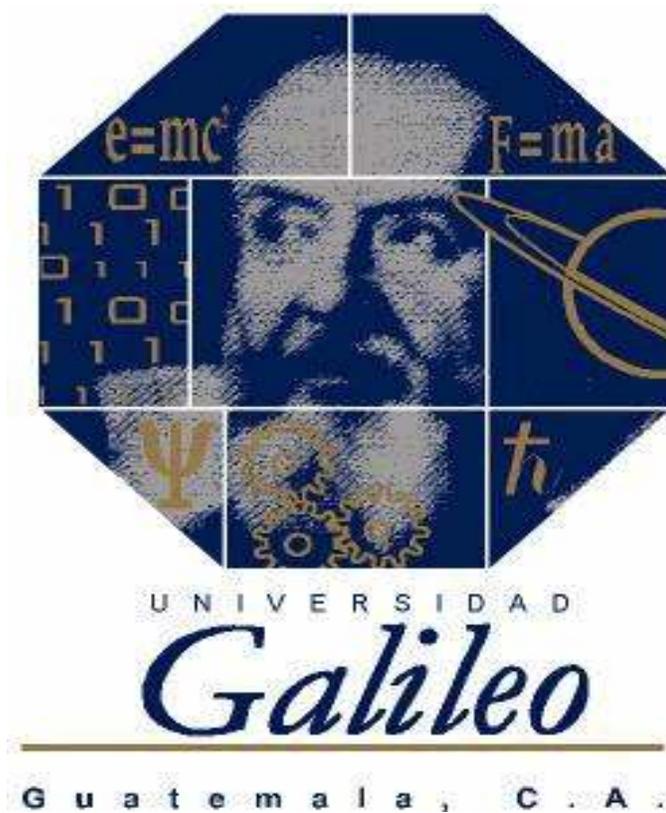


**Gustavo Adolfo Peláez Coronado**

**Sistema de control y monitoreo de motores trifásicos con  
soporte de LabVIEW**



**FACULTAD DE INGENIERÍA DE SISTEMAS,  
INFORMÁTICA Y CIENCIAS DE LA COMPUTACIÓN**

**GUATEMALA, 2007**

Esta tesis fue elaborada por el autor como requisito previo a obtener el título de INGENIERO EN ELECTRÓNICA INFORMÁTICA Y CIENCIAS DE LA COMPUTACIÓN

Guatemala, diciembre de 2007

## **Agradecimientos**

A Dios, a mi familia y el resto de personas que ayudaron, ustedes saben quienes son.

# **INDICE**

CAPÍTULOS	PÁGINA
<b><u>I. Introducción</u></b>	1
<b><u>II. Estándar RS-232C</u></b>	
A. Codificación NRZ	6
B. Comunicación serial	7
C. Niveles de voltaje asociados	8
D. Baudaje y bits por segundo	9
E. Comunicación síncrona y asíncrona	10
F. Conversión de estándares TTL-RS232	12
G. Tipos de conectores	14
<b><u>III. Dispositivos HMI</u></b>	
A. Definición de HMI	16
B. Importancia de HMI	18
C. Tipos de HMI	20
D. Estándares internacionales	23
<b><u>IV. Motores trifásicos asíncronos</u></b>	
A. Inducción electromagnética	25
B. Modelo eléctrico	26
C. Velocidad de sincronismo y rotación	27
<b><u>V. Variadores de frecuencia</u></b>	
A. Definición de variador de frecuencia	29
B. Construcción interna	29
C. PWM y señales sinusoidales	31
D. Importancia de la variación de frecuencia	32

## **VI. Programación en LabVIEW**

A. Tipo de programación	36
B. Configuración de puerto serial	37
C. Construcción de cadenas de caracteres	39
D. Secuencias y ciclos	41

## **VII. Aplicación desarrollada**

A. Características generales	43
B. Inicio de sesión dentro del programa	45
C. Adquisición de datos dentro del programa	46
D. Actualización de parámetros dentro del programa	47
E. Requisitos de operación	49
F. Funcionamiento	49
1. Lecturas de parámetros	50
2. Actualización de parámetros	52
3. Interpretación de mensajes recibidos	53
4. Otras funciones	54

## **VIII. Identificación y resolución de problemas comunes**

A. Problemas de comunicación	57
1. Alambrado	58
2. Mensajes erróneos	59
3. Velocidades de transmisión	60
4. Estado del variador de frecuencia	61

## **IX. Conclusiones**

## **X. Recomendaciones**

## **XI. Bibliografía**

## I. Introducción

Un sistema de control y monitoreo es necesario en diversas partes de la industria de acuerdo a las necesidades del proceso. Dicho problema involucra una serie de dispositivos y conocimientos que van desde métodos de transmisión de mensajes entre dispositivos hasta protocolos y circuitos eléctricos. Normalmente estos sistemas de control se limitan a funciones de encendido, puesta en marcha, apagado o detención del motor y estas funciones son consideradas básicas pero esenciales para el control. Supervisar variables de un proceso en una planta puede resultar difícil si la complejidad del proceso es grande. Esto conlleva un conjunto de variables bastante grande para supervisar en su totalidad, que a su vez implica de instrumentos capaces de manejar toda esta complejidad. Si lo que se está supervisando no es un proceso, sino un dispositivo, entonces la observación de ciertas variables puede ser muy importante porque, de ser un dispositivo sensible y caro, hemos de tener cuidado con aquellas propiedades que lo dañen, como sería un aumento inesperado en el voltaje y la corriente de alimentación.

Sin embargo, conforme avanza la tecnología, es común que estos procesos aumenten en su complejidad, y por ende, exista una mayor probabilidad de ocurrencia de errores humanos en su operación. El éxito o fracaso del proceso dependerá de dos variables que pueden estar presentes en un mismo procedimiento, estas son, la exactitud de los valores consultados y la simplicidad con la que estos se presenten al exterior, para que el usuario pueda tomar la decisión correcta. La última variable puede omitirse de acuerdo al nivel de automatización del proceso pero dejando caer mayor peso sobre la exactitud de la adquisición de parámetros y confiando en la toma de decisiones del programa que gobierna el proceso. Un proceso totalmente automatizado puede, a la larga, resultar más económico, ya que implica un mayor rendimiento entre costo y eficiencia de trabajo a pesar del mantenimiento que haya que brindar a la máquina. Pero aún cuando un proceso esté automatizado en su mayoría, siempre habrá interacción con el ser humano, de ahí que resulta necesario hacer la interfaz sencilla y fácil de entender, al considerar los factores que limiten al operario para el manejo.

Una respuesta a esta problemática se encuentra en las interfaces de hombre y máquina (HMI por sus siglas en inglés), que son medios de comunicación entre el operario y el proceso de control y tienen como objetivo reducir la complejidad en un

proceso que resulte difícil de operar. Esto se lleva a cabo cuando se simplifica la cantidad e información mostrada al reducirle hasta sólo desplegar lo esencial para su entendimiento y realiza procedimientos transparentes para el usuario cuando este indique la ejecución de una tarea. La mejor forma de que el usuario reciba información de un proceso es por medio de señales visuales, las cuales pueden variar en una gamma bastante amplia desde un indicador con una lámpara incandescente hasta un diagrama de flujo con información de parámetros esenciales de cada parte que lo conforma. Actualmente se ha decidido por estas últimas, gracias a los avances en las herramientas de software, que permiten una mejor representación gráfica de lo que está sucediendo dentro del proceso para que el operario reciba de mejor forma la información importante. Estas pantallas pueden representar el proceso en diagramas de bloques que se identifican con el nombre del proceso que se está ejecutando, mientras o en diagramas pictográficos donde cada parte del diagrama de flujo, se representa con figuras o símbolos similares a los reales.

Por lo tanto, el presente trabajo es una propuesta de implementar una aplicación tanto en hardware como en software, con el objetivo de resolver la complejidad y dificultad en el control de un motor trifásico. En sí, el control se desarrolla por medio de envío de mensajes a un variador de frecuencias que adquiere los datos de acuerdo al comportamiento del motor conectado a él. No solamente estos parámetros propios del motor están disponibles, también los del variador de frecuencias como las frecuencias límite, estado general y voltaje aplicado, entre otros. Además, permite el control con un grupo de funciones para poder efectuar una gamma de tareas que el variador de frecuencias realiza sobre el motor trifásico pero como es de pensarse, estas tareas deben ser accedidas por medio de un comando enviado al variador de frecuencias. Como se verá a detalle, existe una tabla que resume estas funciones para el uso de una forma sencilla pero muy precisa al momento de construir un mensaje para controlar. Es bastante común ver un variador de frecuencias en la industria para manejar un motor trifásico por los parámetros disponibles con esta herramienta. La importancia de estos en la industria reside, entre otras, por los parámetros y en la manera robusta pero sencilla de controlar un motor. Dado que la precisión del instrumento es bastante buena, un variador de frecuencias puede manejar motores que muevan una banda transportadora hasta aspiradoras industriales.



Figura 1.1: Aspiradoras industriales a base de motores trifásicos de inducción

Además de las características que tiene un variador de frecuencias, debe considerarse que en el proceso de automatización de una planta, esta herramienta resulta ser una pieza indispensable. Aunque existen otros modelos para reemplazar el variador de frecuencias utilizado, debe tomarse en cuenta que el esquema sigue siendo el mismo, solamente debe cambiarse el formato del mensaje enviado para adaptarse al nuevo dispositivo. Por lo tanto, la aplicación que se ha desarrollado permite cierta flexibilidad para adaptarse a otros dispositivos, sin sacrificar la simplicidad del despliegue de parámetros esenciales y utilizar poca ayuda de otros dispositivos como el variador de frecuencias y el cable para comunicación serial. Se asume que el motor a supervisar está dentro del grupo de motores que el variador de frecuencias puede manejar, por lo que no se toma en cuenta esta posibilidad pero si debe ser así porque podrían obtenerse errores, a pesar de que el sistema funcione de forma adecuada.

La operación del variador de frecuencias debe realizarse desde un punto remoto, dando desventajas como la interfaz entre el operario y el variador, así como la necesidad de estar muy cerca del lugar donde funciona el variador, ya que pone en riesgo al operario. El trabajo desarrollado en la presente tesis resuelve dicho problema con el uso del variador de frecuencias marca Danfoss modelo VLT-2020 al permitir la visualización más agradable de ciertos parámetros, así como la modificación de

parámetros importantes del variador y el control del motor desde un punto separado de ellos. El sistema de control y monitoreos puede ser instalado en cualquier computadora personal que cumpla con los requisitos mínimos para su funcionamiento, por otra parte, el motor y el variador de frecuencias podrán estar funcionando si se cuenta con la infraestructura, específicamente de alimentación eléctrica.

La aplicación desarrollada permite la posibilidad de modificaciones menores para adaptarse mejor a la tarea que se le ha asignado. Se pueden adquirir los mismos parámetros de otros dispositivos si se cuenta con los sensores necesarios. El papel de esta aplicación, con leves modificaciones puede controlar y consultar varios parámetros con sólo presionar un botón de consulta o actualización de variables. La realización de esta acción resulta posible gracias al envío de mensajes simultáneos hacia el dispositivo que se está controlando.

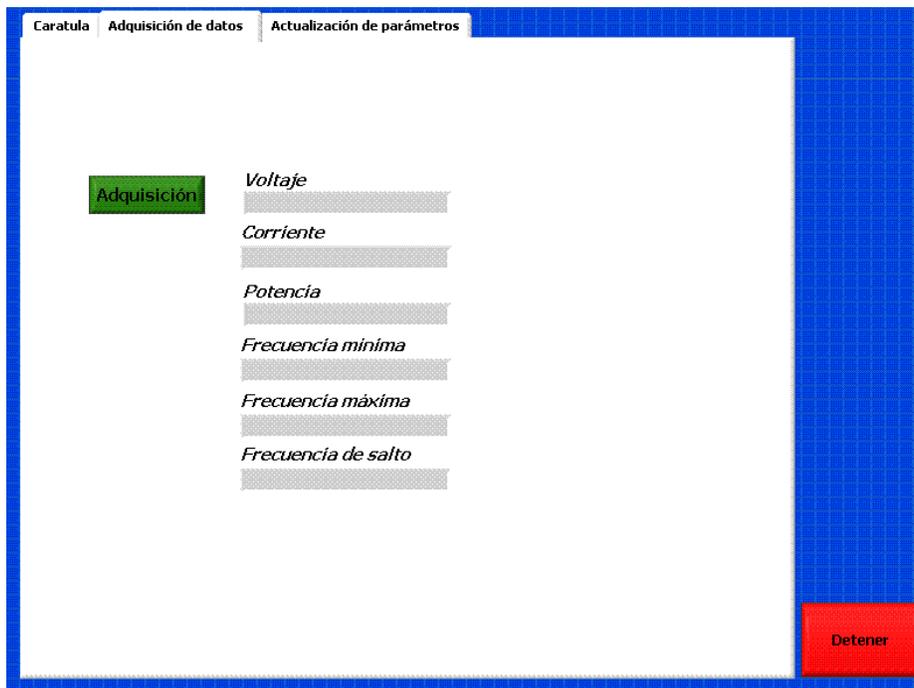


Figura 1.2: Sección de adquisición de datos de la aplicación desarrollada

Fue tomado en cuenta el tiempo que el variador o instrumento tarde en entregar los resultados hacia la computadora, lo cual permite que los mensajes enviados sean ejecutados para obtener un resultado del estado o valor de una variable, así como una función deseada para el dispositivo que se está controlando.

En la siguiente figura se observa un diagrama que explica los componentes utilizados para la realización de dicho proyecto. Se usó una computadora con el programa de monitoreo instalado, y este se comunica a través del protocolo RS232 con el variador de frecuencias de marca Danfoss VLT 2020. El variador de frecuencias recibe alimentación monofásica de 220V con frecuencia de 60Hz y este se conecta al motor trifásico que requiere alimentación trifásica proveniente del variador de frecuencias con velocidades y frecuencias variables para regular su velocidad, como se explicará posteriormente.

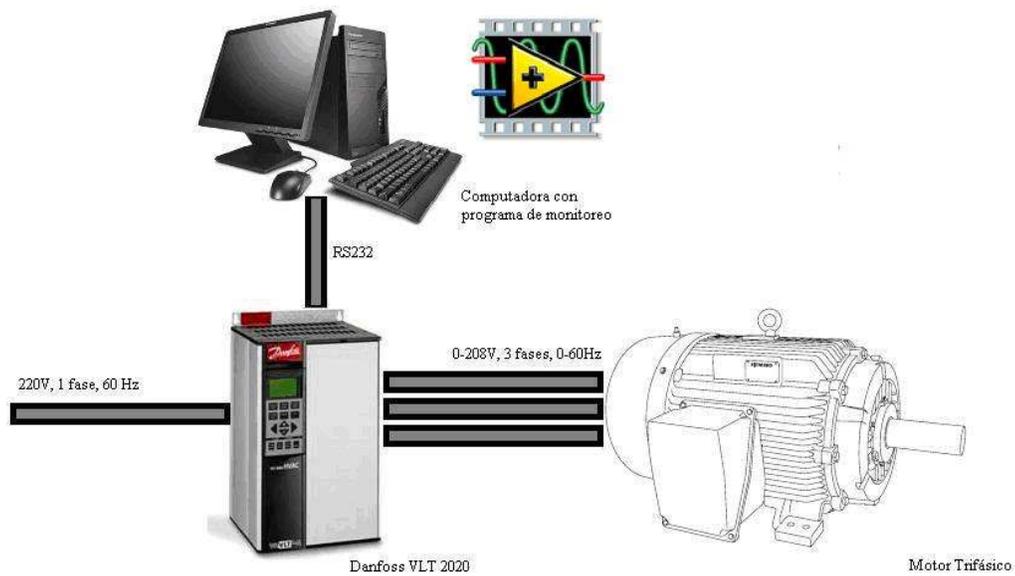


Figura 1.3: Diagrama de aplicación desarrollada.

## II. Estándar RS-232C

### A. Codificación NRZ

La codificación NRZ (non return to zero), también llamado NRZL (non return to zero level), se refiere a todo aquel modo de transmitir datos a través de un medio de transmisión que permita definir estados lógicos como niveles de voltaje distintos a cero. Este método de transmisión se puede utilizar en diversos tipos de comunicación ya existentes de acuerdo a los requerimientos deseados. Existen diversas ventajas de esta codificación sobre otras, por ejemplo, utiliza la mitad del ancho de banda que en el caso de la codificación Manchester dada una tasa de bits constante. A esto hay que agregar que los estados lógicos, que tienen un voltaje asociado para su representación, no vuelven en ningún momento a un estado distinto del voltaje a representar. Este estado se le conoce como estado de descanso o neutral y la ausencia de dicho estado es la principal característica de dicha codificación.

