

**ESCUELA SUPERIOR POLITECNICA DEL LITORAL**  
**Facultad de Ingeniería en Electricidad y Computación**



“ IMPLEMENTACIÓN DEL SISTEMA DE CONTROL “PACMOTION” EN LA  
AUTOMATIZACIÓN DE PROCESOS, ORIENTADOS A LA FORMACIÓN  
PROFESIONAL DE LOS ESTUDIANTES DE LA ESPECIALIZACIÓN DE  
AUTOMATIZACIÓN INDUSTRIAL ”

**INFORME DE PROYECTO DE GRADUACIÓN**

**Previo a la obtención del título de:**

**INGENIERO EN ELECTRICIDAD ESPECIALIZACIÓN ELECTRÓNICA Y  
AUTOMATIZACIÓN INDUSTRIAL.**

**Presentado por:**

**Carlos Alberto Saldaña Enderica**

**Miguel Segundo Bermúdez Centeno**

**GUAYAQUIL-ECUADOR**

**2012**

## **AGRADECIMIENTOS**

A Dios

A mi madre

Al Ingeniero Manzur

Carlos Alberto Saldaña Enderica

## **AGRADECIMIENTOS**

A Dios

A mis Padres y Hermanos

A la familia Romero Andrade

Al Ingeniero Manzur

A mis Amigos

Miguel Segundo Bermúdez Centeno

## **DEDICATORIA**

A mi Madre

A todos mis amigos

A Marisol.

Carlos Alberto Saldaña Enderica

## **DEDICATORIA**

A Dios

A mis Padres y Hermanos

A mis Amigos

Miguel Segundo Bermúdez Centeno

## TRIBUNAL DE SUSTENTACIÓN

.....  
Dr. Boris Vintimilla

**SUBDECANO DE LA  
UNIDAD ACADÉMICA**

.....  
Ing. Alberto Manzur

**DIRECTOR DEL  
PROYECTO DE GRADUACIÓN**

.....  
Ing. Holger Cevallos

**PROFESOR DELEGADO POR  
LA UNIDAD ACADÉMICA**

## DECLARACIÓN EXPRESA

“La responsabilidad del contenido de este Informe, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITECNICA DEL LITORAL”.

(Reglamento de graduación de la ESPOL)

---

Carlos Alberto Saldaña Enderica

---

Miguel Segundo Bermúdez Centeno

## RESUMEN

En la economía de hoy, la diferencia entre empresas está en obtener mejores resultados y mayor rentabilidad con menos recursos y todo esto sin perder de vista los exigentes requisitos de los clientes y operaciones de manejo que son cada vez más globales en el medio.

En la actualidad la tendencia en la industria se basa en las soluciones que ayudarán a maximizar su productividad y establecer la diferencia entre sus competidores gracias al status productivo de cada una de ellas. Estar a la altura de estas exigencias en el manejo de nuevas tecnologías que permitan cumplir con esos desafíos en la automatización industrial es unos de los objetivos de este proyecto.

Los sistemas de automatización cada vez crecen en tamaño y complejidad es por ello, la importancia de simplificar su estructura de control además de tener la capacidad para integrar equipos dispares tanto en la líneas de producción como en las redes de sistemas, este es el motivo del proyecto, la introducción en el estudio del PACSystems y el desarrollo práctico del PACMotion en la formación profesional de los futuros estudiantes de la



carrera de Ingeniería Eléctrica, especialización Electrónica y Automatización Industrial.

PACSystems es un procesador de alta velocidad con una tecnología patentada que ofrece una mayor producción sin cuellos de botella en la información. El resultado de esto es la llegada rápida de un producto al mercado gracias al control integrado a diferentes disciplinas las cuales nos brindan el uso de menos recursos y mayor información.

PACMotion es un nuevo controlador de movimiento para el RX3i PACSystems el cual maximiza el potencial de una empresa en la línea de producción cuando se tienen necesidades de alto rendimiento que incluyen aplicaciones de alta velocidad y precisión de movimiento. PACMotion proporciona el rendimiento, la flexibilidad y la productividad para hacer la mayoría de estas labores.

Nuestro proyecto se desarrolla de la siguiente manera:

En el capítulo 1 se hace una breve reseña de lo que ha sido la automatización industrial y el impacto que ha causado en estos tiempos, con la única intención de enlazar los capítulos que se han venido estudiando en

la especialización de Electrónica y Automatización Industrial con los nuevos capítulos que se revisarán en este proyecto de graduación.

En el capítulo 2 se hace una descripción detallada de la instalación del PACSystems además de conocer las características y sus especificaciones. De igual manera se describe el sistema PACMotion con cada uno de sus elementos para lograr el movimiento de los servomotores, los requerimientos e instalación del hardware y su configuración además de las recomendaciones que se deben seguir para obtener buenos resultados en la instalación.

En el capítulo 3 se realiza la configuración del sistema RX3i por medio del software de programación "Proficy Machine Edition" además de adquirir un conocimiento básico del programa y la configuración de los módulos del RX3i junto a los modos de comunicación que posee el sistema.

En el capítulo 4 se explica el control y programación del sistema PACMotion estudiaremos la mayoría de los bloques de programación del PACMotion además de realizar un control básico del equipo de entrenamiento que poseemos en el laboratorio de Automatización II.

En el capítulo 5 entraremos a la programación avanzada del sistema PACMotion para eso estudiaremos algunos conceptos previos antes de

desarrollar algunos proyectos didácticos para el aprendizaje de este tema, los proyectos serán realizados en el Software de programación Proficy Machine Edition versión 7.0 y el software para Scada InTouch versión 10.1

Finalmente se presentarán las conclusiones y recomendaciones respectivas en la elaboración de nuestro proyecto.

## ÍNDICE GENERAL

RESUMEN

ABREVIATURAS

ÍNDICES DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICES DE DIAGRAMAS

INTRODUCCIÓN

CAPÍTULO 1 .....	1
AUTOMATIZACIÓN INDUSTRIAL.....	1
1.1 Automatización dedicada o estándar .....	1
1.2 Automatización flexible .....	2
1.3 Nuevo enfoque de la automatización industrial .....	4
1.4 Herramientas de la automatización industrial .....	5
1.4.1 Controlador de automatización programable (PAC). .....	6
1.4.2 PACSystems Rx3i .....	7
1.5 PACMotion.....	8
CAPÍTULO 2.....	10
PACSystems RX3i.....	10

2.1	Características.....	10
2.2	Descripción del sistema PACMotion.....	12
2.2.1	Universal Backplane (IC695CHS012).....	12
2.2.2	Fuente de poder. ....	14
2.2.3	CPU (IC695CPU310).....	15
2.2.4	PACMotion.....	17
2.2.5	Bloque de terminales E/S. ....	22
2.2.6	Terminales y conectores .....	24
2.2.7	βi Series servo systems.....	27
2.3	Instalación del hardware .....	27
2.3.1	Instalación física de la placa base universal .....	28
2.3.2	Barra de puesta a tierra .....	30
2.3.3	Consideraciones para la colocación de los módulos en la placa universal .....	30
2.3.4	Fuente de poder .....	35
2.4	Especificaciones técnicas.....	39
2.4.1	Unidad de fuente de poder .....	39
2.4.2	Unidad central de procesamiento .....	41
2.4.3	PACMotion.....	44

2.4.4	Fiber Terminal.....	49
2.4.5	Servo Motor .....	57
2.4.6	Amplificador .....	58
2.5	Estructura física del sistema PACMotion .....	59
2.6	Práctica No 1. ....	61
2.7	Práctica No 2. ....	62
CAPITULO 3.....		63
CONFIGURACIÓN Y PROGRAMACIÓN DEL SISTEMA RX3i .....		63
3.1	Modos de comunicación. ....	63
3.2	Comunicaciones TCP/IP Ethernet para PACSystems. ....	66
3.3	Estación de manejo. ....	66
3.3.1	Comandos para modificar parámetros.....	66
3.3.2	Comandos de monitoreo.....	67
3.3.3	Comandos de la estación de manejo.....	68
3.3.4	Conexión local. ....	72
3.3.5	Asignación de los pines. ....	73
3.4	Software De Programación “Proficy Machine Edition”. ....	73
3.4.1	Manejo básico del programa.....	73
3.4.2	Configuración de los módulos del PACSYSTEMS. ....	75

3.4.3	BETA I SERVO TRAINER. ....	77
3.5	Introducción a los bloques de movimientos. ....	83
3.5.1	Variables de uso general .....	83
3.5.2	Bloques de funciones.....	87
3.6	Práctica No 3. ....	94
CAPITULO 4.....		96
INTRODUCCIÓN AL CONTROL Y PROGRAMACIÓN DEL SISTEMA		
PACMOTION .....		96
4.1	Programación de los Bloques de funciones de movimiento.....	96
4.1.1	Mc_Power.....	97
4.1.2	Mc_Setposition .....	99
4.1.3	Mc_Jogaxis.....	101
4.1.4	MC_READACTUALPOSITION.....	104
4.1.5	MC_MOVERELATIVE .....	105
4.1.6	MC_MOVEABSOLUTE.....	108
4.1.7	MC_ReadDigitalInput.....	112
4.1.8	MC_WriteDigitalOutput .....	113
4.2	Programación de las I/O: .....	115
4.3	Programación del MC_Power: .....	119

4.4	Programa de control de velocidad manual.....	120
4.5	Control básico del equipo de entrenamiento BETA I SERVO.....	127
4.6	Práctica No 4. ....	131
4.7	Práctica No 5. ....	132
CAPITULO 5.....		133
Automatización Industrial bajo el sistema PACMotion .....		133
5.1	Conceptos básicos de desarrollo para la programación .....	133
5.1.1	Unidades de Usuario .....	133
5.1.2	Sincronización .....	137
5.2	Aplicaciones y desarrollo de proyectos didácticos.....	138
5.2.1	Configuración y programación de los cuatro ejes de los servomotores para la simulación del comportamiento de un reloj. ....	138
5.2.2	Configuración y programación de los cuatro ejes de los servomotores para la sincronización de los mismos.....	155
5.2.3	Simulación y programación para el direccionamiento de paneles solares. 163	
5.2.4	Simulación y programación para el proceso de llenado y sellado de envases. ....	171
5.2.5	Simulación y programación de un brazo de robótico en dos coordenadas. ....	185



5.3	Práctica No 6. ....	193
5.4	Práctica No 7. ....	194
5.5	Práctica No 8. ....	195
5.6	Práctica No 9. ....	196
5.7	Práctica No 10. ....	197

CONCLUSIONES Y RECOMENDACIONES

BIBLIOGRAFIA

## **ABREVIATURAS**

ANN: Artificial Neural Network

BCD: Binario Decimal Codificado

CSV: Comma Separated Values

DCS: Distributed Control System

DDE: Dinamic Data Exchange

DIN: Data Interface Module

E/S: Entradas y Salidas

OEM: Original Equipment Manufacturer

PAC: Programmable Automation Controller

PCI: Peripheral Component Interconnect

PCL: Programmable Logic Control

RX3i: Procesador Intel

SCADA: Supervisory Control And Data Acquisition

SNP: Sequence Number Protection

## ÍNDICES DE FIGURAS

Figura 2.1 Universal Backplane .....	12
Figura 2.2 Fuente de Poder .....	14
Figura 2.3 Unidad Central de Procesamiento. ....	15
Figura 2.4 Módulo PACMotion .....	18
Figura 2.5 Fiber I/O PACMotion [1].....	19
Figura 2.6 Fiber Terminal block .....	22
Figura 2.7 Terminal TB1 [1] .....	24
Figura 2.8 SLOT 0 de la Placa Base [1].....	25
Figura 2.9 Slot de Expansión [1].....	26
Figura 2.10 <i>βi series servo</i> [1].....	27
Figura 2.11 Dimensiones para instalación del Backplane [1].....	28
Figura 2.12 Dimensiones laterales [1].....	29
Figura 2.13 Posición del Backplane [1].....	30
Figura 2.14 CPU en slot 0 [1].....	32
Figura 2.15 Fuente de poder slot 6 [1].....	32
Figura 2.16 CPU en slot 1.....	33
Figura 2.17 Fuente de poder slot 11 [1].....	33

Figura 2.18 Ubicación incorrecta de CPU [1] .....	34
Figura 2.19 Ubicación incorrecta en el slot 0 [1] .....	34
Figura 2.20 Características externas [1] .....	35
Figura 2.21 Terminales de Conexión [1] .....	37
Figura 2.22 Protección contra sobre tensiones [1].....	38
Figura 2.23 Leds Indicadores del PACMotion.....	44
Figura 2.24 Leds Indicadores del FTB .....	49
Figura 2.25 Estructura física del sistema PACMotion [1] .....	59
Figura 2.26 Diagrama de bloques del sistema PACMotion [1].....	60
Figura 3.1 Diagrama de redes de comunicación.....	64
Figura 3.2 Puertos de comunicación del PACSystems [1].....	65
Figura 3.3 Puertos COM1 y COM2 .....	65
Figura 3.4 Asignación De Pines.....	73
Figura 3.5 Crear proyecto .....	74
Figura 3.6 Navigator .....	75
Figura 3.7 Configuración de módulos .....	77
Figura 3.8 Configuración FTB.....	79
Figura 3.9 Parámetros Open Load Detect desactivados .....	79
Figura 3.10 Parámetros de configuración del servo motor .....	80

Figura 4.1 Mc_Power .....	97
Figura 4.2 Mc_Setposition .....	99
Figura 4.3 Mc_Jogaxis .....	101
Figura 4.4 Mc_Readactualpos .....	104
Figura 4.5 Mc_Moverelative .....	105
Figura 4.6 Mc_Moveabsolute .....	108
Figura 4.7 Mc Read Digital .....	112
Figura 4.8 Mc Write Digital .....	113
Figura 4.9 Programación de las entradas i0001- i0002 .....	115
Figura 4.10 Programación de la salida Q0001 .....	116
Figura 4.11 Programación de la salida q0002 .....	117
Figura 4.12 Programación de las entradas analógicas .....	118
Figura 4.13 Bloque Mc_Power y Mc_Setposition .....	119
Figura 4.14 LLlamado de las subrutinas .....	121
Figura 4.15 Pulsadores de encendido y apagado del servomotor. ....	123
Figura 4.16 Encendido del eje 1 del modulo 4, usando el bloque .....	123
Figura 4.17 Uso del bloque Mc_Setposition .....	124
Figura 4.18 Pulsadores ON y OFF para habilitar el movimiento del eje. ....	125
Figura 4.19 Uso del bloque Mc_Stop para detener el eje. ....	126

Figura 4.20 Elección del sentido de giro del eje.....	127
Figura 4.21 Escalamiento y conversión del tipo de dato.....	128
Figura 4.22 Ingreso de valor de velocidad por teclado. ....	129
Figura 4.23 Bloque Mc_Jogaxis utilizado para variar la velocidad en el eje. .....	130
Figura 5.1 Número de conteos en 360 grados.....	134
Figura 5.2 Simulación y programación del control de los ejes de los servomotores. ....	138
Figura 5.3 Configuración Settings.....	140
Figura 5.4 Configuración parámetros.....	141
Figura 5.5 Configuración FTB Inputs .....	142
Figura 5.6 Configuración de la pestaña terminal.....	145
Figura 5.7 Habilidad de pestaña terminal.....	145
Figura 5.8 Configuración para el eje .....	146
Figura 5.9 Diagrama de bloques PACMotion.....	151
Figura 5.10 Comunicación del PACSystems con InTouch.....	154
Figura 5.11 MC_GEARIN. ....	159
Figura 5.12 Esquema de las variables desde el Scada. ....	163
Figura 5.13 SVC REQ.....	166

Figura 5.14 Rol Word.....	166
Figura 5.15 Mc_Move absolute.....	167
Figura 5.16 Control principal de los ejes.....	170
Figura 5.17 Scada paneles solares.....	170
Figura 5.18 Configuración de ejes. ....	173
Figura 5.19 MC_GEARINPOS.....	176
Figura 5.20 Ventana SETEO del Scada. ....	182
Figura 5.21 Ventana PROCESO del Scada.....	183
Figura 5.22 Ventana HISTÓRICOS del Scada. ....	184
Figura 5.23 Control de posición de un brazo robótico en dos ejes .....	192

## ÍNDICE DE TABLAS

Tabla I. Módulos disponibles.....	12
Tabla II. Unidad de Fuente de poder .....	39
Tabla III. Estado de leds del CPU .....	41
Tabla IV. Especificaciones Técnicas del CPU .....	43
Tabla V. Leds indicadores de estado de operación de los ejes de PMM ....	45
Tabla VI. Leds indicadores de estado de operación del de PMM .....	45
Tabla VII Leds indicadores de estado de operación de las I/O del FTB .....	50
Tabla VIII. Leds indicadores de estado de operación del FTB.....	51
Tabla IX. Especificaciones técnicas del FTB .....	54
Tabla X. Especificaciones técnicas del servo motor .....	57
Tabla XI. Especificaciones técnicas del amplificador.....	58
Tabla XII. Comandos de la estación de manejo.....	68
Tabla XIII. Funciones de los bloques de programación .....	87
Tabla XIV. Límites de velocidad para los cuatro ejes .....	137
Tabla XV. Rangos Mínimos y Máximos del Eje Maestro. ....	156



## ÍNDICES DE DIAGRAMAS

Diagrama 1. Control de los cuatro ejes de los servomotores.....	153
Diagrama 2. Control del movimiento de paneles solares.....	169
Diagrama 3. Control de la sincronización del proceso de llenado y sellado. .....	180
Diagrama 4. Control del movimiento en dos ejes de un brazo robótico. ....	191



# INTRODUCCIÓN

El proyecto de graduación está centrado en la formación profesional de los futuros Ingenieros Eléctricos de la Escuela Superior Politécnica del Litoral. La necesidad de avanzar en los conocimientos de nuevas tendencias en la Automatización Industrial realizando un trabajo didáctico para los alumnos de la especialización de Electrónica y Automatización Industrial, añadiendo al currículum de estudio la introducción al PACSystems y un estudio avanzado del PACMotion los cuales son sistemas de automatización Industrial patentados por General Electric y ampliamente reconocidos en el mercado mundial.

El objetivo principal que seguimos en este proyecto es integrar los nuevos conocimientos relacionados al PACSystems a los ya estudiados en la especialización de Electrónica y Automatización Industrial en el área de Automatización logrando una mayor preparación y competencia en nuestro campo, así para lograr este objetivo desarrollamos lo siguiente:

1. Configuración del hardware y comunicación entre los módulos de PACSystems y los equipos de entrenamiento "Beta i Servo" que se

encuentran instalados en el laboratorio de Automatización Industrial II.

2. Configuración de los módulos del PACSystems específicamente en la línea PACMotion para el manejo y aprendizaje de los alumnos.
3. Aplicaciones y desarrollo de proyectos de Automatización Industrial didácticos en el área de PACMotion para el desarrollo de las clases del curso de Automatización II.

Las actividades de este proyecto han sido gracias a un estudio completo de dos equipos de entrenamientos “Beta i Servo” con su respectivo controlador donados por la compañía General Electric a la Escuela Superior Politécnica del Litoral los cuales se encuentran en el laboratorio de Automatización Industrial II.

# CAPÍTULO 1

## AUTOMATIZACIÓN INDUSTRIAL

### 1.1 Automatización dedicada o estándar

Los elementos que representan a esta tecnología son los llamados relevadores electromecánicos, los cuales una vez instalados, hacen indeseable la acción de llevar a cabo modificaciones en su lógica de operación, debido a lo problemático y conflictivo que resulta cambiar el diseño de un sistema de control. Los relevadores electromecánicos están constituidos por una bobina que al energizar sus terminales produce un campo magnético, que a su vez provoca que una serie de contactos móviles se cierren o abran, interrumpiendo o permitiendo respectivamente el paso de la corriente eléctrica a través de ellos, tomando en cuenta esta manera de actuar, un relevador electromecánico tiene dos estados de operación, uno cuando su bobina se encuentra sin energía que equivale al 0 lógico y cuando su bobina se encuentra energizada que equivale al 1 lógico.

El estado de los contactos de los relevadores electromecánicos se puede ramificar a muchas otras ubicaciones, haciendo sentir así sus efectos en varios puntos a lo largo del circuito de control. Aprovechando las características anteriores, los relevadores electromecánicos a través de sus contactos alimentan las bobinas de otros relevadores, esto es, controlan otros bloques de relevadores, que al estar agrupados en circuitos realizan las operaciones lógicas del sistema de control automático.

## **1.2 Automatización flexible**

Los sistemas de control que trabajan bajo esta filosofía, basan su toma de decisiones por medio de la ejecución de instrucciones codificadas, las cuales están almacenadas en un circuito de memoria e interpretadas por un microprocesador o microcontrolador. Lo importante de la automatización flexible es que si resulta necesario modificar el sistema de control, basta con cambiar las instrucciones codificadas.

La automatización flexible está conformada por un sistema de Control Lógico Programable (Programable Logic Control "PLC") capaz de realizar el procesamiento de señales binarias basándose en un programa establecido por el usuario, y que contiene puertos de entrada, salida y transmisión de datos con la debida interacción para su

operación. De esta manera, con las señales de entrada y salida se pueden controlar directamente secuencias mecánicas, o procesos fabriles.

Los campos donde puede tener aplicación un sistema PLC prácticamente son innumerables. Estos sistemas de control se destinan principalmente para las funciones de control de procesos, donde se encargan de que cada paso o fase del proceso sea efectuado en el orden cronológico correcto y sincronizado que previamente fue establecido.

Un sistema PLC se basa en un tipo de computadora de uso específico, diseñada para ambientes de trabajo en donde su misión primordial es el control de procesos industriales, que pueden ser constituidos por diversos tipos de maquinaria, robots, líneas de ensamble, etc.

La tecnología que se propone con la utilización de los PLC es muy versátil en lugares donde se requiere de automatizar un proceso industrial. Prácticamente esta tecnología puede adaptarse a cualquier ambiente de operación fácilmente y sin mayores problemas, por otra parte, se trata de una tecnología que se encuentra dentro de los llamados sistemas de automatización flexibles, por lo que tiene una amplia gama de prestaciones adicionales.

Cuando se usa el enfoque de automatización flexible, a la secuencia completa de instrucciones que confeccionan al programa que controla el desempeño del proceso de producción se le llama “programa de control”. Este programa tiene que ser desarrollado por el usuario en función de los requerimientos que son propios del proceso que tiene que ser automatizado, por lo que se tiene que recurrir al empleo de diagramas de flujo para que todos los detalles queden plasmados en el programa de control.

En resumen, al sistema de control automático basado en la tecnología del PLC se le considera como un “sistema programable”, y además se le reconoce como uno de los principales precursores del enfoque de automatización flexible.

### **1.3 Nuevo enfoque de la automatización industrial**

El nuevo enfoque de la automatización está siendo aplicado principalmente en mejorar al máximo la calidad del proceso y luego mantener esta calidad en el producto a través de operadores calificados. Otro cambio importante en el nuevo enfoque de la automatización es la demanda por la flexibilidad de la línea de producción. Las empresas tienen una producción flexible cuando tienen la habilidad de cambiar la manufactura de un producto A por una



manufactura diferente para un producto B sin tener que reconstruir completamente la línea de producción, también una línea de producción es flexible cuando se pueden cambiar parámetros bases como la producción por día o adición o remoción de procesos dentro de la línea sin afectar la calidad del producto Esta capacidad de cambios de células de producción es fácilmente implementada con un buen diseño previo en la automatización de la línea de producción.

#### **1.4 Herramientas de la automatización industrial**

La tecnología informática, junto con los mecanismos y procesos industriales, pueden ayudar en el diseño, implementación y monitoreo de sistemas de control. Un ejemplo de un sistema de control industrial como ya mencionamos es el controlador lógico programable (PLC). La Interfaz hombre-máquina (HMI) o interfaz hombre-computadora, se suelen utilizar para comunicarse con los PLC's y otros equipos, pero también existen diferentes tipos de herramientas para la automatización tales como:

ANN - Artificial neural network

DCS - Distributed Control System

SCADA - Supervisory Control and Data Acquisition

PLC - Programmable Logic Controller

PAC - Programmable automation controller

Instrumentación, control de movimiento, robótica.

#### **1.4.1 Controlador de automatización programable (PAC).**

A diferencia de los sistemas PLC, que son limitados en términos de escalabilidad o la capacidad de adaptación, un controlador de automatización programable puede llevar libertad y flexibilidad a todas las necesidades de la empresa. PACSystems de General Electric es una plataforma inteligente que permite tomar el control, proporcionando conectividad en tiempo real y simplificando la integración en gran medida a los procesos de producción.

PACSystems ofrece una funcionalidad multi-dominio incluyendo la lógica, el control de movimiento, HMI, y control de procesos en un solo andén, cuenta con una plataforma de desarrollo común impulsada por Proficy Machine Edition, el cual es el software de desarrollo, quien permite la migración rápida y sin problemas al tiempo que permite la comunicación más allá de la planta a todas las áreas de la empresa, debido a que está construido sobre plataformas basadas en estándares ya establecidos.

### **1.4.2 PACSystems Rx3i**

El PACSystems RX3i es un controlador programable de automatización industrial de la familia PACSystems.

Al igual que el resto de la familia, el RX3i PACSystems cuenta con un motor de control y entorno de programación universal el cual ofrece la portabilidad de aplicaciones a través de múltiples plataformas de hardware y una convergencia de las opciones de control.

Entre las características del RX3i tenemos:

Un controlador de alto rendimiento con un procesador Intel de 300 MHz y 10 MB de memoria de avanzada programación y rendimiento.

Una placa base universal de alta velocidad con bus PCI la cual funciona a 27 MHz para datos de alto rendimiento que optimiza el rendimiento, además de módulos de entrada-salida analógica, módulos de entrada-salida discreta, módulo neumático, módulos de expansión, modulo de movimiento.

Como ejemplo tenemos que los módulos de entrada proporcionan la interfaz entre el PLC y los dispositivos externos de entrada, tales como sensores de proximidad, pulsadores, interruptores y el calibrador de

BCD, los módulos de salida proporcionan la interfaz entre el PLC y los dispositivos externos de salida tales como contactores, interponiendo relés, pantallas BCD y luces indicadoras.

## **1.5 PACMotion**

Dentro del paquete RX3i se encuentra PACMotion el cual es el controlador de automatización programable de movimiento el cual será objeto de estudio en este proyecto.

General Electric ofrece controles de movimiento para un amplio rango de aplicaciones y la configuración del sistema provee la flexibilidad para escoger un sistema que optimiza la presentación y los costos de inversión en una empresa.

El PACMotion es un versátil controlador de servomotores que combina los beneficios en mejoras para la productividad con la capacidad de tiempo real de sincronización para más de 40 ejes.

Hoy en día los sistemas de automatización deben mejorar la flexibilidad y la escalabilidad para mantenerse al ritmo de la introducción de nuevos productos al mercado. Como podemos ver en el embalaje de productos se requiere de rápida rotación además de una mayor variabilidad esto lo

podemos ver en las industrias como la farmacéutica, alimenticia, industria de bebidas y productos de consumo que son envasados.

PACMotion es una herramienta de alto rendimiento, multi-eje de control de movimiento, integrado estrechamente con un proceso del controlador de automatización (PAC), interfaz de operador y amplias opciones de comunicación, todos vinculados por un software de gran alcance que nos entrega la capacidad para empaquetado, llenado y sellado de productos, además de transportadores inteligentes, encartonadoras, etiquetado de alta velocidad, etc. Así PACMotion es parte de un sistema de automatización completo que integra todo lo necesario para el control de movimiento de manera fácil, precisa y segura.

# CAPÍTULO 2

## PACSYSTEM RX3i

El controlador PACSystems RX3i es una incorporación a la familia PACSystems de controladores de automatización programables (PAC). Al igual que el resto de la familia, el PACSystems RX3i se caracteriza por tener un motor de control y un entorno de programación universal, que facilita la portabilidad de aplicaciones a diversas plataformas de hardware y ofrece una convergencia de todas las opciones de control del PACSystems.

### 2.1 Características

- Controlador de alto rendimiento con microprocesador Intel de 300 MHz y 10 MB de memoria de usuario que elimina la necesidad de disponer de varios controladores y simplifica el control.

- Placa base universal con bus PCI de alta velocidad (27 MHz) para una rápida transmisión de datos en operaciones de E/S complejas y un bus serie para entradas y salidas. La placa base universal admite también el intercambio en tiempo de ejecución del proceso para reducir al mínimo los tiempos de inactividad.
- Dispone de una amplia gama de módulos de E/S (más de 40 tipos disponibles) para todo tipo de aplicaciones, sencillas o complejas, así como diversos módulos de red.
- PACSystems RX3i incluye una plataforma que proporciona redundancia, ofreciendo así el mejor rendimiento y la mayor facilidad de uso.
- El motor de control portátil de PACSystems proporciona un elevado rendimiento en diferentes plataformas, haciendo posible que los fabricantes de equipos originales (OEM) y usuarios finales que usan aplicaciones muy diferentes puedan escoger el hardware del sistema de control que mejor se adapte a sus necesidades.

## 2.2 Descripción del sistema PACMotion

A continuación se listan los módulos disponibles en el laboratorio.

**Tabla I. Módulos disponibles.**

Nombre	Código
1. Power Supply	IC695PSA040
2. Central Processing Unit	IC695CPU310
3. Multi-axis Motion Controller	IC695PMM335
4. Universal Backplane	IC695CHS012.
5. Fiber I/O Terminal Block	IC695FTB001
6. Motor Model	ZA06B-0115-B203 ( $\beta$ 0.5/6000iS)
7. Amplifier	$\beta$ SVM1-20i *ZA06B-6130-H002

### 2.2.1 Universal Backplane (IC695CHS012)

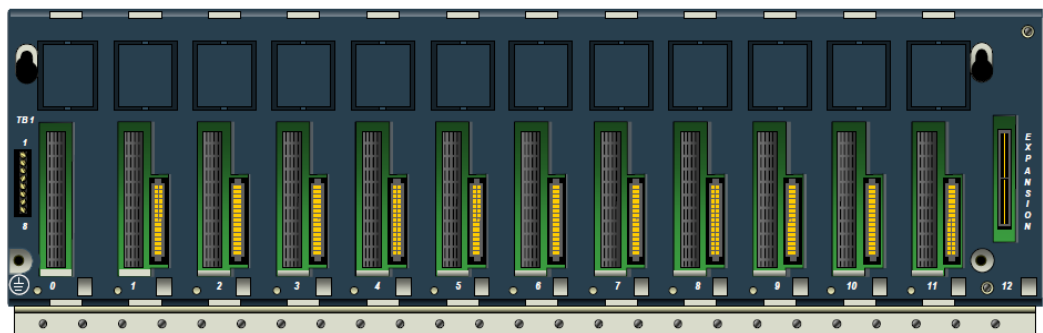


Figura2.1 Universal Backplane

La Placa Base Universal soporta módulos con base PCI, de la serie IC695, así como base serial IC694.



**Características:**

- Terminal Strip al final del extremo izquierdo con entrada de +24V aislados.
- Punto de puesta a tierra.
- Barra integral de puesta a tierra para conectar los módulos.
- Conector de expansión serie para la conexión de expansión serial y placas bases remotas.
- Slots con números impresos en la base para ser usados como referencia en la configuración.

### 2.2.2 Fuente de poder.

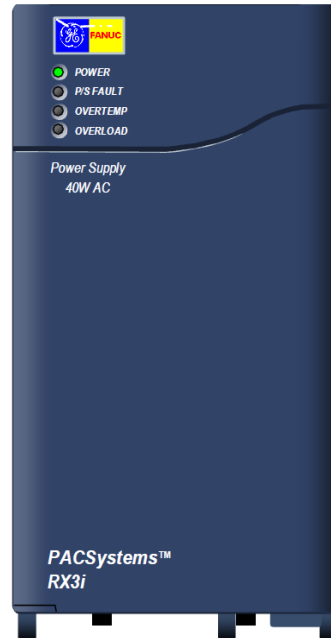


Figura 2.2 Fuente de Poder

La fuente de poder IC695PSA040 tiene una potencia de 40 Watts y opera con una alimentación que va desde 85 hasta 264 VAC o 100 VDC hasta 300 VDC.

Esta fuente de poder proporciona tres salidas:

- (+) 5.1 VDC.
- (+) 24 VDC utilizados para energizar relés de potencia.
- (+) 3.3 VDC esta salida es usada por interminables módulos RX3i de la serie IC695.

Si el número de módulos instalados en el sistema excede la potencia que entrega nuestra fuente de poder entonces será necesario instalar estos módulos en una placa de expansión o una placa base remota.

### 2.2.3 CPU (IC695CPU310)

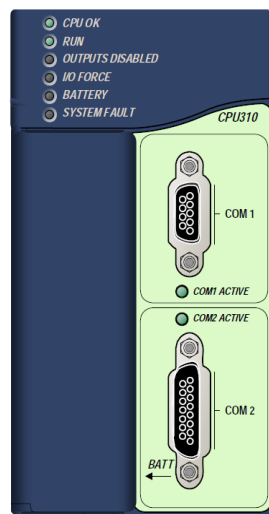


Figura 2.3 Unidad Central de Procesamiento.

El RX3i CPU es usado para desempeñar control en tiempo real de procesos. El CPU se comunica con el programa y equipos HMI por medio de comunicación serial a través de un puerto usando protocolo SNP esclavo, se comunica con entradas y salidas, módulos inteligentes a través del bus del Backplane.

## CARACTERÍSTICAS PRINCIPALES

Alta velocidad de bus PCI, para un mayor rendimiento de nuevas y avanzadas I/O.

