12	14	16

CIRCLE *Set_Clear* **,** *Xpos* **,** *Ypos* **,** *Radius*

Dibuja un círculo en un LCD gráfico.

CLEAR	12	14	16

CLEAR *Variable* or *Variable.Bit*

Pone una variable en un estado bajo. Si ninguna variable está presente después del orden, todo la área de la RAM en usó del Microcontrolador PIC se aclara.

CLEARBIT	12	14	16

CLEARBIT *Variable* , *Index*

Aclara un bit de una variable o registro que usan un índice inconstante al bit de interés.

CLS	12	14	16

CLS

Aclara los LCD alfanuméricos o gráficos y lugares en el cursor a la posición, es decir línea 1, posición 1.

Complement	12	14	16

El operador del Complemento **(~)**, saca el complemento de los bits de un número. Cada vez que en un número que contiene un 1 se cambia a 0 y cada bit que contienen 0 se cambia a 1. Este proceso es conocido como un "bitwise NO".

CONFIG	12	14	16

CONFIG { *configuration fuse settings* }

Habilita o Desactiva las escenas del fusible particulares para el tipo de Microcontrolador PIC usado.

COS	12	14	16

Asignación Variable = **COS** *Variable*

Deduce el Coseno de un valor.

COUNTER	12	14	16

Variable **= COUNTER** *Pin* , *Period*

Cuenta el número de pulsos que aparecen en el pin durante el período, y guarde el resultado en la variable.

CREAD	12	14	16

Variable **= CREAD** *Address*

Lee los datos en cualquier parte de en la memoria.

CURSOR	12	14	16

CURSOR *Line , Position*

Mueve la posición del cursor en el LCD a una línea especificada y posición.

CWRITE	12	14	16

CWRITE *Address* , **[** *Variable* { , *Variable…*} **]**

Escribe los datos en cualquier parte de la memoria.

DATA	12	14	16

DATA {*alphanumeric data* }

Pone la información en la memoria del código que usa la instrucción de RETLW.

DCD	12	14	16

DCD

Acepta un valor de 0 a 15, e ingresa un número del 16 bits, con ese número de bits del bit puesto a 1.

DEC	12	14	16

DEC *Variable*

Realiza el decremento de una variable es decir VAR1 = VAR1 – 1

DECLARE	12	14	16

{**DECLARE**} *code modifying directive = modifying value*

Ajusta ciertos aspectos del código producido, es decir la frecuencia de cristal, puerto y pines del LCD, la proporción serial del baudio etc.

DELAYMS	12	14	16

DELAYMS *Length*

Tarda la ejecución del programa en milisegundos (ms). Los Retrasos pueden durar 65535ms (65.535 segundos) de tiempo.

DELAYUS	12	14	16

DELAYUS *Length*

Tarda la ejecución del programa en microsegundos (us). Los Retrasos pueden durar 65535us (65.535 milisegundos) de tiempo.

DEVICE	12	14	16

DEVICE *Device number*

Informa al compilador que el dispositivo de Microcontrolador PIC está usándose.

DIG	12	14	16

Variable **= DIG** *Value, Digit number*

Devuelve el valor de un digito decimal.

DIM	12	14	16

DIM *Variable* { *as* } { *Size* }

Todos los usos definidos deben declararse utilizando la declaración DIM.

DTMFOUT	12	14	16

DTMFOUT *Pin* , { *OnTime* } , { *OffTime*, } **[** *Tone* {, *Tone*…} **]**

Produce un Tono DTMF en la secuencia del pin.

EDATA	12	14	16

EDATA *Constant1* {*Constant* etc}

Declara lugares constantes o stings directamente en la memoria EEPROM del Microcontrolador PIC compatible**.**

ENABLE	12	14	16

Habilita el proceso de la interrupción que fue previamente habilitado.

END	12	14	16

END

La declaración del Fin, detiene la compilación de fuente, y crea una vuelta infinita.

EREAD	12	14	16

Variable **= EREAD** *Address*

Lee la información de la EEPROM disponible en algunos tipos de MicrocontroladorPIC.

EWRITE	12	14	16

EWRITE *Address* , **[** *Variable* {, *Variable*…etc } **]**

Escribe la información de la EEPROM disponible en algunos tipos de Microcontroladores PIC.

EXP	12	14	16

Assignment Variable = **EXP** *Variable*

Deduce la función exponencial de un valor. Éste es el valor de **e,** de dónde **e** es la base de logaritmo natural.

FOR NEXT	12	14	16

FOR *Variable* **=** *Startcount* **TO** *Endcount* **[STEP** { *Stepval* } **]**
{*code body*}
NEXT

El **for...next** es usado para ejecutar una declaración, o serie de declaraciones en una cantidad predeterminada de tiempos.

FREQOUT	12	14	16

FREQOUT *Pin* , *Period* , *Freq1* { , *Freq2*}

Genera uno o dos tonos de la señal, o las mismas frecuencias, para un período especificado.

GETBIT	12	14	16

Variable **= GETBIT** *Variable*, *Index*

Examina un bit de una variable, o registro.

GOSUB	12	14	16

GOSUB
Label
GOSUB *Label* **[***Variable***,** {*Variable*, *Variable*... etc}**]**

GoSub salta el programa a una etiqueta definida y continúa la ejecución de allí en adelante. Una vez que el programa encuentra el Retorno el programa devuelve a la instrucción que sigue.

GOTO	12	14	16

GOTO *Label*

Salta a una etiqueta definida y continúa la ejecución de allí.

HBRESTART	12	14	16

HBRESTART

Envía un REINICIO la condición al bús de I2C que usa el módulo de MSSP del Microcontrolador PIC.

HBSTART	12	14	16

HBSTART

Envíe una condición de la SALIDA al bús de I2C que usa el MSSP del Microcontrolador PIC.

HBSTOP	12	14	16

HBSTOP

Envía una condición de PARADA al bús de I2C que usa el módulo de MSSP del Microcontrolador PIC.

HIGH	12	14	16

HIGH *Port* or *Port.Bit*

Pone un bit del Puerto en un estado alto. Para un Puerto, esto significa la salida llenándolo de unos.

HPWM	12	14	16

HPWM *Channel* , *Dutycycle* , *Frequency*

Saca un tren de ancho de pulsos modulados que usa los módulos de CCP el hardware de PWM, disponible en algún Microcontrolador PICs. Los pulsos de PWM producidos pueden correr continuamente en el fondo mientras el programa está ejecutando otras instrucciones.

HRSIN	12	14	16

Variable = **HRSIN** , **{** *Timeout* , *Timeout Label* **}**

HRSIN { *Timeout* , *Timeout Label* **}** , **{** *Parity Error Label* **}**

Recibe uno o más valores del puerto serie en dispositivos que contienen un hardware USART.

HRSOUT	12	14	16

HRSOUT *Item* { , *Item...* }

Transmite uno o más Artículos del hardware el puerto de serie (USART) en dispositivos que apoyan las comunicaciones de serie asíncronas en el hardware.

HSERIN	12	14	16

HSERIN *Timeout* , *Timeout Label* , *Parity Error Label* ,

[*Modifiers* , *Variable* {, *Variable...* }**]**

Recibe uno o más valores del puerto serie en dispositivos que contienen un hardware USART.

HSEROUT	12	14	16

HSEROUT [*Item* { *Item...* }

Transmite uno o más Artículos del hardware del puerto serie en dispositivos que apoyan las comunicaciones serie asíncronas en el hardware.

IF...THEN...ELSE	12	14	16

Syntax

IF *Comparison* **THEN** *Instruction* **:** { *Instruction* }
IF *Comparison* **THEN** *Instruction* **:** { *Instruction* } **: ELSEIF** *Comparison* **THEN** *Instruction* **: ELSE** *Instruction*

IF *Comparison* **THEN**
Instruction(s)
ELSEIF *Comparison* **THEN**
Instruction(s)
{
ELSEIF *Comparison* **THEN**
Instruction(s)
}
ELSE
Instruction(s)
ENDIF

Evalúa la comparación y, si la comparación no se cumple la instrucción se ignora, se lleva a cabo hasta que el End If se encuentre.

INC	12	14	16

INC *Variable*

Incrementa una variable VAR1 = VAR1 + 1

INCLUDE	12	14	16

INCLUDE "*Filename*"

Incluye otro archivo al punto actual en la compilación. Todas las líneas en el nuevo archivo se compilan como si ellos estuvieran en el archivo actual.

INKEY	12	14	16

Variable **= INKEY**

Examina un teclado pequeño y pone el valor devuelto en la variable.

INPUT	12	14	16

INPUT *Port* **.** *Pin*
Fija una entrada en el pin del Puerto especificado.

LCDRED	12	14	16

Variable **= LCDREAD** *Line Number* , *Xpos*

Lee un byte de un LCD gráfico.

LCDWRITE	12	14	16

LCDWRITE *Line number* , *Xpos* , **[** *Value* ,{ *Value etc*…} **]**

Escribe un byte en un LCD gráfico.

LDATA	12	14	16

LDATA { *alphanumeric data* }

Pone la información del código en la memoria que usa la instrucción de RETLW cuando se usa con los dispositivos de centro de 14 bits y memoria flash.

LINE	12	14	16

LINE *Set_Clear* **,** *Xpos Start* **,** *Ypos Start* **,** *Xpos End* **,** *Ypos End*

Dibuja una línea recta en cualquier dirección en un LCD gráfico.

LINETO	12	14	16

LINETO *Set_Clear* **,** *Xpos End* **,** *Ypos End*

Dibuja una línea recta en cualquier dirección en un LCD gráfico, empezando de la posición del fin del orden de la Línea anterior.

LOADBIT	12	14	16

LOADBIT *Variable* , *Index* , *Value*

Aclara, Pone un bit de una variable o registro que usan un índice inconstante para apuntar al bit de interés.

LOG	12	14	16

Assignment Variable = **LOG** *Variable*

Deduce el Logaritmo Natural de un valor.

LOG10	12	14	16

Assignment Variable = **LOG10** *Variable*

Deduce el Logaritmo de un valor.

LOOKDOWN	12	14	16

Variable **= LOOKDOWN** *Index* , **[** *Constant* { , *Constant*…etc } **]**

Busca constantes para el valor del índice. Si el índice empareja una de las constantes, entonces guarda la posición de la constante emparejando la variable.

LOW	12	14	16

LOW *Port* or *Port.Bit*

Pone un Puerto o bit en un estado bajo. Para un puerto, esto significa la llenándolo de ceros.

MAX	12	14	16

Retorna el máximo de dos números. Su uso es limitar los números a un valor específico. Su sintaxis es:

VAR2 = VAR1 **MAX** 100 ' Set VAR2 to the larger of VAR1 and 100 (VAR2 will lie between values ' 100 and 255)

MIN	12	14	16

Retorna el mínimo de dos números. Su uso es limitar los números a un valor específico. Su sintaxis es:

ON GOTO	12	14	16

ON *Index Variable* **GOTO** *Label1* {,...*Labeln* }

Causa al programa un salto a las situaciones diferentes basado en un índice inconstante. En un dispositivo Microcontrolador PIC con sólo una página de memoria.

ON_INTERRUPT	12	14	16

ON_INTERRUPT {Goto} *Label*

ON_HARDWARE_INTERRUPT {Goto} *Label*

Salta a un subprograma cuando una interrupción del HARDWARE ocurre.

ON_LOW_INTERRUPT	12	14	16

ON_LOW_INTERRUPT *Label*

Salta a un subprograma cuando una interrupción de HARDWARE prioridad baja ocurre en un dispositivo de centro de 16 bits.

ORG	12	14	16

ORG *Value*

Pone el origen del programa para el código subsecuente a la dirección definida en el Valor.

PEEK	12	14	16

Variable **= PEEK** *Address*

Recupera el valor de un registro y lo pone en una variable.

PIXEL	12	14	16

Variable **= PIXEL** *Ypos* , *Xpos*

Lee la condición de un pixel individual en un LCD gráfico 64x128. El valor devuelto será 1 si el pixel está en clear, y 0 si el pixel no está en clear.

PLOT	12	14	16

PLOT *Ypos* , *Xpos*

Setea un pixel individual en un 64x128 elemento LCD gráfico.

POKE	12	14	16

POKE *Address* , *Variable*

Asigna un valor a un registro.

POT	12	14	16

Variable **= POT** *Pin*, *Scale*

Lee un potenciometro, thermistor, fotocelda, u otra resistencia de la variable.

PRINT	12	14	16

PRINT *Item* { , *Item...* }

Envía el Texto a un módulo LCD.

RCIN	12	14	16

Variable = **RCIN** *Pin* , *State*

Cuenta el tiempo del pin en el estado, normalmente usa la medida de la carga / tiempo de la descarga del circuito resistor/capacitor (RC).

READ	12	14	16

READ *Variable*

Lee el próximo valor de una tabla de DATOS y pone en la variable asignada.

REM	12	14	16

REM *Comments* or **'** *Comments* or **;** *Comments*

Inserta los recordatorios en el código básico de la fuente. Estas líneas no se compilan y se usan para proporcionar la información a la persona que ve el programa fuente.

REPEAT...UNTIL	12	14	16

REPEAT *Condition*
Instructions
Instructions
UNTIL *Condition*

Ejecuta un bloque de instrucciones hasta que una condición sea verdad.

RESTORE	12	14	16

RESTORE *Value*

Mueve el pin en una tabla de datos a la posición especificada por el valor.

RSIN	12	14	16

Variable **= RSIN , {** *Timeout Label* **}**

RSIN { *Timeout Label* **}**, *Modifier*..*Variable* { , *Modifier*.. *Variable*...}

Recibe uno o más bytes de un pin predeterminado a una proporción del baudio predeterminada en formato asíncrono normal que usa 8 bits de los datos, sin paridad y 1 bit de parada (8N1). El pin es hecho automáticamente una entrada.

RSOUT	12	14	16

RSOUT *Item* { , *Item...* }

Envía uno o más Artículos a un pin predeterminado a una proporción del baudio predeterminada en formato asíncrono normal que usa 8bits de los datos, sin paridad y 1 bit de parada (8N1). El pin es automáticamente hecho una salida.

SELECT...CASE	12	14	16

SELECT *Expression*
 CASE *Condition(s)*
  Instructions
 {
 CASE *Condition(s)*
  Instructions

 CASE ELSE
  Statement(s)
  }
ENDSELECT

Evalúa una expresión ejecutando un bloque de código BÁSIC basado en las comparaciones de las condiciones. Después de ejecutar un bloque de código, el programa continúa a la línea que sigue el EndCase. Si ninguna condición se encuentra para ser Verdad y un bloque Case Else es incluido, el código después del Case Else que lleva al EndSelect y se ejecutará.

SERIN	12	14	16

SERIN *Rpin* { **** *Fpin* } , *Baudmode* , { *Plabel*, } { *Timeout* , *Tlabel*, } **[** InputData **]**

Recibe los datos serie asíncronos (es decir los datos de RS232).

SEROUT	12	14	16

SEROUT *Tpin* { **** *Fpin* } , *Baudmode* , { *Pace*, } { *Timeout* , *Tlabel*, } **[** *OutputData* **]**

Transmite los datos serie asíncronos (es decir los datos de RS232).

SERVO	12	14	16

SERVO *Pin* , *Rotation Value*

Controla un mando remoto tipo servo motor.

SET	12	14	16

SET *Variable* or *Variable.Bit*

Pone una variable o bit en un estado alto. Para una variable, esto significa con unos.

SETBIT	12	14	16

SETBIT *Variable* , *Index*

Pone un bit de una variable o registro que usan un índice inconstante al bit de interés.

SHIN	12	14	16

SHIN *dpin* , *cpin* , *mode* , **[** *result* { *bits* } { ,*result* { *bits* }...} **]**
Var = **SHIN** *dpin* , *cpin* , *mode , shifts*

Cambia los datos de un dispositivo síncrono serie.

SHOUT	12	14	16

SHOUT *Dpin*, *Cpin*, *Mode*, **[***OutputData* {*Bits*} {,*OutputData* {*Bits*}..} **]**

Cambia los datos fuera de un dispositivo serie síncrono.

SIN	12	14	16

Assignment Variable = **SIN** *Variable*

Deduce el Seno de un valor.

SLEEP	12	14	16

SLEEP { *Length* }

Los lugares del Microcontrolador PIC entran en el modo de bajo poder para la aproximación en segundos de n.

SNOOZE	12	14	16

SNOOZE *Period*

Entra en el modo de descanso para un período corto. El consumo de Power se reduce a aproximadamente 50 µA que no asumen las cargas que están manejándose.

SOUND	12	14	16

SOUND *Pin*, **[** *Note*,*Duration* {,*Note*,*Duration...*} **]**

Genera un tono o el ruido blanco en el pin especificado. El pin es automáticamente hecho una salida.

SQR	12	14	16

Assignment Variable = **SQR** *Variable*

Deduce la Raíz Cuadrada de un valor.

STOP	12	14	16

STOP

Para el programa de ejecución enviando el Microcontrolador PIC a una vuelta infinita.

SWAP	12	14	16

SWAP *Variable, Variable*

Intercambia los valores de cualquiera de las dos variables entre sí.

SYMBOL	12	14	16

SYMBOL *Name* { = } *Value*

Asigna un alias a un registro, variable, o valor constante**.**

TAN	12	14	16

Assignment Variable = **TAN** *Variable*

Deduce la tangente de una variable.

UNPLOT	12	14	16

UNPLOT *Ypos* , *Xpos*

Aclara un pixel individual en un elemento LCD gráfico de 64x128.

VAL	12	14	16

Variable = **VAL (***Variable* , *Modifier***)**

Convierte un byte array o sting que contiene DECIMAL, HEXADECIMAL o BINARIO en el equivalente entero.

VARPTR	12	14	16

Assignment Variable = **VARPTR (** *Variable* ***)***

Retorna la dirección de la variable en la RAM. Normalmente conocido como un puntero a una variable.

WHILE...WEND	12	14	16

WHILE *Condition*
Instructions
Instructions
WEND

WHILE *Condition* { *Instructions* ***:*** } **WEND**

Ejecuta un bloque de instrucciones mientras una condición es verdad.

XIN	12	14	16

XIN *DataPin* , *ZeroPin* , {*Timeout* , *Timeout Label*} , **[***Variable*{,...}**]**

Recibe los datos de X-10 y guarda el Código de origen y el Código Importante en una variable.

XOUT	12	14	16

XOUT *DataPin* , *ZeroPin* , **[***HouseCode******KeyCode* {*****Repeat*} { , ...}**]**

Transmite un código de origen seguido por una línea de código en el formato X-10.

- **OPERADORES DE COMPARACIÓN.**

Se usan en declaraciones IF... THEN para comparar una expresión con otra. Los operadores soportados se indican en la tabla 2.1.

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

Tabla. 2.1 *Operadores de Comparación*

- **OPERADORES LÓGICOS.**

Los operadores lógicos difieren de las operaciones de bit inteligente, entregan un resultado CIERTO / FALSO de su operación. Valores 0 son tomados como falso. Cualquier otro valor es cierto; se usan junto a operadores de comparación en una declaración IF... THEN.

Los operadores soportados se indican en la tabla 2.2.

Operador	Descripción
AND o &&	AMD lógico
OR o I	OR lógico
XOR o ^ ^	OR exclusivo lógico
NOT AND	NAND lógico
NOT OR	NOR lógico
NOT XOR	NXOR lógico

Tabla. 2.2 *Operadores Lógicos*

2.1.2 Síntesis del Manejo del Software IC – PROG

El IC - PROG es una herramienta fundamental para el programador de microcontroladores en general, ya que éste permite grabar el archivo .HEX en el micro.

En la figura 2.3 se puede apreciar que al iniciar el programa IC - PROG el código del programa aparece 3FFF. En todas las direcciones, la ventana abrir, buscamos la carpeta donde se tenga el programa realizado, y seleccionamos el archivo .HEX que deseamos cargar en el PIC.

Al cargar el programa .HEX, cambian los valores de dirección, por otros distintos a 3FFF, así podremos darnos cuenta que el programa a sido cargado en el programador.

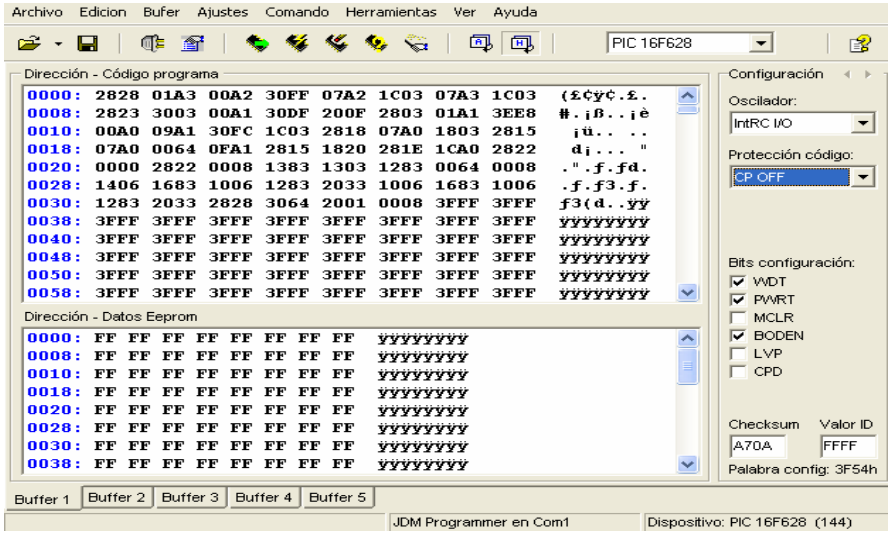
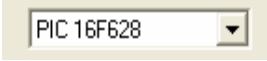


Fig 2.3 *Ventana del Software IC-PROG*

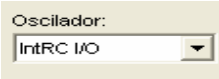
Permite leer el contenido del microcontrolador PIC

Este botón realiza la programación del microcontrolador, carga el archivo hexadecimal al PIC.

Permite borrar el contenido del microcontrolador *y* dejarlo en blanco para poder ser grabado nuevamente.



Como en el MicroCode Studio, en esta ventana, seleccionamos el microcontrolador que se esta utilizando.



En la ventana del oscilador, seleccionamos la configuración deseada, podemos cambiar al cristal interno de 4MHz que posee el PIC a la necesidad del usuario.

Los Bits de configuración, es recomendable desactivar la opción del MCLR *y* dejarlo siempre como se muestra en la figura.

