

## **Diseño de una tarjeta de adquisición de datos para la supervisión y control de procesos y desarrollo de su interfaz utilizando LabVIEW 7.1**

**Director de Tesis:** Ing Alberto Larco, [dlarco@fiec.espol.edu.ec](mailto:dlarco@fiec.espol.edu.ec)

**Integrantes:** Juan Carlos García, [jcgarcia@fiec.espol.edu.ec](mailto:jcgarcia@fiec.espol.edu.ec)  
José Luís Parrales, [jparral@fiec.espol.edu.ec](mailto:jparral@fiec.espol.edu.ec)  
Alfonso Mazacón Baño, [amazacon@fiec.espol.edu.ec](mailto:amazacon@fiec.espol.edu.ec)

Facultad de Ingeniería en Electricidad y Computación (FIEC)  
ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL (ESPOL)  
Campus Gustavo Galindo, vía Perimetral Km 30.5, Guayaquil, Ecuador

### **Resumen**

*En este documento se describe el funcionamiento de la tarjeta de adquisición de datos. La misma que posee 4 entradas digitales y 4 entradas analógicas de 0 a 5 Voltios, las mismas que sirven para monitorear el estado de las variables existentes en un proceso o máquina. Además esta tarjeta posee 4 salidas digitales de rele de 10 Amperios 240 Voltios y una salida analógica con un rango de de 0 a 10 Voltios con una precisión de 40 mV. El elemento de control de este módulo es un microcontrolador PIC 16F877, y el software utilizado para el desarrollo de la interfaz gráfica es LabVIEW 7.1 de National Instrument.*

**Palabras Claves:** *Microcontrolador, Comunicación serial, Computadora Personal.*

### **Abstract**

*This article describes the basic operation of a module designed for serial data acquisition, data is coming from four digital and five analogue inputs, implemented to simulate changes in variables of a process. This module have eight relay outputs and one output for pulse width modulation (PWM), also this module has a switch for data transmission speed selection between the microcontroller and a personal computer (PC). The main control element in this module is a microcontroller PIC 16F877, and for developing its graphic interface we have used software as LabVIEW 7.1*

# 1. Introducción

Actualmente en el mercado existen varias opciones al momento de elegir que hardware o software a utilizar para automatizar un proceso, en este ámbito el uso de Controladores Lógicos Programables (PLCs) y sistemas para la Supervisión, Control y Adquisición de Datos (SCADAs), es muy generalizado, debido a su simplicidad de manejo y su capacidad de expansión, sin embargo muchas soluciones resultan muy costosas para nuestro medio, por lo que es necesario buscar otras alternativas, que aunque involucran más tiempo de desarrollo permiten ofrecer una solución satisfactoria a numerosos problemas de control e instrumentación.

Una de esas alternativas es la utilización de tarjetas o módulos para la adquisición de datos y el control de variables de un proceso, que en la actualidad son cada vez más utilizadas debido a que ofrecen soluciones a la medida de los requerimientos de la aplicación. Comercialmente existen muchas tarjetas con diferentes características aunque también cabe la posibilidad de desarrollarlas como es el caso del módulo propuesto.

## 2. Características del Sistema

En esta sección describiremos las características técnicas de la tarjeta de adquisición diseñada, la misma que consta de tres entradas digitales, tres entradas analógicas de 0 a 5 Vdc, cuatro salidas digitales a relé y una salida analógica de 0 a 10 Vdc y está diseñada para ser la interfaz para la comunicación entre una PC y el variador de velocidad, instalado en el motor del “Grid Picker”, en la banda #1 de la máquina empastadora.

Lo que se desea es que este módulo sea capaz de realizar la adquisición de los datos analógicos y digitales generados por un variador de velocidad cualquiera, enviarlos a la computadora a través del puerto serial para que sean procesados, visualizados y luego establecer algún tipo de control en las salidas, que serán las que actuarán en las entradas del variador.

### 2.1 Diagrama de Bloques del Sistema.

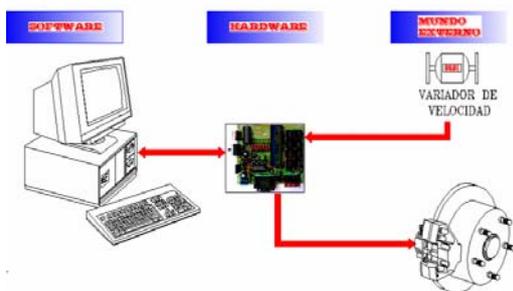


FIGURA 1 Diagrama de bloques del sistema

La figura 1 nos muestra el diagrama de bloques en el que se muestran las diferentes etapas de las que está constituido nuestro proyecto, la misma que consta de cuatro bloques principales que se detallan a continuación:

Nuestro primer bloque es el del software que está desarrollado bajo la plataforma del LabVIEW 7.1 – lo cual describimos más adelante - el objetivo de este bloque es que sea capaz de comunicarse con el Hardware (la tarjeta de adquisición), para transferir y recibir datos, además de tomar ciertas decisiones sobre el comportamiento del hardware.

El segundo bloque está constituido por el hardware necesario para que haya comunicación de la computadora con los bloques siguientes, en otras palabras el hardware es el enlace físico necesario para poder realizar la comunicación entre el primer y el tercer bloque. Para que nuestra tarjeta se comunique con el mundo externo tiene tres entradas digitales y tres entradas analógicas además de cuatro salidas digitales y una salida analógica.

Nuestro tercer bloque está compuesto por un variador de velocidad, que en nuestro caso es el mundo externo, aunque por el diseño nuestra tarjeta puede acoplarse sin ningún problema a cualquier tipo de requerimientos que se le exijan.