- Backplane serie para facilitar la migración de la serie existente 90-30 I/O.
- Comunicación Ethernet vía rack por medio del modulo IC695ETM001.
- Tiempo de sincronización con el servidor de tiempo SNTP en red Ethernet cuando se utiliza con un módulo de Ethernet con la versión de firmware 5.0 o posterior.
- Contiene 10 Mbyte de memoria respaldados por la batería y 10 Mbyte de memoria flash no volátil.
- Suministra acceso a la dimensión de memoria usando la tabla de referencia %W.
- Datos configurables y memoria del programa.
- Programación en Diagrama de escalera, texto estructurado, diagrama de bloques funcionales, y C.
- Soporta auto-ubicación de variables simbólicas que pueden utilizar cualquier cantidad de memoria de usuario.
- Tabla de referencia, tamaños de 32Kbits para %I y %Q y hasta 32 Kwords para analógica %AI y %AQ.

- Soporta la mayoría de los módulos Series 90-30 y racks de expansión.
- Admite hasta 512 bloques de programación. El tamaño máximo de un bloque es de 128KB.
- Referencias de Bit-de-palabra cruzadas le permite especificar los bits por palabra en memoria retentiva como las entradas y salidas de las expresiones booleanas, módulos funcionales.
- Sistema de firmware actualizable.
- Dos puertos serial, un puerto serial RS-485 y un puerto serial RS-232.

#### **2.2.4 PACMotion.**

El PAC Motion Multi-axis Motion Controller (PMM335) es un módulo de alto rendimiento, fácil de utilizar para el control de servomotores e integrado con el CPU de la familia PACSystems RX3i. Este controlador de movimiento combina movimientos integrados y lógica de maquina con el rendimiento, flexibilidad y escalabilidad necesarias para la automatización de maquinas avanzadas.

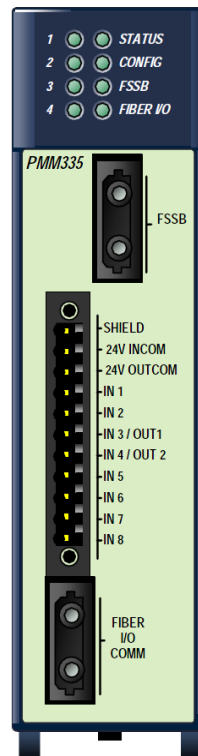


Figura 2.4 Módulo PACMotion

### Características principales

**Cuatro ejes:** Además de un eje virtual, cada módulo PMM335 puede controlar hasta cuatro ejes. Un eje virtual adicional (tiempo de base) y un encoder externo adicional puede ser configurado en él. Se pueden emplear ocho módulos en un rack para un total de 40 ejes.

**Faceplate I/O:** El faceplate I/O incorporado en el PMM334 tiene cuatro entradas digitales de propósito general de 24 Voltios, dos entradas digitales de alta velocidad de 24 Voltios con detención de ausencia de cable, y dos configurables como entradas o salidas digitales de

propósito general de 24 voltios, además de servir como E/S, es configurable para funciones de movimiento específicas tales como entradas de palpación, límite de finales de carreras, switch de entrada, marcador de entradas.

**Fiber i/o terminal block:** El PMM335 soporta además al módulo Fiber I/O Terminal Block (FTB) IC965FTB001.

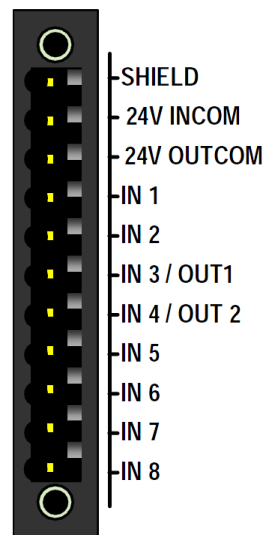


Figura 2.5 Fiber I/O PACMotion[1]

**Rendimiento:** La frecuencia de actualización del lazo de posición del PMM335 es de 500us. La tasa del lazo de actualización es de 125us. La planificación de rutas de movimiento se hace cada 1 ms para todos los cinco ejes.

**Sincronización:** El PMM335 provee una sincronización o puesta en marcha de hasta ocho ejes, para aplicaciones electrónicas de levas o de engranajes, los ejes en el rack pueden ser usados como máster para uno o más ejes esclavos en cualquiera de los módulos en un rack.

**Perfiles Cam:** Las levas electrónicas se pueden crear usando el built-in editor en el software Proficy Machine Edition o importarlos a través de un archivo CSV (variable separada por comas). El editor de levas permite agregar a máster y esclavos puntos en una tabla o mediante la manipulación gráfica. Datos de puntos se pueden montar utilizando el 1er, 2do, 3ero, o 5to grado de ajuste de ranura de la curva. Un perfil de leva se puede subdividir en segmentos (3 puntos mínimo por segmento) con un ajuste de grado diferente de curva para cada segmento. Hasta 2048 perfiles de leva pueden ser descargados y almacenados en el RX3i CPU al mismo tiempo. Hasta 256 perfiles de leva pueden ser seleccionados y almacenados en cada módulo. Los perfiles de leva pueden ser recogidos de forma dinámica según las necesidades.

**Interrupciones:** Se proporcionan hasta tres interrupciones para cada módulo PACMotion. Una interrupción puede ser configurada basada en el tiempo de actualización el cual es configurable hasta 2ms.



Cualquiera de las tres interrupciones se puede configurar como E/S de evento impulsado por las interrupciones.

**Digital Cam Switch:** El Digital Cam Switch (conmutador límite programable) del PMM335 ofrece hasta cuatro salidas (pistas) utilizando salidas ya sea de forma regular o alta velocidad en el faceplate o el FTB. Cada pista puede tener hasta 8 interruptores.

**Diagnóstico Lógico de Bloque:** Para ayudar en la puesta en marcha y limpiar, el software Proficy Machine Edition tiene la capacidad del diagnóstico lógico de bloques el ser cual puede descargado en el RX3i CPU y ejecutado sin alterar la lógica principal del programa.

**Registro de Datos:** El PMM335 proporciona la capacidad de registro de datos durante el tiempo de ejecución. Utilizando Proficy Machine Edition, estos datos pueden ser posteriormente cargados y mostrados.

**PLC open Compliance:** El módulo PMM335 esta designado para ser compatible con la especificación del PLCopen. Toda funcionalidad de movimiento es controlada por funciones especializadas y bloques integrados en el CPU. Las capacidades de los bloques de movimiento funcionales incluyen mezcla y almacenamiento en búfer de bloques, velocidad, aceleración, limitación del movimiento y torque.

## 2.2.5 Bloque de terminales E/S.

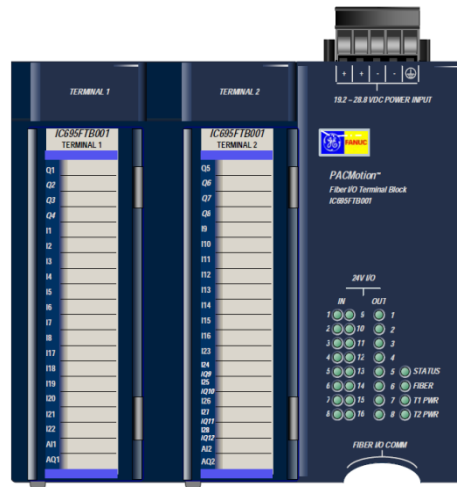


Figura 2.6 Fiber Terminal block

El Fiber Terminal Block (FTB) proporciona una interfaz activa de fibra que extiende las E/S del Controlador Multi-ejes PACMotion. El FTB proporciona un amplio rango de E/S configurables para los ejes del PMM335.

El FTB permite realizar control de movimientos ubicando este a 100 metros de separación del rack RX3i. Los servo amplificadores y E/S de movimiento pueden ser distribuidas con un máximo de 100 metros entre nodos, lo que permite 400 metros como máximo desde el controlador.

Además de las E/S, FTB I/O es configurable para movimientos específicos tales como entradas TouchProbe, finales de carrera, etc.

### **Características principales**

- Doce entradas de alta velocidad de 5 voltios con detección de falla. Seis pueden ser configuradas para operar como simples entradas. Cuatro pueden ser configurados para operar como entradas o salidas diferenciales.
- Dieciséis ópticamente aisladas de 24 voltios.
- Ocho salidas ópticamente aisladas de 24 voltios con protección de corto circuito.
- Dos entradas análogas de propósito general de -10 voltios.
- Dos salidas análogas de propósito general de -10 voltios.
- Montaje en riel DIN que permite una cómoda ubicación de I/O.
- Bloque extraíble RX3i de terminales para brindar facilidad de uso.
- Una fuente de alimentación 5VDC para encoder externo de cuadratura.

## 2.2.6 TERMINALES Y CONECTORES

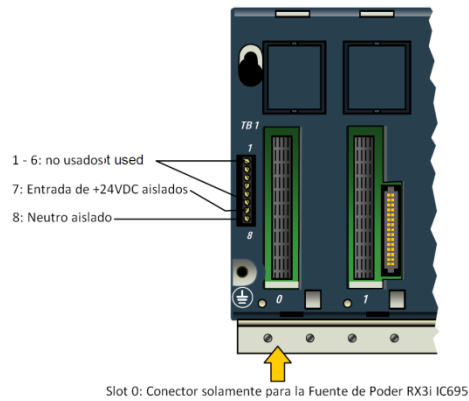


Figura 2.7 Terminal TB1 [1]

**TB1.-** El terminal de conexión TB1 está disponible para conectar una fuente de alimentación externa aislada de +24VDC en los terminales 7 y 8 específicamente.

Esta fuente es requerida por algunos módulos de las series IC693 e IC694.

Se puede usar cable individual desde el número 14 hasta 22 AWG. Los terminales del 1 al 6 no son utilizados.

**SLOT 0.-** El Slot 0 está ubicado en la esquina izquierda del Universal Backplane. En este Slot solo se permite instalar la fuente de poder de las serie IC695. Dependiendo de la fuente de alimentación a utilizar,

esta puede llegar a ocupar dos slots debido a su tamaño pero en principio solo ocupa un slot quedando el otro sin uso alguno.

El CPU se instala en la placa base montado sobre dos slots ocupando en principio solo uno de ellos. Aunque el CPU físicamente está conectado al slot de la parte derecha, en la programación del hardware se toma como referencia el slot anterior, es decir, el slot que está del lado izquierdo. El CPU puede instalarse sobre los slots 0 y 1 sin ninguna preocupación de tocar el slot 0 ya que este se conecta en realidad en el slot 1.

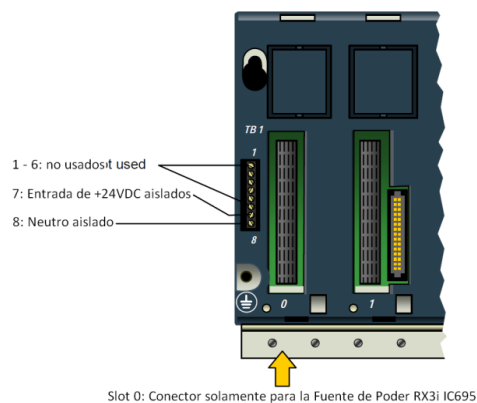


Figura 2.8 SLOT 0 de la Placa Base [1]

## SLOT 1 AL SLOT 11

Los Slots del 1 al 11 tienen dos conectores:

- Para el sistema RX3i basados en bus PCI
- Para el sistema RX3i basado en bus Serial

Cada uno de estos Slots acepta módulos compatibles

- IC695 fuente de poder.
- IC695 unidad central de procesamiento
- IC695/IC694/IC963 I/O
- Módulos opcionales

### SLOT DE EXPANSION Y BARRA PUESTA A TIERRA

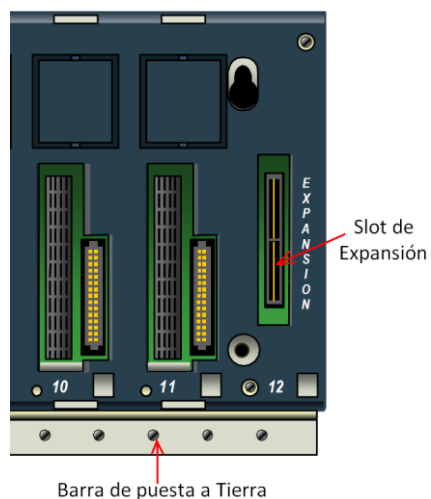


Figura 2.9 Slot de Expansión [1]

En la parte final derecha de la placa base se encuentra el Slot 12 el cual es un conector de expansión. Este conector solo puede ser utilizado para un Módulo Serial de Expansión RX3i (IC695LRE001). Dos módulos de expansión no pueden ocupar esta misma ranura.

### 2.2.7 $\beta$ i SERIES SERVO SYSTEMS

La serie  $\beta$ i de servomotores tienen un alto rendimiento y baja inercia con un encoder serial incorporados. Cada servomotor debe ser usado con su correspondiente amplificador.



Figura 2.10  $\beta$ i series servo [1]

### 2.3 Instalación del hardware

Se debe de tener mucho cuidado en los siguientes puntos:

- No instalar componentes que no son compatibles con el sistema.
- Al momento de cambiar módulos o realizar cualquier tipo de instalación apagar la fuente de alimentación.

El incumplimiento de estos puntos mencionados anteriormente puede llegar a generar peligro de explosión y dañar componentes del sistema.

### 2.3.1 Instalación física de la placa base universal

Las dimensiones recomendadas por el fabricante para la instalación de la placa base universal se muestran a continuación.

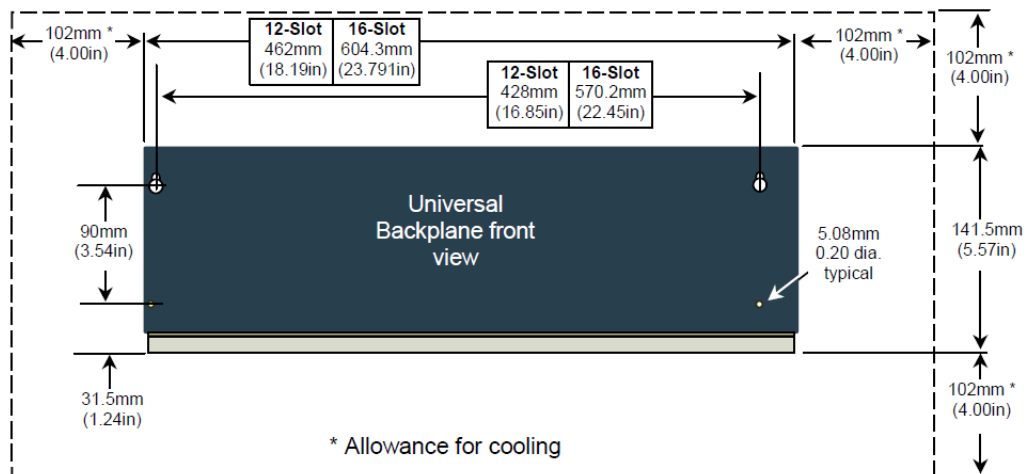


Figura 2.11 Dimensiones para instalación del Backplane[1]

Las dimensiones establecidas tienen que brindar un fácil montaje de los módulos además de un cómodo acceso a los usuarios a las partes eléctricas manipulables. El alojamiento debe proporcionar una ventilación adecuada para evitar que los módulos tengan un sobrecalentamiento que de origen a posibles fallas, además debe constar con una protección equivalente a NEMA/UL Tipo 1 o un grado de protección IP20 (IEC60529).



A continuación se muestran las dimensiones laterales recomendadas por el fabricante. Este dimensionamiento no incluye el espacio libre para el cableado y conectores adicionales.

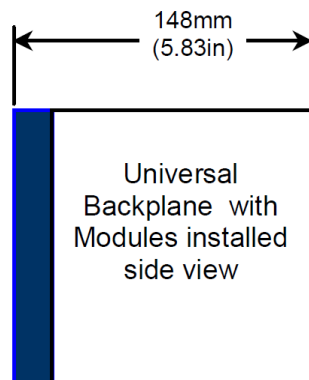


Figura 2.12 Dimensiones laterales [1]

### **Orientación del montaje de la placa base.**

La orientación de la placa base debe de ser horizontal ya que esta posición proporciona una facilidad al flujo del aire para la ventilación de los módulos. Una posición distinta de la horizontal puede afectar el rendimiento del sistema, por lo tanto no es recomendable.

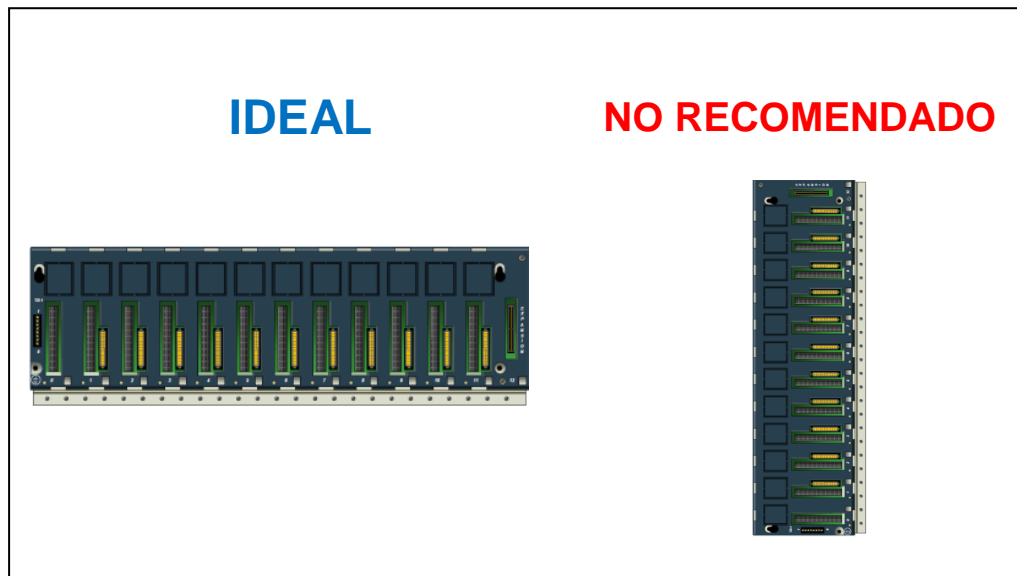


Figura 2.13 Posición del Backplane [1]

### 2.3.2 Barra de puesta a tierra

Los módulos instalados en la placa base pueden ser aterrizados en la barra de puesta a tierra que esta posee usando tornillos tamaño M3. El torque recomendado es de 4 in/lb como máximo.

### 2.3.3 Consideraciones para la colocación de los módulos en la placa universal

- Los módulos que corresponde a las fuentes de alimentación IC695 pueden ser instalados en cualquiera de los slots. Las fuentes DC ocupan un solo slot mientras que las fuentes AC ocupan dos slots.

Las fuentes de alimentación de las series IC694 e IC693 no pueden ser instaladas en esta la placa base.

- Los módulos de entradas y salidas I/O pueden ser instalados en cualquiera de los slots con excepción del slot 0. Estos módulos solo aceptan fuentes de alimentación de la serie IC695 y slot de expansión.
- El último slot de la derecha es un módulo de expansión y solo puede ser usado para un módulo de expansión serial opcional IC695LRE001.
- El CPU puede ser instalado en cualquiera de los slots excepto en el slot de expansión.
- La ubicación de la ranura configurada para el CPU debe coincidir con la ubicación real del CPU.

## Ejemplos de instalación correcta de los módulos en el Backplane

En los gráficos siguientes se muestra la instalación correcta.

En la figura 2.14 se encuentra instalado el CPU en el slot 0 mientras que la fuente de poder en el slot 2

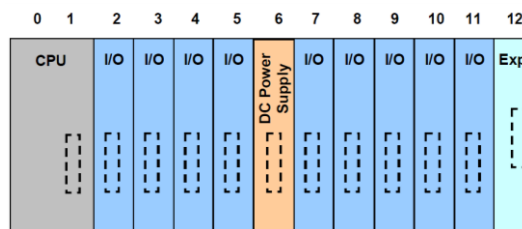


Figura 2.14 CPU en slot 0 [1]

En la figura 2.15 el CPU se encuentra instalado en el slot 0 mientras que la fuente de poder en el slot 6. Nótese que esta fuente de DC y solo ocupa un slot.

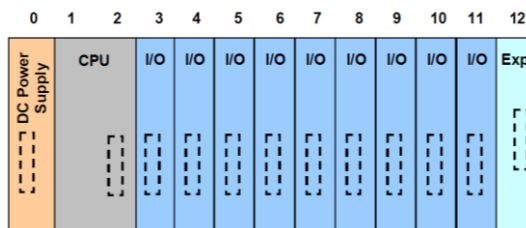


Figura 2.15 Fuente de poder slot 6 [1]

En la figura 2.16 se muestra el CPU instalado en el slot1 con la fuente de poder (DC) instalada en el slot 0.

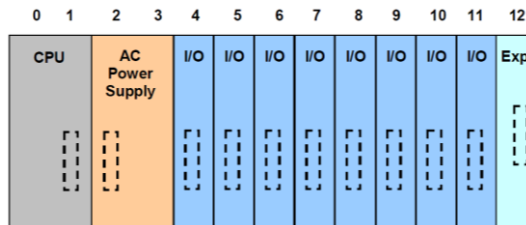


Figura 2.16 CPU en slot 1[1]

### Instalación incorrecta de los módulos en el Backplane

La fuente de poder puede ser instalada en cualquiera de los slots disponibles del Backplane pero no debe de ser instalada en el slot 11.

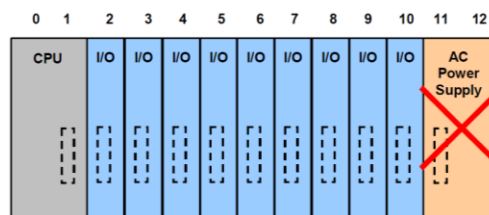


Figura 2.17 Fuente de poder slot 11 [1]

El CPU puede ser instalado en cualquiera de los slots disponibles del Backplane pero no puede ser instalado en el slot 11.

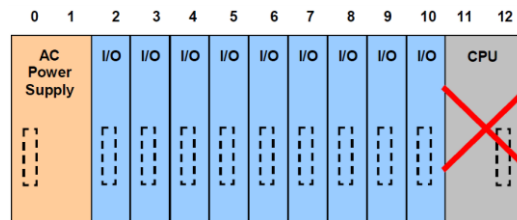


Figura 2.18 Ubicación incorrecta de CPU [1]

El slot 0 solo puede ser utilizado en la instalación de las fuentes de poder y el CPU, (físicamente ocuparía el slot2). Ningún otro módulo puede ser instalado en este slot.

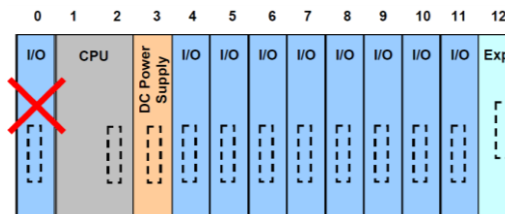


Figura2.19 Ubicación incorrecta en el slot 0 [1]

### 2.3.4 Fuente de poder

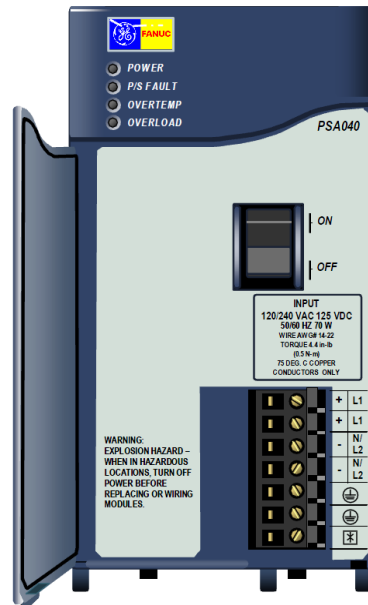


Figura 2.20 Características externas [1]

#### Leds indicadores

La fuente de poder posee cuatro leds indicadores. Estos son:

- POWER: (color verde o ámbar)

Cuando este led está de color verde indica que la fuente se encuentra conectada al Backplane. Cuando este led cambia a color ámbar es porque la fuente no está recibiendo la alimentación, está en off.

- PS Fault:(color rojo)

Cuando este led se encuentra encendido indica que la fuente de poder tiene una falla y no puede suministrar el voltaje suficiente requerido por el Backplane.

- Over Temperatura (color ámbar)

Cuando este led se encuentra encendido indica que la fuente de poder está muy cerca o excediendo la máxima temperatura de operación.

- Overload (color ámbar)

Cuando este led esta encendido indica que la fuente de poder está muy cerca o excediendo el límite máximo de su salida.

### **Switch On/Off**

La fuente de poder posee un Switch ON/OFF localizado en la parte frontal. Este Switch controla la operación de alimentación hacia el Backplane. Este Switch no interrumpe la línea de alimentación hacia la fuente de poder.



## Terminales de conexión

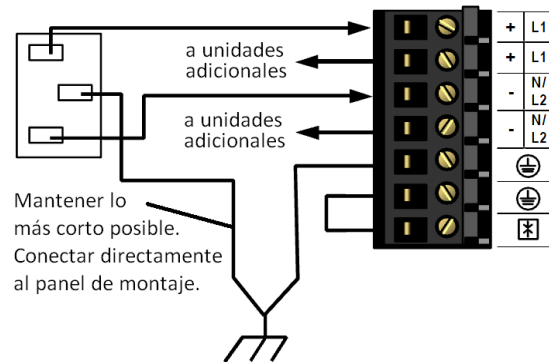


Figura2.21 Terminales de Conexión [1]

Los terminales de la fuente tanto para la alimentación, puesta a tierra modo de desconexión aceptan cables número 14 al 22 AWG individuales.

### Advertencia

Si la misma fuente de alimentación de CA externa se utiliza para proporcionar energía a dos o más fuentes de alimentación RX3i en el sistema, la polaridad de conexión debe ser idéntico al de cada fuente de alimentación. La resultante en el diferencial de potencial puede herir al personal o causar daños al equipo. Además, cada placa base debe estar conectada a una tierra común del sistema.

## Protección contra sobretensiones de entrada

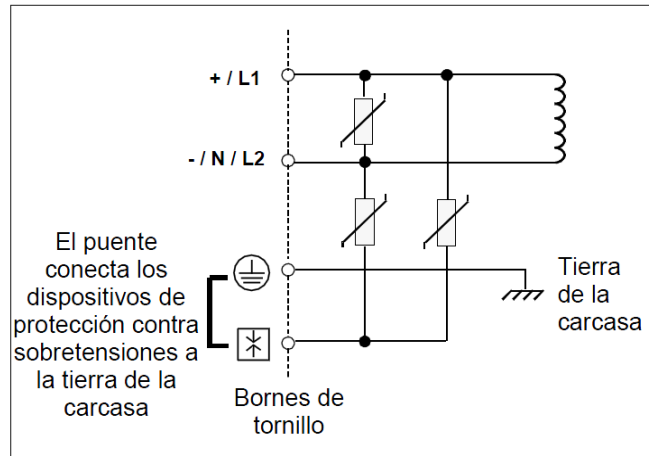


Figura 2.22 Protección contra sobre tensiones [1]

El borne inferior se conecta normalmente a la tierra de la carcasa mediante un puente instalado por el usuario, como se muestra en la parte inferior derecha de la figura. Si no se requiere protección contra sobretensión o si se suministra en sentido ascendente, el puente no es necesario. Para realizar la prueba de alta tensión de la alimentación, deberá desactivarse la protección contra sobretensiones durante la misma retirando el puente. Tras la prueba habilite de nuevo la protección de sobretensión volviendo a instalar el puente.

En sistemas con una entrada neutra flotante (la línea neutra no está referenciada a la tierra de protección), este puente NO deberá instalarse. Además, en un sistema neutro flotante, los dispositivos de

protección contra sobretensiones, como MOVs deben instalarse de L1 a la tierra, y de L2 (Neutro) a la tierra.

## 2.4 Especificaciones técnicas

### 2.4.1 Unidad de fuente de poder

**Tabla II. Unidad de Fuente de poder**

<b>Característica</b>	<b>Rango</b>
Tensión Nominal Rango de Voltaje de Entrada AC DC	120/240 VAC o 125 VDC  85 a 264 VAC 100 a 300 VDC
Potencia de Entrada (Máxima a Plena Carga) Corriente transitoria de conexión	70 Watts máximo 4 Amps, 250 milisegundos máximo
Potencia de Salida	40 Watts máximos en total 5.1 VDC = 30 Watts máximos 3.3 VDC = 30 Watts máximos La máxima potencia de salida depende de la temperatura a la Que se encuentre la fuente de poder.

Tensión de salida	24 VDC: 19.2 VDC a 28.8 VDC 5.1 VDC: 5.0 VDC a 5.2 VDC (5.1 VDC nominal) 3.3 VDC 3.1 VDC a 3.5 VDC (3.3 VDC nominal)
Corriente de salida	24 VDC: 0 a 1.6 Amps 5.1 VDC: 0 a 6 Amps 3.3 VDC: 0 a 9 Amps
Aislamiento (Entrada al Backplane)	250 VAC continuamente; 1500 VAC por 1 minuto
Fluctuación (en Todas las Entradas)	150 mV
Ruido (en Todas las Entradas)	150 mV
Margen de tiempo tras caída de tensión	Este es el tiempo que la fuente de alimentación mantiene las salidas válidas si la fuente de alimentación se interrumpe.
Cableado de los terminales	Cada uno de los terminales aceptan cables número 14 AWG al 22 AWG
Corriente por terminal	6 Amps
Número de fuentes PSA040 en cadena tipo margarita	Hasta 4

## 2.4.2 Unidad central de procesamiento

Tabla III. Estado de leds del CPU

LED	ESTADO DEL LED	ESTADO DE OPERACIÓN DEL CPU
CPU OK	Verde ON	El CPU ha pasado su diagnóstico de arranque y está funcionando correctamente
	OFF	Indica problemas en el CPU. Los Leds RUN y OUTPUTS ENABLED pueden estar intermitentes en un patrón de código de error. Esta condición de error debe ser consultada con un representante de soporte técnico
CPU OK, OUTPUTS ENABLED, RUN	Verde, parpadeando al mismo tiempo	El CPU está en modo de arranque y está a la espera de una actualización del firmware a través del puerto serie.

RUN	Verde ON	El CPU está en modo de marcha
	OFF	El CPU está en modo de Stop
OUTPUTS ENABLED	Verde ON	El escaneo de las Salidas esta activado
	OFF	El escaneo de las Salidas esta desactivado
I/O FORCE	Ámbar	Indica que se existen referencias forzadas en el programa
BATTERY	Rojo, parpadeando	Indica que la batería con poca carga
	Rojo, ON	Indica la ausencia de la batería o esta no tiene carga
SYSTEM FAULT	ON	El CPU esta en modo Stop/Falla o en Stop/Detenido
COM1 COM2	Verde, parpadeando	Indica señal activa en el puerto

**Tabla IV. Especificaciones Técnicas del CPU**

Batería: retención de memoria	Para estimar la duración de la batería consultar el manual de referencia GFK-2222
Almacenamiento de programa	10 Mbyte RAM respaldados por batería, 10 Mbyte de memoria flash no volátil.
Requerimientos de potencia	+3.3 VDC: 1.25 Amps nominal, +5 VDC: 1.0 Amps nominal.
Temperatura de Operación	0 a 60 °C (32°F a 140°F)
Punto flotante	Si
Velocidad de ejecución booleana, típica	0.195ms por cada 1000 instrucción booleana.
Tiempo de precisión del reloj en un día	Desplazamiento máximo de 2 segundos por día.
Precisión de tiempo del reloj (tiempo interno)	0.01% máximos.
Comunicaciones integradas	RS-232, RS-485
Protocolos seriales soportados	Modbus RTU esclavo, SNP, Serial I/O
Backplane	Soporta bus dual Backplane: RX3i PCI y serie 90-30 serial
Compatibilidad PCI	Sistema diseñado para ser eléctricamente compatible con PCI 2.2 estándar.

Bloques del programa	Hasta 512 bloques de programa. El tamaño máximo para un bloque es de 128KB.
Memoria	%I y %Q: 32Kbits para discretas %AI y %AQ: configurables hasta 32Kwords %W: configurable hasta el máximo de la memoria RAM disponible.

### 2.4.3 PACMotion



Figura2.23 Leds Indicadores del PACMotion

El PMM tiene ocho Leds que proporcionan indicaciones de estado. Cuatro de los Leds indican el estado de los ejes por individual. Dos Leds indican el estado de funcionamiento del módulo. Los otros dos Leds indican el estado de la FSSB y enlaces de comunicaciones FTB.



**Tabla V.Leds indicadores de estado de operación de los ejes de PMM**

<b>LED</b>	<b>ESTADO DEL LED</b>	<b>ESTADO DE OPERACIÓN DEL PMM335</b>
1 (Eje 1) 2 (Eje 2) 3 (Eje 3) 4 (Eje 4)	Verde ON	EL drive correspondiente al LED está habilitado
	Rojo ON	Un paro normal ha ocurrido en este eje por causa de un error
	Rojo, parpadeando a intervalos de 500ms	Una parada rápida ha ocurrido por causa de un error

**Tabla VI.Leds indicadores de estado de operación del de PMM**

<b>LED</b>	<b>ESTADO DEL LED</b>	<b>ESTADO DE OPERACIÓN DEL PMM335</b>
TODOS	OFF	El PMM no está alimentado
	Verde ON	El PMM335 está funcionando correctamente y no

STATUS		hay errores en el módulo
	Verde, parpadeando a intervalos de 1 segundo	Indica que ha ocurrido una advertencia o un error que no requiere de un paro
	Verde, parpadeando a intervalos de 500ms	Indica que ha ocurrido un error requiriendo un rápido o normal paro
	Verde, parpadeando intermitente; CONFIG y Leds OFF (de los ejes)	El patrón de parpadeo indica que un error fatal ha ocurrido en el módulo. Anote el número de parpadeos en la secuencia y el póngase en contacto con el soporte técnico para obtener información adicional.
	Verde, parpadeando simultáneamente CONFIG con Leds (de	El módulo se encuentra en la actualización del

	los ejes)	Firmware.
	Ámbar ON	Se ha producido un grave error en el hardware del módulo de out en el tiempo de espera.
	Verde ON	El PMM ha recibido una configuración válida desde el CPU RX3i.
CONFIG	Verde, parpadeando	El PMM aún no ha recibido una configuración válida del programador.
	Ámbar ON	El PMM se encuentra en modo de arranque.
	Ámbar, parpadeando	El PMM recibió una configuración inválida del programador.
	OFF	La comunicación FSSB está deshabilitada.
FSSB	Verde ON Verde ON	La comunicación FSSB está habilitada.

