

UNIVERSIDAD DEL VALLE DE  
GUATEMALA

Facultad de Ingeniería

Implementación de una Red de Variadores de Frecuencia en el  
Laboratorio de Operaciones Unitarias para la Automatización de  
Equipos Industriales

Héctor Estuardo Bonilla Díaz

Guatemala

2010



Implementación de una Red de Variadores de Frecuencia en el  
Laboratorio de Operaciones Unitarias para la Automatización de  
Equipos Industriales

UNIVERSIDAD DEL VALLE DE  
GUATEMALA

Facultad de Ingeniería

Implementación de una Red de Variadores de Frecuencia en el  
Laboratorio de Operaciones Unitarias para la Automatización de  
Equipos Industriales

Trabajo de graduación presentado  
para optar al grado académico de Licenciatura en Ingeniería  
Electrónica

Guatemala

2010

Vo. Bo. :

(f) \_\_\_\_\_  
Ing. Carlos Esquit

Tribunal Examinador:

(f) \_\_\_\_\_  
Ing. Carlos Burmester

(f) \_\_\_\_\_  
Ing. Gamaliel Zambrano

(f) \_\_\_\_\_  
Ing. Carlos Esquit

Fecha de aprobación: Guatemala, 1 de diciembre del 2010

## PREFACIO

Este trabajo de graduación es parte del esfuerzo que se ha hecho junto a fases anteriores, para automatizar equipos que pertenecen al laboratorio de operaciones unitarias de la Universidad del Valle de Guatemala y poder convertirlo en una planta piloto.

Este trabajo de graduación surgió de la necesidad de buscar un método moderno de monitoreo para algunos equipos del laboratorio de operaciones unitarias. El método tradicional consiste en encender cada equipo y disponer de la variable de interés estrictamente en el display del variador de frecuencia, sin contar con un registro de los mismos. La modificación es altamente beneficiosa, principalmente por razones de versatilidad en el muestreo y adquisición de datos necesarios para estudios de comportamiento de los motores.

Además, este trabajo de graduación fue posible gracias a Dios y a mi familia que me han dado su cariño y apoyo incondicional, a mis catedráticos que han compartido su conocimiento a lo largo de estos años y a mis compañeros de estudio y amigos que junto a sus risas y desvelos hemos construido este tiempo maravilloso de retos saldados y por saldar.

## CONTENIDO

PREFACIO.....	v
LISTA DE TABLAS .....	vii
LISTA DE GRAFICOS.....	ix
RESUMEN.....	xi
I. INTRODUCCIÓN.....	1
II. MARCO TEÓRICO Y ANTECEDENTES.....	2
A. Variadores de frecuencia .....	2
1. Etapa rectificadora.....	3
2. Inversor .....	3
3. Etapa de control.....	4
B. Bomba centrífuga.....	5
C. Reactor químico .....	8
D. Torque del motor de inducción AC:.....	9
E. Modbus y RS-485: .....	12
1. Transmisión serial de Modbus .....	14
2. Direcciones de Modbus .....	15
III. JUSTIFICACIÓN .....	19
IV. OBJETIVOS.....	20
A. Generales .....	20

B. Específicos.....	20
V. METODOLOGÍA .....	21
VI. RESULTADOS .....	22
A. Diagrama general del módulo de variadores de frecuencia .....	22
B. Banco hidráulico.....	23
C. Elevador de cangilones y de tornillo sin fin .....	28
D. Reactor químico.....	33
E. Especificaciones de motores.....	34
F. Convertidor RS232 – RS485.....	35
G. Diagramas de flujo de la programación de Modbus .....	38
VII. ANÁLISIS DE RESULTADOS .....	42
A. Mediciones en los variadores instalados y gráficas:.....	42
B. Comunicación con los variadores desde estación piloto:.....	43
VIII. CONCLUSIONES .....	45
IX. RECOMENDACIONES.....	46
X. BIBLIOGRAFÍA.....	47
XI. GLOSARIO.....	50
XII. ANEXOS .....	51

## LISTA DE TABLAS

Tabla 1. Longitud máxima del cable hacia el motor .....	4
Tabla 2. Relación de torque en función de RPM y número de polos .....	10
Tabla 3. Comparación de los estándares RS232 y RS485 .....	12
Tabla 4. Campos y forma de la trama de Modbus.....	13
Tabla 5. Propiedades de Modbus ASCII y RTU .....	14
Tabla 6. Rangos de direcciones del protocolo Modbus (10) .....	15
Tabla 7. Códigos del estándar Modbus soportado por los variadores de Allen Bradley...	16
Tabla 8. Registros del variador relacionados con la comunicación .....	16
Tabla 9. Descripción de los 8 bytes de Modbus (11) .....	17
Tabla 10. Registro 8192 del variador PowerFlex 40 .....	18
Tabla 11. Mediciones tomadas al variador del Banco Hidráulica.....	25
Tabla 12. Datos tomados del elevador de cangilones .....	30
Tabla 13. Especificaciones de motores.....	34

## LISTA DE GRÁFICOS

Ilustración 1. Principios de operación del motor de inducción .....	3
Ilustración 2. Diagrama de bloques de un variador (9) .....	4
Ilustración 3. Portadora y salida del variador (9).....	5
Ilustración 4. Curvas características de una bomba centrífuga y punto de operación .....	6
Ilustración 5. Potencia asociada a cada uno de los métodos de control de flujo .....	7
Ilustración 6. Curva típica del torque vs frecuencia de un motor de inducción.....	10
Ilustración 7. Respuesta del variador Powerflex 40, frecuencia vs torque (13).....	11
Ilustración 8. Comportamiento transiente del variador PowerFlex 40 (13).....	11
Ilustración 9. Diagrama lógico del proyecto .....	22
Ilustración 10. Diagrama del Banco Hidráulico.....	23
Ilustración 11. Banco Hidráulico .....	24
Ilustración 12. Corriente de salida del parámetro d003 del variador .....	25
Ilustración 13. Corriente de salida presente en una de las líneas del motor. ....	26
Ilustración 14. Voltaje de salida presente en una de las líneas del motor .....	26
Ilustración 15. Voltaje de salida presente en una de las líneas trifásicas .....	27
Ilustración 16. Frecuencia vs corriente de torque .....	27
Ilustración 17. Frecuencia vs torque.....	28
Ilustración 18. Diagrama de elevadores .....	29
Ilustración 19. Variador instalado en el elevador de cangilones.....	30
Ilustración 20. Voltaje de variador vs frecuencia .....	31
Ilustración 21. Corriente de variador vs frecuencia.....	32
Ilustración 22. Corriente de línea vs frecuencia .....	32

Ilustración 23. Voltaje de línea vs frecuencia .....	33
Ilustración 24. Diagrama del reactor CSTR .....	33
Ilustración 25. Reactor químico y variador .....	34
Ilustración 26. Motor Baldor M3542 instalado en el sistema de cangilones (4) .....	35
Ilustración 27. Diagrama del circuito de conversión de RS232 a RS485 (12) .....	35
Ilustración 28. Circuito convertidor prototipo .....	36
Ilustración 29. PCB del circuito convertidor de RS232 a RS485.....	37
Ilustración 30. Circuito convertidor de RS232 a RS485 .....	37
Ilustración 31. Diagrama de flujo de la lectura de los holding registers. ....	38
Ilustración 32. Diagrama de flujo de la escritura de los registros. ....	39
Ilustración 33. Lectura de algunos registros por el Maestro de la red .....	41

## RESUMEN

La electrónica ha tenido un papel influyente en la automatización industrial. La utilización de los variadores de frecuencia en la industria ha aumentado en los últimos años, debido a la necesidad de mejorar los procesos y los consumos de energía en máquinas que utilizan motores.

Este trabajo es parte del proyecto de automatización que se lleva a cabo en el laboratorio de operaciones unitarias de la Universidad del Valle de Guatemala. En el cual han tomado un papel protagónico estudiantes de ingeniería electrónica e ingeniería química.

En diversos procesos que se llevan a cabo en el laboratorio de operaciones unitarias, es necesario variar la velocidad de operación de los equipos involucrados. Por lo que se ha recurrido a instalar variadores de frecuencia marca Allen Bradley de la línea PowerFlex, en los equipos que poseen motores tanto trifásicos como monofásicos. Estos variadores son del tipo PWM, por lo que operan con el principio de modulación de ancho de pulso.

En este trabajo se intervino, en procesos que tienen distintos tipos de cargas, las de par constante en donde se requiere optimizar el proceso, tal como lo son las máquinas transportadoras como los elevadores de cangilones, el tornillo sin fin y el reactor químico. Por otro lado está el banco hidráulico que posee una bomba centrífuga que también se intervino para agregar el variador y poder tener control sobre esta carga de par variable.

A su vez se ha monitoreado el comportamiento de los aparatos con carga, obteniendo mediciones de parámetros de interés, tales como la corriente y voltaje tanto de línea como del variador.

Posteriormente se diseñó una red de comunicación con topología de bus, bajo el protocolo RS485. Un cable especializado se ha colocado en una canaleta aérea montada a lo largo del laboratorio.

Dado que la topología lógica de la red es cliente servidor, se diseñó un programa que consiste en el maestro de la red de variadores que actúa como el servidor de la red, este permite monitorear los variadores y todos los registros que contienen información de interés, desde la frecuencia a la que está operando el motor hasta la corriente de torque que está proveyendo. Adicional al monitoreo se tiene la capacidad de ejercer control sobre el variador que se monitoree, dando la posibilidad de encender y apagar el variador, así como también introducir la frecuencia de operación deseada.

El que el usuario tenga acceso a este tipo de automatización juega un papel determinante en la preparación del alumno que interactúa con los equipos automatizados. A los que se enfrentará en la industria en su ejercicio profesional.

## I. INTRODUCCIÓN

Este trabajo de graduación, trata de la implementación de una red de variadores de frecuencia para los motores de diversos equipos industriales, en los cuales se tuvo la necesidad de mejorar su rendimiento en el consumo y la producción. Algunos de los equipos que se sometieron a dicha automatización son los siguientes: un expeller, un elevador neumático, una tolva de alimentación, dos reactores, un secador rotativo, dos elevadores de cangilones, un banco hidráulico y un chiller o enfriador. Dichos equipos se encuentran instalados en el laboratorio de operaciones unitarias de la Universidad del Valle de Guatemala.

Se instalaron a los motores de cada equipo los variadores de frecuencia y se colocaron los parámetros necesarios de acuerdo a las necesidades puntuales de cada equipo, para maximizar la eficiencia de estos, incluyendo el mejoramiento de su sistema de arranque y paro.

Se monitoreó de forma remota cada uno de los parámetros disponibles que interesan al operador en cada uno de los equipos, se modificó su arranque, paro y frecuencia de operación para cada motor, remotamente por medio de la interfaz humano-computador, instalada estratégicamente en la planta piloto y conectada a sus esclavos por un bus de datos que maneja un protocolo de comunicación estandarizado; como lo es Modbus-RTU implementado en un lenguaje de alto nivel sobre RS485.

## II. MARCO TEÓRICO Y ANTECEDENTES

### A. Variadores de frecuencia

Consisten en uno de los muchos equipos que ayudan al ahorro del consumo y demanda de recursos en la industria, los variadores de frecuencia tienen la capacidad de permitir arranques suaves de motores, erradicando picos de corriente que se generan frecuentemente al arrancar motores produciendo calentamientos y daños al equipo.

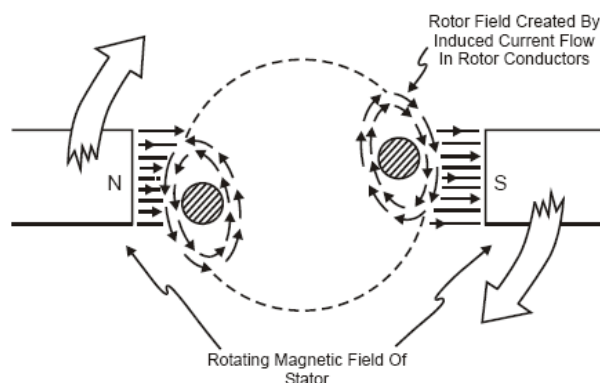
Permiten el control de la velocidad de los mismos, logrando un mejor manejo y control sobre la producción y proporciona versatilidad al momento de adaptar motores a distintos procesos.

En el caso de los variadores PWM (Modulación de Ancho de Pulsos), el variador de frecuencia regula la frecuencia del voltaje aplicado al motor, logrando modificar su velocidad.

Simultáneamente con el cambio de frecuencia, debe variarse el voltaje aplicado al motor para evitar la saturación del flujo magnético con una elevación de la corriente que dañaría el motor.

El número de polos y la frecuencia aplicada en los motores de inducción (Fig. 1) determinan la velocidad de giro del rotor (Ecuación.1). Esta ecuación incluye un efecto denominado “deslizamiento” que es la diferencia entre la velocidad del rotor y la rotación del campo magnético en el estator. Los campos magnéticos del rotor tratarán de ponerse en sincronía a la rotación de los campos del estator. [12]

Ilustración 1. Principios de operación del motor de inducción



Ecuación 1. Revoluciones por minuto (RPM)

$$velocidad\ del\ eje = \frac{120 * frecuencia}{No. Polos} - deslizamiento$$

Los variadores de frecuencia PWM contienen las siguientes partes principales, con pequeñas diferencias en componentes de hardware y software:

1. Etapa rectificadora. Convierte la tensión alterna en continua mediante rectificadores de diodos y un filtro para suavizar la tensión rectificada y reducir la emisión de armónicos y así mejorar el factor de potencia. [6]

2. Inversor. Convierte la tensión continua en otra de tensión y frecuencia variable mediante la generación de pulsos. Actualmente se emplean IGBT's (Isolated Gate Bipolar Transistors) para generar los pulsos controlados de tensión. [12]

La frecuencia portadora de los IGBT se encuentra entre 2 a 16kHz. Una portadora con alta frecuencia reduce el ruido acústico del motor pero disminuye el rendimiento del motor y la longitud permisible del cable hacia el motor (Figura.2). [12]

Por otra parte, los IGBT's generan mayor calor. Los equipos más modernos utilizan IGBT's inteligentes que incorporan un microprocesador con todas las protecciones por: sobre corriente, sobre tensión, baja tensión, cortocircuitos, puesta a tierra del motor, sobre temperaturas, etc. [12]

**Tabla 1. Longitud máxima del cable hacia el motor**

**Maximum cable distance AFD to motor**

Switching Frequency	460 Volt Premium	230 Volt Premium	380 Volt Premium	575 Volt Premium
3Khz	196 ft	481 ft	295 ft	53 ft
6Khz	138 ft	340 ft	209 ft	37 ft
9Khz	113 ft	278 ft	170 ft	31 ft
12Khz	98 ft	241 ft	148 ft	26 ft
15Khz	88 ft	215 ft	132 ft	24 ft
20Khz	76 ft	186 ft	114 ft	21 ft

For voltages and switching frequencies not listed above please use the following formula to determine maximum cable distance:

$$\frac{1328 - (2.15 \times \text{Voltage})}{\text{Square Root (Switching Freq)}} = \text{Feet}$$

3. Etapa de control. Esta etapa controla los IGBT para generar los pulsos variables de tensión y frecuencia. Además controla los parámetros para arranque, parada y variación de velocidad; en donde las señales externas de referencia están aisladas galvánicamente para evitar daños en sensores o controles y evitar ruidos en la etapa de control.

**Ilustración 2. Diagrama de bloques de un variador (9)**

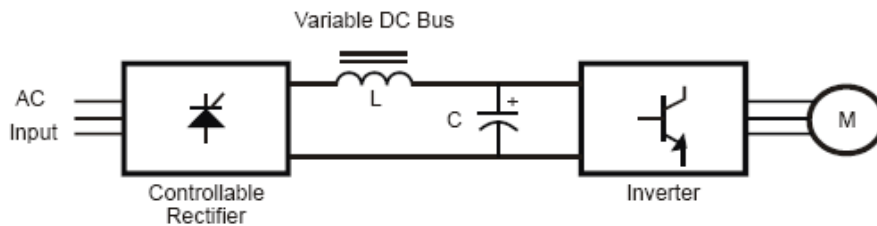
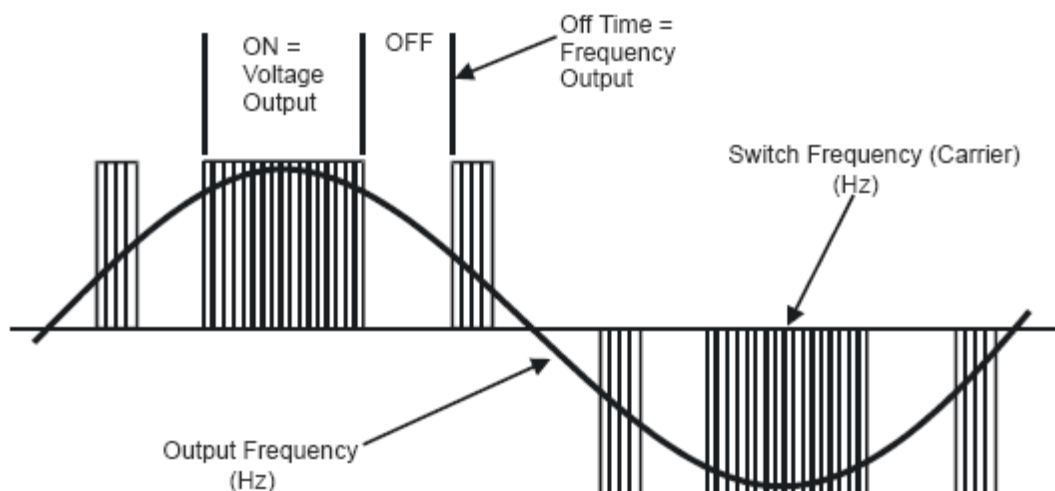


Ilustración 3. Portadora y salida del variador (9)



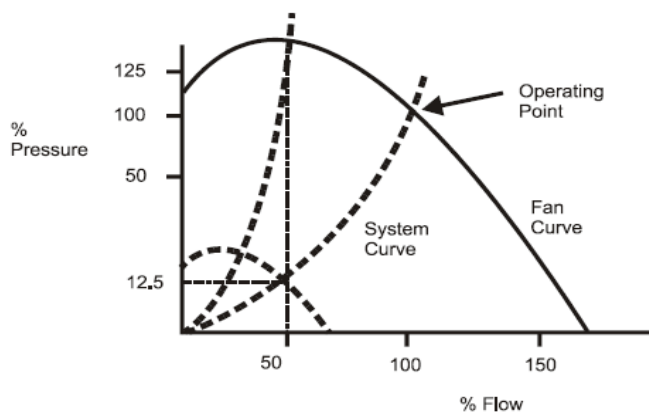
## B. Bomba centrífuga

Las bombas y ventiladores son aplicaciones comúnmente usadas en el ambiente industrial. Los métodos habitualmente utilizados para el control de flujo son:

- Estrangulamiento de válvula de salida: El flujo es controlado a través del cierre parcial de la válvula ubicada a la salida de la bomba
- Uso de un bypass, que conecta la salida con la entrada.
- Uso de un control ON-OFF: Con este método, la bomba es encendida y apagada, logrando un flujo promedio.
- Uso de variadores de frecuencia