Finalmente el cuarto bloque esta constituido por el elemento que es controlado por el variador, es un motor de 3 Hp. Trifásico. A este motor consideramos necesario el acople de un sistema de realimentación hacia la tarjeta que nos permita visualizar la velocidad del mismo en tiempo real.

## 3 Diseño del Hardware.

El objetivo de este diseño es hacer una tarjeta de adquisición de datos basada en un microcontrolador de tres entradas analógicas, tres entradas digitales, tres salidas digitales y una salida analógica.

### 3.1 Entradas

Consta de tres entradas digitales que corresponden al puerto E, ubicados en los pines 8, 9 y 10, está configurado para que cumpla con las funciones de RE0, RE1 Y RE2 respectivamente del PIC 16F877A, el voltaje de polarización de las entradas digitales es de 5 Vdc, podría ser diferente ya que estas entradas están aisladas por opto-acopladores, también consta de tres entradas analógicas que corresponden al puerto A, ubicados en los pines 2 al 4

que están configurados para que cumplan las funciones RA0, RA1 Y RA2 respectivamente, con un rango de trabajo de 0 a 5 Vdc.

### 3.2 Salidas.

Las salidas digitales constan de cuatro relés de 12 Vdc, que se encuentran ubicadas en el puerto B, configurados como salidas digitales, desde RB0 a RB3, los mismos que se encuentran ubicados

desde los pines 33 al 36 del PIC correspondiente, las salidas digitales se encuentran aisladas del relé por medio del integrado ULN2003AN -refiérase al anexo D1.1-. Este aislamiento es necesario ya que las corrientes que manejan las salidas del PIC son muy bajas para polarizar los relés, 10A a 120Vac y 6A 28Vdc. La salida analógica se la obtiene a través de un conversor digital analógico TLC7628CN -refiérase al anexo D1.2-. Este integrado convierte la información enviada en forma digital desde el puerto D del PIC el mismo que se encuentra configurado como salida digital. El puerto D está ubicado en los pines del 19 al 22 y del 27 al 30, es decir que nuestra salida analógica tiene 8 bits de resolución. Para obtener un voltaje de salida de 0 a 10 Vdc fue necesario utilizar el integrado TL084 -refiérase al anexo D1.3.-, configurado como acoplador de impedancia y como amplificador con ganancia 2.

### 3.3 Alimentación.

La alimentación de la tarjeta es de 120 Vac. En el interior de la tarjeta se han construido tres fuentes de +12, -12 y +5 Vdc, para obtener estos voltajes se trabaja con los reguladores 7812, 7912 y 7805 respectivamente. La fuente de -12 Vdc fue necesaria para polarizar el integrado TL084. La fuente de +12 Vdc fue necesaria para polarizar a los integrados y a los relés mientras que la de +5 Vdc se la utilizó para polarizar el PIC.

### 3.4 Selección del Microcontrolador.

Para seleccionar el microcontrolador a usar se debió considerar los siguientes aspectos: cantidad de entradas y salidas, recursos internos que debería tener el micro, accesibilidad y precios. La capacidad mínima que debía tener el PIC era de 15 entradas y salidas digitales, 3 canales analógicos, capacidad de comunicación serial y convertidores analógicos/digitales para el tratamiento de los datos analógicos. En el mercado actualmente los microcontroladores más usados son los de la familia MICROCHIP, por esta razón el integrado que usamos es el PIC16F877A.

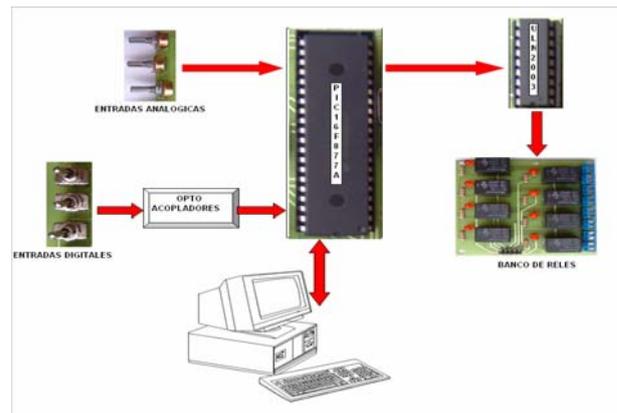


FIGURA 2 Diagrama de bloques de la tarjeta

La figura 2 nos muestra como está dividida la tarjeta de adquisición. El bloque central está constituido por el PIC 16F877A. El bloque de las entradas digitales está representado por tres botoneras, el bloque de opto-acopladores está constituido por tres acopladores tipo 4N25 y uno 4432 -refiérase al anexo D1.5-. El bloque de las entradas analógicas está representado por tres potenciómetros que varían de 0 a 5 Vdc, se encuentran conectado al bloque central a través del OPA4342 que es un OPAM configurado en modo acoplador de impedancia o ganancia unitaria. El bloque de comunicación entre el computador y el PIC se logra mediante el USART incluido en el PIC y el integrado MAX232 que se encarga de convertir las señales TTL a niveles de voltaje establecidos por el protocolo RS232.

El bloque de acoplamiento de las salidas digitales lo constituye el integrado ULN2003 que contiene 8 salidas tipo DARLINGTON, para manejar el bloque de salida compuesto por 4 relés con las características mencionadas anteriormente. El bloque del convertidor digital analógico está compuesto por TLC6528, este bloque se encarga de convertir la información digital proveniente del bloque central en una señal analógica, esta señal analógica pasa por el bloque conformado por TL084 que es el que se encarga de convertir la señal analógica a niveles de voltajes deseados en el bloque de salida analógica.