FSSB	Verde, parpadeando	La configuración FSSB está en progreso o los servos amplificadores todavía no están disponibles. Los ejes se pueden utilizar con motor sintético. Esta es una operación normal.
	Rojo ON Rojo ON	La comunicación FSSB ha fallado
	OFF	La comunicación Fiber I/O está inactivado
FTB	Verde ON	La comunicación Fiber I/O esta activado
	Verde parpadeando Verde parpadeando	La configuración Fiber I/O está en progreso
	Rojo ON	La comunicación Fiber I/O ha fallado
		Indica un error de

	OFF	ID del FTB. El PMM está tratando de comunicarse con el FTB que no tiene la correcta configuración PMM/FTB ID de comunicación.
	OFF	

#### 2.4.4 Fiber Terminal



Figura 2.24 Leds Indicadores del FTB

#### Leds indicadores

El FTB tiene un LED para cada uno de los 24 puntos I/O, lo que indica el estado del punto de I/O para permitir la depuración visual. Los cuatro LED adicionales indican el estado de funcionamiento de la FTB.

**Tabla VII.Leds indicadores de estado de operación de las I/O del FTB**

<b>LED</b>	<b>ESTADO DEL LED</b>	<b>ESTADO DE OPERACIÓN DEL FTB</b>
Entradas desde la 1 a la 16	Verde	La entrada esta ON
	Off	La entrada esta OFF
Salidas desde la 1 a la 8	Verde	La salida esta ON
	Off	La salida esta OFF
	Red	La salida tiene un estado de falla

**Tabla VIII.Leds indicadores de estado de operación del FTB**

<b>LED</b>	<b>ESTADO DEL LED</b>	<b>ESTADO DE OPERACIÓN DEL FTB</b>
STATUS	Off	No se aplicado alimentación al FTB.
	Verde	Correcto funcionamiento.
	Ámbar	La configuración no ha sido recibida aún.
FIBER	Off	No se aplicado alimentación al FTB.
	Verde	Enlace E/S establecido.
	Verde, parpadeando	Se está transfiriendo la configuración de enlace E/S.
	Verde y Rojo parpadeando alternativamente	Indica un error de ID en el FTB. El PMM está tratando de comunicarse con un FTB que no tiene el correcto enlaces de comunicaciones PMM/FTB.
	Rojo	Enlace de comunicación E/S no establecido.
	Off	La configuración de

T1 PWR		detección de poder en las salidas 1 – 4 del FTB esta desactivada.
	Verde	La configuración de detección de poder en las salidas 1 – 4 del FTB esta activada, y 24V de alimentación son aplicados al terminal 1 (OUT POWER) y al terminal 19 (OUT COMMON) del bloque de terminales 1.
	Rojo	La configuración de detección de poder en las salidas 1 – 4 del FTB esta activada y no se ha aplicado alimentación a los terminales 1 y 19 del bloque de terminales 1.
T2 PWR	OFF	La configuración de detección de poder en las salidas 5 – 8 del FTB esta desactivada.
		La configuración de



T2 PWR	Verde	detección de poder en las salidas 5 – 8 del FTB esta activada y 24V de alimentación son aplicados a los terminales 1 y 19 del bloque de terminales 2.
	Rojo	La configuración de detección de poder en las salidas 5 – 8 del FTB esta activada y no se aplicado alimentación a los terminales 1 y 19 del bloque de terminales 2.

**Tabla IX. Especificaciones técnicas del FTB**

<b>Especificaciones detalladas</b>		<b>Comentarios</b>
Dimensiones de montaje	35 mm Riel DIN	Debe montarse sobre una superficie horizontal para su correcta refrigeración.
Pulgadas	5.56 W x 4.94 H x 2.46 D	
mm	141.2 W x 125.5 H x 62.5	
Interface con el módulo PACMotion	Cable de Fibra Óptica	100 metros máximos de longitud de cable de fibra óptica; la interfaz utiliza un ID único seleccionable por el usuario para cada par de PMM/FTB para prevenir la conexión.
Interface con el módulo PACMotion	Cable de Fibra Óptica	
Requerimientos de potencia	19.2VDC – 28.8VDC; 0.45 Amps a 24V	Un cable AWG #14 (2.1mm <sup>2</sup> ) o dos cables AWG #16 (1.3mm <sup>2</sup> ).

Salidas de 24V	8 ópticamente aisladas, fuente, cargas abiertas & detección de corto	2 grupos de 4, 1.5 Amps máximo por punto. 4 Amps máximos por grupo.
Entradas de propósito general de 24V	16 ópticamente aisladas. fuente/sumidero	4 grupos de 4.
Entradas/Salidas (diferencial) 5V	4	RS422 Driver de línea con protección de corto circuito; 48 mA máximo.
Entradas/Salidas (diferencial) 5V	4	
Entradas de 5V	8 (6 diferenciales/un terminal; 2 diferenciales)	RS422/RS485 Receptor de línea con detección de fallos.
Entradas Analógicas	2, $\pm 10V$ diferencial	14 bits de resolución
Salidas Analógicas	2, $\pm 10V$ de un solo terminal	12 bits de resolución
Salidas de 24V de Poder		Inversión de polaridad protegida con fusible reemplazable.

Salidas de 5V de Poder	0.5 Amps máximo	Protección electrónica contra sobrecarga
QuadEncoder Detección de circuito abierto	Si	
E/S Asignación de funciones	Configurable	E/S Las Funciones son asignadas durante la configuración del hardware

### 2.4.5 Servo Motor

A continuación se muestran algunas de las especificaciones básicas del servo motor utilizado.

**Tabla X. Especificaciones técnicas del servo motor**

<b>Motor</b>	<b>Torque Nominal</b>	<b>Encoder</b>	<b>Amplificador Requerido</b>	<b># Catálogo del Motor</b>
$\beta$ 0.5/6000is	0.65 Nm (5.8 lbf-in) torque continuo; 6000RPM	B64ia (64K)	$\beta$ SVM1-20i ZA06B-6130-H002	Motor Solo: ZA06B-0115-B203 Motor con Freno: ZA06B-0115-B503

## 2.4.6 Amplificador

A continuación se muestra el modelo del amplificador usado

**Tabla XI. Especificaciones técnicas del amplificador**

<b>Motor</b>	<b>Modelo del Amplificador</b>	<b># Catalogo del Amplificador</b>	<b># Catálogo de Paquete del Amplificador</b>
$\beta$ 0.5/6000is	$\beta$ SVM1-20i	ZA06B-6130-H002	IC800BIK020

A conveniencia del usuario y del proceso se debe elegir el correcto sistema de serie servo  $\beta$ i.

El Amplificador  $\beta$ i-Series contiene lo siguiente:

- Un  $\beta$ SVM1-20i 20A amplificador (ZA06B-6130-H002)
- Un Fusible de repuesto de 24 VDC (ZA06B-6073-K250)
- Una Resistencia de Descargar de 20 Watts (ZA06B-6130-H401)
- Un Kit Conector de Poder CZ7 (ZA06B-6130-K200)
- Un Kit Conector CXA19 de 24 VDC (ZA06B-6130-K201)
- Un Kit de Conector de Termostato de Descarga CXA20
- Un Kit de Conector CX29 MCC (ZA06B-6130-K203)
- Un Kit de Conector E-stop CX30 (ZA06B-6130-K204)

## 2.5 Estructura física del sistema PACMotion

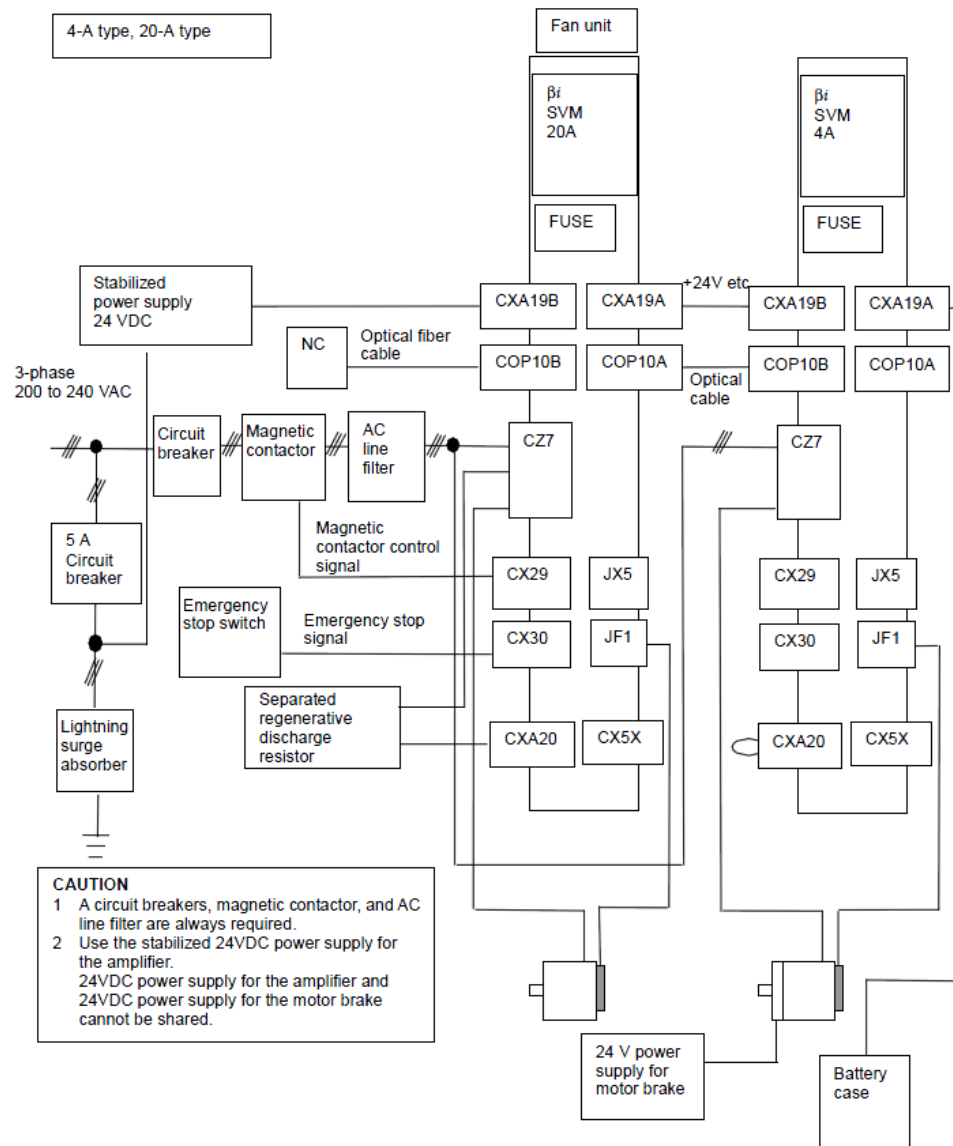


Figura 2.25 Estructura física del sistema PACMotion [1]

A continuación se muestra el sistema conectado utilizando una PC, CPU, la fuente de poder, un PMM, dos servomotores con sus respectivos Amplificadores y cable de fibra óptica (FSSB), y un Fiber I/O Terminal Block.

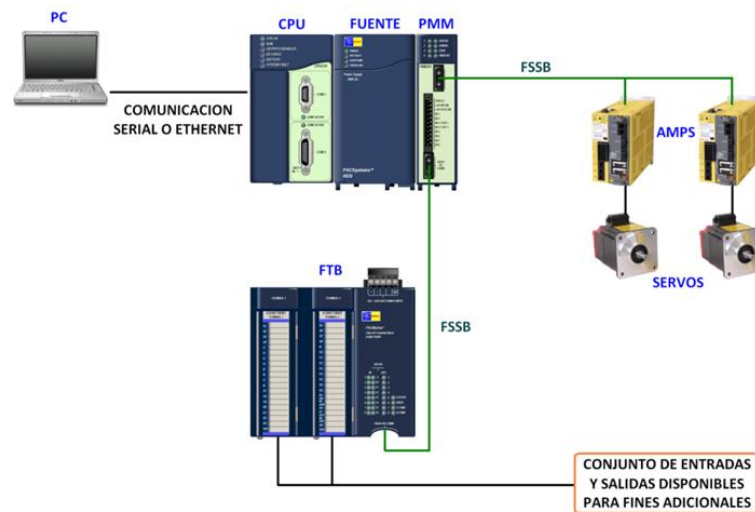


Figura 2.26 Diagrama de bloques del sistema PACMotion [1]



## 2.6 Práctica No 1.

Descripción general del sistema (PacMotion).Características básicas de sus componentes. (Primera parte)

### **Objetivos:**

- Realizar una descripción general del sistema PACMotion.
- Conocer las características principales de cada uno de los componentes del sistema PACMotion.

### **Enunciado de la práctica:**

En esta práctica se introducirá al estudiante al sistema de control de servomotores PACMotion de General Electric los cuales se encuentran instalados en el laboratorio de Automatización Industrial II y así más adelante realizar una variedad de prácticas sobre el mismo.

La ejecución de esta práctica pretende desarrollar las habilidades para la identificación de cada uno de los equipos que vamos a utilizar, de esta manera el estudiante estará preparado para identificar cada uno de los módulos utilizados con su respectiva descripción además de la integración e instalación en un solo sistema.

## **2.7 Práctica No 2.**

Descripción general del sistema (PacMotion).Características básicas de sus componentes. (Segunda parte)

### **Objetivos:**

- Realizar una descripción general del sistema PACMotion.
- Dar a conocer las características principales de cada uno de los componentes del sistema PACMotion.

### **Enunciado de la práctica:**

En esta práctica continuaremos con las especificaciones técnicas que tenemos que conocer al configurar el sistema de control de servomotores PACMotion de General Electric.

# CAPITULO 3

## CONFIGURACIÓN Y PROGRAMACIÓN DEL SISTEMA RX3i

### 3.1 Modos de comunicación.

El reconocimiento de Ethernet TCP/IP, tanto en las organizaciones como en Internet, lo ha convertido en el estándar de la comunicación actual. Su extendido uso supone una reducción de los gastos de conexión, un mayor rendimiento y la incorporación de nuevas funciones, elementos que, combinados, garantizan su durabilidad.

Gracias a su elevada velocidad, la red ya no limita el rendimiento de la aplicación además que la arquitectura puede evolucionar sin ninguna dificultad.

Los productos o dispositivos siguen siendo compatibles, lo que garantiza la durabilidad a largo plazo del sistema.

Ethernet es una red no determinista de tipo bus CSMA/CD, normalizada e independiente de todo constructor. Una red fácil de poner en marcha, coste reducido, ampliable, abierto, que permite la

conexión a todos los niveles de una empresa desde las oficinas hasta el nivel de E/S distribuidas y variadores de velocidad.

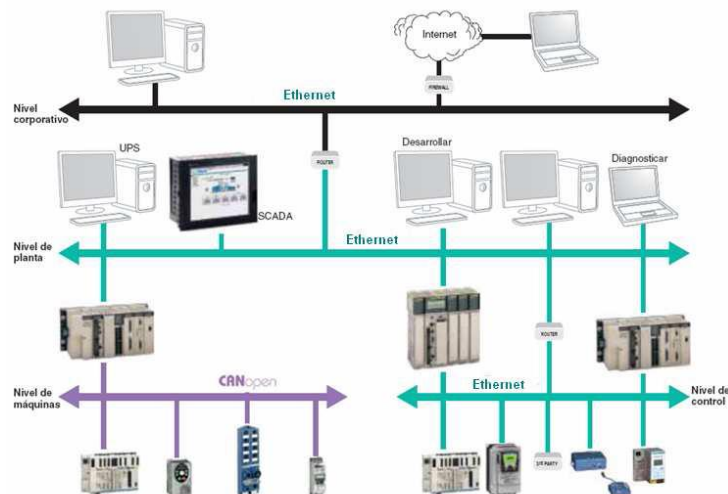


Figura3.1 Diagrama de redes de comunicación

Toda la programación de PMM se realiza a través del software de programación.

El entorno de programación RX3i tiene dos opciones de comunicación. Se puede conectar el programador directamente a uno de los puertos COM de la CPU, o bien puede comunicarse vía red Ethernet, para esto necesita una dirección IP la cual se debe asegurar en el RX3i antes que la conexión Ethernet pueda ser establecida.

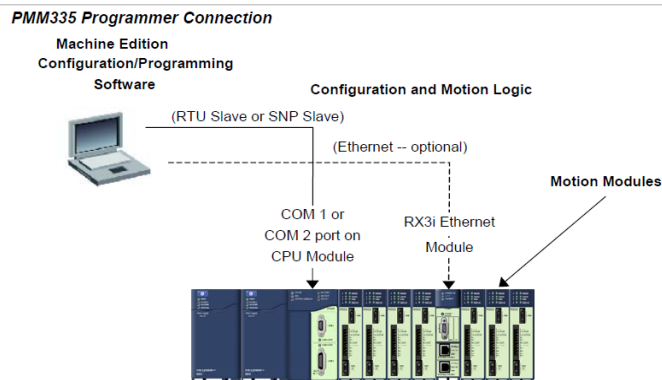


Figura 3.2 Puertos de comunicación del PACSystems [1]

La CPU tiene dos puertos seriales independientes, a los que se puede acceder por la parte frontal del módulo.

Los puertos 1 y 2 ofrecen interfaces en serie a dispositivos externos. Cualquiera de los puertos se puede utilizar para actualizaciones de firmware, el protocolo usado es el RTU.

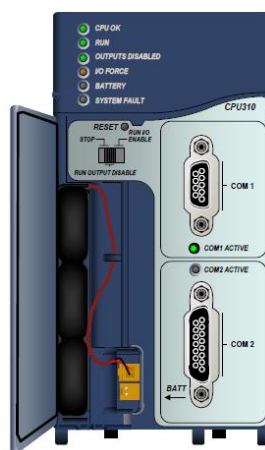


Figura 3.3 Puertos COM1 y COM2

### **3.2 Comunicaciones TCP/IP Ethernet para PACSystems.**

Existen soluciones de E/S, Control y SCADA gracias a la red homogénea que permite operar en zonas de alto ruido, cubrir grandes distancias en tiempo real y el manejo de un gran volumen de transferencias de datos a alta velocidad.

### **3.3 Estación de manejo.**

La estación de manejo es una función integrada la cual se usa para controlar la interfaz de Ethernet y comprobar su funcionamiento en la red, la estación de manejo dispone lo siguiente:

#### **3.3.1 Comandos para modificar parámetros.**

Son los comandos que realizan funciones de modificación de la interfaz de Ethernet, el acceso para los comandos de modificación está protegido por contraseña, la protección por contraseña ayuda a prevenir el uso incorrecto o involuntario de los comandos.

En la sesión de modificar, si no se ejecutan los comandos dentro de un período de tiempo de espera configurable la sesión expira y tendrá que volver a iniciar sesión, de forma predeterminada, el tiempo de espera es de 10 minutos. Se puede cambiar según sea necesario utilizar para eso usamos el comando CHLTIME.

Entre los parámetros que podemos manipular tenemos:

Comandos para el modo de observar o modificar, comandos para leer la pantalla, cambiar el Formato de pantalla, comprobación de direcciones IP, comprobación de la dirección IP de la interfaz de Ethernet, verificar que la dirección IP de la interfaz Ethernet sea única, permiten cambiar los parámetros de copia de seguridad de la interfaz de Ethernet, verificación de Comunicaciones de la Red, etc.

### **3.3.2 Comandos de monitoreo.**

Están disponibles para cualquiera que utilice la Estación. Estos comandos proporcionar información acerca de la interfaz de Ethernet y la red. Ejecutar el Monitor de comandos no afecta el funcionamiento de la interfaz de Ethernet o la red, entre las funciones que pueden realizar estos comandos tenemos:

Visualización de información acerca de un nodo, visualización del registro de Excepción, etc.

La estación de manejo funciona en segundo plano cuando la interfaz de Ethernet se encuentra funcionando, no se puede acceder a ella al momento de arranque de diagnósticos o cuando se está usando el software de programación de escalera.

### 3.3.3 Comandos de la estación de manejo.

Gracias a estos comandos vamos a tener un control total ya sea para modificar o monitorear el comportamiento de los datos a través de la red Ethernet.

**Tabla XII. Comandos de la estación de manejo.**

<b>COMANDO</b>	<b>FUNCIÓN</b>	<b>MODOS DISPONIBLES</b>
bootp	Asigna temporalmente los parámetros de TCP/IP.	Modifica y monitorización
Cd	Cambia de directorio de trabajo	Modifica y monitorización
Channel	Muestra la información de un canal de comunicación individual	Modifica y monitorización
chltime	Cambia el tiempo de espera de inactividad de un usuario	Modificar
Chparm	Cambia los parámetros de la copia de seguridad	Modificar
Chsosw	Cambia los parámetros de la configuración de la copia de seguridad de	Modificar



	Ethernet	
Chtime	Cambia el reloj interno de Ethernet	Modificar
Clear	Limpia los ítem seleccionados	Modificar
Del	Borra archivos del sistema que se encuentran en el directorio de trabajo	Modificar
Dir	Muestra el contenido de los archivo del directorio	Modifica y monitorización
Egdcmd	Envía un comando EGD a un nodo remoto	Modificar
Egdread	Muestra el intercambio de datos a través de Ethernet	Modifica y monitorización
Egdwrite	Modifica el intercambio de datos a través de Ethernet	modifica
Exs	Muestra es estado para comnReqs del PLCLogic	Modifica y monitorización
Formats	Muestra formatos de referencia en el servidor	Modifica y monitorización
?, help	Muestra la ayuda del station manager	Modifica y monitorización

s	Elimina una conexión modbus/TCP	Modificar
Killss	Elimina una conexión SRTP server	Modificar
Log	Muestra los actuales registros de usuario	Modifica y monitorización
Login	Inicio de sesión a un nivel de modificación	Modifica y monitorización
Logout	Salida de sesión a un nivel de modificación	Modificar
ltime	Muestra el tiempo de inactividad de la sesión	Modificar y monitorización
mkdir	Crea un nuevo archivo de sistema en el directorio	Modificar
Net	Forza a la red a un estado de offline/online	Modificar
Node	Muestra una identificación básica de un nodo	Modificar y monitorización
Ok	Se puede restablecer STAT (o registro vacío) LED (registro no se borra)	Modificar
Parm	Muestra los parámetros avanzados	Modificar y monitorización

Ping	Envía un eco de respuesta tipo ICMP	modificar
Plcread	Muestra la memoria del CPU	Modificar y monitorización
Plcwrite	Modifica la memoria del CPU	Modificar
Pluginapp	Maneja plugins opcionales para aplicaciones de Ethernet	Modificar
Rem	Envía comandos a un nodo remoto	Modificar
Rename	Renombra un archivo del directorio de trabajo	Modificar
repp	Muestra los últimos resultados del comando ping	Modificar
Restart	Reinicia el firmware de Ethernet	Modificar
Rmdir	Remueve archivos del directorio de sistema	Modificar
Sosw	Muestra la configuración de Ethernet	Modificar y monitorización
Stat	Muestra varios estado de operación	Modificar y monitorización
stopp	Detiene el proceso de	Modificar

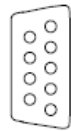
	ping	
Tally	Muestra varios recuento de mostradores de operación	Modificar y monitorización
Time	Muestra el reloj interno de Ethernet	Modificar y monitorización
Trace	Muestra la actividad de depuración	Modificar
Xchange	Muestra el intercambio de información individual EGD	Modificar y monitorización

### 3.3.4 Conexión local.

Para la operación local, debemos conectar el ordenador al puerto RS-232 de la estación de manejo de la interfaz de Ethernet del PACSystems utilizando un estándar de un puerto serial través de nueve pines tipo RS-232.

### 3.3.5 Asignación de los pines.

La asignación de los pines en el puerto RS-232 se muestra a continuación.



<i>Pin Number</i>	<i>Signal</i>	<i>Direction</i>	<i>Description</i>
1*	DCD	IN	Data Carrier Detect
2	TX	OUT	Transmit Data
3	RX	IN	Receive Data
4	DSR	IN	Data Set Ready
5	GND		Signal Ground
6	DTR	OUT	Data Terminal Ready
7	CTS	IN	Clear to Send
8	RTS	OUT	Ready to Send
9	RI	IN	Ring Indicator

\* Pin 1 is located at the bottom right of the serial port connector as viewed from the front of the module.

Figura 3.4 Asignación De Pines

## 3.4 Software De Programación “Proficy Machine Edition”.

### 3.4.1 Manejo básico del programa.

1. Iniciamos el Software Proficy Machine Edition, versión 7.0.

Una vez iniciado el programa tenemos dos opciones:

- Create a new Project
- Open an existing Project

Elegimos Create a new Project y seleccionamos “Machine Edition template”. Esta opción, a diferencia de Empty Project, nos permite

en un solo paso crear un nuevo proyecto y agregar el módulo que vamos a usar.

A continuación hacemos clic en OK.

2. En la siguiente ventana de dialogo colocamos el nombre del nuevo proyecto, escribimos el nombre del proyecto que deseamos y seleccionamos el modulo a usar, en este caso vamos a escoger siempre “GE Intelligent Platforms PACSystems RX3i”. Luego damos OK.

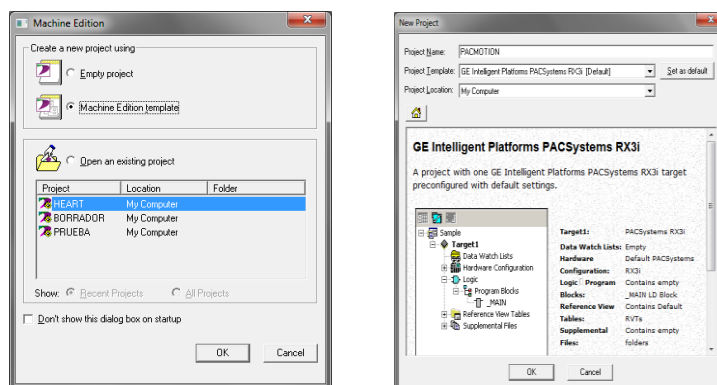


Figura 3.5 Crear proyecto

### 3.4.2 Configuración de los módulos del PACSYSTEMS.

1. En el siguiente paso agregamos los módulos que vamos a utilizar para realizar nuestra programación, para esto hacemos uso de la ventana llamada Navigator, en caso de no aparecer al inicio del programa, la podemos obtener haciendo clic en “Tools” o también presionando las teclas Shift+F4.

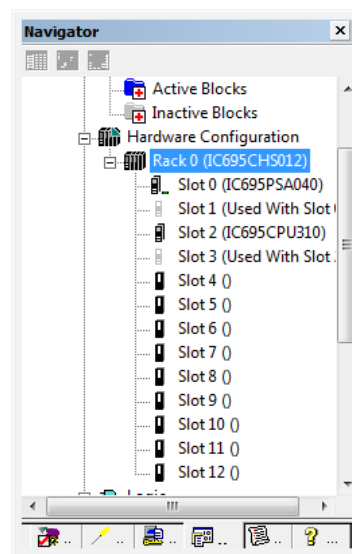


Figura 3.6 Navigator

2. Dentro de la ventana llamada Navigator, podemos observar un explorador, hacemos clic en “Hardware Configuration” y abrimos la

pestaña "Rack 0" (IC695CHS012) que es el Universal Backplane de 12 Slots que estamos utilizando.

3. Dentro del Rack 0 (IC695CHS012) podemos observar que por defecto se agrega la fuente de alimentación en el slot 0, y el CPU en el slot 2, los mismos que ocupan el slot que le sigue debido a su tamaño. Por razones de entrenamiento vamos a quitar la fuente de alimentación con la tecla "Supr".
4. Ahora vamos a colocar los diferentes módulos a usar, para esto primero vamos a colocar el CPU, luego La fuente de alimentación y por último el modulo que se encarga los servos.

Con el puntero del mouse arrastramos el CPU desde el slot 2 hasta el slot 0. Hacemos doble clic en la pestaña de nombre slot 2, e inmediatamente aparecerá la ventana de catálogo, en la pestaña llamada "Power Supplies" vamos a tomar la fuente de alimentación con la siguiente descripción, UNIVERSAL 120/240VAC, 125VDC 40W Power Supply.

5. Por último agregamos en el slot 4 el módulo controlador IC695PMM335 quien realiza el control de los ejes, de la misma manera como lo hicimos anteriormente, hacemos doble clic sobre el slot 4, en la ventana catálogo seleccionamos la pestaña



“Motion” buscamos el módulo correspondiente, en este caso PACMotion, Digital Motion Control Module, damos OK.

6. Con esto finalizamos la instalación de los módulos en el Universal Backplane.

### 3.4.3 BETA I SERVO TRAINER.

Ahora para tener control del Beta I Servo Trainer debemos de configurar cada modulo instalado en el backplane, para esto vamos a realizar los siguientes pasos:

#### Configuración del CPU (IC695CPU310).

Para configurar el CPU hacemos clic derecho en el slot 0 donde se encuentra ubicado el modulo del CPU, a continuación en “Configure”.

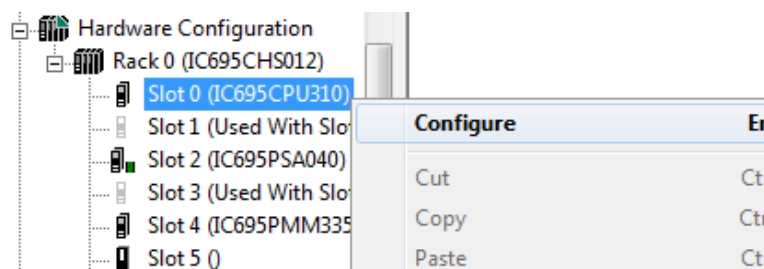


Figura 3.7 Configuración de módulos

En la siguiente ventana encontramos los parámetros del CPU, estos parámetros se encuentran por default, en nuestro caso solo modificaremos los que nos interesan.

Dentro de la pestaña Settings, en Passwords cambiamos del modo “Enabled” a Disabled.

### **Configuración del PMM (IC695PMM335).**

- De la configuración de este módulo dependerá el funcionamiento del sistema. En este modulo configuramos los parámetros del servo a usar. En la siguiente figura se pueden observar las pestañas del PMM.
- Para tener una mejor información del funcionamiento y evitar posibles errores por la presencia de “Warning” que puedan provocar nuevos errores cambiamos el parámetro “Log Messages in I/O FaultTable” a Errors\_Warnings.
- En el caso de “Axes1 mode” y demás por default se encuentra FSSB servo, esto nos indica que estamos usando fibra para controlar los servos, este es nuestro caso.
- Ya que disponemos del Fiber Terminal Block vamos a hacer uso de sus entradas y salidas habilitamos el mismo cambiando el parámetro “Fiber Terminal BlockSetting” a Enabled.
- Tenemos que tomar en cuenta que en esta configuración se encuentran habilitados los parámetros “FTB INxxFaultDetect” (donde xx va desde 17 hasta 28) el cual detecta la ausencia del

cableado en la entrada xx, mostrando un warning por cada entrada en caso de darse tal caso. Para evitar esta molestia cambiamos el parámetro “FTB INxxFaultDetect” a Disabled ya que estas entradas no están físicamente cableadas.

InfoViewer (0.4) IC695PMM335	
Settings   FP I/O   FTB Inputs   FTB Outputs   I/O Interrupts   Axis 1   Axis 2   Advanced   Power Consumption	
Parameters	Value
<i>FTB IN17</i>	Fast Digital Input
FTB IN17 Input Ref	M4_FTb_IN17
<i>FTB IN17 Mode</i>	Differential
<i>FTB IN17 Fault Detect</i>	<b>Disabled</b>
<i>FTB IN18</i>	Fast Digital Input
FTB IN18 Input Ref	M4_FTb_IN18
<i>FTB IN18 Mode</i>	Differential
<i>FTB IN18 Fault Detect</i>	<b>Disabled</b>
<i>FTB IN19</i>	Fast Digital Input

Figura 3.8 Configuración FTB

Al igual que en las entradas aquí deshabilitamos el parámetro “FTB OUT xx Open Load Detect” para evitar generar Warnings durante la ejecución del programa.

InfoViewer (0.4) IC695PMM335	
Settings   FP I/O   FTB Inputs   FTB Outputs   I/O Interrupts   Axis 1   Axis 2   Advanced   Power Consumption	
Parameters	Values
<i>FTB OUT4</i>	Digital Output
FTB OUT4 Ref	M4_FTb_OUT4
<i>FTB OUT4 Open Load Detect</i>	<b>Disabled</b>
<i>FTB OUT5</i>	Digital Output
FTB OUT5 Ref	M4_FTb_OUT5
<i>FTB OUT5 Open Load Detect</i>	<b>Disabled</b>
<i>FTB OUT6</i>	Digital Output

Figura 3.9 Parámetros Open Load Detect desactivados

La configuración para el servomotor, como son rangos de posición, velocidad, aceleración, torque, etc. Lo tenemos en la siguiente figura.