Las curvas características de una bomba centrífuga están representadas esquemáticamente en la Ilustración 4. La intersección entre la curva de la bomba y la curva del sistema define el punto de operación de la bomba. [1]

Ilustración 4. Curvas características de una bomba centrífuga y punto de operación



La potencia relativa, asociada a los diferentes métodos de control de flujo, puede ser estimada a través del área entre los ejes así también el punto de operación

Ecuación 2. Potencia en función de la presión y el flujo

$$P = Q * H$$

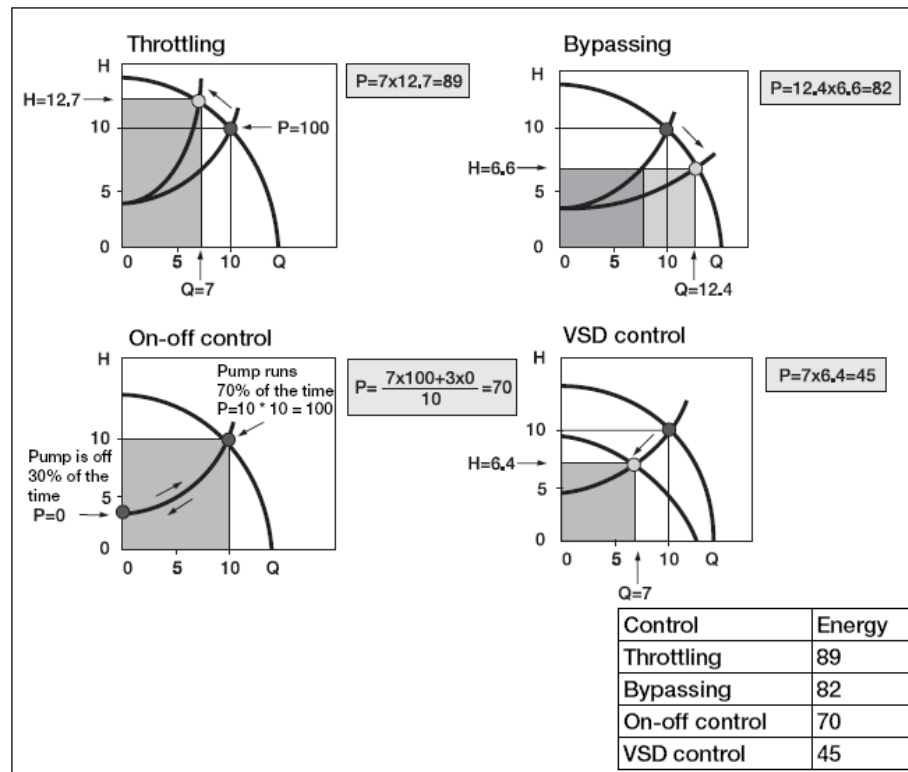
P = Potencia

H = Presión

Q = Flujo

Mediante el siguiente ejemplo, considerando un flujo promedio de 70%, es posible mostrar la diferencia entre la potencia demandada por cada uno de los diferentes métodos de control de flujo previamente señalados (Ver Ilustración 5). [6]

Ilustración 5. Potencia asociada a cada uno de los métodos de control de flujo



El uso de variadores de frecuencia en aplicaciones de bombeo, presenta las siguientes ventajas:

1. Disminución del costo de vida de la bomba: Los principales costos (ordenados de mayor a menor) asociados a un sistema de bombeo son la energía consumida, inversión inicial y su correspondiente mantenimiento.
2. Menor costo de mantenimiento y reparación: Al usar variadores de frecuencia se pueden reducir los costos de mantenimiento y reparación de las bombas, según: [1]
  - a. Reducción del estrés mecánico de la bomba.
  - b. Reducción de riesgos de cavitación.

- c. Reducción de daño en bomba debido a cambios bruscos de flujo, asociados a la partida de ésta.
- d. Permite la aplicación de bombas en paralelo, controladas según la demanda de flujo solicitada.

Los variadores de frecuencia ofrecen la ventaja adicional de poder hacer trabajar a la bomba a la velocidad óptima o de mayor rendimiento.

### C. Reactor químico

Este tipo de equipo tiene como objetivo tener en su interior distintos tipos de reacciones químicas y los ingenieros los diseñan para maximizar la eficiencia y que los productos netos de cierta reacción estén presentes en la salida. [14]

En un Reactor continuo tipo tanque agitado (CSTR), uno o más reactivos líquidos se introducen en un reactor de tanque equipado con un impeller, mientras que el reactor retira residuos. El impulsor se mueve dentro del tanque para asegurar que los reactivos se mezclen adecuadamente. Al dividir el volumen del tanque por el promedio del caudal volumétrico a través del tanque se obtiene la cantidad promedio de tiempo que una cantidad de reactivo pasa en el interior del tanque. [14]

El comportamiento de un CSTR suele aproximarse o modelada por la de un continuo ideal reactor de tanque agitado (CISTR). Esta aproximación es válida para propósitos de ingeniería. El modelo CISTR se utiliza a menudo para simplificar los cálculos.

Los impulsores en los tanques se utilizan para mezclar líquidos. Esto puede ser usado para combinar materiales: sólidos, líquidos o gaseosos. Esta mezcla es muy importante si hay variaciones en condiciones como la temperatura o la concentración. [10]

## D. Torque del motor de inducción AC:

Los caballos de fuerza HP relacionan la velocidad de rotación del eje del motor, entre más HP mayor velocidad del eje. Por definición, 1 HP equivale a  $3.3 \times 10^4$  ft-lb por minuto.

Usando la siguiente ecuación se puede determinar la potencia (hp) entregada por un motor AC de inducción

**Ecuación 3. Torque en lb-ft**

$$T = \frac{HP * 5252}{RPM}$$

En donde:

T = torque en lb-ft

HP = potencia en hp

RPM = Revoluciones por minuto

O en unidades métricas se da por:

**Ecuación 4. Potencia en Watts**

$$P = T * w$$

En donde:

P = potencia en W

T = torque en N\*m

w = velocidad angular en rad/s

Podemos ver en la siguiente tabla que entre mayor sea la velocidad, el motor entrega menos torque y también se observa que entre mayor es el número de polos del motor, mayor es el torque entregado

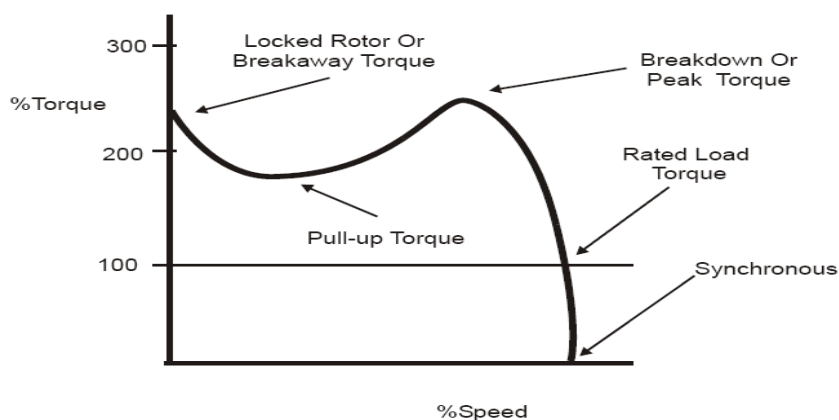
Tabla 2. Relación de torque en función de RPM y número de polos

Número de polos	Sync. RPM (60Hz)	Torque
2	3600	1.5 lb-ft
4	1800	3 lb-ft
6	1200	4.5 lb-ft
8	900	6 lb-ft
10	720	7.5 lb-ft
12	600	8.75 lb-ft

El momento de torsión en el motor al arrancar es mayor que su toque a plena carga, por lo que este arrancará soportando cualquier carga que le introduzca a plena potencia. [7]

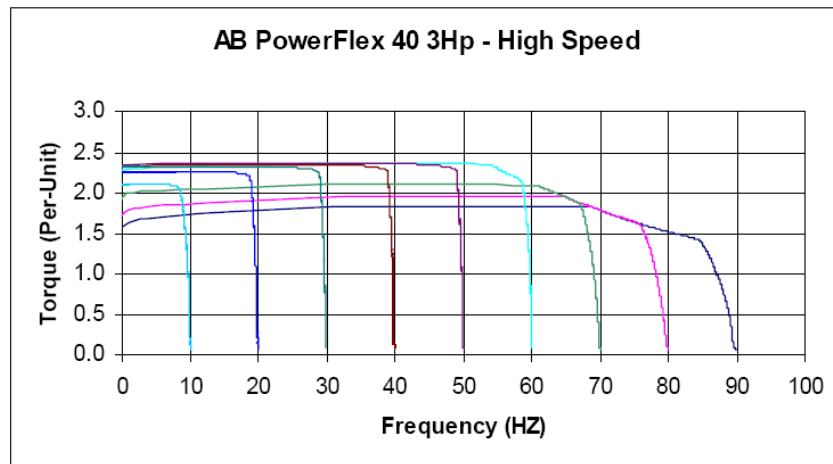
Como se observa en la Ilustración 6. Existe un momento de torsión máximo que no puede sobrepasarse, que es dos o tres veces el momento nominal a plena carga del motor. [7]

Ilustración 6. Curva típica del torque vs frecuencia de un motor de inducción



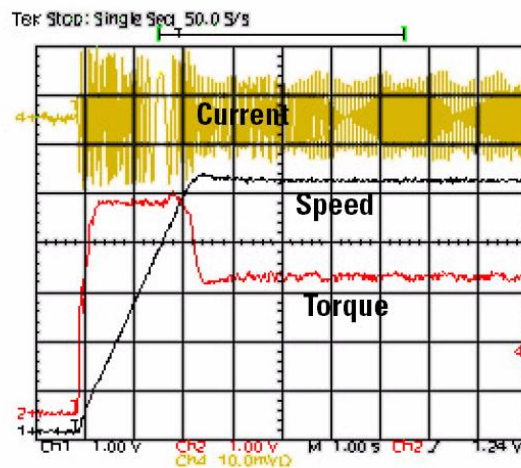
Un Variador de Frecuencia ofrece diferentes curvas frecuencia vs torque, tal y como se muestra en la Ilustración 7. El PowerFlex40 es un variador del tipo sensorless vector control, en donde el variador regula la velocidad y tiene la capacidad de dar un alto torque en las distintas frecuencias. [13]

Ilustración 7. Respuesta del variador Powerflex 40, frecuencia vs torque (13)



La Ilustración 8, demuestra la capacidad del variador de acelerar hasta 150 veces la carga, en donde se puede observar la regulación de corriente, la linealidad en la aceleración y un tiempo de respuesta corto. [13]

Ilustración 8. Comportamiento transiente del variador PowerFlex 40 (13)



## E. Modbus y RS-485:

El estándar de comunicación con el que se realiza la interfaz del variador es RS485, utilizado para la transmisión en serie de datos de alta velocidad a grandes distancias, estándar que encuentra creciente aplicación en el sector industrial. La RS485 está concebida como un sistema Bus bidireccional de hasta 32 participantes. En la Tabla 3 se muestra las distintas características del bus RS485 junto con el estándar RS232

**Tabla 3. Comparación de los estándares RS232 y RS485**

	RS232	RS485
Diferencial	No	Si
Número máximo de variadores	1	32
Modos de Operación	half dúplex full dúplex	half dúplex
Topología de red	PPP	multipunto
Distancia Max. (Acorde Estándar)	15 m	1200 m
Velocidad Max. ( 12 m)	20 kbs	35 Mbs
Velocidad Max. (1.2 Km)	(1 kbs)	100 kbs
Sensibilidad de entrada	±3 V	±200 mV
Voltaje de entrada	±15 V	-7..12 V
Tension de salida max.	±25 V	-7..12 V
Tension de salida min.	±5 V	±1.5 V

El variador powerflex40 soporta algunas funciones del protocolo Modbus este se construye alrededor de mensajes. El formato de estos mensajes es independiente del tipo de interfaz físico. En RS232 están los mismos mensajes usados que en Modbus/TCP sobre Ethernet. El mismo protocolo se puede utilizar sin importar el tipo de conexión. Debido a esto, Modbus da la posibilidad para aumentar fácilmente la estructura del hardware de una red industrial, sin la necesidad de grandes cambios en el software. Un dispositivo puede también comunicarse con varios nodos de Modbus inmediatamente,

incluso si están conectados con diversos tipos del interfaz, sin la necesidad de utilizar diversos protocolos para cada conexión. [6]

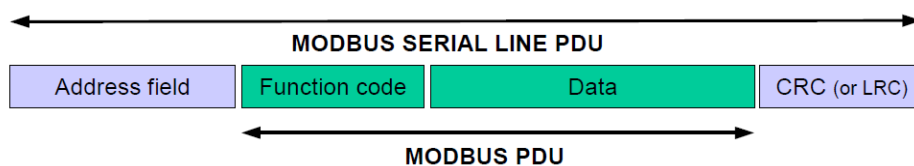
Cada mensaje de Modbus tiene la misma estructura. Cuatro elementos básicos están presentes en cada mensaje y la secuencia de estos elementos es igual para todos.

La comunicación es iniciada por el maestro en la red. Un maestro de Modbus envía determinado mensaje para que el esclavo tome determinada acción.

La dirección en el encabezado del mensaje se utiliza para definir qué dispositivo debe responder a un mensaje. El resto de los nodos en la red de Modbus no hacen caso del mensaje si el campo de dirección no corresponde al suyo.

Tabla 4. Campos y forma de la trama de Modbus

CAMPO	DESCRIPCIÓN
Dirección	Dirección del dispositivo
Function code	Código que define el tipo de mensaje
Data	Datos con información adicional
Error check	Valor numérico que verifica errores en comunicación



1. Transmisión serial de Modbus. Las conexiones seriales de Modbus pueden utilizar dos modos de transmisión básicos, ASCII o RTU (Remote Terminal Unit). El modo de transmisión en comunicaciones seriales define la manera que se cifran los mensajes de Modbus. El formato de Modbus/RTU utiliza la codificación binaria que reduce el tamaño de cada mensaje y permite más intercambio de datos en el mismo intervalo de tiempo. Todos los nodos en un segmento de la red de Modbus deben utilizar el mismo modo de transmisión serial. [9]

Modbus/RTU utiliza bloques de silencio en el tiempo para subdividir la trama de comunicación. Cada mensaje se debe preceder por un bloque con una longitud mínima de 3.5 caracteres. Si un receptor detecta un bloque de por lo menos 1.5 caracteres, asume que está viniendo un nuevo mensaje y el buffer es limpiado. [9]

Tabla 5. Propiedades de Modbus ASCII y RTU

	Modbus/ASCII	Modbus/RTU
Caracteres	ASCII 0..9 and A..F	Binary 0..255
Error Check	LRC Longitud Redundancy Check	CRC Cyclic Redundancy Check
Inicio de Trama	character ':'	3.5 chars silence
Fin de Trama	characters CR/LF	3.5 chars silence
Gaps	1 sec	1.5 times char length
Start bits	1	1
Data bits	7	8
Paridad	even/odd none	even/odd none
Stop bits	1 2	1 2

2. Direcciones de Modbus. La primera información en cada mensaje de Modbus es la dirección del receptor. Este parámetro contiene un byte de información. Las direcciones válidas están en la gama 0..247. Los valores 1..247 se asignan a los dispositivos individuales de Modbus y se utiliza 0 como dirección broadcast.

Los mensajes enviados a la última dirección serán aceptados por todos los esclavos. Cuando responde él utiliza la misma dirección que el master en la petición. De esta manera el master puede ver que el dispositivo está respondiendo realmente a la petición. [9]

Dentro de un dispositivo de Modbus, los holdings registers, las entradas y las salidas se asignan un número entre 1 y 10000. Uno esperaría, las mismas direcciones en los mensajes de Modbus para leer o para fijar valores. Desafortunadamente éste no es el caso. En los mensajes de Modbus las direcciones se utilizan con un valor entre 0 y 9999. Si se desea leer el valor de la salida (coil) 18 por ejemplo, se tiene que especificar el valor 17 en el mensaje de la pregunta de Modbus. Para la entrada y holding registers se debe restar de la dirección de dispositivo para conseguir la estructura apropiada del mensaje de Modbus. Esto conduce a los errores comunes. [6]

La tabla siguiente muestra la gama de direcciones para las coils, las entradas y los registros y la manera en que las direcciones en el mensaje están calculadas dando la dirección actual del dispositivo en el esclavo.

Tabla 6. Rangos de direcciones del protocolo Modbus (10)

Dirección dispositivo	Dirección Modbus	Descripción
1...10000*	address – 1	Coils (outputs)
10001...20000*	address – 10001	Inputs
40001...50000*	address – 40001	Holding registers

El segundo parámetro en cada mensaje de Modbus es el function code. Esto define el tipo de mensaje y el tipo de acción requeridos al esclavo. El parámetro contiene un byte de información.

Cuando un esclavo de Modbus contesta a una solicitud, utiliza el mismo function code que en la petición. Sin embargo, cuando se detecta un error, el MSB del function code se activa. De esa manera el master puede detectar errores. [8]

**Tabla 7. Códigos del estándar Modbus soportado por los variadores de Allen Bradley**

Modbus Function Code (Decimal)	Comando
03	Lectura del Holding Register
06	Escritura de Single Register
16 (10 Hexadecimal)	Escritura de Multiples Registros

**Tabla 8. Registros del variador relacionados con la comunicación**

Parameter	Details
P036 [Start Source]	Set to 5 "RS485 (DSI) Port" if Start is controlled from the network.
P038 [Speed Reference]	Set to 5 "RS485 (DSI) Port" if the Speed Reference is controlled from the network.
A103 [Comm Data Rate]	Sets the data rate for the RS485 (DSI) Port. All nodes on the network must be set to the same data rate.
A104 [Comm Node Addr]	Sets the node address for the drive on the network. Each device on the network requires a unique node address.
A105 [Comm Loss Action]	Selects the drive's response to communication problems.
A106 [Comm Loss Time]	Sets the time that the drive will remain in communication loss before the drive implements A105 [Comm Loss Action].
A107 [Comm Format]	Sets the transmission mode, data bits, parity and stop bits for the RS485 (DSI) Port. All nodes on the network must be set to the same setting.

Los valores internos en un dispositivo de Modbus se almacenan en holding registers. Estos registros son de dos bytes y se pueden utilizar para varios propósitos. Algunos registros contienen los parámetros de la configuración y otros se utilizan para

devolver los valores medidos por un sensor por ejemplo. Los registros en un dispositivo compatible de Modbus comienzan a contar en 40001. Están direccionados en la estructura del mensaje de Modbus con direcciones que empiezan desde 0. La función 03 de Modbus se utiliza para solicitar uno o más valores de un holding register en un dispositivo. Solamente un dispositivo esclavo se puede direccionar en una solicitud. No se puede implementar solicitudes broadcast con la función 03. [9]

Tabla 9. Descripción de los 8 bytes de Modbus (11)

Byte	Valor	Description
1	1...247	Dirección del esclavo
2	3	Function code
3	0...255	Dirección de inicio, high byte
4	0...255	Dirección de inicio, low byte
5	0...255	Número de registros, high byte
6	0...255	Número de registros, low byte
7(...8)	LRC/CRC	Chequeo de error

Después de procesar la solicitud, el esclavo de Modbus devuelve los 16 bits solicitados del holding register. El primer byte de datos contiene el más significativo y el segundo el byte bajo del registro. [8]

El mensaje de la respuesta de Modbus comienza con la dirección de dispositivo esclavo y el código 03 de la función. El byte siguiente es el número de los bytes de datos que siguen. Este valor es dos veces el número de registros devueltos. Un último byte que se añade para que el host compruebe si ocurrió un error de la comunicación. [8]

Como se observa en la Tabla 10 el variador tiene en su registro 8192, 16 bits para modificar su estado. El apagado-encendido, la forma de aceleración-desaceleración y la referencia de la velocidad de operación.