Aunque posee la ventaja de utilizar menor ancho de banda, tiene el problema de la interpretación correcta del 1 y 0 correspondientes, ya que tienen duración específica que puede no darse de la misma forma en el receptor al crear errores de recepción del mensaje.

Existen diversos tipos de codificación NRZ, es decir, varias formas de implementar esta serie de condiciones y características de la codificación NRZ. Entre las principales hablaremos de NRZL, NRZI (non return to zero inverse), NRZM (non return to zero mark) y NRZS (non return to zero space).

De acuerdo a la definición, NRZI es precisamente NRZ pero invertido, en donde un 1 lógico estará representado como un 0 lógico de NRZ y viceversa. En el caso de NRZM, podemos definirla como una codificación donde un 1 lógico no viene representado como un nivel de voltaje, sino más bien como un cambio de nivel de voltaje. Es decir, si estábamos en voltajes positivos, ahora usaremos voltajes negativos cuando se presente el siguiente 1 lógico, al contrario si se presenta un 0 lógico seguiremos en voltajes positivos. NRZS es el contrario de NRZM y básicamente consiste en manejar la misma lógica de no representar con niveles de voltaje los estados lógicos, sino con cambios en la línea. Pero a diferencia de NRZS, NRZM tiene la

característica de cambiar el estado de la línea cuando vea un 0 lógico y si ve un 1 lógico mantendrá el estado en la línea.

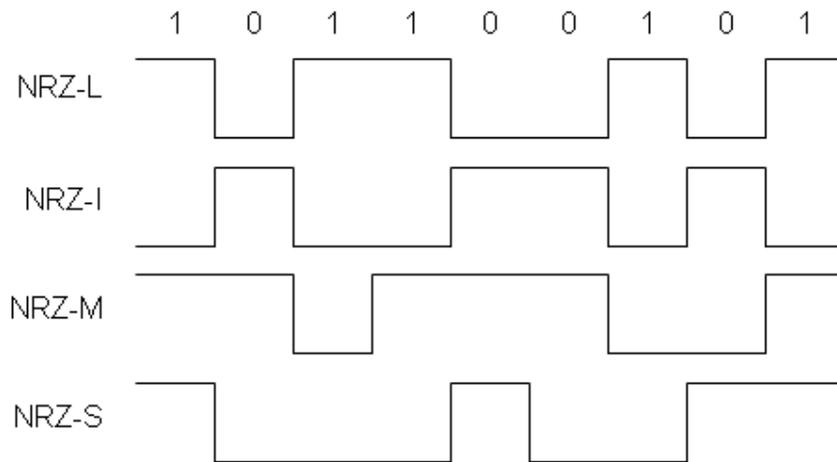


Figura 2.1: Diversos tipos de NRZ

De acuerdo a la figura 1, podemos observar precisamente la inversión de NRZI con NRZL. De la misma forma, pero no tan obvia, NRZM y NRZS se complementan para el cambio de estado de líneas.

## B. Comunicación serial

Entre los diversos tipos de comunicación que existen para transmitir datos, hay algunos que nos brindan mayores ventajas sobre otros, tanto en precios de material, como en forma en que transmitiremos los datos, así como el método o cantidad de bits a enviar por unidad de tiempo.

De estos tipos, utilizaremos la comunicación serial que se define como el envío de una serie de datos entre un emisor y un receptor por un canal tal que un solo BIT de información se envía a la vez. Se utiliza comúnmente para la comunicación donde necesitamos una reducción de pines significativa y por ende reducir costos de implementación. Si bien es cierto, la comunicación serial podría dar la apariencia de que maneja menor cantidad de bits por unidad de tiempo, esto no es cierto. Al contrario, normalmente es la transmisión serial la que maneja mayor cantidad de bits por segundo.

La cantidad de bits por segundo que se manejen por unidad de tiempo va a depender de la forma en la que estemos enviando el mensaje, esto es, a qué velocidad se estará transmitiendo un 1 ó 0 lógico. Sin embargo, cabe mencionar que esta velocidad está limitada por el medio de transmisión elegido, así como la longitud entre el emisor y el receptor.

La comunicación serial presenta ventajas en lo que respecta a su costo de implementación y en la mayoría de casos, velocidad de transmisión de bits. Sin embargo, también posee ciertas desventajas que es importante hacer notar. La ya mencionada dependencia del tiempo, hace que nuestros circuitos necesiten de un método para poder identificar cuándo se está transmitiendo un bit de información y cuándo se dejó de enviar dicho bit. Es decir, la duración que tenga el bit transmitido es de un tiempo determinado que debe ser bastante preciso, para ello deben estar en excelente sincronía o mantener una comunicación constante entre sí. La decodificación del mensaje también es parte importante y conlleva a utilizar herramientas de software capaces de recibir el mensaje de forma serial e interpretarlo de forma correcta, lo cual resulta más sencillo de implementar en comunicación paralela.

Por lo tanto, siendo este un medio de comunicación con características tales que nos permite reducir en forma significativa el número de conectores para realizar la comunicación, un costo de implementación más económico y sin sacrificar la cantidad de bits transmitidos por segundo, vemos que la comunicación serial es la preferida para ser implementada en la mayoría de aplicaciones donde hemos de transmitir información al reemplazar entonces a la comunicación paralela.

## **C. Niveles de voltaje asociados**

Dado que se estará trabajando con estados lógicos, generalmente binarios, es necesario asociar a cada estado un nivel de voltaje correspondiente para poder identificarlos de la forma correcta en el lado del receptor.

Es dependiente del número de estados lógicos que se manejen en la comunicación el número de niveles de voltaje necesarios para transmitir un código. Por lo general, si estamos en el caso de transmisión binaria serán necesarios dos estados. Siendo de nuestro interés la comunicación serial del tipo NRZI, hablaremos entonces de

dos niveles de voltaje distintos de cero donde un 1 lógico tendrá asociado un nivel de voltaje negativo y un 0 lógico uno positivo.

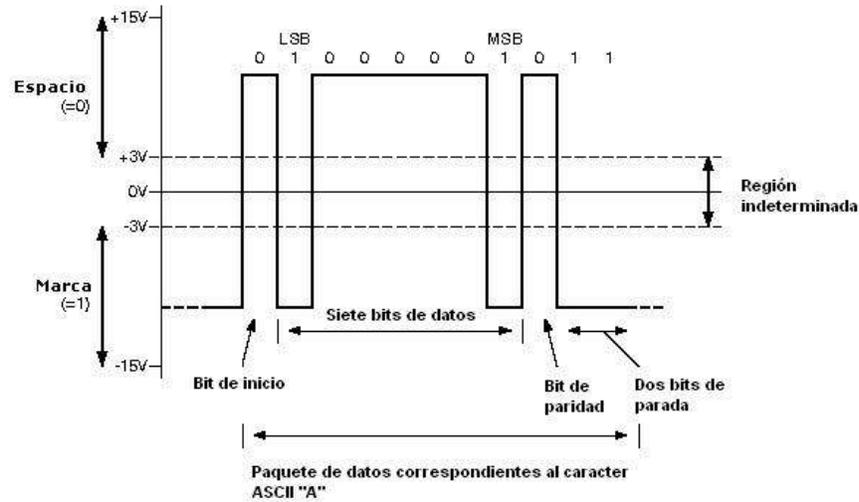


Figura 2.2: Niveles de voltaje asociados.

Estos niveles de voltajes positivos y negativos deben ser precisos y estar definidos dentro de un rango. El motivo de que sea un intervalo y no un valor específico de voltaje correspondiente al valor lógico es debido a que existen diversos factores que puedan hacer poco realizable o imposible el envío y recepción correctos de BITS de información.

## D. Baudaje y bits por segundo

Cuando estamos hablando de comunicación serial, hemos de hacer énfasis en que se estarán enviando diversos valores lógicos. Estos valores o niveles lógicos son niveles de voltajes distintos que varían en cantidad y valor de acuerdo al protocolo implementado para comunicarse.

Sin embargo, independiente de estas variaciones podemos asegurar que la señal estará cambiando de niveles de voltaje de sus líneas. A estos niveles de voltaje de la línea se les conocen como estados de la línea y serán sujeto de explicación en este punto.

Dado que la información que estamos enviando se medirá en bits, podemos establecer una relación entre el número de bits enviados por unidad de tiempo. A esto se le conoce como bits por segundo o bps y es una medida de la cantidad de información enviada por unidad de tiempo. Esto nos lleva a hacer notar que en el medio de transmisión se estarán dando estados en sus líneas de acuerdo al bit enviado al que le corresponde un nivel de voltaje. Por lo tanto, el medio de transmisión tendrá una cantidad definida de estados en sus líneas de acuerdo a los bits por segundo.

Al número de estados de línea por unidad de tiempo se le conoce bajo el nombre de baudios. Esta unidad indica únicamente la cantidad de estados que habrá en la línea por unidad de tiempo.

Ahora bien, aunque el cambio de estados en la línea bien puede representar la cantidad de bits enviados, no siempre se cumple esta afirmación. Sucede que al utilizar algoritmos de compresión y codificación, se deja de cumplir el hecho de que un estado en la línea represente un bit de información. Más bien, representa un código que tiene, como función ejemplificar un conjunto de bits en un número determinado de estados y por lo tanto, los bps son distintos del baudaje.

Comúnmente, para baudajes relativamente bajos, ambos términos resultan ser iguales entre sí. Numéricamente hablando, esto se cumple para baudajes menores a 2400 baudios que es el baudaje máximo para la telefonía. Dado que se desea transmitir a una velocidad mayor, necesitaremos comprimir y codificar dichos datos para que, a una misma velocidad de transmisión, mayor información esté siendo enviada.

## **E. Comunicación síncrona y asíncrona**

Anteriormente mencionamos la necesidad de establecer un correcto parámetro de tiempo para enviar y recibir datos. Esta es la principal ventaja y desventaja de la comunicación serial. De no establecerse un convenio entre el emisor y el receptor, se estarán presentando errores en la interpretación del mensaje. Si bien es cierto, con sólo configurar el receptor para que asocie cada bit con una duración de tiempo determinada, podríamos obtener una transmisión correcta, mas no es así en la vida real donde se presentan problemas de ruido, atenuación y distorsión de la señal.

Por esto es necesario corregir dichos errores a manera de garantizar la recepción correcta de la señal a su destino. Entre las soluciones existentes, mencionamos dos tipos de formas para la comunicación serial que resuelven el problema del manejo de tiempos adecuados.

La primera forma es la comunicación síncrona y recibe este nombre debido a que utiliza un mismo reloj para el emisor y el receptor que dicta el comportamiento temporal de la señal para garantizar que el mensaje sea enviado con los tiempos correctos y el receptor interprete cada bit en el momento adecuado. Para hacer esto, se envían constantemente datos que vienen a ser la señal de reloj que mantendrá el sincronismo entre ambas partes a comunicarse. Aun cuando no se envían señales de información o datos, el canal se mantiene ocupado mientras se transmiten datos para la señal de reloj que, entre otras funciones, sirve para mostrar el estado del emisor hacia el receptor.

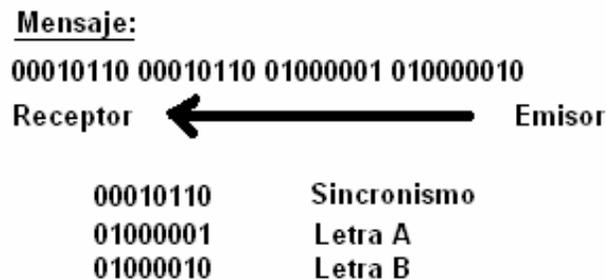


Figura 2.3: Ejemplo de comunicación síncrona

La otra forma es la comunicación asíncrona, que a diferencia de la anterior, utiliza un protocolo o enmarcado para el envío de mensajes con el objetivo de evitar el envío de mensajes de sincronismo. Este tipo de comunicación permite entonces un menor procesamiento de datos por parte de la electrónica en el lado del emisor, que se refleja en un menor consumo de potencia y por ende más económico. Sin embargo, tiene la desventaja de que, para enviar información de bastantes bits, esta resulta ser ineficiente ya que agrega bits para su transmisión tanto al inicio como al final del mensaje. Esto también muestra una velocidad menor de datos ya que, mientras se transmite un byte de información, hemos de agregar estos dos caracteres que no son de utilidad para el mensaje pero sí para el sincronismo entre el emisor y receptor.

Esto es, se envía un bit de inicio para indicar al receptor del mensaje que se va a transmitir información. Posteriormente, se envía una serie de niveles de voltaje correspondientes a bits de información que forman parte del mensaje a transmitir.

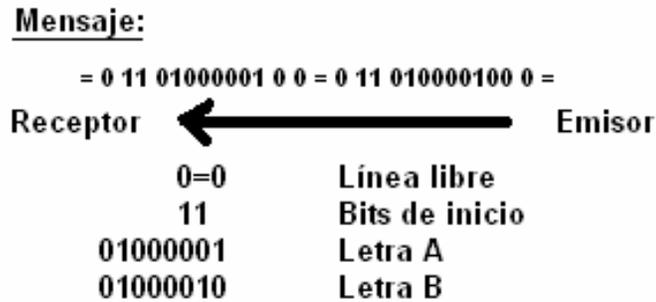


Figura 2.4: Ejemplo de comunicación síncrona

## F. Conversión de voltajes en TTL-RS232

Para definir esta conversión de voltajes, hemos de aclarar en primera instancia que un voltaje TTL se refiere al nivel de voltaje que un estado lógico tiene asociado para ser representado. Es decir, un 1 lógico en TTL está comprendido entre 2.5V y 3V y un 0 lógico es entre 0V y 0.8V. De la misma forma los niveles de voltaje en RS232-C tienen correspondencia en sus estados lógicos. Comúnmente usados, de -3V a -15V para representar un 1 lógico y de 3V a 15V para un 0 lógico, se utilizan estos rangos de voltaje para la comunicación serial. No importando el tipo de comunicación que se esté implementando ya sea síncrona o asíncrona.

El objetivo de realizar dicha conversión reside en la necesidad de comunicar dispositivos de diversos puntos entre sí a través del protocolo RS232 y por ende hay que resolver dicho problema de compatibilidad de niveles de voltaje.

Para realizar esta tarea, existe una familia de circuitos integrados de marca **Maxim** que, a un bajo costo, nos ofrece una solución a dicho problema sin tener que utilizar fuentes de alimentación con voltajes superiores a los 5V. El nombre de este chip es MAX232 y permite dicha conversión de voltajes sin importar la dirección del flujo de datos. Es decir, puede convertir de señales TTL a RS232 y recibir señales RS232 para convertirlas a TTL pero cabe notar la ventaja de que contamos con dos canales

para TTL-RS232 y otros dos para RS232-TTL. Cada canal es independiente del otro, por lo que podemos convertir diversas señales al mismo tiempo sin temor de alguna interferencia.