InfoViewer (0.4) IC695PMM335	
Settings   FP I/O   FTB Inputs   FTB Outputs   I/O Interrupts   Axis 1   Axis 2   Advanced   Power Consumption	
Parameters	Values
Stop Axis on FTB Error	Enabled
<i>Position Feedback Source</i>	Motor Encoder
<i>Axis Positioning Mode</i>	<b>Rotary</b>
<i>Motor Encoder Mode</i>	<b>Incremental</b>
Motor Encoder User Units	<b>360.0</b>
Motor Encoder Counts	<b>65536</b>
Motor Encoder Position Range (uu)	<b>360.0</b>
Motor Encoder Low Position Limit (uu)	-8388608.0
Motor Encoder Counts Per Motor Revolution	65536
Motor Encoder Maximum Positive RPM Limit	8191
Motor Encoder Maximum Negative RPM Limit	8191
<i>External Device</i>	None
Over Travel Limit Switch	<b>Disabled</b>
Axis Direction	Normal
<i>Software End of Travel</i>	<b>Disabled</b>
Max Velocity System (RPM)	<b>6000.0</b>
Equivalent Velocity (uu/sec)	<b>36000.0</b>
Max Acceleration System (RPM/sec)	40000.0
Equivalent Acceleration (uu/sec <sup>2</sup> )	<b>240000.0</b>
Max Deceleration System (RPM/sec)	40000.0
Equivalent Acceleration (uu/sec <sup>2</sup> )	<b>240000.0</b>
Max Deceleration System (RPM/sec)	40000.0
Equivalent Deceleration (uu/sec <sup>2</sup> )	<b>240000.0</b>
Max Jerk (uu/sec <sup>3</sup> )	<b>1000000000000.0</b>
Drive Disable Delay (ms)	100
Motor Type	<b>281</b>
Motor Velocity Limit (RPM)	4000
Torque Limit (%)	<b>95.0</b>
<i>Position Lag Monitoring</i>	<b>Enabled</b>
Max Position Lag (uu)	<b>3000.0</b>
Max Position Error (Motor Revs)	10.0
Equivalent Position Error (uu)	<b>3600.0</b>
In Position Zone (uu)	10.0
Position Loop Time Constant (ms)	100.0
Velocity Feedforward (%)	0.0
Load Inertia Ratio (256 = 1:1)	128
Error Stop Deceleration (uu/sec <sup>2</sup> )	<b>100000.0</b>
Error Stop Jerk (uu/sec <sup>3</sup> )	10000000.0
Master Axis Velocity Filter (ms)	8
Feedback Moving Deadband (uu/sec)	100.0
Command Moving Deadband (uu/sec)	0.0
Sync Master Position Deadband (Master uu)	10.0
Disabled Direction Deadband (uu)	0.0

Figura 3.10 Parámetros de configuración del servo motor

**Los parámetros a modificar son los siguientes:**

- **Axis Position Mode:** (Rotary) esto nos indica que el servo va a estar rotando indefinidamente una vez dada la orden.
- **Motor EncoderMode:** (Incremental) ya que para el modo Absolute se requiere de una batería interna en el amplificador para que la información no se pierda una vez apagado el sistema.
- **Motor EncoderUserUnits:** (360) porque deseamos manejar 360 grados como unidades de usuario. Esto es configurable de acuerdo a las unidades que el usuario decida manejar.
- **Motor EncoderCounts:** (65536) este valor define la resolución del movimiento del servo que se aplica a las unidades de usuario, este es el máximo permitido para nuestro tipo de servomotor.
- **OverTravelLimit Switch:** (Disabled) este parámetro está deshabilitado ya que en el modo Rotary no se requiere de un final de carrera.
- **Software End of Travel:** (Disabled) al igual que el parámetro anterior, éste se encuentra deshabilitado por la misma razón.
- **Max Velocity System (RPM):** 6000.0 de acuerdo a las características encontradas en el manual 6000 RPM es la velocidad máxima para éste servomotor.
- **Max Jerk (uu/sec\*\*3):** (1000000000000.0) se disminuye a este valor ya que el que se encontraba por defecto no se encuentra

dentro de los rangos permitidos para las unidades de usuario establecidas.

- **Motor Type:** (281) este es el código asignado al servomotor  $\beta 0.5/6000iS$ .
- **Torque Limit (%):** (95.0) para evitar que el servomotor trabaje al límite disminuimos este valor va de 100% a 95%.
- **Position Lag Monitoring:** (Enable) este parámetro nos permite monitorear continuamente la posición del servomotor y comparar la posición actual con la posición ordenada.
- **Max Position Lag (uu):** (3000.0) este valor especifica el máximo retraso entre la posición actual y la posición comandada. Si se sobrepasa este valor se enviara un error y se detendrá el servomotor.
- **Error Stop Deceleration (uu/sec \*\*2):** (100000.0) al igual que en Max Jerk, este parámetro se disminuyó del que estaba por defecto ya que debido al cambio en las unidades de usuario esto cambia los cálculos.

De la misma manera configuramos Axis 2.

Las pestañas FPI/O, I/O INTERRUPTS, ADVANCED, POWER CONSUMPTION no son tomadas en cuenta por ahora.

## 3.5 Introducción a los bloques de movimientos.

### 3.5.1 Variables de uso general

En esta sección se describen algunas de las funciones que son comunes en la mayor parte de los bloques.

#### **Variables de entrada.**

**Enable:** Determina la acción o el comportamiento especificado dentro del bloque de función. La entrada para estos bloques debe estar en ejecución continua.

**Execute:** Esta entrada cumple la función de ejecutar la orden del bloque, se activa solamente con un flanco de subida en la entrada.

**Jerk y JerkUnits:** Es la tasa de cambio de la aceleración con respecto al tiempo, por lo general su unidad está dada en  $Uu/sec^3$  (Unidades de usuario por segundos al cubo), aunque también se lo puede expresar en porcentaje si se lo desea, si la entrada Jerk es "Cero" entonces se considera una tasa de cambio ilimitada y una aceleración limitada.

El parámetro "JerkUnits" especifica las unidades que van a ser utilizadas por el parámetro "Jerk", estas pueden ser "UserUnitsPerSecondCubed" o "PercentJerkLimiting".

El parámetro "UserUnitsperSecondCubed" utiliza  $Uu/sec^3$  como unidad y especifica el Jerk limitante como la máxima tasa de cambio de la aceleración.

El parámetro “PercentJerkLimiting” utiliza un porcentaje para limitar el Jerk, va desde 0% hasta 100%. Su valor es especificado en una variable de tipo LREAL en la cual se indica el porcentaje del Jerk al que la aceleración o desaceleración llegara partiendo de cero para al final lograr la velocidad deseada.

Usando el parámetro “PercentJerkLimiting” con un valor de 0% se tendrá una aceleración constante o de forma lineal, mientras que con un valor de 100% se tendrá una curva S de aceleración a full.

**BufferMode:** Parámetro usado para determinar el comportamiento de los bloque de función de Movimiento.

#### **Los modos de operación son los siguientes:**

**Aborting:** Este comando ignora cualquier movimiento en curso y actúa de inmediato.

**Buffered:** Este comando comienza el movimiento tan pronto como en el bloque actual de movimiento se active la salida Done, InPosition o InVelocity.

**BlendingLow:** El segundo comando de bloque de movimiento empieza el movimiento tan pronto cuando el primer comando de bloque de movimiento haya finalizado, las velocidades de ambos bloques se juntan en una velocidad baja para ambos.



**Blending Previous:**

El segundo comando de bloque de movimiento empieza tan pronto el primer comando de bloque de movimiento haya finalizado. La transmisión de la velocidad usa la posición final del primer bloque de movimiento.

**BlendingNext:**

El segundo comando de bloque de movimiento empieza el movimiento tan pronto cuando el primer comando de bloque de movimiento haya finalizado, a diferencia del anterior la transmisión de la velocidad utiliza la velocidad inicial del segundo bloque y la final del primer bloque.

**BlendingHight:**

El segundo comando de bloque de movimiento empieza el movimiento tan pronto cuando el primer comando de bloque de movimiento haya finalizado y la transmisión de la velocidad utiliza las velocidades altas entre ambos bloques y la posición final del primer bloque.

### **Parámetros de salida comunes**

**Axis:** Acepta tipo de dato AXIS\_REF, sirve para indicar el eje sobre el cual está actuando el bloque.

**Done:** Dato tipo "PowerFlow", esta salida se activa cuando el bloque ha alcanzado su objetivo principal.

**Busy:** Dato tipo "PowerFlow", indica si el bloque está habilitado y actualmente ejecutándose.

**Valid:** Dato tipo "PowerFlow", se activa si las salidas están disponibles.

**Active:** Dato tipo "PowerFlow", se activa cuando el bloque ha tomado dominio del eje.

**ComandAborted:** Dato tipo "PowerFlow", se activa cuando el comando fue abortado por otro comando.

**Warning:** Dato tipo "PowerFlow", que se activa cuando se presenta alguna advertencia con respecto al bloque.

**Error:** Dato tipo "PowerFlow", que se activa cuando ocurre un error en la ejecución del bloque.

**ErrorID:** Variable tipo de dato Word, que nos dará a conocer la identificación de un error cuando este se presente.

### 3.5.2 Bloques de funciones

Las instrucciones de movimiento consisten en un estándar de bloques de funciones los cuales se describen a continuación.

**Tabla XIII. Funciones de los bloques de programación**

<b>TITULO</b>	<b>ESPECIFICACIÓN</b>	<b>BLOQUE</b>
RESET	Limpia los errores de los servos y los módulos.	MC_Reset. Modulo Reset.
AXISPOWER UPAND INITIALIZATION	Aplica potencia a los servos. Estabiliza o recalibra la posición de los servos. Ejecuta una búsqueda para estabilizar los servos a una posición valida actual.	MC_Power.  MC_SetPosition.  MC_Home.
TOUCH PROBE	Captura la posición actual del eje.	MC_TouchProbe.
DISCRETE MOTION	Establece una posición actual del eje. Mueve el eje a una posición específica. Adiciona un movimiento al eje además del ya programado por la instrucción principal.	MC_Home.  MC_MoveAbsolute .  MC_MoveAdditive.

DISCRETE MOTION	Mueve una distancia específica relativa a la actual posición en el momento de la ejecución del programa.	MC_MoveRelative.
CONTINUOUS MOTION	Jog hacia adelante o atrás. Mueve los ejes a una velocidad específica. Mueve el eje una distancia específica en relación a la actual posición en el momento de la ejecución del programa.	MC_JogAxis. MC_MoveVelocity. MC_Additive.
SETTING OVERRIDE PARAMETERS	Configura velocidad, aceleración, desaceleración y torque para todas las funciones que comandan el movimiento sobre los ejes.	MC_SetOverride.
CONTROLLED STOPS	Detiene los ejes Bloquea los ejes	MC_Stop MC_Halt.
AXIS SYNC	Empiezan todos los ejes al mismo tiempo. Empiezan todos los ejes con un retardo de tiempo relativo entre cada uno.	MC_SyncStart. MC_DelayedStart.

<p>AXIS GEARING</p>	<p>Sincroniza la velocidad de radio de un eje esclavo en relación al eje maestro.</p> <p>Sincroniza la velocidad y posición de un eje esclavo en relación al eje maestro.</p> <p>Finaliza la ejecución de los bloques de función MC_GearIn y MC_GearOut.</p>	<p>MC_GearIn.</p> <p>MC_GearOut.</p> <p>MC_GearOut.</p>
<p>SUPERIMPOSED MOTION</p>	<p>Mueve una distancia específica en adición a un movimiento existente.</p>	<p>MC_MoveSuperimposed.</p>
<p>CAM PROGRAMMING</p>	<p>Carga el archivo de la CAM del CPU sobre un PMM.</p> <p>Capta el archivo de la CAM sobre un eje maestro y esclavo.</p> <p>Libera el eje de un esclavo de un archivo de la CAM.</p> <p>Borra un archivo de la CAM de un PMM.</p>	<p>MC_CamTableSelect.</p> <p>MC_CamIn.</p> <p>MC_CamOut.</p> <p>MC_CamTableDeselect.</p>
<p>PHASING</p>	<p>Controla el cambio de fase de un eje esclavo</p>	<p>MC_Phasing</p>
	<p>Lee un archivo de la</p>	<p>MC_CamFileRead.</p>

CAM FILE MANAGEMENT	CAM del RX3i guardado en la memoria de referencia del CPU. Copia un archivo de la CAM de la memoria de referencia del CPU dentro del sistema RX3i. Recupera información de los archivos de la CAM usado en un PMM.	MC_CamFileWrite.  MC_LibraryStatus.
MONITORING AXIS OPERATION	Lee el estado actual de los ejes. Nos dice el tipo de error o la advertencia en relación al eje. Lee la posición actual del eje. Lee la velocidad actual del eje. Lee el torque actual del eje.	MC_ReadStatus.  MC_ReadAxisError.  MC_ReadActualPosition.  MC_ReadActualVelocity.  MC_ReadTorqueCommand.
READING WRITING I/O REFERENCE IDS	Lee el estado de las entradas discretas del FTB. Lee el estado de la salida discretas del FTB. Escribe una salida	MC_ReadDigitalInput.  MC_ReadDigitalOutput.

	<p>discreta en el FTB.</p> <p>Lee el estado de las entradas analógicas del FTB.</p> <p>Lee el estado de las salidas analógicas del FTB.</p> <p>Escribe una salida analógica en el FTB.</p>	<p>MC_WriteDigitalOutput.</p> <p>MC_ReadAnalogInput.</p> <p>MC_ReadAnalogOutput.</p> <p>MC_WriteAnalogOutput.</p>
<p>READING</p> <p>WRITING</p> <p>AXIS AND</p> <p>MODULE</p> <p>PARAMETERS</p>	<p>Lee el valor de un parámetro booleano de la configuración del hardware o la bandera del estatus del eje.</p> <p>Lee el valor de un arreglo de parámetros booleano de la configuración del hardware o las banderas del estatus de los ejes.</p> <p>Lee el valor de un arreglo de parámetros tipo DWORD.</p> <p>Lee el valor de un parámetro tipo LREAL</p> <p>Lee el valor de un arreglo de parámetros tipo LREAL.</p>	<p>MC_ReadBoolParameter.</p> <p>MC_ReadBoolParameter.</p> <p>MC_ReadDwordParameters.</p> <p>MC_ReadParameter.</p> <p>MC_ReadParameters.</p>

	<p>Cambia el valor de un parámetro booleano de la configuración del hardware o la bandera del estatus del eje.</p> <p>Cambia el valor de un arreglo de parámetros booleano de la configuración del hardware o las banderas del estatus de los ejes.</p> <p>Cambia el valor de un arreglo de parámetros tipo DWORD.</p> <p>Cambia el valor de un parámetro tipo LREAL</p> <p>Cambia el valor de un arreglo de parámetros tipo LREAL.</p>	<p>MC_WriteBoolParameter.</p> <p>MC_WriteBoolParameters.</p> <p>MC_WriteDwordParameters.</p> <p>MC_WriteParameter.</p> <p>MC_WriteParameters.</p>
MODULE-LEVEL OPERATIONS	<p>Controla un punto base de salida sobre la posición y tiempo del eje.</p> <p>Especifica los parámetros de los datos que van a ser monitoreados por el Data Logging Windows.</p> <p>Comienza la introducción de los datos especificados por el MC_</p>	<p>MC_DigitalCamSwitch.</p> <p>MC_DL_Configure.</p> <p>MC_DL_Activate</p>



MODULE- LEVEL OPERATIONS	DL_ Configure. Escribe los datos del archivo del MC_ DL_ Configure en la memoria de referencia del CPU.	MC_DL_Get
	Borra la configuración del MC_DL_Configure.	MC_DL_Delete
	Limpia todos los errores sobre un módulo.	MC_ModuleReset
	Lee los primeros 100 eventos del PMM.	MC_ReadEventQueue

Los bloques de funciones tienen datos de ejemplos los cuales consisten en una estructura, estos ejemplos están alojados en la memoria del RX3i y son automáticamente asignados cuando la función es usada la primera vez.

Por ejemplo la función MC\_MoveAbsolute posee los correspondientes elementos de entrada y salida para los parámetros así como los datos internos para realizar la función determinada, en otras palabras las funciones están diseñadas para que nosotros solo coloquemos los parámetros de entrada.

### **3.6 Práctica No 3.**

Configuración del equipo de entrenamiento “Beta i servo” por medio del software de programación “Proficy Machine Edition”.

#### **Objetivos:**

- Familiarización con el entorno de programación.
- Conocer los distintos modos de comunicación del sistema, protocolo, puertos.
- Configuración de los parámetros del PacMotion para su correcto funcionamiento.

#### **Enunciado de la práctica:**

Esta práctica pretende que el alumno conozca los modos de comunicación del sistema además identifique los módulos y parámetros a configurar para el correcto funcionamiento del equipo.

# **CAPITULO 4**

## **INTRODUCCIÓN AL CONTROL Y PROGRAMACIÓN DEL SISTEMA PACMOTION**

### **4.1 Programación de los Bloques de funciones de movimiento.**

El sistema PACMotion cuenta con una variedad de instrucciones, una librería de bloques que ha sido diseñada para su total funcionamiento. Algunos de los bloques principales para la programación se muestran a continuación con su respectiva descripción:

### 4.1.1 Mc\_Power.

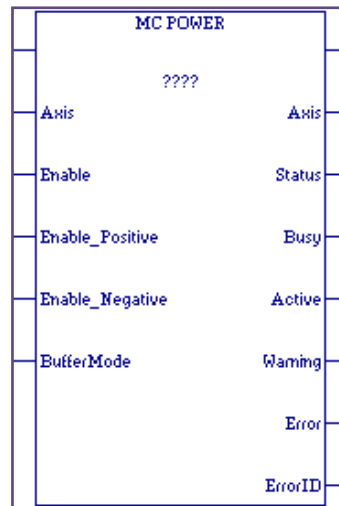


Figura 4.1 Mc\_Power

MC\_POWER es el bloque se encarga de controlar al amplificador de potencia de cada eje, se usa uno por cada servomotor.

A este bloque le asignamos una variable que servirá de referencia para almacenar los datos tanto de sus entradas como sus salidas. Además es necesario usar la entrada para habilitar el bloque, "Axis" acepta solo tipos de datos AXIS\_REF, sirve para especificar el eje que se desea energizar, "enable" acepta entradas de tipo contactos con flujo de energía constante, "PowerFlow" ordena el encendido, Enable\_Positive y Enable\_Negative, son entradas de tipo contactos "PowerFlow", son utilizados para habilitar los sentidos de movimientos del eje, si ninguna de estas dos entradas se encuentra

habilitada entonces el eje no girará, “BufferMode”, acepta tipos de datos “MC\_BufferMode” variable o constante, esta entrada sirve para definir el comportamiento que tendrá el bloque en la acción con respecto a las entradas y salidas.

En las salidas tenemos: “Axis” acepta tipo de dato AXIS\_REF, sirve para indicar el eje sobre el cual está actuando el bloque, “Status” acepta salidas del tipo bobinas o cualquier otro elemento del tipo “PowerFlow”, indica si el bloque está activado, “Busy”, dato de tipo “PowerFlow”, indica si el bloque está habilitado y actualmente ejecutándose, “Active” es un dato tipo “PowerFlow”, se activa cuando el bloque ha tomado dominio del eje, “Warning” también es un dato tipo “PowerFlow”, que se activa cuando se presenta alguna advertencia con respecto al bloque, “Error” se activa cuando ocurre un error en la ejecución del bloque y “ErrorID” en una variable de tipo de dato Word, que nos dará a conocer la identificación de un error cuando este se presente.

### 4.1.2 Mc\_Setposition

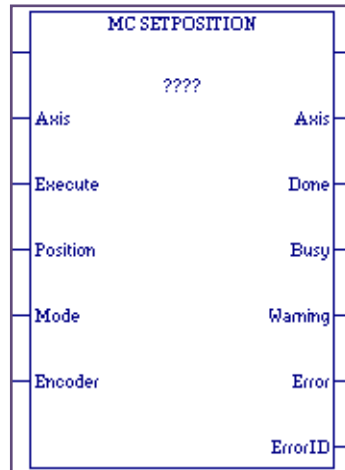


Figura 4.2 Mc\_Setposition

MC\_SETPOSITION fija una posición determinada dentro del rango preestablecido en la configuración del eje.

Este bloque requiere de una variable para guardar sus datos de entradas de y salidas. Será usado para fijar la posición que indique la entrada Position. Por ejemplo la posición “cero” del eje.

En cuanto a las entradas tenemos: Axis, acepta solo tipos de datos AXIS\_REF, sirve para especificar el eje al cual se fijara la posición, “Execute” que acepta datos tipo “PowerFlow” de un solo pulso necesariamente, esta entrada dará la orden para fijar la posición especificada en “position”, “Position” acepta datos tipo LREAL constantes o variables, aquí se especifica la posición que se desea

fijar en el eje, "Mode" acepta datos tipo "PowerFlow", esta entrada sirve para especificar el modo en que se toman los datos de la entrada "Position", si Mode es 0 se establece el Modo Absoluto el cual toma el valor de Position y lo escribe en el eje como la posición actual, si Mode es 1 se establece el Modo Relativo el cual toma el valor de position lo incrementa a la posición actual del eje y escribe el resultado como la nueva posición actual.

Entre las salidas se tiene: Axis, acepta tipo de dato AXIS\_REF, sirve para indicar el eje sobre el cual está actuando el bloque, "Done" en un dato tipo "PowerFlow", esta salida se activa cuando el bloque ya actuó correctamente sobre el eje, "Busy" es un dato tipo "PowerFlow", indica si el bloque está habilitado y actualmente ejecutándose, "Warning" es un dato tipo "PowerFlow", que se activa cuando se presenta alguna advertencia con respecto al bloque, "Error" en un dato tipo "PowerFlow", que se activa cuando ocurre un error en la ejecución del bloque; y "ErrorID" es una variable de tipo de dato Word, que nos dará a conocer la identificación de un error cuando este se presente.

### 4.1.3 Mc\_JogAxis.

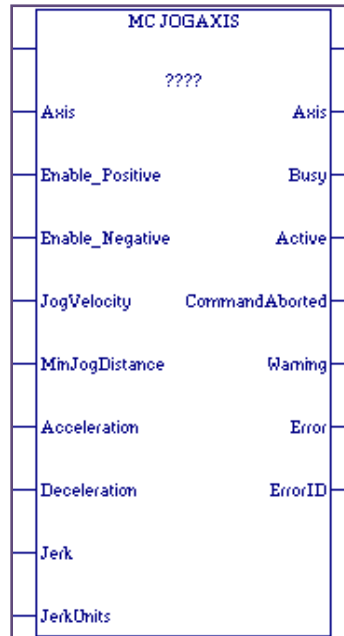


Figura 4.3 Mc\_Jogaxis

MC\_JOGAXIS da movimiento al eje definido en la entrada Axis. En este bloque se especifica la velocidad en unidades de usuario la cual puede ser constante o variable. Requiere de una variable para almacenar sus datos de entrada y salida.

En cuanto a entradas se tiene: Axis, acepta solo tipos de datos AXIS\_REF, aquí se especifica el eje que se desea controlar, "Enable\_Positive", acepta datos tipo "PowerFlow", un contacto en esta entrada sirve para habilitar la dirección positiva de movimiento del eje, "Enable\_Negative", acepta datos tipo "PowerFlow", un contacto en



esta entrada sirve para habilitar la dirección negativa de movimiento del eje, "JogVelocity", acepta datos de tipo LREAL constante o variable, aquí se especifica la velocidad deseada de movimiento para el eje, "MinJogDistance", acepta datos de tipo LREAL constante o variable, aquí se especifica un valor mínimo de distancia en unidades de usuario de recorrido permitido para cuando se desactive el sentido actual de movimiento del eje, "Acceleration", acepta datos de tipo LREAL constante o variable, sirve para establecer la aceleración a la cual se desea alcanzar la velocidad establecida, la unidad de este valor está en unidades de usuario por segundos al cuadrado y debe de estar dentro del rango preestablecido en la configuración, "Deceleration", acepta datos de tipo LREAL constante o variable, sirve para establecer la desaceleración cuando se desea descender hacia una la velocidad establecida, la unidad de este valor está en Unidades de Usuario por Segundos al cuadrado y debe de estar dentro del rango preestablecido en la configuración, "Jerk", acepta datos de tipo LREAL constante o variable, en esta entrada se especifica la tasa de cambio de la aceleración o desaceleración, si Jerk tiene valor 0 entonces se considera una tasa de cambio ilimitada, "JerkUnits", acepta datos de tipo MC\_JerkUnits constante o variable, se usa para especificar el tipo de unidades que toma Jerk, tiene dos opciones: UserUnitsPerSecondCubed y PercentJerkLimiting, la primera

establece como unidad a las unidades de usuario por segundos al cubo, la segunda representa un porcentaje de aceleración o desaceleración yendo desde 0% hasta 100% en el tiempo.

En cuanto a las salidas tenemos: Axis, acepta tipo de dato AXIS\_REF, sirve para indicar el eje sobre el cual está actuando el bloque, "Busy", dato tipo "PowerFlow", indica si el bloque está habilitado y actualmente ejecutándose, "Active", dato tipo "PowerFlow", se activa cuando el bloque ha tomado dominio del eje, "ComandAborted", dato tipo "PowerFlow", se activa cuando la entrada Enable del bloque MC\_POWER se deshabilita ó cuando otro bloque MC\_JOGAXIS ha tomado el control del eje, "Warning", dato tipo "PowerFlow", que se activa cuando se presenta alguna advertencia con respecto al bloque, "Error", dato tipo "PowerFlow", que se activa cuando ocurre un error en la ejecución del bloque y "ErrorID", variable tipo de dato Word, que nos dará a conocer la identificación de un error cuando este se presente.

#### 4.1.4 MC\_READACTUALPOSITION

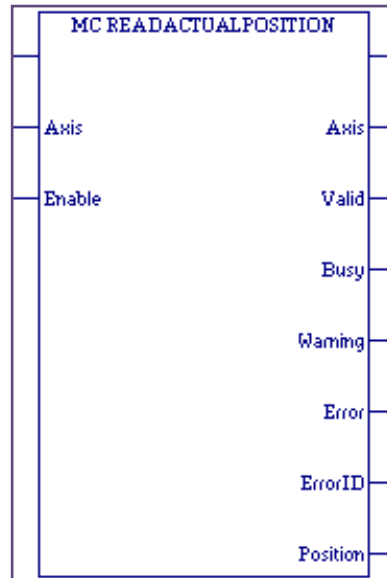


Figura4.4 Mc\_Readactualpos

EL bloque MC\_READACTUALPOSITION lee la posición actual en la que se encuentra el eje que indiquemos en la entrada. En las entradas tenemos: Axis, acepta solo tipos de datos AXIS\_REF, aquí se especifica el eje al cual se tomará lectura de la posición, "Enable", acepta datos tipo "PowerFlow", es la entrada de habilitación para que el bloque haga lectura de la posición del eje.

En cuanto a las salidas se tiene: Axis, acepta tipo de dato AXIS\_REF, sirve para indicar el eje sobre el cual está actuando el bloque, "Valid", dato tipo "PowerFlow", esta se activa si las salidas

están disponibles, “Busy”, es un dato tipo “PowerFlow”, indica si el bloque está habilitado y actualmente ejecutándose, “Warning”, dato tipo “PowerFlow”, que se activa cuando se presenta alguna advertencia con respecto al bloque, “Error”, dato tipo “PowerFlow”, que se activa cuando ocurre un error en la ejecución del bloque, “ErrorID”, variable tipo de dato Word, que nos dará a conocer la identificación de un error cuando este se presente y “Position”, escribe datos de tipo LREAL, en este parámetro se muestra la posición donde se encuentra actualmente el eje.

#### 4.1.5 MC\_MOVERELATIVE

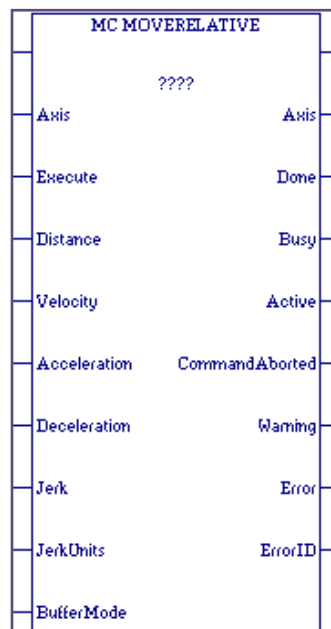


Figura 4.5 Mc\_Moverelative

El bloque MC\_MOVERELATIVE es usado para dar movimientos relativos a los ejes, es decir incrementa la posición actual del eje especificadas en "Distance", esto es cada vez que la entrada "Execute" reciba un flanco de positivo. A este bloque se le asigna una variable y una línea de habilitación.

En cuanto entradas tenemos: Axis, acepta solo tipos de datos AXIS\_REF, aquí se especifica el eje al cual se tomará lectura de la posición, "Execute", acepta datos tipo "PowerFlow" necesariamente de flanco positivo, esta entrada es la orden para ejecutar el movimiento de tipo relativo en el eje especificado, "Distance", acepta datos de tipo LREAL constante o variable, aquí se especifica la distancia que se incrementará en el eje, "Velocity", acepta datos de tipo LREAL constante o variable, aquí se especifica la velocidad a la cual se desea alcanzar la posición a incrementar, "Acceleration", acepta datos de tipo LREAL constante o variable, sirve para establecer la aceleración a la cual se desea alcanzar la velocidad establecida, la unidad de este valor está en unidades de usuario por segundos al cuadrado y debe de estar dentro del rango preestablecido en la configuración, "Deceleration", acepta datos de tipo LREAL constante o variable, sirve para establecer la desaceleración cuando se desea descender hacia una la velocidad

establecida, la unidad de este valor está en unidades de usuario por segundos al cuadrado y debe de estar dentro del rango preestablecido en la configuración, "Jerk", acepta datos de tipo LREAL constante o variable, en esta entrada se especifica la tasa de cambio de la aceleración o desaceleración, si Jerk tiene valor 0 entonces se considera una tasa de cambio ilimitada, "JerkUnits", acepta datos de tipo MC\_JerkUnits constante o variable, se usa para especificar el tipo de unidades que toma Jerk, tiene dos opciones: UserUnitsPerSecondCubed y PercentJerkLimiting la primera establece como unidad unidades de usuario por segundos al cubo y la segunda representa un porcentaje de aceleración o desaceleración desde 0% hasta 100% en el tiempo, "BufferMode", acepta tipos de datos MC\_BufferMode variable o constante, esta entrada sirve para definir el comportamiento que tendrá el bloque en la acción con respecto a las entradas y salidas.

En cuanto a las salidas tenemos: Axis, acepta tipo de dato AXIS\_REF, sirve para indicar el eje sobre el cual está actuando el bloque, "Done", dato tipo "PowerFlow", esta salida se activa cuando el eje ha alcanzado la posición establecida según la entrada "Distance" en el bloque, "Busy", dato tipo "PowerFlow", indica si el bloque está habilitado y actualmente ejecutándose, "Active", dato tipo "PowerFlow", se activa cuando el bloque ha tomado dominio del

eje, ComandAborted, dato tipo “PowerFlow”, se activa cuando el comando fue abortado por otro comando, “Warning”, dato de tipo “PowerFlow”, que se activa cuando se presenta alguna advertencia con respecto al bloque, “Error”, dato tipo “PowerFlow”, que se activa cuando ocurre un error en la ejecución del bloque y “ErrorID”, variable de tipo de dato Word, que nos dará a conocer la identificación de un error cuando se presente.

#### 4.1.6 MC\_MOVEABSOLUTE

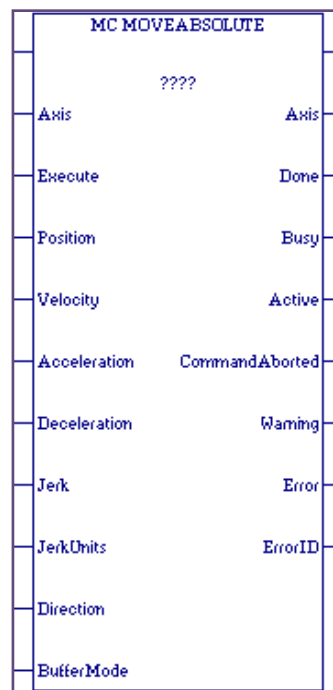


Figura 4.6 Mc\_Moveabsolute

El bloque MC\_MOVEABSOLUTE sirve para realizar movimientos de posiciones precisos con respecto a un punto de referencia.

En cuanto a entradas tenemos: Axis, acepta solo tipos de datos AXIS\_REF, aquí se especifica el eje al cual se tomará lectura de la posición, "Execute", acepta datos tipo "PowerFlow" necesariamente de flanco positivo, esta entrada es la orden para ejecutar el movimiento de tipo relativo en el eje especificado, Position, acepta datos de tipo LREAL constante o variable, aquí se especifica la posición a la cual se desea llevar el eje, esto es en Unidades de Usuario, "Velocity", acepta datos de tipo LREAL constante o variable, aquí se especifica la velocidad a la cual se desea alcanzar la posición a incrementar, "Acceleration", acepta datos de tipo LREAL constante o variable, sirve para establecer la aceleración a la cual se desea alcanzar la velocidad establecida, la unidad de este valor está en unidades de usuario por segundos al cuadrado y debe de estar dentro del rango preestablecido en la configuración, "Deceleration", acepta datos de tipo LREAL constante o variable, sirve para establecer la desaceleración cuando se desea descender hacia una la velocidad establecida, la unidad de este valor está en unidades de usuario por segundos al cuadrado y debe de estar dentro del rango preestablecido en la configuración, "Jerk", acepta datos de tipo LREAL constante o variable, en esta entrada se



especifica la tasa de cambio de la aceleración o desaceleración, si Jerk tiene valor 0 entonces se considera una tasa de cambio ilimitada, "JerkUnits", acepta datos de tipo MC\_JerkUnits constante o variable, se usa para especificar el tipo de unidades que toma Jerk, tenemos "UserUnitsPerSecondCubedyPercentJerkLimiting", la primera establece como unidad unidades de usuario por segundos al cubo, la segunda representa un porcentaje de aceleración o desaceleración yendo desde 0% hasta 100% en el tiempo, "Direction", acepta datos de tipo MC\_Direction constante o variable, sirve para especificar el sentido en el cual se moverá el eje para alcanzar la posición deseada. Presenta cuatro opciones: PositiveDirection, el eje se moverá en sentido "positivo" según lo previamente programado en la configuración de los parámetros. NegativeDirection, el eje se moverá en sentido "negativo" según lo previamente programado en la configuración de los parámetros. ShortestWay, el eje se moverá en el sentido en el cual haya menos distancia que recorrer. CurrentDirection, el eje no cambia de dirección, "BufferMode", acepta tipos de datos "MC\_BufferMode" variable o constante, esta entrada sirve para definir el comportamiento que tendrá el bloque en la acción con respecto a las entradas y salidas.