**Tabla 10. Registro 8192 del variador PowerFlex 40**

Function Code 06 writes to register address 8192 (Logic Command). P036 [Start Source] must be set to 5 “RS485 (DSI) Port” in order to accept the commands. Use Function Code 03 to read this register.

Logic Command			
Address (Decimal)	Bit(s)	Description	
8192	0	1 = Stop, 0 = Not Stop	
	1	1 = Start, 0 = Not Start	
	2	1 = Jog, 0 = No Jog	
	3	1 = Clear Faults, 0 = Not Clear Faults	
	5,4	00 = No Command	
		01 = Forward Command	
		10 = Reverse Command	
		11 = No Command	
	6	Not Used	
	7	Not Used	
	9,8	00 = No Command	
		01 = Accel Rate 1 Enable	
		10 = Accel Rate 2 Enable	
		11 = Hold Accel Rate Selected	
	11,10	00 = No Command	
01 = Decel Rate 1 Enable			
10 = Decel Rate 2 Enable			
11 = Hold Decel Rate Selected			
14,13,12	000 = No Command		
	001 = Freq. Source = P036 [Start Source]		
	010 = Freq. Source = A069 [Internal Freq]		
	011 = Freq. Source = Comms (Addr 8193)		
	100 = A070 [Preset Freq 0]		
	101 = A071 [Preset Freq 1]		
	110 = A072 [Preset Freq 2]		
	111 = A073 [Preset Freq 3]		
15	Not Used		

### III. JUSTIFICACIÓN

En los últimos años la industria se ha visto en la necesidad de automatizar sus equipos. El uso de sistemas computarizados para controlar los procesos es una necesidad actual, ya que beneficia a los operadores asistiéndolos físicamente; reduciendo el error del operador notablemente. Esto es muy deseable porque se gana en la eficacia con la que se desarrolla el proceso.

Lo que se busca es hacer uso de equipo para la automatización en el laboratorio de operaciones unitarias y así contar con mejoras en el aprendizaje del estudiante, que se involucra en procesos automatizados, con los que seguramente interactuará en la industria.

El proyecto se enfoca en utilizar variadores de frecuencia en distintos motores, logrando un ahorro en el consumo de energía, controlando el arranque y la velocidad de los motores, traducido en la industria en un mejor manejo sobre la producción.

## IV. OBJETIVOS

### A. Generales

1. Monitorear y dirigir de forma computarizada, procesos que impliquen la utilización de motores.
2. Mejorar el aprendizaje de los estudiantes que operan los equipos, mediante el desarrollo tecnológico de estos basándose en estándares industriales

### B. Específicos

1. Poner en marcha los variadores de frecuencia y desarrollar el control automatizado de los equipos.
2. Tener disponibles las variables solicitadas por los usuarios de los distintos motores, desarrollando una interfaz grafica para que el usuario manipule los variadores desde la estación de trabajo en la planta.
3. Formar una red con los variadores de frecuencia, utilizando un protocolo sobre el bus RS485.

## V. METODOLOGÍA

Se recurrió como principal apoyo al manual de usuario técnico de los variadores de frecuencia Allen Bradley de la línea PowerFlex, para su instalación adecuada y colocación de los parámetros en cada uno de los equipos en los cuales se instaló.

Se estudiaron los lineamientos y normas del protocolo Modbus provenientes de fuentes confiables tanto en el internet como en libros para su implementación. Se investigaron e hicieron consultas, directamente a los operadores de los equipos para cumplir con sus requerimientos.

Los datos obtenidos de todos los motores se envían cuando se solicita a una hoja de cálculo por medio de acceder a un archivo generado por el programa. Y así poder estudiar el comportamiento de la variable de interés y tomar decisiones en base a ellos.

La aplicación de este proyecto está orientada al estudiante que se empleara en la industria alimenticia y en la industria química pues debemos tomar en cuenta que en estos aparatos industriales donde se implemento la utilización de los variadores de frecuencia; con el fin de obtener equipos automatizados.

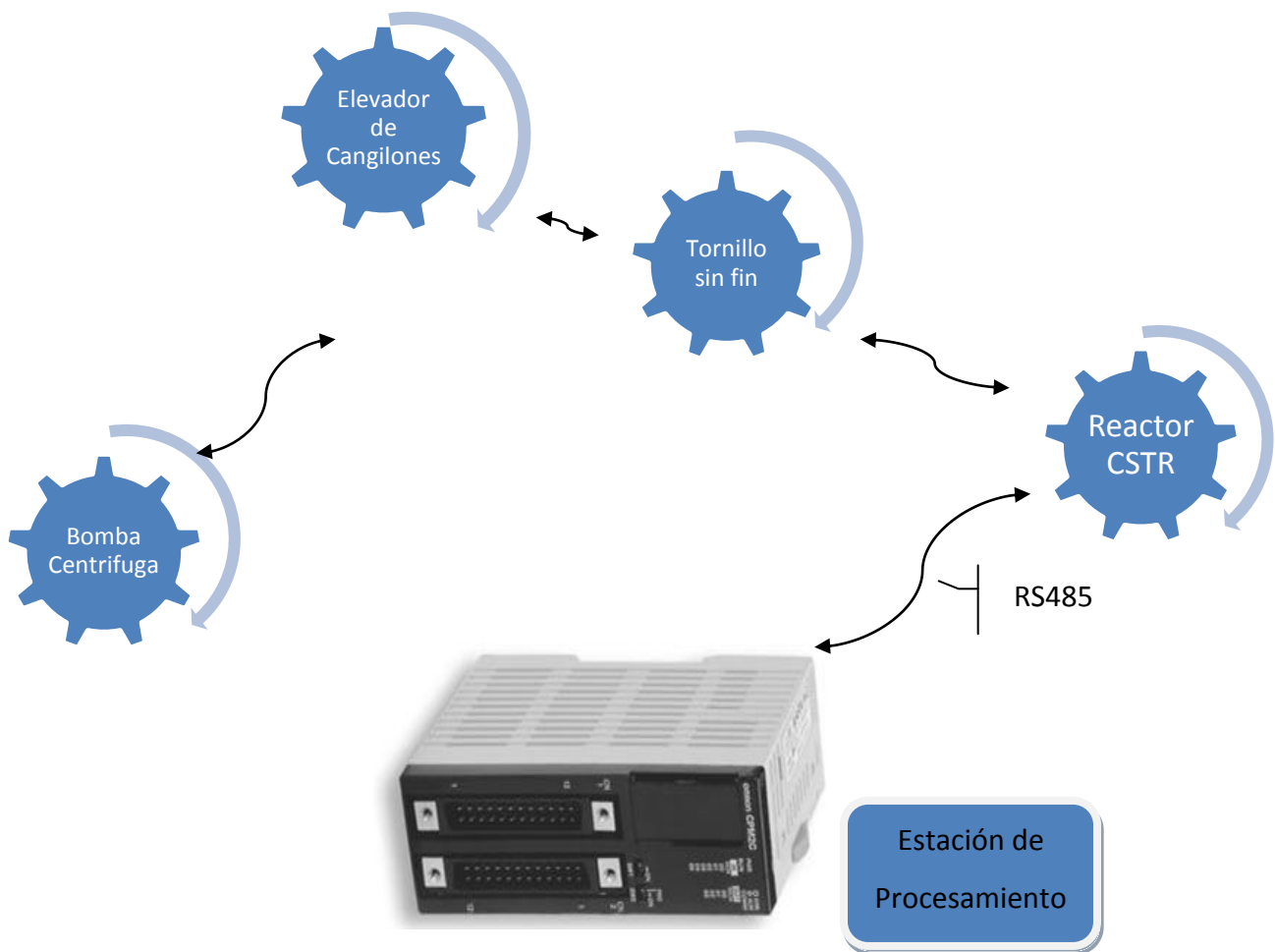
Con la implementación de este monitoreo, es deseable ver resultados comparativos en los gastos de consumo eléctrico. Para ello se necesario calcular la potencia ahorrada en una determinada cantidad de tiempo y compararla al escenario cuando no se contaban con variadores instalados.

Adicionalmente se verificó la funcionalidad de los equipos en el proceso al que se somete.

## VI. RESULTADOS

### A. Diagrama general del módulo de variadores de frecuencia

Ilustración 9. Diagrama lógico del proyecto



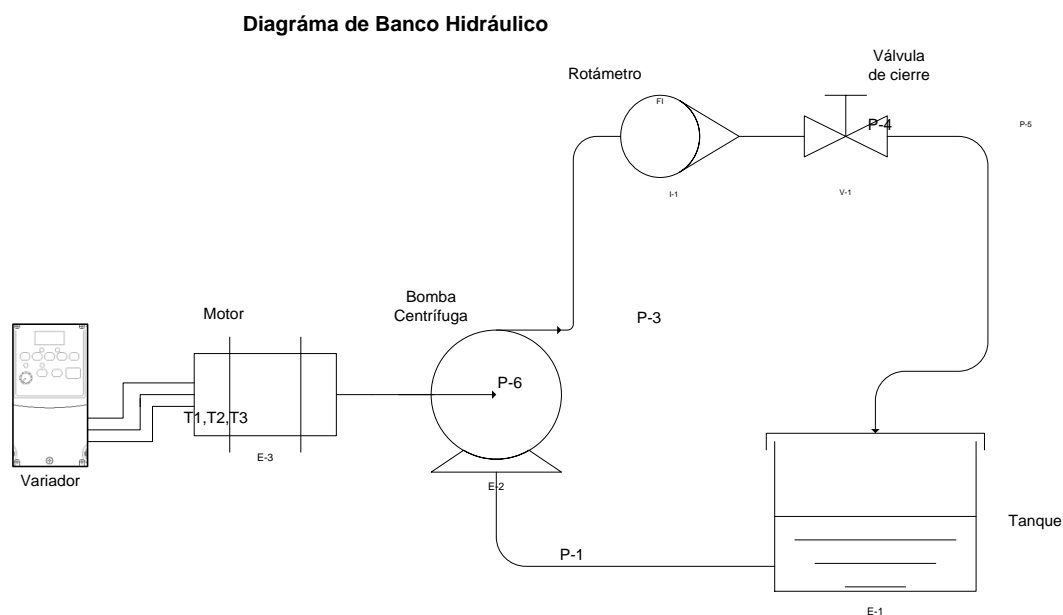
Se observa en la Ilustración 9 los distintos equipos que se interconectan: El Reactor químico, el tornillo sin fin que a su vez es alimentado en el proceso por el

elevador de cangilones y la bomba centrífuga del banco hidráulico. Estos cuatro equipos conforman la red. Estos van a una estación piloto que es el nodo maestro de la red.

## B. Banco hidráulico

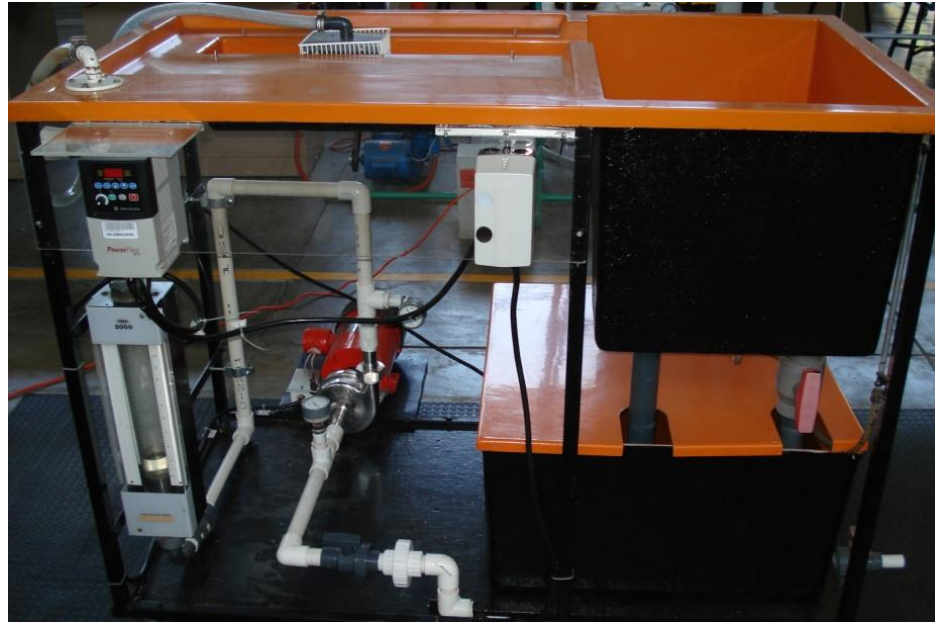
La Ilustración 10, corresponde al diagrama del banco hidráulico en donde se observan los distintos elementos que conforman el sistema, entre ellos el variador y el motor que conjunto que permite la variación de los parámetros que interesan observar, la bomba centrífuga que provee de propulsión al flujo y el rotámetro que permite ver el caudal del líquido que lo atraviesa.

Ilustración 10. Diagrama del Banco Hidráulico



El equipo diagramado anteriormente se muestra en la Ilustración 11. Se observa en esta figura el variador instalado en el banco hidráulico sobre la placa de fibra de vidrio, en la parte inferior izquierda se observa el rotámetro y al fondo la bomba centrífuga.

Ilustración 11. Banco Hidráulico



Se tomaron mediciones al variador instalado en el banco hidráulico, en donde se puede observar varios parámetros de interés descritos en la Tabla 11.

La primera columna es la frecuencia en la que opera el variador, se optó por muestrear cada 5 Hz con el objetivo de obtener 12 muestras para llegar a la frecuencia máxima de operación de los motores con los que contamos.

La corriente de variador, el voltaje de variador y la corriente de torque son valores que nos proporciona el driver en los respectivos registros.

La corriente de línea y el voltaje de línea fueron medidos directamente en el equipo con un multímetro.

Los últimos tres parámetros, que son la potencia, las revoluciones por minuto (RPMs) y el torque fueron mediciones calculadas indirectamente con los datos obtenidos directamente y mencionados en párrafos anteriores. En la sección de análisis de resultados se describe como se calcularon estas variables.

Tabla 11. Mediciones tomadas al variador del Banco Hidráulica

Frec. (HZ)	Corriente de Variador	Corriente de línea	Voltaje de Variador	Voltaje de línea	Corriente de torque	Potencia (W)	HP	RPM	Torque lb-ft
5	1.85	0.578	20.2	174.3	0.43	8.686	0.011643	300	0.203837676
10	2.05	1.095	38.5	174.8	0.43	16.555	0.022192	600	0.194251251
15	2.13	1.453	56.8	175.8	0.43	24.424	0.03274	900	0.191055776
20	2.2	1.743	75.9	176.1	0.77	58.443	0.078342	1200	0.342876046
25	2.29	1.954	94.6	176.5	0.95	89.87	0.120469	1500	0.421802717
30	2.42	2	113.8	176.8	1.13	128.594	0.172378	1800	0.502960745
35	2.6	2.08	133	177.3	1.35	179.55	0.240684	2100	0.601938338
40	2.8	2.41	152.3	177.6	1.63	248.249	0.332773	2400	0.728219252
45	3.07	2.71	171.3	177.3	2.1	359.73	0.482212	2700	0.937991242
50	3.42	3.12	190.2	176.4	2.5	475.5	0.637399	3000	1.115873995
55	3.86	3.53	210.4	173.1	2.95	620.68	0.832011	3300	1.324157673
60	4.43	4.17	230	168.4	3.71	853.3	1.143834	3600	1.668726393

Se tomaron muestras de valores de corriente y voltaje, estas se incluyen de forma gráfica a continuación:

Ilustración 12. Corriente de salida del parámetro d003 del variador

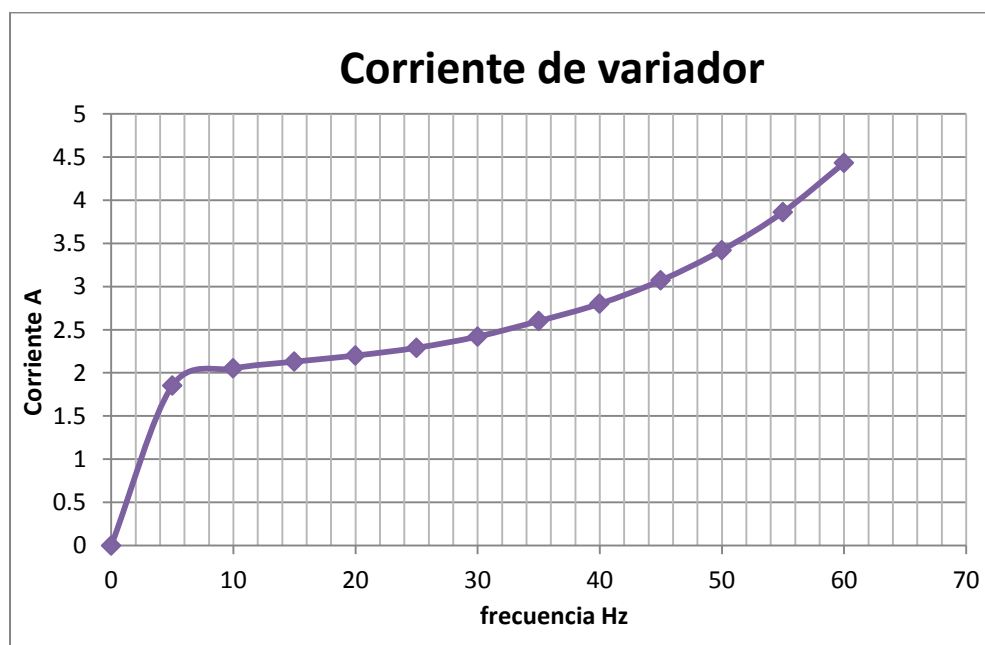


Ilustración 13. Corriente de salida presente en una de las líneas del motor.