2.1.3 Síntesis del Manejo del Software PROTEUS

PROTEUS es una herramienta que permite realizar simulaciones de los proyectos futuros a ser implementados.

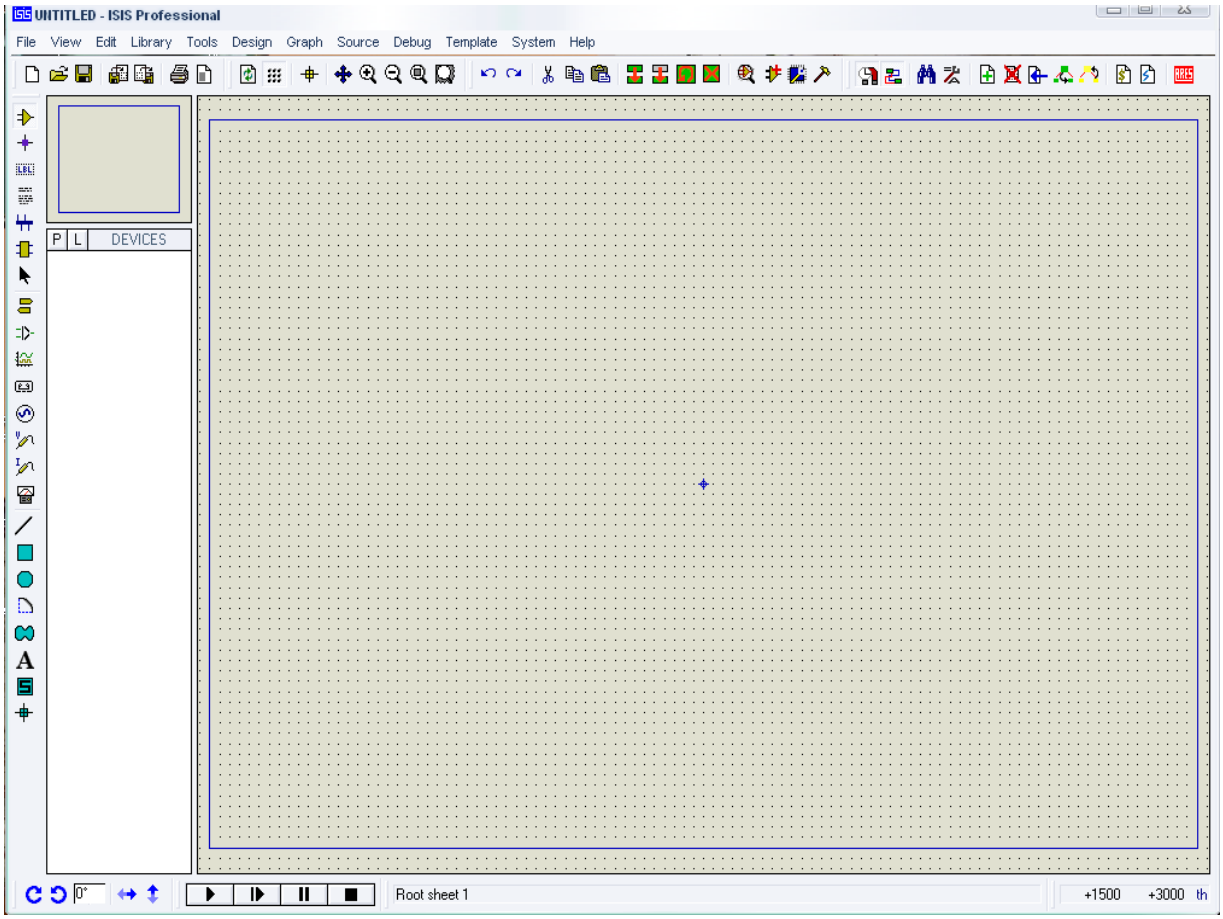
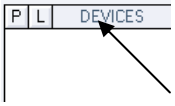


Fig.2.4 *Ventana del programa de PROTEUS*

Este icono permite seleccionar el elemento que deseamos utilizar, de la siguiente manera:



Seleccionamos el icono del componente dando un clic, seguido damos un clic en la letra P.

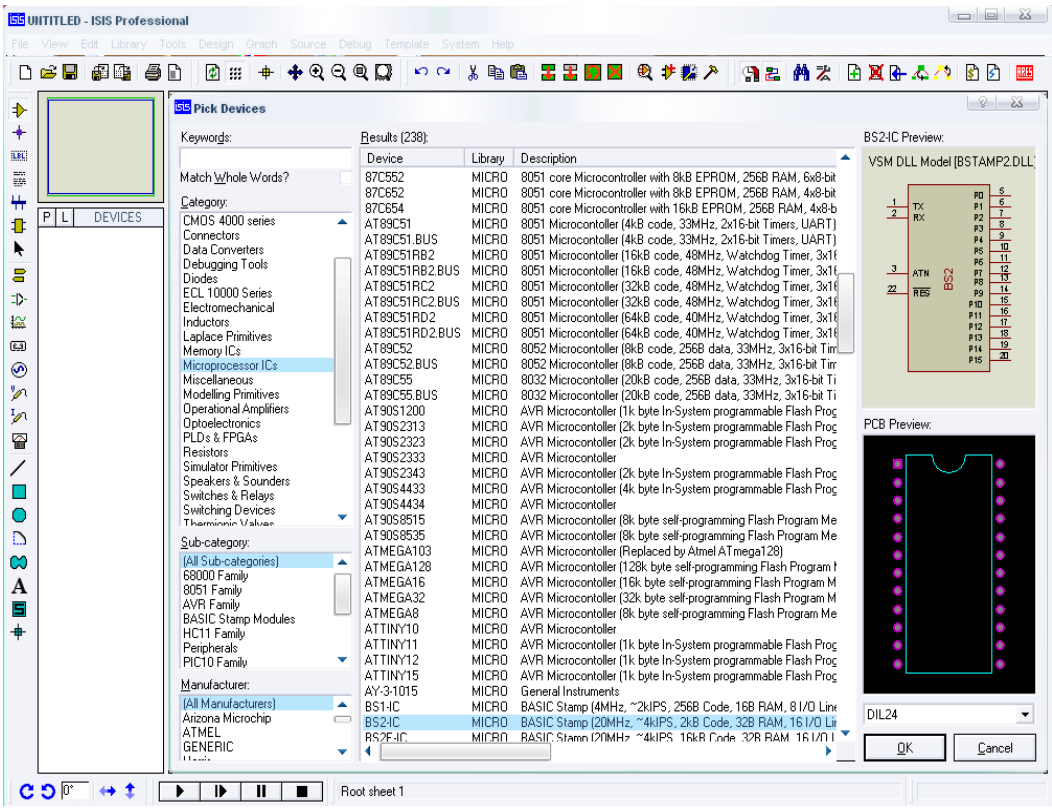


Fig2.5 *Ventana de componentes del programa de PROTEUS*

En la parte de Keywords: escribimos el elemento a utilizar, y en Results (1): damos doble clic para agregar el elemento a la pantalla del Proteus.

2.2 DISEÑO DEL SISTEMA DE MONITOREO Y CONTROL DEL PROCESO.

2.2.1 Diseño e implementación del hardware.

Los elementos necesarios para la implementación de los contadores para las cardas, constan principalmente de un microcontrolador PIC 16F877A[1], este circuito integrado es el que realiza el proceso de conteo de la producción diaria de hilo, cambios de turno y chequeos de fallas en la producción.

La tarjeta principal donde se realiza las conexiones de la red de los microcontroladores que componen cada uno de los contadores para las cardas, tiene conectores, para conectar respectivamente periféricos de entrada-salida,

[1] Anexo A1

dichos periféricos son colocados en la PC en donde se realiza el monitoreo de cada una de las máquinas de cardado.

- **Diagramas de Bloques del Sistema**

El sistema ésta conformado por los siguientes bloques que forman parte del sistema de conteo implementado.

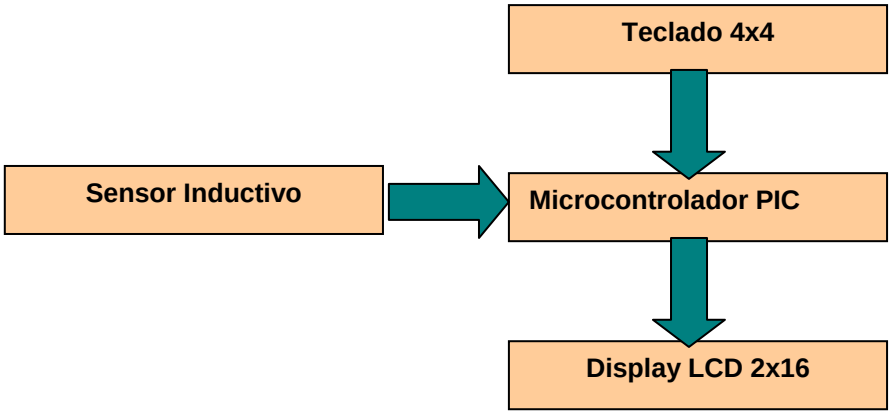


Figura 2.6 *Diagrama de Bloques de los Contadores.*

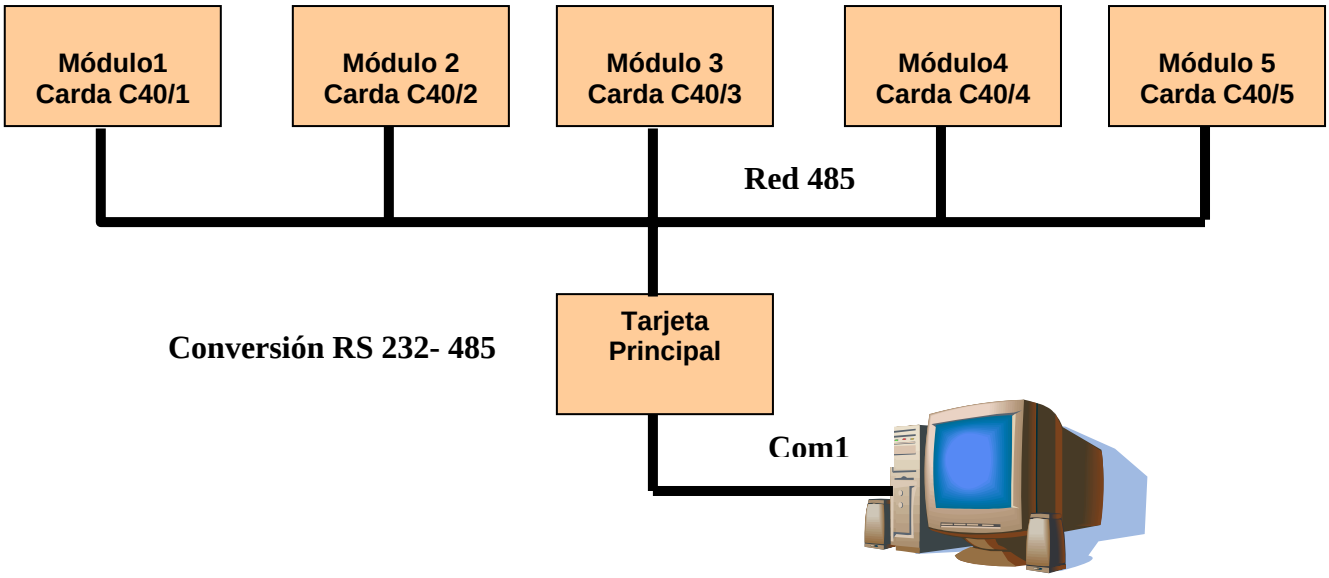


Figura 2.7 *Diagrama de Bloques de la red de Contadores.*

Donde:

- **Software de aplicación.-** Se utiliza una interfase gráfica, para el monitoreo de las velocidades de conteo, producción diaria, cambios de turno, este software fue implementado en LabVIEW7.1.

- **Computador personal.-** Debe utilizarse mínimo un computador Pentium II, instalado el sistema operativo Windows 98, o versiones y características superiores.

- **Microcontrolador.-** los microcontroladores utilizados en este proyecto son los PIC 16F877A, de la casa estadounidense Microchip.

- **Puerto Serie COM1.-** Puerto serial utilizado para la comunicación entre el PIC y la PC

- **Cables de conexión**. Cable UTP

Sensor Inductivo. Tipo NPN, este sensor tiene tres terminales, necesita de una fuente de alimentación de 12 ó 24Vdc, un terminal que envía la señal y el terminal de tierra.

2.2.2 Descripción del hardware del sistema

El sistema está diseñado y construido, aprovechando la versatilidad del puerto serie COM1, el cual conecta la computadora con la red realizada entre los microcontroladores que conforman cada uno de los contadores para las 5 máquinas encargadas de realizar el cardado. La tarjeta principal se encarga de realizar la red entre los microcontroladores utilizando la plataforma de la interfase RS485.

• **Distribución de pines del PIC 16F877A para el Sistema de Conteo de Cardado.**

El PIC 16F877A, dispone de 5 puertos de entrada salida configurables, en total cuenta con 32 líneas, las mismas que se distribuyen, en los distintos periféricos de entrada-salida, como se muestran en la tabla 2.3

PIC 16F877A	
Puerto	**Pin**
D.0	19
D.1	20
D.2	21
D.3	22
E.1	10
E.2	9

Tabla 2.3 *Pines que manejan los puertos del PIC16F877A*

• **LCD 2x16**[2]

Este periférico de salida, que se utiliza para la comunicación entre el usuario y el PIC, es conectado al puertos D del microcontrolador, distribuyendo la conexión de sus pines como se indica en la tabla 2.4

LCD 6 Líneas	
Línea	**Pin**
D4	27
D5	28
D6	29
D7	30
E	20
R/S	19

Tabla 2.4 *Pines que manejan el LCD*

[2] Anexo B1.2

- **Teclado 4x4**[3]

Este periférico de entrada, que se utiliza para seteo de parámetros que los realiza el usuario, es conectado al puerto B del microcontrolador.

La conexión de pines entre el PIC y el teclado se indica en la tabla 2.5.

PIC 16F877A		Teclado 8 Líneas
Puerto	**Pin**	**Descripción**
B.0	33	Fila A
B.1	34	Fila B
B.2	35	Fila C
B.3	36	Fila D
B.4	37	Columna 1
B.5	38	Columna 2
B.6	39	Columna 3
B.7	40	Columna 4

Tabla 2.5 *Pines que manejan el teclado*

- **Descripción funcional de la tarjeta principal.**

Esta tarjeta contiene los siguientes dispositivos electrónicos:

-1 CI SP232ACP[4]
-1 CI SN75176[5]
-Elementos Varios. (Resistencias, condensadores, conectores)

La tarjeta electrónica está principalmente conformada por la conversión de los niveles lógicos de voltaje entre la PC y la red de los microcontroladores, para lo

[3] Anexo B1.1
[4] Anexo A2
[5] Anexo A3

que utilizamos el CI SP232ACP para convertir niveles de voltaje TTL a RS232, seguido del CI SN75176, el cual nos permite realizar la conversión de RS232-485 para poder transmitir los datos a largas distancias.

- **FUENTE DE ALIMENTACIÓN**

La energía suministrada a cada uno de los módulos de conteo y a la tarjeta principal proviene de una fuente de alimentación que entrega voltajes de +5 VCC. La fuente es utilizada para alimentación del PIC, CI SP232ACP, CI SN75176, LCD y el sensor, que se encuentran tanto en los módulos contadores como en la tarjeta principal.

La figura 2.7 muestra el diagrama circuítal de la fuente de alimentación implementada en la tarjeta principal y en cada uno de los módulos de conteo.

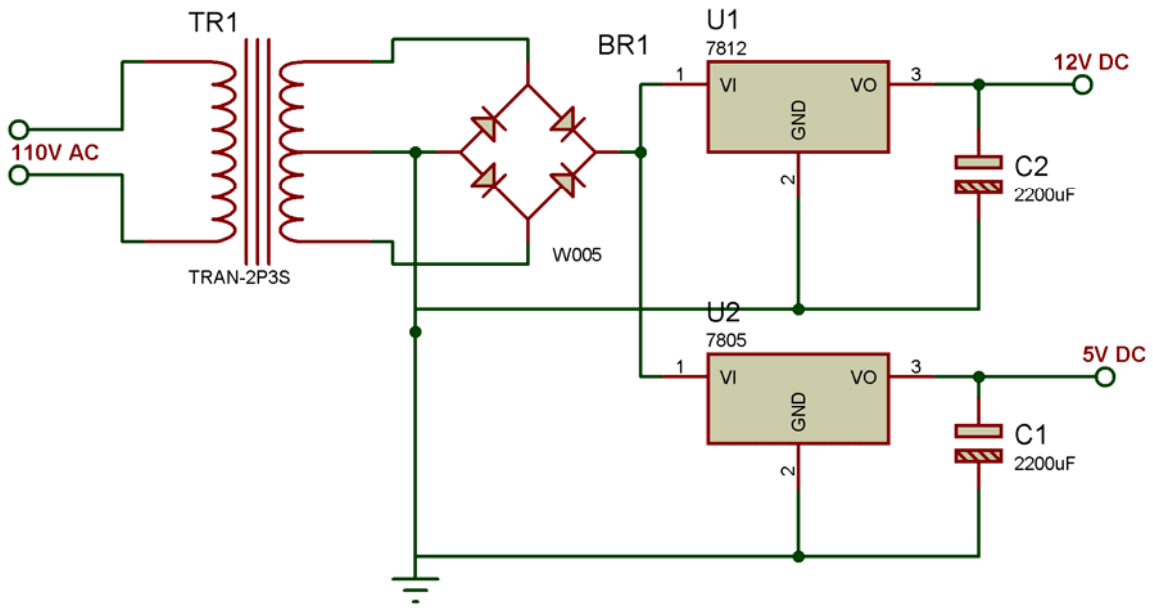


Figura 2.7 *Esquema de la fuente de alimentación*

2.2.3 Descripción de la Plataforma utilizada para la implementación del software.

El software fue desarrollado en LabVIEW7.1

INTRODUCCIÓN A LabVIEW 7.1

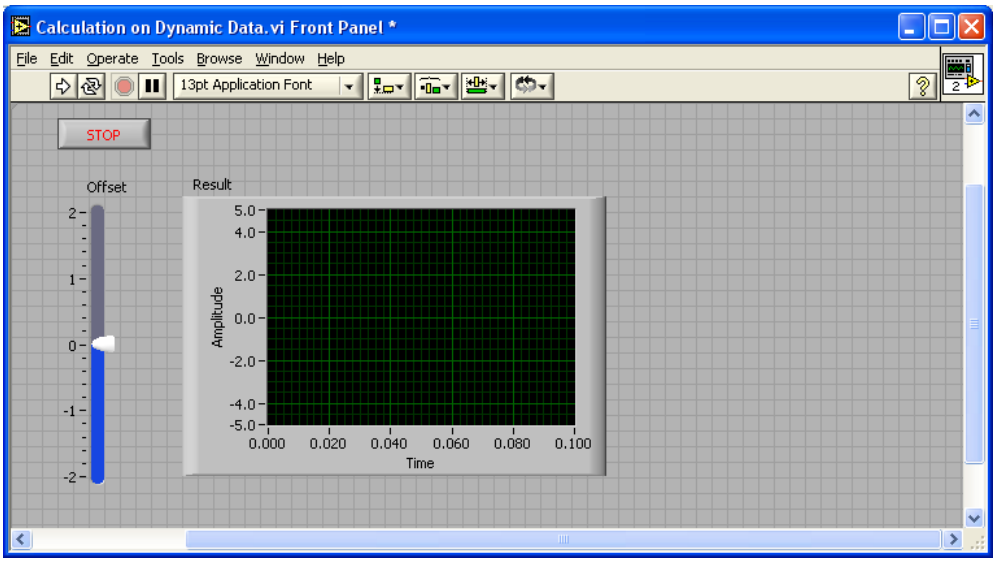
LabVIEW: Laboratory Virtual Instrument Engineering Workbench.
LabVIEW es un lenguaje de programación orientado a la instrumentación. Utiliza un entorno gráfico en lugar de líneas de texto para crear aplicaciones.

INSTRUMENTOS VIRTUALES

Un programa hecho en LabVIEW se denomina instrumento virtual (VI) porque generalmente modela la apariencia y funcionamiento de un instrumento físico.
Los VI's están compuestos de dos pantallas principales: El PANEL FRONTAL y el DIAGRAMA DE BLOQUES.

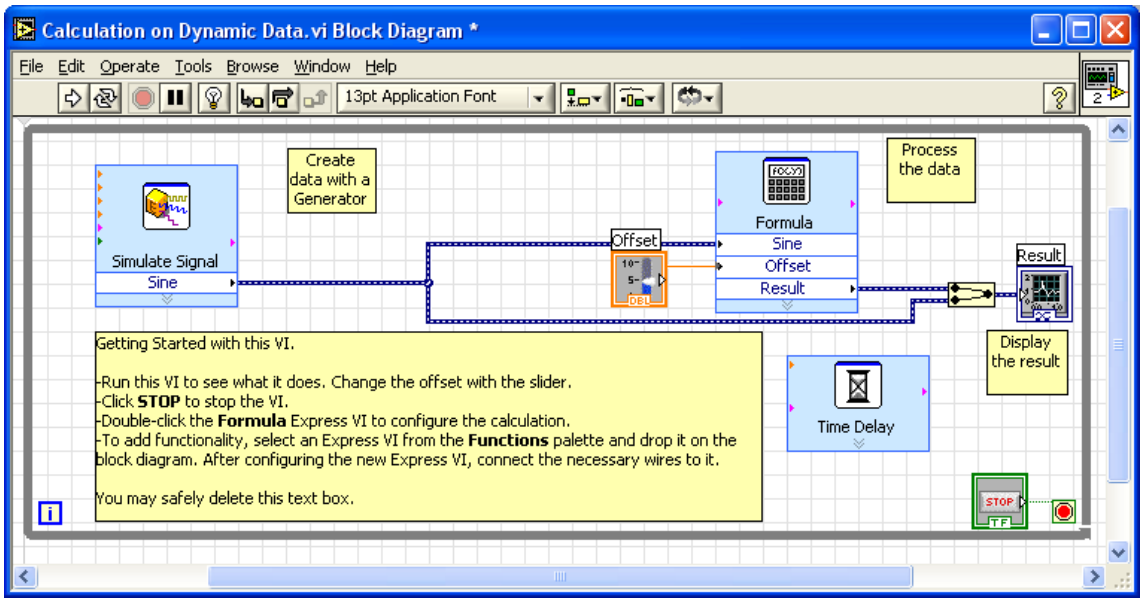
EL PANEL FRONTAL

Es la interfase de usuario. Contiene los controles e indicadores que son los terminales de entrada y salida del VI, respectivamente.