En cuanto a las salidas tenemos: Axis, acepta tipo de dato AXIS\_REF, sirve para indicar el eje sobre el cual está actuando el bloque, "Done", dato tipo "PowerFlow", esta salida se activa cuando el eje ha alcanzado la posición establecida según la entrada "Distance" en el bloque, "Busy", dato tipo "PowerFlow", indica si el bloque está habilitado y actualmente ejecutándose, "Active", es un dato tipo "PowerFlow", se activa cuando el bloque ha tomado dominio del eje, "ComandAborted", dato tipo "PowerFlow", se activa cuando el comando fue abortado por otro comando, "Warning", dato tipo "PowerFlow", que se activa cuando se presenta alguna advertencia con respecto al bloque, "Error", dato tipo "PowerFlow", que se activa cuando ocurre un error en la ejecución del bloque, y "ErrorID", variable tipo de dato Word, que nos dará a conocer la identificación de un error cuando este se presente.

### 4.1.7 MC\_ReadDigitalInput

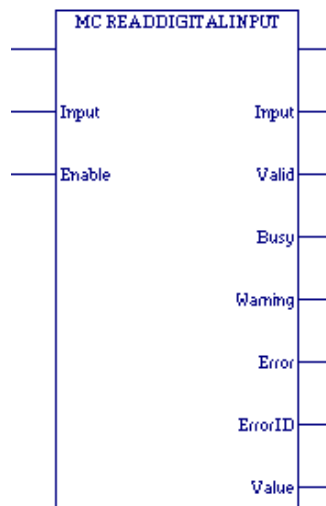


Figura 4.7 Mc Read Digital

Devuelve el valor del faceplate de la entrada discreta o FTB a una referencia específica en este caso a las entradas digitales como lo es I0001, lo que nos indica esto es que para crear automáticamente una variable del tipo de datos INPUT\_REF, primero se crea el parámetro de entrada para MC\_ReadDigital\_Input, de esta forma podemos ver en nuestra programación el comportamiento de todas las entradas ya sean normalmente abiertas o cerradas.

Todos los bloques necesitan un variable donde almacenar la información a esta variable se la llama variable de instancia, y en la programación en escalera vamos a darle nombre a esta variable en

la parte superior central de cada bloque, el nombre que lleve esta variable puede ser el que nosotros deseemos.

#### 4.1.8 MC\_WriteDigitalOutput

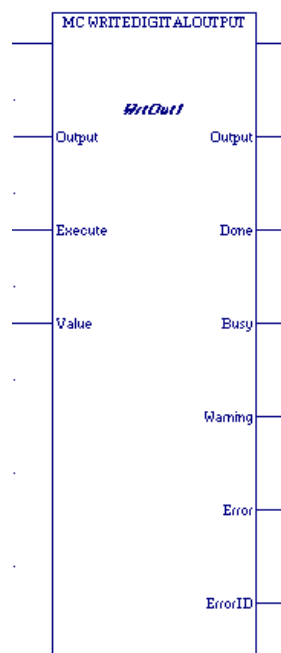


Figura 4.8 Mc Write Digital

El bloque de función MC\_WriteDigitalOutput escribe un valor, una vez que el FTB o faceplate ha sido especificado por el parámetro de entrada.

Para ejecutar otra operación la salida "ejecute" del bloque debe estar en un flanco de bajada.

En estos bloques solo vamos a tener tres parámetros de entrada los cuales son: output, execute y value los cuales cumplen funciones específicas, así por ejemplo en el parámetro output colocamos la variable de salida a la que queremos dirigir nuestra señal de salida, en este ejemplo, M4\_FTB\_out1 se ha definido como la variable de salida número uno del módulo cuatro del Fiber terminal block, “execute” ordena al bloque cumplir con función.

## 4.2 Programación de las I/O:

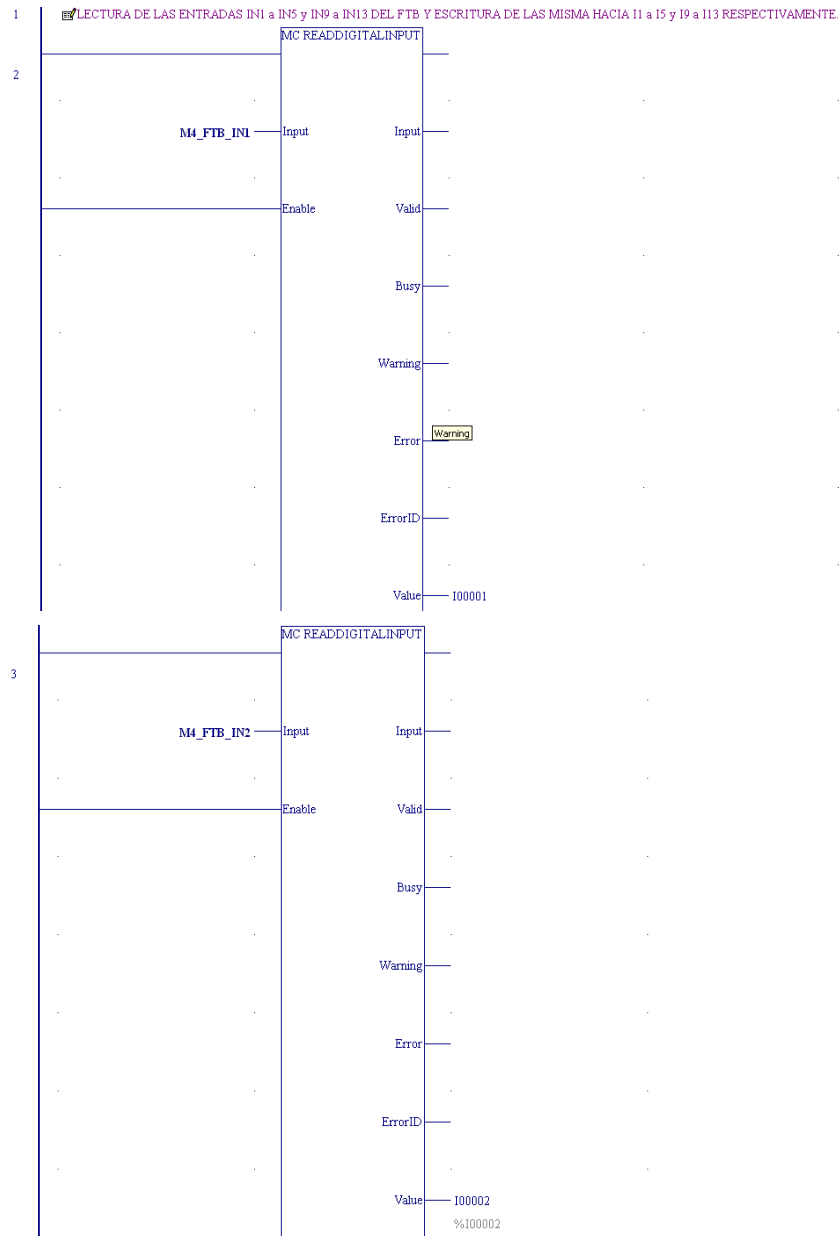


Figura 4.9 Programación de las entradas i0001- i0002

De esta manera seguimos programando las demás entradas.

### Programación de la lectura de las salidas Q al FTB

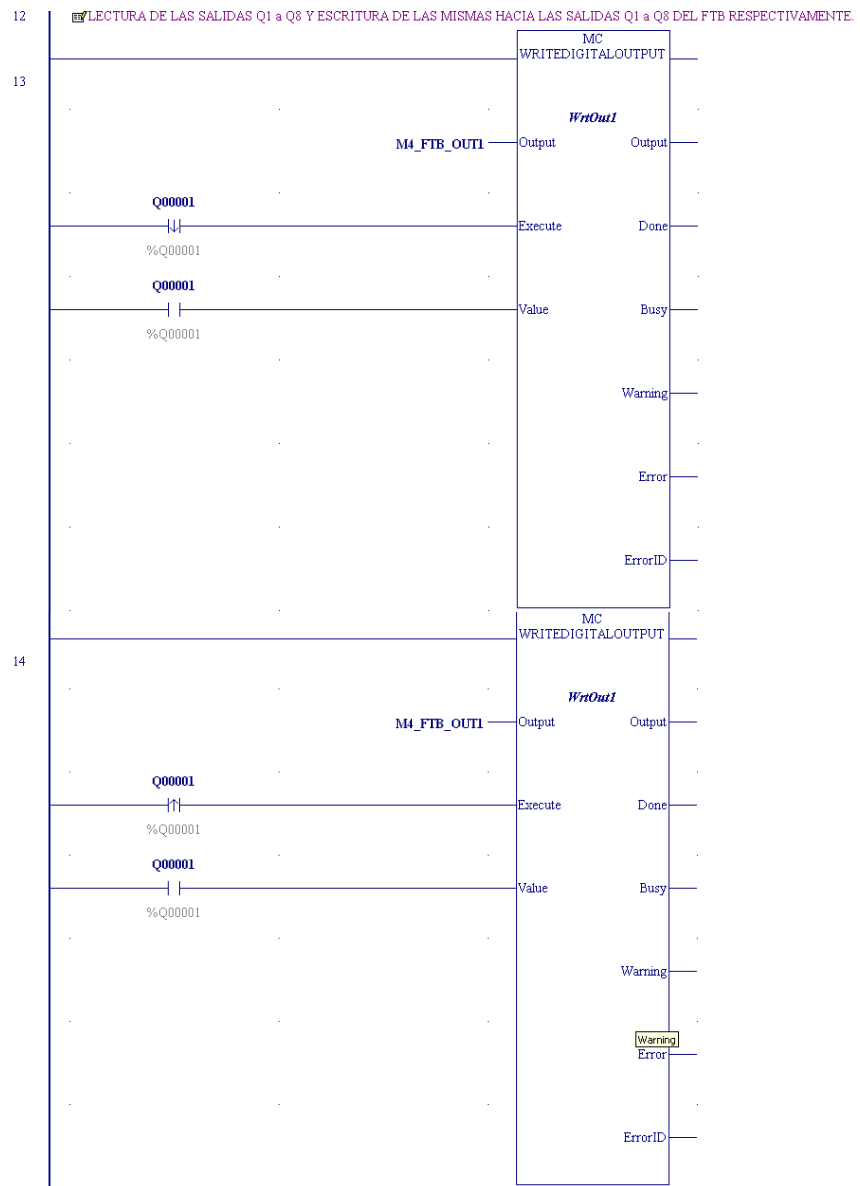


Figura 4.10 Programación de la salida Q0001

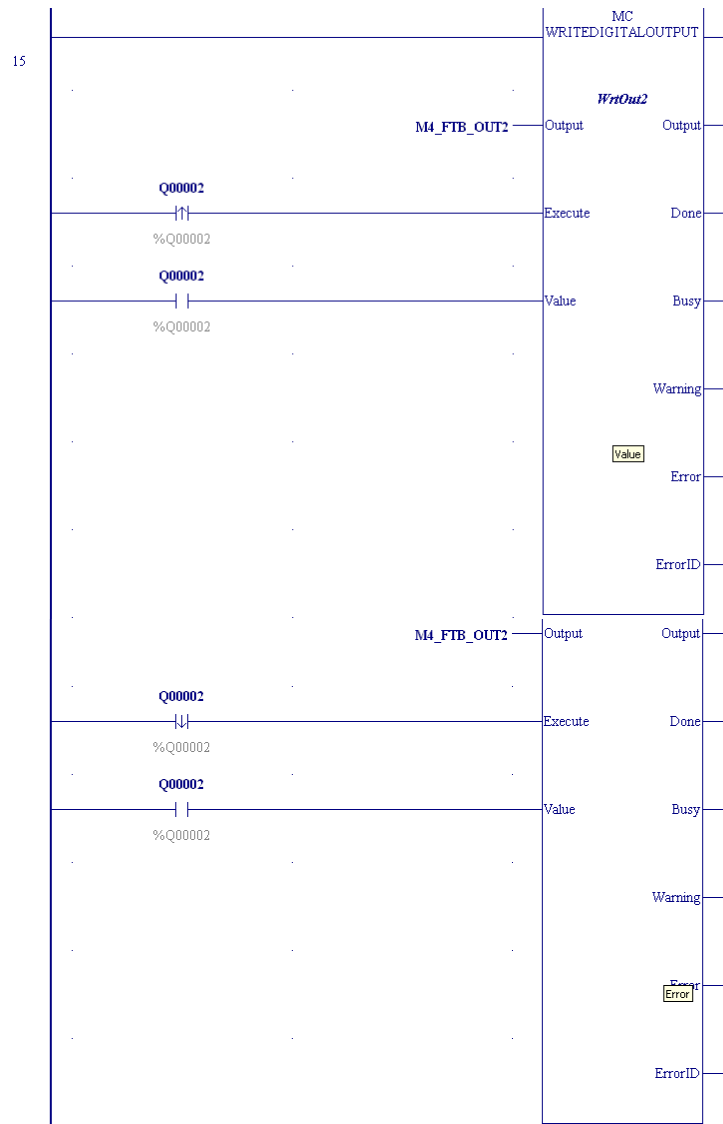


Figura 4.11 Programación de la salida q0002

De esta manera seguimos programando las demás salidas.



### Programación de la lectura de las salidas analógicas

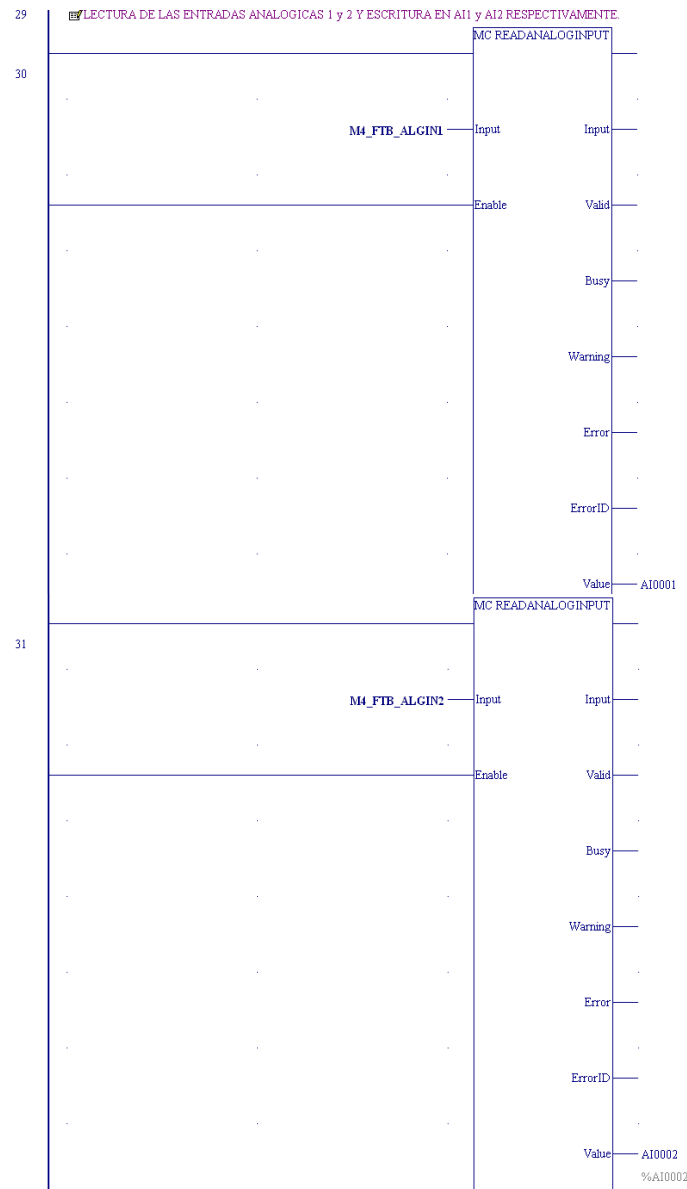


Figura 4.12 Programación de las entradas analógicas

### 4.3 Programación del MC\_Power:

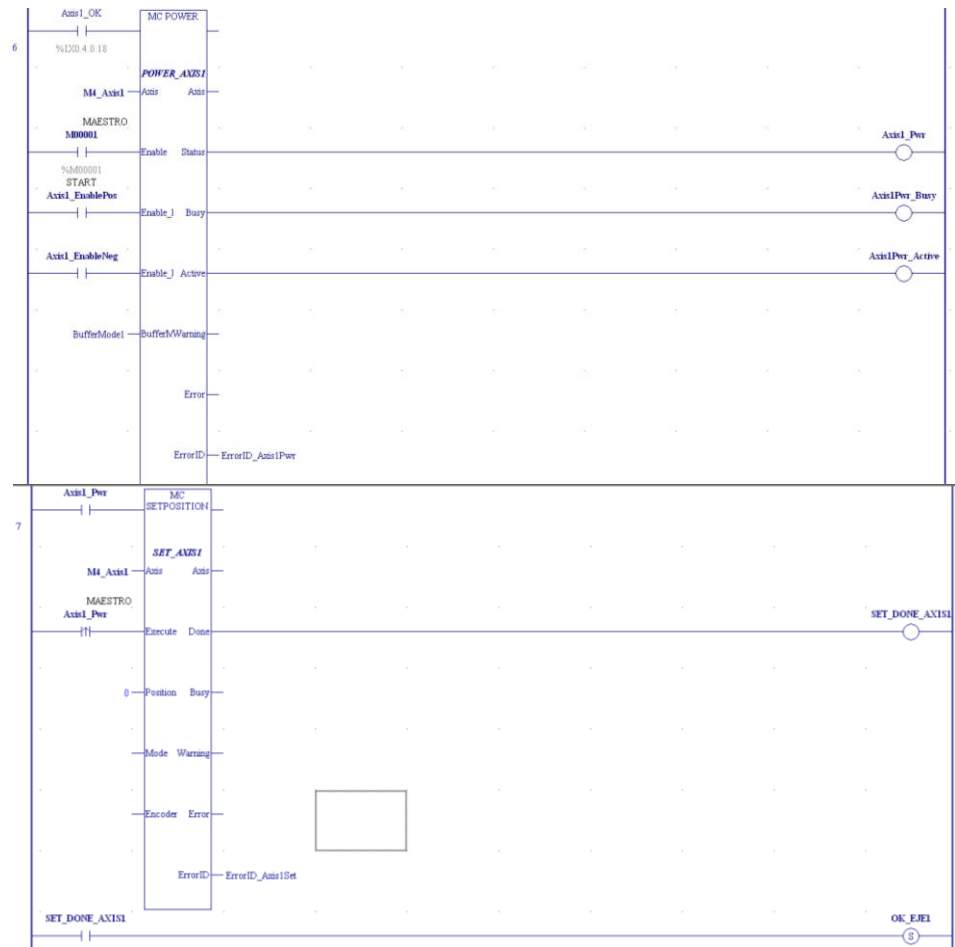


Figura 4.13 Bloque Mc\_Power y Mc\_Setposition

Una vez programado los bloques principales para la ejecución de los servomotores vamos a llevar a la práctica cada uno de los bloques antes mencionados.

#### **4.4 Programa de control de velocidad manual**

##### **Descripción general**

El objetivo de realizar la programación de un control de velocidad manual para uno de los cuatro servomotores es la introducción en el uso de los bloques básicos de movimiento.

Vamos a manipular la velocidad de un eje ya sea ingresando valores por teclado dentro del software Proficy Machine Edition durante la ejecución del programa o mediante el uso de una entrada analógica del Fiber Terminal Block, además, se hace uso de sus entradas digitales para simular pulsadores de Marcha y Paro.

## Especificaciones de la programación



Figura 4.14 Llamado de las subrutinas

Para facilidad y orden en la programación dentro del software se hace uso de las subrutinas.

Se empieza con el llamado de todas las subrutinas en el programa principal.

Dentro del programa se observan cuatro subrutinas:

- **Subrutina M4\_FTB.**

Dentro de esta subrutina esta la programación necesaria para leer y escribir datos digitales y analógicos del Fiber Terminal Block.

- **Subrutina POWER.**

En esta subrutina se encuentra la programación para el encendido del servoamplificador del eje que se va a controlar.

- **Subrutina ON\_STOP.**

Se la utiliza para dar marcha y paro del programa que controla la velocidad del servomotor.

- **Subrutina VELOC.**

Dentro de esta subrutina esta la programación para el control manual de la velocidad del servomotor.

### **Encendido del servoamplificador.**

Para el control de encendido y apagado del servoamplificador se utilizan las entradas del FTB I1 e I2 para encender y apagar respectivamente. Adicional se colocan en paralelo bobinas auxiliares si se desea hacer un control desde un software Scada.

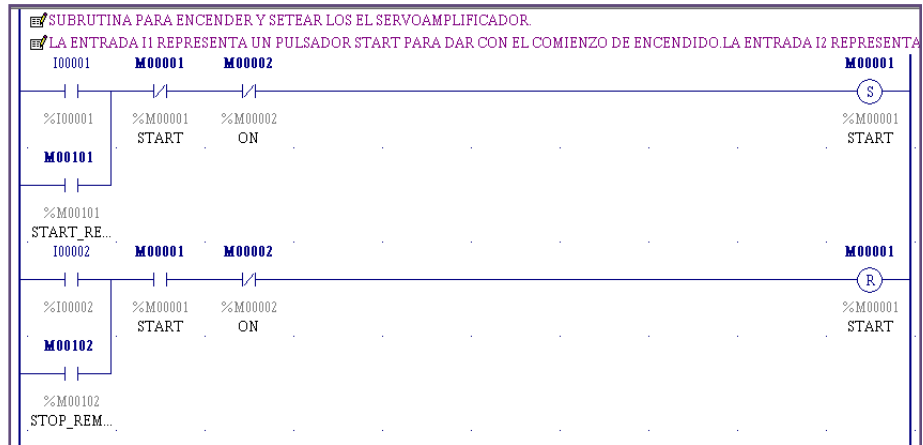


Figura 4.15 Pulsadores de encendido y apagado del servomotor.

La bobina M1 (START) envía la orden de encendido hacia el bloque MC\_POWER donde se especifica el eje que se va a controlar, en este caso se trata del primer eje del cuarto modulo (M4\_Axis1).

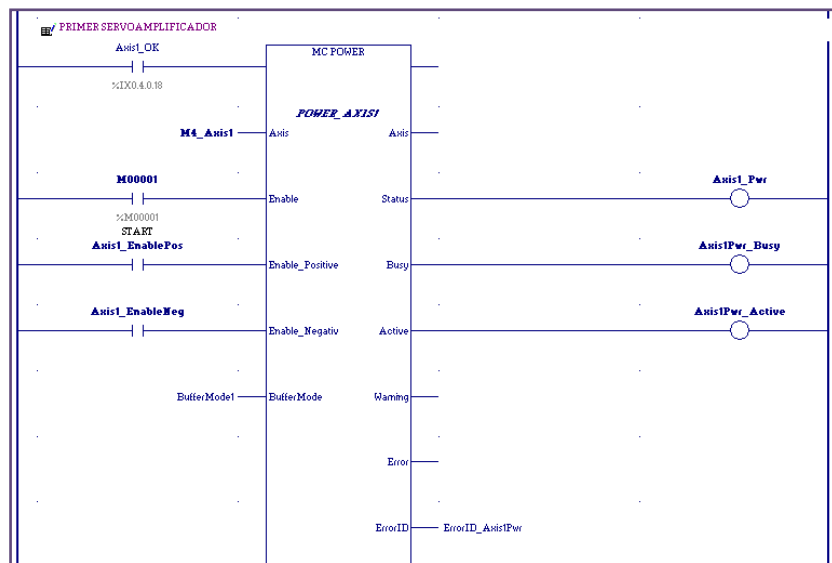


Figura 4.16 Encendido del eje 1 del modulo 4, usando el bloque Mc\_Power

Nótese que el bloque lleva como variable interna “POWER\_AXIS1” en donde se direccionan y almacenan sus entradas y salidas.

La habilitación del bloque se realiza por medio de un contacto el cual esta referenciado a un parámetro del Status Data del PMM. Este parámetro es el # 2019 está ubicado en la dirección 18 del Status Data y se le ha colocado el nombre de "Axis1\_OK".

Dentro de la programación del encendido del servoamplificador se hace también uso del bloque MC\_SETPOSITION el cual es usado en este punto para establecer la posición "cero" dentro del rango de las unidades de usuario especificadas en la configuración del eje. Cabe recalcar que este eje ha sido configurado con 100uu correspondiente a una revolución.

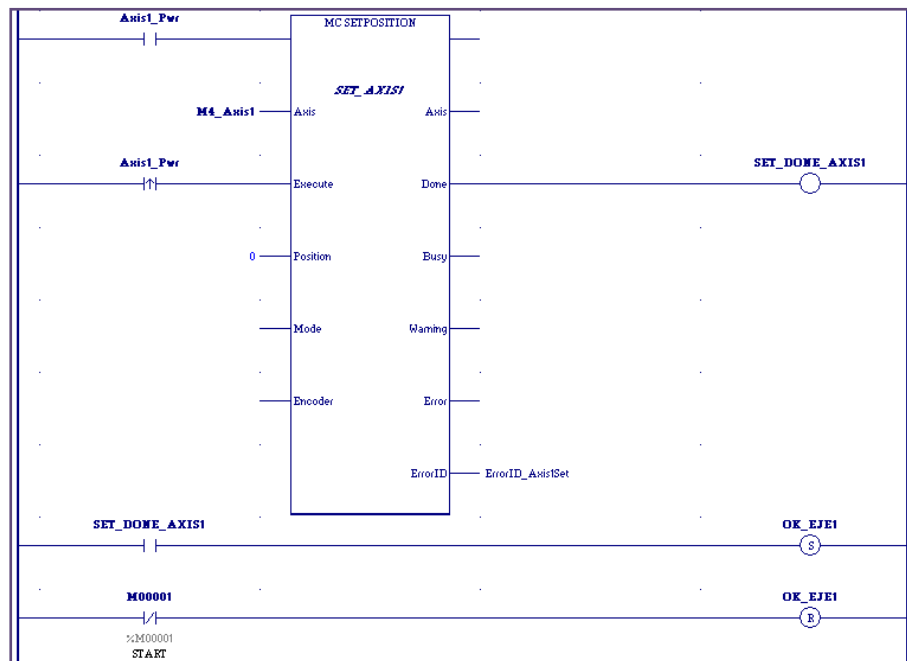


Figura 4.17 Uso del bloque Mc\_Setposition

La variable de control de este bloque tiene el nombre de “SET\_AXIS1”. Solo hace falta un flaco positivo para fijar la posición. Cuando este bloque se ejecuta correctamente la bobina “SET\_DONE\_AXIS1” se activa lo cual es utilizado para activar la bobina “OK\_EJE1” que a su vez habilita el pulsador de marcha en la subrutina ON\_STOP.

### Marcha y paro del eje.

Dentro de la subrutina ON\_STOP se encuentran dos entradas del FTB I3 e I4 que simulan los pulsadores ON y OFF respectivamente para comenzar con el movimiento del eje. Adicional a estas entradas se encuentran en paralelo bobinas auxiliares si se desea realizar un control desde un software Scada.

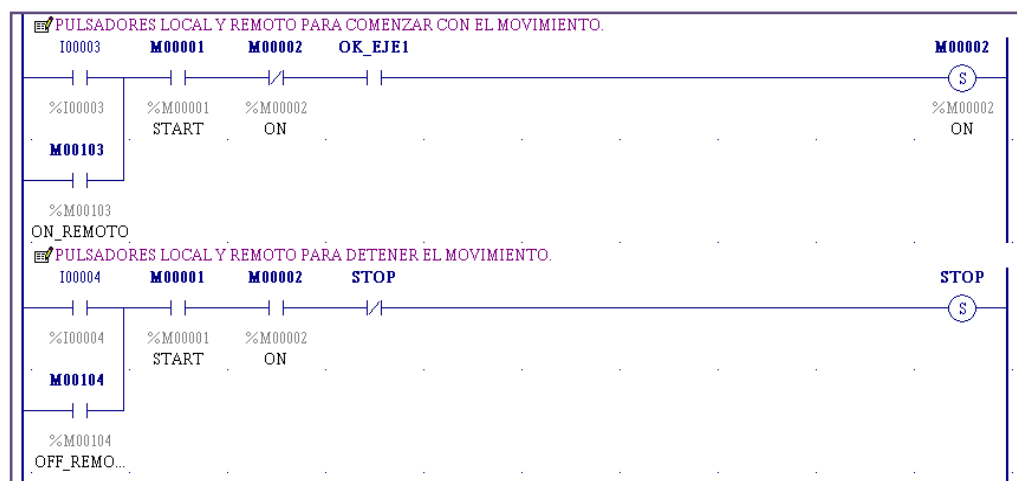


Figura 4.18 Pulsadores ON y OFF para habilitar el movimiento del eje.



Nótese que la bobina M2 que es la encargada de dar paso al bloque de movimiento, solo es habilitada si el eje fue encendido correctamente en la subrutina POWER.

El bloque MC\_STOP es el encargado de detener correctamente el eje en movimiento para evitar cualquier error durante el apagado del servoamplificador.

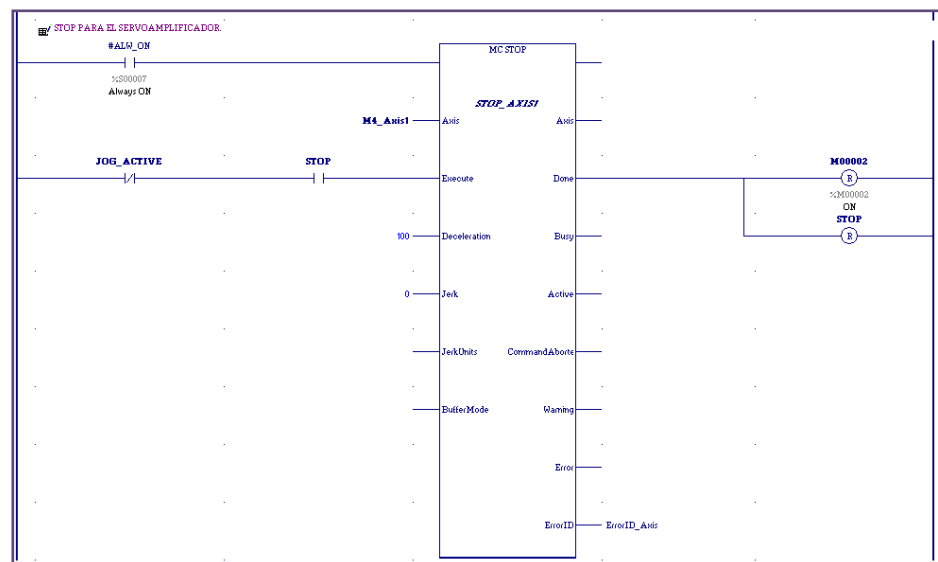


Figura 4.19 Uso del bloque Mc\_Stop para detener el eje.

Este bloque lleva como variable de referencia "STOP\_AXIS1" y está programado para que actúe solamente si el eje ya no está siendo manipulado por el bloque encargado de darle movimiento, esto es posible gracias a un contacto normalmente cerrado llamado

“JOG\_ACTIVE” el cual proviene del bloque MC\_JOGAXIS (indicado más adelante) e indica si este último bloque esta activo en el eje.

#### 4.5 Control básico del equipo de entrenamiento BETA I SERVO.

Previo al bloque de control de velocidad se ha colocado una bobina auxiliar M3 que por defecto está habilitando el movimiento en sentido positivo según lo configurado dentro de los parámetros del PMM. Esta bobina puede ser remplazada posteriormente por una entrada del FTB para tener elección sobre el sentido de giro del eje. M3 también puede ser manipulada desde un software Scada como lo es InTouch.

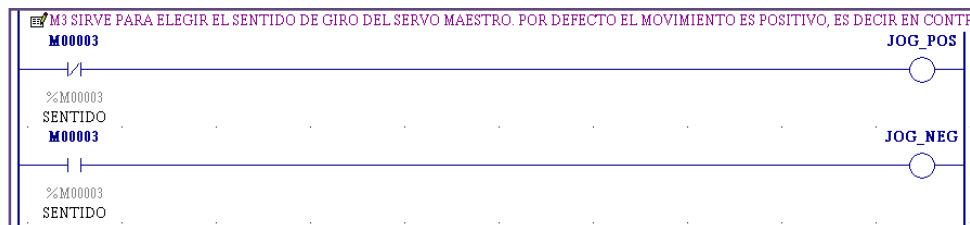


Figura 4.20 Elección del sentido de giro del eje.

La velocidad del eje puede ser comandada de dos formas, por una entrada analógica o por el ingreso de un valor por teclado durante la ejecución del programa. Para esta elección se dispone de la bobina auxiliar M4, que, al igual a M3 puede ser remplazada por una entrada del FTB o manipulada desde un software Scada. Por

defecto M4 está habilitando la entrada analógica la cual tiene un valor máximo en bits de hasta 32767. Este valor es llevado a un escalamiento de 0 como mínimo a 100 como máximo. De acuerdo al máximo valor especificado en las Unidades de Usuario (100uu) entonces se tiene que la velocidad máxima es de 1 revolución por segundo.