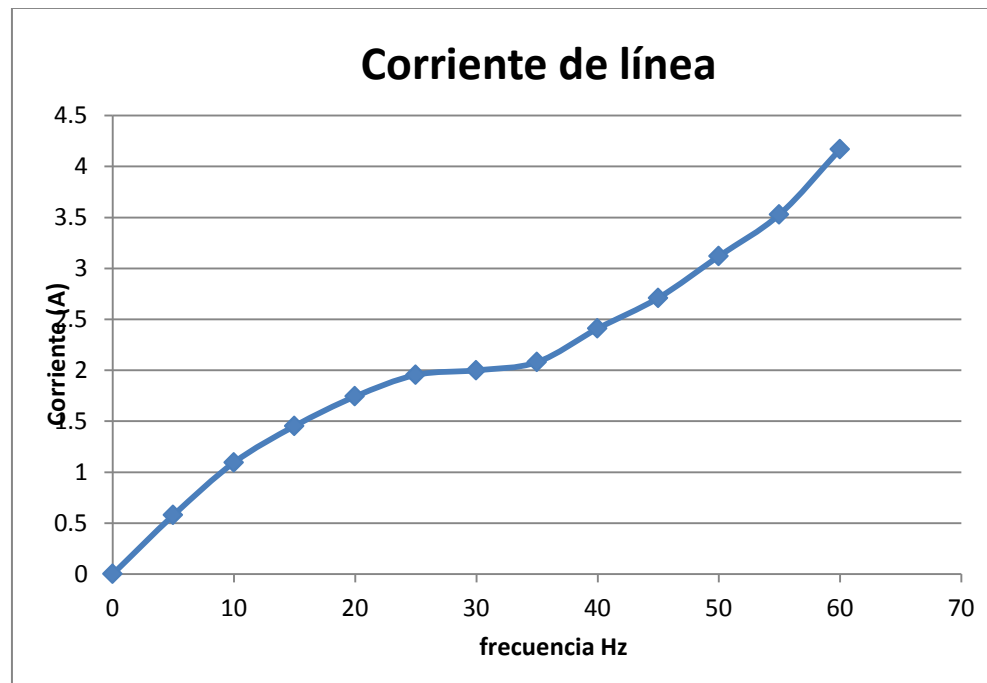


Ilustración 14. Voltaje de salida presente en una de las líneas del motor

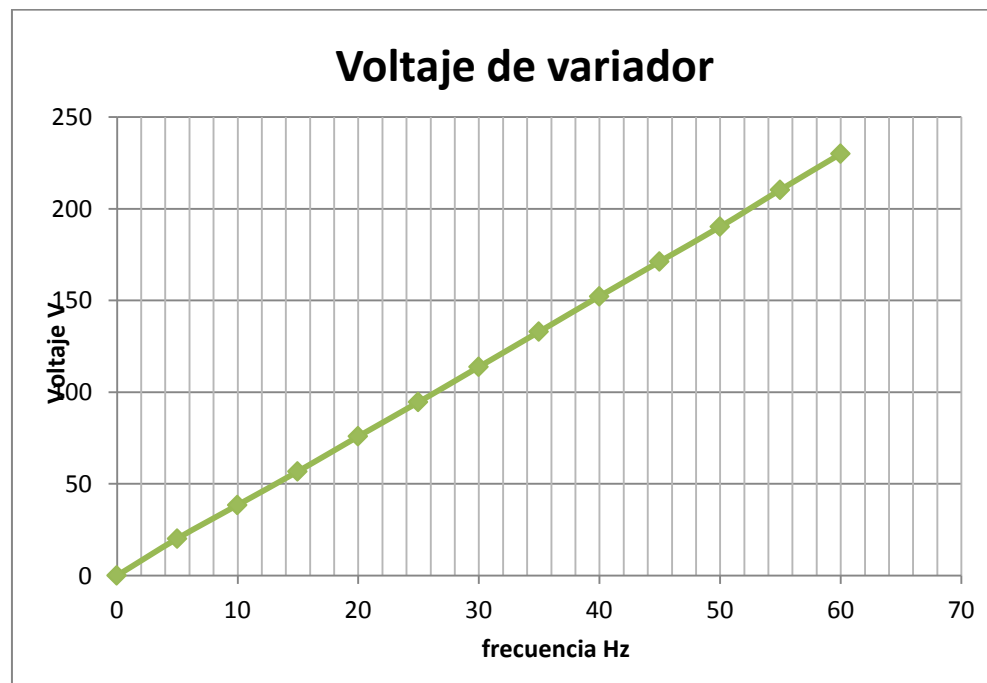


Ilustración 15. Voltaje de salida presente en una de las líneas trifásicas

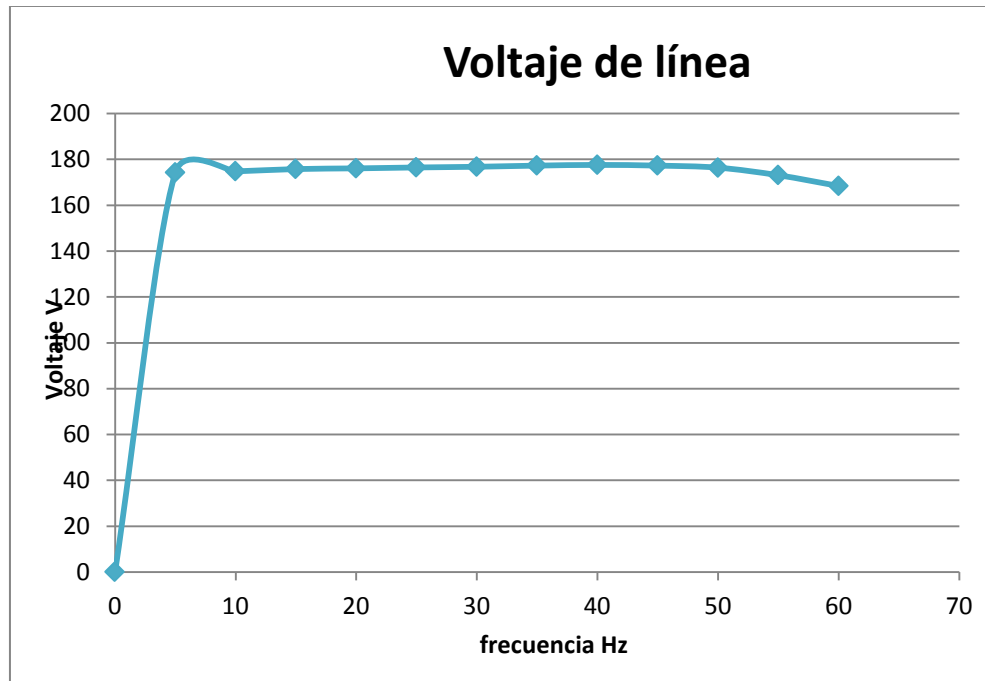


Ilustración 16. Frecuencia vs corriente de torque

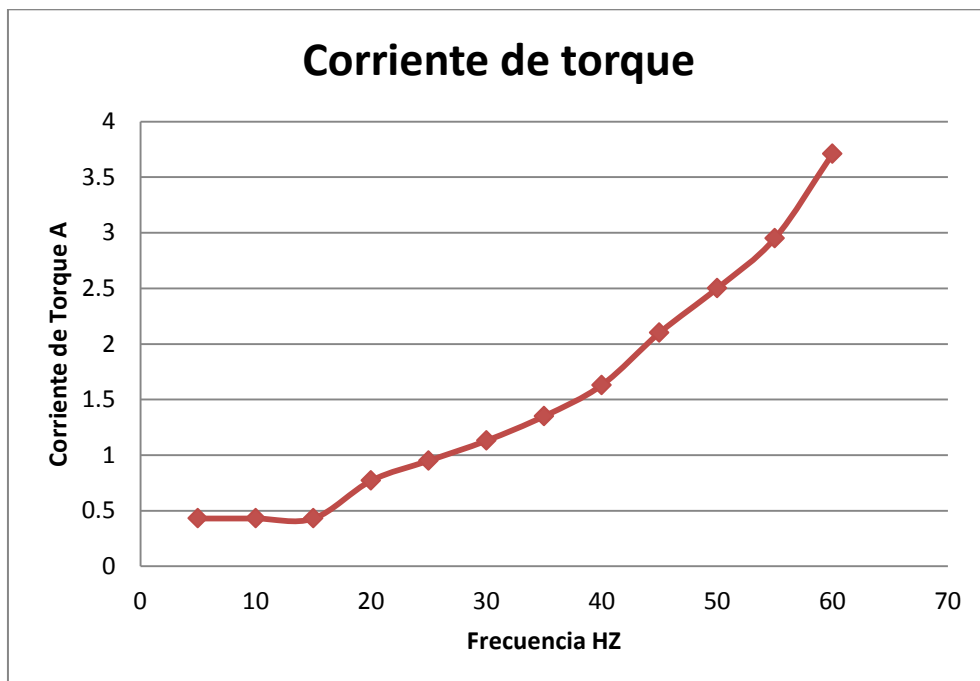
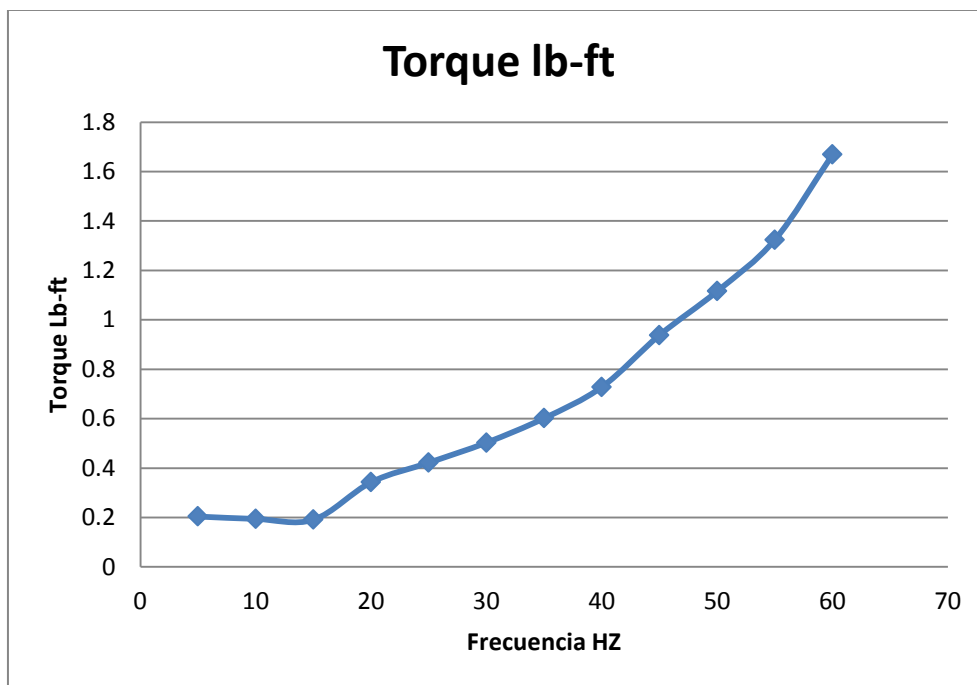


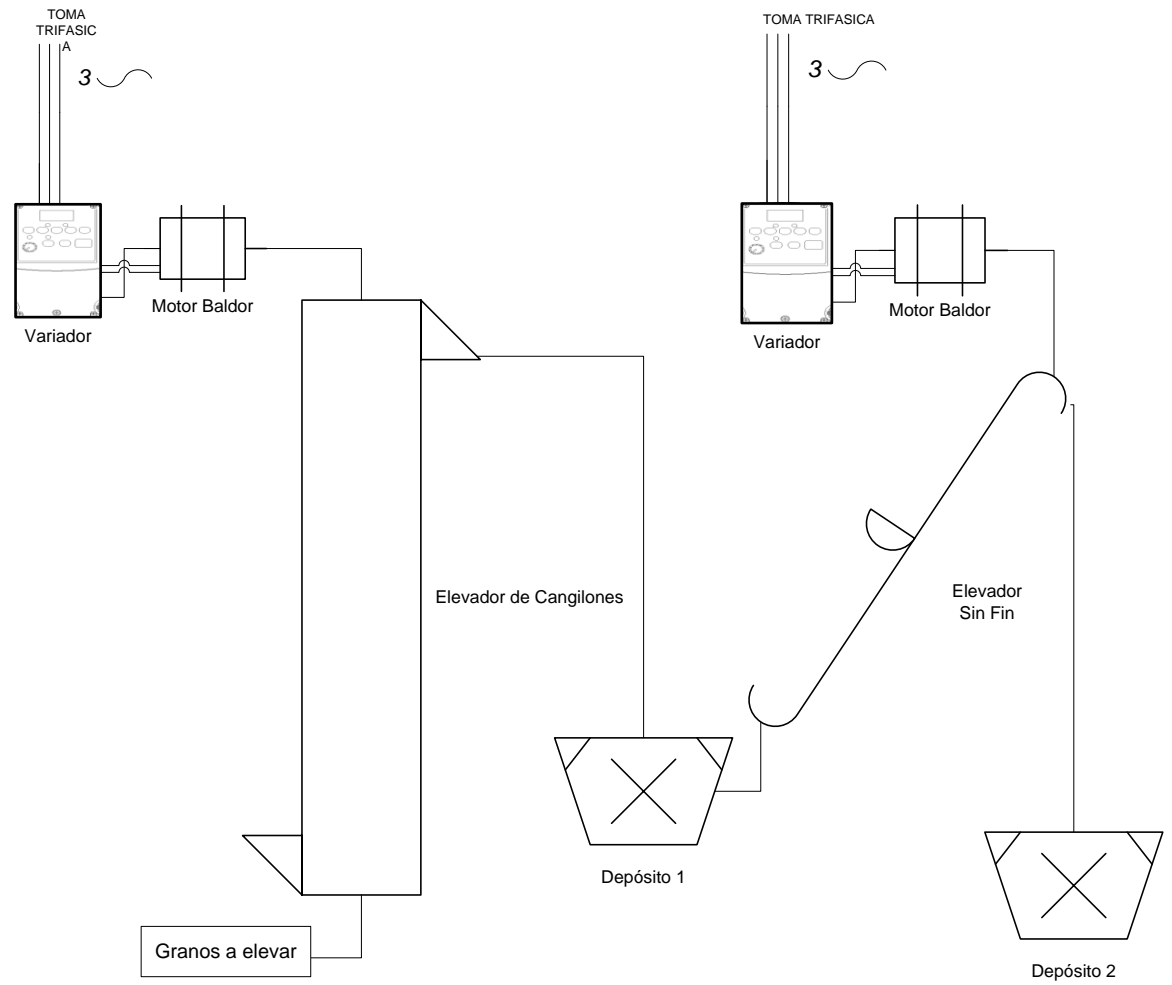
Ilustración 17. Frecuencia vs torque



### C. Elevador de cangilones y de tornillo sin fin

En la Ilustración 18 se diagrama el sistema de elevador de cangilones en donde se está controlando el motor del equipo para modificar la rapidez en la entrega del producto que se lleva en los cangilones. El otro equipo de elevación es el tornillo sin fin que también se muestra en la ilustración, se le adicionó el variador de frecuencias con el mismo propósito. Ambos son motores trifásicos encargados de llevar el grano hasta un depósito específico.

Ilustración 18. Diagrama de elevadores



El sistema que muestra la Ilustración 19 es el elevador de cangilones con el equipo instalado.

Ilustración 19. Variador instalado en el elevador de cangilones



Se instaló el variador en el elevador de cangilones, los siguientes son los datos tomados en el sistema:

Tabla 12. Datos tomados del elevador de cangilones

Dato #	Frecuencia	Voltaje Var.	Voltaje línea	Corriente Var.	Corriente línea
1	5	24	215	1.98	0.1
2	10	41	215	2.09	0.62
3	15	58	215	2.1	0.9

Tabla 12. Continuación datos tomados del elevador de cangilones

4	20	78	215	2.16	1.17
5	25	96	215	2.19	1.25
6	30	115	215	2.19	1.35
7	35	130	215	2.2	1.42
8	40	149	215	2.2	1.48
9	45	168	215	2.22	1.5
10	50	186	215	2.24	1.54
11	55	205	215	2.25	1.57
12	60	222	215	2.26	1.6