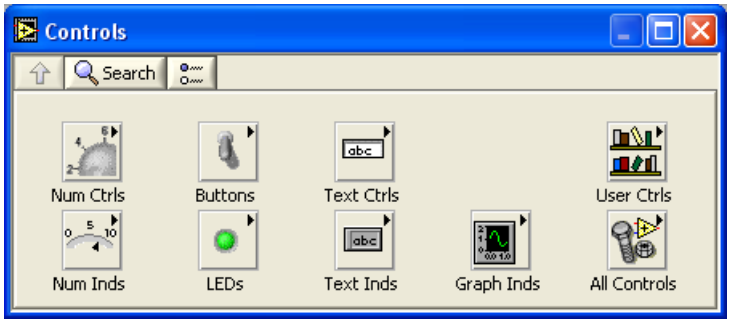


EL DIAGRAMA DE BLOQUES

Contiene el código que controla el funcionamiento de los objetos del panel frontal. Éste código utiliza representaciones gráficas de funciones y operaciones. Tiene una cierta similitud con un diagrama de flujo



LA PALETA DE CONTROLES



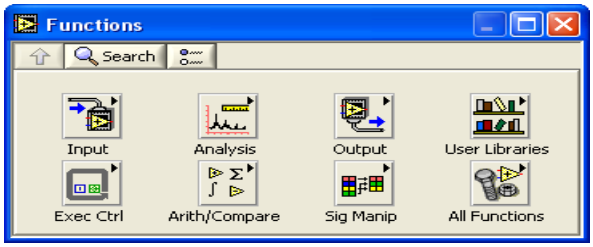
Paleta de controles Express

Paleta de controles Avanzada

Está disponible sólo en el panel frontal. Contiene los controles e indicadores utilizados para construir la interfase gráfica del VI.

Aparece al hacer clic derecho en el panel frontal o seleccionando *Window»Show Controls Palette* en la barra de menús.

LA PALETA DE FUNCIONES



Paleta de funciones Express

Paleta de funciones Avanzada

Está disponible sólo en el diagrama de bloques. Contiene los VI's y funciones utilizados para elaborar el código fuente.

Aparece al hacer clic derecho en el diagrama de bloques o seleccionando Window»Show Functions Palette en la barra de menús.

LA PALETA DE HERRAMIENTAS

Está disponible en el panel frontal y en el diagrama de bloques. Una herramienta es un modo de operación del cursor del mouse.

Aparece al hacer clic derecho mientras se presiona la tecla *Shift* o seleccionando *Window»Show Tools Palette*.

BOTONES DE LA PALETA DE HERRAMIENTAS Y SUS FUNCIONES

BOTÓN	FUNCIÓN
	Selección automática de herramienta**.** Al mover el cursor sobre los objetos, LabVIEW selecciona automáticamente la herramienta correspondiente de la paleta.
	Operación. Cambia el valor de un control o selecciona el texto de un control
	Posicionar. Posiciona, cambia el tamaño y selecciona objetos
	Etiquetar. Crea textos, etiquetas y edita textos existentes.
	Cablear. Une los objetos del diagrama de bloques utilizando líneas de conexión.
	Menú de objeto. Presenta el menú emergente de objeto. También se accede haciendo clic derecho sobre el objeto.

	Desplazamiento. Desplaza la ventana sin utilizar las barras de desplazamiento.
	Breakpoint. Se utiliza para fijar breakpoints (puntos de ruptura) en VIs, funciones, nodos, líneas de conexión y estructuras para pausar la ejecución en ese lugar.
	Punta de prueba. Crea puntas de prueba en las líneas de conexión, las puntas de prueba permiten visualizar rápidamente los valores producidos en el VI.
	Copiar color. Copia colores para luego utilizarlos con la herramienta colorear.
	Colorear. Sirve para fijar colores en los objetos del VI.

Tabla 2.6 *Botones de la Paleta de Herramientas*

LA BARRA DE HERRAMIENTAS

Se utiliza para ejecutar, depurar y editar un VI.

Barra de herramientas del Panel Frontal

Barra de herramientas del Diagrama de Bloques

BOTONES DE LA BARRA DE HERRAMIENTAS Y SUS FUNCIONES

BOTÓN	FUNCIÓN
	Ejecutar VI. El VI sólo puede ejecutarse si el botón tiene este icono.

	VI en ejecución
	El VI en ejecución es un subVI.
	El VI tiene errores. El VI no puede ejecutarse, se deben corregir los errores para que este icono desaparezca.
	Ejecución continua.
	Detener (Abortar) ejecución.
	Pausar y reiniciar ejecución. LabVIEW resalta el diagrama de bloques el lugar donde fue pausado el VI. La ejecución se reanuda al presionar nuevamente este botón.
	Abre un nodo (subVI o estructura) y pausa. Al presionar nuevamente, ejecuta la primera acción del nodo y pausa.
	Ejecuta un nodo y pausa.
	Termina la ejecución de un nodo y pausa.
13pt Application Font	Edición de texto.
	Resaltar ejecución. Presenta una animación de la ejecución en el Diagrama de Bloques.
	Alinear objetos.
	Distribuir objetos.
	Cambiar tamaño objetos.
	Reordenar objetos.
	Ingreso de texto. Le indica al usuario que se está modificando un valor de un control o una etiqueta mediante el teclado.
	Mostrar u ocultar ayuda contextual.

Tabla 2.7 *Botones de la Barra de Herramientas*

TIPOS DE DATOS

TIPOS DE DATOS DE CONTROLES E INDICADORES

La siguiente tabla muestra los símbolos para los diferentes tipos de datos. El color de cada símbolo indica el tipo de dato del control o indicador. Los controles tienen un borde más grueso que los Indicadores. En el Diagrama de Bloques aparece una flecha a la derecha si es control o una flecha a la derecha si es indicador.

Control	Indicador	Tipo de dato
		Numérico de punto flotante, precisión simple Rango: $\pm 1.40 \times 10^{-45}$ a $\pm 3.40 \times 10^{+38}$, 32 bits Dígitos decimales: 6
		Numérico de punto flotante, precisión doble Rango: $\pm 4.94 \times 10^{-324}$ a $\pm 1.79 \times 10^{+308}$, 64 bits Dígitos decimales: 15
		Numérico de punto flotante, precisión extendida Rango: $\pm 6.48 \times 10^{-4966}$ a $\pm 1.19 \times 10^{+4932}$, 128 bits Dígitos decimales: de 15 a 33 según de la plataforma
		Numérico complejo de punto flotante, precisión simple Rango: $\pm 1.40 \times 10^{-45}$ a $\pm 3.40 \times 10^{+38}$ (real e imaginario), 64 bits Dígitos decimales: 6
		Numérico complejo de punto flotante, precisión doble Rango: $\pm 4.94 \times 10^{-324}$ a $\pm 1.79 \times 10^{+308}$ (real e imaginario), 128 bits Dígitos decimales: 6
		Numérico complejo de punto flotante, precisión extendida Rango: $\pm 6.48 \times 10^{-4966}$ a $\pm 1.19 \times 10^{+4932}$ (real e imaginario), 256 bits Dígitos decimales: 6

		Entero con signo de 8 bits Rango: –128 a 127
		Entero con signo de 16 bits Rango: –32 768 a 32 767
		Entero con signo de 32 bits Rango: –2 147 483 648 a 2 147 483 647
		Entero sin signo de 8 bits Rango: 0 a 255
		Entero sin signo de 16 bits Rango: 0 a 65 535
		Entero sin signo de 32 bits Rango: 0 a 4 294 967 295

		Time stamp de 128 bits. Almacena valores de tiempo absoluto con alta precisión.
		Tipo enumerado
		Booleano. Tiene sólo dos posibles valores: TRUE (Verdadero) y FALSE (Falso). Su tamaño es de 8 bits.
		String. Contiene una cadena de caracteres. Su tamaño depende del número de caracteres.
		Array. El color y el tipo de dato que aparece entre los corchetes depende de los elementos que contiene.
		Cluster. Agrupa varios tipos de dato. Son de color café si todos los elementos del Cluster son numéricos o de color rosa si los elementos son de tipo diferente al numérico.
		Path. Contiene una ruta de acceso a un archivo.
		Dinámico. (Vis Express) Incluye datos asociados con unas señal, tales como: el nombre de la señal o la hora y fecha en la que fue adquirida.

		Waveform (Forma de onda). Contiene los datos, el tiempo de inicio, y el incremento de tiempo (Δt) de una forma de onda.
		Digital waveform (Forma de onda digital). Contiene el tiempo de inicio, el Δx, los datos digitales y otros atributos de una forma de onda digital.
		Digital. Contiene datos asociados con señales digitales.
		Número de referencia (refnum).
		Variante. Incluye el nombre del control o indicador, la información del tipo de dato y los datos mismos.
		Nombre I/O. Contiene los recursos configurados por el usuario para comunicarse con un instrumento o un dispositivo de medida.
		Picture. Despliega imágenes que contienen líneas, círculos, texto y otro tipo de gráficos.

Tabla 2.8 *Funciones de los Controles e indicadores*

CONTROLES E INDICADORES

Son los elementos fundamentales de un VI. Por medio de ellos es posible ingresar y presentar datos de forma interactiva.

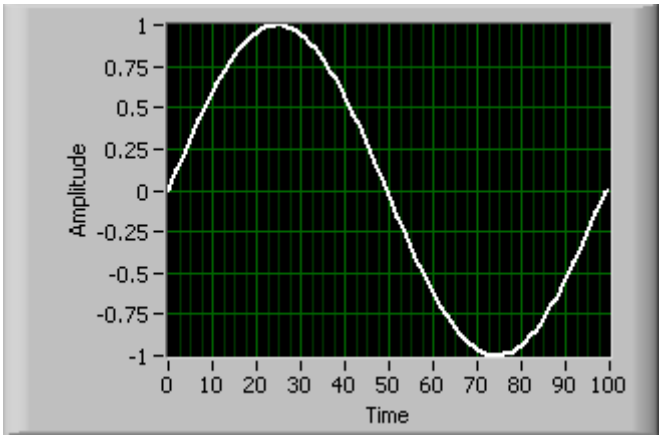
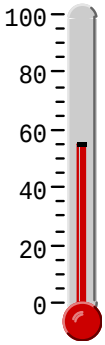
Se representan en el diagrama de bloques como iconos o terminales.

Cualquier objeto del panel frontal puede comportarse como control ó indicador, sin embargo, ciertos objetos están diseñados explícitamente para funcionar de una de estas formas. Por ejemplo una perilla funciona como control o un led funciona como indicador, si bien podrían funcionar de modo diferente, tendrían poca utilidad práctica.

EJEMPLOS DE CONTROLES COMUNES

OBJETO EN EL PANEL FRONTAL	OBJETO EN EL DIAGRAMA DE BLOQUES	
	Icono	Terminal
OK	OK TF	TF
	TF	TF
4 6 2 8 0 10	1.23 U8	U8
50	4 6 2 DBL	DBL
Hola Mundo!	abc abc	abc

EJEMPLOS DE INDICADORES COMUNES

OBJETO EN EL PANEL FRONTAL	OBJETO EN EL DIAGRAMA DE BLOQUES	
	Icono	Terminal
		
		

ARRAYS Y CLUSTERS

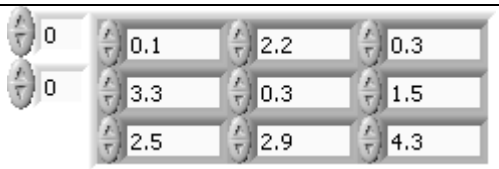
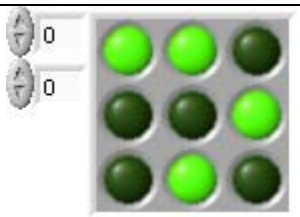
ARRAYS

Es un tipo de dato que agrupa elementos del mismo tipo.

Un array tiene elementos y dimensiones. Los elementos son los datos que componen el array. Las dimensiones son la longitud, altura o profundidad.

Un array de una dimensión se denomina vector, uno de dos dimensiones se denomina matriz.

EJEMPLOS DE ARRAYS

OBJETO EN EL PANEL FRONTAL	OBJETO EN EL DIAGRAMA DE BLOQUES	
	Icono	Terminal
		
		

CLUSTERS

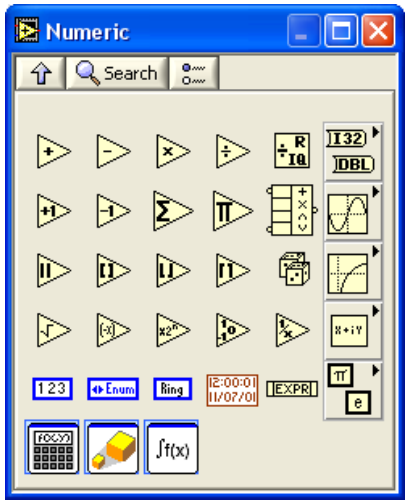
Es un tipo de datos que agrupa datos que pueden ser del mismo tipo o de diferente tipo.

Si el cluster es control todos sus elementos serán controles, si es indicador todos sus elementos serán indicadores.

OPERACIONES MATEMÁTICAS, LÓGICAS Y DE COMPARACIÓN

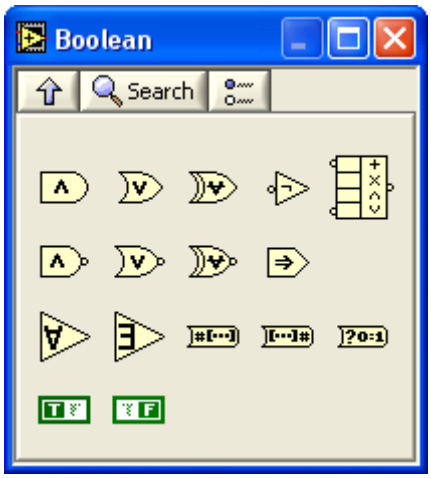
FUNCIONES NUMÉRICAS

Se usan para crear operaciones aritméticas, trigonométricas, logarítmicas y complejas con números y para realizar conversiones entre diferentes tipos de datos numéricos.



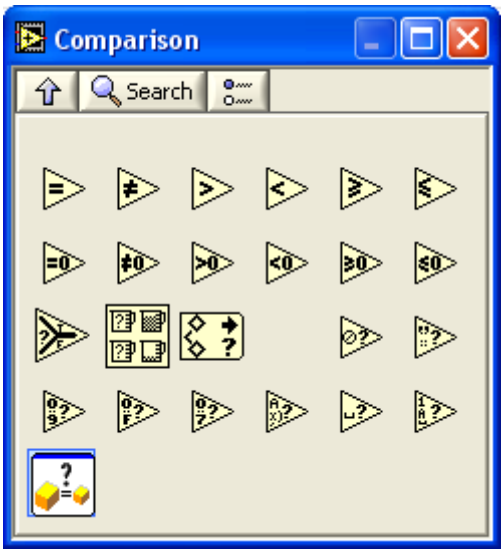
FUNCIONES BOOLEANAS

Se usan para ejecutar operaciones lógicas con valores Booleanos simples o con arrays de valores Booleanos.



FUNCIONES DE COMPARACIÓN

Se usan para comparar valores Booleanos, strings, numéricos, arrays y clústers.



VARIABLES LOCALES

Se utilizan para leer o escribir en los controles o indicadores del panel frontal de un VI.

La lectura y escritura en una variable local tiene el mismo efecto que leer o escribir sobre un control o indicador, respectivamente. Sin embargo, por medio de las variables locales también es posible escribir en un control y leer un indicador.

Existen variables locales de lectura y de escritura sin importar si el objeto es un control o un indicador.

ESTRUCTURAS

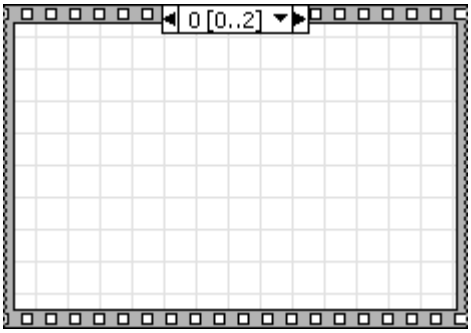
Las estructuras son representaciones gráficas de lazos y sentencias condicionales de los lenguajes de programación basados en texto.

Las estructuras se usan en el diagrama de bloques para repetir bloques de código, ejecutar código en forma condicional o en un orden específico.

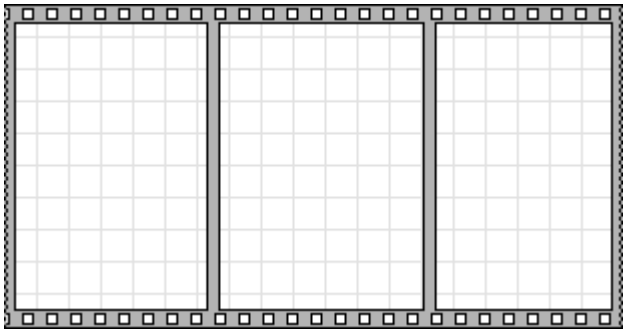
Todas las estructuras tienen un borde de tamaño variable que encierra la sección del diagrama (subdiagrama) de bloques que se ejecuta de acuerdo a las reglas de la estructura.

ESTRUCTURAS STACKED Y FLAT SEQUENCE

Consisten en uno o más subdiagramas, o marcos, que se ejecutan secuencialmente. Se utiliza para asegurar que un subdiagrama se ejecute antes o después de otro.



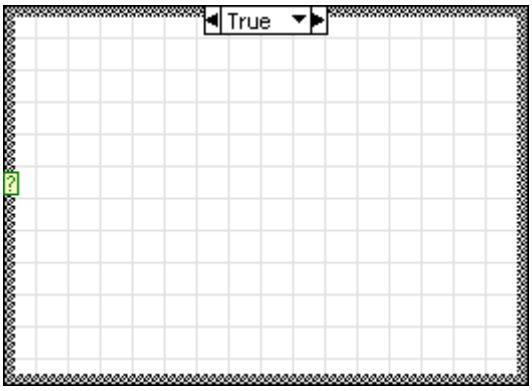
Estructura Stacked Sequence



Estructura Flat Sequence

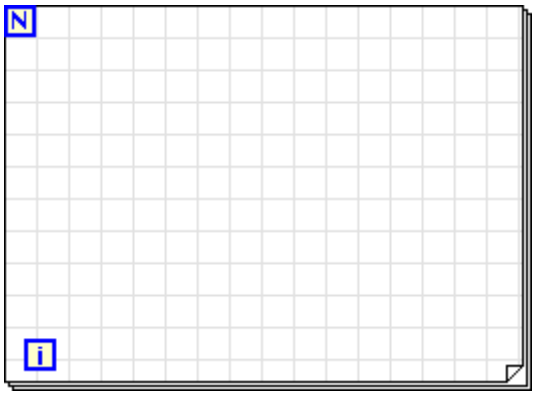
ESTRUCTURA CASE

Tiene uno o más subdiagramas o casos, sólo uno de ellos se ejecuta a vez, dependiendo del valor presente en el selector. Este valor puede ser Booleano, numérico, string ó enumerado.



LAZO FOR

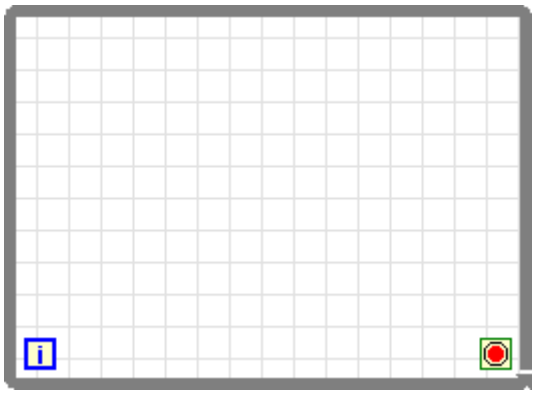
Ejecuta el subdiagrama el número de veces que se indique en el valor conectado en el terminal (N). El terminal (i) cuenta el número de iteraciones del lazo, su rango es de 0 a N-1.



LAZO WHILE

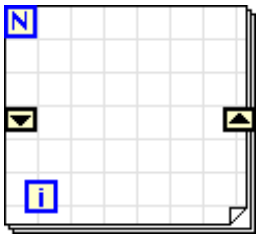
Repite el subdiagrama hasta que el terminal condicional recibe un determinado valor Booleano. Si la condición es *Stop if True* ( ) el lazo se detiene con un valor TRUE. Si la condición es *Continue if True* ( ) el lazo se detiene con un valor FALSE.

El terminal (i) tiene la misma función que en el lazo FOR, sin embargo aquí no tiene un rango definido sino que depende de la condición de parada del lazo.



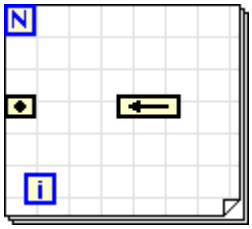
REGISTROS DE DESPLAZAMIENTO

Se utilizan en los lazos For y While para transferir valores de una iteración de lazo a la siguiente.



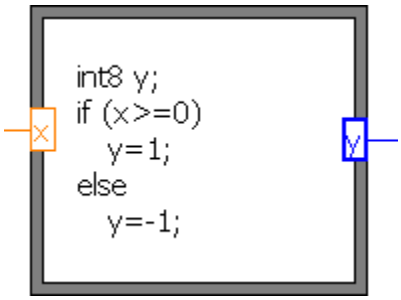
NODOS FEEDBACK

Su función es igual a la de un registro de desplazamiento. Aparece cuando se realiza una realimentación dentro de un lazo For o While. Puede reemplazarse con un registro de desplazamiento y viceversa.



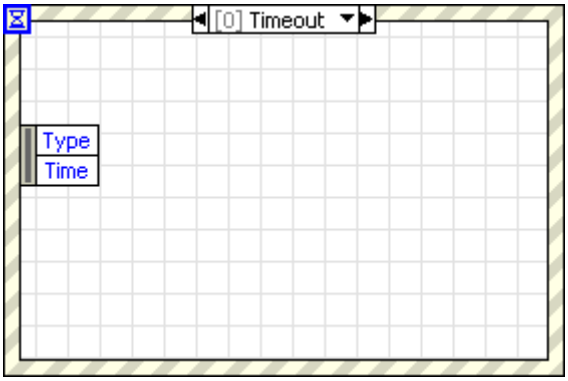
NODO FÓRMULA

Evalúa fórmulas y expresiones matemáticas con una sintaxis similar a la de lenguaje C.