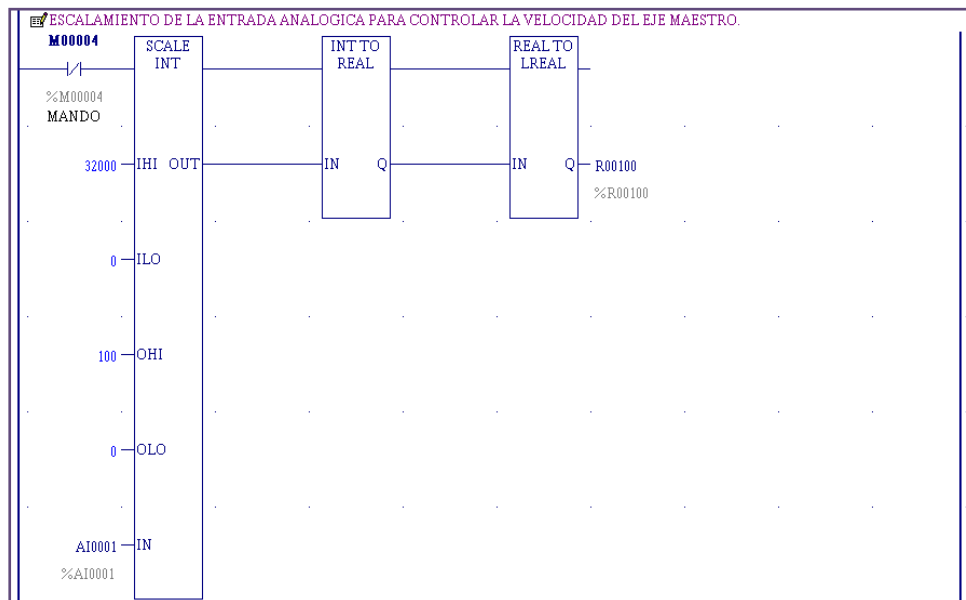


Figura 4.21 Escalamiento y conversión del tipo de dato.

Dado a que la entrada "JogVelocity" del bloque MC\_JOGAXIS solo acepta datos de tipo LREAL se lleva el valor escalado previamente de la entrada analógica hacia un bloque INT TO REAL y por ultimo al bloque REAL TO LREAL donde el dato es almacenado en una nueva variable y colocado en la entrada respectiva.

Para el caso de que se desee ingresar valores desde el teclado para manipular la velocidad del eje, M4 deberá estar activado. El valor se ingresa en el registro R3 (seleccionar el registro, presionar Ctrl + Enter, ingresar valor, enter), éste valor se lleva a REAL y posteriormente a LREAL para su ingreso en la entrada respectiva.

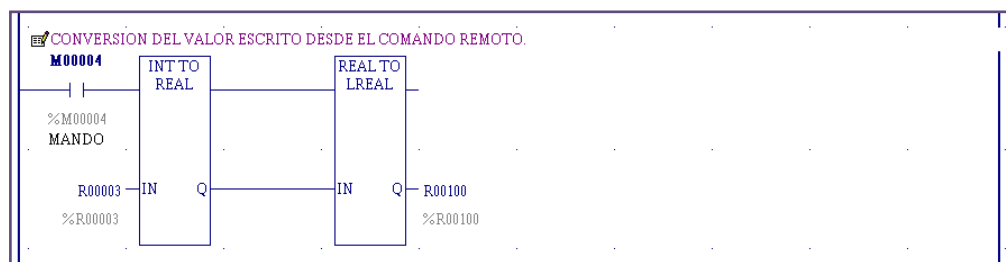


Figura 4.22 Ingreso de valor de velocidad por teclado.

Finalmente el bloque MC\_JOGAXIS que es el encargado de variar la velocidad, actúa sobre el eje indicado siempre que no se haya activado la señal de stop.

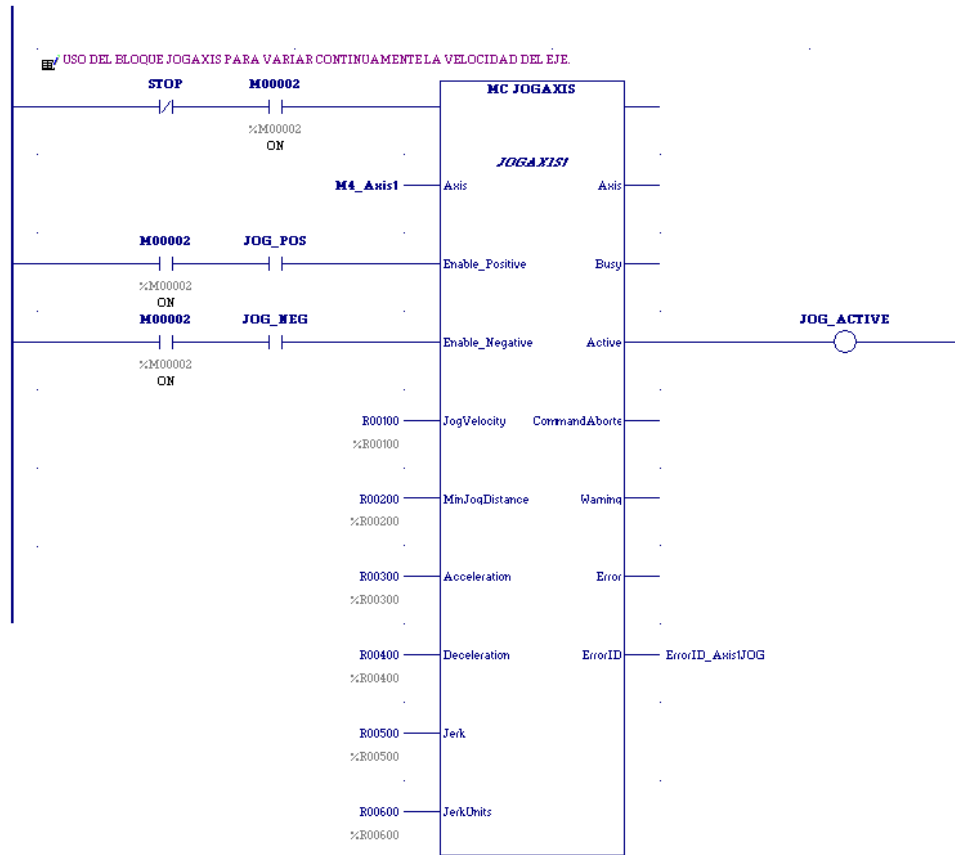


Figura 4.23 Bloque Mc\_Jogaxis utilizado para variar la velocidad en el eje.

Este bloque tiene como variable de referencia “JOGAXIS1” y posee registros donde se encuentran preestablecidos los valores de MinJogDistance, Acceleration, Deceleration, Jerk y JerkUnits.

#### **4.6 Práctica No 4.**

Familiarización con los bloques de movimiento del software de programación “Proficy Machine Edition.

##### **Objetivos:**

- Familiarización con los principales bloques de programación de movimiento.
- Introducción a la programación de las funciones y bloques de movimientos que soporta el modulo PMM.

##### **Enunciado de la práctica:**

Con esta práctica vamos a familiarizarnos con los bloques de movimiento, entradas y salidas digitales, entrada y salida analógica, habilitar los amplificadores, y el funcionamiento de algunos bloques necesarios para las siguientes prácticas.

#### **4.7 Práctica No 5.**

Control básico del equipo de entrenamiento “Beta I Servo” por medio del software de programación “Proficy Machine Edition”.

##### **Objetivos:**

- Control del equipo de entrenamiento “Beta I Servo”, entradas digitales y control de los dos servomotores del equipo.
- Familiarización con los principales bloques de programación.
- Introducción a la programación de las funciones y bloques de movimientos que soporta el modulo PMM.

##### **Enunciado de la práctica:**

El objetivo de realizar la programación de un control de velocidad manual para uno de los cuatro servomotores gracias a los conocimientos adquiridos en la práctica anterior.

Llevar a cabo la manipulación de la velocidad de un eje ya sea ingresando valores por teclado dentro del software Proficy Machine Edition durante la ejecución del programa o mediante el uso de una entrada analógica del Fiber Terminal Block del cual, además, se hace uso de sus entradas digitales para simular pulsadores de Start, Stop, Marcha y Paro.

# CAPITULO 5

## Automatización Industrial bajo el sistema PACMotion

### 5.1 Conceptos básicos de desarrollo para la programación

Antes de entrar por completo a la programación debemos conocer algunos conceptos que no ayudarán a lo largo del desarrollo de la programación.

#### 5.1.1 Unidades de Usuario

Antes de entrar en detalle se debe tener claro cómo definir los parámetros “Motor EncoderUserUnits” (unidades de usuario) y “Motor EncoderCounts” (Conteos).

El parámetro “Motor EncoderUserUnits” está relacionado con las unidades de usuario. El parámetro se refiere a las unidades con las que desea trabajar el programador, el cual realmente es un número



definido por el mismo para cada eje en particular. Ofrece al programador una ventaja sobre las dimensiones que tiene cada movimiento. Su rango permitido va desde  $1.0 \times 10^{-6}$  hasta 131072.0 y se utiliza para calcular la relación UserUnits/Counts.

El parámetro Motor EncoderCounts está relacionado con las divisiones del encoder y también es utilizado para calcular la relación UserUnits/Counts. Su rango permitido va desde 1 hasta 65536 para el caso de nuestro modelo de servomotor. El valor 65536 equivale al número máximo de conteos en un ángulo de  $360^\circ$ .

Visto gráficamente sería de la siguiente manera:

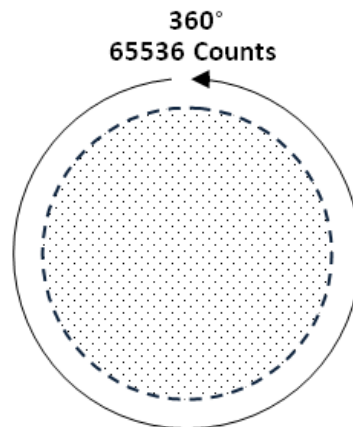


Figura 5.1 Número de conteos en 360 grados.

El modelo de nuestro servomotor dispone de 65536 divisiones para  $360^\circ$ .

La relación UserUnits/Counts establece la equivalencia entre conteos y una unidad de usuario, además bajo esta relación se basan los bloques de programación para realizar movimientos precisos establecidos en las unidades de usuario. Es necesario que esta relación siempre tenga que ser menor a 1.

Por ejemplo, si el parámetro Motor EncoderUserUnits es igual a 360 grados y el parámetro Motor EncoderCounts igual a 65536 entonces la relación UserUnits/Counts es la siguiente:

$$Uu/Cts = 360/65536$$

Con esta igualdad se tiene 1 grado de giro para una unidad de usuario.

La siguiente relación es equivalente a la anterior y por tanto se obtiene el mismo ángulo de giro por unidad de usuario.

$$Uu/Cts = 180/32768$$

Los parámetros Motor Encoder Position Range y Motor EncoderLow Position son importantes para definir mediante un rango la posición actual del servomotor.

El parámetro “Motor EncoderLow Position” establece la posición mínima para el eje mientras que parámetro “Motor Encoder Position Range” define su rango.

Si para el ejemplo anterior cuya relación es  $Uu/Cts = 360/65536$ , definimos Motor EncoderLow Position = 0, y Motor Encoder Position Range = 360 entonces el eje tendrá un rango que va desde 0 hasta 360 unidades de usuario (360 Uu), en otras palabras, con 360 Uu el eje dará una revolución y volverá a estar en su posición inicial cero.

Si en otro caso para la misma relación  $Uu/Cts$  definimos Motor EncoderLow Position = 0, y Motor Encoder Position Range = 720, entonces el eje tendrá que girar dos revoluciones para regresar a su posición inicial cero.

Cada uno de estos parámetros puede ser configurado tanto antes o después de la programación, bien configurando sus valores directamente desde la configuración del hardware desde el software de programación o haciendo uso de los bloques de escritura de parámetros.

### 5.1.2 Sincronización

La sincronización es el método por el cual se logra que varios ejes se ejecuten a la vez con el propósito de completar una tarea y evitar así condiciones de carrera, que pudieran desembocar en un estado inesperado.

Como ejemplo vamos a definir como límites que el último eje (eje esclavo 3) tenga una velocidad máxima de 1 rev/seg, por ende el eje esclavo 2 tendrá una velocidad máxima de 2 rev/seg, el eje esclavo 1 una velocidad máxima de 4 rev/seg. Y el eje maestro una velocidad máxima de 8 rev/seg. Tal como se muestra en la siguiente tabla:

**Tabla XIV. Límites de velocidad para los cuatro ejes**

<b>Eje</b>	<b>Velocidad máxima (rev/seg)</b>
Eje Maestro	8
Eje Esclavo 1	4
Eje Esclavo 2	2
Eje Esclavo 3	1

La velocidad máxima del Eje Esclavo 3 puede ser cambiada a conveniencia del usuario durante la programación.

En la siguiente figura se muestra una analogía del esquema de las dependencias entre los cuatro ejes por medio de poleas:

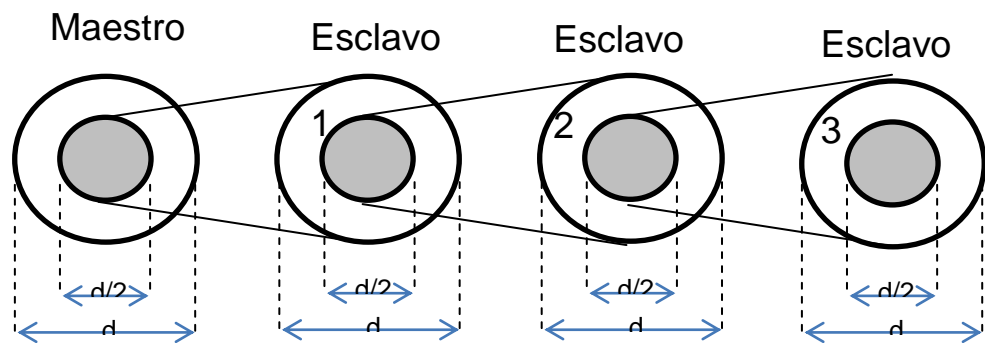


Figura5.2 Simulación y programación del control de los ejes de los servomotores.

## 5.2 Aplicaciones y desarrollo de proyectos didácticos.

### 5.2.1 Configuración y programación de los cuatros ejes de los servomotores para la simulación del comportamiento de un reloj.

Para conseguir la programación de un reloj vamos a realizar lo siguiente:

Cada eje de los servomotores debe de ser configurado para que avance los grados necesarios para marcar el tiempo requerido por la manecilla correspondiente, esto se realiza previamente en la configuración de los parámetros de cada eje.

Se utilizarán los cuatro ejes, el primero para marcar las horas (12 horas), el segundo para marcar los minutos (60 minutos), el tercero para marcar los segundos (60 segundos) y el cuarto como semilla para generar los segundos. Este último tendrá 100 divisiones para una revolución y su velocidad será de una revolución por segundo. La semilla se utilizará para incrementar cada segundo.

El servomotor que actúa como semilla da pulsos cada segundo a medida que pasa por la posición cero de su mismo eje, estos pulsos son recibidos por el servomotor que actúa como segundero y a su vez lo hace avanzar una unidad de usuario respectivamente (un segundo, 6 grados), cuando el segundero pasa por su posición cero envía un pulso hacia el minuterero el cual avanza una unidad de usuario respectivamente (un minuto, 6 grados), de igual manera cuando el servomotor que actúa de minuterero pasa por su posición cero este enviara un pulso hacia el que marca las horas haciendo que este avance una unidad de usuario (una hora, 30 grados).

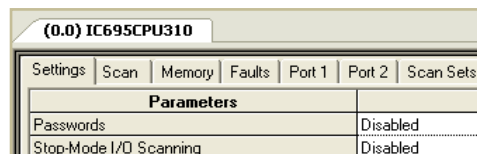
Las posiciones de cada eje se leerán con el bloque `MC_READACTUALPOSITION` el mismo que pasa la información a un registro de tipo `LREAL`, para luego pasar este registro por una serie de bloques que lo convertirán a tipo `INT`.

## CONFIGURACIÓN DE PARÁMETROS

La configuración del hardware para los módulos es la siguiente:

- Slot 0 con el CPU
- Slot 2 con la Fuente de Poder
- Slot 4 modulo PMM
- Slot 5 modulo PMM

### Configuración Del CPU



(0.0) IC695CPU310	
Parameters	
Passwords	Disabled
Stop-Mode I/O Scanning	Disabled

Figura 5.3 Configuración Settings

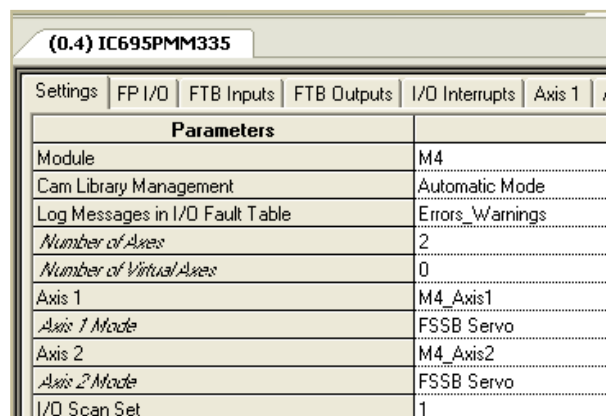
En la configuración del CPU deshabilitamos el Passwords ya que no haremos uso de este parámetro.

EL protocolo del puerto 1 del CPU (RS-232) lo cambiamos a modo SNP Slave para realizar la comunicación con InTouch mediante la interfaz GESNP. Para ello también le damos un SNP ID necesario para la comunicación con el Scada.

## Configuración del PMM

Seleccionamos el módulo IC695PMM335 y en la ventana Inspector cambiamos el parámetro “Variable Mode” a verdadero (True).

Con esto haremos uso de los parámetros del STATUS del PMM para verificar el estado de la configuración de los ejes.



Parameters	
Module	M4
Cam Library Management	Automatic Mode
Log Messages in I/O Fault Table	Errors_Warnings
<i>Number of Axes</i>	2
<i>Number of Virtual Axes</i>	0
Axis 1	M4_Axis1
<i>Axis 1 Mode</i>	FSSB Servo
Axis 2	M4_Axis2
<i>Axis 2 Mode</i>	FSSB Servo
I/O Scan Set	1

Figura 5.4 Configuración parámetros

En la pestaña Settings del módulo 4 cambiamos el parámetro “Log Messages in I/O FaultTable” a “Errors\_Warnings” para que nos indique los errores y advertencias que se presenten en la programación. En el parámetro “Number of Axes” colocamos “2” ya que controlaremos 2 servomotores con cada módulo PMM. Los demás parámetros están con sus valores por defecto.

En la pestaña FTB Inputs habilitamos el “Fiber Terminal Block Settings” para usar y configurar este modulo adicional.



Deshabilitamos el parámetro “Fiber Terminal Block Identification Mode” para que nos permita utilizar ambos FTB en cada modulo.

A partir de la entrada IN 17 hasta la entrada IN 28 deshabilitamos el parámetro “FTB IN xx FaultDetect” para el caso de que estas entradas no estén conectadas.

Si este parámetro está habilitado nos arroja un warning durante la ejecución del programa por cada entrada que no esté físicamente cableada.

(0.4) IC695PMM335					
Settings	FP I/O	FTB Inputs	FTB Outputs	I/O Interrupts	Axis
<b>Parameters</b>					
<i>FTB IN16</i>					Digital Input
		FTB IN16 Input Ref			M4_FT_B_IN16
<i>FTB IN17</i>					Fast Digital Input
		FTB IN17 Input Ref			M4_FT_B_IN17
<i>FTB IN17 Mode</i>					Differential
<i>FTB IN17 Fault Detect</i>					Disabled
<i>FTB IN18</i>					Fast Digital Input
		FTB IN18 Input Ref			M4_FT_B_IN18
<i>FTB IN18 Mode</i>					Differential
<i>FTB IN18 Fault Detect</i>					Disabled
<i>FTB IN19</i>					Fast Digital Input

Figura 5.5 Configuración FTB Inputs

### **PMM - AXIS 1 (Segundos)**

En la pestaña Axis 1 configuramos los parámetros que nos interesan para que se realice el movimiento de la manecilla del segundero.

En “Axis PositioningMode” cambiamos a “Rotary” ya que haremos un control por rotación.

En “Motor EncoderMode” cambiamos a “Incremental” ya que no disponemos de una batería para el amplificador.

En “Motor EncoderUserUnits” escribimos 60 que serán las divisiones para una vuelta del eje, es decir cada segundo.

En “Motor EncoderCounts” escribimos “65536” que es el máximo de conteos permitidos en una vuelta disponibles en nuestro tipo de servomotor. Entonces, por cada unidad de usuario, especificada anteriormente se tendrán  $60/65536$  conteos.

En “Motor Encoder Position Range” escribimos el valor que deseamos como rango para el movimiento del eje 1 del módulo 4. Para nuestro caso “60”, este parámetro se especifica en unidades de usuario.

En “Motor EncoderLow Position Limit” escribimos la mínima posición que deseamos para el movimiento del eje. En nuestro caso “0”, este valor se especifica en unidades de usuario.

Deshabilitamos el parámetro “OverTravelLimit Switch” ya que no especificaremos un final de carrera.

En “Axis Direction” cambiamos a “Reverse” para que el eje gire en sentido de las manecillas del reloj.

Deshabilitamos “Software End of Travel” ya que no usamos un final de carrera.

Colocamos al parámetro “Max Jerk” dentro de un rango permitido.

En el parámetro “Motor Type” escribimos “281” ya que este es el código para el modelo de nuestro servomotor.

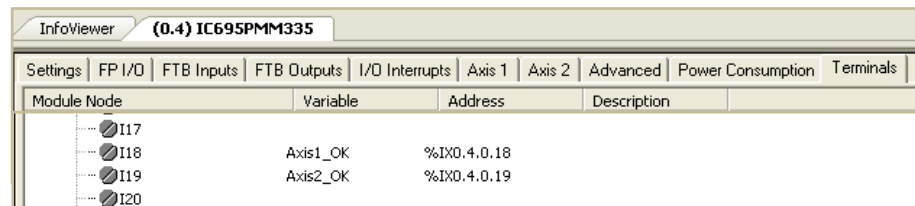
Colocamos al parámetro “Error Stop Jerk” dentro de un rango permitido.

### **PMM - AXIS 2 (Semilla)**

Este eje se configura de la misma manera que el eje 1 con la única diferencia de que las Unidades de Usuario y el Rango cambian a “100”.

No es necesario cambiar la dirección de movimiento ya que este eje será controlado con el MC\_JOGAXIS el cual permite controlar movimientos positivos y negativos. Por último ajustamos “Max Jerk” dentro de un rango permitido.

En la pestaña “Terminals” podemos observar los bits del Status Data del PMM. A medida que en la programación agregamos las variables con sus nombres (por ejemplo un contacto con el nombre de “Axis1\_OK”) automáticamente esta variable se direcciona en el bit correspondiente del Status Data del PMM. De no ser así, hacemos clic en el bit que deseamos asignar y buscamos la variable a la que asignaremos el bit.



Module Node	Variable	Address	Description
I17			
I18	Axis1_OK	%IX0.4.0.18	
I19	Axis2_OK	%IX0.4.0.19	
I20			

Figura 5.6 Configuración de la pestaña terminal

## CONFIGURACIÓN DEL PMM (SLOT 5)

De igual forma que en modulo 4 seleccionamos dicho modulo desde la venta del “Navigator” y en la ventana del “Inspector” cambiamos a “True” en valor del modo de variable. Esto es para habilitar la pestaña “Terminals” en la configuración del PMM.

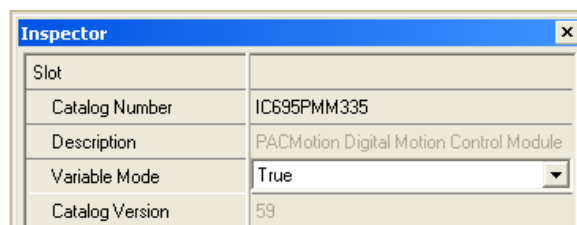


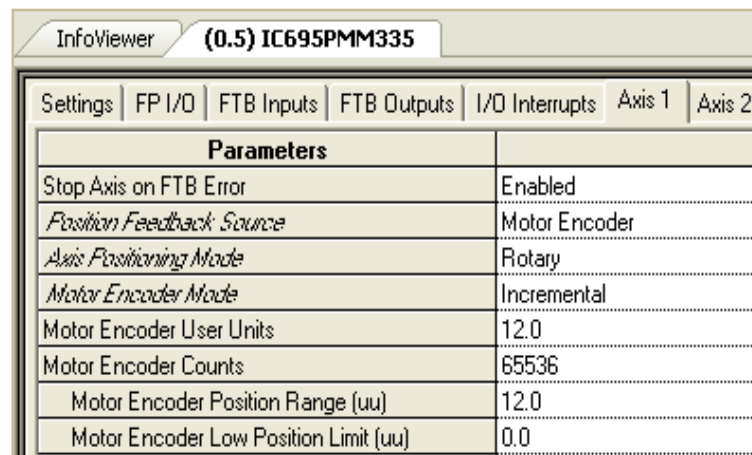
Figura 5.7 Habilitación de pestaña terminal

La configuración del PMM del módulo del slot 5 es similar a la del PMM del slot 4 con la única diferencia de la asignación de las unidades de usuario, las unidades de usuario cambian en el eje que indican la hora, para este caso son solamente 12.

Los demás parámetros vienen por default.

### PMM - AXIS 1 (Horas)

En la pestaña Axis 1 de la configuración del módulo PMM del slot 5 escribimos las unidades de usuario en el parámetro “Motor Encoder User Units”, para este caso “12” ya que este eje representa la manecilla que indica la hora.



Parameters	
Stop Axis on FTB Error	Enabled
<i>Position Feedback Source</i>	Motor Encoder
<i>Axis Positioning Mode</i>	Rotary
<i>Motor Encoder Mode</i>	Incremental
Motor Encoder User Units	12.0
Motor Encoder Counts	65536
Motor Encoder Position Range (uu)	12.0
Motor Encoder Low Position Limit (uu)	0.0

Figura 5.8 Configuración para el eje

### PMM - AXIS 2 (Minutos)

La configuración de los parámetros del eje (Axis 1 – PMM Slot 5) que marcan los minutos es exactamente igual a la configuración del eje

que marcan los segundos (Axis 2 – PMM Slot 4) ya que el valor de las unidades de usuario es “60” para ambos.

Configurados los cuatros ejes con sus unidades de usuario realizamos la programación de los mismos de tal forma que cada eje represente cada manecilla de un reloj, más un eje que funciona como semilla para generar los segundos.

## **PRINCIPALES BLOQUES UTILIZADOS**

### **MC\_POWER**

Usaremos un bloque MC\_POWER por cada servomotor para encender los amplificadores de cada uno de los ejes.

A este bloque le asignamos una variable que le servirá de referencia para almacenar los datos tanto de sus entradas como sus salidas. Además es necesario usar la entrada para habilitar el bloque, Axis para especificar el eje que se desea energizar, Enable, para dar la orden de encendido, Enable\_Positive y Enable\_Negative, para habilitar los sentidos de movimientos del eje, BufferMode, para definir el comportamiento del bloque.

En la salida usaremos Status, para conocer si el bloque está activado, Busy, indica si el bloque se está habilitado y ejecutándose, Active, se

activa cuando el bloque ha tomado dominio del eje, y ErrorID, que nos dará a conocer la identificación de un error cuando este ocurra.

### **MC\_SETPOSITION**

Fija una posición determinada dentro del rango preestablecido en la configuración del eje. Este bloque será usado para fijar la posición cero de cada eje. Se usarán las entradas de habilitación, Axis, Execute, que dará la orden para fijar la posición, Position, donde escribimos la posición que deseamos fijar. Entre las salidas se dará uso a Done, que muestra una señal cuando el bloque ya actuó sobre el eje, y ErrorID, para identificar cualquier clase de error que pueda ocurrir dentro de la ejecución del programa. Este bloque al igual que el anterior requiere de una variable para guardar sus datos de entradas y de salidas.

### **MC\_JOGAXIS**

MC\_JOGAXIS dará movimiento al eje definido como semilla el cual pondrá en marcha a los segundos. En este bloque se especificará la velocidad en unidades de usuario para que el eje responda a una revolución por segundo, para nuestro caso 100Uu/seg.

A este bloque le asignamos una variable de referencia, una línea de habilitación, Axis, el eje que se desea controlar, Enable\_Positive, un contacto para la dirección positiva de movimiento, Enable\_Negative, un contacto para la dirección negativa de movimiento, JogVelocity, la velocidad deseada de movimiento, MinJogDistance, el valor mínimo de distancia de recorrido permitido para cuando se desactive la dirección actual de movimiento, Acceleration, la aceleración deseada, deceleration, la desaceleración deseada.

En cuanto a las salidas tenemos Active, que indica cuando el bloque está en funcionamiento, y ErrorID, que mostrará la identificación de cualquier error que ocurra durante el funcionamiento del programa.

### **MC\_READACTUALPOSITION**

EL bloque MC\_READACTUALPOSITION lee la posición actual en la que se encuentra el eje que indiquemos.

A este bloque se le asigna una línea de habilitación además del Axis, el eje que se desea controlar y Enable para que haga lectura de la posición actual. En cuanto a las salidas tenemos ErrorID, para identificar cualquier error que se presente durante la ejecución del



programa, en esta salida asignaremos un registro donde se irá guardando continuamente la posición actual del eje.

### **MC\_MOVERELATIVE**

El bloque MC\_MOVERELATIVE es usado para dar movimientos a los ejes que representan los segundos, minutos y horas. Mediante un pulso en “Execute” este bloque realiza un incremento a la posición actual (solo se requiere de un pulso para realizar el movimiento y la distancia final es una suma relativa a la posición actual). A igual que los bloques anteriores a este bloque se le asigna todos los parámetros de entrada que nos muestra.

### **MC\_MOVEABSOLUTE**

El bloque MC\_MOVEABSOLUTE lo usaremos para realizar movimientos de posiciones precisos con respecto a un punto de referencia.

A este bloque le asignaremos una variable, una línea de habilitación, también asignamos el eje que se desea controlar y los demás parámetros como son execute, position, además de velocity, acceleration, desaceleración. En cuanto a las salidas en algunos casos usaremos la salida “Active” hacia una bobina.