### 3.5 Diseño de la tarjeta en PROTEL.

Para el diseño de la placa se consideró realizar el revelado de las pistas en una sola capa, aprovechando todos los espacios de tal manera que se puedan reducir costos al máximo. El diseño final de las pistas fue el que se muestra a continuación,

### 3.4 Diagrama de Bloques de la Tarjeta de Adquisición.

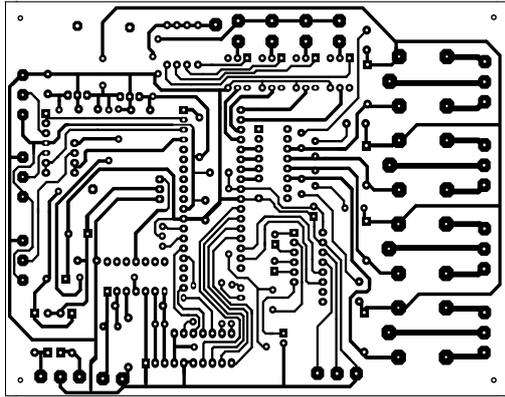


FIGURA 3 Circuito Impreso de las pistas de la Tarjeta.

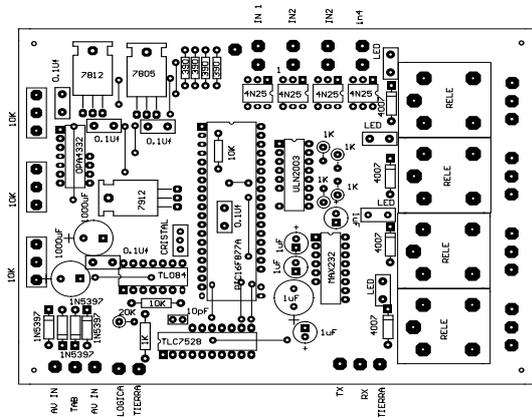


FIGURA 4 Circuito Impreso de los Elementos de la Tarjeta.

### 3.6. Diseño del Software para el PIC.

#### 3.6.1 Introducción.

Una de las partes más complicadas e interesante del desarrollo de la tarjeta, fue realizar el programa que iba hacer cargado en el PIC, el principal problema a resolver era establecer un protocolo de comunicación entre el software desarrollado en LabVIEW (refiérase al capítulo 5), y la tarjeta de adquisición, ya que si bien es cierto la información iba hacer transmitida a través del puerto serie en código ASCII, era necesario diferenciar los tipos de señales.

Envío desde PC	Respuesta del PIC
1	Enciende Salida # 1
2	Enciende Salida # 2
3	Enciende Salida # 3
4	Enciende Salida # 4
5	Apaga Salida # 1
6	Apaga Salida # 2
7	Apaga Salida # 3
8	Apaga Salida # 4

TABLA 1 Comandos que recibe el PIC para actuar en las salidas digitales

Para las salidas digitales se implementó el siguiente código: Cada vez que se envíe una letra ente el 1-8, desde la PC, el PIC activará o desactivará una de las cuatro salidas digitales, como se muestra en la tabla 2

Tal como se muestra en la tabla 1 el funcionamiento es de la siguiente manera: Cuando la PC transmita un “1”, el PIC pondrá en “1” el pin programado como salida digital #1 (RB0), de la misma forma cuando la PC transmita un “5”, el PIC pondrá en “0” el pin programado como la salida digital #1 (RB0). De igual forma se trabaja la salida analógica como se especifica en la tabla 4.7, mostrada a continuación:

Envío desde PC	Respuesta del PIC
l	Decrementa Salida Analógica.
m	Incrementa Salida Analógica
f	Conserva el valor anterior.

Tabla 2. Comandos que recibe el PIC para actuar en la salida analógica.

Cada vez que el PIC recibe una “l” o “m” desde la PC, el PIC decrementa o incrementa la variable F\_DIGITAL, luego envía este valor al puerto D para su conversión, después transmite este valor a la PC, para que sea visualizado. Si el valor recibido por el PIC es una f, el valor de F\_DIGITAL no cambia.

El procesamiento de las entradas analógicas de la tarjeta es como se muestra en la tabla 3, que está detallada a continuación:

Envío desde PC	Respuesta del PIC
0	Envío del canal analógico 0
9	Envío del canal analógico 1
C	Envío del canal analógico 2

Tabla 3 Comandos que recibe el PIC para actuar en las entradas analógicas.

Cuando el PIC recibe uno de los valores de la tabla anterior, inicia la conversión del canal respectivo, luego transmite el valor del canal, hacia la PC, para una mayor facilidad de este proceso se trabajó con 8 bits de resolución de los canales analógicos.

#### 3.6.2 Programación en MPLAB.

El programa en MPLAB que fue cargado en el PIC, está dividido en las siguientes rutinas:

1. El Programa Principal.
2. Servicios de Interrupción.
3. La conversión de los valores Analógicos a Digital.
4. Codificación e identificación de los valores recibidos desde la PC.

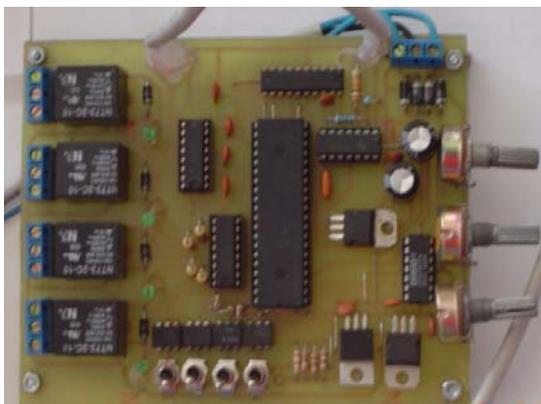
5. El envío de los datos a través del puerto serie a la PC.
6. Las rutinas de retardo.