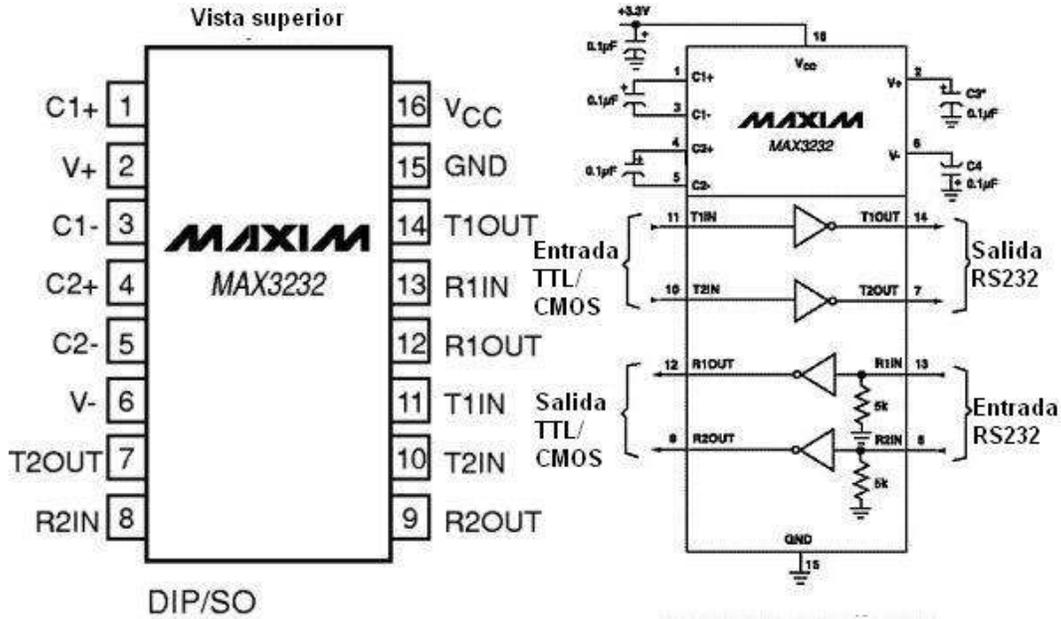


Figura 2.5: Diagrama pictográfico de convertidor MAX3232

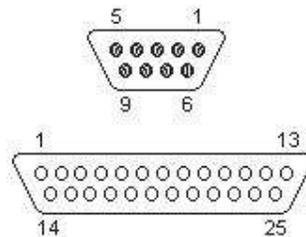
No obstante, este circuito integrado requiere de otros componentes pasivos para realizar esta tarea, pero no son más que capacitores propios para cada chip a utilizar. Con estos capacitores, obtenemos la duplicación de voltaje para llevar de 5V a 10V y también la inversión de polaridad en el voltaje para obtener -10V. Tanto con 10V y -10V representar 1 y 0 lógico para el protocolo RS232.

La ventaja de que sea solamente un circuito integrado y que su único complemento sean capacitores, permite tener a nuestro alcance la solución a las conversiones de voltaje TTL-RS232 con un precio bajo. Aún mas interesante, esta presencia de los capacitores permite que no se necesite de una fuente que entregue mayor voltaje al que ya usamos para nuestros circuitos TTL y por ende obtenemos un menor consumo de potencia en la aplicación a implementar.

## G. Tipos de conectores

El estándar RS232 no especifica el tipo de conector pero recomienda el tipo D sub-miniatura de 25 pines. Sin embargo, debido a que la mayoría de dispositivos usan pocos pines para comunicarse, se pueden usar conectores de menor tamaño como es el caso del DE-9 que ha sido estandarizado como TIA-574. Existen entonces, 9 tipos de pines comúnmente usados y para cada pin existe una función asociada de acuerdo a la aplicación a realizar. Así pues, mientras una aplicación sencilla puede usar solamente 3 pines, siendo estos RX-TX-GND, hay otras que son más complejas y utilizan otros pines para efectuar funciones sofisticadas como la detección de un mensaje en la línea, envío de señales para informar el estado del emisor, del receptor y del canal, entre otros.

**Conectores DE-9 y DE-25 para comunicación serial RS-232C**



Señal RS232	9-Pin	25-Pin
Carrier Detect	1	8
Receive Data	2	3
Transmit Data	3	2
Data Terminal Ready	4	20
System Ground	5	7
Data Set Ready	6	6
Request to Send	7	4
Clear to Send	8	5
Ring Indicator	9	22

Figura 2.6: Conectores DE-9 y DB-25

Dependiendo entonces de la aplicación a desarrollar, así será el conector necesario a utilizar. Pero cuando ya hay un dispositivo con un tipo de conector y necesitamos comunicarlo con otro de cantidad de pines diferente, se ha de usar adaptadores para resolver este problema. Dichos adaptadores son responsables de permitir la conexión correcta entre los pines que se comparten entre los conectores. Está pensado que el adaptador no agregue una impedancia significativa ni que interfiera en la señal que está transmitiendo y que simplemente sea un dispositivo que físicamente permite la unión entre los conectores, pero que para el usuario sea algo transparente a niveles de voltaje y, por lo tanto, a niveles lógicos.

### III Dispositivos HMI

#### A. Definición de HMI

Una HMI (Human Machine Interface) es la intersección entre el hombre y la máquina. Este tipo de dispositivos tiene el objetivo de facilitar el uso de tecnología que puede resolver diversos problemas aun cuando su complejidad sea grande pero que el usuario está en capacidad de manejarlo y usarlo de forma adecuada. Es decir, hacer evidente el uso y capacidades de la tecnología a la que se aplica el HMI.

Se refiere también como la interfaz del usuario con cualquier dispositivo mecánico, eléctrico o electrónico. Por lo tanto, al ser la definición bastante abierta, es de esperar que haya diversos tipos de HMI, según las aplicaciones en las que se ha implementado uno.



Figura 3.1: Ejemplo de una HMI industrial.

Con el objetivo de reforzar esta definición, se puede mencionar como ejemplo un celular. Basta con escribir un número y presionar el botón indicado para realizar una llamada telefónica. No es necesario especificar más parámetros como la frecuencia a utilizar, la modulación de la señal, ni realizar otras tareas, por lo que resulta bastante sencillo para el usuario utilizar un teléfono celular. Todas estas tareas ya son desarrolladas internamente dentro del celular y son transparentes para el usuario. Esta

interfaz del usuario con la red telefónica es simple por la información que se muestra pero también por los comandos y métodos para interactuar de parte del ser humano.

Otras interfaces no son tan simples en este sentido por el uso que tienen, generalmente industriales y esto conlleva a un desarrollo más sofisticado para que el usuario siga encontrando la interfaz lo más fácil posible. Estas nuevas consideraciones residen en el tamaño del instrumento que se desarrolla y cómo el usuario lo utilizará. Por lo tanto, resulta prudente tomar la anatomía del cuerpo humano para colocar los instrumentos al alcance del operario y mostrar la información en un lugar donde resulte visible y cómodo para el usuario. En caso sea necesario utilizar mucho espacio para la implementación de la HMI, también se tomará en cuenta la iluminación del cuarto donde se ubicarán los paneles de control. De la misma manera la posición en la que se hayan acomodado dentro del cuarto. El ángulo entre paneles es importante, ya que permite al operario tener mayor acceso a la información desplegada y a los distintos medios para interactuar con la HMI.



Figura 3.2: Armarios de control industrial según los estándares internacionales.

## B. Importancia de HMI

Muchas actividades que se desarrollan son el resultado de un conjunto de otras actividades menores que trabajan entre sí y algunas de estas pueden resultar siendo muy importantes por lo que es necesario que se ejecuten de forma correcta. Sin embargo, existe la tendencia de que, entre más sofisticada es una tarea, más fácil debe ser para el usuario poder ejecutarla. Es decir, saber cumplir con los objetivos necesarios para realizar la tarea sin complicaciones para el usuario u operario.

Por lo tanto, el problema a resolver de una tarea compleja y transformarla a una tarea sencilla queda en manos de un circuito intermediario capaz de presentar al usuario la información con un nivel de detalle suficiente como para resolver la tarea, pero sin resultar tan abrumador como para confundir al usuario.

La necesidad de que sea fácil de operar y de entender radica precisamente en la eliminación de errores humanos. Estos errores se pueden dar, entre otras cosas, por el mismo cansancio físico, fatiga mental o distracciones entre otras causas. Dichos errores deben ser eliminados ya que puede resultar en grandes desastres y costos altos ya que dependen de la tarea a ejecutar. De hecho, gran parte de las aplicaciones actuales se dedican a hacer más fácil de operar sus dispositivos para que el usuario se sienta más cómodo en la realización de una tarea. Esto se ve reflejado en las aplicaciones profesionales donde 50% del código de programación es para facilitar el uso del dispositivo según el International Engineering Consortium (IEC).



The image shows a control panel with a blue background. At the top left, there is a red button labeled "Actualización". Below it, there are three input fields with labels: "Freq mínima" (value 15), "Freq máxima" (value 40), and "Freq de salto" (value 30). The values are displayed in a white box on the left side of each input field.

Parameter	Value
Freq mínima	15
Freq máxima	40
Freq de salto	30

Figura 3.3: Aplicación diseñada para la modificación de parámetros de forma simultánea. Los valores mostrados están en Hertz.

Sin una HMI implementada en un proceso complejo, se corre el riesgo de cometer errores debidos al usuario por diversas causas, al reducir la productividad y aumentar costos. Los errores que se puedan cometer se deberán en su mayoría al operario que maneje la HMI y ser causados por razones que involucran desde la poca claridad, para mostrar el flujo de control en la planta, hasta fatiga o mal manejo del instrumento por parte del operario. Dichas razones son la causa por la cual se decidió estudiar el cuerpo humano para entender datos que se asumen entendidos pero que son de vital importancia para el éxito o fracaso del proceso, como sería el alcance del brazo tanto vertical como horizontalmente, el espacio visual que se percibe y las variaciones del cuerpo humano cuando está en diversas posiciones.

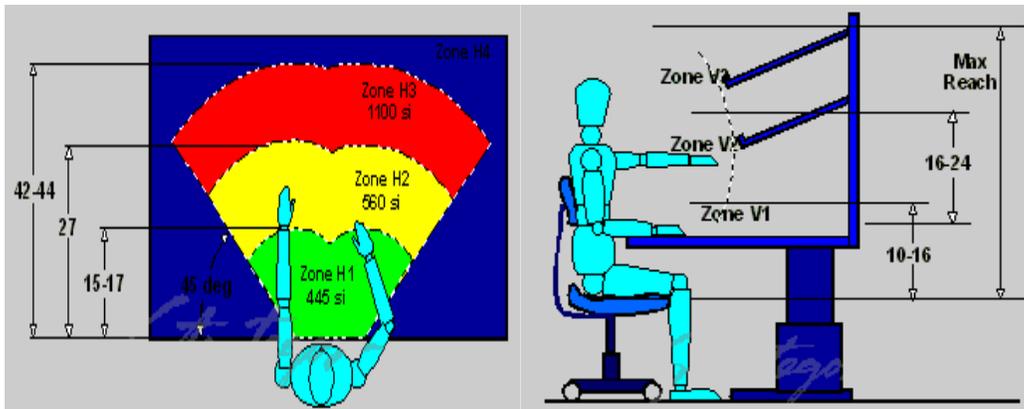


Figura 3.4: Alcances del cuerpo humano, dimensiones en pulgadas.

Esto implica que la ergonomía se aplica también para una HMI dada la necesidad de reducción de errores causados por la fatiga o prevenir lesiones a largo plazo causados por el uso constante de la interfaz. Existen diagramas en conjunto con tablas numéricas creadas para ejemplificar un promedio de dimensiones que el cuerpo humano tiene. Con estos números se puede dar una dimensión aproximada de la interfaz a realizar siendo sujeta a cambios menores debidos a la población del lugar a instalar la HMI y del propio operario que estará a cargo.

## C. HMI virtuales

Si bien es cierto la gran mayoría de interfaces son paneles de control para la industria y dispositivos electrónicos de pequeñas dimensiones para la vida cotidiana, también existen aquellos que se apoyan en las herramientas de software para presentar en la pantalla de una computadora un proceso industrial que se esté llevando a cabo.

La razón por la cual se utiliza esta interfaz y no un panel de control es por la forma gráfica explícita en la que se presentan los resultados, ya que permite al operario controlar y monitorear el proceso desde una sola interfaz.

Es por esto que en la industria se están utilizando ambos tipos de HMI, ya que algunos equipos requieren de grandes cuartos de control con paneles de control dependiendo de qué tan modernos sean o pantallas de cristal líquido (LCD) donde se mostrará una imagen generada por herramientas de software en la que se podrá interactuar por medio de distintos periféricos que pueden ser desde un teclado hasta el contacto directo con la pantalla, si esta lo permite. No es de sorprenderse saber que con el constante empuje de la computación a lo largo de los años recientes, las HMI se hayan desarrollado bastante, sobre todo en el lado de HMI virtuales haciéndolas más simples y entendibles.

Para ello, este diseño debe representar con un buen detalle cada paso que existe con una serie de variables que permitan la realización correcta del procedimiento. Su forma de presentarse en la pantalla debe ser tal, que se asemeje a la planta montada en la vida real con cada instrumento y componente que esta contenga. Pero no se trata solamente de hacer un laberinto de tuberías, actuadores, fajas etc. Dicho procedimiento debe llevar un orden de izquierda a derecha y de arriba hacia abajo. Esta estructuración corresponde a la facilidad del ser humano para interpretar información mostrada por ser similar al movimiento que los ojos tienen en la lectura de un texto. Aunque esta forma de representar el proceso facilitará la adquisición de datos del proceso, debe tomarse en cuenta que la HMI también debe ser propicia para que el usuario pueda alterar datos en el procedimiento de manera simple. Estas modificaciones deben estar disponibles al usuario y estarán agrupadas de acuerdo al segmento del proceso que se desee modificar o por la similitud con otros parámetros entre sí.

From Computer Desktop Encyclopedia  
Reproduced with permission.  
© 2008 Opto 22  
From Computer Desktop Encyclopedia  
Reproduced with permission.  
© 2008 Opto 22

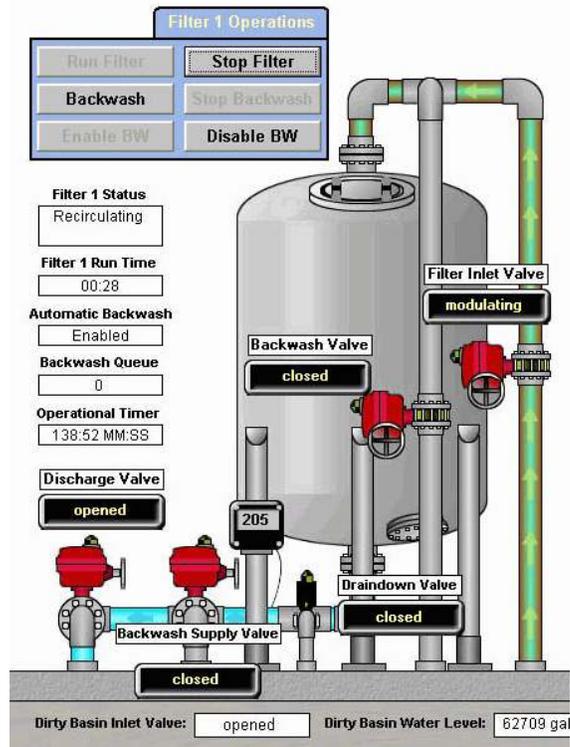


Figura 3.5: Ejemplo de una HMI virtual para el control de tanques y válvulas.

La forma en la que el usuario esté interactuando o modificando valores de un procedimiento será de suma importancia para que se eviten errores en la ejecución por haber ingresado mal un dato. Debe considerarse si el parámetro a cambiar tendrá valores discretos o continuos para luego asignarle un control adecuado a la variable que permita modificaciones adecuadas a ella. Un ejemplo claro consiste en el control que se utilice para encender un motor y la especificación de una temperatura específica de un horno. No sería lógico utilizar un botón de encendido y apagado para controlar temperaturas porque se limitaría a dos posibles valores siendo entonces un control muy pobre. Y tampoco sería adecuado colocar barras de empuje o controles de perilla que tienen una variedad casi infinita de posibles valores a asignar si lo que se pretende es sólo encender o apagar el motor. La manera en que el operario especifique un valor para sus modificaciones debe ser clara y con retroalimentación, en el sentido de que el operario sepa exactamente qué valores está ingresando, muy útil para casos donde la precisión es fundamental.

Un ejemplo de una HMI virtual es la aplicación a futuro que este proyecto puede desarrollar. Donde se controlan de forma continua varios equipos de la línea de producción, fábrica o planta de procesos se ha hecho uso de los requerimientos de una HMI virtual antes mencionados. Cada parte del proceso tiene un color asociado para representar su estado de funcionamiento, y muestra una cantidad mínima de variables para mantener un despliegue liviano de la pantalla. Sin embargo, de ser necesario un nivel de detalle mayor para el monitoreo del equipo, como sería un historial de consumo de energía, basta con seleccionar el componente y presionar dos veces para abrir una nueva pantalla con la información detallada.

Aunque ya había existencia de estos paneles gráficos en el pasado, estos eran bastante rígidos en su estructura por ser fabricados con piezas plásticas de color y lámparas incandescentes para representar lo que actualmente se hace con un indicador en la pantalla. Sin embargo, no es sino hasta el día de hoy que se cuenta con pantallas de LCD, monitores y otros indicadores que se ha dado mayor importancia a las HMI virtuales, por la facilidad de instalación y precios para implementar en la industria.