## Diagrama de bloques

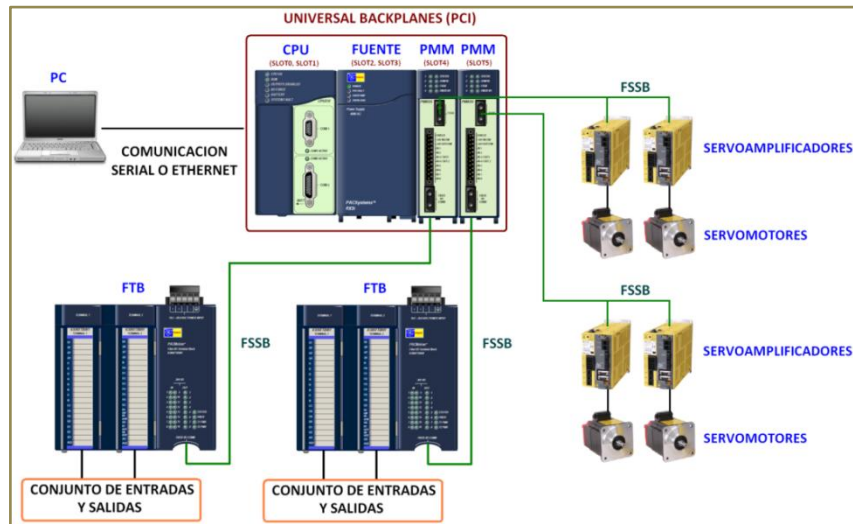
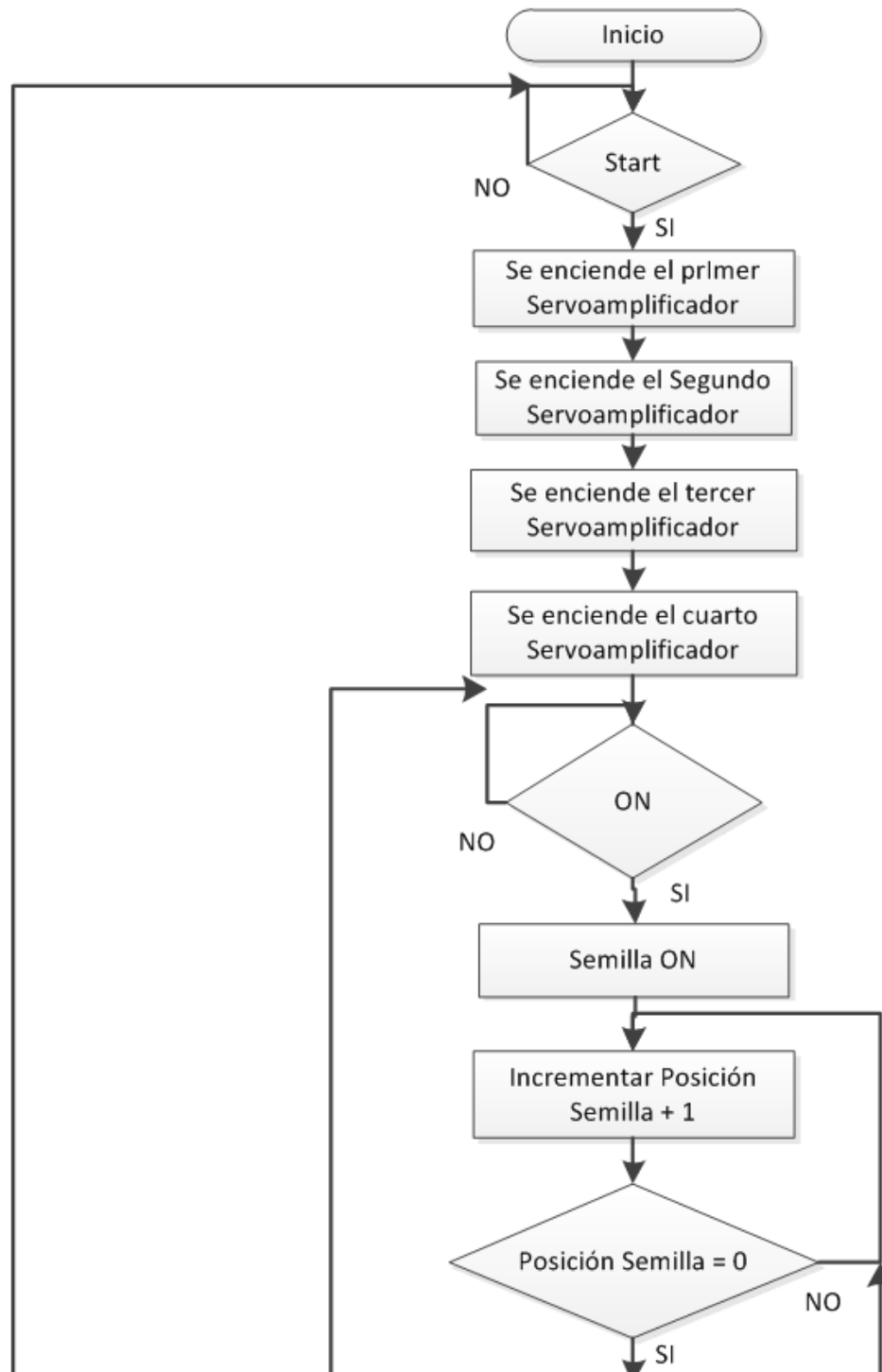


Figura 5.9 Diagrama de bloques PACMotion

## Diagrama de flujo



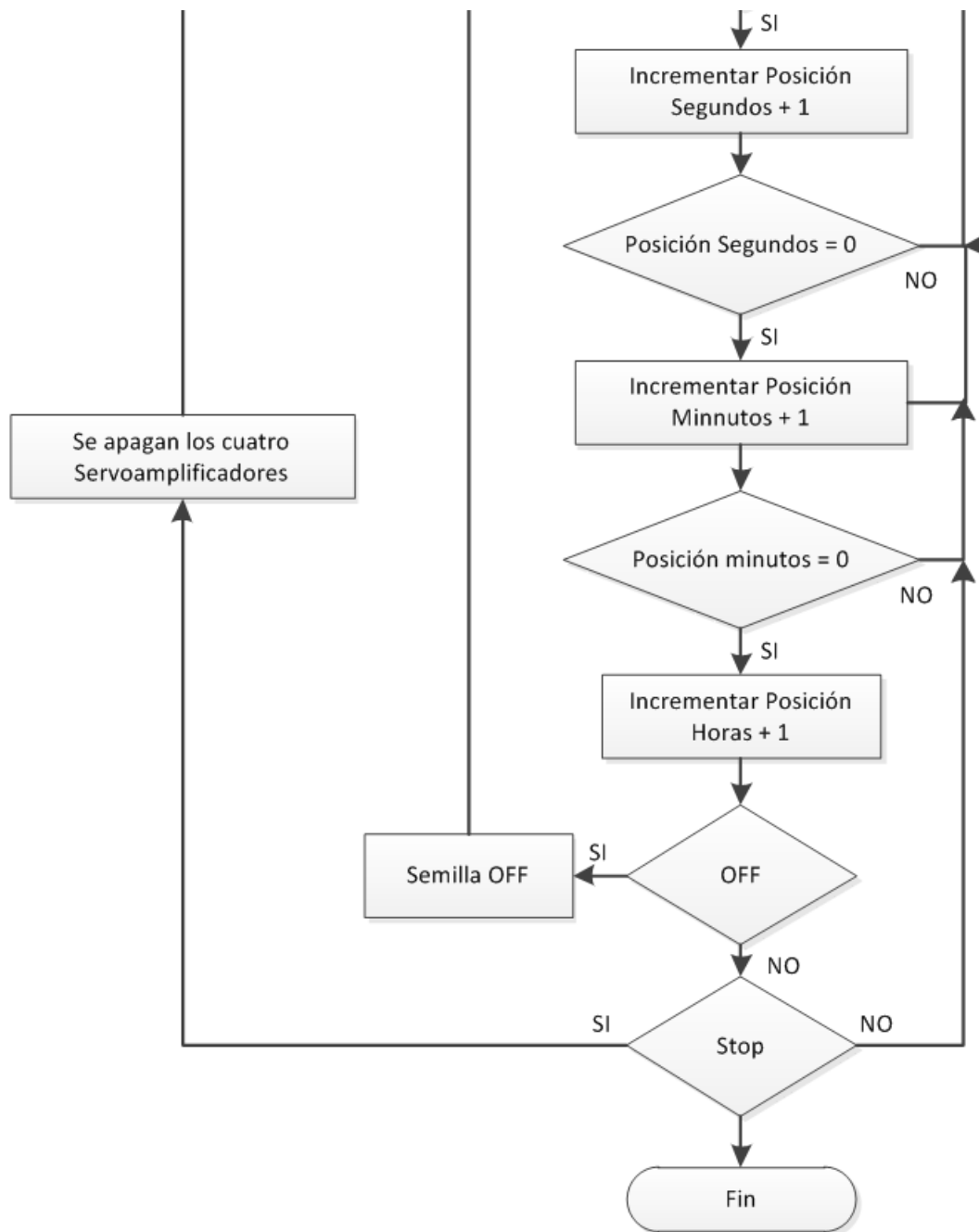


Diagrama 1. Control de los cuatro ejes de los servomotores

## PROGRAMACIÓN EN PROFICY MACHINE EDITION

La programación respectiva se encuentra en el anexo A.

## PROGRAMACIÓN EN INTOUCH

En el programa de InTouch se monitorean las posiciones de los ejes (%R10, %R20, %R30 y %R40) los cuales indican el tiempo.

Además se direccionan las bobinas auxiliares (%M11, %M12, %M13 y %M14) para tener control del encendido, apagado, marcha y paro del programa.

En la figura 5.12 se muestra un ejemplo del programa hecho en InTouch haciendo lectura de la posición de cada eje.

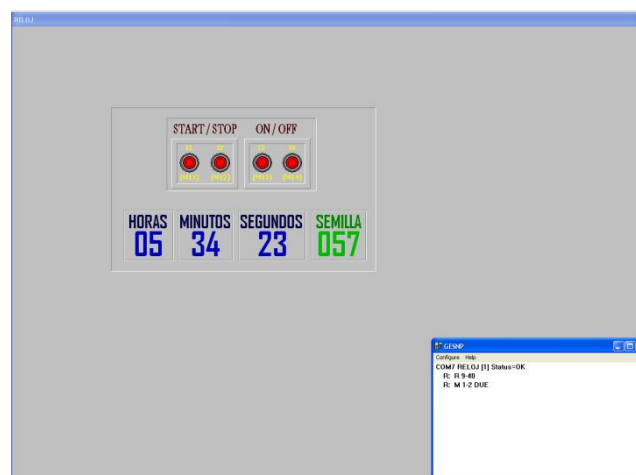


Figura 5.10 Comunicación del PACSystems Con InTouch

### **5.2.2 Configuración y programación de los cuatros ejes de los servomotores para la sincronización de los mismos.**

La sincronización de los cuatro ejes se llevará a cabo gracias al control de un solo eje por parte del programador, los restantes tres ejes dependerán de éste. El programa consiste en el control de la velocidad de un primer eje (eje maestro) mediante una señal analógica variada a voluntad por el usuario.

El segundo eje (eje esclavo 1) tendrá la mitad de la velocidad del eje maestro, el tercer eje (eje esclavo 2) tendrá la mitad de la velocidad del eje esclavo 1 y el cuarto eje (eje esclavo 3) tendrá la mitad de la velocidad del eje esclavo 2.

Para este caso los cuatro servomotores tendrán la misma configuración en cuanto a parámetros se refiere.

Como límites se especifica que el último eje (eje esclavo 3) tenga una velocidad máxima de 1 rev/seg, por ende el eje esclavo 2 tendrá una velocidad máxima de 2 rev/seg, el eje esclavo 1 una velocidad máxima de 4 rev/seg y el eje maestro una velocidad máxima de 8 rev/seg. Tal como se muestra en la siguiente tabla:

La velocidad máxima del Eje Esclavo 3 puede ser cambiada a conveniencia del usuario durante la programación. La velocidad del

Eje Maestro manipulada por el usuario desde una entrada analógica será establecida en porcentajes de velocidad, es así que la velocidad máxima adquirida por éste eje, 8 rev/seg, representa el 100%, de acuerdo a la entrada analógica variable de 0 a 10 voltios tenemos la siguiente tabla de correspondencias de valores mínimos y máximos para el eje maestro:

**Tabla XV. Rangos Mínimos y Máximos del Eje Maestro.**

<b>Mínimo</b>	<b>Máximo</b>
0 rev/seg	8 rev/seg
0 %	100%
0 Volt	10 Volt

De este modo no tenemos que preocuparnos por exceder el límite de velocidad con el potenciómetro.

Los datos de posición y velocidad podrán ser visualizados por medio del scada en una pantalla, además de tener la posibilidad de controlar el encendido y apagado de los servoamplificadores, el inicio y paro de la aplicación y el control de la velocidad del eje maestro por un slider o

una entrada de valor numérico. Para los casos de control de velocidad y sentido de giro esto se dispondrá dentro de la pantalla del scada de un switch que alternará los modos manual y remoto.

## **BLOQUES DE PROGRAMACIÓN IMPORTANTES PARA EL DESARROLLO DE LA APLICACIÓN**

Para la realización de esta práctica se requiere de un bloque en especial llamado MC\_GEARIN, además de los bloques para el encendido de los servoamplificadores MC\_POWER, la inicialización de la posiciones MC\_SETPOSITION, el paro de los servomotores MC\_STOP, la variación de la velocidad del eje maestro MC\_JOGAXIS y otros bloques básicos fuera de la librería PACMotion.

### **MC\_GEARIN**

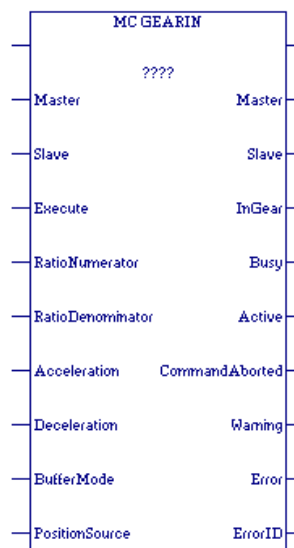
Este bloque se encarga de sincronizar la velocidad de un eje esclavo con respecto a la velocidad de un eje maestro.

Entre sus entradas se tiene, Master, donde se especifica el eje que trabajará como maestro, Slave, donde se especifica el eje que trabajará como esclavo, Execute, que recibe una señal de flanco positivo para iniciar la sincronización, RatioNumerator, que sirve para



especificar la dimensión del radio del eje maestro, RatioDenominator, que sirve para especificar la dimensión del radio del eje esclavo, Acceleration, donde se indica la aceleración que tendrá el eje esclavo para alcanzar la velocidad del eje maestro cuando esta aumenta, deceleration, que indica la desaceleración que tendrá el eje esclavo para alcanzar la velocidad del eje maestro cuando esta disminuya, BufferMode, que define el comportamiento del bloque, Position Source, que tiene dos opciones, ActualPosition, que sigue la posición indicada en el eje actualmente, CommandedPosition, que sigue la posición generada instantáneamente por el modulo PMM internamente.

RatioNumerator y RatioDenominator no son necesariamente los radios de los ejes sino más bien una relación de vueltas entre cada eje.



**Figura 5.11 MC\_GEARIN.**

## DESARROLLO DE LA APLICACIÓN

La aplicación comienza con el encendido de los servoamplificadores por parte del usuario mediante el uso de la entrada física I1, como pulsador, quien corresponde al FTB del módulo cuatro PMM, también mediante el uso de un pulsador desde scada que comanda el contacto de la dirección M101, esta orden energiza una bobina set de dirección M1 denominada "ON" la cual habilita el bloque MC\_POWER para el primer servoamplificador, una vez encendido el primer servo amplificador inmediatamente actúa un bloque MC\_SETPOSITION para fijar la posición actual como posición cero, cuando el primer eje ha sido encendido correctamente y establecido su posición cero se procede automáticamente con el encendido del segundo eje de la misma forma actuando sobre si un bloque para el encendido y uno

para fijar la posición cero, de esta manera se consigue el encendido de los cuatro ejes y el establecimiento de sus posiciones iniciales.

Una vez que se ha conseguido el encendido de los cuatro servoamplificadores se puede dar la orden del inicio de la aplicación mediante el uso de la entrada I3 como pulsador que comanda la entrada IN3 del FTB, o desde el scada con un botón que manipula el estado del contacto M103. Cuando se haya dado la orden del inicio de la aplicación se energiza una bobina set M2 denominada "START" la cual habilita el bloque MC\_JOGAXIS encargado del movimiento del eje maestro, inmediatamente cuando el bloque MC\_JOGAXIS se encuentre activado envía una señal hacia el primer bloque MC\_GEARIN quien realiza la sincronización entre el eje maestro y el eje esclavo 1, este bloque a su vez habilita el siguiente bloque MC\_GEARIN que enlaza la sincronización entre el eje esclavo 1 con el eje esclavo 2, que a su vez finalmente enlaza la sincronización entre el eje esclavo 2 y el eje esclavo 3.

Mientras el programa se encuentra en ejecución la única velocidad que puede ser manipulada es la velocidad del Eje Maestro ya sea desde la entrada analógica AI1 del FTB o desde el scada mediante un slider que manipula el valor del registro R3, así mismo se tiene el control del sentido de movimiento del eje maestro con un switch simulado por la entrada I5 del FTB o desde el Scada con un switch

que manipula el estado del contacto M105, y un switch o mando para elegir el modo desde donde se realizará el control de la velocidad el cual tiene asociado la entrada I6 del FTB o desde el scada manipulando el estado del contacto M106.

Para detener el programa se hace uso de la entrada I4 del FTB como pulsador de parada o desde el scada con un pulsador que manipula el estado del contacto M104, esto envía una señal que energiza una bobina set denominada "STOP" la cual deshabilita el bloque de movimiento MC\_JOGAXIS y luego de esto actúan los bloques MC\_STOP de cada eje perdiéndose la sincronización y parando a cada uno de ellos.

El único estado en el que se puede apagar los servoamplificadores es en el estado de STOP de los ejes, ya que si envía la orden de apagado mientras los ejes se encuentran en movimiento existe una sobrecarga de corriente en los mismo, esto envía una señal de error al CPU del sistema el cual deshabilita la acción de cualquier bloque de movimiento hasta que se corrija el error.

La señal de apagado de los servoamplificadores se realiza mediante la entrada I2 del FTB que simula un pulsador de “OFF” o desde el Scada con el uso de un pulsador que manipula el estado del contacto M102. Esta señal energiza una bobina reset denominada “ON” la cual desenergiza los cuatro bloques MC\_POWER de cada servoamplificador.

### **PROGRAMACIÓN EN PROFICY MACHINE EDITION**

La programación respectiva se encuentra en el anexo A.

### **PROGRAMACIÓN EN INTOUCH**

La aplicación en InTouch incluye dos pulsadores para controlar en encendido y apagado de los servoamplificadores, dos pulsadores para el inicio y paro del programa, un selector para elegir el sentido de giro del eje maestro, un selector para elegir el modo de manejo de la velocidad del eje maestro, un slider para variar la velocidad desde el Scada, en caso de elegir dicha opción con el “selector de Velocidad”, y cuatro indicadores donde se muestra la posición de cada uno de los eje en unidades de usuario.

La comunicación con el RX3i se realiza mediante el protocolo SNP por el puerto 1 del CPU con dirección 1.

La programación solo incluye comunicación de las variables del RX3i más no creación de scripts.

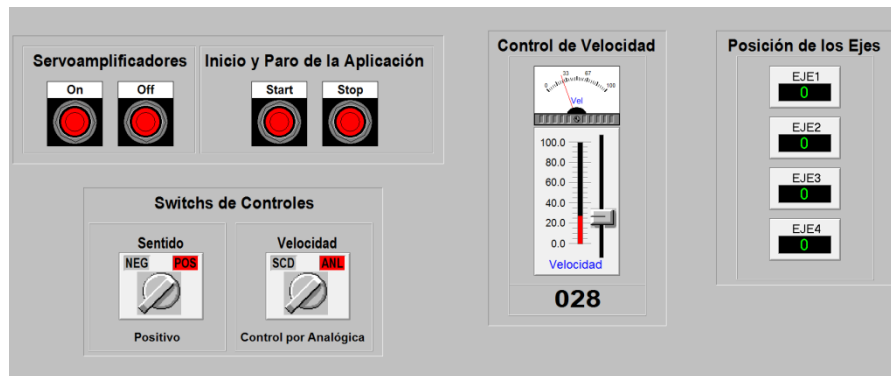


Figura 5.12 Esquema de las variables desde el Scada.

### 5.2.3 Simulación y programación para el direccionamiento de paneles solares.

Este ejercicio comprende la programación de dos ejes de los servomotores para que cada uno de ellos realice la función seguimiento del sol de acuerdo a las fechas del calendario y a las horas del día.

Vamos a realizar el control de dos ejes colocados en un equipo de paneles solares los cuales van a realizar el seguimiento del sol durante toda la época del año, vamos a manejar dos ejes de rotación el primero para controlar el movimiento del panel en un ángulo de 180 grados quien tomará la radiación solar durante el día y el segundo eje tendrá un movimiento angular de 47 grados en sentido norte a sur y

viceversa durante todo el año, esto con el fin de tomar la radiación solar durante las diferentes época del año.

Para llevar a cabo esta práctica además de la programación real del sistema de control la cual ha sido implementada con algunas herramientas que iremos viendo en el desarrollo de la práctica también se ha realizado una programación en paralelo para poder comprobar el funcionamiento del programa.

La programación de la lógica de posicionamiento ha sido gracias al reloj interno de nuestro controlador, de acuerdo a esto se ha realizado la programación en ladder para el movimiento de nuestros ejes.

Para esto tenemos que tomar en cuenta las unidades usuario que colocaremos en cada eje en cuestión, la configuración es exactamente igual a la que hemos venido desarrollando a lo largo de este trabajo.

## **BLOQUES UTILIZADOS**

A continuación una lista de los bloques de movimientos utilizados en la programación:

**MC\_POWER:** Utilizado para el encendido de los servoamplificadores.

**MC\_SETPOSITION:** Utilizado para establecer la posición cero de cada eje.

**MC\_MOVEABSOLUTE:** Usado para mover al eje indicado a una posición exacta.

**MC\_STOP:** Usado para detener los ejes o interferir y perder la sincronización.

**MC\_READACTUALPOSITION:** Usado para leer la posición actual de los ejes. Esto es leído en unidades de usuario.

**SVC\_REQ:** Es un bloque de servicio de respuesta el cual gracias a los parámetros que recibe en el FNC y PRM va a cumplir una determinada función. FNC puede ir una variable tipo INT o constante. La variable es la que identifica el servicio solicitado por ejemplo si deseo cambiar el fondo del modo de ventana de tareas y el valor del temporizador escribo en FNC el numero 5 o si deseo Restaurar el Watch dog Timer escribo el numero 8 e nuestro caso que deseamos obtener la hora del controlador vamos a escribir el numero 7. En el parámetro PRM va una variable de tipo INT o constante la cual definirá el tipo de acción que va a realizar nuestro bloque, en este caso escribimos 0 porque con este valor vamos a poder leer el día y la hora de nuestro controlador, si escribimos 1 vamos a poder escribir pero para nuestro caso solo necesitamos poder leer.



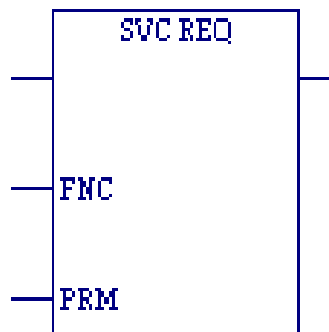


Figura 5.13 SVC REQ.

**ROL WORD:** La función de este bloque es rotar los bits de una palabra, el más significativo al menos significativo, esto lo hacemos con el fin de obtener los segundos, minutos, horas días meses y año este bloque junto con el bloque AndWord realizan ese trabajo.

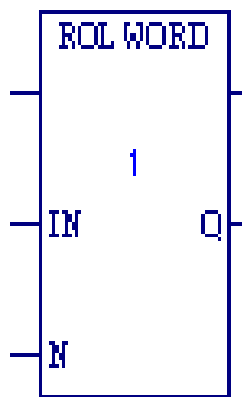


Figura5.14 Rol Word.

## MC\_MOVEABSOLUTE

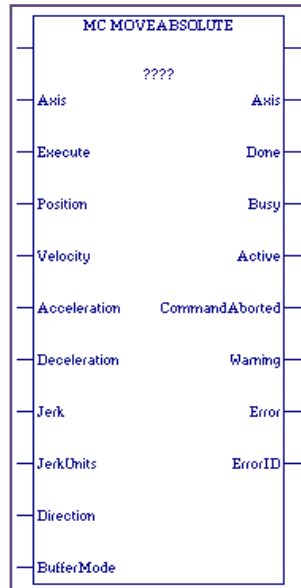
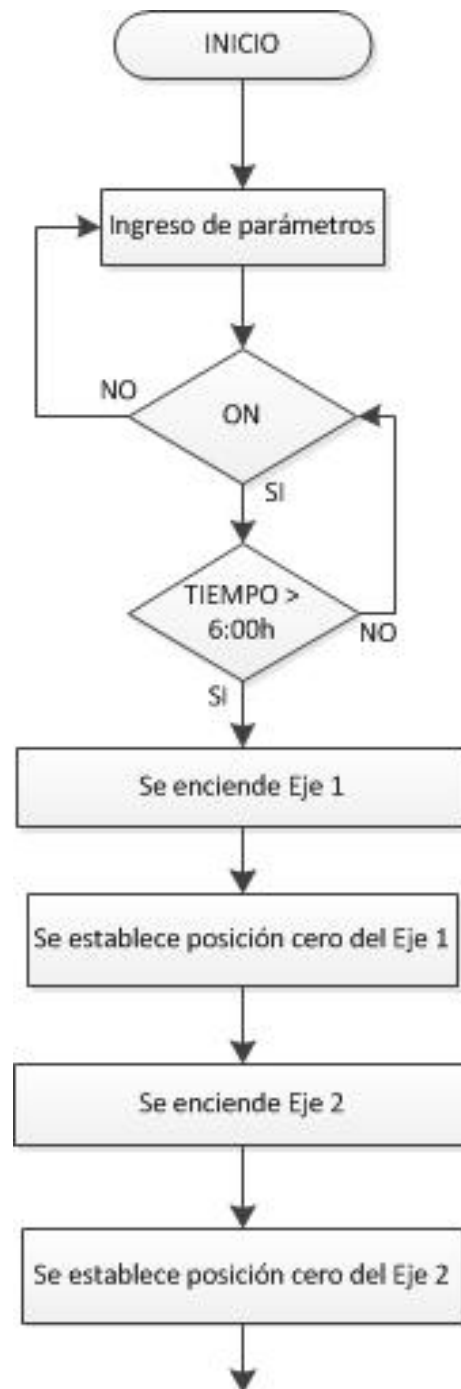


Figura5.15 Mc\_Moveabsolute

El bloque MC\_MOVEABSOLUTE lo usaremos para realizar movimientos de posiciones precisos con respecto a un punto de referencia.

A este bloque le asignaremos una variable, una línea de habilitación y todos los parámetros ya antes descritos. En cuanto a las salidas en algunos casos usaremos la salida "Active" hacia una bobina.

**DIAGRAMA DE FLUJO**

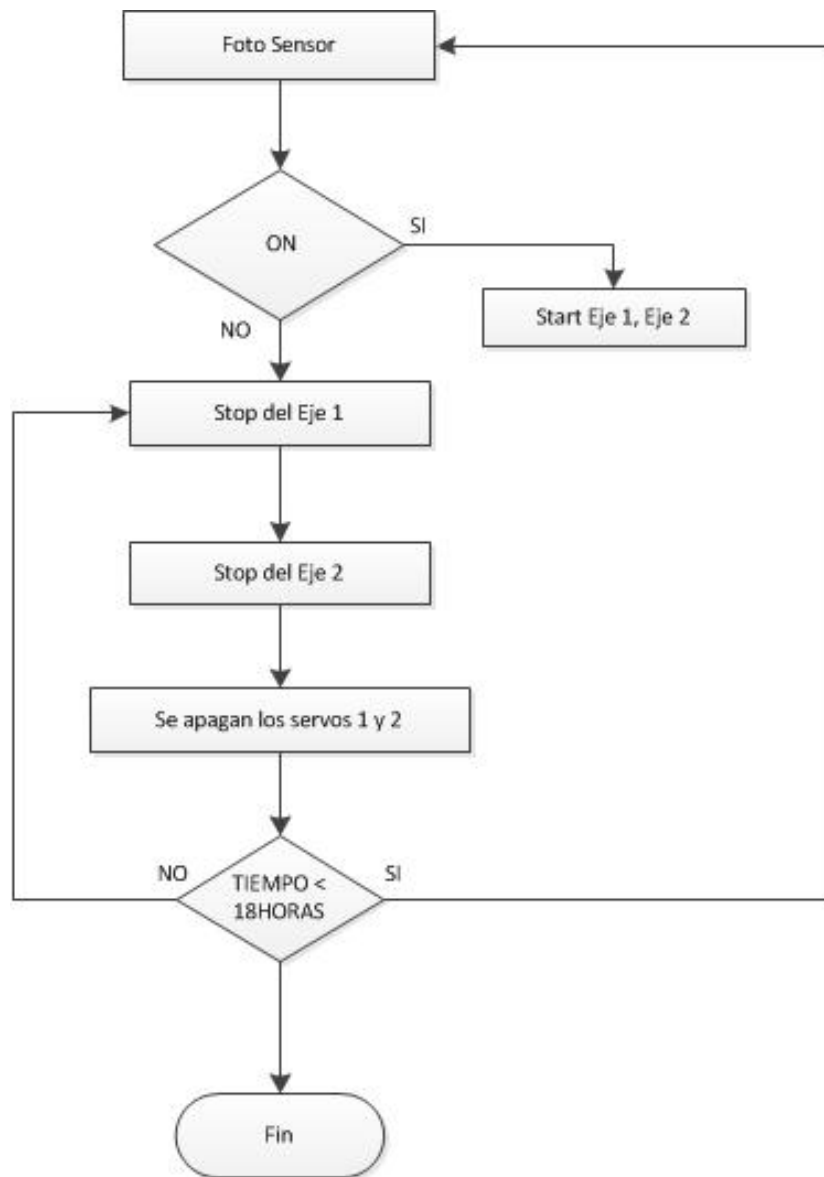


Diagrama 2. Control del movimiento de paneles solares.

## PROGRAMACIÓN EN PROFICY MACHINE EDITION

La programación respectiva se encuentra en el anexo A.

## PROGRAMACIÓN EN INTOUCH

En el programa de InTouch se monitorean las posiciones de los ejes los cuales indican la posición.

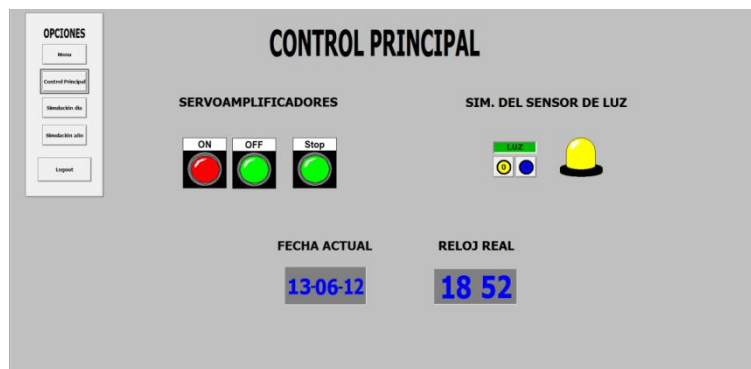


Figura 5.16 Control principal de los ejes



Figura 5.17 Scada paneles solares

#### **5.2.4 Simulación y programación para el proceso de llenado y sellado de envases.**

Este ejercicio está orientado a la sincronización de ejes para realizar el control de un sistema de llenado y sellado de envases. Se dispondrá del uso de los cuatro ejes disponibles físicamente dentro del sistema para realizar:

- El avance de una banda transportadora (eje 1).
- Llenado uniforme del envase mediante el uso de una boquilla móvil (eje 2).
- Bombeo del producto líquido con el que se llenará el envase (eje 3).
- Sellado de la tapa bajo presión (eje 4).

El proceso comienza con el avance de la banda por el que viajan los envases vacíos, todos separados por la misma distancia, este último punto es muy importante ya que la distancia a la que se encuentran separados cada uno de los envases debe ser exactamente la misma pues el eje encargado del avance de la banda siempre se desplazará a esta distancia la cual es ingresada con anticipación por el usuario.

Cada vez que la banda transportadora desplace un envase a la posición donde se encuentra la boquilla de llenado (controlada por el

eje 2), esta descenderá dentro del envase hasta una posición cercana a la base para luego ascender, mientras la boquilla asciende entra en sincronización con la bomba (controlada por el eje 3), de esta forma el envase es llenado uniformemente evitando el desperdicio de producto fuera de él, de igual forma el sellador de tapas (controlado por el eje 4) es sincronizado con la boquilla de tal forma que mientras la boquilla descende el sellador ejerce presión sobre la tapa, en un envase que ya paso por la etapa de llenado, y a medida que la boquilla asciende el sellador regresa a una posición de reposo sobre la tapa del envase.

El proceso dispone de un pulsador de encendido y un pulsador de apagado para el control del encendido de los servoamplificadores, además de un pulsador de marcha y un pulsador de paro para el control de inicio o detención del proceso. El control que realiza cada uno de estos pulsadores se lo puede ejercer desde el Scada del proceso.

Las dimensiones tales como la altura del envase, la separación entre cada uno, el área interna, el volumen de producto a ingresar, las posiciones de descenso y ascenso de la boquilla, las posiciones de presión y reposo del sellador, así como la velocidad y aceleración de los ejes 1 y 2 son ingresadas previamente desde el Scada donde además de poder observar la animación del proceso se mostrarán

valores de lectura del estado de cada eje, las posiciones de cada uno y un gráfico con el histórico de movimiento.

## CONFIGURACIÓN DE PARAMETROS

Los parámetros a configurar para esta práctica son parecidos a los de las prácticas anteriores con la única diferencia de que todos los ejes son configurados con 100 Unidades de Usuario por cada revolución, además de esto el parámetro “Axis PositioningMode” se cambio al modo “Linear” para permitir el incremento de las unidades de usuario a más de 100 tal como se muestra en la siguiente figura:

InfoViewer (0.4) IC695PMM335	
Settings   FPI I/O   FTB Inputs   FTB Outputs   I/O Interrupts   Axis 1	
Parameters	
Stop Axis on FTB Error	Enabled
<i>Position Feedback Source</i>	Motor Encoder
<i>Axis Positioning Mode</i>	Linear
<i>Motor Encoder Mode</i>	Incremental
Motor Encoder User Units	100.0
Motor Encoder Counts	65536
Motor Encoder Counts Per Motor Revolution	65536
Motor Encoder Maximum Positive RPM Limit	8191
Motor Encoder Maximum Negative RPM Limit	8191
<i>External Device</i>	None
Over Travel Limit Switch	Disabled
Axis Direction	Normal
<i>Software End of Travel</i>	Disabled

Figura 5.18 Configuración de ejes.



La configuración del hardware incluye lo siguiente:

- Slot 0 con el CPU
- Slot 2 con la Fuente de Poder
- Slot 4 módulo PMM
- Slot 5 módulo PMM

Se recomienda usar SNP como protocolo de comunicación en el puerto 1 para la comunicación con el Scada.

## **BLOQUES UTILIZADOS**

A continuación una lista de los bloques de movimientos utilizados en la programación:

MC\_POWER: Utilizado para el encendido de los servoamplificadores

MC\_SETPOSITION: Utilizado para establecer la posición cero de cada eje.

MC\_MOVEABSOLUTE: Usado para mover al eje indicado a una posición exacta.

MC\_MOVERELATIVE: Usado para mover al eje indicado a una posición adicional a la actual.

MC\_GEARINPOS: Utilizado para realizar la sincronización de velocidad y posición entre los ejes. Se utilizan dos de estos bloques,

uno para sincronizar la boquilla con la bomba y uno para sincronizar la boquilla con el sellador.

MC\_STOP: Usado para detener los ejes o interferir y perder la sincronización.

MC\_READACTUALPOSITION: Usado para leer la posición actual de los ejes. Esto es leído en unidades de usuario.

### **MC\_GEARINPOS**

La base para el desarrollo de esta práctica esta en el bloque MCGEARINPOS que es el encargado de sincronizar el eje 2 (que controla el movimiento de la boquilla de llenado) con los ejes 3 y 4 (que controlan el bombeo de producto y sellado de tapas respectivamente).

Lo característico de este bloque es que no solo realiza la sincronización de los ejes con respecto a la velocidad sino que además con respecto a la posición donde se encuentre el eje maestro. Es por esta causa que es de vital importancia para el control de llenado con la boquilla móvil.