### 3.7 Fotos de la Tarjeta de Adquisición de Datos.

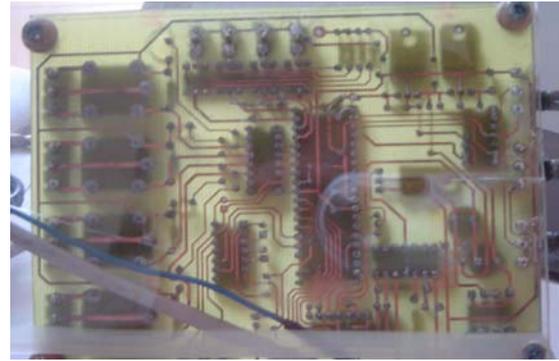
Las siguientes fotos nos muestran todos los detalles de nuestra tarjeta de adquisición de datos, como por ejemplo sus cables de conexión, sus diferentes entradas y salidas digitales, analógicas y así como sus salidas a relé.



**FIGURA 5** Vista superior de la tarjeta de adquisición.



**FIGURA 6** Fotografía general de las entradas y salidas digitales y analógicas.



**FIGURA 7** Fotografía general de las pistas de la tarjeta de adquisición de datos.

## 4. Diseño de la interfaz gráfica

Para controlar un módulo de adquisición de datos es necesario desarrollar una interfaz gráfica, y para ello hay que seleccionar un programa que permita realizar esta tarea de una manera sencilla y eficiente, por ese motivo en este caso se seleccionó el programa LabVIEW 7.1 para el desarrollo de la interfaz., la programación en este software resulta muy sencilla ya que no se utilizan líneas de código porque el desarrollo de aplicaciones se hace seleccionando, configurando y enlazando objetos o drivers, disponibles en el propio programa.

### 4.1. Características y herramientas LabVIEW.

LabVIEW es un programa de instrumentación virtual con un ambiente de desarrollo gráfico (utiliza un lenguaje de programación gráfico) utilizado en ciencias e ingeniería que incluye herramientas funcionales para simulación, adquisición de datos, análisis de mediciones y presentación y almacenamiento de datos. Usando LabVIEW usted puede crear sus propios instrumentos virtuales (VIs) y con esto obtener soluciones fáciles y eficientes a problemas comunes en el área industrial e investigativa, los instrumentos virtuales emulan el trabajo de un instrumento real.

Se habla de instrumento virtual cuando se emplea la computadora para realizar funciones de un instrumento clásico pudiendo agregarle al instrumento otras funciones a voluntad, a los programas desarrollados en LabVIEW se los conoce como instrumentos virtuales.

### 4.2. Funciones utilizadas en la interfaz creada en LabVIEW

Debido a que la base de todo VI es el diagrama de bloques, es necesario conocer bien las funciones que

tenemos disponibles en LabVIEW para el desarrollo de cualquier interfaz.

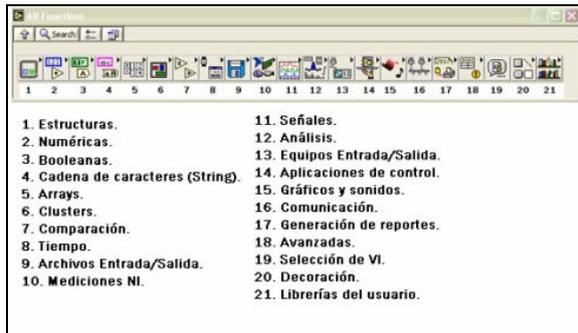


Figura 8. Lista de funciones disponibles en LabVIEW 7.1.

LabVIEW utiliza un lenguaje gráfico de programación para programar primero, se deben conocer los tipos de datos, la ejecución del flujo de datos, conocer los ciclos Mientras (While) y Para (For) y también estructuras de caso.

LabVIEW tiene una variada gama de tipos de datos. Cuando se pone un control o indicador en el panel frontal, LabView pone un terminal correspondiente a este control o indicador en el diagrama en bloque. Este terminal desaparece solo cuando se borra el control o indicador. Los enlaces o alambres son los caminos de los datos entre los terminales fuente y los terminales destino. No se permite enlazar dos terminales fuentes ni dos terminales destinos, y si se permite enlazar un terminal fuente a varios terminales destino.

En La figura 9 se muestra la representación de los tipos de datos en LabVIEW.

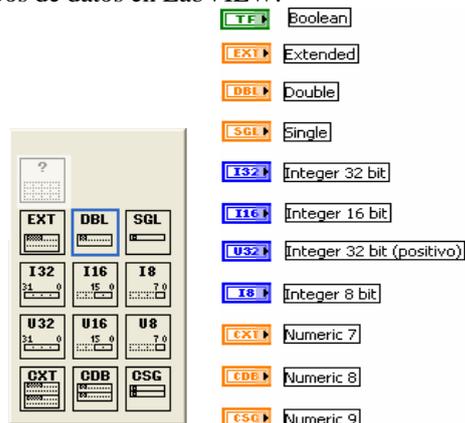


Figura 9. Tipos de datos en LabVIEW

LabVIEW sigue un modelo de flujo de datos para ejecutar los VIs. Un nodo del diagrama de bloque se ejecuta cuando todas sus entradas están disponibles. Cuando un nodo completa la ejecución, suministra datos a sus terminales de salida y pasa los

datos de salida al siguiente nodo en la trayectoria del flujo de datos.