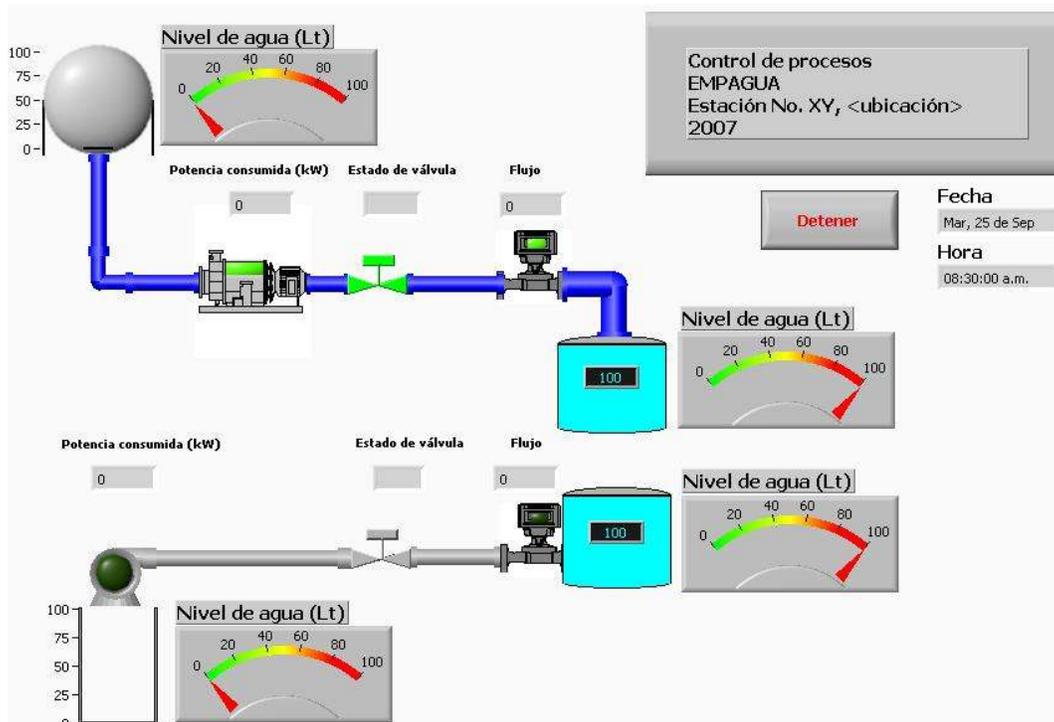


Figura 3.6: Panel gráfico de control para estación de agua

Todos estos puntos resultan de suma importancia, aunque no existen reglas específicas para el desarrollo de una HMI en general, si se puede adaptar una existente con estos requerimientos ya descritos para la función a realizar. Es cuestión de quien desarrolla la HMI y del operario para estar de acuerdo en algunos ajustes finales para que la HMI se apegue lo más posible a la forma ideal de control del proceso. Los cambios a realizar en una HMI virtual serán más sencillos de hacer dado que es por medio de software y no por medio de indicadores lumínicos, análogos, botones o palancas dentro de un panel de control que contemplen modificaciones de estructura física. Esto presenta una reducción del tiempo para implementar cambios respecto del panel de control y reducción de costos para quien realiza estas modificaciones, porque no tendrá que comprar nuevas piezas si desea cumplir con los cambios mencionados.

## **D. Estándares internacionales**

Si existe la necesidad de garantizar HMI más eficientes, es necesario entonces establecer reglas y requisitos que sean reconocidos internacionalmente para mantener un estándar mundial de estas interfaces.

Así entonces, existen organizaciones encargadas de establecer parámetros como los antes mencionados. Claro está, algunas discrepan de otras entre compañías pero van dirigidas hacia el mismo fin, que es fabricar HMI más indicados para el usuario.

Entre estas organizaciones, podemos mencionar a International Standards Organization (ISO) que establece varios puntos como el ISO 9241, diseñado para todos los requerimientos ergonómicos en el trabajo de oficina con despliegues visuales. Este punto incluye, entre otras, teclados, espacios de trabajo necesarios hasta diseño de menús. ISO14915 es otro punto que se enfoca más en multimedia y cubre todo aquello relacionado a la navegación del usuario por el panel de control.

Existen muchos otros puntos acerca de estándares para las HMI pero ninguna se contradice, más bien se complementan para cubrir de mejor manera el diseño de estos dispositivos. Se menciona también al American National Standard Institute (ANSI) que se dedica en cierta parte a la ergonomía del software en las interfaces, pero en su mayoría se concentra a los estándares de informática, en el área de programación y de las telecomunicaciones. Tal es el caso del Information Infrastructure Standards Panel

(IISP), desarrollado bajo ANSI para acelerar el desarrollo de un estándar a nivel mundial que sería crucial para la infraestructura informática de los Estados Unidos. Para el área de la Internet, existe la World Wide Web Consortium (W3C), responsable de definir estándares específicos para la red que incluye el formato del lenguaje de hipertexto de HTML.

Como ejemplo de estándares de ISO y ANSI se mencionan los siguientes:

- ISO 9241 "Ergonomic Requirements for Office Work with Visual Display Terminals." Este involucra diversas áreas desde teclados hasta requerimientos en los lugares de trabajo. Maneja también el diseño de un menú y el uso de los colores según la aplicación deseada.
- ISO 14915 es un estándar para multimedia, contempla el diseño de controles y navegación.
- ISO-IEG 11581 "Graphical Symbols on Screens".
- ISO-IEG 13714 "User Interface to Telephone-Based Services—Voice Messaging Applications".
- ISO-IEG 11580 "Names and Descriptions of Objects and Actions Commonly Used in the Office Environment".
- ANSI/HFES 200 "Ergonomic Requirements for Software User Interfaces"  
Es una extensión del ISO 9241, e incluye un capítulo con guías para hacer interfaces de usuarios para gente con diversas discapacidades.
- “i.e. Información Infrastructure Standards Panel” (IISP), también creado por ANSI fue elaborado para acelerar el desarrollo de estándares que serían cruciales para la construcción de la infraestructura de informática de Estados Unidos de Norteamérica.
- ANSI T1M1.5 es una contribución técnica para la arquitectura de telecomunicaciones, interfaces y protocolos para la ITU (International Telecommunications Union).

## IV. Motores trifásicos asíncronos

### A. Inducción electromagnética

Cuando una corriente eléctrica atraviesa un conductor, este tiene asociado un campo magnético que gira alrededor de la corriente. Este campo magnético es proporcional a la corriente del conductor por lo que al aumentar o disminuir la corriente, también lo hace la intensidad del campo. Un fenómeno físico que resulta esencial para entender el comportamiento de un motor eléctrico, en particular, uno de inducción o asíncrono. Las ecuaciones de Maxwell nos explican que, este campo magnético se rige bajo el convenio dextrógiro o regla de la mano derecha.

Sin embargo, no sólo una corriente produce un campo magnético, también se da el fenómeno recíproco, que consiste en la inducción de una corriente dado un campo magnético que rodea a un conductor. A esto se le conoce como la Ley de Faraday.

En ambos casos, tanto uno que induce a otro, la dirección que tienen es dependiente de la dirección del otro fenómeno físico. Como se explicó antes, el convenio dextrógiro o la regla de la mano derecha determina tanto el flujo de la corriente o la dirección del campo magnético

Ahora bien, dependiendo de la forma del conductor, los campos magnéticos pueden dar origen a los electroimanes, que tienen el mismo comportamiento que un imán natural pero que se puede encender o apagar para regular su comportamiento magnético. Como es de esperarse, también tiene dos polos y una intensidad de campo asociada.

Para obtener un electroimán debemos construir una espira o alambre bobinado alrededor de un eje o núcleo que suele ser no ferromagnético. Esta forma del conductor, que tendrá una corriente que fluye por el, permite que el campo magnético que gira alrededor del conductor forme un bipolo magnético.

Siendo más específicos para nuestro motor trifásico, el campo magnético desarrollado dentro de él es constante para cualquier instante del tiempo. Aunque en dirección no lo es debido a la rotación que este tiene según el comportamiento de las bobinas, sí lo es la magnitud del campo. Esto es muy importante para su funcionamiento correcto y una rotación suave sin fluctuaciones en el torque y velocidad.

## B. Modelo eléctrico

Siendo de nuestro interés el estudio de motores trifásicos asíncronos o de inducción, hemos de estudiar su construcción interna, modelarlo como un circuito eléctrico y establecer ecuaciones para entender su funcionamiento.

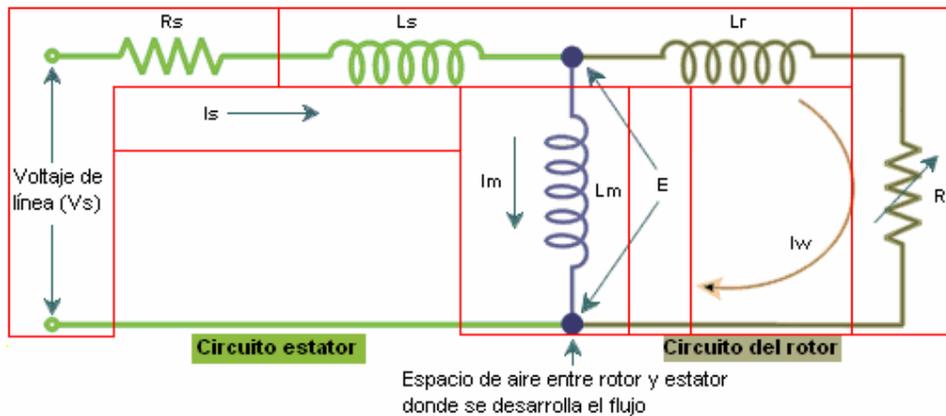


Figura 4.1: modelo eléctrico por fase de motor trifásico de inducción.

Es de gran importancia, analizar el motor por cada una de sus fases, sin embargo, debido a que todas sus fases son idénticas entre sí, nos concentraremos en el estudio de una sola fase y por ende, desarrollaremos un modelo para dicha fase que se aplica al resto de las que le acompañan.

Como primer punto de este circuito, podemos ver que existe un voltaje de línea  $V_s$ , que es la alimentación del motor. Este voltaje estará con las especificaciones requeridas por el motor y se tomará como punto de referencia la envoltura del motor o chasis. Debido a que este voltaje alimentará las bobinas del estator, se coloca entonces un resistor denominado  $R_s$  en serie a una bobina  $L_s$  que modelan la impedancia característica de la bobina, tanto por su resistencia interna debida al cable como a la inductancia asociada de la bobina. Por estos dos componentes atraviesa una corriente denominada  $I_s$  que se deriva en dos corrientes que se explican a continuación. Posteriormente, modelamos la inductancia que representa la inducción del campo magnético del estator hacia el rotor, para esta inducción, es necesaria una corriente  $I_m$  y para ello un voltaje entre los puntos donde atraviesa la corriente. Esta corriente es,

comúnmente, de 30% de la corriente  $I_s$  y al igual que el flujo magnético, esta es proporcional al voltaje  $E$  y a la frecuencia. Se asume en este diagrama que no existen pérdidas en el núcleo. Sin embargo, si se desea tener un modelo más real, debe considerarse esta pérdida al colocar un resistor en paralelo a la bobina.

Ahora bien, el circuito del rotor está explicado también por una bobina y un resistor en serie. Este resistor tiene una resistencia variable de acuerdo a la temperatura: a mayor temperatura, mayor resistencia. Aumentar la resistencia del rotor conlleva a una reducción de la corriente  $I_w$ , que es la responsable de producir el torque del motor, que a su vez es directamente proporcional a la carga que tenga conectada a su eje.

Es importante mencionar que para obtener la máxima potencia en este último segmento del circuito, debe cumplirse el teorema de la máxima transferencia de potencia, esto es, la impedancia de la bobina del rotor debe ser igual a la resistencia del rotor y con ello, el torque será máximo.

### **C. Velocidad de sincronismo y rotación**

Un motor de inducción funciona por medio del fenómeno de inducción electromagnética y es a través de la atracción y repulsión de polos que se da la rotación del eje del rotor dentro del estator.

De aquí podemos observar que existen dos velocidades distintas dentro del motor funcionando al mismo tiempo. Una es la velocidad de sincronismo, resultante de la velocidad de polarización de las bobinas dentro del estator. La otra velocidad es la de rotación, representante de la velocidad a la que el eje del rotor está girando.

Ambas velocidades, como es de esperarse, son distintas y en cierto caso, iguales. Para entender mejor esto, hay que recordar que la velocidad de rotación es debida a la atracción y repulsión de polos magnéticos. Esta velocidad es dependiente precisamente de estas fuerzas que atraen y repelen al polo que a su vez son causadas por el estator y sus cambios de polaridad entre sus bobinas. Por lo tanto, la velocidad de sincronismo es mayor que la de rotación y será siempre mayor o igual a la de rotación. En ningún momento puede ser mayor la de rotación a la de sincronismo, esto implicaría innegablemente la presencia de una fuerza externa que acompañara al eje en la misma dirección de rotación para que girara aún más rápido de lo normal. Cuando esta

condición se presenta, el motor deja de funcionar como un dispositivo pasivo que transforma energía eléctrica a mecánica sino más bien, transformaría la energía mecánica de rotación del eje del rotor a energía eléctrica por la inducción del campo magnético sobre las bobinas del estator y por lo tanto, una corriente de retorno del motor hacia el resto del sistema eléctrico que lo acompaña.

La modificación de dichas velocidades tiene asociado al torque que el motor puede ejercer. Dada la ecuación  $E = K \frac{d\phi}{dt}$  puede observarse que el voltaje  $E$  es dependiente del flujo  $\Phi$  que atraviesa las espiras por unidad de tiempo  $t$  del estator y según la ecuación  $T = k\Phi I \omega$  podemos observar la importancia del voltaje en el torque resultante. De darse velocidades iguales entre el sincronismo y la rotación, no habría flujo atravesando las espiras del rotor, esto a su vez afecta directamente al torque que es proporcional a este flujo. Por lo tanto, una variación en el flujo debido a la variación de la diferencia entre la velocidad de rotación y sincronismo, afecta directamente al torque que este genere.

## V. Variadores de frecuencia

### A. Definición de variador de frecuencia

Es un dispositivo electrónico para el control de velocidad de motores de corriente alterna que, haciendo una conversión del voltaje y frecuencia introducidos, permite la salida de señales con parámetros variables.

Estas señales de entrada deben ser de corriente alterna, generalmente de 110V a 120V en amplitud y entre 50Hz a 60Hz de frecuencia de forma sinusoidal. Dependiendo del fabricante, puede requerir tanto 3 líneas de este tipo provenientes de alimentación trifásica o una sola línea de 220V a 240V en amplitud con la misma frecuencia y forma ya descrita. La salida está conformada por una serie de pulsos rectangulares que simulan el comportamiento de señales sinusoidales desfasadas  $120^\circ$  entre sí para alimentar al motor trifásico.

### B. Construcción interna

Para realizar esto se llevan a cabo pasos de rectificación de señales AC a DC y de vuelta a AC con los parámetros deseados tanto en amplitud como en frecuencia. Este método de regulación de velocidad es bastante común en la industria de estos días.

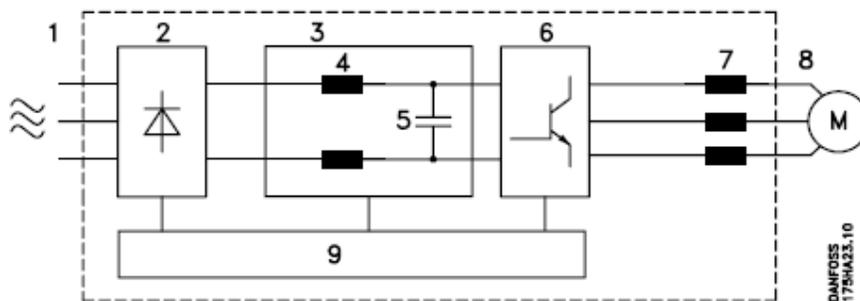


Figura 5.1: Diagrama de bloques de construcción interna de un variador de frecuencias, según manual de usuario.

Esta herramienta se puede considerar según varios bloques funcionales conectados entre sí que realizan una tarea distinta para que en conjunto lleven a la realidad dicho variador de frecuencias.