Ilustración 20. Voltaje de variador vs frecuencia

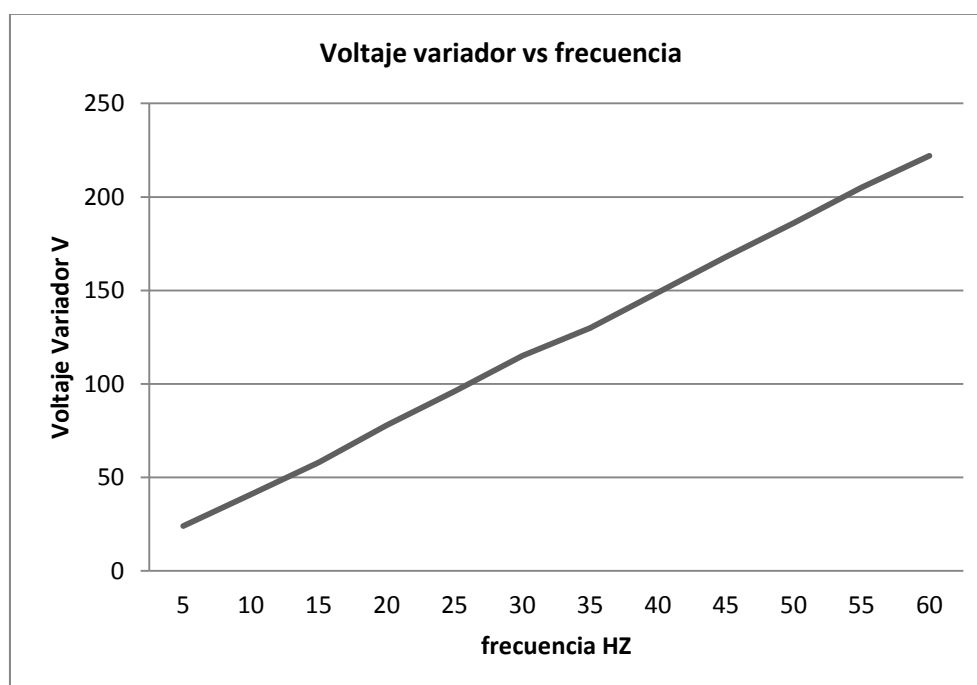


Ilustración 21. Corriente de variador vs frecuencia

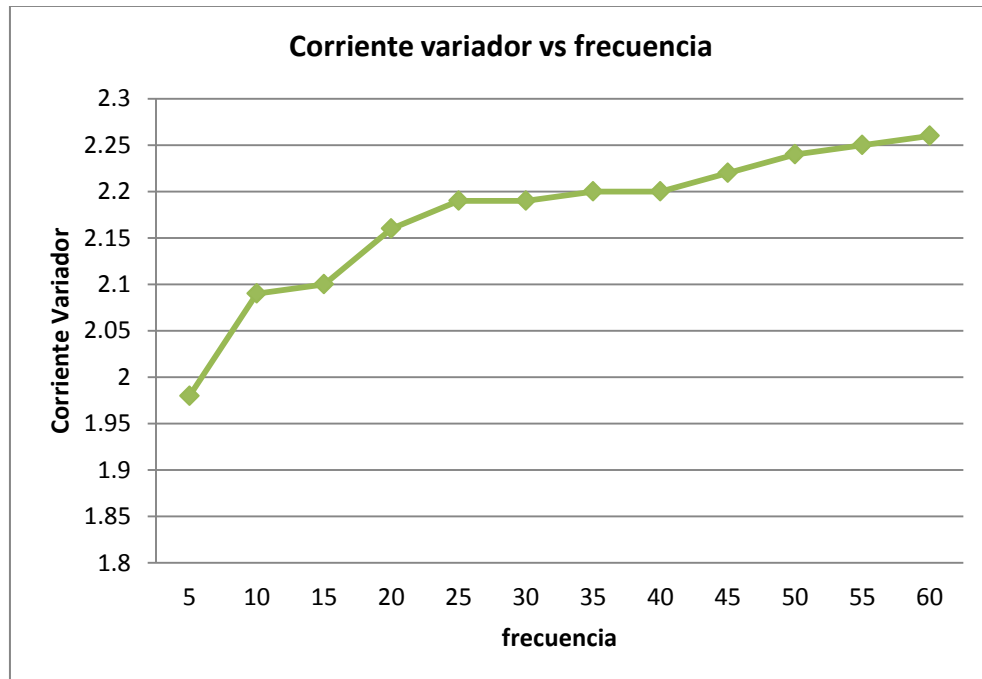


Ilustración 22. Corriente de línea vs frecuencia

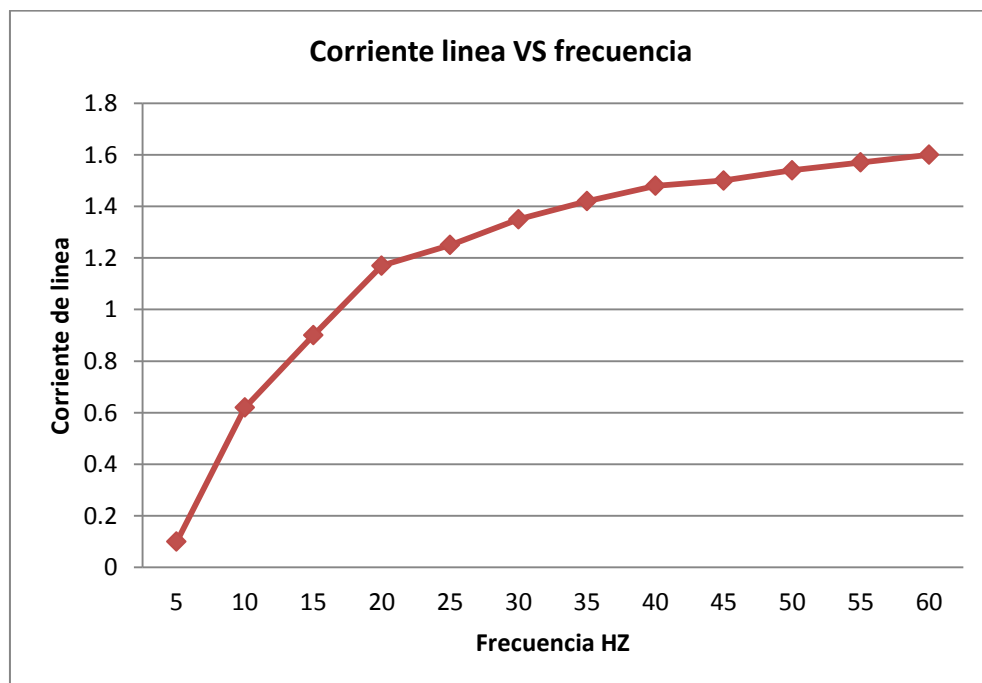
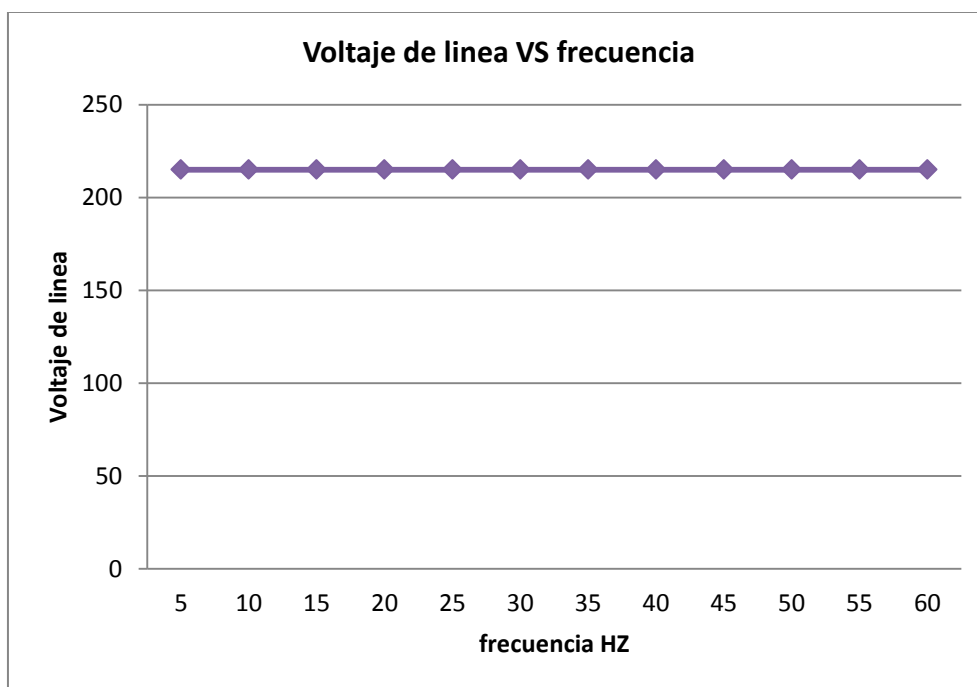


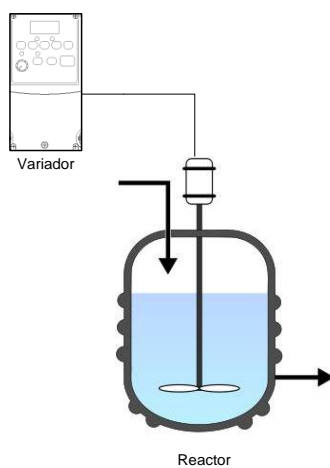
Ilustración 23. Voltaje de línea vs frecuencia



## D. Reactor químico

El diagrama de la Ilustración 24 es el que corresponde al equipo del reactor químico el motor está directamente conectado al impulsor que se encarga de la mezcla apropiada para obtener un proceso de homogenización adecuado.

Ilustración 24. Diagrama del reactor CSTR



La siguiente es una imagen del reactor y se observa la caja del variador en donde está instalado adicionalmente un indicador de temperatura.

Ilustración 25. Reactor químico y variador



## E. Especificaciones de motores

El equipo que se instaló es de acuerdo a la capacidad de los motores y los siguientes requerimientos:

Tabla 13. Especificaciones de motores

Especificaciones	Elevador de Cangilones	Bomba Centrífuga	Reactor Químico
Modelo	<i>Baldor (M3542)</i>	<i>Reliance Triclover-C114</i>	<i>Wwg NBR7094</i>
Horsepower	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{3}{4}$
Voltaje	208-230/460 v	208-230/460 v	208-230/460 v
Fases	3	3	3
Corriente	3.2-3.0 / 1.5 A	4.62 / 2.31 A	4.43/2.56/2.22
RPM	1725	3450	1770

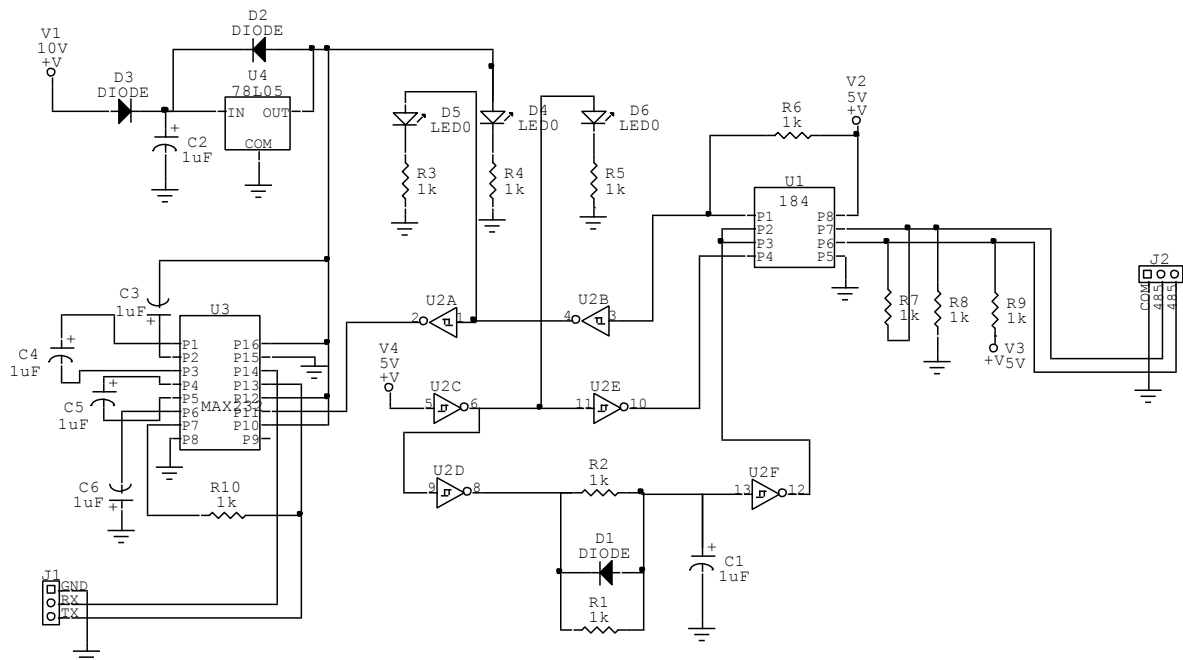
Ilustración 26. Motor Baldor M3542 instalado en el sistema de cangilones (4)



## F. Convertidor RS232 – RS485

Se necesitó convertir la interfaz serie RS232 de la máquina de la planta piloto, con la interfaz industrial del protocolo RS485. Para esto se obtuvo un circuito electrónico para tal efecto, con transmisión de datos half-duplex, que convirtiera ambos protocolos de manera asíncrona y bidireccionalmente.

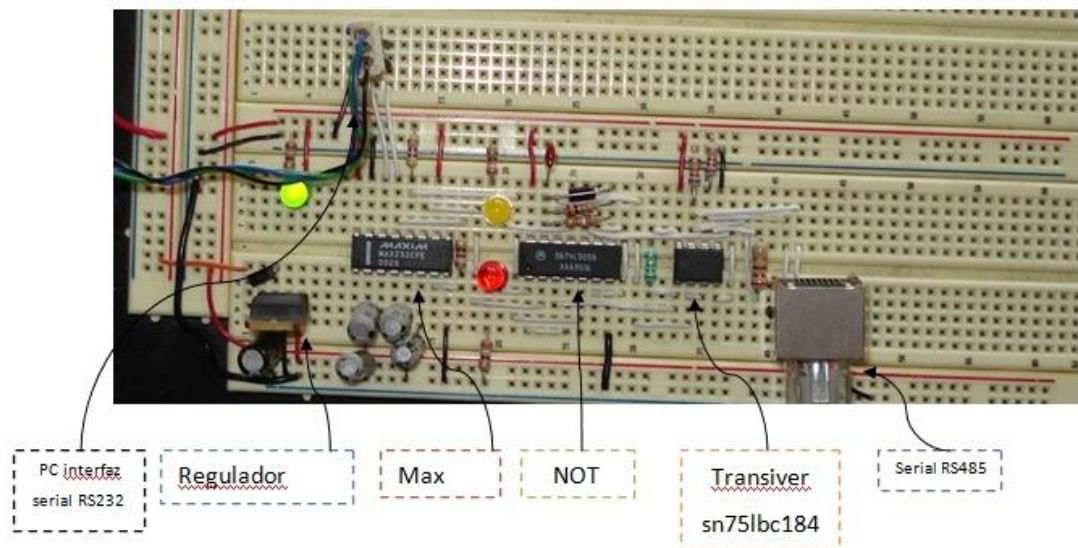
Ilustración 27. Diagrama del circuito de conversión de RS232 a RS485 (12)



Este circuito llenó los requisitos para la utilización en la red, requiere de un voltaje de 9 voltios DC que se provee por un transformador y se regula a 5 voltios DC con el integrado 7805. Consume menos de  $\frac{1}{2}$  Watt y el integrado sn75lbc184 es el transiver dedicado encargado de la conversión, cuenta protección para transientes de voltaje y tiene una velocidad máxima de operación de 115.2kbps.

Se construyó el circuito encargado de convertir los protocolos RS232 a RS485, inicialmente en etapa de pruebas se construyó en un proto-board, para poder ponerlo en funcionamiento y probarlo en la red, este se observa en la Ilustración 28.

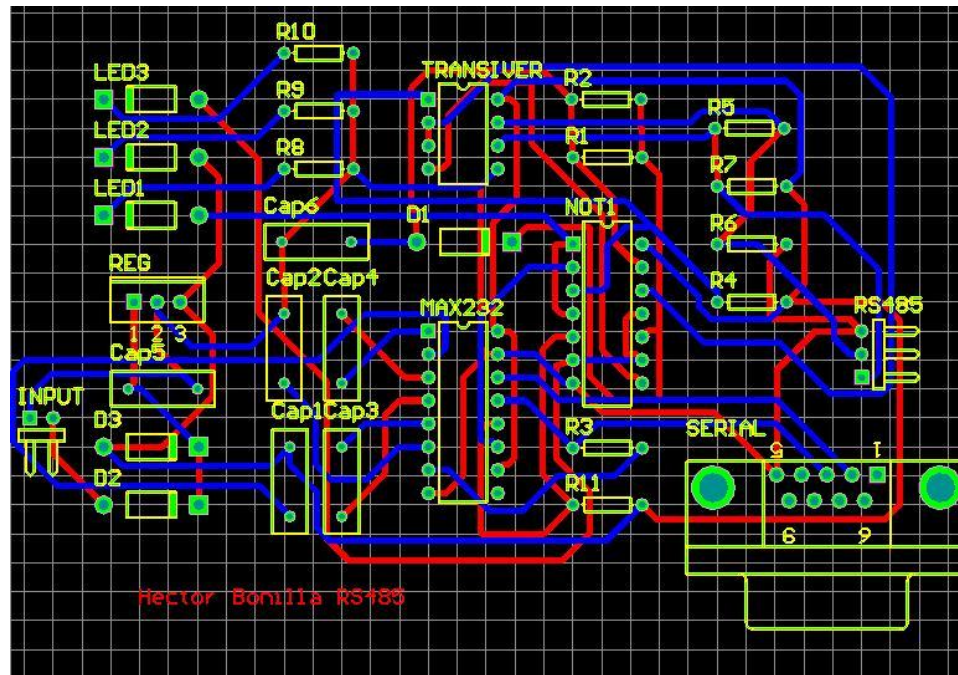
Ilustración 28. Circuito convertidor prototipo



El circuito se redibujó en un simulador para generar los archivos necesarios y realizar el circuito PCB. Que se muestra en la Ilustración 29.

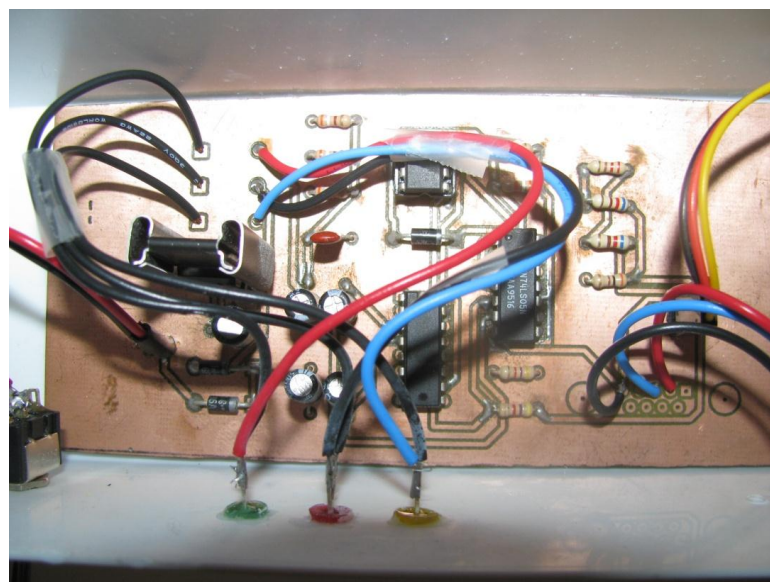
Altium es el simulador que se utilizó, esta poderosa herramienta tiene la capacidad de hacer las rutas automáticamente en las capas que el diseñador disponga, en este circuito convertidor se generó los archivos Gerber tanto de la top y botton layer como el archivo para el barreno.

Ilustración 29. PCB del circuito convertidor de RS232 a RS485



Posterior a ello se obtuvo la placa impresa mediante una maquina fresadora CNC y finalmente se soldaron los dispositivos en ella y resultó un circuito que se adecuó en una caja protectora como se observa en las Ilustraciones 30 y 31.

Ilustración 30. Circuito convertidor de RS232 a RS485



## G. Diagramas de flujo de la programación de Modbus

Ilustración 31. Diagrama de flujo de la lectura de los holding registers.