En LabVIEW existen los ciclos Mientras (While) y Para (For) están localizados en la paleta "Functions Structures". El ciclo Para difiere del ciclo Mientras en que el ciclo Para se ejecuta una serie de veces predeterminada. El ciclo Mientras se deja de ejecutar solo si el valor en la condición terminal existe.

**Ciclo Mientras.-** Similar al ciclo Haga (Do) o al ciclo Repita-hasta (Repeat-Until) en lenguajes de programación basados en texto, un ciclo Mientras, ejecuta un sub diagrama hasta que la condición sea cumplida.

El ciclo Mientras ejecuta el sub diagrama hasta que la terminal dependiente, recibe un valor Booleano específico. El comportamiento y la apariencia de la terminal dependiente es Continue if True (continúe si es Verdadero), mostrado a la derecha de la figura 10, en el diagrama de bloques. La terminal de iteración (una terminal de salida), mostrada a la izquierda, contiene el numero de iteraciones completas. El conteo de iteraciones siempre empieza en cero. Durante la primera iteración, la terminal de iteración regresa a cero.

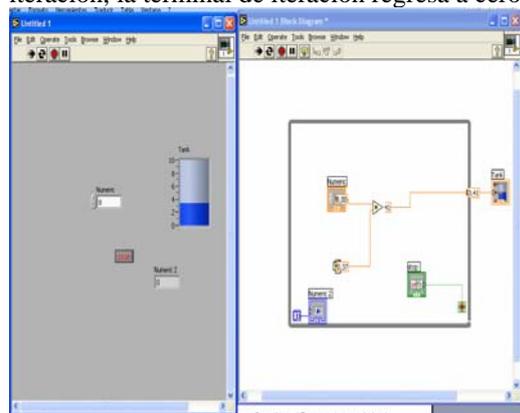


Figura 10. Ciclo Mientras (While) continúa si es verdadero

Si el ciclo es Stop if true (Detener si es verdadero) tiene la misma configuración del ciclo anterior, la única diferencia, es que el ciclo se detiene si la acción booleana es verdadero.

### Ciclo For (Para)

Un ciclo For (Para), mostrado en la figura 11, ejecuta una serie varias veces. El valor en la terminal de conteo (una terminal de entrada) representada por la N, indica cuantas veces repetir el sub diagrama y para el caso de la figura 11 es de 99 (N-1). La terminal de iteración (una entrada de salida), contiene el número de iteraciones completas. El conteo de iteraciones siempre empieza en cero.

Durante la primera iteración, la terminal de iteración regresa a cero.

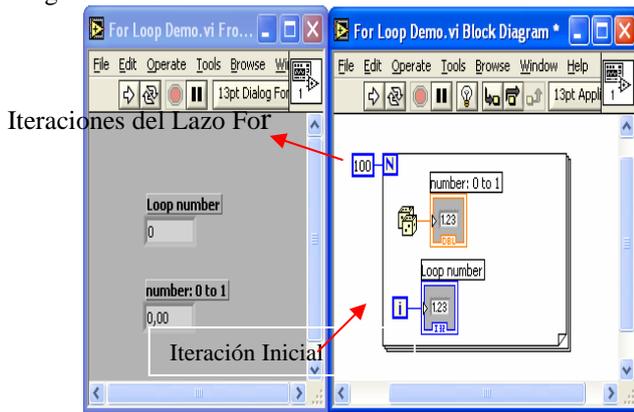


Figura 11. Ciclo For (Para)

Para la comunicación serial se utilizaron las funciones disponibles en el VISA, las mismas que se muestran en la figura 12, estas funciones nos permiten abrir o cerrar el puerto serial, así como escribir o leer en el mismo.



Figura 12. Funciones para comunicación serial.

Además de las funciones para la elaboración del diagrama de bloques LabVIEW permite emplear otros VIs, como si fueran objetos para ser utilizados dentro del diagrama de bloques de otra aplicación y ahí reciben el nombre de SubVIs. En nuestro caso se han empleado tres SubVIs dentro de la interfaz creada, que realizan tareas específicas y entregan los resultados al diagrama de bloques del programa principal.

### 4.3. Diseño del Panel de Control en LabVIEW para la tarjeta de adquisición.

En la figura 13 se muestra el panel de control diseñado con el objeto de monitorear y controlar el variador de velocidad del motor “Grid Picker” descrito en los capítulos anteriores.

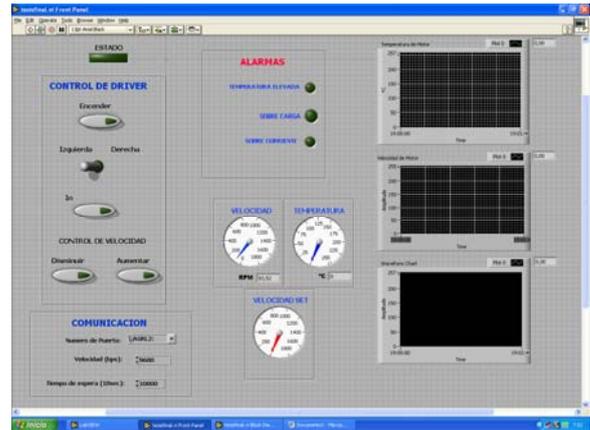


Figura 13. Panel de Control diseñado para el control de un variador de velocidad.