El primer segmento consiste en la alimentación de señales de corriente alterna. Dependiendo del variador de frecuencias, así será el número de señales que este recibirá, ya que puede ser alimentación monofásica o trifásica. Para el caso del variador de frecuencia Danfoss VLT 2020, es indistinto qué tipo de alimentación se introduzca, ya que está configurado para aceptar tanto alimentación de una fuente monofásica o en su defecto trifásica. Es a partir de estas señales que, teniendo asociado un voltaje pico, una frecuencia y un valor promedio, se genera la variación de frecuencia de tres fases que alimentarán al motor trifásico. Primero debe rectificarse el tipo de señal que se recibe, es decir, convertirlo a un valor DC correspondiente al seno que entra al sistema. Esto se logra por medio de un arreglo de diodos que va a depender del fabricante en lo que respecta a su arreglo y configuración. Lo que estos componentes pretenden realizar es la rectificación de la señal para obtener solamente valores positivos sinusoidales. Ya que existe una variedad de rectificadores que involucran desde transformadores hasta un solo diodo, se puede implementar este bloque funcional de diversas maneras tomando en cuenta los costos monetarios y pérdidas energéticas asociadas.

Luego, se filtra por medio de un arreglo de capacitores y resistores que tienen como objetivo mantener una señal DC constante muy parecida al valor pico de la señal rectificada que entra a esta parte del circuito. Como se observa en el diagrama, la configuración de los componentes anteriormente mencionados es precisamente la de un filtro pasa altas. Dado que la impedancia del capacitor aumenta conforme disminuye la corriente, entonces al ser conectado a una señal DC, este componente presenta una impedancia bastante alta, similar a un circuito abierto por lo que se obtiene una medición muy similar a la que recibió de entrada pero sin las pequeñas fluctuaciones que le acompañaban.

De esta señal de amplitud constante, vamos a alimentar a los transistores IGBT conectando el voltaje DC a los colectores del transistor. El emisor va conectado a tierra y la base a un circuito, generalmente un microcontrolador, que tendrá la responsabilidad de permitir la conmutación entre las regiones de corte y saturación del transistor. Esto con el objetivo de ver en la salida, conectada al colector del transistor, un voltaje que puede variar entre VCC que viene a ser el voltaje DC de los colectores y 0V, que es el caso cuando se alcanza la región de saturación dando una conexión de nuestra salida a

tierra. Los transistores juegan el papel de interruptores eléctricos que enviarán una señal rectangular en sus salidas con una amplitud de VCC, frecuencia y ciclo de trabajo controlados por el microcontrolador interno. Esto se realiza tres veces de forma paralela con el objetivo de tener tres señales con este mismo valor de voltaje que estarán desfasadas 120 grados y así obtener la alimentación trifásica del motor.

Finalmente, estas señales cuadradas alimentan a las bobinas del motor trifásico que empieza a funcionar como si se le hubiera conectado una alimentación trifásica sinusoidal.

### **C. PWM y señales sinusoidales**

El motivo por el cual funciona es porque el circuito equivalente para el motor trifásico presenta un circuito pasa bajas en su interior. Para entenderlo mejor, observe que las bobinas del motor son modeladas como un resistor en serie con un inductor donde el resistor representa la resistencia interna del material conductor y la bobina o inductor representa la inductancia asociada que permite la existencia de campos magnéticos en el estator. Así, entonces, al alimentar a este filtro pasa bajas con una señal cuadrada de anchos de pulso variable y polaridad cambiante, obtenemos una representación muy parecida a la señal sinusoidal que estaría entregando una alimentación trifásica.

Sin embargo, este dispositivo no pretende simular solamente una fuente de alimentación trifásica. Más bien, un controlador de señales trifásicas que permite la variación de velocidad del motor a través del parámetro de la frecuencia de la señal que se está enviando. Así pues, la conmutación de la señal cuadrada y la frecuencia a la que realiza dichas tareas nos permite regular la velocidad del motor sin pérdidas adicionales en el sistema.

## **D. Importancia de la variación de frecuencia**

Sin duda alguna existen diversos métodos para controlar la velocidad de un motor trifásico de inducción. Entre ellos, contamos con la variación de la resistencia interna del rotor, que repercute directamente en el voltaje sobre las barras del rotor y el torque asociado. También con la variación del número de polos, ya que al depender de la cantidad de polos en funcionamiento, así será la velocidad de sincronismo que es inversamente proporcional al número de polos. Finalmente el control del deslizamiento, un método que se basa en el flujo que atraviesa a las espiras de cada rotor para definir así la velocidad de sincronismo y rotación pero siempre tomando en cuenta en qué punto de operación se encuentra el motor.

No obstante, todos los métodos anteriormente mencionados no son más que formas de desviar toda la energía disponible hacia otros puntos ajenos a la rotación. Es decir, agregando pérdidas adicionales al sistema para que la velocidad sea menor y por ende no reduzca la cantidad de energía consumida al disminuir su velocidad.

La variación de frecuencia en el motor es distinta, principalmente porque la modificación de este parámetro no agrega pérdidas adicionales, mas bien, se beneficia de la relación de voltaje y frecuencia para un torque constante. Es, entonces, un método que al exigirle menos al motor, presenta un menor consumo de energía y por ende más económico. Finalmente, podemos hacer funcionar motores trifásicos sin contar con la estructura para este tipo de alimentación, dependiendo del fabricante, el mismo variador funcionará con una sola línea de 220V AC.

A niveles industriales, es bastante común encontrar esta técnica de control de velocidad por las ventajas anteriormente mencionadas, siendo entonces una solución económica y rentable para la industria.

Para el caso del variador de frecuencias utilizado, Danfoss VLT 2020, esta regulación se realiza por medio de diversos procesos en serie que convierten el voltaje y la frecuencia de alimentación de carácter constante a valores variables. Con un voltaje de alimentación que puede ser monofásico de 220V a 50-60Hz hasta trifásicos de 460V a 50-60Hz, poner en funcionamiento un dispositivo de estos resulta ser más flexible si es comparado con aquellos que solamente pueden funcionar con alimentación trifásica a voltajes y frecuencias constantes.

Este variador de frecuencias puede entregar, en sus salidas, corriente alterna correspondiente al 10-100% del voltaje de alimentación, con una frecuencia variable entre 0-500Hz. Para un motor de 0.75KW, este variador resulta idóneo dada la potencia máxima de salida que corresponde precisamente a 0.75KW. Haciendo uso de tecnología conocida como “Voltage Vector Control” (VVC) Danfoss entrega un variador de frecuencias superior a otros que utilizan “Pulse Width Modulation” (PWM) tradicional para regular la velocidad de un motor y a esto hay que agregar que se reducen las pérdidas del inversor hasta en 30%, comparado con otros variadores de frecuencias. Para entenderlo mejor, las siguientes figuras tomadas del manual del variador de frecuencias comparan el voltaje que se entrega al motor y el patrón de conmutación simplificado tanto por VVC como PWM. Para ambas gráficas los rectángulos representan el patrón de conmutación por fase, en este caso la fase U, la línea gris oscura corresponde al voltaje por fase, y la línea gris clara al voltaje de fase a fase.

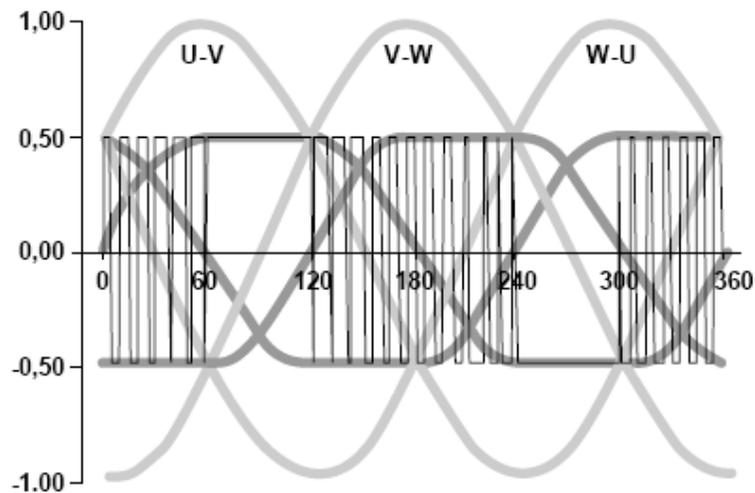


Figura 5.2: Gráfica del patrón simplificado de conmutación del principio VVC y voltaje del motor.

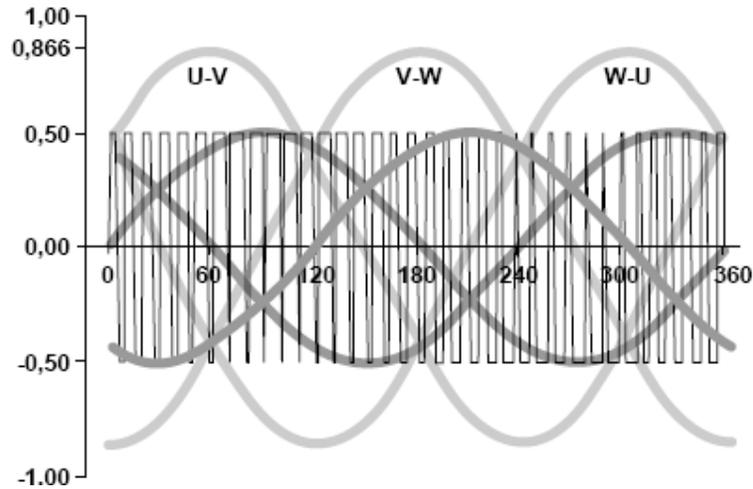


Figura 5.3: Gráfica del patrón simplificado de conmutación de PWM tradicional y voltaje del motor.

La programación de los parámetros se realiza a través de una pantalla de cristal líquido y teclas disponibles con funciones predeterminadas, en algunos casos. La modificación de los valores es posible debido al microprocesador que lleva dentro, así como otros módulos como la memoria EPROM, EEPROM y RAM. Como lo muestra la siguiente figura, tomada del manual del VLT 2020, una serie de bloques interconectados permite la regulación del voltaje de alimentación a salidas con voltajes y frecuencias correspondientes a variables que el usuario ha especificado.

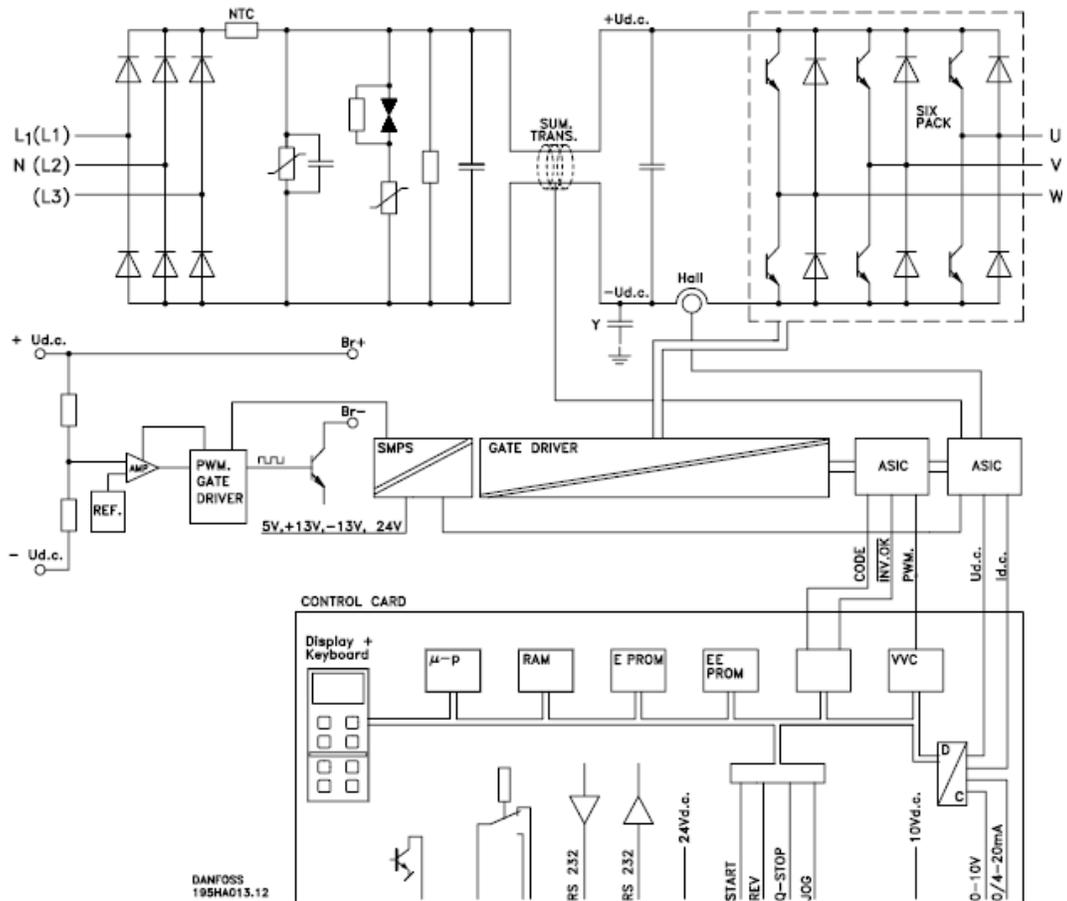


Figura 5.4: Segmento del diagrama de bloques del variador de frecuencias.

## VI. Programación en LabVIEW

### A. Tipo de programación

LabVIEW es un lenguaje de programación creado por la empresa norteamericana National Instruments. Es un lenguaje gráfico, casi único, en el método en el que el código es construido y guardado. No existe tal cosa como código de texto, más bien una visión de un diagrama para indicar cómo fluyen los datos a través del programa y sus diversos módulos que lo componen. Esta herramienta es de gran preferencia por ingenieros y otras personas que prefieren ver el flujo de datos, en lugar de un programa con código en texto como la programación convencional.

A estos programas se les llama instrumentos virtuales, que constan de tres partes fundamentales siendo estas: el panel de control, el diagrama de bloques y la interconexión entre estos bloques.

Este flujo de datos es fundamental en lo que respecta a LabVIEW y su forma de escribir código. Se maneja la filosofía de que el paso de los datos por los módulos del programa define el orden de ejecución de esta cadena de eventos que son el funcionamiento y ejecución de dicho programa. Diversos módulos o bloques funcionales tienen una serie de entradas y salidas que, conectadas entre sí, pueden realizar diversas funciones, según sea el acomodo que defina el programador de estas piezas.

Se tiene a disposición una gran gama de herramientas que incluyen controles análogos, digitales para el manejo de datos así como diversos métodos para mostrar resultados que pueden ser ya sea de forma numérica o gráfica. También se incluyen herramientas para el análisis de señales, manejo de procesos de control y comunicación con el exterior por medio de Hardware especializado en la adquisición de datos, así como en el envío de señales análogas y digitales de baja potencia a través de este mismo modo.

LabVIEW nos da en conjunto una serie de opciones para resolver de manera práctica y rápida los problemas de la vida real, con la ventaja de implementar estas aplicaciones con relativa facilidad, ya que además se cuenta con herramientas para el

análisis profundo del flujo de datos y entender a detalle cómo se están desarrollando las tareas.

Un ejemplo de esta programación se puede ver en la siguiente figura: este programa permite controlar el llenado de un tanque a un determinado nivel. El control de llenado del tanque se conecta hacia el indicador gráfico del tanque, este es consultado por medio de una variable local que lleva el mismo nombre. La razón de leer la variable del tanque es para compararla contra un nivel de referencia y determinar si se ha alcanzado este nivel en el tanque. De ser así, la ejecución del programa se suspenderá dado que la desigualdad ya no se cumple.

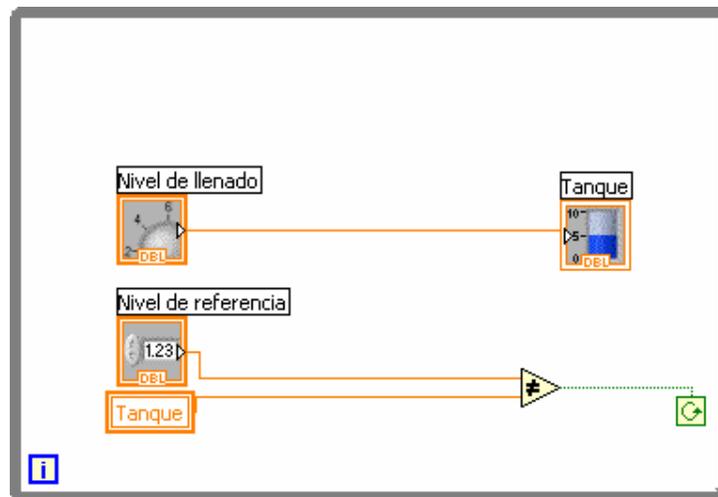


Figura 6.1: Ejemplo de un diagrama de bloques en un instrumento virtual (VI)

## B. Configuración de puerto serial

La configuración inicial de la comunicación serial requiere de un módulo especial para desarrollarse. Este bloque maneja parámetros esenciales para establecer la comunicación como se necesita, y resultan ser comunes para la mayoría de usuarios.