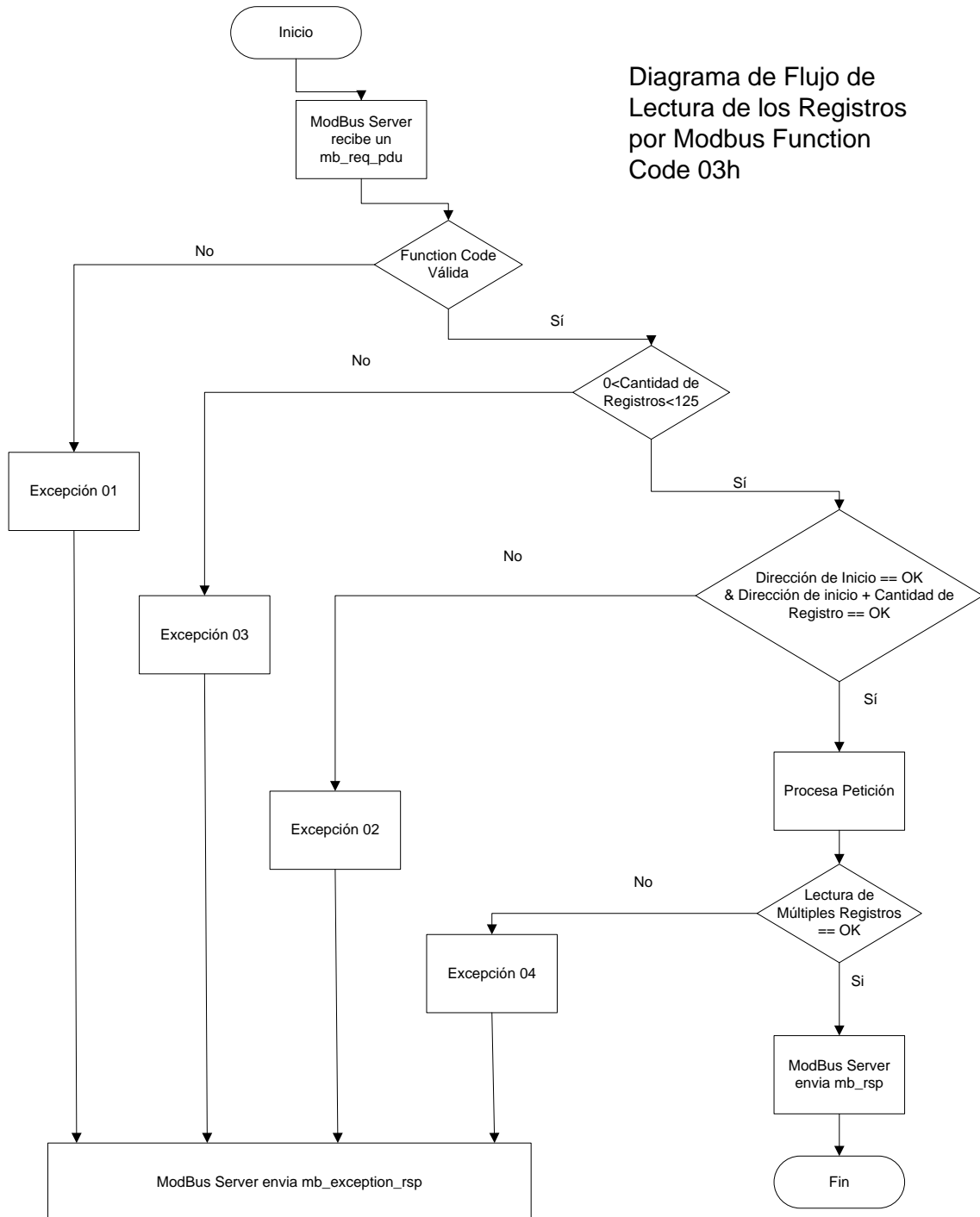
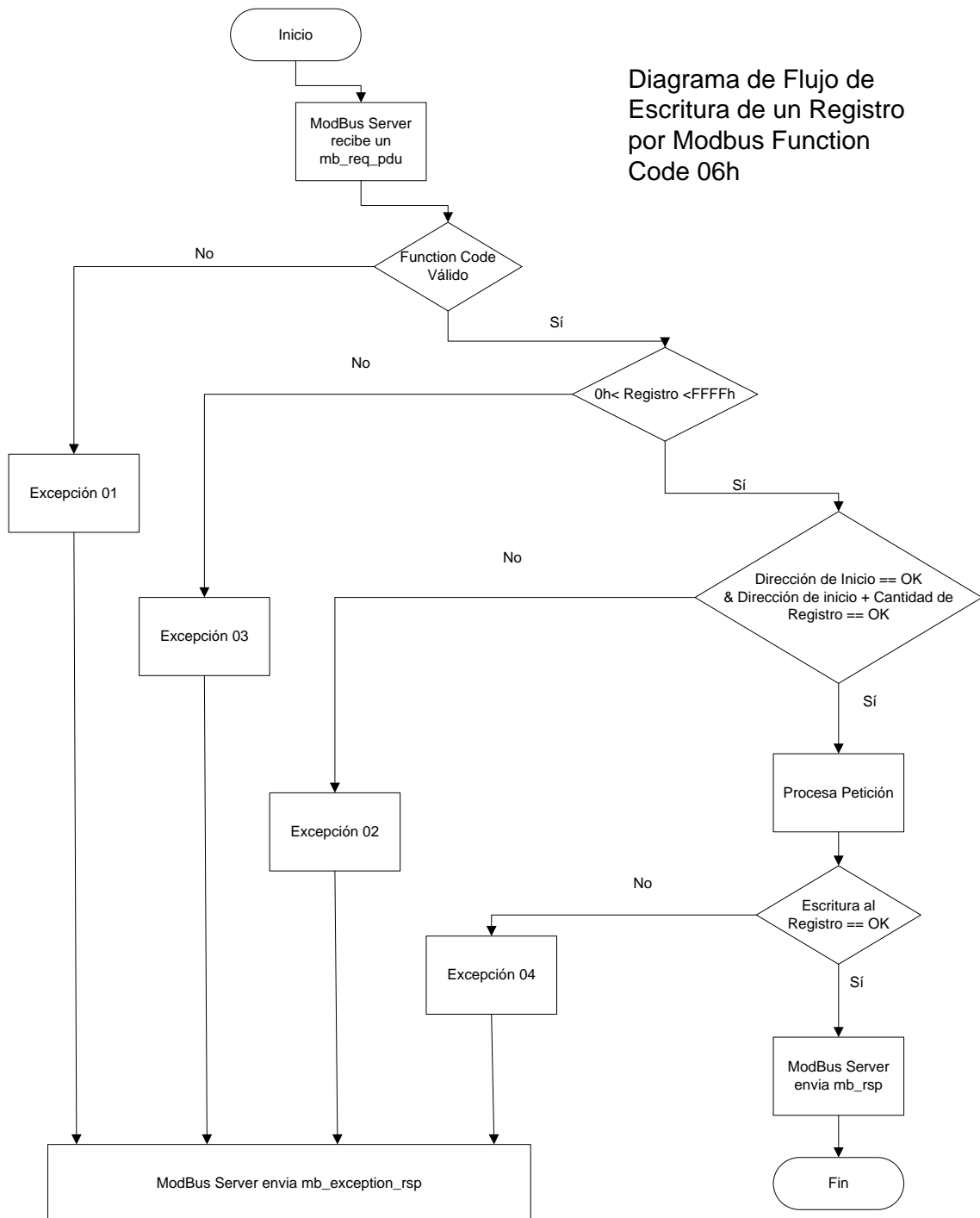


Ilustración 32. Diagrama de flujo de la escritura de los registros.



En la interfaz de usuario del maestro de la red de variadores, el programa detecta automáticamente los puertos disponibles de la computadora en donde se está ejecutando. Para la información del puerto se habilitan dos Combo Boxes en donde se colocan el puerto COM disponible y la velocidad de transferencia: 38.4Kbps (que es la máxima velocidad que permite el circuito convertidor). Adicionalmente se puede colocar las siguientes tasas de transferencia 19.2Kbps, 9.6Kbps. Dependiendo de la cantidad de datos a en la trama es deseable aumentar la tasa de transferencia si se incrementa los datos solicitados.

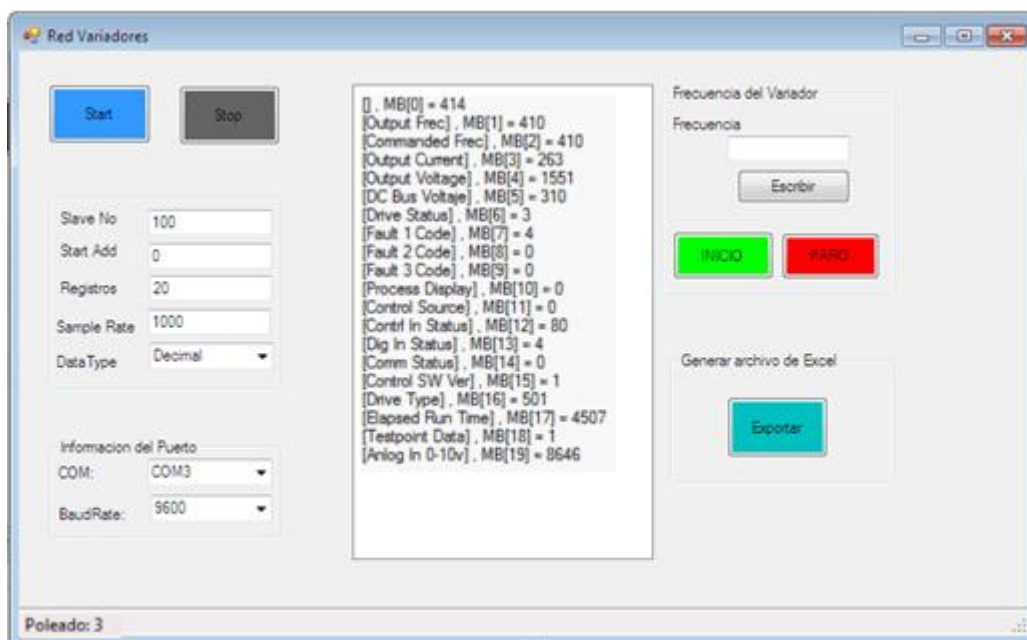
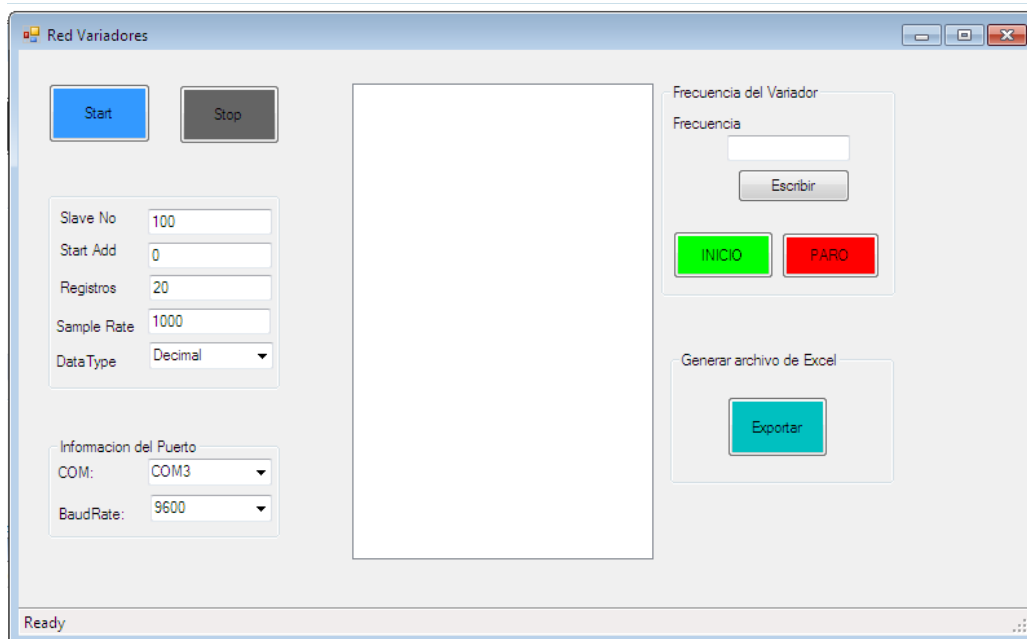
Después se llenan los campos que corresponden a la información del protocolo. Tal es el caso de la dirección del esclavo "Slave No" que es un campo necesario para la identificación del dispositivo al cual se está comunicando, este valor es establecido en el variador en el parámetro A104, el registro en donde se inicia la lectura "Start Addr" que es otro campo necesario para poder solicitar los datos deseados y la cantidad de registros que se leerán "Registros".

Posteriormente la tasa de muestreo que se establece en milisegundos "Sample Rate", por ejemplo, si se desea muestrear cada segundo se coloca 1000 y el tipo de dato con el que se quiere visualizar los datos "Data Type"

Al dar click en el botón Start el programa inicia a desplegar en el ListBox central los registros que se solicitan tal como se observa a continuación en la interfaz de usuario de la Ilustración 32.

Adicionalmente se tiene la opción de escribir a determinado registro para poder encender o apagar el variador, así como también modificar su frecuencia de operación.

Ilustración 33. Lectura de algunos registros por el Maestro de la red



## VII. ANÁLISIS DE RESULTADOS

### A. Mediciones en los variadores instalados y gráficas:

En la Tabla 10 en donde se tomaron mediciones del banco hidráulico se observan datos que son obtenidos indirectamente con el objetivo de calcular el torque en lb-ft que ocupa la última columna, estos se detallan a continuación:

La potencia se obtuvo con la relación

$$P = |V| * |I| * \cos(\theta)$$

En donde V corresponde al voltaje de la línea, I es la corriente de torque, provista por el variador y  $\cos(\theta)$  es el factor de potencia que se asumió igual a 1.

La potencia en HP se convirtió con la siguiente relación:

$$1\text{HP} = 746\text{watts}$$

Se observa que la máxima potencia obtenida (a 60 Hz) fue de 1.14HP que equivale al 76% de la potencia del motor (1.5HP).

Las RPMs se hallaron multiplicando la frecuencia por 60.

Finalmente se calculó el torque en lb-ft con la ecuación 3

$$T = \frac{\text{HP} * 5252}{\text{RPM}}$$

Se observa en las gráficas de corriente de variador vs frecuencia que se mantiene un crecimiento dentro de los límites de corriente de ambos motores. Se

observa una corriente máxima cercana a los 4.5A en el caso de la bomba centrífuga en donde la corriente de placa del motor es 4.62A.

En el caso del elevador de cangilones se observa una variación en la corriente sin un patrón definido, eso es debido a que no se le estaba sometiendo a la misma carga en todas las mediciones ya que por tratarse de granos, los cangilones no llevan exactamente la misma cantidad de grano. Pero si es posible observar; como es de esperarse, que la corriente presenta crecimiento al aumentar la frecuencia. Mostrando un valor máximo de 2.26A, casi 1A menos que su corriente de máxima potencia.

En el caso del voltaje del variador (que es el voltaje que llega a cada una de las líneas del motor) se observa tanto en la gráfica de la bomba centrífuga como en la del elevador de cangilones su crecimiento lineal frente al crecimiento de la frecuencia en la que opera el motor, tal y como se esperaría.

También se tomaron los voltajes presentes en una de las líneas trifásicas y estos se mantuvieron cercanos a su valor nominal 208/230 V.

El cableado de la red se colocó en una canaleta aérea diseñada exclusivamente para el adecuado cableado de la red y se utilizó el cable Siemens Simatic Net Profibus 22AWG (blindado) que tiene excelentes propiedades para aislarse del medio, con doble blindaje. Necesario para mantener la comunicación íntegra y que los motores no interfieran introduciendo ruido a la red.

## B. Comunicación con los variadores desde estación piloto:

La interfaz de usuario del programa maestro, es una interfaz diseñada para facilitar el monitoreo de los equipos y la mejora de la obtención de datos en las pruebas de cada equipo.

Dicho desarrollo presenta la facilidad de que detecta automáticamente los puertos serie utilizados cuando se abre la aplicación para que el usuario seleccione el que utiliza para la comunicación.

En la interfaz final se busca que el operador no tenga necesidad de ingresar los parámetros que pueden presentarle alguna complicación.

Debido a que se decidió poder mostrar todos los parámetros que se tienen a disposición en los registros de lectura, se optó por dejar que el usuario decida el rango de los registros de lectura que le interesan y polearlos a la tasa de muestreo.

La red funciona adecuadamente encendiendo, modificando la frecuencia y apagando el equipo por medio de la función de escritura. Dos de los cuatro equipos están actualmente configurados para esta característica, son el caso del elevador de cangilones y el banco hidráulico. Para el elevador del tornillo sin fin y el reactor se necesita modificar la parametrización actual de los variadores para que se pueda realizar, pero en estos momentos es requerimiento de parte de los operadores de los equipos dejarlos sólo con monitoreo por parte de la red y encenderlos, apagarlos y modificar su frecuencia localmente.

## VIII. CONCLUSIONES

1. Se logró instalar los variadores en los equipos bajo los parámetros adecuados.
2. El circuito que se ha implementado convirtió los protocolos RS232/RS485 correctamente a una tasa de transferencia máxima de 115.2Kbps
3. Se tiene la red de 4 variadores expandible hasta para 32 dispositivos en cascada para una red cliente servidor.
4. Se implementó el maestro de la red de variadores con el protocolo Modbus y se cuenta con el desarrollo de la lectura y escritura de los registros
5. Se interpretó el parámetro de la corriente de torque para la obtención del par que entrega el motor.
6. La interfaz y el circuito permiten conexión directa e individual a variadores de frecuencia tipo Power Flex.

## IX. RECOMENDACIONES

1. De ser necesario controlar los variadores a una frecuencia mayor a los 60Hz se necesita invertir en motores vectoriales que no se dañen al ser sometidos a frecuencias más elevadas.
2. Es necesario verificar el estado de la red eléctrica y su posible presencia de armónicos, por la creciente utilización de variadores, dichos armónicos podrían producción desperfectos en los sistemas.
3. Para incrementar la portabilidad del circuito se recomienda su eventual conversión futura a conectividad USB.

## X. BIBLIOGRAFÍA

1. ABB drives. "Using variable speed drives (VSDs) in pump applications", Internet: [http://library.abb.com/global/scot/scot201.nsf/veritydisplay/f78dcbfe1b99a353c125715e0027f769/\\$File/ApplicationGuide1\\_Pumps\\_RevA\\_lowres.pdf](http://library.abb.com/global/scot/scot201.nsf/veritydisplay/f78dcbfe1b99a353c125715e0027f769/$File/ApplicationGuide1_Pumps_RevA_lowres.pdf) 26 de abril de 2006. [Accesado en noviembre de 2008]
2. ABB Inc. "Drives and power electronics, variable speed drives & controls", 2001. Internet: [http://www.ioliettech.com/what\\_is\\_a\\_variable\\_frequency\\_drive.htm](http://www.ioliettech.com/what_is_a_variable_frequency_drive.htm) [Accesado el 15 de noviembre de 2008]
3. "Adjustable Frecuency AC Driver PowerFlex40, User Manual, Rockwell Automation", Internet: [http://literature.rockwellautomation.com/idc/groups/literature/documents/um/22b-um001\\_en-e.pdf](http://literature.rockwellautomation.com/idc/groups/literature/documents/um/22b-um001_en-e.pdf) Enero de 2007. [Accesado en mayo de 2009]
4. Baldor Electric Company, "Specifications: M3542". Internet: [http://www.baldor.com/products/specs.asp?1=1&competitor=Lincoln&catalog=M3542&product=AC+Motors&family=General+Purpose|vw\\_ACMotors\\_GeneralPurpose&winding=34WG0883&rating=40CMB-CONT](http://www.baldor.com/products/specs.asp?1=1&competitor=Lincoln&catalog=M3542&product=AC+Motors&family=General+Purpose|vw_ACMotors_GeneralPurpose&winding=34WG0883&rating=40CMB-CONT) 2008. [Accesado en noviembre de 2008]

5. Bartos, Frank J. "Control engineering, AC drives stay vital for the 21st century", Internet:[http://www.controleng.com/index.asp?layout=article&articleid=CA450388&article\\_prefix=CA&article\\_id=450388](http://www.controleng.com/index.asp?layout=article&articleid=CA450388&article_prefix=CA&article_id=450388) 9 de enero del 2004. [Accesado en noviembre de 2008]
6. Bies, Lammert. "Modbus interface tutorial". Internet: <http://www.lammertbies.nl/comm/info/modbus.html> Abril, 2008. [Noviembre de 2008]
7. Chapman, Stephen. 2004. *Máquinas Eléctricas*. 4ª. ed. McGraw-Hill Professional, 746 págs.
8. "Modbus over serial line specification and implementation guide V1.02". Internet:[http://cars9.uchicago.edu/software/epics/Modbus\\_over\\_serial\\_line\\_V1\\_02.pdf](http://cars9.uchicago.edu/software/epics/Modbus_over_serial_line_V1_02.pdf) 20 de diciembre de 2006. [Accesado en mayo de 2009]
9. Modbus-IDA, "Modbus Application Protocol Specification V1.1b", Internet:[http://www.modbus.org/docs/Modbus\\_Application\\_Protocol\\_V1\\_1b.pdf](http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf) 28 de diciembre de 2006. [Accesado en mayo de 2009]
10. Mohd Fazalul, Chemical Reactor. Disponible en Internet: <http://hezrif.uitm.edu.my/ese652/53%20Chemical%20Reactor%20DOC%20090915.pdf>
11. Pérez Ludueña, Jesús. Mayo 2007. *Ahorrando energía en aplicaciones de bombeo*. Disponible en: <http://www.emb.cl/electroindustria/articulo.mv?xid=693&rank=1> [Accesado en noviembre de 2009]

12. Polka, Dave. 2003. *Motors and drives: A practical technology guide*. ISA. 385 págs.
13. "PowerFlex® 4 and PowerFlex® 40 AC Drives, technical data". Rockwell Automation", Disponible en Internet:  
  
<http://www.aschome.com/PDFs/Powerflex%2040%20Specifications.pdf>  
[Accesado en abril de 2009]
14. Schmidt, Lanny D. 1998. *The Engineering of Chemical Reactions*. New York: Oxford University Press.
15. Siemens Industry USA. "Modbus information". Disponible en Internet:  
<http://www.sea.siemens.com/us/SiteCollectionDocuments/WSSResources/Internet/Products/ModbusInformation.pdf>. [Accesado en octubre de 2010]
16. Smith, R.E. "Convertidor rs485/rs232", Internet:  
<http://rs485.com/pdf/files/converters/asc24.pdf> 28 de julio de 1999. [Accesado noviembre de 2009]
17. Turkel, Solomon S., "Understanding variable speed drives", Apr 1, 1997 12:00 PM.  
Internet:[http://ecmweb.com/mag/electric\\_understanding\\_variable\\_speed\\_3/index.html](http://ecmweb.com/mag/electric_understanding_variable_speed_3/index.html) 1 de abril de 1997, 12:00 PM. [Accesado noviembre de 2009]

## XI. GLOSARIO

Bomba centrífuga: Es la encargada de impulsar un líquido mediante una fuerza centrífuga.