El Indicador “Estado”, nos indica si el sistema está habilitado o no desde la tarjeta de adquisición. Este habilitador es la Entrada Digital número 0, la programación realizada en el Diagrama de Bloques de LabVIEW pregunta por esta entrada inmediatamente después de abrir el puerto. Como vemos en las figuras 14 y 15 y según el protocolo desarrollado para la comunicación entre la tarjeta de adquisición y la PC, para saber el estado de la entrada digital número 0 debemos enviar el carácter “a”.

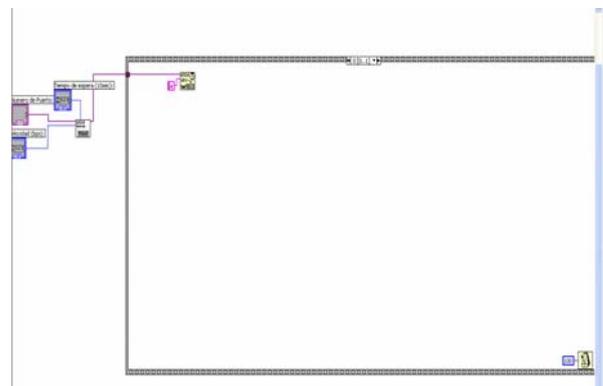
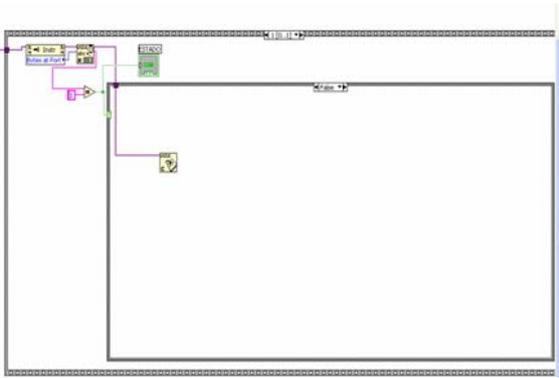


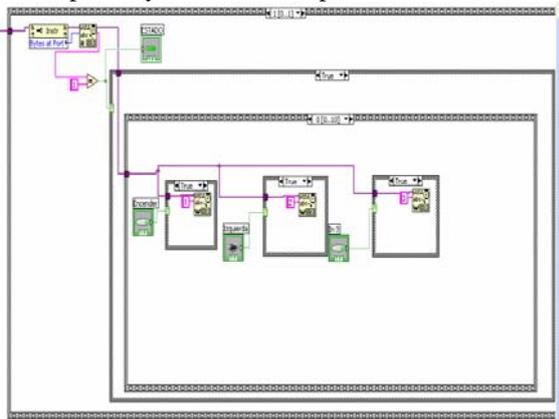
Figura 14 Diagrama de Bloques con el primer paso del programa diseñado.

Para realizar eso utilizamos la estructura “Stacked Sequence”, en la frama 0 (Figura 14) se escribe en COM 1 el carácter “a”, luego en la frama 1 (Figura 15), leemos el puerto y revisamos el contenido de este.



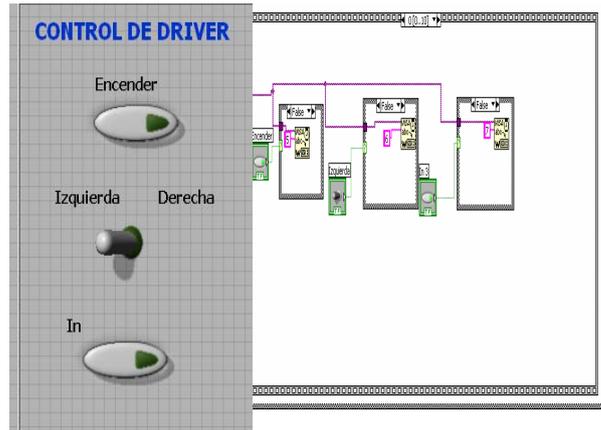
**Figura 15.** Diagrama de Bloque Trama 1 de la estructura principal y la ejecución por FALSE de la estructura CASE

Para el siguiente paso utilizamos una estructura "CASE", si el carácter leído desde COM 1 es "1" (Figura 16), se ejecuta la opción por TRUE de la estructura y el resto del proceso está habilitado; si el carácter leído desde COM 1 es "0" (Figura 15), se ejecuta la opción por FALSE de la estructura que cierra el puerto y continua en espera del "1".



**Figura 16.** Diagrama de Bloque Trama 1 de la estructura principal y la ejecución por TRUE de la estructura CASE

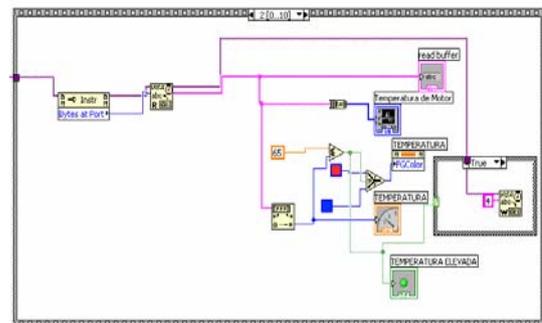
Mientras la entrada digital 0 esté habilitada, se ejecuta las instrucciones contenidas en la condición TRUE de la estructura CASE. La siguiente estructura interior es un "Stacked Sequence" de 11 framass, la trama 0 se muestra en la figura 16 y muestra los tres controles que manejan las salidas digitales de la tarjeta de adquisición utilizadas para el control del driver de velocidad. En la figura 17 se muestra la parte del panel de control de las salidas digitales y del diagrama de bloques correspondiente al manejo de las salidas digitales. En esta trama escribimos en el COM 1 el código descrito en las páginas anteriores de este documento de acuerdo al estado de cada uno de los controles.