Se asignan valores constantes a algunos parámetros de entrada como es el carácter de terminación del mensaje, y de aceptar esta terminación, hemos de especificar el tipo de carácter que reconocerá LabVIEW. En caso de detectar comunicaciones erradas por inactividad, también escribiremos cuánto tiempo en milisegundos será necesario para enviar la señal de dicho error. Se escribe el nombre del puerto a utilizar que

normalmente es COM1 para RS-232C. Luego se especifica el baudaje a utilizar, seguido del número de bits del dato en el mensaje. Se especifica si se desea utilizar paridad par, impar o ninguna para establecer una forma liviana para detección de errores en el mensaje. El número de bits de parada se puede establecer entre 3 opciones, 1 bit, 1.5 bits o 2 bits. Finalmente, el parámetro del control de flujo para manejar el envío de datos por el canal y esto se utiliza, entre otros casos, cuando el buffer de información no es de un tamaño grande o suficiente para el manejo de los datos y se necesita regular cuando se envía información de acuerdo al llenado del buffer.

Dada esta gran gama de opciones para configurar comunicación serial, no hay problemas para establecer una comunicación serial de primera con las características y requisitos necesarios de acuerdo al medio y la forma en la que estemos transmitiendo el mensaje.

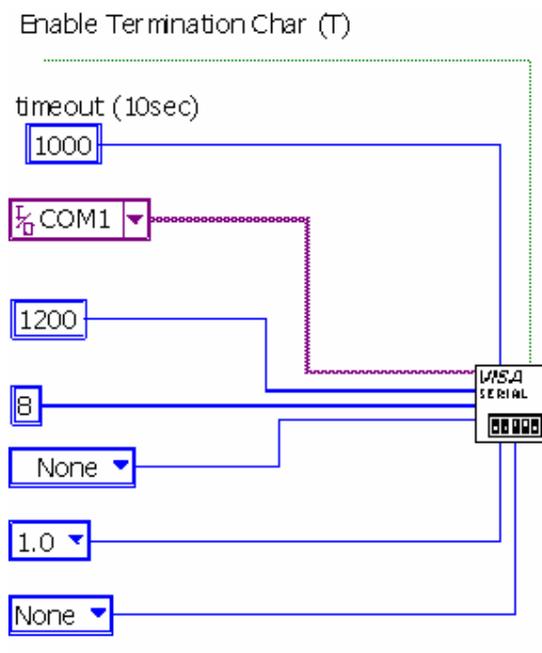


Figura 6.2: Bloque para configurar comunicación serial y algunos parámetros.

De acuerdo a la figura anterior, se puede observar cómo diversas constantes se conectan a un solo bloque de configuración serial. Al ser mencionados en orden descendente según su aparición en la figura, se especifica la existencia de un carácter de terminación del mensaje, un tiempo máximo de espera para la recepción de mensajes antes de mostrar un error, el nombre del puerto al cual se hace referencia para escribir los mensajes, el baudaje que determinará la velocidad de transmisión del mensaje, la

cantidad de bits de datos que conformarán al mensaje. También se puede especificar el tipo de paridad deseada en la comunicación a realizar, luego se especifica la duración del bit de parada y finalmente el tipo de control, ya sea por hardware, software o ninguno.

## **C. Construcción de cadenas de caracteres**

LabVIEW posee muchas herramientas para enviar información y manejar el flujo de datos por el código del programa. Entre estas herramientas contamos con los caracteres y cadenas que se pueden formar entre sí. Debido a que varios bloques o módulos requieren entre sus parámetros cadenas de caracteres, es de suponer que existan herramientas para el manejo de estos objetos, tanto para la adquisición, envío y manipulación por medio de cadenas de caracteres y texto.

Resulta sencillo construir una constante de este tipo en el diagrama de bloques, la cual inicia vacía y resulta posible escribir para asignarle un valor deseado. No obstante, este valor no es invariable, más bien, como se mostrará posteriormente, podemos hacer una serie de procesos con esta cadena inicial de caracteres para adaptarla al requisito que el programador tenga en mente.

A esta operación se le conoce como construcción de cadenas de caracteres y no está restringida a ser construida solamente con caracteres, sino también con cadenas de diversos largos para que se concatenen y formen una sola cadena de una longitud mayor.

El orden en el que se adjunten estos segmentos de la cadena a ser formada puede ser definido por el programador. No es necesario que una cadena tenga que ser construida para obtener una nueva cadena, más bien, podemos tomar una ya existente y concatenarle en cualquier posición de la cadena un nuevo segmento. Es decir, un nuevo segmento puede adjuntarse al final, al inicio o insertarse en el medio de la cadena de caracteres.

No obstante, la eliminación de ciertos segmentos en la cadena existente es posible al dejar al resto dentro de la estructura intacta. Borrar segmentos de cualquier posición de la cadena brinda la libertad de recortar la cadena en el punto que sea necesario. No es obligatorio que sea en un extremo, ya sea al inicio o al final, y la

longitud del segmento a borrar puede ser cualquiera, siempre que se encuentre entre un número mayor a cero, pero también menor a la longitud máxima

La consulta de un elemento resulta interesante y recibe el nombre de indexación de una cadena de caracteres. No es más que la selección de un elemento o un segmento de una determinada longitud dentro de una cadena.

Para facilitar todas estas tareas, existen propiedades de las cadenas que indican al usuario un valor de acuerdo a la cadena a la que se está haciendo referencia. Entre estos se encuentra la longitud de la cadena y la indicación de la posición de los caracteres en el objeto, entre las principales.

Como ejemplo para entender esta construcción de cadenas, la siguiente figura muestra una aplicación de construcción de cadenas de caracteres en la aplicación desarrollada. El primer bloque permite la combinación de caracteres, en este caso, la cadena de ceros con el valor de la variable de *frecuencia máxima*. El tamaño de este segmento será dependiente de la longitud del valor de la variable y será insertado a partir de la posición inicial de la cadena de ceros. Luego, la concatenación de diversas cadenas se realiza con el siguiente bloque que construye una sola cadena con los segmentos que recibe en cada una de sus posiciones. Coloca en el principio la cadena que se encuentra en la posición superior, luego la que está en la casilla inferior y así continúa en forma descendente. Es muy importante colocar en el orden correcto estos segmentos, ya que el mensaje podría tener partes intercambiadas entre sí. Finalmente, una vez construido el mensaje deseado, se envía por el puerto serial con el bloque de escritura hacia el puerto designado para esta tarea.

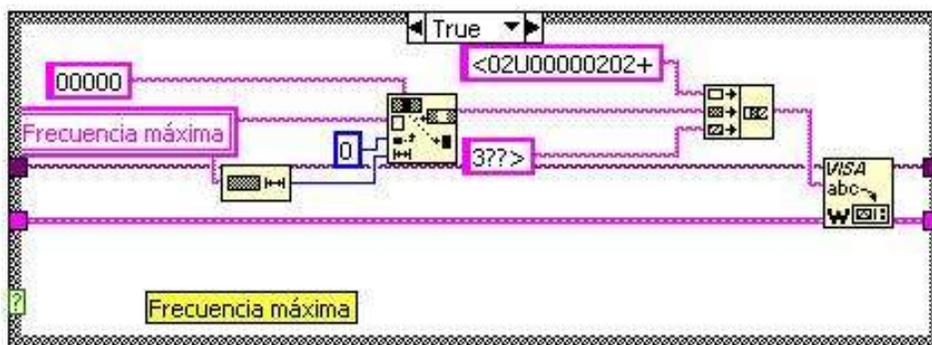


Figura 6.3: Construcción de cadenas de caracteres por segmentos

## D. Secuencias y ciclos

Una programación convencional requiere de varias instrucciones esenciales, siendo la principal la elección respecto de una variable pero si necesitamos generar una secuencia o repetir una serie de pasos, resulta de gran importancia contar con una instrucción que permita realizar esta acción. LabVIEW cuenta con estructuras cíclicas para llevar a cabo varias tareas repetitivas que pueden ser desarrolladas hasta un número finito de veces o hasta que el usuario decida detener la ejecución del ciclo. Podemos elegir si se desea ejecutar el diagrama mientras la condición sea verdadera, o continuar si es verdadera. Cabe mencionar que este ciclo siempre se ejecuta al menos una vez y es posible llevar el conteo del número de iteraciones realizadas por esta estructura. El siguiente ejemplo se muestra en la siguiente figura, este es un ejemplo de un ciclo que permite el control de nivel de un indicador de nombre *Tank* por medio de un control análogo llamado *Dial*. Acompañando a este código, se encuentran dos variables *X* y *Y* que activarán un indicador lumínico si ambas variables están encendidas. Para detener todo este código, se encuentra el botón *Stop*.

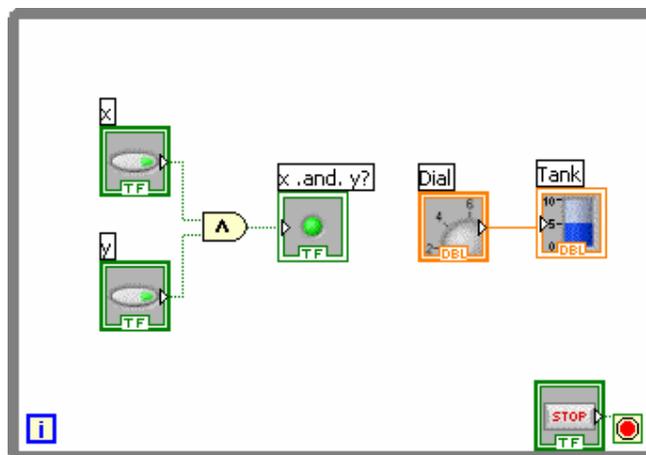


Figura 6.4: Ejemplo de ciclo controlado por el usuario

Cuando se desea dar un control de flujo donde una instrucción preceda a otra, se puede utilizar una secuencia de eventos. Esta estructura permite que la ejecución de tareas se dé en forma consecutiva y una tarea se ejecutará solamente cuando la predecesora ya haya terminado. Resulta de gran importancia el manejo de una secuencia si se desea adquirir un dato, pero hay que esperar que esté disponible. De forma similar, el envío de una serie de mensajes hacia el exterior en un orden determinado se puede

realizar con esta estructura. La estructura secuencial no tiene tamaño predefinido, mas bien, resulta ser una estructura dinámica a la cual se le pueden agregar o borrar segmentos de la secuencia. Nuevamente, esta estructura se usa básicamente para garantizar la ejecución de un diagrama o conjunto de instrucciones, antes o después de otro conjunto de instrucciones.

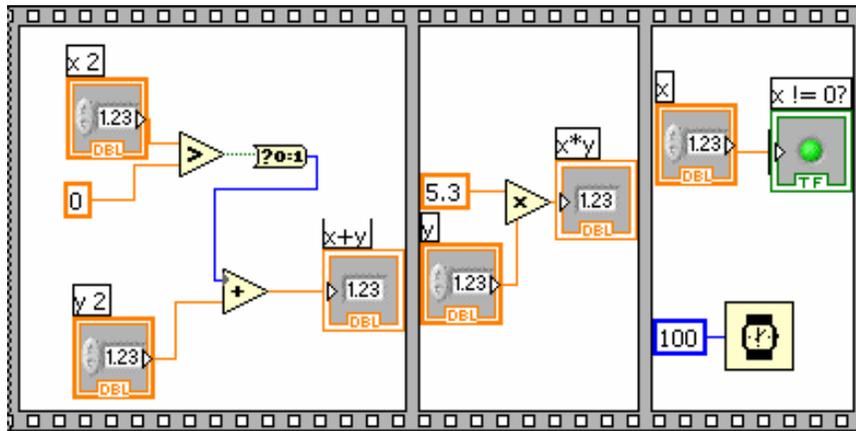


Figura 6.5: Ejemplo de estructura secuencial

La figura anterior muestra una secuencia de pasos a realizar en un programa. Consta de 3 pasos que deben ejecutarse de forma secuencial y no paralela como normalmente sucedería si se colocaran uno después del otro. El primer paso compara la variable  $X2$  para determinar si es mayor que cero. Si fuera cierto, entonces la condición verdadera la convierte a un valor lógico ya sea 1 ó 0, e ingresa a un bloque de suma. Este a su vez toma la variable  $Y2$  y le suma el valor lógico convertido. Finalmente el resultado es mostrado en el indicador  $X+Y$ . El siguiente paso se da una vez terminado el proceso anterior y consiste en multiplicar por 6.3 el valor de la variable  $Y$  para luego mostrarlo en el indicador  $X*Y$ . Luego el tercer y último paso no es más que un indicador de la variable  $X$  para encenderse si éste es igual a 0, mientras se da esta decisión también se estará esperando un tiempo de 100ms.

## VII. Aplicación desarrollada

### A. Características generales

Un sistema de control y monitoreo de parámetros comunes en un motor trifásico es de lo que consta la aplicación que se ha desarrollado. La consulta de parámetros importantes para el correcto funcionamiento de un motor trifásico se puede realizar con sólo presionar un botón y al instante se obtendrán desplegados en pantalla los valores de variables como la frecuencia mínima, máxima y de salto para lo que está programado el variador de frecuencias que controla al motor trifásico. De la misma forma, la modificación del funcionamiento del motor trifásico con tareas que deben realizarse en conjunto una a la vez, ahora se puede hacer con sólo presionar un botón con la función predeterminada como el ejemplo de la puesta en marcha, el apagado y el cambio de giro del motor.



Figura 7.1: Carátula de la aplicación desarrollada

La principal ventaja de esta aplicación sobre la operación del variador desde la pantalla de cristal líquido con las teclas que este trae para su operación, reside precisamente en la facilidad que el usuario tiene para modificar parámetros de una forma más simple y evitar así posibles errores de configuración y daños al equipo que maneja. El resultado obtenido de respuesta del variador de frecuencias al presionar un botón de actualización de parámetros permite al usuario obtener de forma inmediata una mejor idea del estado del motor trifásico.

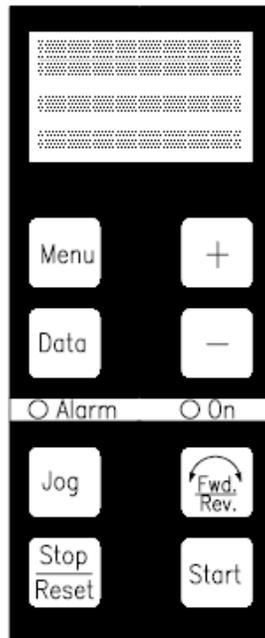


Figura 7.2: Diagrama del panel para interfaz del variador de frecuencias con el usuario

Con el objetivo de facilitar el uso de la aplicación, esta se ha diseñado en forma tabulada y separada por secciones claramente identificadas por pestañas en una ventana principal. Cada pestaña contiene una función independiente del resto, son excluyentes entre sí y resulta fácil saber en dónde se encuentra el operario con sólo ver el nombre de la pestaña de la ventana. La ventaja de tener pestañas en el programa radica en la ventaja de utilizar un espacio reducido para mostrar gran cantidad de información, sin resultar confuso para quien utiliza este programa.

Siendo esta una aplicación diseñada para simplificar tareas complejas de la industria, debe ser fácil utilizarla, ordenada para entender lo que se realiza y por lo tanto evitar errores en su uso que puedan darse hasta en el equipo que se está controlando.

## B. Inicio de sesión dentro del programa

Para poner en marcha la aplicación, el usuario debe seleccionar la pestaña de inicio de sesión del programa. En esta pestaña, se encontrarán dos campos, uno correspondiente al nombre de la cuenta del usuario y otro a la clave de dicha cuenta. Una vez ingresados ambos datos, se debe presionar el botón de “Login” para iniciar la sesión del programa.

Además del inicio de sesión, el usuario cuenta con otras opciones, cada una de estas opciones está disponible por medio de un botón que se encuentra en la parte inferior de la pantalla. Este inicio de sesión tiene como objetivo mantener un nivel de seguridad de la aplicación para evitar la ejecución errónea o imprudente de personas ajenas al proceso, y garantizar que el operario de la aplicación esté debidamente autorizado para su uso. La siguiente figura muestra la pantalla de inicio de sesión del sistema.