Cangilón: Es un recipiente encargado del transporte de material como granos, agua, piedras o metales.

Impeller: Es la parte de un elemento rotativo en un compresor que imparte energía al flujo del medio mediante fuerza centrífuga. Se compone de una serie de palas montadas que giran con el eje.

Modbus: Es un protocolo de comunicación basado en una arquitectura Maestro/Esclavo. El medio de transmisión son Ethernet y RS485. Ampliamente expandido en la industria y la automatización.

Variador de frecuencia: (siglas VFD, del inglés: Variable Frequency Drive o bien AFD Adjustable Frequency Drive) es un sistema para el control de la velocidad rotacional de un motor de corriente alterna (AC) por medio del control de la frecuencia de alimentación suministrada al motor

## XII. ANEXOS

```

//*****
****
//Autor: Hector Bonilla
//Fecha: 1-10-2010
//Clase: Modbus Class
//*****
****
using System;
using System.Collections.Generic;
using System.Text;
using System.IO.Ports;

namespace Variadores
{
    class Modbus
    {
        private SerialPort serial = new
SerialPort();
        public string modbusStatus;

        #region Constructor / Destructor
        public Modbus()
        {
        }
        ~Modbus()
        {
        }
        #endregion

        #region Metodos Open / Close
        public bool Open(string portName,
int baudRate, int databits, Parity
parity, StopBits stopBits)
        {
            //Aseguro que no esta abierto
            if (!serial.IsOpen)
            {
                //Se ingresa los
parametros seriales deseados
                serial.PortName =
portName;
                serial.Parity = parity;
                serial.BaudRate =
baudRate;
                serial.DataBits =
databits;
                serial.StopBits =
stopBits;

                //Timeouts
                serial.ReadTimeout =
1000;
                serial.WriteTimeout =
1000;

                try
                {
                    serial.Open();
                }
                catch (Exception err)
                    {
                        modbusStatus = "Error
iniciando puerto " + portName + ": " +
err.Message;
                        return false;
                    }
                modbusStatus = portName +
"Puerto abierto exitosamente ";
                return true;
            }
            else
            {
                modbusStatus = portName +
"Puerto previamente abierto";
                return false;
            }
        }
        public bool Close()
        {
            //Asegura que este abierto
para cerrarlo:
            if (serial.IsOpen)
            {
                try
                {
                    serial.Close();
                }
                catch (Exception err)
                {
                    modbusStatus = "Error
cerrando puerto " + serial.PortName + ":
" + err.Message;
                    return false;
                }
                modbusStatus =
serial.PortName + " Cerrado
exitosamente";
                return true;
            }
            else
            {
                modbusStatus =
serial.PortName + " No esta abierto";
                return false;
            }
        }
    }
}
#endregion

```

```

    #region Computo de CRC
    private void GetCRC(byte[]
message, ref byte[] CRC)
    {
        //Espera un mensaje modbus de
cualquier tamaño y 2 bytes de CRC

        ushort CRCFull = 0xFFFF;
        byte CRCHigh = 0xFF, CRCLow =
0xFF;
        char CRCLSB;

        for (int i = 0; i <
(message.Length) - 2; i++)
        {
            CRCFull =
(ushort) (CRCFull ^ message[i]);

            for (int j = 0; j < 8;
j++)
            {
                CRCLSB =
(char) (CRCFull & 0x0001);
                CRCFull =
(ushort) ((CRCFull >> 1) & 0x7FFF);

                if (CRCLSB == 1)
                    CRCFull =
(ushort) (CRCFull ^ 0xA001);
            }
            CRC[1] = CRCHigh =
(byte) ((CRCFull >> 8) & 0xFF);
            CRC[0] = CRCLow =
(byte) (CRCFull & 0xFF);
        }
        #endregion

        #region Metodo BuildMessage
        private void BuildMessage(byte
address, byte type, ushort start, ushort
registers, ref byte[] message)
        {
            //Arreglo que recibe el CRC
            byte[] CRC = new byte[2];

            message[0] = address;
            message[1] = type;
            message[2] = (byte) (start >>
8);
            message[3] = (byte) start;
            message[4] = (byte) (registers
>> 8);
            message[5] = (byte) registers;

            GetCRC(message, ref CRC);
            message[message.Length - 2] =
CRC[0];
            message[message.Length - 1] =
CRC[1];
        }
        #endregion

        #region Metodo CheckResponse
        private bool CheckResponse(byte[]
response)
        {
            //Hace un chequeo basico de
CRC

            byte[] CRC = new byte[2];
            GetCRC(response, ref CRC);

            if (CRC[0] ==
response[response.Length - 2] && CRC[1]
== response[response.Length - 1])
                return true;
            else
                return false;
        }
        #endregion

        #region Metodo GetResponse
        private void GetResponse(ref
byte[] response)
        {
            //Lee del buffer

            for (int i = 0; i <
response.Length; i++)
            {
                response[i] =
(byte) (serial.ReadByte());
            }
        }
        #endregion

        #region Funcion 16 - Write
        Register
        public bool SendFc16(byte
address, ushort start, ushort registers,
short[] values)
        {
            //Puerto esta abierto:
            if (serial.IsOpen)
            {
                //Limpia buffers:

                serial.DiscardOutBuffer();
                serial.DiscardInBuffer();
                //Mensaje 1 addr + 1 fcn
+ 2 start + 2 reg + 1 count + 2 * reg
vals + 2 CRC

                byte[] message = new
byte[9 + 2 * registers];
                //Respuesta es fixcada en
8 bytes

                byte[] response = new
byte[8];

                //Se agrega bytecount al
mensaje

                message[6] =
(byte) (registers * 2);
                //Escribir valores en el
mensaje antes de mandar
                for (int i = 0; i <
registers; i++)
                {
                    message[7 + 2 * i] =
(byte) (values[i] >> 8);
                    message[8 + 2 * i] =
(byte) (values[i]);
                }
                //Construye el mensaje
que sale

                BuildMessage(address,
(byte) 16, start, registers, ref message);

                //Manda el mensaje Modbus
por el puerto

                try

```

```

        {
            serial.Write(message,
0, message.Length);
            GetResponse(ref
response);
        }
        catch (Exception err)
        {
            modbusStatus = "Error
en la escritura: " + err.Message;
            return false;
        }
        //Evaluate message:
        if
(CheckResponse(response))
        {
            modbusStatus =
"Escrito exitosamente";
            return true;
        }
        else
        {
            modbusStatus = "CRC
error";
            return false;
        }
    }
    else
    {
        modbusStatus = "Puerto
cerrado";
        return false;
    }
}
#endregion

#region Function 3 - Read
Registers
public bool SendFc3(byte address,
ushort start, ushort registers, ref
short[] values)
{
    //Puerto esta abierto
    if (serial.IsOpen)
    {
        //Limpia Buffers

        serial.DiscardOutBuffer();
        serial.DiscardInBuffer();
        //Mensaje es de 8 bytes
        byte[] message = new
byte[8];
        //Respuesta de lectura en
el buffer
        byte[] response = new
byte[5 + 2 * registers];
        //Construye el mensaje
que envia
        BuildMessage(address,
(byte)3, start, registers, ref message);
        //Envia al puerto
        try
        {
            serial.Write(message,
0, message.Length);
            GetResponse(ref
response);
        }
        catch (Exception err)
        {
            modbusStatus = "Error
en lectura: " + err.Message;
            return false;
        }
        //Evalua el mensaje
        if
(CheckResponse(response))
        {
            //Devuelve los
valores
            for (int i = 0; i <
(response.Length - 5) / 2; i++)
            {
                values[i] =
response[2 * i + 3];
                values[i] <= 8;
                values[i] +=
response[2 * i + 4];
            }
            modbusStatus =
"Lectura Exitosa";
            return true;
        }
        else
        {
            modbusStatus = "CRC
error";
            return false;
        }
    }
    else
    {
        modbusStatus = "Puerto
cerrado";
        return false;
    }
}
}
#endregion

}
}

//*****
//Autor: Hector Bonilla
//Fecha: 1/10/2010
//Clase: Manejo de eventos y GUI de
Modbus
//*****
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Timers;
using System.IO.Ports;

namespace Variadores
{
    public partial class Form1 : Form
    {
        Modbus Mbus = new Modbus();
        SerialPort serial = new
SerialPort();
    }
}

```

```

        System.Timers.Timer timer = new
System.Timers.Timer();
        bool poleando = false;
        string tipoDato;
        int counter;

        #region GUI
        public delegate void
GUIDelegate(string parametros);
        public delegate void GUIClear();
        public delegate void
GUIStatus(string parametros);
        #endregion

        public Form1()
        {
            InitializeComponent();
            CargaComboBox();
            timer.Elapsed += new
ElapsedEventHandler(timer_Elapsed);
        }

        #region Funciones Clear Status
Update del GUI
        public void DoGUIClear()
        {
            if (this.InvokeRequired)
            {
                GUIClear delegateMethod =
new GUIClear(this.DoGUIClear);
                this.Invoke(delegateMethod);
            }
            else
            {
                this.ValoresRegistros.Items.Clear();
            }
            public void DoGUIStatus(string
parametros)
            {
                if (this.InvokeRequired)
                {
                    GUIStatus delegateMethod
= new GUIStatus(this.DoGUIStatus);
                    this.Invoke(delegateMethod, new object[]
{ parametros });
                }
                else
                {
                    this.LableStatus.Text =
parametros;
                }
            }
            public void DoGUIUpdate(string
parametros)
            {
                if (this.InvokeRequired)
                {
                    GUIDelegate
delegateMethod = new
GUIDelegate(this.DoGUIUpdate);
                    this.Invoke(delegateMethod, new object[]
{ parametros });
                }
                else
                {
                    this.ValoresRegistros.Items.Add(parametro
s);
                }
            }
        }
        #endregion

        #region Timer Elapsed Event
Handler
        void timer_Elapsed(object sender,
ElapsedEventArgs e)
        {
            PollFunction();
        }
        #endregion

        #region ComboBoxes
private void CargaComboBox()
        {
            //Carga - puertos, baudrates,
datatype.
            // Puertos Disponibles:
            string[] puertos =
SerialPort.GetPortNames();

            foreach (string puerto in
puertos)
            {
                cPorts.Items.Add(puerto);
            }
            cPorts.SelectedIndex = 0;

            //Baudrates:
            string[] baudrates = {
"115200", "57600", "38400", "19200",
"9600" };

            foreach (string baudrate in
baudrates)
            {
                cBaudRate.Items.Add(baudrate);
            }
            cBaudRate.SelectedIndex = 4;

            //Tipo de dato:
            string[] dataTypes = {
"Decimal", "Hexadecimal", "Float", };

            foreach (string dataType in
dataTypes)
            {
                cDataType.Items.Add(dataType);
            }
            cDataType.SelectedIndex = 0;

            //Textbox defaults:
            tNumReg.Text = "20";
            tSampleo.Text = "1000";
            tSlave.Text = "100";
            tStartAdd.Text = "0";
        }
        #endregion

        #region Funciones Start y Stop
private void StartPoll()
        {
            counter = 0;

            //Abrir COM port con los
setings provistos
            if
(Mbus.Open(cPorts.SelectedItem.ToString()
,
Convert.ToInt32(cBaudRate.SelectedItem.To
String()),

```

```

        8, Parity.None,
StopBits.One))
    {
        //Desabilitar doble
Start
        bStart.Enabled = false;
        tipoDato =
cDataType.SelectedItem.ToString();

        //Bandera de poleando
        poleando = true;

        //Inicializa timer
        timer.AutoReset = true;
        if (tSampleo.Text != "")
            timer.Interval =
Convert.ToDouble(tSampleo.Text);
        else
            timer.Interval =
1000;

        timer.Start();
    }
    LableStatus.Text =
Mbus.modbusStatus;
}
private void StopPoll()
{
    //Para timer y cierra el COM
port
    poleando = false;
    timer.Stop();
    Mbus.Close();

    bStart.Enabled = true;

    LableStatus.Text =
Mbus.modbusStatus;
}
private void
bStart_Click_1(object sender, EventArgs
e)
{
    StartPoll();
}
private void bStop_Click_1(object
sender, EventArgs e)
{
    StopPoll();
}
#endregion

#region Funcion de poleado
private void PollFunction()
{
    //Update GUI:
    DoGUIClear();
    counter++;
    DoGUIStatus("Poleado: " +
counter.ToString());

    //Crea arreglo con los
valores leidos:
    short[] values = new
short[Convert.ToInt32(tNumReg.Text)];
    ushort pollStart;
    ushort pollLength;

    if (tStartAdd.Text != "")
        pollStart =
Convert.ToUInt16(tStartAdd.Text);
    else
        pollStart = 0;
    if (tNumReg.Text != "")
        pollLength =
Convert.ToUInt16(tNumReg.Text);
    else
        pollLength = 20;

    //Lee los registros y
despliega los datos:
    try
    {
        while
(!Mbus.SendFc3(Convert.ToByte(tSlave.Text
), pollStart, pollLength, ref values));
    }
    catch (Exception err)
    {
        DoGUIStatus("Error en la
lectura: " + err.Message);
    }

    string itemString;
    // string[] RegName = new
string[20] {"", "Output Frec", "Commanded
Frec", "Output Current", "Output
Voltage",
// "DC Bus Voltaje", "Drive
Status", "Fault 1 Code", "Fault 2 Code",
"Fault 3 Code", "Process Display",
// "Control Source", "Control
In Status", "Dig In Status", "Comm
Status", "Control SW Ver", "Drive Type",
// "Elapsed Run Time",
"Testpoint Data", "Anlog In 0-10v"};

    switch (tipoDato)
    {
        case "Decimal":
            for (int i = 0; i <
pollLength; i++)
            {
                // itemString =
 "[" + Convert.ToString(pollStart + i +
40001) + "], MB[" +
                // itemString =
 "[" + RegName[i] + "], MB[" +
                itemString =
"MB[" +
Convert.ToString(pollStart + i) + "] = "
+ values[i].ToString());
                DoGUIUpdate(itemString);
            }
            break;
        case "Hexadecimal":
            for (int i = 0; i <
pollLength; i++)
            {
                itemString = "["
+ Convert.ToString(pollStart + i + 40001)
+ "], MB[" +
Convert.ToString(pollStart + i) + "] = "
+ values[i].ToString("X");
                DoGUIUpdate(itemString);
            }
            break;
        case "Float":

```

```

        for (int i = 0; i <
(pollLength / 2); i++)
        {
            int intValue =
(int)values[2 * i];
            intValue <= 16;
            intValue +=
(int)values[2 * i + 1];
            itemString = "["
+ Convert.ToString(pollStart + 2 * i +
40001) + "]" , MB[" +
Convert.ToString(pollStart + 2 * i) + "]"
= " +
(BitConverter.ToSingle(BitConverter.GetBy
tes(intValue), 0)).ToString();
DoGUIUpdate(itemString);
        }
        break;
    }
}
#endregion

#region Escribir Frecuencia
/*private void WriteFunction()
{
    //StopPoll();

    if (tEscribeReg.Text != "" &&
tEscribeValor.Text != "" && tSlave.Text
!= "")
    {
        byte address =
Convert.ToByte(tSlave.Text);
        ushort start =
Convert.ToUInt16(tEscribeReg.Text);
        short[] value = new
short[1];
        value[0] =
Convert.ToInt16(tEscribeValor.Text);

        try
        {
            while
(!Mbus.SendFcl6(address, start,
(ushort)1, value)) ;
        }
        catch (Exception err)
        {
            DoGUIStatus("Error en
la escritura: " + err.Message);
        }

        DoGUIStatus (Mbus.modbusStatus);
    }
    else
        DoGUIStatus("Llene todos
los campos antes de escribir");

    //StartPoll();
}
private void
bEscribir_Click(object sender, EventArgs
e)
{
    WriteFunction();
}
}
#endregion

*/

private void WriteFrequency()
{
    if (tEscribeValor.Text != ""
&& tSlave.Text != "")
    {
        byte address =
Convert.ToByte(tSlave.Text);
        ushort start = 8193;
        short[] value = new
short[1];
        value[0] =
(short) (10*Convert.ToInt16(tEscribeValor.
Text));

        try
        {
            while
(!Mbus.SendFcl6(address, start,
(ushort)1, value)) ;
        }
        catch (Exception err)
        {
            DoGUIStatus("Error en
la escritura:" + err.Message);
        }

        DoGUIStatus (Mbus.modbusStatus);
    }
    else
        DoGUIStatus("Llene los
campos necesarios para envio de
frecuencia");

    //StartPoll();
}
private void
bEscribir_Click(object sender, EventArgs
e)
{
    WriteFrequency();
}
}
#endregion

#region Encendido del variador
private void Encender()
{
    if (tSlave.Text != "")
    {
        byte address =
Convert.ToByte(tSlave.Text);
        ushort start = 8192;
        short[] value = new
short[1];
        value[0] = 12314;

        try
        {
            while
(!Mbus.SendFcl6(address, start,
(ushort)1, value)) ;
        }
        catch (Exception err)
        {
            DoGUIStatus("Error en
la escritura de Encendido:" +
err.Message);
        }
    }
}
}

```

```

DoGUIStatus (Mbus.modbusStatus);
    }
    else
        DoGUIStatus("Llene los
campos necesarios para Encender");

        //StartPoll();
    }

private void bInicio_Click(object
sender, EventArgs e)
{
    Encender();
}

#endregion

#region Apagado del Variador
private void Apagar()
{
    if (tSlave.Text != "")
    {
        byte address =
Convert.ToByte(tSlave.Text);
        ushort start = 8192;
        short[] value = new
short[1];
        value[0] = 12313;

        try
        {
            while
(!Mbus.SendFc16(address, start,
(ushort)1, value)) ;

        }
        catch (Exception err)
        {
            DoGUIStatus("Error en
la escritura de Apagado:" + err.Message);
        }
    }
}

DoGUIStatus (Mbus.modbusStatus);
    }
    else
        DoGUIStatus("Llene los
campos necesarios para apagar");

        //StartPoll();
    }

private void bParo_Click(object
sender, EventArgs e)
{
    Apagar();
}

#endregion

#region Cambio DataType ?
private void
cDataType_SelectedIndexChanged(object
sender, EventArgs e)
{
    //Inicia el poleado si el
tipo de dato cambio en el proceso
    if (poleando)
    {
        StopPoll();
        tipoDato =
cDataType.SelectedItem.ToString();
        StartPoll();
    }
}
#endregion
}
}

```

RED DE VARIADORES DE FRECUENCIA

---

Laboratorio de Operaciones Unitarias UVG

# Manual de Usuario

UNIVERSIDAD DEL VALLE DE GUATEMALA

# **Manual de usuario de la red de variadores de frecuencia**

---

Héctor Bonilla D.  
e-mail: [hbonill@gmail.com](mailto:hbonill@gmail.com)

## Índice

<b>Requisitos del Sistema</b> .....	3
<b>Instalación</b> .....	3-7
<b>Uso de la Aplicación</b> .....	8
Área de configuración de comunicación .....	9
Parámetros programables de comunicación en el variador .....	10
Parámetros del grupo de visualización .....	11
Área de mando hacia el variador .....	12

*Lea este manual en su totalidad, antes de trabajar con el controlador de la red de variadores de frecuencia del laboratorio de Operaciones Unitarias de la UVG.*

I C O N K E Y	
	Advertencia
	Precaución
	Nota

**Tome en cuenta estos iconos que serán utilizados en este manual.**

## Requisitos del Sistema

1. Sistema operativo Windows
2. Una computadora con interfaz serie COM para conectarse a la red.

 Es factible utilizar un convertidor USB-SERIAL si en caso no se posee este puerto de forma nativa en la PC a utilizar.