**Figura 17.** Diseño en LabVIEW para el manejo de las salidas digitales de la tarjeta de adquisición.

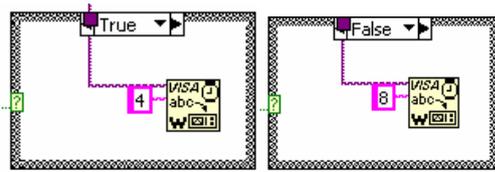
La trama número 1 escribe en el puerto el carácter "C" y genera un retardo de 50 ms. Como se describió anteriormente cuando la tarjeta recibe el carácter "C" espera un tiempo y luego envía por el COM 1 el contenido del canal analógico número 2 que simula la entrada desde un sensor de temperatura LM 35 instalado en la carcasa del motor.

En la trama 2 se realiza el procedimiento de lectura del canal analógico número 2, como se muestra en la figura 18.



**FIGURA 18.** Trama # 2 correspondiente a la adquisición y visualización del canal analógico # 2

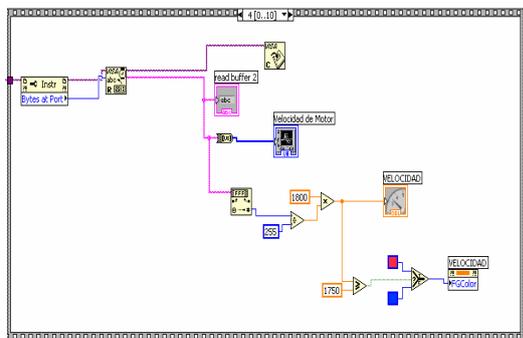
En la figura anterior realizamos el procedimiento para visualizar el canal 2 mediante el gauge "TEMPERATURA" así como graficar mediante el Waveform Chart "Temperatura de Motor". También realizamos una comparación para encender la alarma "TEMPERATURA ELEVADA" en caso de que el valor de temperatura exceda los 65° C, con esta condición se activa también la salida digital numero 4 de la tarjeta de adquisición, esto se hace con la estructura CASE mostrada en la parte derecha de la figura 18 y en la figura 19.



**FIGURA 19.** Estructura CASE para el manejo de la salida digital número 4.

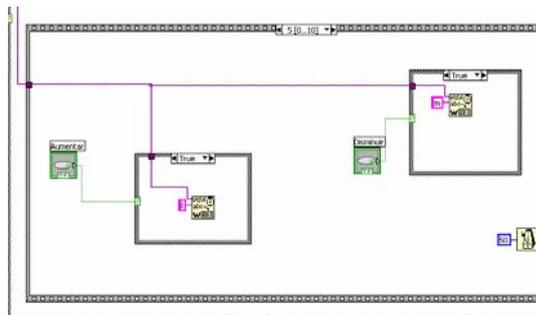
La frama número 3 escribe en el puerto el caracter “0” y genera un retardo de 50 ms. Como se describió anteriormente cuando la tarjeta recibe el caracter “0” espera un tiempo y luego envía por el COM 1 el contenido del canal analógico número 1 que simula la entrada desde un sensor para la velocidad del motor.

En la frama 4 se procede a leer el contenido del COM 1 correspondiente a la entrada analógica 1, como vemos en la figura 20 la programación es similar a la realizada en la frama 2.



**FIGURA 20.** Frama 4 realizada para la adquisición y visualización del canal analógico 1, correspondiente a la velocidad del motor.

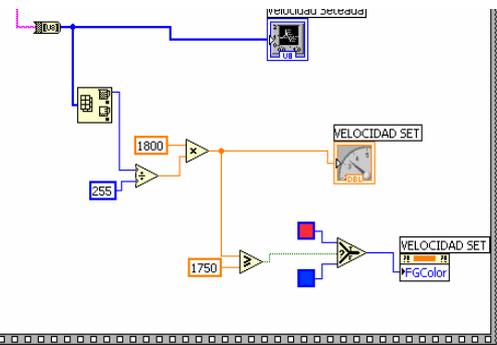
La frama 5 posee los controles digitales “Aumentar” y “Disminuir” como se muestra en la figura 21, el objetivo de los controles nombrados anteriormente es manejar el acumulador utilizado para la salida analógica de la tarjeta -la forma de trabajo de los controles es tipo botonera-. Cuando el control “Aumentar” es verdadero, se escribe en el COM 1 el caracter “1”, Si el control “Disminuir” es verdadero, se escribe en el COM 1 el caracter “m”



**FIGURA 21** Frama 5 realizada para el manejo de la salida analógica, se muestran los ciclos tanto para Aumentar o Disminuir su Valor

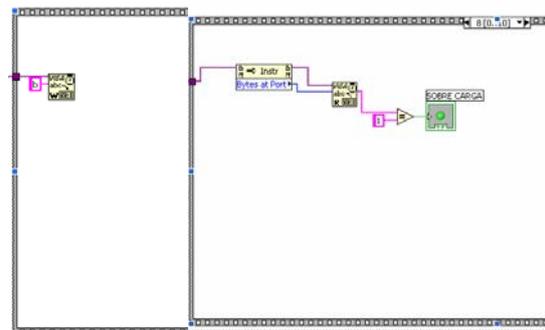
En la figura 5.24 se muestra la frama número 6, esta frama contiene la programación que se realizó para mostrar el valor que tiene almacenado el microcontrolador, al cual le asignamos el nombre “VELOCIDAD SET”, con este canal intentamos, este valor es manejado desde los controles “AUMENTAR” y “DISMINUIR” explicados anteriormente.