Figura 7.3: Inicio de sesión del programa

Una vez iniciada la sesión para el operario, se podrá usar la sección de adquisición y actualización que ofrece el programa.

Estas dos secciones serán descritas a continuación para dar un mejor entendimiento de su funcionamiento dentro del programa.

### C. Adquisición de datos dentro del programa



Figura 7.4: Adquisición de datos del motor a través del variador de frecuencias.

En la figura anterior se muestra la pantalla de adquisición de datos y alertas comunes que deben tomarse en cuenta para el variador de frecuencias. La toma de valores para estos parámetros se realiza al presionar una sola vez el botón de adquisición y el programa entregará los resultados de las 6 variables consultadas. Esta simplicidad permite que múltiples valores se desplieguen sin la necesidad de construir un mensaje para cada consulta y cometer errores en este proceso. Las alarmas se muestran arregladas a la derecha del panel, por lo que se verán con facilidad cuando una de estas se encuentre activa dado el contraste entre el color del indicador encendido con el fondo de la pantalla. Los resultados son obtenidos de la misma manera en que el mensaje se ha construido, por lo que debe entenderse el formato que este maneja para interpretar correctamente el valor que el variador de frecuencias ha entregado.

En algunos casos particulares de parámetros consultados, no se obtendrá el valor exacto de la variable consultada, sino más bien un número que, de acuerdo al manual del variador, corresponde a una constante del motor.

## D. Actualización de parámetros dentro del programa

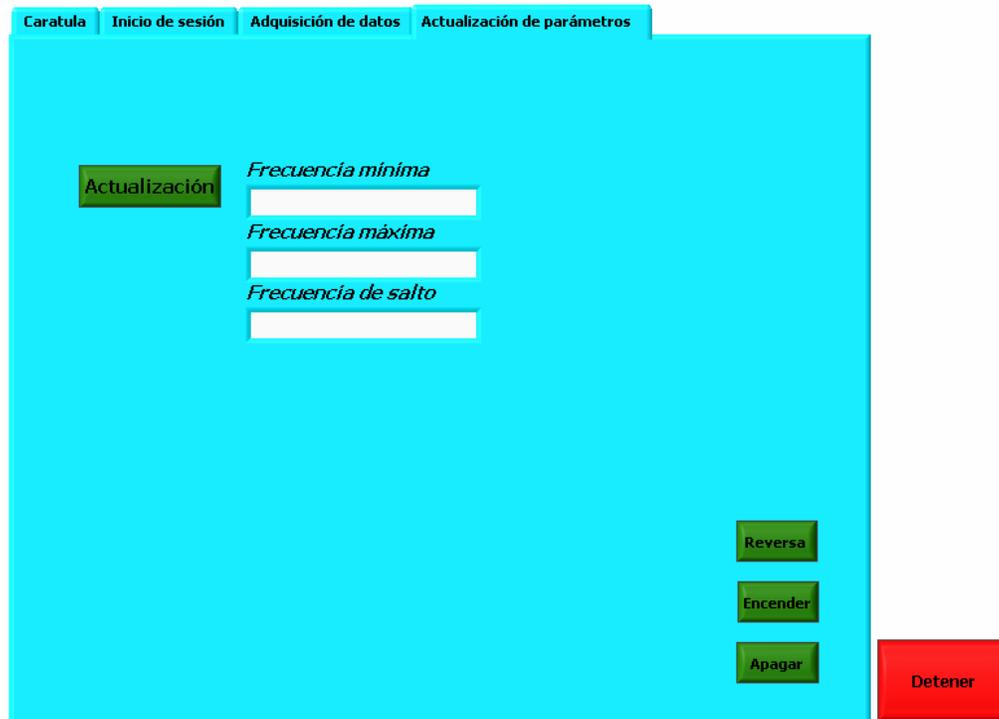


Figura 7.5: Actualización de variables del motor y algunos comandos predeterminados.

Finalmente, la actualización de parámetros presenta 3 espacios para ingresar las frecuencias en las que se estará operando el motor trifásico. Tanto la frecuencia mínima, máxima y de salto, permiten valores reales, lo que ofrece exactitud para poner a funcionar el motor en un valor muy específico. De forma distinta también se encuentra la modificación de dirección del giro del motor identificado con un botón de reversa. Al presionar este botón, no será necesario ingresar ningún número, dado que el programa se encargará de construir el mensaje con los valores necesarios para cambiar la dirección de giro. Acompañando a estos botones, se encuentran las opciones de encender y apagar el motor, ambas son similares a la reversa en el sentido de que sólo se debe presionar un botón para realizar la tarea deseada.

El encendido o puesta en marcha del motor es predeterminado de una serie de distintos tipos de arranque disponibles ofrecidos por el variador de frecuencias. Sin embargo, la selección del arrancado puede ser alterada cambiando ligeramente el mensaje que el botón tiene asociado. De igual manera el apagado puede realizarse de diversas formas y si resultara necesario cambiar la forma en la que el motor se apaga, bastará con modificar el mensaje del botón de acuerdo a la tabla del variador de frecuencias.

Esta aplicación se comunica con el variador de frecuencias por medio del protocolo RS-232C a través del cual se estarán enviando comandos dentro de mensajes con un formato determinado para la consulta y modificación de parámetros del variador de frecuencias. La longitud de la computadora desde la que se controla el proceso hasta el variador de frecuencias, dependerá de la velocidad a la que se está transmitiendo, así como el tipo de cable que se utilice, ya que si se cuenta con un cable blindado no ha de considerarse el ruido electromagnético que pueda afectar al medio de transmisión. De manera similar el baudaje que se elige para transmitir, permitirá establecer una longitud máxima segura para su funcionamiento.

Una interfaz como la desarrollada en esta ocasión, es un claro ejemplo de cómo un proceso que puede resultar sofisticado se convierte en algo más simple de manejar con la ayuda de herramientas de software, para este caso en particular LabVIEW. Es con este programa que se desarrolla la interfaz entre el proceso y el operario, que tiene como objetivo reducir la complejidad del proceso de controlar un motor trifásico de inducción, lo cual lleva asociado una reducción de errores comúnmente relacionados al ser humano.

## **E. Requisitos de operación**

La aplicación desarrollada requiere de condiciones mínimas para su funcionamiento correcto. Estas condiciones son, tanto para el software que se utiliza, como para el hardware.

A continuación se listan los requerimientos.

- Procesador Pentium III, Celeron o cualquier equivalente a 866MHZ
- Memoria RAM de 256MB
- Resolución mínima de 1024x768 píxeles
- Disco duro de 1.2GB
- Fuente de alimentación de corriente alterna de 220V a 60Hz

Otros materiales como destornilladores planos, cable AWG #12 y un conector serial DE-9 con 3 salidas de alambre UTP son necesarios y se asume la presencia de dichos materiales.

## **F. Funcionamiento**

El sistema de control y monitoreo permite la adquisición de valores en tiempo real del motor trifásico. La lectura la realiza por medio del envío de comandos con un formato determinado por medio de la comunicación serial apegados al protocolo RS232-Cc que a su vez son respondidos por el variador de frecuencia, que realiza las consultas de los parámetros. De la misma forma, por medio del envío de mensajes debidamente configurados, también se puede realizar la escritura o actualización de parámetros en el variador de frecuencia. Los mensajes que se mencionan con anterioridad tienen un formato específico y adecuado para la adquisición o actualización de los parámetros según la función a desarrollar.

Será explicado en diversas, partes tanto para la actualización como para la escritura, se mencionará cómo es que estas actividades son desarrolladas dentro del programa.

## 1. Lecturas de parámetros

De acuerdo con el variador de frecuencias utilizado, Danfoss VLT 2020, es necesario enviar un mensaje con un “frame” o formato determinado para realizar la lectura de algunos parámetros en el motor trifásico a utilizar. Esto es, construir una cadena de caracteres donde se especifique la dirección del variador, el comando a realizar y la dirección del parámetro a leer.

Para ello, construiremos una cadena de caracteres de acuerdo a este formato en el programa desarrollado en LabVIEW. Iniciamos con una primer cadena donde estará el mensaje “<02” esto es el primer carácter y la dirección con la que se identifica el variador de frecuencia. Luego, de acuerdo al formato, encontramos el segmento de 4 bits que corresponden al estatus del variador de frecuencia. Sin embargo, esto está reservado para el variador que nos devolverá el estado en este segmento. Para nuestra función de lectura, podemos escribir cualquier secuencia de 4 caracteres ya que igualmente serán reemplazados por la respuesta y se decidió enviar la cadena “0000”. Ahora se debe especificar el parámetro a acceder en un espacio de 4 bits, esto se hace escribiendo el número correspondiente al parámetro. Sin embargo, este número en el manual solamente tiene 3 dígitos, por lo que debemos anteponer un carácter “0” al mensaje, para cumplir con el requisito de 4 dígitos del mensaje.

Continuando con la construcción del mensaje, se especifica el signo del valor numérico de los próximos 5 bits. Esto se hace en caso se quiera actualizar el valor de un parámetro. Dado que no necesitamos hacer esto, podemos colocar cualquier carácter en el bit correspondiente al signo y en los próximos 5 bits designados para un valor numérico. Luego, el bit asignado para la coma o punto decimal también será omitido porque no vamos a actualizar ningún parámetro. En la sección del check sum, designada para la corrección de errores, hemos de especificar el valor correspondiente para el mensaje enviado. Para un caso donde esto no sea necesario, como lo es en la lectura, se ha de escribir el par de caracteres “??” donde se especifica que no habrá control por check sum. Finalmente se coloca el carácter “>” para terminar la cadena de caracteres.

Un ejemplo de esta cadena construida sería así: **<02R00000201+000003??>** que tiene el objetivo de comunicarse con la unidad número 2 para indicarle la lectura del comando 107, correspondiente a la frecuencia mínima del motor. El resto de

caracteres que componen la cadena, no son necesarios para realizar la función de lectura, solamente para cumplir con el formato que requiere el variador. En la siguiente figura, se puede observar el envío de un mensaje de lectura desde la PC hacia el variador de frecuencias. Basta con escribir la cadena de caracteres que se desea enviar al módulo de escritura desde el puerto y para el caso de la figura, se está enviando a la unidad 02 el comando de lectura del parámetro 104 correspondiente al voltaje del motor. El resto de valores, son despreciables pero deben estar ocupados para que se cumpla la longitud de 22 caracteres requerida en el mensaje a enviar.

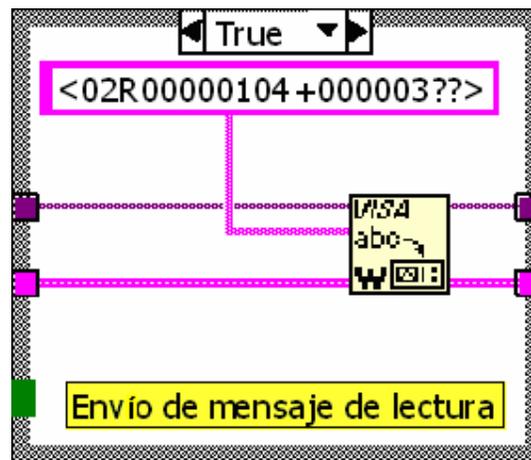


Figura 7.6: Envío del mensaje para ordenar la lectura de un parámetro

## 2. Actualización de parámetros

Así como podemos realizar la consulta de parámetros de la forma anteriormente expuesta, también resulta posible modificarlos. Esta tarea se realiza de la misma manera, al enviar un mensaje con el formato que requiere el variador de frecuencias y actualizará el parámetro con el valor que se ha designado.

Este mensaje es similar al anterior en lo que respecta, tanto al carácter de inicio como al del final. También en el uso de la dirección del variador en el segundo y tercer bit al igual que la especificación del parámetro en el espacio designado a partir del noveno bit. La respuesta del estatus del variador que viene en el mensaje desde el quinto bit no es necesaria para la actualización.

Únicamente resulta útil para esta configuración el bit de control, el parámetro a modificar, el signo y la posición de la coma o punto decimal en el valor que reemplazará al que contenga actualmente el parámetro. El resto de bits no es de interés. El primer bit a utilizar es el de control, resulta importante escribir el carácter importante ya que podría confundirse entre la función de actualización y control. Se ha de escribir el carácter “U”, letra inicial de la palabra en inglés *update*. En el bit posterior al parámetro se debe especificar el signo del valor a escribir en el parámetro, esto es, si es positivo o negativo. Danfoss tiene el convenio de que, si no es el carácter ASCII “-”, cualquier otro carácter se considera positivo. El siguiente conjunto de bits a modificar consiste en un conjunto de 5 bits donde se escribirá el valor numérico del parámetro empezando desde el primer bit contiguo al signo. Debe ocuparse todo el espacio si fuera el caso de utilizar decimales, que deben estar incluidos en estos 5 espacios. Finalmente, para considerar el caso de ambigüedad entre la parte entera y la decimal se encuentra este último espacio de 1 bit que indica cuantos números quedan hacia la derecha del punto decimal. Este bit, en conjunto con los 5 anteriores, es de suma importancia para especificar un valor numérico con precisión o una magnitud bastante grande de acuerdo al punto.

Para entender mejor la actualización, ser verá un ejemplo. Siendo la cadena de caracteres `<02U00000202+400003??>` se estaría especificando, dado el carácter “U”, la actualización de un parámetro, en este caso, el número 201 correspondiente a la frecuencia máxima. Se le estará asignando el valor de 40.000 debido al “3” que acompaña al número “40000”. Puede notarse que de no colocar de forma correcta el

número decimal se asignaría otro muy distinto, 10 veces mayor o menor, al parámetro deseado.

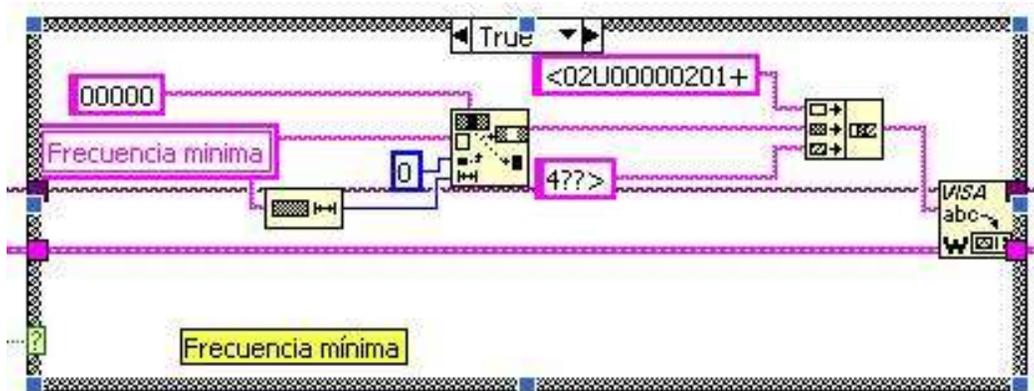


Figura 7.7: Envío del mensaje para la actualización de un parámetro

### 3. Interpretación de mensajes recibidos

Es de suma importancia entender los resultados que el variador de frecuencias nos devuelve cuando se ha hecho una solicitud de lectura o del estatus del variador. Sin embargo, la manera de entender este mensaje debe ser de la misma manera que se hizo en la construcción de la cadena de caracteres. Esto es, analizar por segmentos cada una de las partes del mensaje y así dar una interpretación correcta del variador.

El variador de frecuencias envía un mensaje de respuesta de la misma forma como le fue enviado. Pero el variador actualiza el mensaje al agregar bits de status en el lugar predeterminado y con valores numéricos en el espacio designado, para entregar el valor que se ha solicitado del parámetro específico en el mensaje.

Como se dijo anteriormente, partes del mensaje no serán modificadas, estas son: la dirección del variador que seguirá constante, el carácter de control con el comando de lectura o escritura hacia el variador, el número de parámetro a consultar y los caracteres de inicio y fin. El resto será reemplazado por los bits de estatus que indican el estado de operación del variador de frecuencias, así como el valor del parámetro. Debe tomarse en cuenta el bit de control para saber qué se está obteniendo y tomar los segmentos necesarios para la decodificación.