ESTRUCTURA DE EVENTOS

Tiene uno o más subdiagramas o casos de evento. Sólo uno de ellos se ejecuta a la vez. Esta estructura espera hasta que ocurra un evento en el panel frontal, entonces ejecuta el caso apropiado para manipular dicho evento.



INDICADORES GRÁFICOS

WAVEFORM GRAPH Y WAVEFORM CHART

Estos indicadores se diferencian por la forma de presentar y actualizar los datos.

Los indicadores Waveform Graph toman los datos de un array y luego los grafican en función de su índice, es decir, de la posición de cada elemento en al array.

El indicador Waveform Chart, en cambio, adjunta nuevos datos a los ya existentes creando así un gráfico histórico, es decir, en un Waveform Chart se puede visualizar los datos actuales conjuntamente con los datos previamente adquiridos.

Cada uno de los puntos del eje horizontal tienen una separación constante y se puede asignar un solo valor, por esta razón estos indicadores se utilizan para graficar señales adquiridas a una tasa de muestreo constante.

2.3 IMPLEMENTACIÓN DEL SOFTWARE.

2.3.1 Programación del microcontrolador PIC 16F877A.

El firmware que contiene el microcontrolador PIC 16F877A, se programó en el compilador micro Code Studio, y su programa se muestra en el ANEXO B.

2.3.2Interfase gráfica

Para el desarrollo de la interfase gráfica se utiliza la plataforma LabVIEW7 Express. En la cual se muestra el monitoreo de los parámetros de producción de cada una de las cardas. (Ver Capitulo III)

Esta interfase permite la comunicación entre el usuario y el sistema, la misma que es de fácil entendimiento y amigable para el operador.

CAPÍTULO III

ANÁLISIS DE RESULTADOS.

3.1 Descripción funcional

El sistema tiene por objeto en forma conjunta o individual entre la PC y la tarjeta principal, realizar el monitoreo de parámetros en la máquina de cardado RIETER C4, utilizando la interfase gráfica realizada en LABVIEW 7.1 o mediante un teclado principal de 4*4 colocado en el panel principal de la máquina, la tarjeta principal se encarga de comunicar cada uno de los contadores a través de la interfase RS485 al HMI. La construcción de los contadores de producción de hilo se diseñaron para poder monitorear localmente y remotamente con los microcontroladores PIC 16F877A los mismos que se encargan de recibir la señal externa proveniente de un sensor inductivo, posteriormente esta es a la vez acondicionada para poder filtrar el ruido proveniente de los motores, el monitoreo en tiempo real que se realizan a las cardadoras son: las frecuencias del tren de pulsos que nos proporciona el sensor, la velocidad expresada en RPM de cada uno de los motores que se encuentran en las cardas, la producción por cada uno de los turnos y la velocidad de entrega.

El cardado es un proceso en el cual se endereza parcialmente las fibras y forma con ellas una trama delgada que se unen en una cuerda suave conocida como cinta de cardado. La máquina para cardar se compone de cilindros cubiertos con una guarnición gruesa y pesada, de elementos llamados chapones y otros elementos como el peinador, otros cilindros de salida y de una apiladora donde se recoge la cinta en botes que es el resultado del proceso de esta máquina.

El material previamente procesado ingresa a la tolva de alimentación, la cual guía y regula la cantidad de material que va ingresar al cilindro alimentador que es controlado electrónicamente para obtener una alimentación uniforme dependiendo tanto de los dispositivos de la entrada que es la tolva, en la salida los cilindros calandradores que es por donde se obtiene la cinta de carda se mide la variación de la cinta, un motor controla el sistema de regulación mediante la variación de velocidad para tener una cinta de material uniforme a la salida.

Los cilindros de alimentación son los que lleva el material al likerin el cual es el encargado de abrir y limpiar el material, mediante estiramientos se logra obtener una fina capa de material en forma de velo el cual es llevado por el gran tambor, el mismo es forrado por una guarnición en forma de dientes de sierra, los chapones son los encargados de limpiar, sacar fibra corta y da uniformidad al velo que sigue la trayectoria hacia el peinador, el mismo dispone de unas guarniciones que arrastra el material y mediante rodillos se lleva hacia los cilindros calandradores procesos en el cual el velo se transforma en cinta que es recogida en botes, función que realiza la apiladora.

En el mismo eje de estos cilindros calandradotes se encuentra acoplado un disco con agujeros que da la señal al sensor de proximidad, este sensor es que da la información de la cantidad de material que se está entregando.

3.1.1 MÓDULO DE MONITOREO

El hardware principal del sistema esta formado por los siguientes subsistemas:

- Tarjeta Principal.
- Contadores de producción.

TARJETA PRINCIPAL.

Este subsistema tiene la capacidad de realizar el monitoreo del hardware que está interconectado a este, además realiza el proceso de comunicación con la PC, para dar la posibilidad de monitoreo mediante una interfase gráfica desarrollada en LabVIEW 7.1. Esta tarjeta está colocada en la sala de control. La figura 3.1 muestra el diagrama circuital. El ANEXO C (c.1) indica el la placa de circuito impreso de la tarjeta.

Esta tarjeta contiene los siguientes dispositivos electrónicos:

1 CI SP232ACP[1]
1 CI SN75176[2]

Elementos Varios. (Resistencias, condensadores, conectores)

La tarjeta electrónica está principalmente conformada por la conversión de los niveles lógicos de voltaje entre la PC y la red de los microcontroladores, para lo que utilizamos el CI SP232ACP para convertir niveles de voltaje TTL a RS232, seguido del CI SN75176, el cual nos permite realizar la conversión de RS232-485 para poder transmitir los datos a largas distancias y a la vez interconecta el software con el hardware. Está construido en una placa de baquelita con ruteo a un solo lado.

[1] Anexo A2
[2] Anexo A3

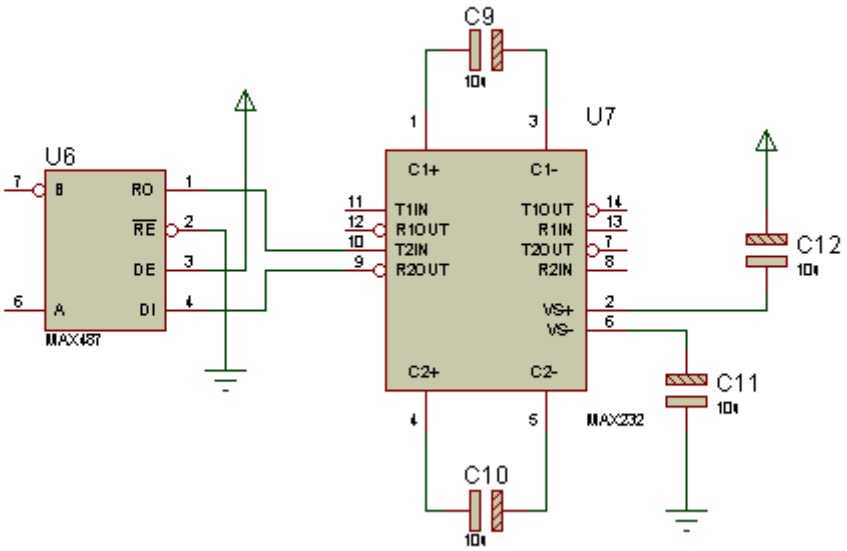


Figura 3.1 *Diagrama circuital de la tarjeta principal.*

CONTADORES DE PRODUCCION.

Este subsistema realiza la adquisición de datos de producción de hilo, la señal analógica que entrega el sensor de proximidad tipo inductivo. Su señal mediante un amplificador operacional, es acondicionada a un rango de 1 a 5 V, para ser enviada que ingrese al microcontrolador, el cual a través del pin 3 recibe la señal y mediante el firmware interno del PIC interpreta valores de producción por turnos, producción total, velocidad del motor interno de la máquina y la frecuencia de trabajo a través de un LCD. El ANEXO C(c.2) indica el la placa de circuito impreso de cada uno de los contadores.

La figura 3.2, muestra el diagrama circuital de esta placa.

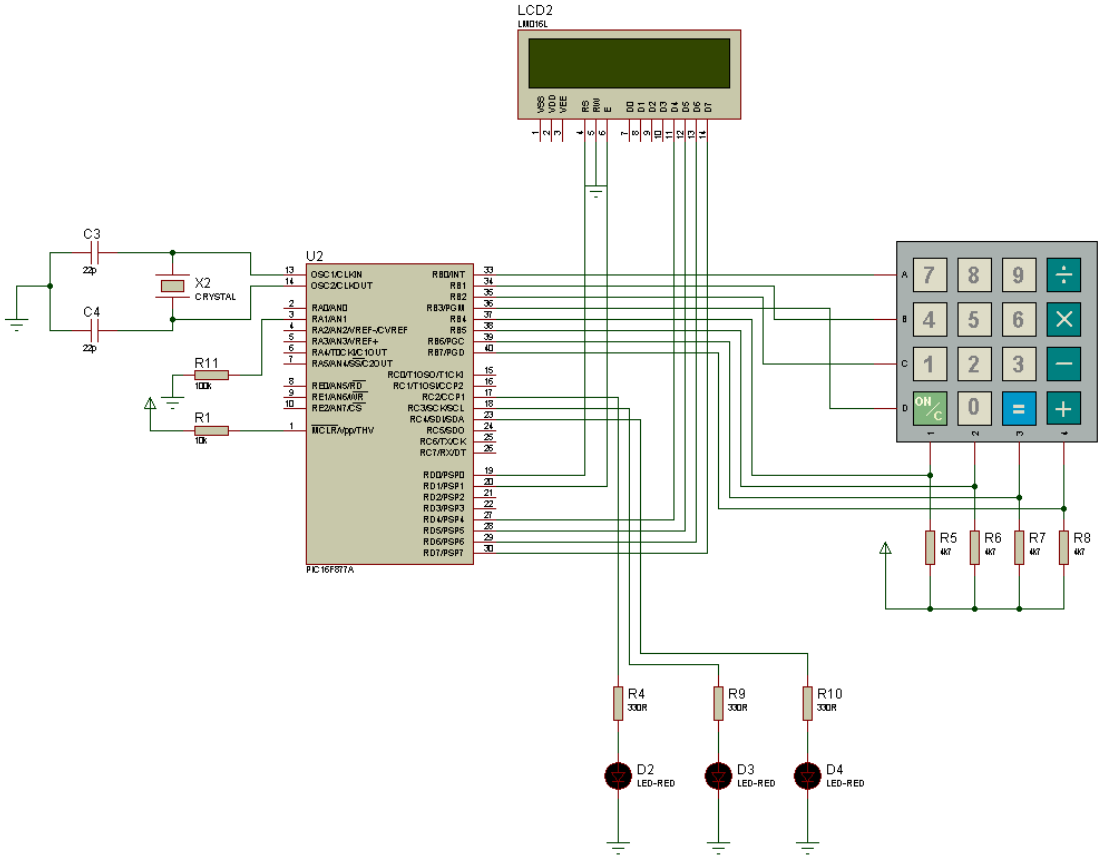


Figura 3.2 *Diagrama circuital de los contadores de producción.*

3.1.2 Cable de comunicación

El proceso se comunica con el software de aplicación mediante un cable construido especialmente para este proyecto.

El cable se conecta entre el puerto serie COM1 de la computadora y en el conector de la tarjeta principal.

Todas las señales enviadas y recibidas por la computadora están distribuidas en el cable, como se muestra en la figura 3.3.

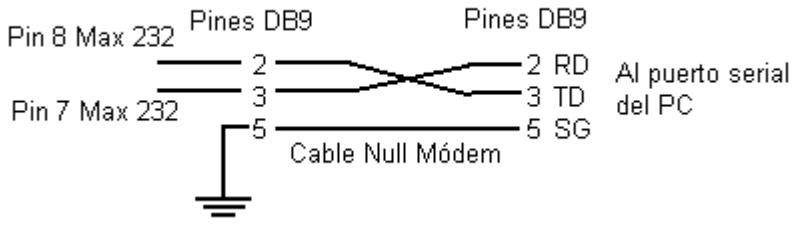


Figura 3.3 *Conexión entre la PC y el PIC mediante el C.I. Max 232.*

La figura 3.4 muestra una fotografía del cable construido para este proyecto.

Figura 3.4 *Cable para la comunicación*

3.2 Pruebas del sistema

El sistema fue sometido a varias pruebas experimentales presentando resultados exitosos. Las pruebas que se realizaron fueron tanto de la parte de Software como de Hardware.

Las pruebas experimentales del sistema en funcionamiento son las siguientes:

- Prueba funcional del Software
- Prueba funcional del Hardware
- Prueba en modo local

3.2.1 Prueba funcional del Software

Consiste en accionar controles y visualizar parámetros desde el computador. Para estas pruebas, el conector DB9 esta conectado al puerto serie COM1 del computador y a la tarjeta principal, y ésta a su vez conectada a la red de contadores a través de la interfase RS 485.

Las pruebas realizadas fueron las siguientes:

Ejecución del programa

Pulsando doble clic en el icono de acceso directo al programa de monitoreo LabVIEW 7.1 se verifica el ingreso al programa. La figura 3.5 indica el icono de acceso directo.

Figura 3.5 *Icono de acceso directo al programa*

- **Inicio de la Interfase**

Cuando se ingresa al sistema, la interfase principal mostrada en la figura 3.6 se inicia y está lista para monitorear variables del proceso de hilatura, ya que está configurado de esta manera.

En el caso de que se presione la tecla “CLOSE”, es necesario volver a ejecutar la interfase desde el acceso directo, ya que este control cierra la interfase.

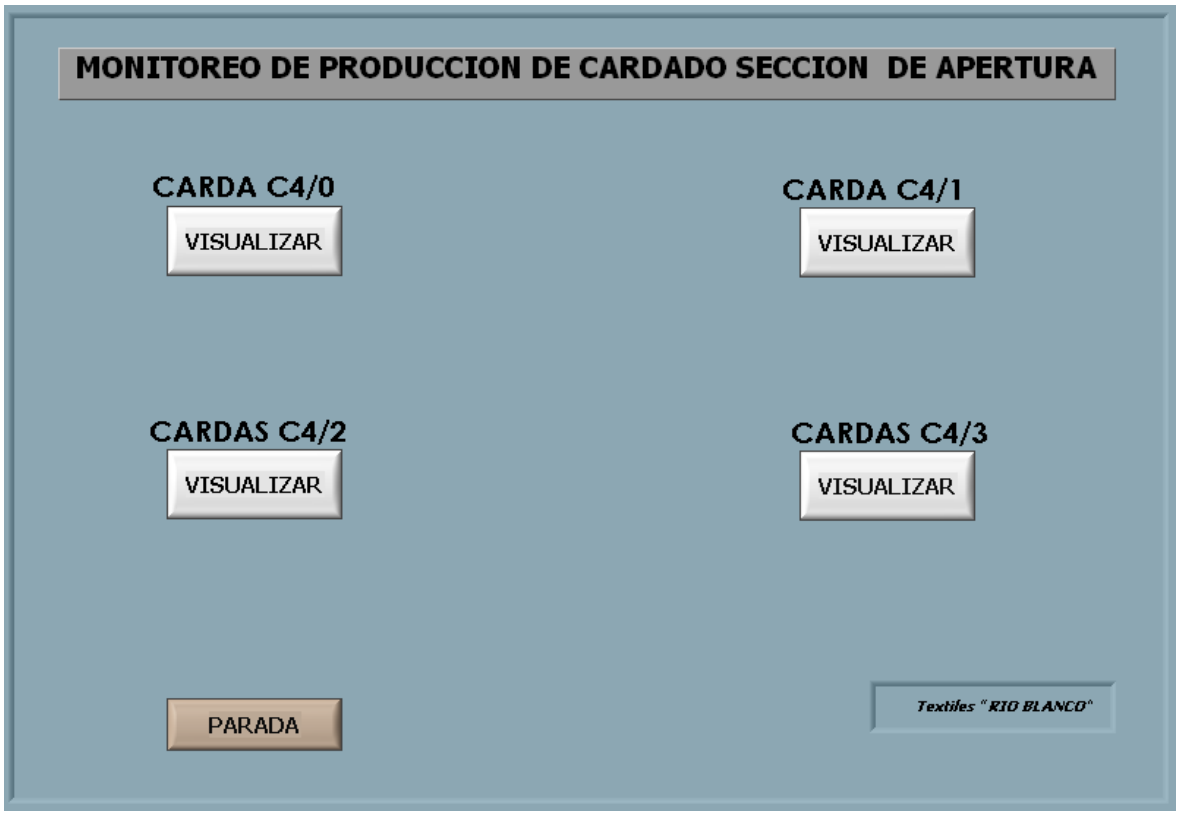


Figura 3.6 *Pantalla de acceso a los contadores de las cardas*

La pantalla de operador principal permite ingresar al monitoreo de los parámetros de cada una de las cardas. En la figura 3.7 se puede observar la plataforma de adquisición de datos de una de las cardas, pudiéndose observar los tres turnos diarios de producción de hilo, así también la producción por turnos, velocidad de entrega del material, velocidad del motor de la cardadora y la frecuencia del sistema.

Figura 3.7 *Pantalla de Operación del proceso de cardad.*

Con el botón “PARADA”, en la ventana principal se comprueba efectivamente la finalización de la interfase, con esto se cierra la posibilidad de monitorear el proceso, pero el dicho proceso de la máquina no se altera ya que este es controlado directamente por el PIC.

Los indicadores luminosos muestran cual de los turnos se encuentra vigente siendo tres turnos por día, estos indicadores están visibles tanto en el panel de monitoreo local tanto como en el panel de operador que se encuentra en la sala de control.

3.2.2 Prueba funcional del Hardware.

El equipo fue sometido a diferentes pruebas, verificando el buen funcionamiento de cada una de sus partes. Es decir el correcto funcionamiento del ingreso de los

números por teclado, comunicación entre el PIC y la (HMI) y la visualización de la producción por turnos diario de cada una de las cardadoras a través de los cuatro LCD gráficos los mismos que se encuentran en cada uno de los módulos de conteo de las cardadoras.

Pruebas del microcontrolador

Se hicieron varias pruebas las mismas que se detallan a continuación:
Se verificó la lectura y escritura de datos en la memoria EEPROM del microcontrolador, éstos permanecen almacenados en su respectiva localidad de memoria sin importar si se quita o no la alimentación.

El microcontrolador es sumamente versátil. Gracias a su amplia funcionalidad y a su fácil manejo.

Se realizó muchas operaciones de borrado y escritura tanto de la memoria flash como de la memoria EEPROM, y no hay inconveniente en su funcionamiento, el PIC está diseñado para soportar muchos procesos de lectura y escritura.

Se probó que el reset externo funciona efectivamente, este se ejecuta a través del pin 1, conectado con una resistencia a Vcc, para provocar el reset dicho pin se aterriza a tierra con un pulsador normalmente abierto, este se utiliza para el apagado de la parte de potencia, el mismo que está ubicado en el panel frontal de la máquina.

No se tuvo ningún inconveniente la comunicación entre el PIC y la plataforma LabVIEW 7.1, no hubo errores en la transmisión y recepción de datos, de tal manera que resulta confiable el intercambio de datos.

3.2.3 Prueba en modo local.

Esta prueba, se lo realiza solo localmente, está basado en prender el sistema de la máquina de cardado, específicamente el motor, el mismo que nos proporciona

la frecuencia a la cual se produce las fibras de hilo, variamos la velocidad del motor para tener diferentes velocidades de producción, y observamos navegando con el teclado personalizado cada uno de los parámetros que el operador necesita saber acerca del la producción diaria. El funcionamiento y las rutinas de operación que el trabajador manipula se detallan a continuación:

Cuando se energiza el panel de la máquina, aparece un menú de inicio con la que dá la bienvenida al operador de turno permitiéndole escoger el turno en el cual se va a monitorear la producción, estos son tres turnos por día. En el Anexo B (b3) se puede observar el teclado de operación del modulo de conteo.

Una vez que se prende la parte de potencia, el sistema comienza a tomar valores de medición de producción diaria de hilo, velocidad de entrega del producto, frecuencia y velocidad del motor de la cardadora, los mismos que son mostrados en el LCD. En el Anexo C (c4) se presenta el modulo completo de conteo.

Los indicadores Mostrados en el Anexo C (c5) presentan los diferentes turnos de operación, siendo estos tres turnos por día.

3.3 Alcances y limitaciones

3.3.1 Alcances

El proyecto desarrollado permite realizar las siguientes funciones:

Permite el monitoreo en tiempo real ya sea desde la PC o del panel principal de la máquina. En el caso de la PC se visualiza en red las cuatro cardadoras, permitiendo al usuario, en este caso, al operador de turno monitorear y manipular los diferentes parámetros de la maquina mediante una interfase gráfica; y en el caso que se desee controlar y visualizar parámetros desde el panel propio de la máquina, se lo hace mediante un menú que se despliega en el LCD controlado por un teclado 4*4.

- Permite controlar de la PC o desde el panel al mismo tiempo.

- La red realizada permite poder monitorear las cuatro cardadoras en tiempo real sin que se sea prioritario realizar el monitoreo localmente.

- Si durante la ejecución, se sale del programa LabVIEW 7.1, el proceso continua sin ninguna alteración, ya que el proceso es manejado directamente por el microcontrolador PIC 16F877A.

- Este proyecto está enfatizado a brindar una solución de ahorro económico y de mano de obra a las distintas empresas que se dedican a la producción de materiales, en el cual su principio se base en realizar el conteo de producción en metros por hora y las distintas equivalencias que se desglosan de esta.

- La red creada bajo la interfase RS 485 nos permite comunicar a las máquinas con la sala de control a larga distancia.

-

3.3.2 Limitaciones

- Si el sistema trabaja en funciones de multitarea se deberá utilizar una computadora de tecnología Pentium II en adelante, para que el sistema funcione adecuadamente.

- El sistema no permite guardar históricos ya que no se vió la necesidad de almacenar este tipo de información.

- El proyecto no maneja registro de usuario ya que la empresa no vio la necesidad de realizarla, debido a que el operador solo tiene acceso a las pantallas del panel frontal.

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES.