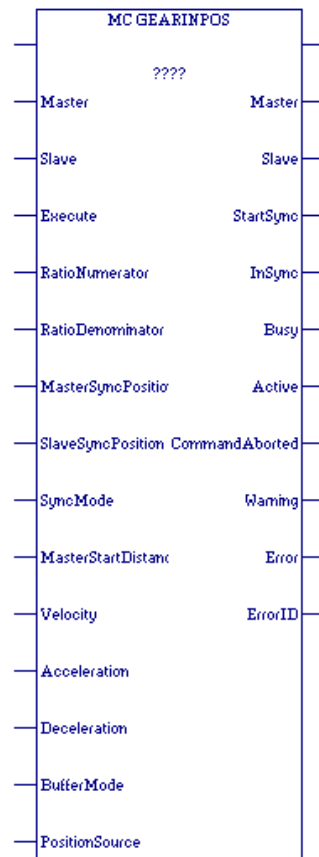


Figura 5.19 MC\_GEARINPOS.

El bloque requiere de una variable donde se almacenarán los datos de entradas y salidas del mismo.

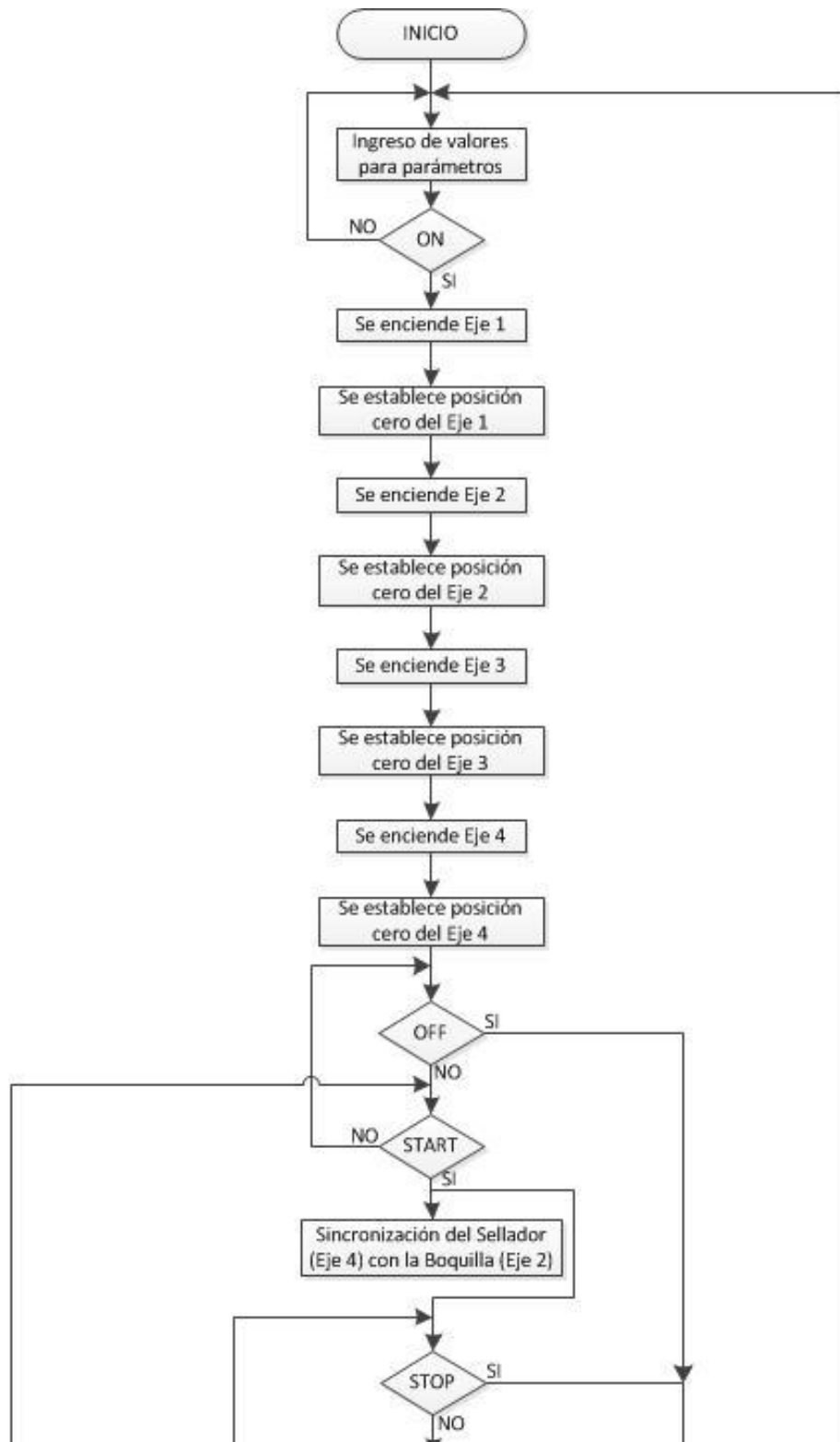
En cuanto a entradas dispone tenemos:

- Una entrada de habilitación que es la primera línea a la vista.
- Master: donde se especifica el eje que actuará como eje maestro.
- Slave: donde se especifica el eje que actuará como eje esclavo.

- **Execute:** que recibe una señal de flanco de subida para ejecutar la acción especificada en el bloque.
- **RatioNumerator:** donde se representa el radio del eje maestro.
- **RatioDenominator:** donde se especifica el radio del eje esclavo, estos radios son solo representaciones de la proporción en cuanto a revoluciones sobre sus ejes.
- **MasterSyncPosition:** especifica la posición del eje maestro de sincronización con el eje esclavo, es decir la posición donde entrará en sincronía con el eje esclavo.
- **SlaveSyncPosition:** Especifica la posición del eje esclavo de sincronización con el eje maestro, es decir la posición donde entrará en sincronía con el eje maestro.
- **SyncMode:** dispone de dos opciones, **BackupAllowed**, que regresa al eje esclavo automáticamente a su posición inicial antes del sincronismo cuando se pierde la sincronización con el eje maestro y **NoBackupAllowed**, que deja al eje esclavo en la última posición donde se perdió la sincronización con el eje maestro.
- **MasterStartDistance:** aquí se especifica una distancia fija que el eje maestro cubre durante la rampa de sincronización con el eje esclavo.

- **Velocity:** especifica la velocidad máxima permitida para que el eje esclavo alcance la posición de sincronismo con el eje maestro.
- **Acceleration:** especifica la aceleración máxima permitida para que el eje esclavo alcance la velocidad para llegar a la posición de sincronismo con el eje maestro.
- **Deceleration:** especifica la desaceleración máxima permitida para alcanzar la posición de sincronismo.
- **BufferMode:** especifica el comportamiento del bloque.
- **PositionSource:** dos opciones, `ActualPosition`, que establece como posiciones de sincronización las posiciones actuales donde se encuentran los ejes cuando se ejecuta el bloque, y `CommandedPosition`, que establece como posiciones de sincronización las posiciones definidas en los parámetros `MasterSyncPosition` y `SlaveSyncPosition`.

En cuanto a las salidas las más relevantes son: `StartSync`, que se activa cuando los ejes han comenzado la rampa de sincronización para alcanzar la posición de sincronismo, `InSync`, que se activa cuando los ejes se encuentran sincronizados, `Busy`, indica que el bloque está ocupado y aun no termina de cumplir su acción, `Active`, que se activa cuando el bloque esta activado y en correcto funcionamiento.

**Diagrama de flujo**

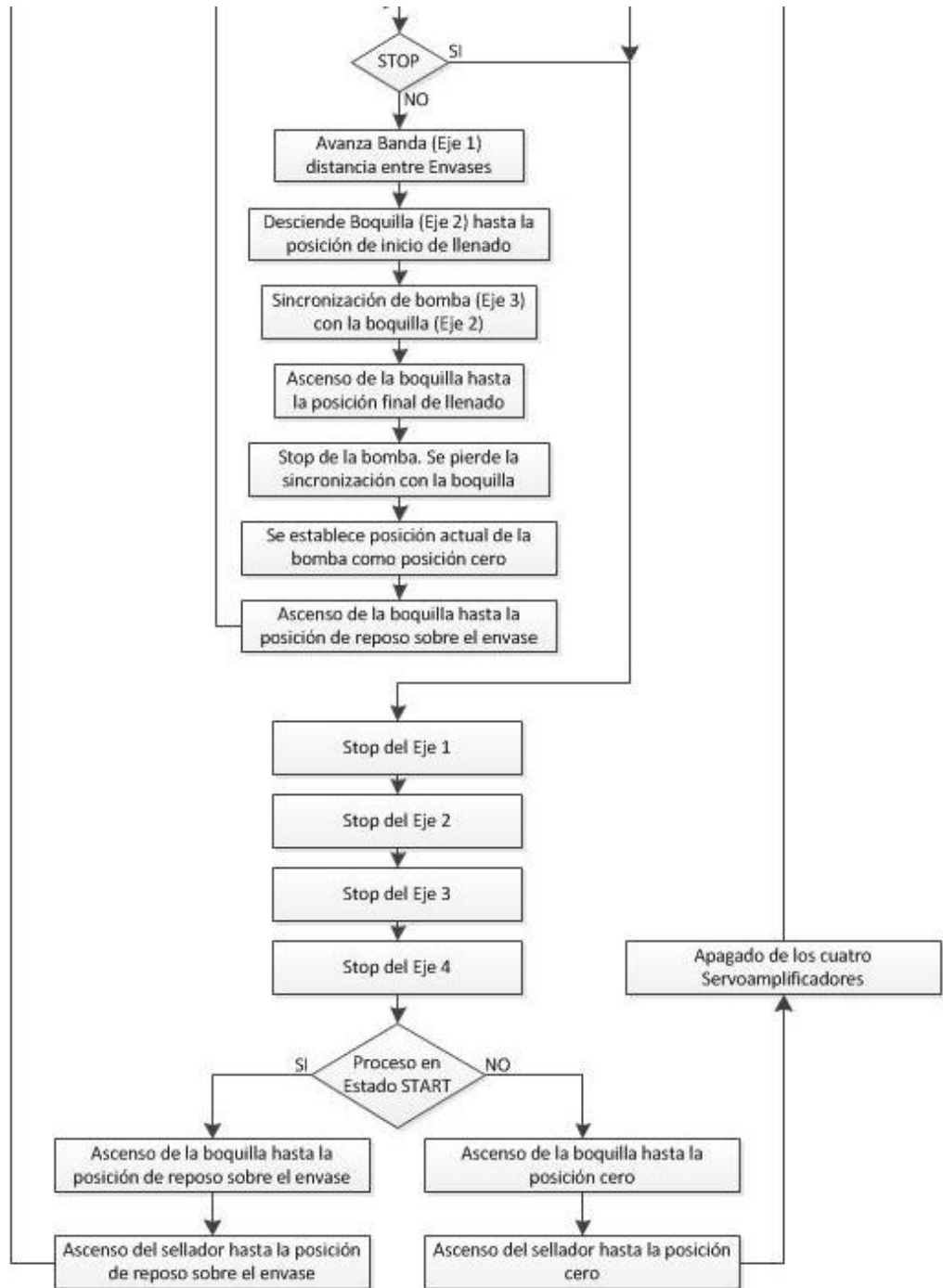


Diagrama 3. Control de la sincronización del proceso de llenado y sellado.

## **PROGRAMACIÓN EN PROFICY MACHINE EDITION**

La programación respectiva se encuentra en el anexo A.

## **PROGRAMACIÓN EN INTOUCH**

La aplicación en InTouch incluye tres ventanas, una llamada “SETEO” donde se escriben los parámetros previos al inicio del proceso, una llamada “PROCESO” donde se puede observar el proceso en tiempo real, y una última llamada “HISTÓRICOS” donde se presentan gráficos históricos de las señales de movimiento de los servomotores. En la ventana llamada “SETEO” se ingresan los parámetros, para establecer la distancia recorrida o volumen ingresado por revolución de cada eje.

- Centímetros por revolución del eje 1
- Centímetros por revolución del eje 2
- Centímetros cúbicos por revolución del eje 3
- Centímetros por revolución del eje 4

Límites de velocidad, aceleración y desaceleración

- Velocidad eje 1
- Aceleración y Desaceleración eje 1
- Velocidad eje 2
- Aceleración y Desaceleración eje 2



## Especificaciones de envases

- Distancia entre envases
- Posición de descenso de la boquilla
- Volumen ingresar
- Área interna del envase
- Altura del envase
- Distancia de la boquilla sobre el envase

## Sellado del envase

- Posición de presión para sellado
- Posición de reposo del sellador

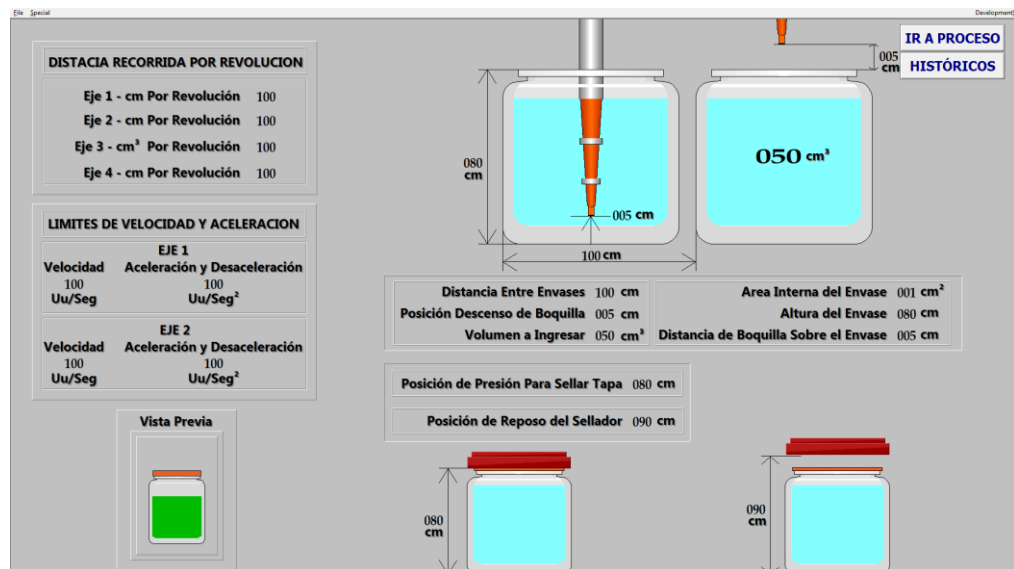


Figura 5.20 Ventana SETEO del Scada.

En la ventana llamada “PROCESO” se tiene el control del encendido y apagado de los servoamplificadores, el inicio y paro del proceso, luces indicadoras del estado de los servoamplificadores, lecturas de las posiciones actuales de cada eje tanto en unidades de usuario como en centímetros, además de la simulación del proceso de acuerdo a la lectura de las posiciones de los ejes.

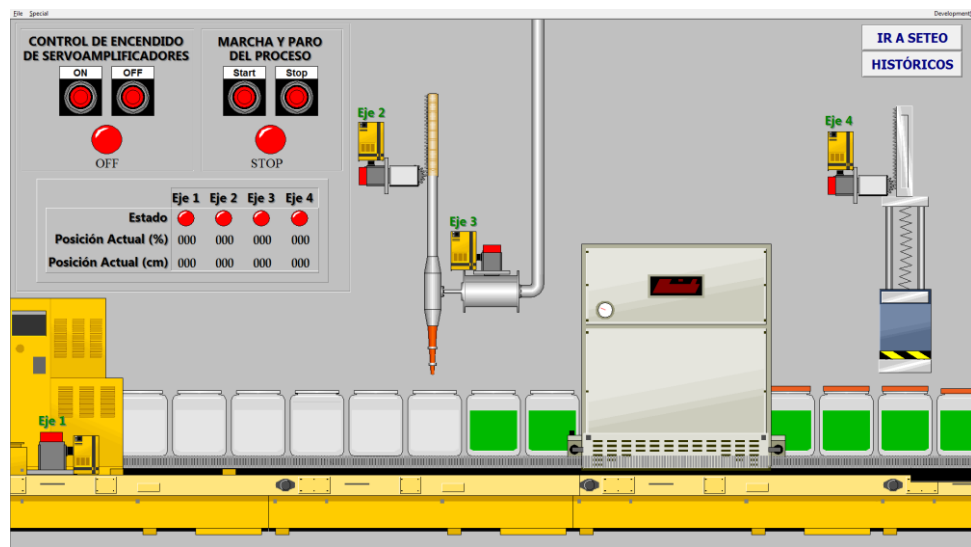


Figura 5.21 Ventana PROCESO del Scada.

En la ventana llamada “HISTÓRICOS” se puede observar las lecturas de las posiciones en un gráfico histórico. Cada eje posee un grafico donde se observa la posición a lo largo del tiempo tanto en centímetros como en unidades de usuario, y un gráfico extra para las posiciones de los cuatro ejes.

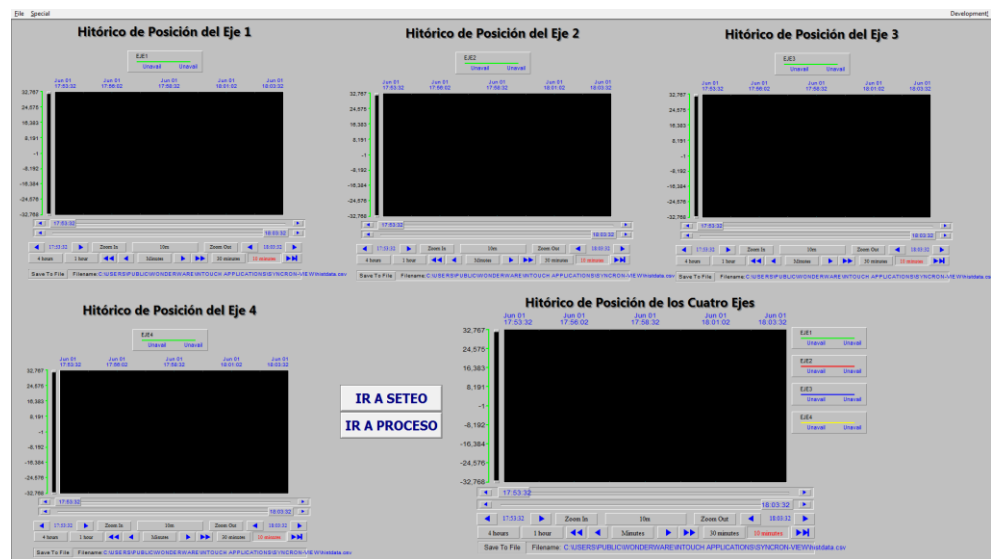


Figura 5.22 Ventana HISTÓRICOS del Scada.

La comunicación con el RX3i se realiza mediante el protocolo SNP por el puerto 1 del CPU con dirección 1.

### **5.2.5 Simulación y programación de un brazo de robótico en dos coordenadas.**

En esta práctica vamos a realizar el control de los dos ejes del uno de los Trainer los cuales van a simular el movimiento en dos ejes (x, y) para lograr el posicionamiento de un brazo mecánico, el cual cumplirá la función de mover objetos desde un punto en el eje hasta otro o cumplirá la función de seguir una trayectoria en ambos casos por un ciclo indeterminado o definido por el usuario, para realizar esta tarea la programación desde Proficy no cambiará mucho en relación a las anteriores prácticas, nuestro trabajo se centra en la herramienta de crear , guardar, y leer archivos desde el scada para lograr la migración de los datos de estos archivos en tiempo real al Proficy, una vez realizado esto estaremos en capacidad de realizar la programación de movimiento de manera más sencilla.

### **CONFIGURACIÓN DE PARAMETROS**

Tomar en cuenta las unidades usuario que colocaremos en cada eje en cuestión, la demás configuración es exactamente la misma que hemos venido desarrollando a lo largo de este trabajo.

## **BLOQUES UTILIZADOS**

A continuación una lista de los bloques de movimientos utilizados en la programación:

MC\_POWER: Utilizado para el encendido de los servoamplificadores.

MC\_SETPOSITION: Utilizado para establecer la posición cero de cada eje.

MC\_MOVEABSOLUTE: Usado para mover al eje indicado a una posición exacta.

MC\_STOP: Usado para detener los ejes o interferir y perder la sincronización.

MC\_READACTUALPOSITION: Usado para leer la posición actual de los ejes. Esto es leído en unidades de usuario.

## **ESCRITURA Y LECTURA DE DATOS EN ARCHIVO .TXT**

El control de los ángulos para los servomotores depende en gran parte de una pequeña base de datos creada desde el scada. Esta base de datos son en sí un conjunto de coordenadas de movimientos seleccionadas en la pantalla de control bajo las cuales el programa realiza una serie de cálculos para encontrar los ángulos equivalentes a los que cada eje debe de moverse. En el Scada se utiliza una función especial la cual muestra la posición en pixeles del objeto sobre el cual se posiciona el mouse. Esta función es \$ObjHor para la lectura de la posición horizontal y \$ObjVer para la lectura de

la posición vertical. Utilizando estas funciones y realizando cálculos para definir los límites de las coordenadas dentro del rango definido (0,0) como mínimo y (9,9) como máximo se procede a seleccionar las coordenadas que seguirá el brazo.

Dentro del Scada se tienen las opciones de encendido y apagado de los servoamplificadores, el inicio y paro del programa, dos opciones de selección de puntos, dos opciones de movimientos, un switch para inicializar valores y borrar datos guardados, un mensaje para indicar la habilitación o deshabilitación de la escritura, y el área de selección de coordenadas.

Dentro del área de coordenadas se posee cien posibles opciones de movimientos, con indicadores de la posición que se está seleccionando y el número de datos tomados.

Es muy importante definir la posición del área de trabajo haciendo doble clic en el extremo superior izquierdo y en el extremo inferior derecho, esto es para que el programa hecho dentro del Scada sitúe los límites del área de selección de coordenadas para que se fijen los límites dentro del rango establecido. Esto se lo realiza antes de la captura de datos y debe hacerse con el switch de Borrar Archivos activado, luego se eligen las opciones de movimientos, al menos dos segundos después se desactiva el switch y se procede a la captura de datos.

Los datos son guardados en un archivo .txt que se crea automáticamente en una dirección especificada dentro del disco duro.

La función utilizada para la escritura de datos se denomina "FileWriteMessage", esta función dentro de cualquier programación Script en el Scada es capaz de guardar los datos especificados en una memoria de tipo "message". La función posee los siguientes parámetros de entrada:

**FileWriteMessage (filename, offset, messagetag, linefeed).**

Donde:

**Filename:** Es la dirección donde se encuentra el archivo donde se almacenaran las coordenadas. Por ejemplo "c:\Data\File.txt" donde el archivo tiene por nombre File.txt y no es necesario crearlo. La dirección "c:\Data\" debe existir dentro del disco duro para que el archivo pueda ser creado.

**Offset:** Localización en bytes en el archive donde se empezara a escribir. Se coloca -1 para empezar a escribir al final del último dato dentro del archivo.

**Messagetag:** Es el dato que se va escribir dentro del archivo. Este dato debe ser del tipo "message".

**Linefeed:** Especifica si debe escribir un salto de línea después de cada escritura de dato, para ello se coloca 1. Si se desea escribir datos en una sola línea se coloca 0.

Para nuestro caso en la escritura de datos horizontales y verticales tenemos las funciones como se muestran en las siguientes líneas:

```
FileWriteMessage("C:\DATOS\FileH.txt",-1,HT,0),
```

```
FileWriteMessage("C:\DATOS\FileV.txt",-1,VT,0),
```

Para la lectura de los datos se utiliza una función similar llamada "FileReadMessage" la cual lee los datos de un archivo especificado establecido dentro de los parámetros ingresados.

La función tiene los siguientes parámetros de entrada:

**FileReadMessage** (filename, offset, messagetag, charstoread)

Donde:

**Filename:** Es la dirección donde se encuentra el archivo de donde se van a leer los datos.

**Offset:** localización en bytes del archivo donde se empezará a leer.

**Message tag:** Es la variable donde se va a almacenar el dato leído desde el archivo. Esta variable debe ser del tipo "message".

**Charstoread:** Número de bytes que se van a leer a partir del primer dato.



Para nuestro caso en la lectura de datos tanto horizontales como verticales tenemos las funciones como se muestra a continuación:

```
FileReadMessage ("C:\DATOS\FileH.txt", DATO, LEC_TH, 1),
```

```
FileReadMessage ("C:\DATOS\FileV.txt", DATO, LEC_TV, 1),
```

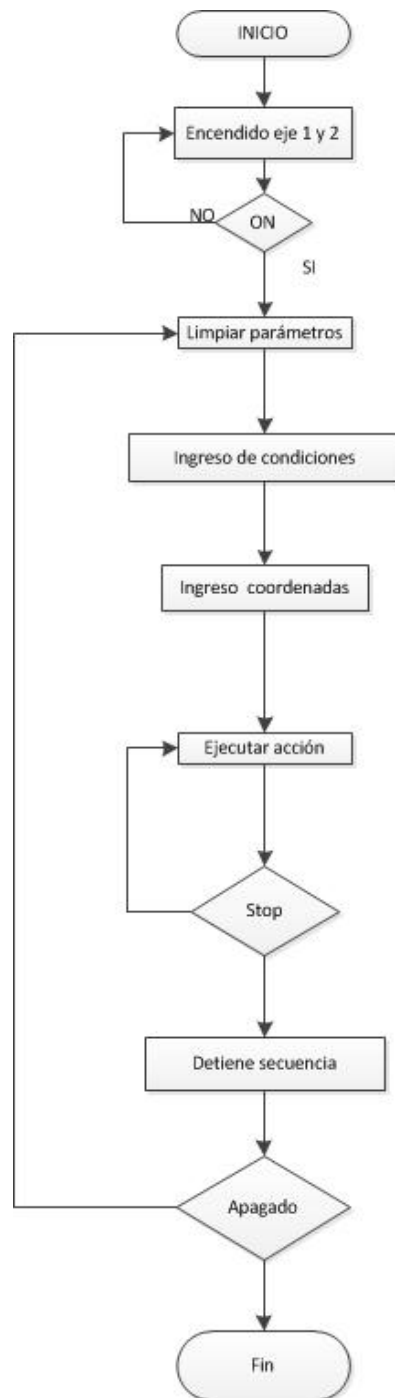
**DIAGRAMA DE FLUJO**

Diagrama 4. Control del movimiento en dos ejes de un brazo robótico.

## PROGRAMACION EN PROFICY MACHINE EDITION

La programación respectiva se encuentra en el anexo A.

## PROGRAMACION EN INTOUCH

En el programa de InTouch se monitorean las posiciones de los ejes los cuales indican la posición.

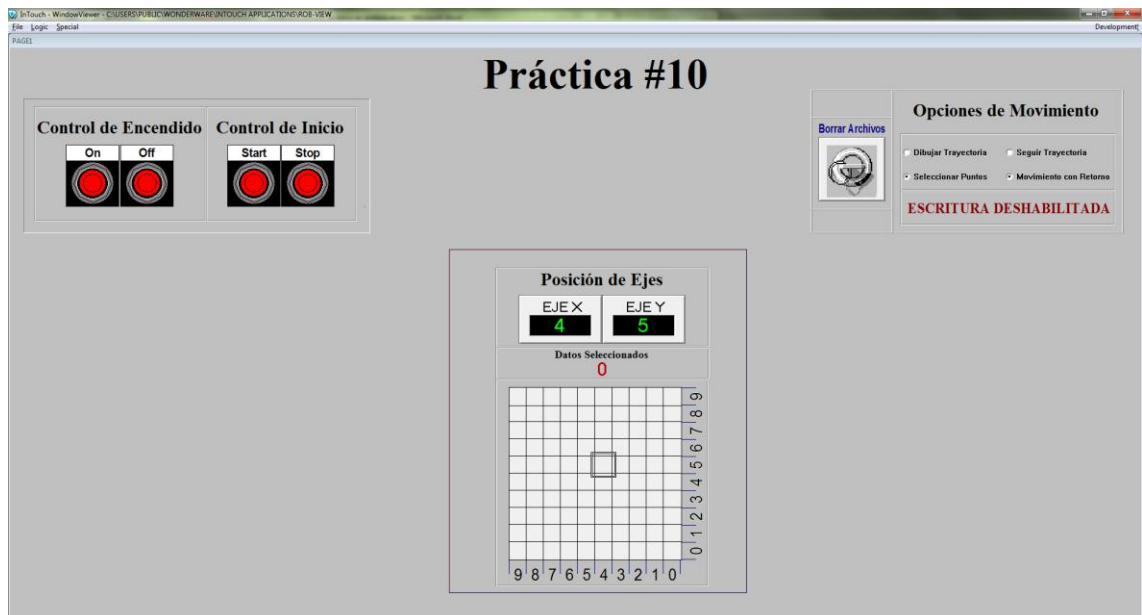


Figura 5.23 Control de posición de un brazo robótico en dos ejes

### **5.3 Práctica No 6.**

#### **Programación de un reloj utilizando los cuatro servomotores del sistema PacMotion.**

##### **Objetivos:**

- Configurar correctamente los parámetros de cada uno de los cuatro servomotores.
- Aprender el uso correcto de los bloques MC\_SETPOSITION para fijar una posición determinada, MC\_MOVERELATIVE y MC\_MOVEABSOLUTE para movimiento de los ejes.
- Relacionarse con la definición de las Unidades de Usuario “UserUnits” (Uu) aplicadas para cada uno de los cuatro ejes.

##### **Enunciado de la práctica.**

Esta práctica comprende la programación de los cuatro servomotores para que cada uno de ellos realice la función de marcar el tiempo como un reloj de manecillas.

## **5.4 Práctica No 7.**

### **Simulación y programación para la sincronización de los servomotores.**

#### **Objetivos:**

- Comprender el modo de sincronización en el sistema PACMotion.
- Sincronización de los ejes de los servomotores por medio del bloque MC\_GEARIN.
- Configurar correctamente los parámetros de los cuatro servomotores.

#### **Enunciado de la práctica.**

Esta práctica está orientada a la sincronización de los cuatro ejes, por ende se llevará a cabo el control de un solo eje por parte del programador, los restantes tres ejes dependerán de éste.

## **5.5 Práctica No 8.**

Simulación y programación para el proceso de llenado y sellado de envases

### **Objetivos:**

- Configurar correctamente los parámetros de los cuatro servomotores.
- Introducción al manejo del bloque MCGEARINPOS para la realización de la sincronización entre ejes.
- Realizar un programa de aplicación en la industria básico para el llenado y sellado de envases.

### **Enunciado de la práctica**

Esta práctica está orientada a la sincronización de ejes para realizar el control de un sistema de llenado y sellado de envases. Se dispondrá del uso de los cuatro ejes disponibles en el equipo.

## **5.6 Práctica No 9**

Programación del control de direccionamiento de paneles solares.

### **Objetivos:**

- Configurar correctamente los parámetros de los servomotores.
- Aprender el uso correcto de los bloques MC\_MOVEABSOLUTE para movimiento de los ejes.
- Relacionarse con la definición de las Unidades de Usuario “UserUnits” (Uu) aplicadas para cada uno de los cuatro ejes.

### **Enunciado de la práctica.**

Esta práctica comprende la programación de dos ejes de los servomotores para que cada uno de ellos realice la función seguimiento del sol de acuerdo a las fechas del calendario y a las horas del día.

## **5.7 Práctica No 10**

Programación del control de un brazo robótico en dos coordenadas.

### **Objetivos:**

- Programación y control del funcionamiento de un brazo robótico sobre dos coordenadas.
- Configurar correctamente los parámetros de los servomotores.
- Programación y control en Scada para PACMotion.

### **Enunciado de la práctica**

Esta práctica comprende la programación y control del funcionamiento de un brazo robótico sobre dos coordenadas desde el Scada para ello realizaremos la configuración respectiva en cada eje y a la vez la programación en Scada para realizar los movimientos deseados de los ejes.



## **CONCLUSIONES Y RECOMENDACIONES**

Al culminar la implementación del sistema de control “PACMotion” en la automatización de procesos logramos:

1. Obtener un mayor conocimiento sobre los sistemas de automatización industrial además de generar mayores destrezas en el uso de los mismos.
2. Gracias a la implementación de este sistema logramos conocer nuevas herramientas de automatización, los beneficios que ellas nos brindan y los alcances que tienen en el control y la automatización de procesos.
3. Cada práctica ha sido diseñada para un estudio didáctico del PACMotion de tal manera que el alumno logre adquirir los conocimientos para la instalación, configuración y programación del sistema.
4. Con este proyecto hemos logrado incorporar al estudio de la materia de automatización industrial los sistemas de control de movimiento en

los procesos de producción utilizando PACMOTION de la empresa General Electric.

5. Se recomienda que para futuros proyectos se agreguen los módulos que hacen falta para tener completo el equipo PACSystems.
6. Se recomienda la adquisición de nuevos equipos para la facilidad de los estudiantes en el momento de realizar el debido entrenamiento pues el laboratorio solo cuenta con dos equipos de entrenamiento y un solo controlador.
7. Se recomienda hacer las indicaciones pertinentes a la seguridad en el momento de la ejecución de las prácticas.
8. Se recomienda una lectura clara y comprensible de las prácticas antes de entrar a realizar algún tipo de instalación, configuración o programación del equipo.

## BIBLIOGRAFIA

[1]General Electric, PACMotion Multi-Axis Motion Controller User's Manual, GFK-2448B, Enero 2011

[2]General Electric, PACSystems™ RX3i IC695CPU310 Central Processing Unit, GFK-2329Z, Junio 2010

[3]General Electric, "PACSystems™ RX3i IC695CHS012 IC695CHS016 Universal Backplanes", GFK-2554, Enero 2010

[4]General Electric, PACMotion™ RX3i IC695FTB001 GFK-2467 Fiber I/O Terminal Block, GFK-2467, Noviembre 2008

[5]General Electric, PACMotion™ RX3i IC695PMM335-AB Multi-axis Motion Controller, GFK-2449C, Marzo 2009

[6]General Electric, Servo Amplifier  $\beta$ i Series Descriptions Manual,  
GFZ-65322EN/01, Agosto 2003