En esta frama también se realiza una pequeña programación para que el color del gauge “VELOCIDAD SET” cambie a rojo una vez que el valor seteado sea mayor a 1750 RPM.



**FIGURA 22** Frama 6 realizada para mostrar la salida analógica en el panel de control

El control de las entradas digitales 2 y 3 se las realiza siguiendo el protocolo explicado en el capítulo cuatro, primero escribimos el carácter “b” en la frama 7 (figura 23) y luego de un retardo leemos en la frama 8, (mostrada en la figura 23) si el valor que me envió la tarjeta es un “1” o un “0”, dependiendo de el carácter enviado procedemos a encender o apagar el indicador “SOBRE CARGA” ubicado en el cuadro ALARMAS del panel de control (Figura 13).



**FIGURA 23.** Framas 7 y 8 utilizadas para leer el valor de la entrada digital #2

Finalmente las framas 9 y 10 nos ayudan a visualizar la entrada digital número 3 estas framas poseen una

programación similar a las framas 7 y 8 con la diferencia que el carácter enviado desde LabVIEW es el “c” y el indicador sobre el que se actúa es “SOBRE CORRIENTE” ubicado en el cuadro ALARMA del panel de control.

## 5. Costos.

Enero 12 del 2006  
Proyecto: Tarjeta de Adquisición de Datos.

ITEM	DESCRIPCION	UN	CANT	P/UN	SUBTOTAL
<b>LISTA DE MATERIALES PARA ARMAR LA TARJETA</b>					
1	Resistenciasde 10k	U	6	0,05	0,30
2	Resistenciasde 1k	U	5	0,04	0,20
3	Resistenciasde 20k	U	2	0,12	0,24
4	Regulador LM 7812	U	1	0,75	0,75
5	Regulador LM 7805	U	1	0,75	0,75
6	Regulador LM 7912	U	1	0,85	0,85
7	CI 390	U	4	0,55	2,20
8	CI 40N25	U	4	0,70	2,80
9	CI ULN2003	U	1	1,50	1,50
10	CI MAX232	U	1	3,00	3,00
11	Capacitores 0,1uf	U	10	0,15	1,50
12	Capacitores 1000uf	U	2	0,50	1,00
13	Capacitores 10pf	U	1	0,30	0,30
14	4007	U	4	0,80	3,20
15	Diodo Emisor de luz	U	4	0,10	0,40
16	Relay 12 Vd-12A	U	4	1,50	6,00
17	Cristal de Cuarzo 4MHz.	U	1	0,80	0,80
18	CI TL084	U	1	1,50	1,50
19	CI TLC7528	U	1	1,60	1,60
20	PIC16F877A	U	1	9,00	9,00
21	1N5397	U	4	1,20	4,80
22	OPA4332	U	1	0,90	0,90
23	Potenciómetro de 10k	U	3	0,40	1,20
24	Cable de Comunicación	U	1	1,50	1,50
25	Borneras	U	3	0,20	0,60
26	Interruptores	U	4	0,30	1,20
27	Adaptadores 12Vdc: fijo	U	2	4,25	8,50
28	Circuito Impreso	Gb	1	20,00	20,00
29	Zócalo de 40 pines	U	1	0,25	0,25
30	Zócalo de 18 pines	U	1	0,10	0,10

<b>VALOR DEL PRESUPUESTO</b>	<b>USD 76,94</b>
<b>IVA 12%</b>	<b>USD 9,23</b>
<b>VALOR TOTAL DEL PRESUPUESTO</b>	<b>USD 86,17</b>

**Tabla 4.** Lista de los materiales utilizados y sus respectivos precios en el mercado local.

El valor total del proyecto como podemos observar en la tabla 4 ascendió a \$86.17, el diseño de esta tarjeta está hecho con el mejor material, una excelente placa lo que garantiza un mejor funcionamiento y una durabilidad muy buena al proyecto.

## 6. Conclusiones

La tarjeta de adquisición diseñada y su sistema de monitoreo y control desarrollado en LabVIEW tiene múltiples aplicaciones, no solamente en el control de un variador de velocidad, puede ser muy útil no solamente en aplicaciones industriales si no para fines pedagógicos e inclusive para incentivar a adolescentes para ingresar a este inimaginable mundo que es la programación industrial.

Aparte de las señales que están siendo mostradas en el panel de control, la programación que posee el microcontrolador nos permitiría añadir otra entrada analógica así como otra entrada y otra salida digital, nosotros no las incluimos en esta aplicación por que no eran necesarias para este caso específico.

El costo total del proyecto es de 86 dólares aproximadamente, este costo es el 20% del costo de una sencilla tarjeta de adquisición de National Instruments (La cual asciende a los 500 dólares), la diferencia se debe a que si nosotros usamos una tarjeta de National Instruments, es menos complejo el diseño en LabVIEW y nos evitamos el trabajo del diseño de la tarjeta y programación del controlador.

## 7. Referencias

- [1] Bishop, R, “Learning LabVIEW 7.1”, University of Texas, United States, Prentice Hall, 2004, pp. 175-236.
- [2] Angulo, J., and Romero, S., “Microcontroladores PIC Diseño Práctico de aplicaciones Segunda parte PIC 16F877 PBasic y ensamblador” , Universidad de Deusto, España, 1981, pp. 105-206.
- [3] National Instruments., “Lookout Developer`s Manual”, United States, November 2001, pp. .