4.1 Conclusiones:

- Se diseñó y construyó el sistema de contadores de producción basado en microcontroladores que cumplen con las características y funciones iguales que los contadores de la marca WHALI tipo EZA1 propias de las cardas.

- La red construida es capaz de Monitorear el proceso productivo y los datos de producción en tiempo real.

- El sistema integra: un software de aplicación, un computador, y cuatro módulos de conteo de producción en red, que proporciona un instrumento que permite monitorear el proceso de cardado.

- El software fue diseñado utilizando como plataforma LabVIEW 7.1, el cual brinda un ambiente de diseño que proporciona al usuario un máximo de productividad para aplicaciones de automatización.

- La implementación de este proyecto ha permitido que este tipo de maquinaria que existe en la industria textilera pueda contar con contadores más económicos y que brinden mejores prestaciones que los ya existentes.

- Se implementó este proyecto sin la necesidad de invertir mucho dinero, razón importante para poner en funcionamiento las cardadoras que existen en la empresa las mismas que no contaban con un monitoreo de producción.

- La implementación del hardware permite monitorear en forma local los parámetros de producción del proceso de cardado.

- El sistema fue diseñado para poder comunicar los módulos de conteo mediante la interfase RS485 la misma que brinda varios recursos en la comunicación serial a larga distancia y la velocidad a la que los dispositivos se comunican con la PC.

- El sistema implementado ha resuelto un problema real, como es el de poner nuevamente a funcionar este tipo de máquinas.

- El diseño realizado en este proyecto puede utilizarse como modelo, para el monitoreo de cualquier proceso de producción de materiales en el cual se deseen saber la cantidad de producto en metros por hora o sus unidades equivalentes.

- Cada dispositivo conectado a la red tiene su código propio de reconocimiento con la interfase LabVIEW.

- El número máximo de componentes de la red utilizada en el proyecto es de 32 elementos.

- Empleando el circuito integrado MAX232 se logra integrar correctamente la interfase RS485 con la PC.

- La velocidad de la comunicación serie influye en la velocidad de muestreo para el envio y la recepción de datos desde la PC a cada uno de los módulos de conteo y viceversa.

- Con la comunicación serial asincrónico se logra disminuir a un número de tres líneas el cable empleado para la comunicación serial. Una para la transmisión, otra para la recepción de los datos y una tercera para tierra.

- El programa LabView, tiene instrucciones para manejar el puerto serial en forma amigable con el usuario, y facilita la programación que permite adquirir y enviar datos a la red creada.

- El sistema permite el monitoreo ya sea desde la PC o del panel local de la máquina. En el caso de la PC se visualiza la producción de las cuatro cardadoras, en sus tres turnos diarios mediante una interfaz gráfica; y en el caso que se desee visualizar parámetros desde el panel propio de la máquina se lo hace mediante un menú que se despliega en el LCD controlado por un teclado 4*4.

- Si durante la ejecución de la interfaz principal de operador, se oprime el botón parada desde el HMI, el proceso continua sin ninguna alteración, ya que el proceso es manejado directamente por el microcontrolador PIC 16F877A.

- La red proporciona el monitoreo de la producción diaria directamente desde la sala de control ahorrando así a la empresa personal de operación, recursos económicos en lo que respecta a tener un mejor sistema que evite pérdidas y errores de conteo en la producción diaria.

- El sistema no permite guardar históricos tampoco tiene la opción de registro de usuarios ya que la empresa no vió la necesidad de almacenar este tipo de información.

4.2 Recomendaciones

- Para garantizar una funcionabilidad óptima del sistema, se debe utilizar correcta y cuidadosamente el hardware y software.

- Antes de iniciar el proceso, se debe revisar que todas las conexiones entre la PC y la máquina estén correctas.

- La conexión entre el módulo y la computadora puede hacerse en cualquier momento que se requiera.

- El cable para realizar el intercambio de datos conectado al puerto serial, es único y debe estar conectado adecuadamente.

- El cable de comunicación entre la PC y la tarjeta principal no debe exceder de quince metros debido a especificaciones propias de esta interfase.

- El proyecto utiliza sistema operativo Windows 98 o superior.

- Si el sistema trabaja en funciones de multitarea se deberá utilizar una computadora de tecnología Pentium 1 en adelante, para que el sistema funcione adecuadamente.

- En la implementación del circuito se recomienda utilizar capacitores de aproximadamente 22 pF que van conectados de la salida del oscilador a tierra con el objetivo de eliminar el ruido.

- El conductor que se emplea como medio de transmisión de la red RS485 debe ser de preferencia UTP o un cable trenzado que permita eliminar ruido en la red.

- Se debe colocar a tierra las líneas que no sean utilizadas, para evitar tener niveles flotantes, que generen una información considerada como basura en la red.

- Fomentar nuevos trabajos con esta red en el área de la instrumentación, superando las limitaciones del sistema.

- Al realizare un proyecto de grado, incentivar a los alumnos por investigar y crear nuevos sistemas, que cubran las necesidades de la industria moderna.

GLOSARIO

ASCII.- American Standard Code for Information Interchange - Código estándar americano para el intercambio de información.

ASM.- (Asembler). Código ensamblador.

Banco de memoria.- Grupo de localidades en un espacio especificado de memoria.

BIT.- Es la unidad más pequeña de información y la unidad base en las comunicaciones.

Byte.- Conjunto de 8 bits continuos mínimos que hacen posible un direccionamiento de información en un sistema computarizado.

C A/D.- Conversor Analógico Digital

CI.- Circuito Integrado

CMOS.- (Complementary Metal Oxide Silicon). Silicio de oxido metálico.

Codificación.- Códigos metódicos y sistemáticos.

COM.- (Common Object Model). Modelo de objetos genéricos. Es un conjunto de redes distribuidas diseñadas para permitir a las aplicaciones Windows acceder a las aplicaciones y objetos sobre otras plataformas, COM integra un sistema de vinculación de objetos OLE.

Comunicación asíncrona.- Envío de datos no acoplado a una señal de reloj.

Comunicación síncrona.- Envío de datos acoplado a una señal de reloj.

Controlador.- Dispositivo que opera automáticamente para regular una variable controlada.

CPU.- (Center Process Unit). Unidad de procesamiento Central.

DCE.- (Data Communications Equipment). Equipos de conversión entre el DTE y el canal de transmisión.

DSP.- Procesador digital de señales.

DTE.- (Data Terminal Equipment). Equipos que son la fuente y destino de los datos.

E/S.- Entradas y Salidas.

EEPROM.- Memorias de solo lectura, borrable eléctricamente.

Flash memory.- Es una memoria que puede ser grabada y borrada en el circuito embebido.

GPR.- Registros de propósito general de microcontroladores Microchip.

Hardware.- Conjunto de componentes electrónicos, eléctricos y mecánicos que soportan la información y realizan operaciones para las cuales fueron diseñados.

HMI.- (Human Machine Interface). Interfase entre el humano y la máquina.

Interfases.- Conexión que permite la comunicación entre dos o más entidades.

LCD.- Display de cristal líquido.

MCU.- Microcontrolador.

MHz.- Unidad de medida de frecuencia (velocidad)

PC.- Computador personal, unión de hardware y software.

PLC.- Controlador lógico programable, sistema lógico que combina hardware y software, de modo que sus instrucciones codificadas puedan modificarse; proporciona automatización “flexible” en vez de “dedicada”.

PIC.- Peripheral Interface Controller (Controlador de Interfaz Periférico).

PSP.- Puerta paralela esclava

PWM.- Modulación por ancho de pulso.

RAM.- (Random Access Memory). Memoria de escritura y lectura de tipo volátil.

RS232C.- Norma internacional de transmisión serial entre dos dispositivos, que utiliza señales de voltaje para comunicarse. Debido a que se emplea una señal de voltaje, el RS-232C no puede transmitir señales a largas distancias (máximo 15 metros).

Set Point.- Punto en que una señal se establece bajo ciertos parámetros deseados. Es un punto de consigna para valor de la señal de la variable.

Sistema.- Es una combinación de componentes que actúan conjuntamente y cumplen un objetivo.

Sistema digital de monitoreo y control.- El término hace referencia a cualquier dispositivo basado en instrumentación y sistemas de computadoras o bien basados en microprocesadores, para funciones de control o de adquisición de datos.

Software.- Conjunto de instrucciones (programas) y datos que, almacenados en la memoria de la máquina, describen el trabajo a realizar.

SFR.- Registros de funciones especiales, permiten la configuración de microcontroladores de la empresa Microchip.

SSP.- Puerto serie síncrono

Tierra.- Potencial de la tierra física.

Transceiver.- Convertidor de señales.

Transductor.- Dispositivo que recibe una o varias señales provenientes de la variable medida y pueden modificarla o no en otra señal.

Transmisor.- Capta la señal del elemento primario de medida y la transmite a distancia en forma eléctrica, neumática, hidráulica, mecánica y ultrasónica.

TTL.- Lógica transistor – transistor, circuitos integrados de la serie 74XXX

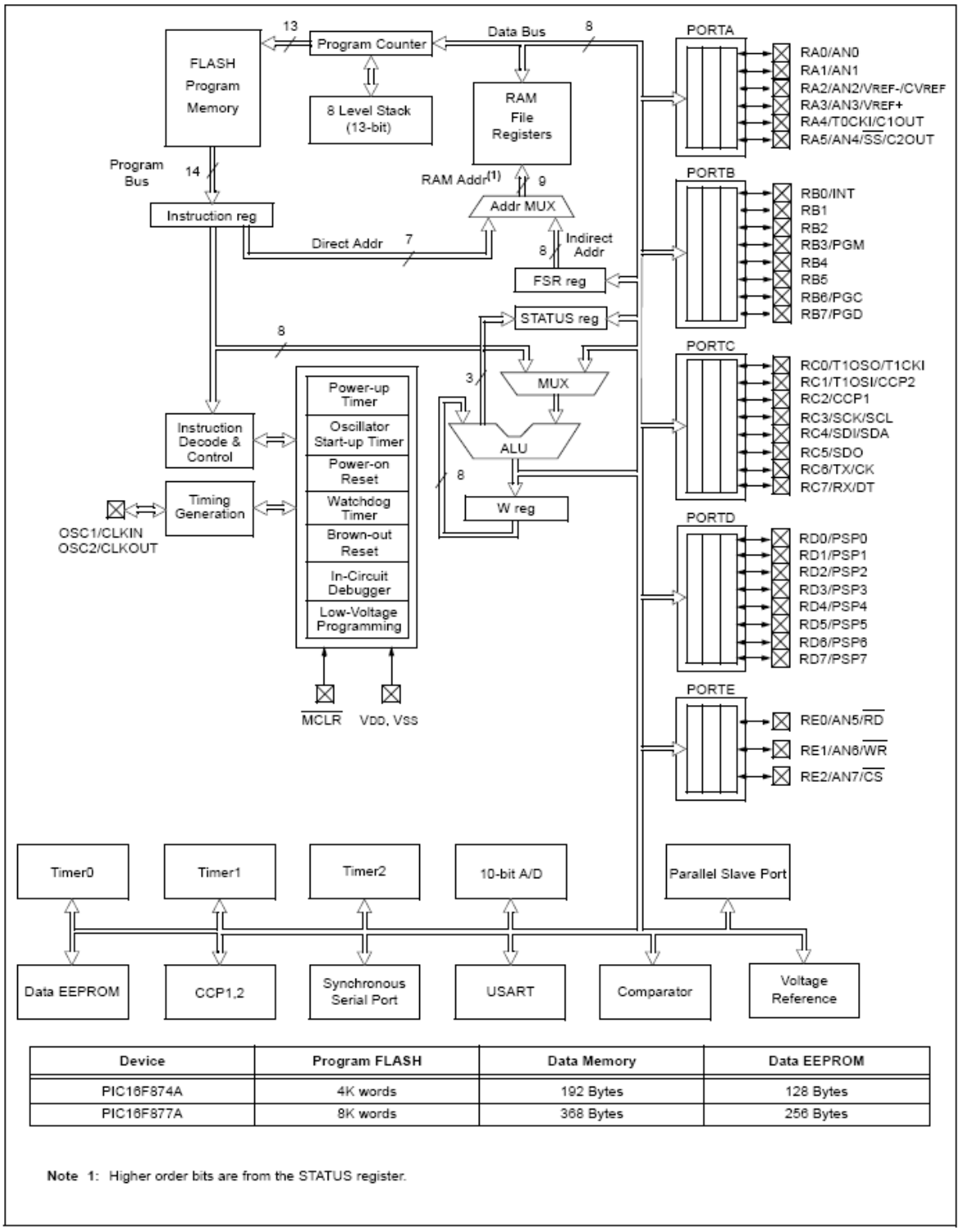
UART.- (Universal Asynchronous Receiver Transmitter). Transmisor receptor asíncrono universal, permite controlar la interfase RS-232.

ANEXOS

ANEXO A

DESCRIPCIÓN DE LOS CIRCUITOS INTEGRADOS

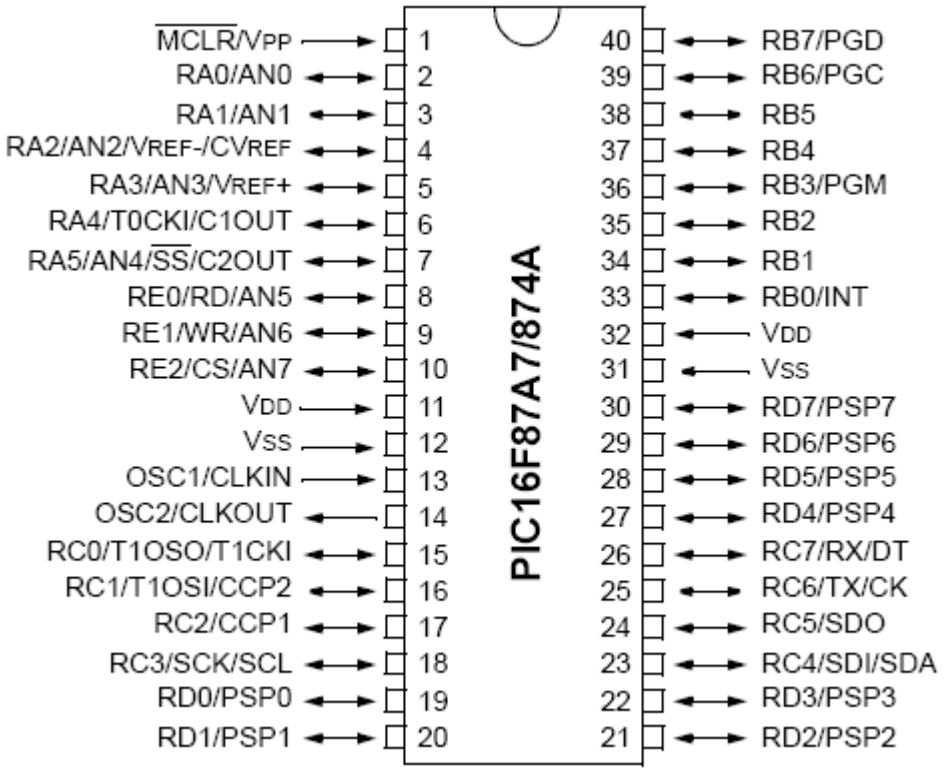
ANEXO A.1



Device	Program FLASH	Data Memory	Data EEPROM
PIC16F874A	4K words	192 Bytes	128 Bytes
PIC16F877A	8K words	368 Bytes	256 Bytes

Note 1: Higher order bits are from the STATUS register.

a.1.1- Esquema general interno del microcontrolador 16F877A



a.1.2.- Disposición de pines del PIC 16F877A.

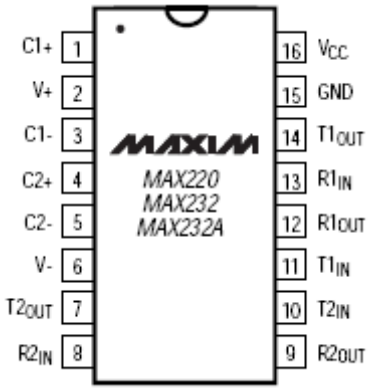
File Address	Bank 0	File Address	Bank 1	File Address	Bank 2	File Address	Bank 3
00h	Indirect addr.(*)	80h	Indirect addr.(*)	100h	Indirect addr.(*)	180h	Indirect addr.(*)
01h	TMR0	81h	OPTION_REG	101h	TMR0	181h	OPTION_REG
02h	PCL	82h	PCL	102h	PCL	182h	PCL
03h	STATUS	83h	STATUS	103h	STATUS	183h	STATUS
04h	FSR	84h	FSR	104h	FSR	184h	FSR
05h	PORTA	85h	TRISA	105h		185h	
06h	PORTB	86h	TRISB	106h	PORTB	186h	TRISB
07h	PORTC	87h	TRISC	107h		187h	
08h	PORTD(1)	88h	TRISD(1)	108h		188h	
09h	PORTE(1)	89h	TRISE(1)	109h		189h	
0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah	PCLATH
0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh	INTCON
0Ch	PIR1	8Ch	PIE1	10Ch	EEDATA	18Ch	EECON1
0Dh	PIR2	8Dh	PIE2	10Dh	EEADR	18Dh	EECON2
0Eh	TMR1L	8Eh	PCON	10Eh	EEDATH	18Eh	Reserved(2)
0Fh	TMR1H	8Fh		10Fh	EEADRH	18Fh	Reserved(2)
10h	T1CON	90h		110h	General Purpose Register 16 Bytes	190h	General Purpose Register 16 Bytes
11h	TMR2	91h	SSPCON2	111h		191h	
12h	T2CON	92h	PR2	112h		192h	
13h	SSPBUF	93h	SSPADD	113h		193h	
14h	SSPCON	94h	SSPSTAT	114h		194h	
15h	CCPR1L	95h		115h		195h	
16h	CCPR1H	96h		116h		196h	
17h	CCP1CON	97h		117h		197h	
18h	RCSTA	98h	TXSTA	118h		198h	
19h	TXREG	99h	SPBRG	119h		199h	
1Ah	RCREG	9Ah		11Ah		19Ah	
1Bh	CCPR2L	9Bh		11Bh		19Bh	
1Ch	CCPR2H	9Ch	CMCON	11Ch		19Ch	
1Dh	CCP2CON	9Dh	CVRCON	11Dh		19Dh	
1Eh	ADRESH	9Eh	ADRESL	11Eh		19Eh	
1Fh	ADCON0	9Fh	ADCON1	11Fh		19Fh	
20h	General Purpose Register 96 Bytes	A0h	General Purpose Register 80 Bytes	120h	General Purpose Register 80 Bytes	1A0h	General Purpose Register 80 Bytes
		EFh		16Fh		1EFh	
		F0h	accesses 70h-7Fh	170h	accesses 70h-7Fh	1F0h	accesses 70h - 7Fh
7Fh		FFh		17Fh		1FFh	

Unimplemented data memory locations, read as '0'.
* Not a physical register.

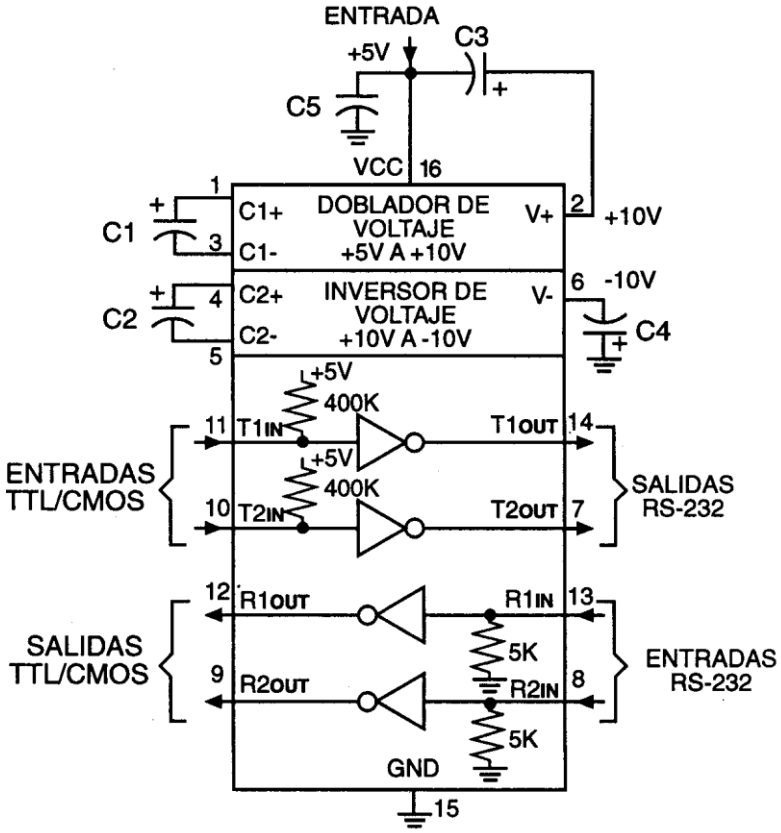
Note 1: These registers are not implemented on the PIC16F876A.
2: These registers are reserved, maintain these registers clear.

a.1.3.- Organización interna de datos, en la que se encuentran los registros del PIC 16F877A, además indica el banco en el que se encuentran estos registros.