En el caso de haber hecho una lectura, debe considerarse el número de parámetro consultado y los próximos 7 bits que corresponden, tanto a la parte decimal, entera y signo del parámetro. Ciertos parámetros no mostrarán un número exacto para el parámetro que se está consultando, más bien, la respuesta estará expuesta de acuerdo a un convenio establecido por el variador. Por lo tanto, resulta necesario consultar el manual de usuario de este dispositivo para saber qué significado tiene cada uno de estos números enviados. Por ejemplo, si se consulta la frecuencia de salto, se obtendrá un número con una parte entera y posiblemente una parte decimal de acuerdo a su configuración. Pero en el caso de la consulta del voltaje del motor, no estaremos viendo una cantidad en el orden de los cientos, sino un número, que es el índice para consultar en el manual de usuario, en la región de este parámetro, para luego entender a qué voltaje corresponde este índice.

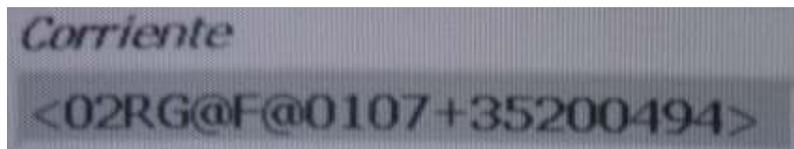


Figura 7.8: Muestra de un mensaje recibido del variador de frecuencias

#### 4. Otras funciones

El proyecto permite tanto la actualización y lectura de parámetros del variador de frecuencias que está controlando un dispositivo trifásico. Sin embargo, no está únicamente limitado a estas funciones, también puede desarrollar algunas tareas básicas para el control del dispositivo que maneja el variador.

Tanto el encendido como el apagado del motor trifásico resulta fácil de realizar, ya que sólo se debe presionar un botón para que este deje de funcionar o bien, si está energizado y apagado, pueda encenderse.

Estas tareas no son precisamente de modificación de parámetros en el variador como se explicó anteriormente, más bien consiste en manejar los parámetros de control del variador de frecuencias.

Para realizar estas tareas predeterminadas, se debe construir el mensaje con el carácter de control “C” acompañado de 4 bits que componen la parte de control sobre el

motor trifásico. Cada bit tiene asignado un carácter ASCII que cumple con una función del motor. En la siguiente tabla puede entenderse mejor esta función.

ASCII		CONTROL															
		Byte 8				Byte 7				Byte 6				Byte 5			
		N O F U N C T I O N / R E V E R S I N G	C H O I C E / O F / S E T - U P / 2	C H O I C E / O F / S E T - U P / 1	N O F U N C T I O N / / C A T C H - U P	N O F U N C T I O N / / S L O W D O W N	D A T A / N O T / V A L I D / / V A L I D	J O G / O F F / / O N	J O G / O F F / / O N	N O F U N C T I O N / / R E S E T	R A M P / S T O P / / S T A R T	H O L D / / R A M P / / E N A B L E D	Q U I C K - S T O P / / R A M P	C O A S T / / / E N A B L E D	O F F / / / 3	O F F / / / 2	O F F / / / 1
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00		
@	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
A	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	
B	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	
C	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	
D	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	
E	0	1	0	1	0	0	1	0	1	0	1	0	0	1	0	1	
F	0	1	1	0	0	1	1	1	0	1	0	0	1	1	0	1	
G	0	1	1	1	0	1	1	1	1	1	1	0	1	1	1	1	
H	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	
I	1	0	0	1	1	0	0	1	1	0	1	1	0	0	1	0	
J	1	0	1	0	1	0	1	1	1	0	1	1	0	1	0	1	
K	1	0	1	1	0	0	1	1	0	1	1	1	0	1	1	1	
L	1	1	0	0	1	1	0	1	1	1	0	1	1	1	0	1	
M	1	1	1	0	1	1	0	1	1	1	0	1	1	0	1	1	
N	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	
O	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

Figura 7.9: Tabla de control proporcionada por el manual del variador de frecuencias.

Como puede observarse, estas tareas están pensadas para el uso común de un motor trifásico, tanto la variación de la velocidad actual hacia la de salto, como distintos tipos de apagados así como cambiar el giro de rotación del rotor.

Es importante hacer referencia al tercer bit del séptimo byte, ya que el valor de este parámetro debe ser 1 para que el resto de comandos sea válido. Así, entonces, al elegir los bits necesarios para las funciones a realizar, se asigna un carácter ASCII en el byte que le corresponde.

Para ejercer control sobre el motor, debe ser construido un mensaje donde se especifique el carácter de control, la dirección del variador de frecuencia y el conjunto

de caracteres ASCII designados a cada byte para su función. El resto de parámetros no es necesario para el control, por lo tanto pueden tener otros valores.

## **VIII. Identificación y resolución de problemas comunes**

Como cualquier procedimiento para resolver un problema, este también puede presentar sus problemas propios. Las razones de dichos contratiempos han de variar desde un simple alambre mal conectado hasta velocidades distintas de comunicación y por ende una incompatibilidad de comunicación entre dispositivos.

Se recopila una serie de posibles fallas que afectarían al proyecto, así como las soluciones para superar este problema. Debe hacerse mención que las fallas aquí listadas no son todas las que el usuario pueda experimentar. Sin embargo, son las más probables a suceder dados los errores típicos de operación del ser humano.

No obstante, el sistema de control trabajará de manera óptima si se cumplen los requisitos mínimos de su funcionamiento e idealmente la posterior revisión de personal calificado para su visto bueno. El usuario final puede estar seguro de que no habrá problemas en interpretación de comandos que pongan en riesgo el funcionamiento del motor trifásico, siempre y cuando se cumplan los requisitos necesarios.

### **A. Problemas de comunicación**

Se discutirá como primer punto a fallar la etapa de comunicación, siendo esta la más vulnerable e importante del sistema de control.

Dado que este sistema de control se basa en la comunicación entre la computadora remota y el variador de frecuencias, es prudente considerar que existan errores de comunicación entre ambos dispositivos.

No solamente habría errores de Hardware, sino que también de Software debido a la mala configuración de la comunicación. Ambos tipos de errores pueden dar resultados que sean desde una mala recepción de mensajes hasta la imposibilidad de establecer un vínculo de comunicación adecuado con el variador de frecuencias.

Aunque pueden darse problemas de ruido electromagnético sobre el medio de transmisión, en este caso un cable UTP (Unshielded twisted pair), no se consideró este problema dadas las condiciones de prueba donde la longitud es menor a un metro y la ausencia de otros dispositivos emisores de electromagnetismo no afectó la comunicación.

Es recomendable utilizar un cable STP (Shielded twisted pair) cuando el ambiente en el que se sitúa el sistema de control está rodeado de interferencia electromagnética significativa. Dicha interferencia alterará el mensaje que se transmita por el cable, a menos que se proteja con un blindaje electromagnético.

## **1. Alambrado**

Un problema común al instalar este sistema se encuentra en las conexiones entre dispositivos. Debe asegurarse que el conector DE-9 de la computadora esté debidamente conectado sin hacer falsos contactos. El extremo opuesto, que se conecta al variador de frecuencias, debe tener debidamente atornillados los contactos. No basta con tan solo colocarlos en sus posiciones adecuadas, debe hacerse girar el tornillo de cada contacto para que los sujetadores hagan presión adecuada sobre el alambre.

Las terminales que han de alambrarse para la comunicación serial son: la número 20 asociada a la referencia digital, 71 a la recepción, y 72 a la transmisión. Debe notarse que existen dos nodos de referencia, es importante conectar la referencia digital a la del medio de transmisión. La falta de cumplimiento a este punto resultará en una comunicación errada dado que no se cuenta con un buen nivel de referencia.

En el caso de haber alambrado correctamente el cable de comunicación serial y no recibir una respuesta del variador de frecuencias, debe verificarse la conexión de +24V hacia las terminales especificadas por el manual de operación del variador de frecuencias. Es decir, la terminal número 12 debe estar conectada a las terminales 18 y 27 correspondientes a DIN1 y DIN2, respectivamente.

Puede observarse la forma en la que se debería alambra el variador de frecuencias entre sus terminales, algunas de ellas no son necesarias para la aplicación. Los cables están haciendo énfasis al STP dado que presenta los entornos de los cables conectados al nodo de referencia física.

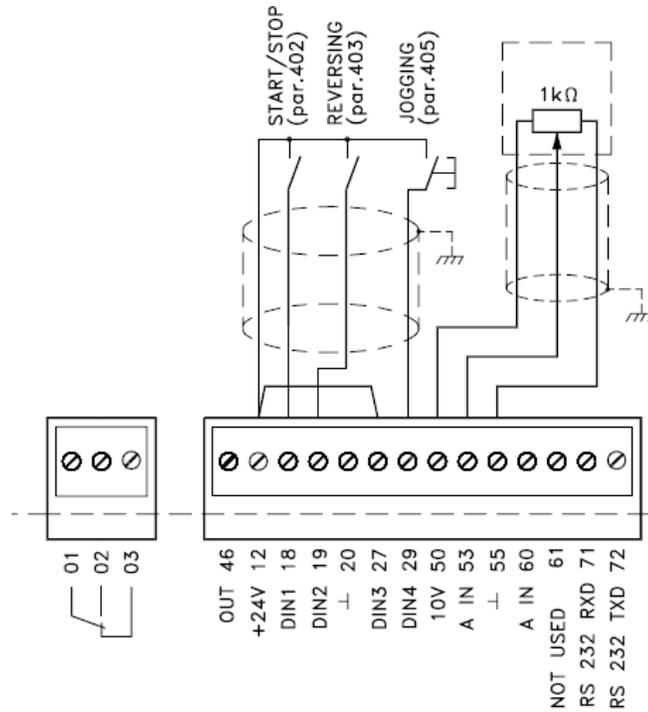


Figura 8.1: Alambrado tradicional, tomado del manual del variador de frecuencias.

## 2. Mensajes erróneos

Una comunicación adecuada consiste en aquella que se logra establecer para enviar y recibir problemas, sin ningún problema de interpretación de datos. Pero aun cuando la comunicación se establece, debe considerarse la calidad del mensaje que se ha recibido y si cumple con las características impuestas para el formato del mensaje.

Si se recibe un mensaje incompleto, se considera erróneo, así como también un mensaje con una longitud mayor a la que se espera. Esta variación en la longitud puede acarrear errores a la hora de la interpretación, si se trata de un programa o máquina que se encargará de esta tarea. Un ser humano podría resolver este problema dada la flexibilidad y detección de los segmentos separados del mensaje.

Estos segmentos separados son básicamente mensajes que se adjuntan a otros, ya que no se completó el mensaje anterior para lograr los 22 bits que normalmente componen al mensaje. Por lo tanto, los bits que no fueron entregados anteriormente serán entregados en el siguiente mensaje. Este fenómeno ocurre nuevamente y no habrá un orden en la recepción de los comandos a interpretar del mensaje.

La solución a este problema está en indicar un tiempo de espera o “time out” mayor al que ya se tenga configurado. También debe especificarse al receptor de mensajes que muestre exactamente 22 bits y no los bits que se encuentren en el canal. Otras soluciones también pueden implementarse como la detección de los caracteres de inicio y fin del mensaje para garantizar que se han recibido con éxito los 22 bits.

### **3. Velocidades de transmisión**

Tal y como se describió anteriormente, la velocidad de transmisión, o baudaje, puede variar entre los dispositivos según sea la conveniencia al tomar en cuenta la longitud del cable, así como el ruido en el medio ambiente.

Por lo que, siendo una velocidad configurable tanto para el variador de frecuencias como para el sistema de control, puede existir la posibilidad de que ambas velocidades sean distintas entre sí al dar entonces una incompatibilidad en la comunicación y errores de interpretación de mensajes.

Dicho error no permitirá una comunicación adecuada y de existir, será con mensajes erróneos por la duración de cada bit de información que sería distinta para el variador de frecuencias comparada con la del sistema de control.

Ha de consultarse la velocidad del variador de frecuencias para configurar el sistema de control de acuerdo a este parámetro que se mide en baudios. De la misma manera, puede configurarse el variador a la velocidad del sistema de control pero en forma más restringida para los baudajes disponibles, por lo tanto no tiene la misma flexibilidad que ofrece el programa. Es por esto que se recomienda la primera forma de compatibilidad de baudajes en donde el sistema de control se ajusta al variador de frecuencias.

#### **4. Estado del variador de frecuencias**

El variador de frecuencias es una pieza fundamental en el desarrollo de un control y monitoreo de parámetros para un motor trifásico. Su estado de operación es importante para el correcto funcionamiento del motor y el sistema de control. De presentarse una falla en él, existirán algunos parámetros inaccesibles o dependiendo de la gravedad del estado del variador de frecuencias, puede no accederse a ningún parámetro.

## IX. Conclusiones

- La implementación de una HMI en la industria permite un grado de supervisión mayor al que un variador de frecuencias ofrecería por medio de la pantalla LCD, dada la información disponible y las tareas simultaneas que se pueden realizar con solo presionar un botón.
- Reducir la cantidad de errores debidos al ser humano permite que un proceso sea más seguro y económico, por lo que es importante que el operario tenga la mejor información disponible para tomar las mejores decisiones de acuerdo al proceso que desarrolla.
- LabVIEW es un programa distinto a muchos otros disponibles en el mercado, ya que la programación por bloques o módulos para construir un instrumento virtual resulta adecuado para estas aplicaciones por las múltiples ventajas que presenta. Basta con mencionar la configuración de una conexión al puerto serial, publicación de datos en la red, generación de reportes y la reducción significativa de tiempo en el desarrollo de nuevas aplicaciones comparado con otros lenguajes de programación.
- La velocidad fue regulada al variar la frecuencia del voltaje de alimentación pero a diferencia de otras técnicas de control de velocidad, esta no presenta pérdidas agregadas como sería aumentar la resistencia interna del rotor.
- Esta aplicación y la comunicación por puerto serial no se limitan solamente al tipo de variador de frecuencias que se utilizó. Más bien, puede modificarse levemente para aceptar otros protocolos de comunicación de otros dispositivos para leer y escribir por el puerto serial.

## **X. Recomendaciones**

- La implementación de esta aplicación a otras áreas de la industria resulta adecuada para un proceso donde se desee incluir automatización así como reducción de errores humanos.
- Dada la facilidad de uso que esta interfaz presenta al usuario, es recomendable su uso para este tipo de aplicaciones donde se puede simplificar un proceso industrial
- Implementar otras funciones a las ya existentes en la aplicación actual como la publicación de resultados y control a través de la Internet dará mayor flexibilidad y fortaleza a la aplicación.
- Modificar los protocolos de comunicación entre los equipos ya sea computadoras, variadores de frecuencias o monitores de potencia permitirá que la aplicación controle una mayor gama de dispositivos industriales.

## XI. BIBLIOGRAFÍA

- Béla G. Lipták. 1999. *Process Control*. 3. era edición. Boca Ratón, Florida. CRC Press LLC 1551 Págs.
- Chapman, Stephen J. 1995. *Máquinas eléctricas*. 2. da edición. Octavio Posada Salazar. Santafé de Bogotá, Colombia. McGraw-Hill 740 Págs.
- Rizzoni, Giorgio. 2001. *Principios y aplicaciones de la ingeniería eléctrica*. 3. era edición. Roberto Ríos Martínez. Bogotá, Colombia. Mc Graw Hill. 959 Págs.
- Bloomfield, Louis A. *How Things Work*  
<http://howthingswork.virginia.edu/>
- Christopher E. Strangio. *The RS232 Standard*. CAMI Research Inc.  
[http://www.camiresearch.com/Data\\_Com\\_Basics/RS232\\_standard.html](http://www.camiresearch.com/Data_Com_Basics/RS232_standard.html)
- InetDaemon.com Enterprises. *Non-Return-To-Zero(NRZ)*  
[http://www.inetdaemon.com/tutorials/theory/modulation/non\\_return\\_to\\_zero.shtml](http://www.inetdaemon.com/tutorials/theory/modulation/non_return_to_zero.shtml)
- International Engineering Consortium. *The Human-Machine Interface(HMI)*.  
<http://www.iec.org/online/tutorials/hmi/index.html>
- Jupitermedia Corporation. *Webopedia*  
<http://www.webopedia.com/>
- Siemens Energy and Automation. *Basics of AC Drives*  
[http://www.sea.siemens.com/step/templates/lesson.mason?ac\\_drives:1:1:1](http://www.sea.siemens.com/step/templates/lesson.mason?ac_drives:1:1:1)
- Siemens Energy and Automation. *Basics of AC Motors*  
<http://www.sea.siemens.com/step/templates/lesson.mason?motors:1:1:1>
- Wikipedia. *Non-return-to-zero*.  
<http://en.wikipedia.org/wiki/NRZI>