## Instalación

### Paso 1

No es necesario ningún tipo de instalador. El archivo Variadores.exe es el ejecutable del programa directamente, traslade este a su escritorio y ejecútelo directamente haciendo doble CLICK sobre el icono.



Este es el Icono del archivo Variadores.exe

## Paso 2

Conecte a la alimentación el circuito convertidor COM->RS485.

## Paso 3

Conecte su PC al circuito (Figura 1) por medio de un cable DB9 serial (Figura 2) o un convertidor serial/USB (Figura 3) en el plug COM->RS485.



Figure 1. Circuito Convertidor



Figure 2. Cable Serial DB9 Macho/Hembra



Figure 3. Convertidor USB/Serial

**Paso 4**

Conecte el cable Siemens Profibus (Figura 4) en el plug verde de un costado del circuito (Figura 5) asegurando este con un destornillador de precisión, tal como se observa en la Figura 6.

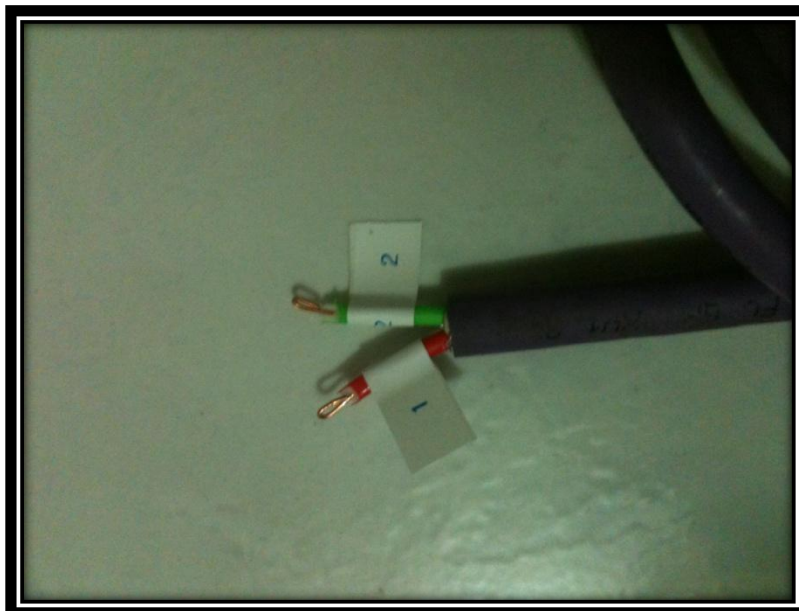


Figure 4. Cable Siemens Profibus



Figure 5. Plug del cable Profibus de la Red

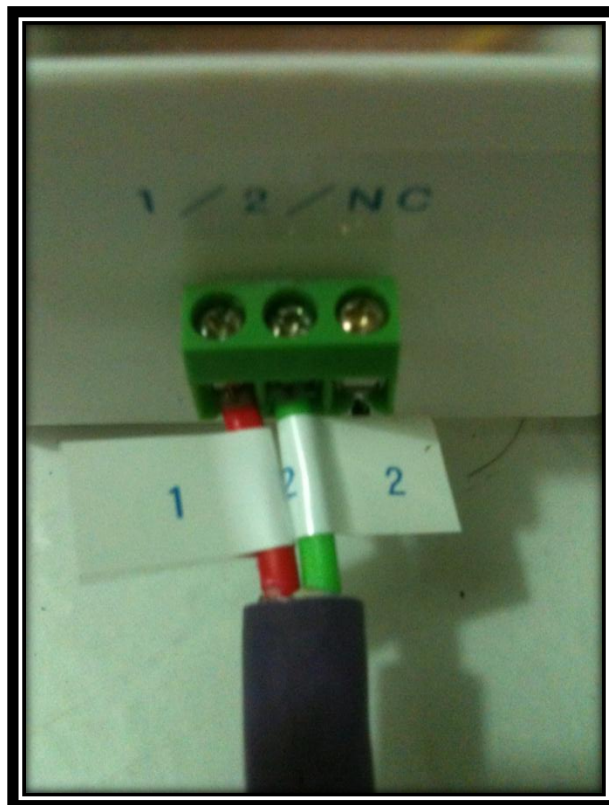


Figure 6. Conexion Cable Profibus

**Paso 5**

Conectar el terminal RJ45 (Figura 7) al variador.

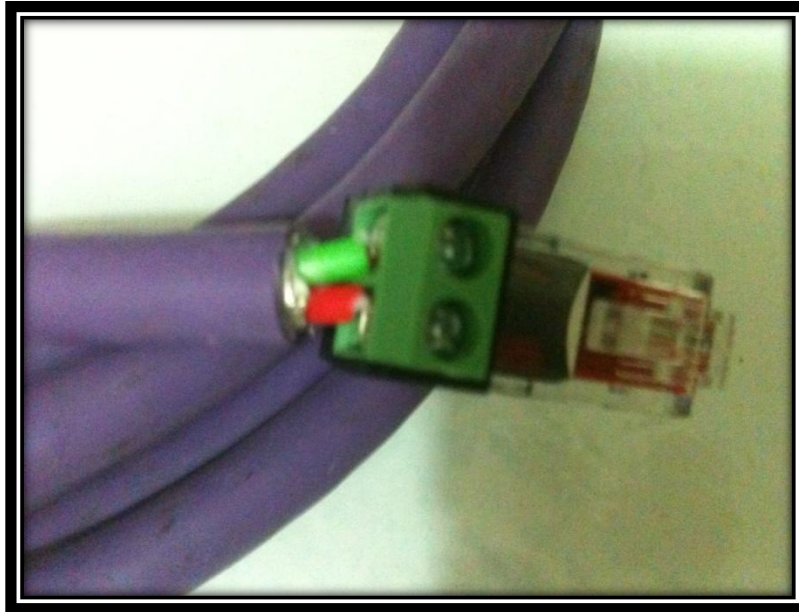
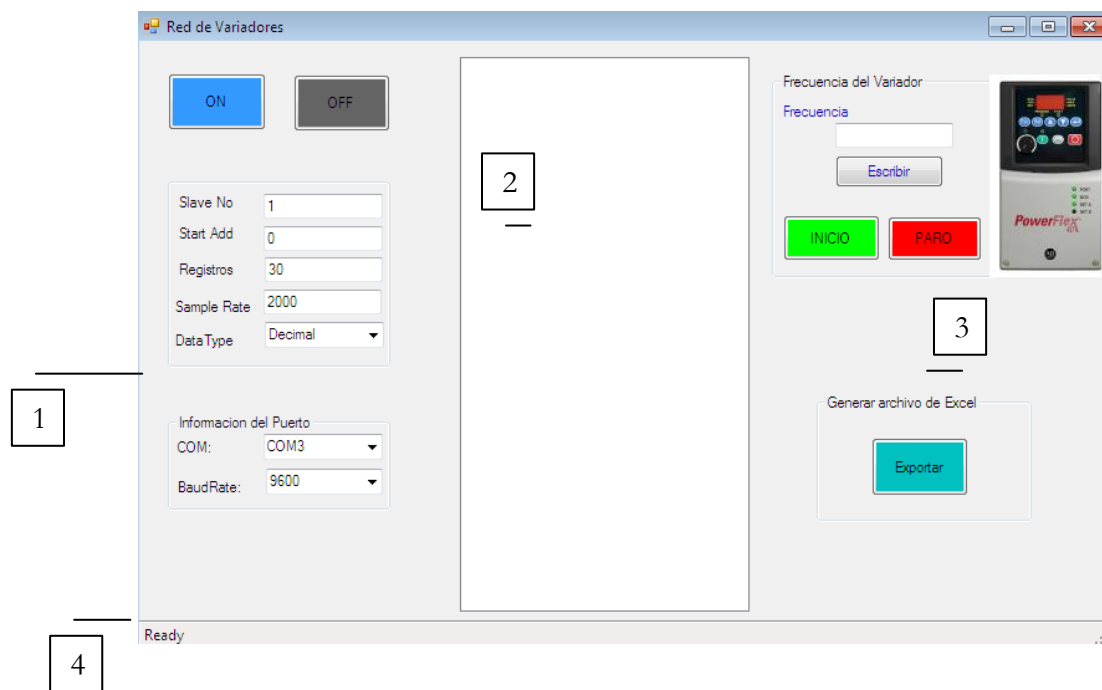


Figure 7 Terminal RJ45 a conectar con el variador

# Uso de la Aplicación

## Cuenta con cuatro partes principales:

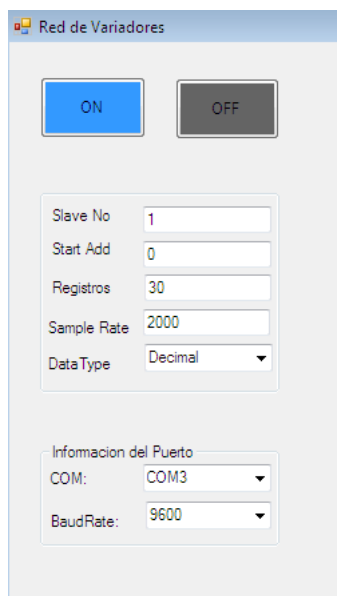
1. Área de configuración de comunicación
2. Área de monitoreo de datos
3. Área de mando hacia el variador
4. Área de notificación





---

## ÁREA DE CONFIGURACIÓN DE COMUNICACIÓN

---



- 1 En esta área se observan los botones **ON** y **OFF**. Ellos dan inicio y fin a las lecturas que se tendrán de los registros del variador.
- 2 **Slave No** consiste en un número del *1 al 247* que es el identificador del variador. Programable en el parámetro **A104**.
- 3 **Start Add** es el primer valor de los registros que desplegará en el área de *monitoreo de datos*.  
 No modifique este valor si pretende obtener los 30 parámetros de visualización completos. Y coincidan con lo generado en el archivo de Excel.
- 4 **Registros** es la cantidad de registros que se desplegaran en el área de *monitoreo de datos*.  
 No modifique este valor si pretende obtener los 30 parámetros de visualización completos. Y coincidan con lo generado en el archivo de Excel.
- 5 **Sample Rate** Es la tasa de muestreo que tendrá la captura de la información en *mili-segundos*. El valor por defecto es de 2 segundos.
- 6 **COM:** consiste en el número de puerto serie al cual que se está interconectando el equipo.
- 7 **BaudRate:** Esta es la velocidad de la transferencia del Puerto serie. Por defecto se tiene una velocidad de **9.6Kbps** la máxima velocidad es de **38.4Kbps**. Esta debe coincidir con el parámetro **A103** programado en el variador.



Si aparecen múltiples opciones de números de puerto en el COM y usted desconoce cuál de estos corresponde puede verificar en el Gestor de Dispositivos haciendo **CLICK DERECHO** en:  
**Computer -> Manage -> Device Manager -> Ports (COM & LPT)**

**PARAMETROS PROGRAMABLES DE COMUNICACIÓN EN EL VARIADOR**

A continuación los parámetros que se deben de configurar en el variador para que la comunicación sea exitosa

Parámetro	Descripción	Rango de Conf.	Valores	Default						
A103	[Comm Data Rate]	0/5	<table border="1"> <tr> <td>0="1200"</td> <td>3="9600"</td> </tr> <tr> <td>1="2400"</td> <td>4="19.2K"</td> </tr> <tr> <td>2="4800"</td> <td>5="38.4K"</td> </tr> </table>	0="1200"	3="9600"	1="2400"	4="19.2K"	2="4800"	5="38.4K"	3
0="1200"	3="9600"									
1="2400"	4="19.2K"									
2="4800"	5="38.4K"									
A104	[Comm Node Addr]	1/247	1	100						
A105	[Comm Loss Action]	0/3	<table border="1"> <tr> <td>0="Fault"</td> <td>2="Stop"</td> </tr> <tr> <td>1="Coast Stop"</td> <td>3="Continu Last"</td> </tr> </table>	0="Fault"	2="Stop"	1="Coast Stop"	3="Continu Last"	0		
0="Fault"	2="Stop"									
1="Coast Stop"	3="Continu Last"									
A106	[Comm Loss Time]	0.1/60.0 segundos	0.1 segundos	5.0 Seg						



Si se llegaran a perder datos en la velocidad de transmisión, ante todo si se está monitoreando los equipos más distantes a la estación de trabajo.



Equipo	Comm Node Addr
Reactor	1
Elevador de Cangilones	2
Tornillo Sin Fin	3
Banco Hidraulico	4


**PARAMETROS DEL GRUPO DE  
VISUALIZACION**


No.	Parámetro	Mín./máx.	Pantalla /opciones
d001	[Output Freq]	0.0/[Maximum Freq]	0.1 Hz
d002	[Commanded Freq]	0.0/[Maximum Freq]	0.1 Hz
d003	[Output Current]	0.00/(amperios variador × 2)	0.01 Amps
d004	[Output Voltage]	0/voltaje nominal del variador	1 VCA
d005	[DC Bus Voltage]	Basado en capacidad nominal del variador	1 VCC
d006	[Drive Status]	0/1 (1 = Condición verdadera)	<u>Bit 3</u> Desacelerando <u>Bit 2</u> Acelerando <u>Bit 1</u> Avance <u>Bit 0</u> En marcha
d007- d009	[Fault x Code]	F2/F122	F1
d010	[Process Display]	0.00/9999	0.01 – 1
d012	[Control Source]	0/9	<u>Dígitos 1 = Comando de veloc.</u> <u>Dígito 0 = Comando de arranque</u> (Vea P038; 9 = "Frecuencia Test")    (Vea P036; 9 = "Avan/impuls")
d013	[Contrl In Status]	0/1 (1 = Entrada presente)	<u>Bit 3</u> Trans DB Enc <u>Bit 2</u> Ent. Paro <u>Bit 1</u> Ent/Dir/Rev <u>Bit 0</u> Ent/Arr/Ava
d014	[Dig In Status]	0/1 (1 = Entrada presente)	<u>Bit 3</u> En. digit. 4 <u>Bit 2</u> En. digit. 3 <u>Bit 1</u> En. digit. 2 <u>Bit 0</u> En. digit. 1
d015	[Comm Status]	0/1 (1 = Condición verdadera)	<u>Bit 3</u> Error <u>Bit 2</u> DSI <u>Bit 1</u> Tx <u>Bit 0</u> Rx
d016	[Control SW Ver]	1.00/99.99	0.01
d017	[Drive Type]	1001/9999	1
d018	[Elapsed Run Time]	0/9999 Hrs	1 = 10 Hrs
d019	[Testpoint Data]	0/FFFF	1 Hex
d020	[Analog In 0–10 V]	0.0/100.0 %	0.1 %
d021	[Analog In 4–20 mA]	0.0/100.0 %	0.1 %
d022	[Output Power]	0.00/(Potencia variador × 2)	0.01 kW
d023	[Output Powr Fctr]	0.0/180.0 grados	0.1 grados
d024	[Drive Temp]	0/120 °C	1 °C
d025	[Counter Status]	0/9999	1
d026	[Timer Status]	0.0/9999 segundos	0.1 segundos
d028	[Stp Logic Status]	0/7	1
d029	[Torque Current]	0.00/(amperios del variador × 2)	0.01 Amps

**AREA DE MANDO HACIA EL  
VARIADOR**



- 1 En esta área se observan los botones **INICIO** y **PARO**. Estos permiten dar marcha al variador energizado y detener la operación del mismo.
- 2 **Frecuencia** En este campo se coloca la frecuencia en  $H_z$  que se desea opere el variador, este realizará el mando cuando se haga CLICK el botón **Escribir**.
- 3 **Generar archivo de Excel** en este campo se observa el botón **Exportar** que nos sirve para guardar los primeros 100 datos muestreados de los 30 parámetros del grupo de visualización.

 El exportar generará un archivo con extensión .csv que lo abre Excel.

 Para que estos mandos se ejecuten en el variador, los parametros **P036** y **P038** deben de setarse a **5**

P038	[Speed Reference]	0/7	0 = "Drive Pot" 1 = "InternalFreq" 2 = "0-10 V Input" 3 = "4-20 mA Input" 4 = "Preset Freq" 5 = "Comm Port" 6 = "Stp Logic" 7 = "Anlg In Mult"	0 1
P036	[Start Source]	0/6	0 = "Keypad" <sup>(1)</sup> 1 = "3-Wire" 2 = "2-Wire" 3 = "2-W Lvl Sens" 4 = "2-W Hi Speed" 5 = "Comm Port" 6 = "Momt FWD/REV"	0

<sup>(1)</sup> Cuando está activo, la tecla de Retroceso también está activa, a menos que haya sido inhabilitada por A095 [Reverse Disable].