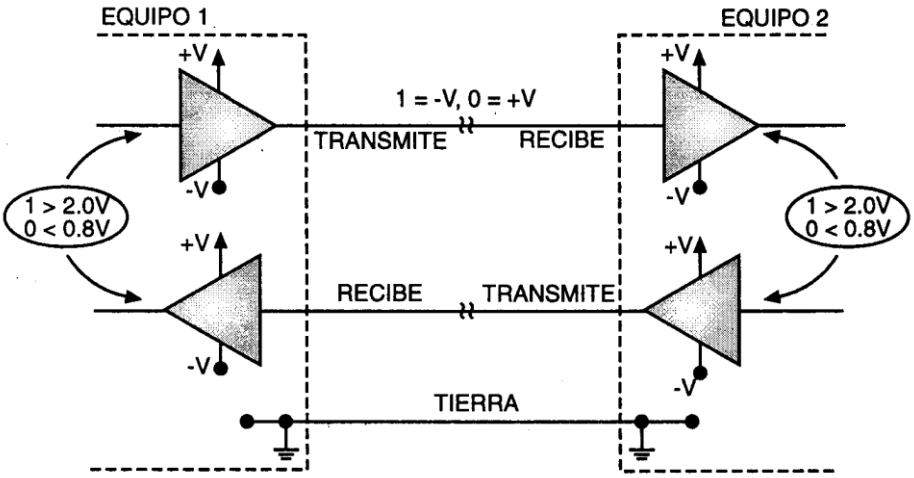
ANEXO A.2



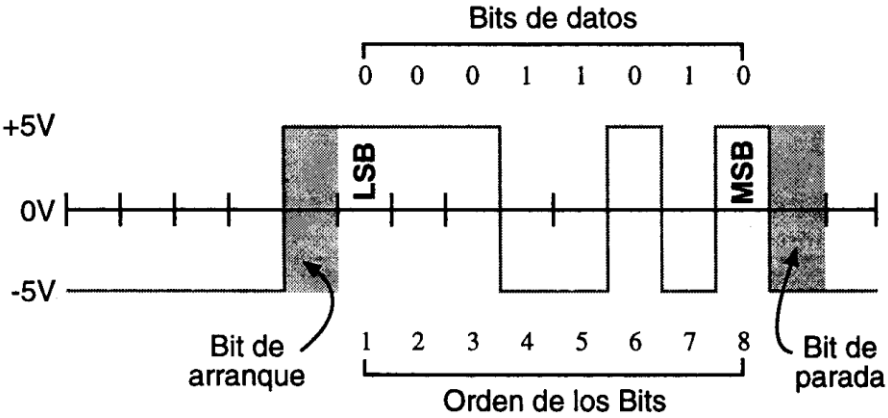
a.2.1- Diagrama de pines del MAX232A



a.2.2- Estructura interna del MAX232A

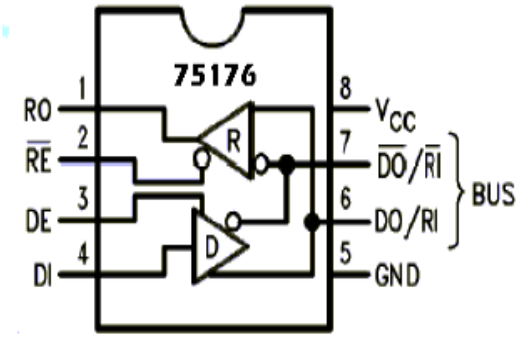


a.2.3- Representación de la interfase RS-232

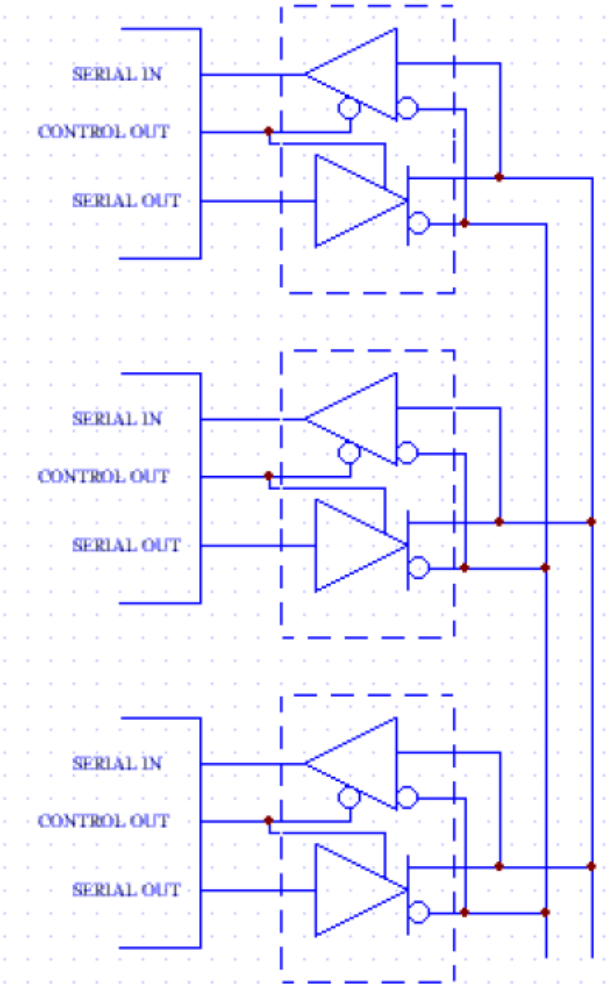


a.2.4- Señal presente sobre una línea RS-232

ANEXO A.3



a.3.1- Diagrama de pines del SN 75176

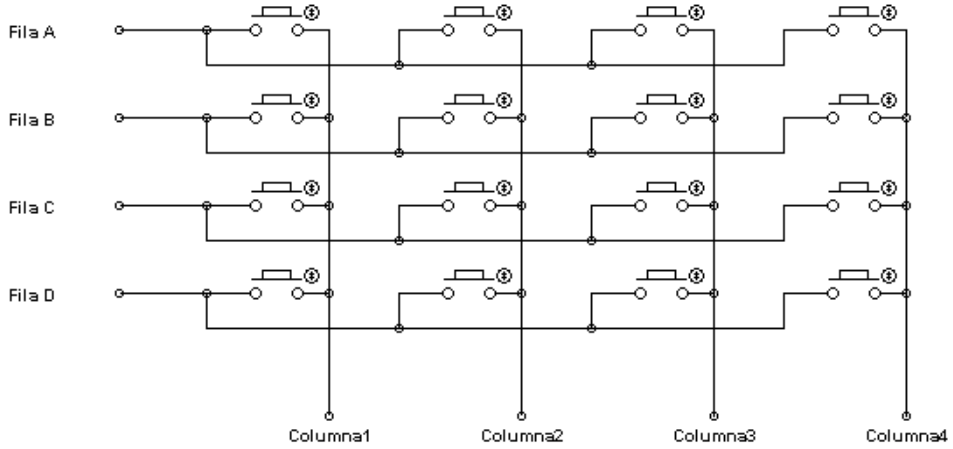


a.3.2 Esquema de una comunicación RS-485 en Modo Half Duplex.

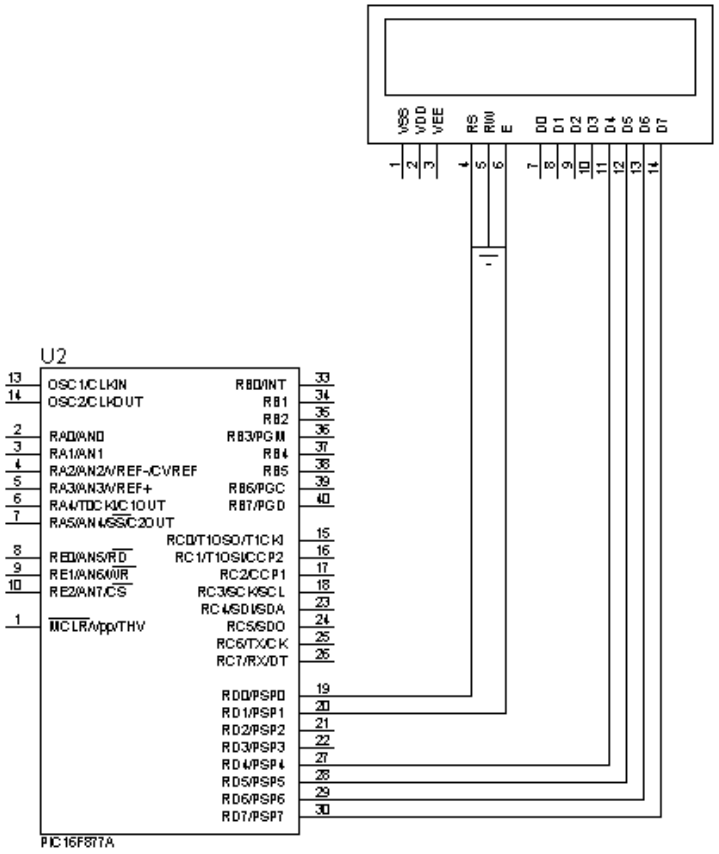
ANEXO B

DIAGRAMAS DE CONEXIONES

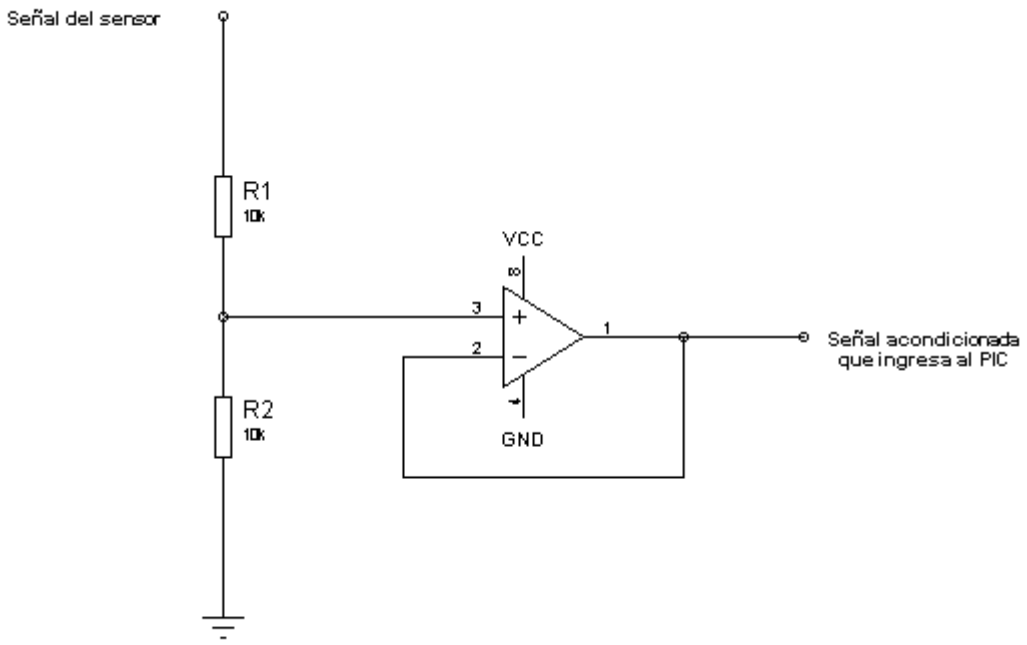
ANEXO B.1



b.1.1- Diagrama de la estructura interna del teclado 4x4.



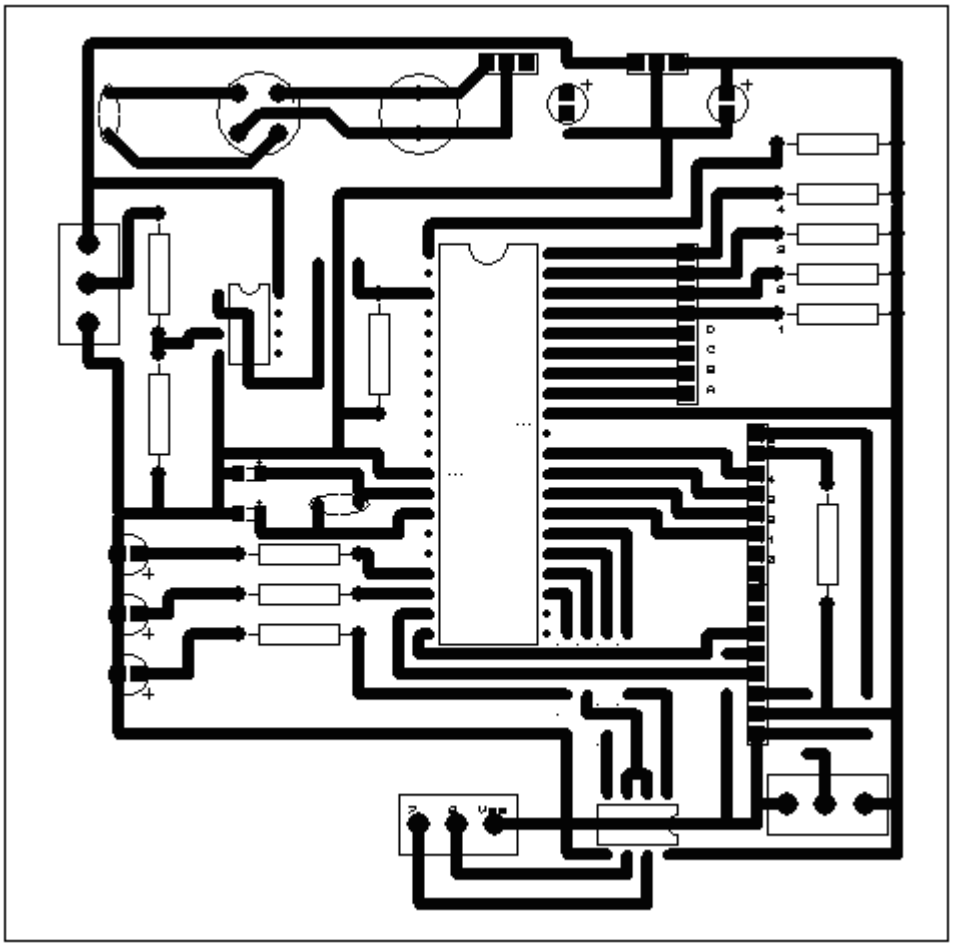
b.1.2.Diagrama de conexiones del lcd.



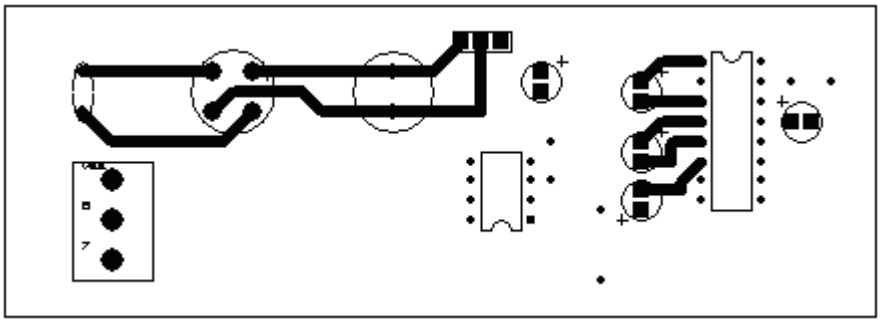
b.1.3- Diagrama circuital del acondicionamiento de señal del sensor.

ANEXO C

CIRCUITOS IMPRESOS



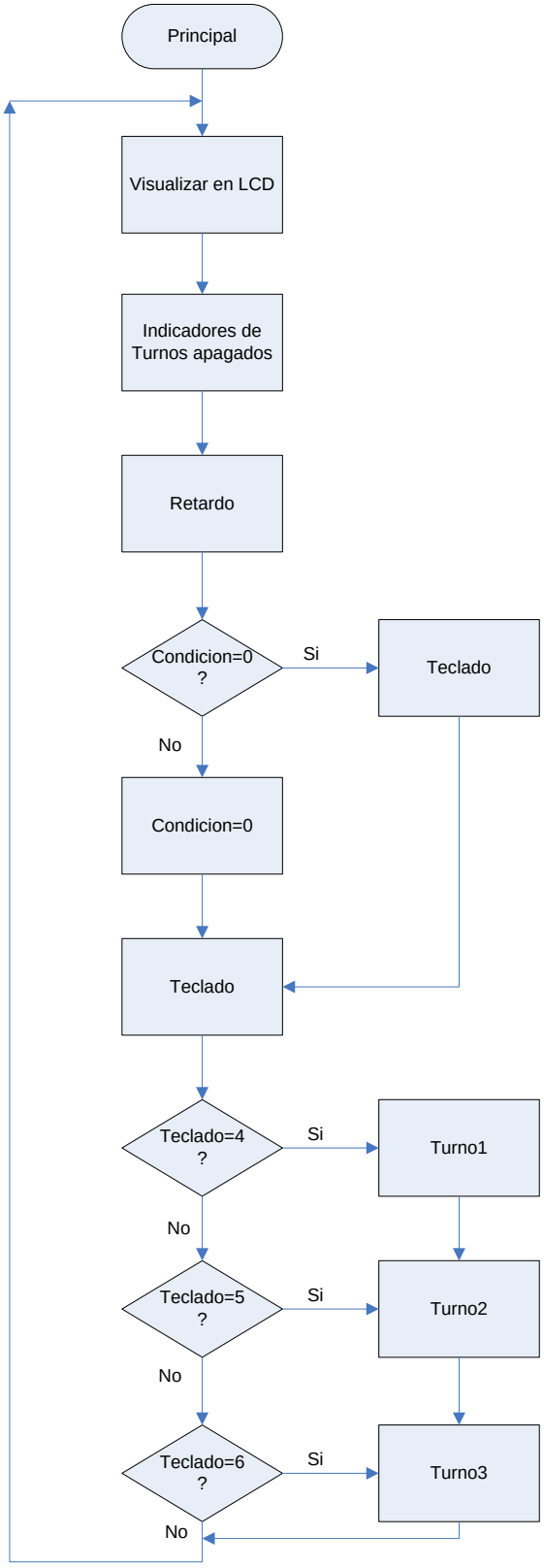
c.1.- Circuito impreso de la tarjeta para cada contador.



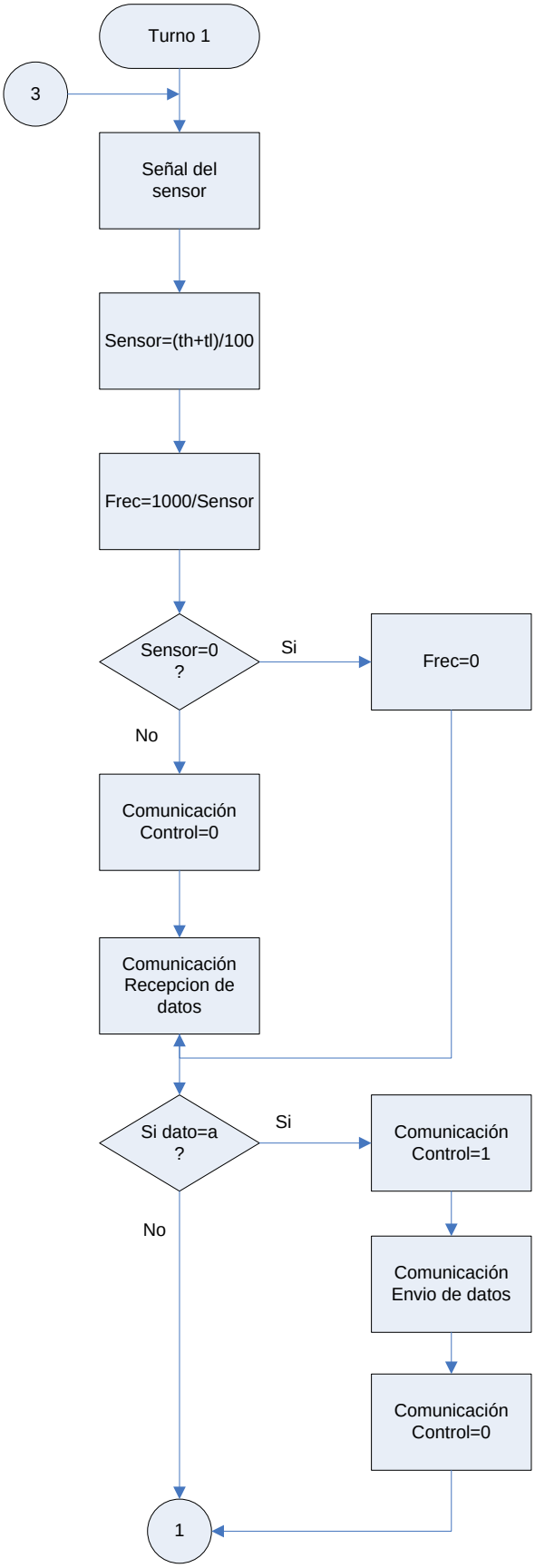
c.2.- Circuito impreso de la tarjeta de adquisición de datos.

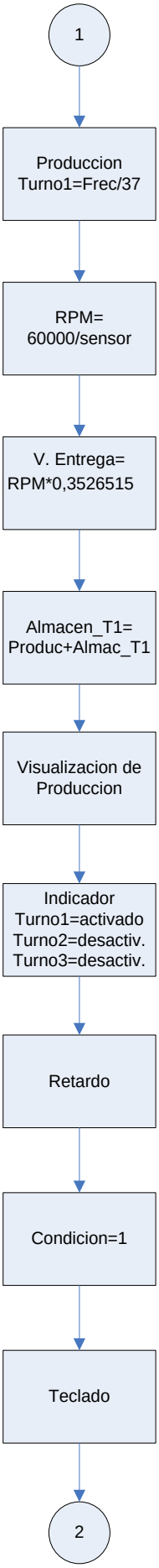
ANEXO D

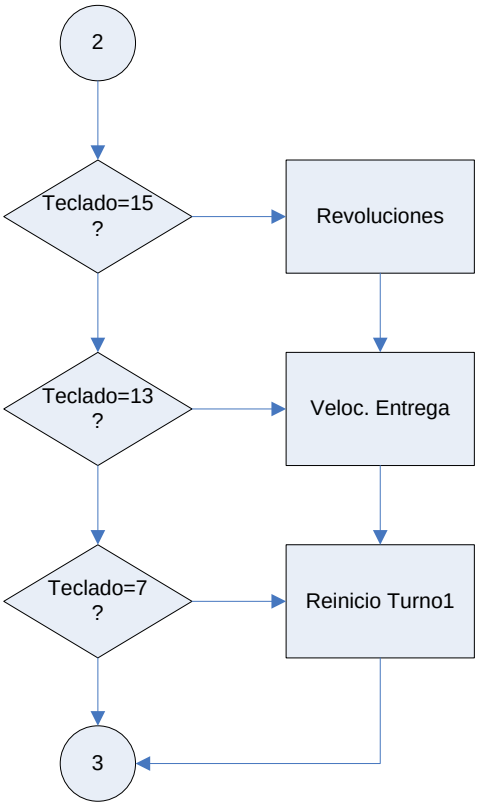
DIAGRAMAS DE FLUJO



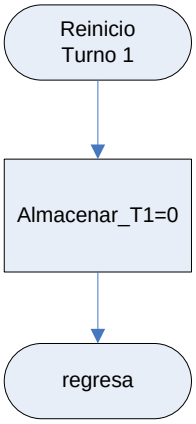
d.1.- Diagrama de flujo del Programa Principal.


Turno 1
3
Señal del sensor
Sensor=(th+tl)/100
Frec=1000/Sensor
Sensor=0 ?
Si
Frec=0
No
Comunicación Control=0
Comunicación Recepcion de datos
Si dato=a ?
Si
Comunicación Control=1
No
Comunicación Envio de datos
Comunicación Control=0
1

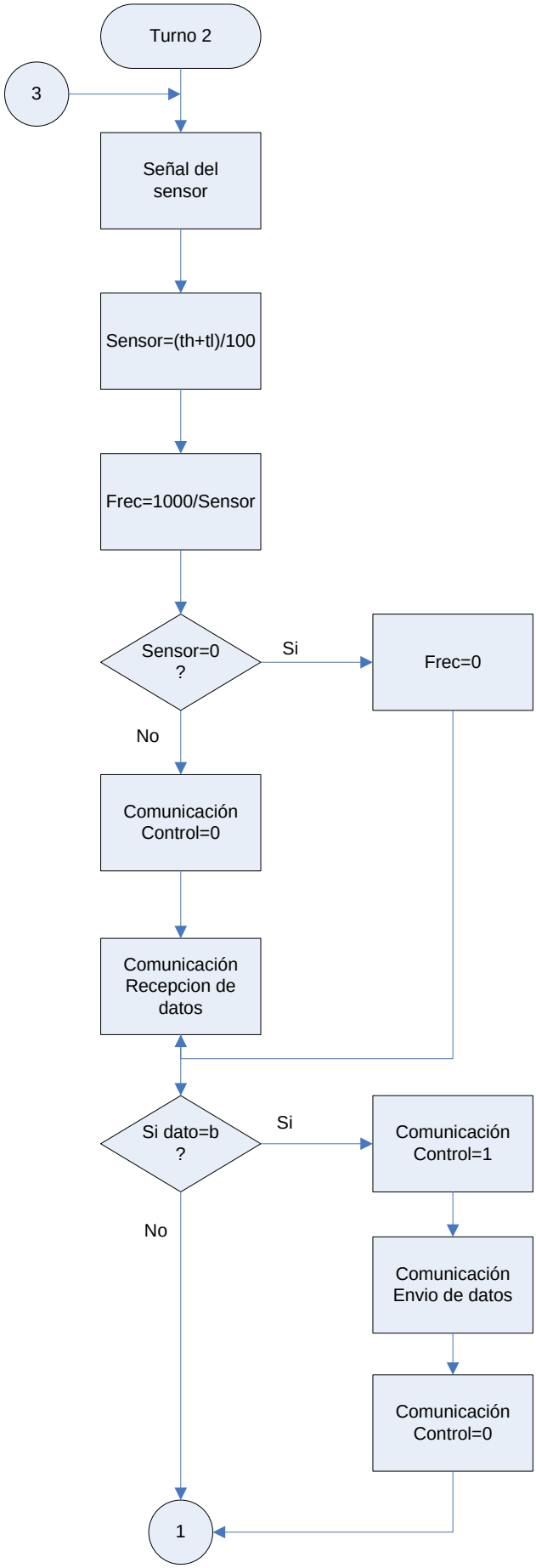

1
Produccion
Turno1=Frec/37
RPM=
60000/sensor
V. Entrega=
RPM*0,3526515
Almacen_T1=
Produc+Almac_T1
Visualizacion de
Produccion
Indicador
Turno1=activado
Turno2=desactiv.
Turno3=desactiv.
Retardo
Condicion=1
Teclado
2

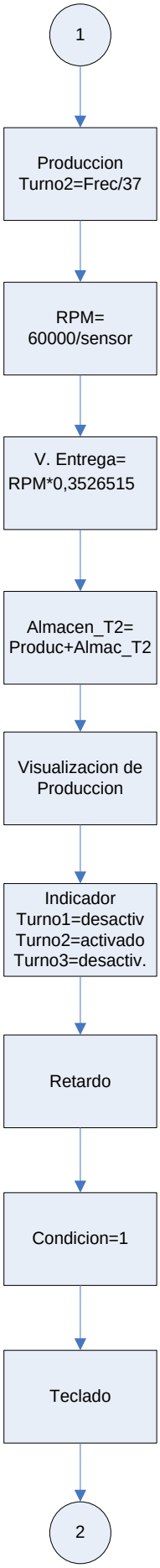


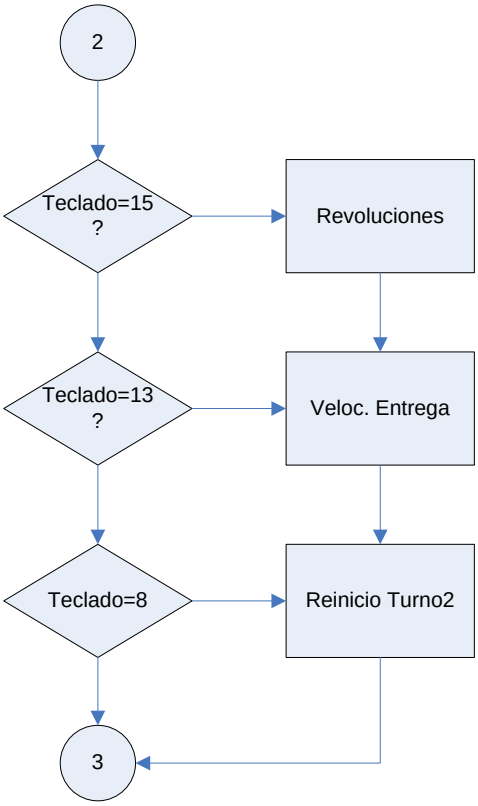
d.2.- Diagrama de flujo de la subrutina Turno 1.



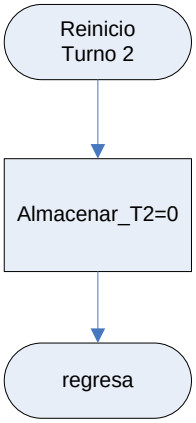
d.3.- Diagrama de flujo de la subrutina Reinicio Turno 1.


Turno 2
3
Señal del sensor
Sensor=(th+tl)/100
Frec=1000/Sensor
Sensor=0 ?
Si
Frec=0
No
Comunicación Control=0
Comunicación Recepcion de datos
Si dato=b ?
Si
Comunicación Control=1
No
Comunicación Envio de datos
Comunicación Control=0
1

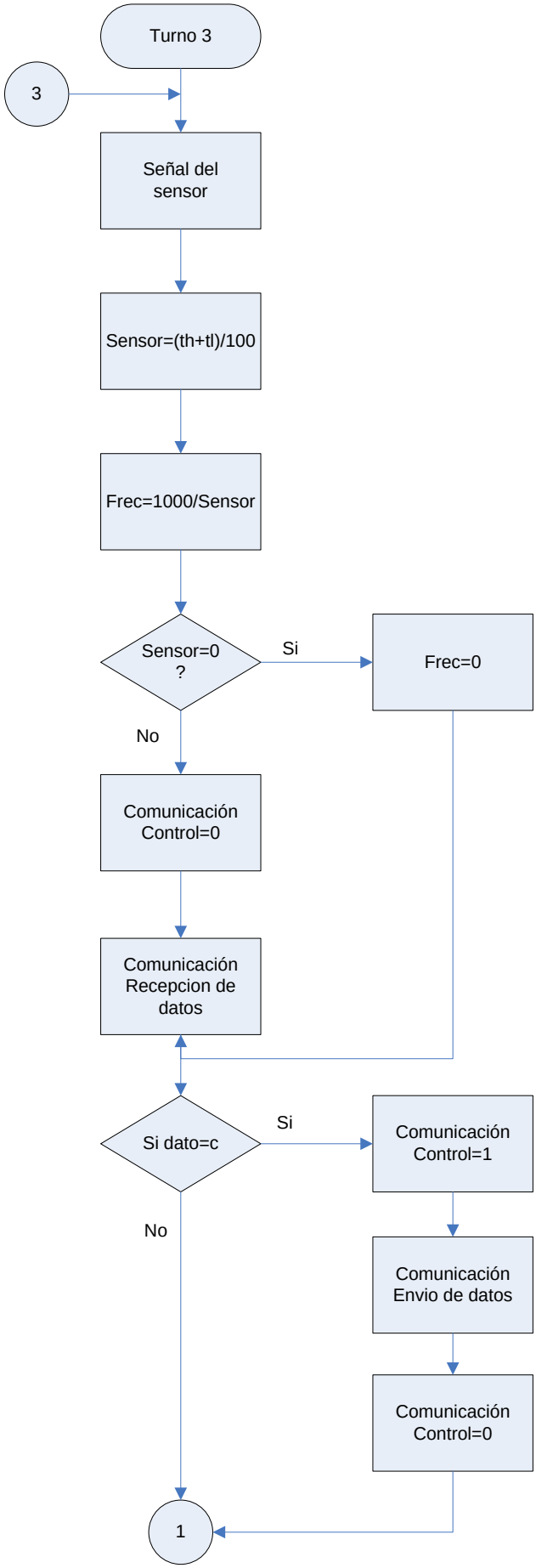

1
Produccion
Turno2=Frec/37
RPM=
60000/sensor
V. Entrega=
RPM*0,3526515
Almacen_T2=
Produc+Almac_T2
Visualizacion de
Produccion
Indicador
Turno1=desactiv
Turno2=activado
Turno3=desactiv.
Retardo
Condicion=1
Teclado
2

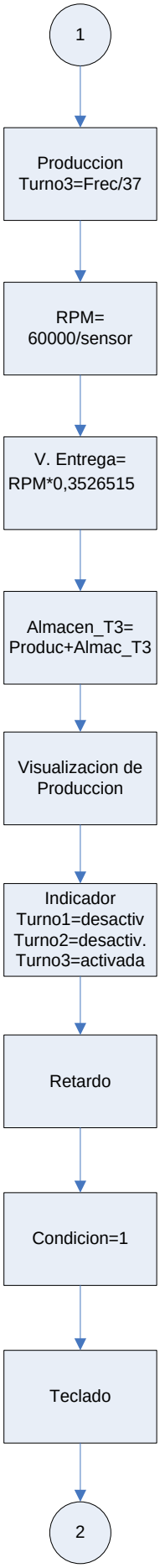


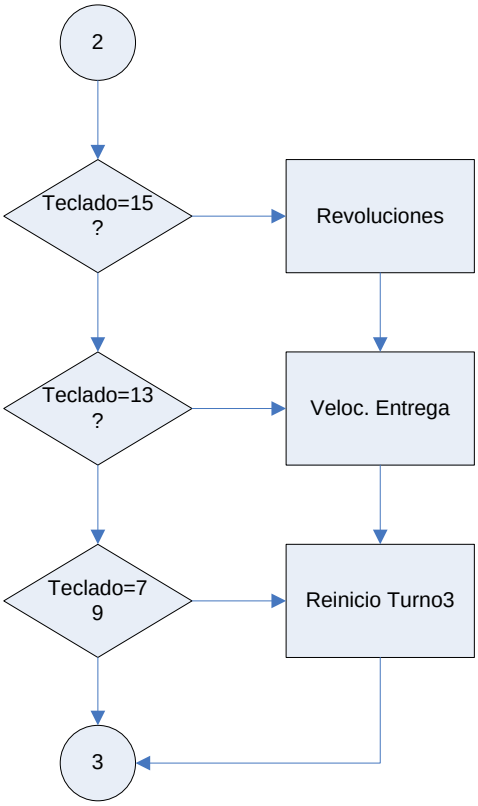
d.4.- Diagrama de flujo de la subrutina Turno 2.



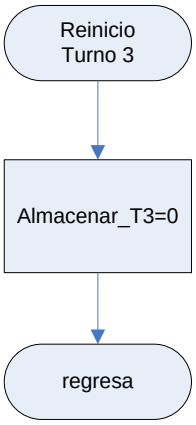
d.5.- Diagrama de flujo de la subrutina Reinicio Turno 2.


Turno 3
3
Señal del sensor
Sensor=(th+tl)/100
Frec=1000/Sensor
Sensor=0 ?
Si
Frec=0
No
Comunicación Control=0
Comunicación Recepcion de datos
Si dato=c
Si
Comunicación Control=1
No
Comunicación Envio de datos
Comunicación Control=0
1

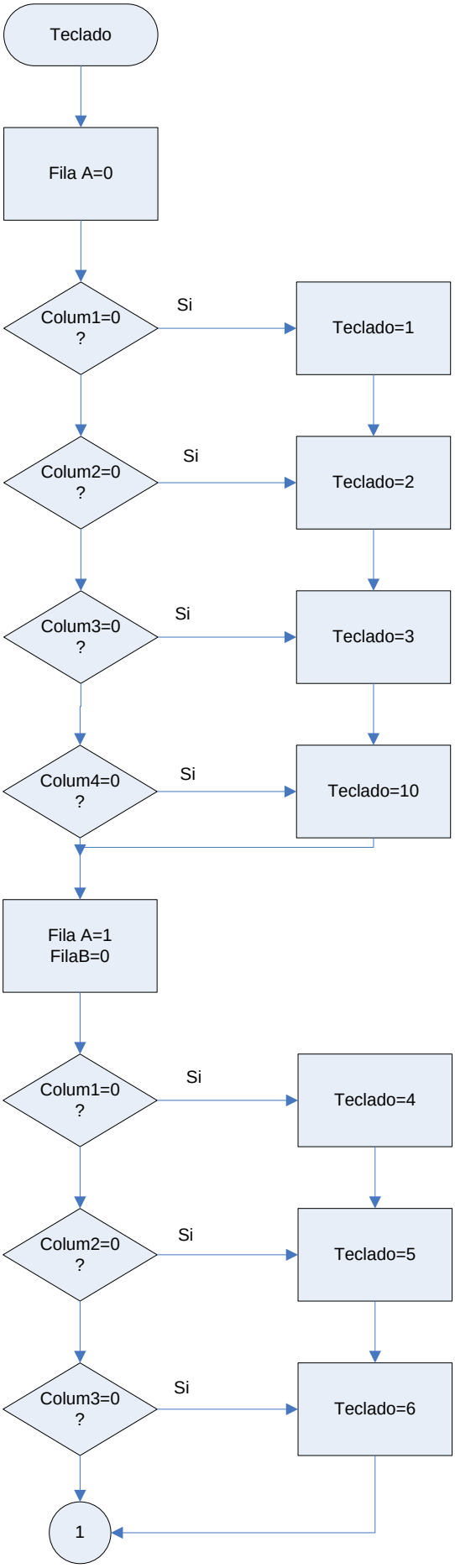

1
Produccion
Turno3=Frec/37
RPM=
60000/sensor
V. Entrega=
RPM*0,3526515
Almacen_T3=
Produc+Almac_T3
Visualizacion de
Produccion
Indicador
Turno1=desactiv
Turno2=desactiv.
Turno3=activada
Retardo
Condicion=1
Teclado
2

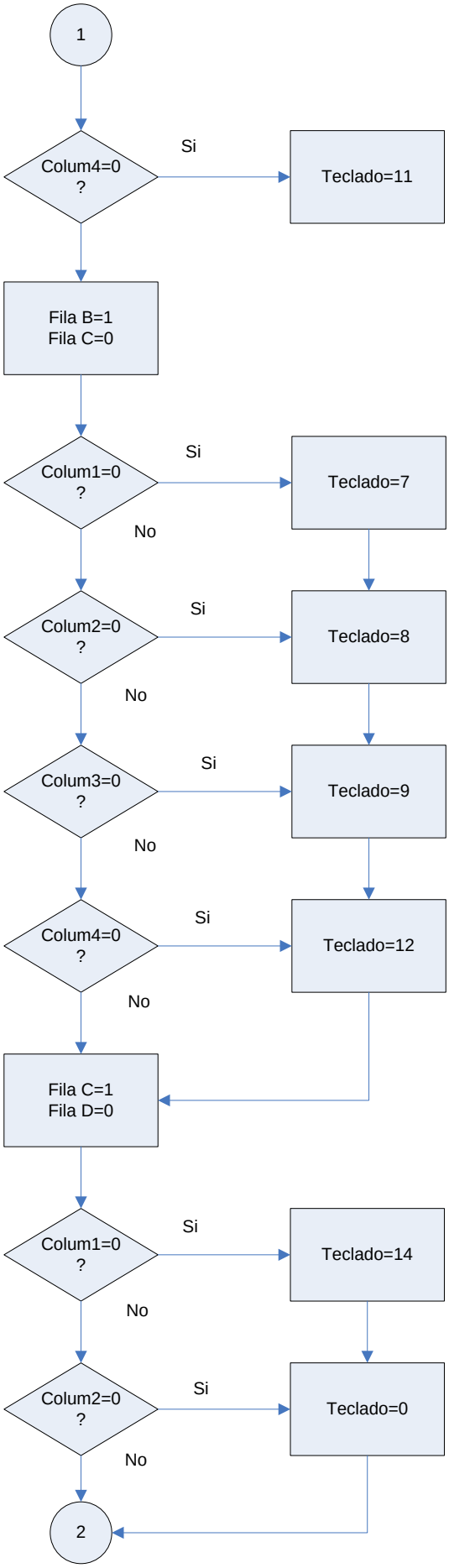


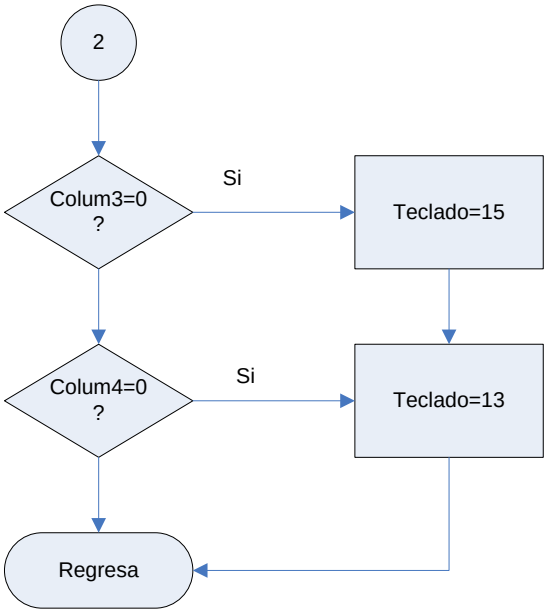
d.6.- Diagrama de flujo de la subrutina Turno 3.



d.7.- Diagrama de flujo de la subrutina Reinicio Turno 3.


Teclado
Fila A=0
Colum1=0
?
Si
Teclado=1
Colum2=0
?
Si
Teclado=2
Colum3=0
?
Si
Teclado=3
Colum4=0
?
Si
Teclado=10
Fila A=1
FilaB=0
Colum1=0
?
Si
Teclado=4
Colum2=0
?
Si
Teclado=5
Colum3=0
?
Si
Teclado=6
1

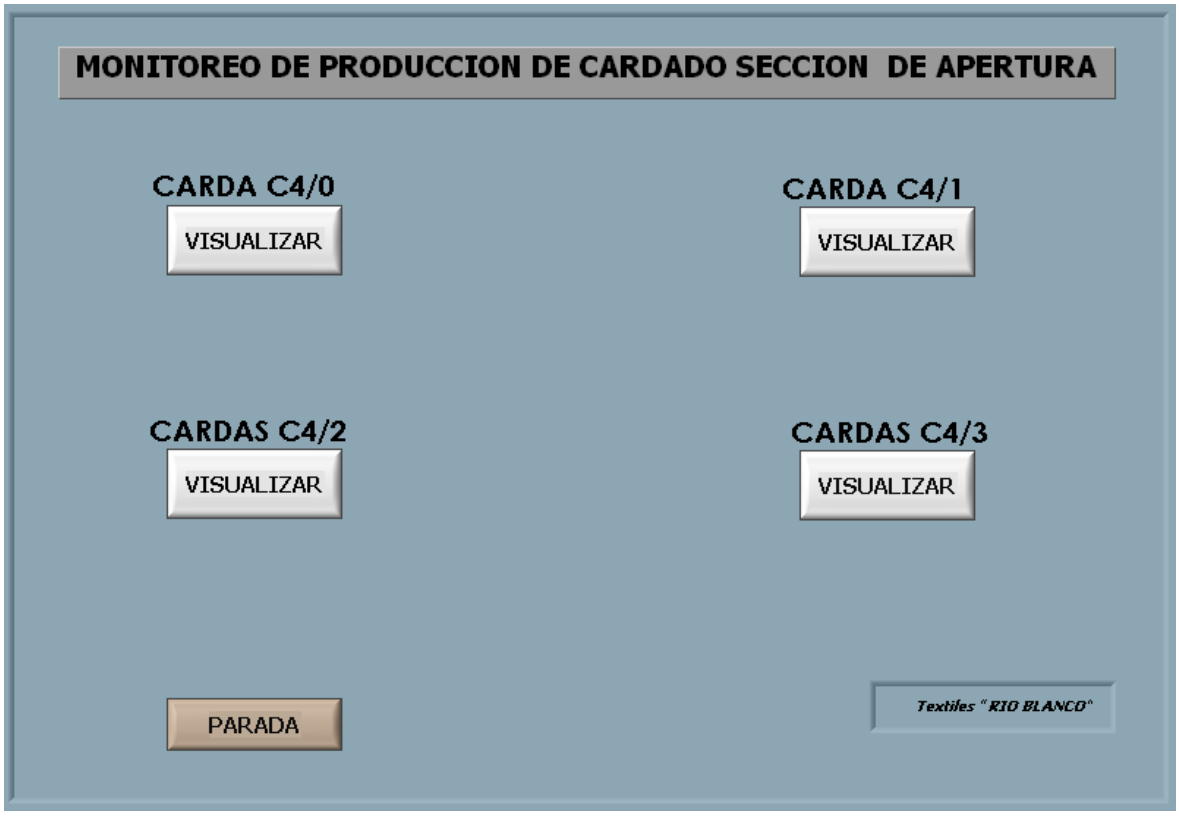

1
Colum4=0
?
Si
Teclado=11
Fila B=1
Fila C=0
Colum1=0
?
Si
Teclado=7
No
Colum2=0
?
Si
Teclado=8
No
Colum3=0
?
Si
Teclado=9
No
Colum4=0
?
Si
Teclado=12
No
Fila C=1
Fila D=0
Colum1=0
?
Si
Teclado=14
No
Colum2=0
?
Si
Teclado=0
No
2



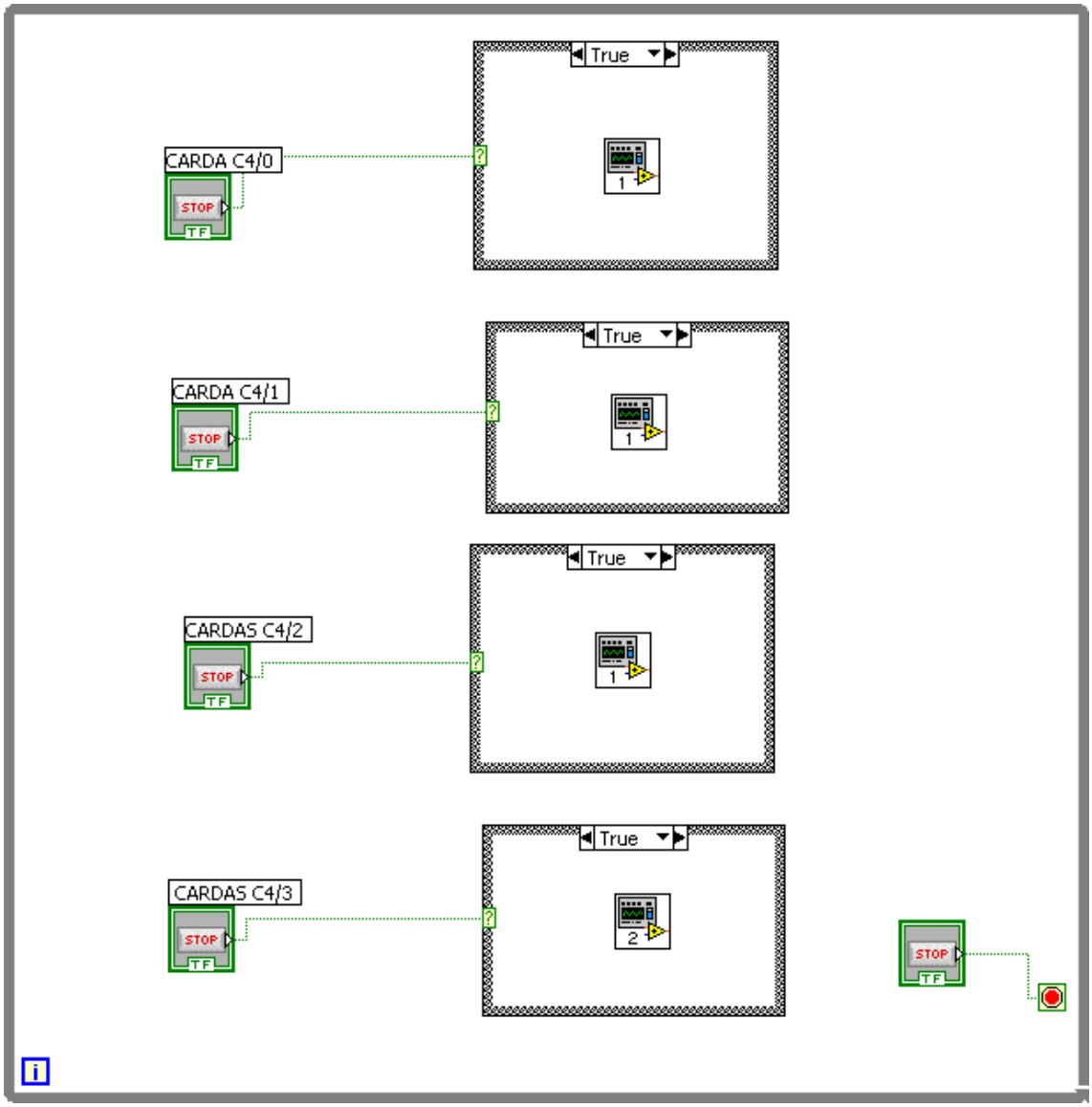
d.8.- Diagrama de flujo de la subrutina Teclado.

ANEXO E

PROGRAMACION REALIZADA EN LABVIEW

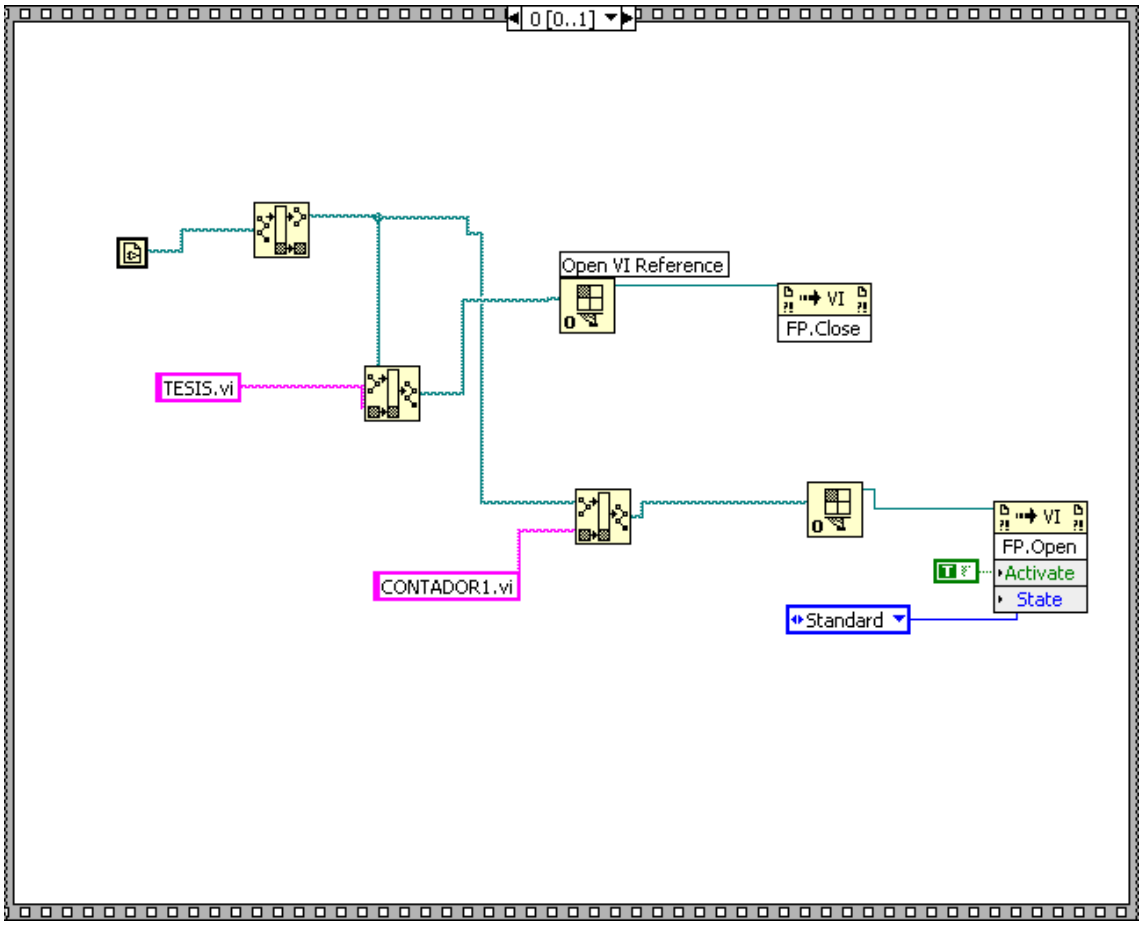


e.1. Panel Frontal del Programa Principal, realizado en LabVIEW 7.1 Profesional.

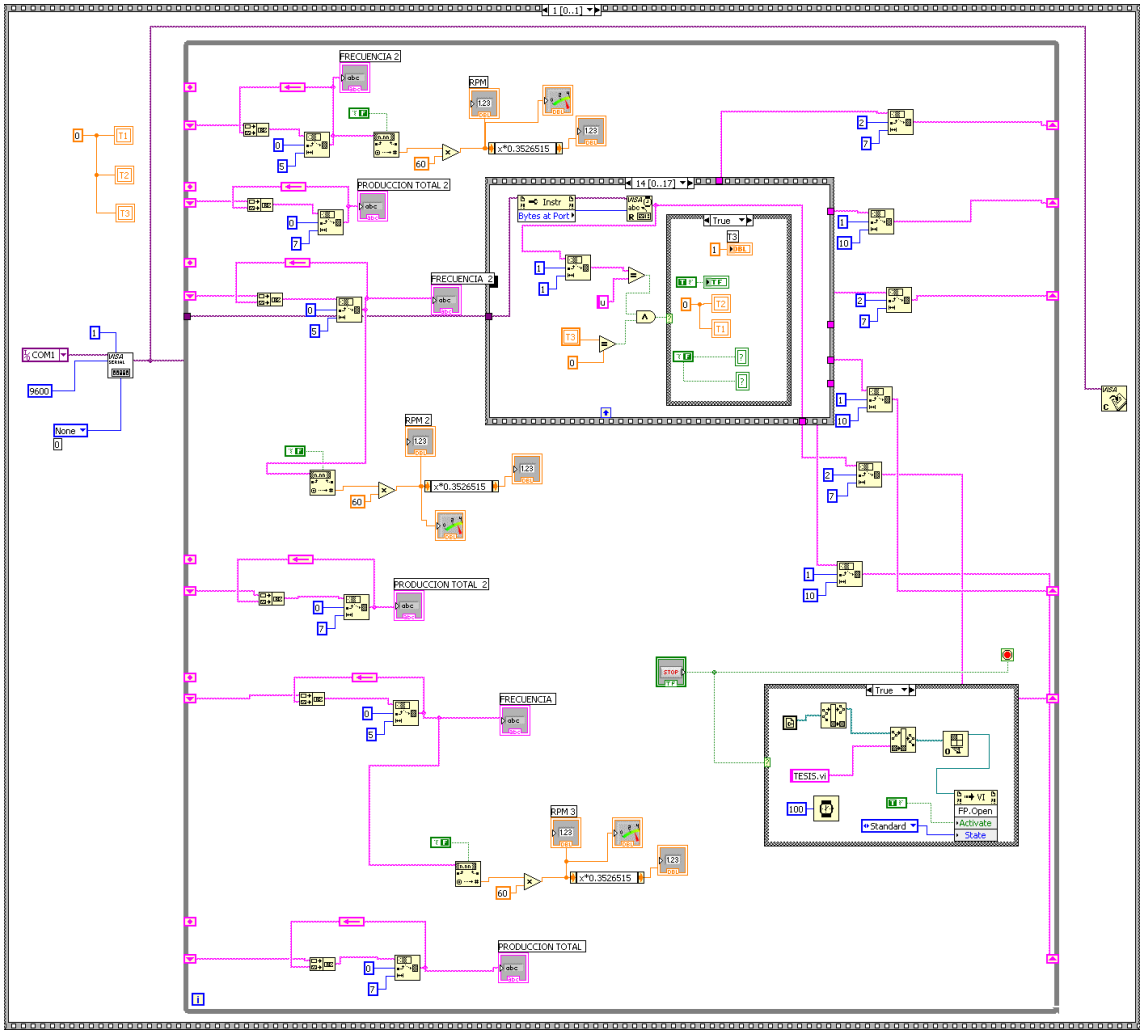


e.2. Diagrama de bloques del Programa Principal, realizado en LabVIEW 7.1 Profesional.

e.3. Panel frontal de la Programación de cada contador, realizado en LabVIEW 7.1 Profesional.



e.4 Diagrama de bloques de la Programación de cada contador, en secuencia 0, realizado en LabVIEW 7.1 Profesional.



e.5 Diagrama de bloques de la Programación de cada contador, en secuencia 1, realizado en LabVIEW 7.1 Profesional.

ANEXO F

MANUAL DE USUARIO

Manual de Usuario:

1.- Inicialmente en la visualización del contador se observa un mensaje como se indica en la figura f.1:

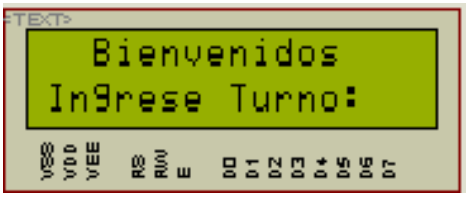


Fig.f.1. *Visualización inicial del contador*

2.- Para empezar la producción se selecciona un turno; ello se realiza mediante teclado pulsando el botón turno 1. Como indica la figura f.2.

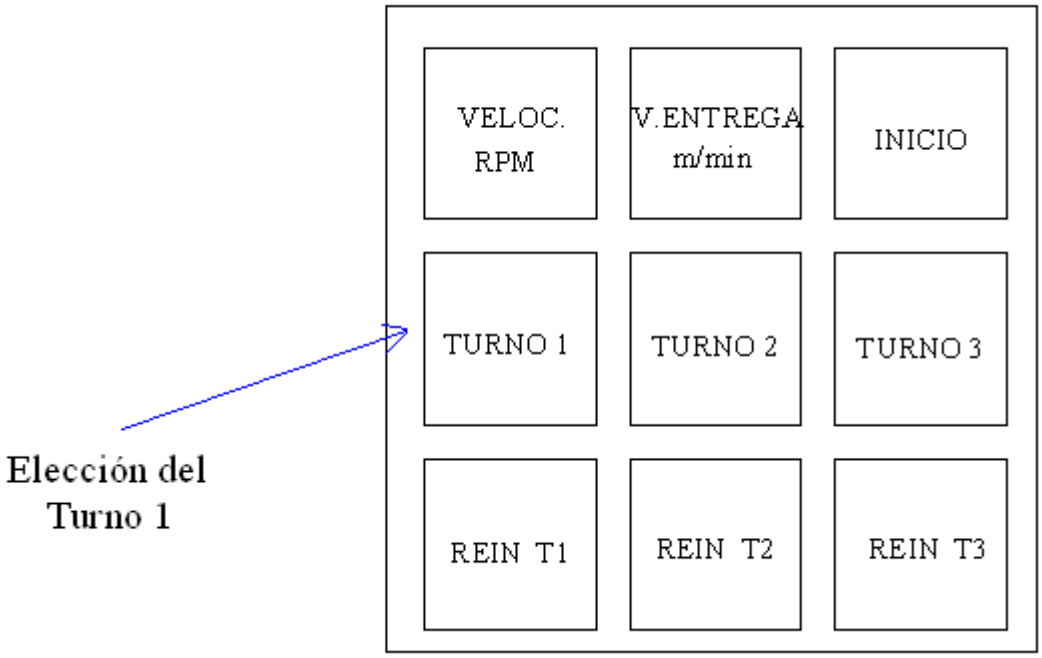


Fig.f.2. *Elección del turno a trabajar*

3.- Realizado el paso anterior se presenta la visualización de la producción (metros) en tiempo real. Como se ilustra en la figura f.3.

Fig.f.3. *Visualización de la producción del contador*

4.- Este contador tiene la capacidad de presentar otros parámetros como son:

a. Velocidad (RPM), para ingresar a esta visualización, en el teclado presionar la tecla que indica velocidad, con ello en el LCD se visualizará la velocidad en RPM como se observa en la figura f.4.b

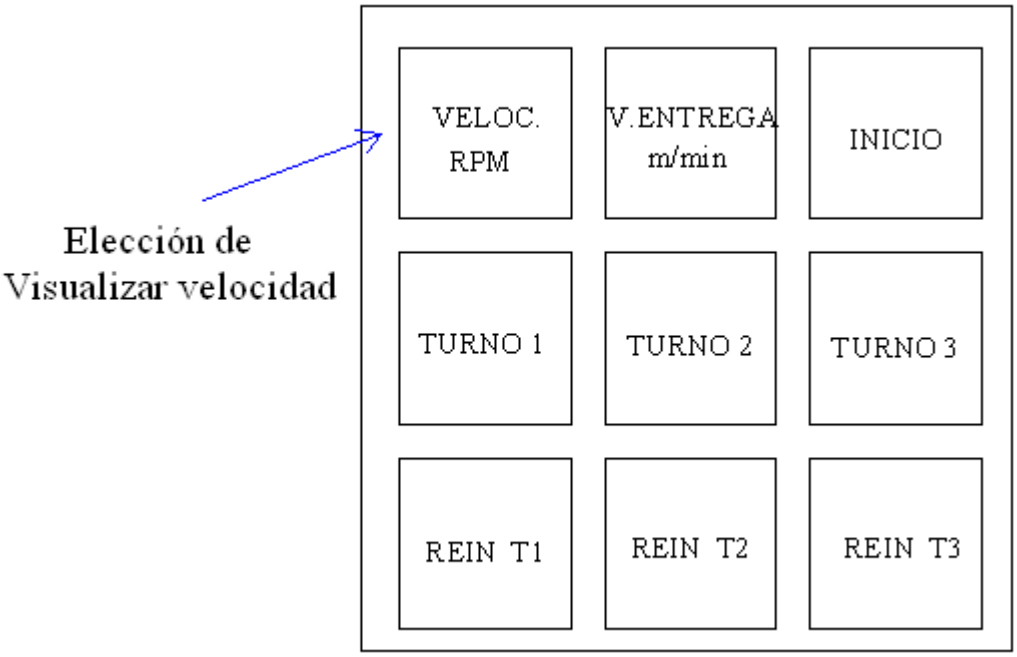


Fig.f.4.a *Selección del botón para visualizar velocidad*

Fig.f.4.b *Visualización de la velocidad del contador*

b. Velocidad de entrega (m/min), para ingresar a esta visualización, en el teclado presionar la tecla que indica velocidad de entrega, con ello en el LCD se visualizara la velocidad de entrega en (m/min) como se observa en la figura f.5.b

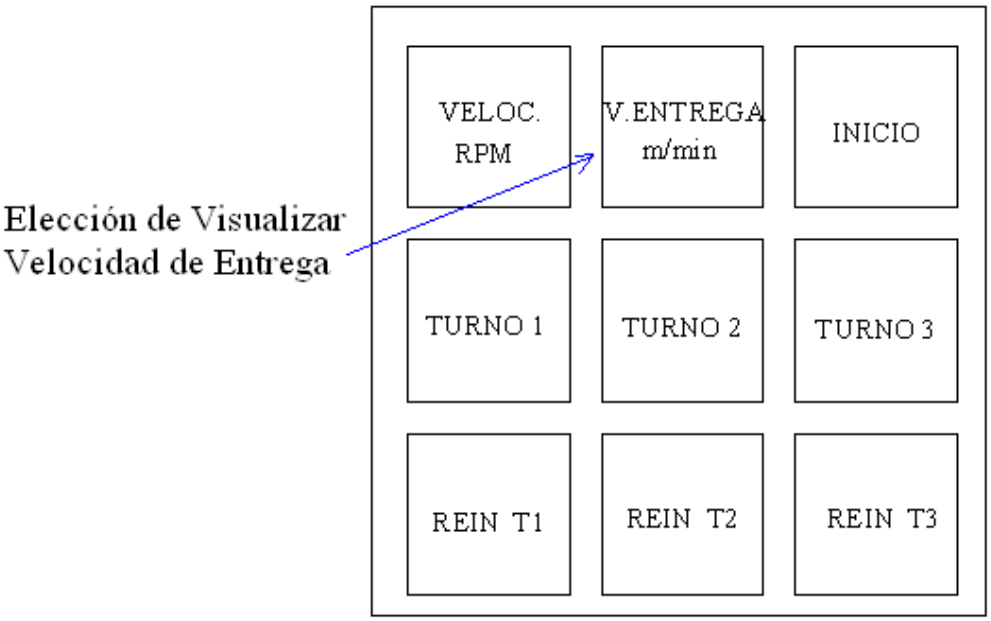


Fig.f.5.a *Selección del botón para visualizar velocidad de entrega*

Fig.f.5.b *Visualización de la velocidad de entrega*

5.- Al entrar en funcionamiento cada turno cuenta con un indicador que se enciende durante el funcionamiento del mismo, como se observa en la figura f.6.

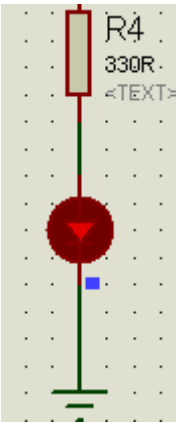


Fig.f.6. *Indicador de funcionamiento de cada turno*

6.- Cada turno cuenta con la opción de reiniciarlo, cuando termine con su período de conteo. Esto se lo realiza pulsando el botón de reiniciar turno de acuerdo al turno que se encuentre en funcionamiento. A continuación se observa como reiniciar el turno 1.

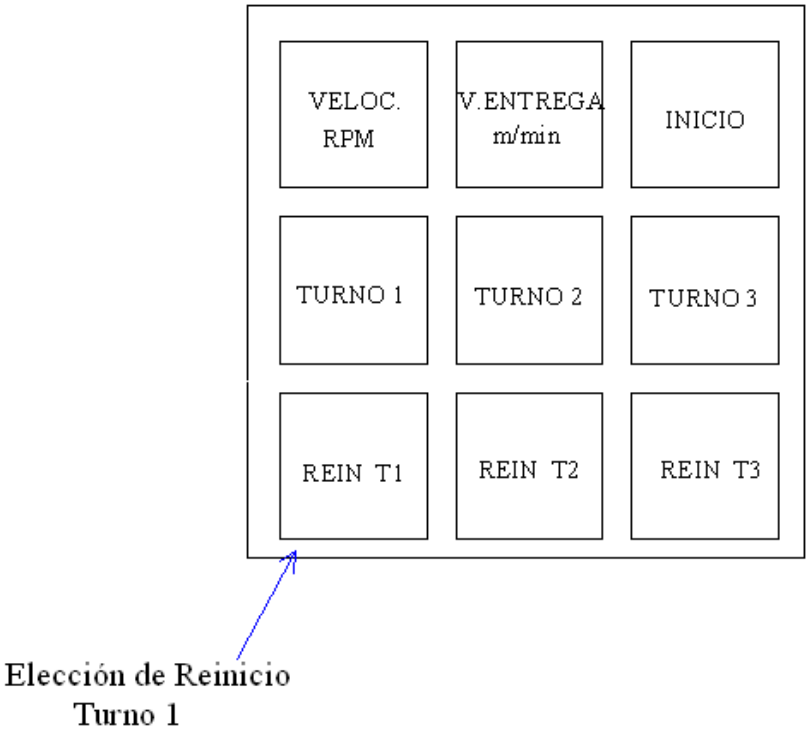


Fig.f.7.a *Selección del botón para reiniciar el turno 1*

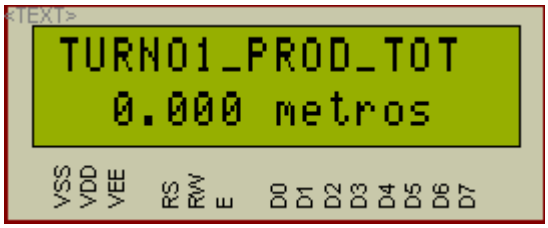


Fig.f.b *Visualización del Turno 1 reiniciado*