

UNIVERSIDAD DE PIURA

FACULTAD DE INGENIERIA



**“Automatización del encendido de los transmisores principales del Radio
Observatorio de Jicamarca”**

Tesis para optar el título de Ingeniero Mecánico – Eléctrico

**Juan Carlos Espinoza Guerra
Asesor: Dr. Ing. Justo Oqueli Cabredo**

Piura, abril 2006

A mis queridos padres Juan e Inés por
todo el esfuerzo realizado y a mis hermanos

RESUMEN

El presente trabajo de tesis corresponde a la ejecución de la segunda etapa del proyecto de automatización de los transmisores principales del Radio Observatorio de Jicamarca. Esta segunda etapa implica la automatización y control de los procesos involucrados con el encendido de las etapas de potencia de los transmisores principales (etapa de salida *PA* y etapa previa *Driver*), como tarea adicional se requiere también el monitoreo de las corrientes de operación de ambas etapas.

Para la ejecución de la mencionada segunda etapa se ha seguido con la misma lógica empleada en la primera etapa del proyecto, esto es utilizar el PLC y el sistema SCADA existentes para controlar y supervisar respectivamente los nuevos procesos a automatizar. Además de diseñar y construir una estación remota, que cuenta con un microcontrolador PIC, como elemento principal, la cual se encarga del control de la tensión de pantalla de ambas etapas del transmisor. Dicha estación remota junto con los módulos ADC, utilizados para el monitoreo de las corrientes, utilizan una red de comunicaciones RS485, protocolo Modbus para intercambiar información con el PLC.

Al finalizar este proyecto de tesis se desarrolló un prototipo del nuevo sistema de control el cual se ha implementado en el transmisor N° 3 del ROJ.

INDICE GENERAL

INTRODUCCIÓN	1
CAPÍTULO 1: ESCRIPCIÓN GENERAL DEL SISTEMA PROPUESTO	3
1.1 INTRODUCCIÓN	3
1.1.1 EL RADAR DE JICAMARCA.	3
1.1.1.1 La antena.	4
1.1.1.2 Transmisores y receptores	4
1.1.1.3 Controlador de radar	5
1.1.1.4 Sistema de adquisición	5
1.2 SISTEMA DE CONTROL DE LOS TRANSMISORES DEL ROJ	5
1.2.1 SISTEMA DE ENFRIAMIENTO	7
1.2.2 HABILITACIÓN DE LOS TRANSMISORES	8
1.2.3 ACTIVACIÓN DE LA FUENTE DE ALTA TENSIÓN	8
1.2.4 ACTIVACIÓN DE LA SEÑAL DE RADIO FRECUENCIA RF	8
1.2.5 MONITOREO DE PARÁMETROS Y SISTEMA DE ALARMAS	8
1.3 DESCRIPCIÓN DEL SISTEMA PROPUESTO	9
1.3.1 PC SUPERVISORA	9
1.3.2 PLC	10
1.3.3 ESTACIONES REMOTAS	10
1.3.4 MÓDULOS ACROMAG	11
CAPÍTULO 2: RED DE COMUNICACIÓN DE DATOS	13
2.1 CAPA FÍSICA	14
2.1.1 VELOCIDAD DE TRANSMISIÓN	14
2.1.2 INTERFASE ELÉCTRICA RS485	14
2.1.3 CABLEADO	15
2.1.4 INTERFASES MECÁNICAS	16
2.1.5 REQUERIMIENTOS EN UN SISTEMA MULTIPUNTO MODBUS	17
2.2 CAPA ENLACE DE DATOS	18
2.2.1 DESCRIPCIÓN DE LA TRAMA MODBUS	18
2.2.2 MODOS DE TRANSMISIÓN	19
2.3 CAPA DE APLICACIÓN	20
2.3.1 DESCRIPCIÓN DEL PROTOCOLO MODBUS	21
2.3.2 TIPOS DE DATOS EN MODBUS	22
2.3.3 CÓDIGOS DE FUNCIÓN MODBUS	22
2.4 DISPOSITIVOS DE RED	23

2.4.1	PLC	23
2.4.2	MÓDULOS ACROMAG	25
2.4.3	ESTACIONES REMOTAS	27

CAPÍTULO 3: CONTROL DE LOS PROCESOS INVOLUCRADOS CON EL ENCENDIDO DE LOS TRANSMISORES **29**

3.1	INTRODUCCIÓN	29
3.2	ENCENDIDO Y APAGADO DE LA ETAPA DE SALIDA <i>PA</i>	30
3.3	ENCENDIDO Y APAGADO DE LA ETAPA PREVIA <i>DRIVER</i>	30
3.4	CONTROL DE LA TENSIÓN DE FILAMENTO DE LA ETAPA DE SALIDA <i>PA</i>	30
3.5	CONTROL DE LA TENSIÓN DE PANTALLA DE AMBAS ETAPAS <i>PA</i> Y <i>DRIVER</i>	31
3.5.1	ESTACIÓN REMOTA	32
3.5.1.1	Adaptador de comunicaciones	32
3.5.1.2	Unidad de control	34
3.5.1.3	Etapas de potencia	35
3.5.1.4	Fuente de alimentación	36
3.5.2	MOTORES PASO A PASO	37
3.5.2.1	Motores paso a paso unipolares	38
3.5.2.2	Motores paso a paso bipolares	38
3.5.2.3	Control de motores paso a paso	39

CAPÍTULO 4: PROTOTIPO DEL SISTEMA **41**

4.1	CONTROL DE LOS PULSADORES ON/OFF	41
4.1.1	INSTALACIÓN DE RELÉS	41
4.2	PROTOTIPO DE LA ESTACIÓN REMOTA	44
4.2.1	DISEÑO DE LOS CIRCUITOS	44
4.2.2	ALOJAMIENTO MECÁNICO Y CONEXIONES	45
4.2.3	SOFTWARE DEL MICROCONTROLADOR PIC	45
4.2.3.1	Rutina de Inicio “RESET”	47
4.2.3.2	Rutina principal “MAIN”	49
4.2.3.3	Rutina de interrupción “INTERRUPT”	50
4.2.3.4	Rutina para la función 0x03 “FUNCT_03”	53
4.2.3.5	Rutina para la función 0x10 “FUNCT_10”	56
4.2.3.6	Rutina de respuesta de excepción Modbus “EXCEPTION”	57
4.2.3.7	Otras subrutinas	59
4.3	MONTAJE DE LOS MOTORES PASO A PASO	59
4.4	INSTALACIÓN DE LOS MÓDULOS ACROMAG	61

CAPÍTULO 5: SOFTWARE DEL PLC Y DEL SISTEMA SCADA **65**

5.1	SOFTWARE DEL PLC	65
5.1.1	PROGRAMA ORIGINAL	68
5.1.1.1	Rutina principal “MAIN”	68
5.1.1.2	Estado Inicial y estado Modo Prueba “S0 y S100”	69
5.1.1.3	Estado Cooling “S1”	70
5.1.1.4	Estados Blower_R y Blower_L “S2 y S3”	71
5.1.1.5	Estado LV “S4”	71
5.1.1.6	Estado RF Permit “S10”	72
5.1.1.7	Estado HV “S11”	72
5.1.1.8	Estados TX1, TX2, TX3 y TX4 “S14, S15, S16 y S17”	73

5.1.1.9	Estado Polling RX y Falla Poll RX “S20 y S21”	73
5.1.2	MODIFICACIONES	74
5.1.2.1	Rutina principal “MAIN”	74
5.1.2.2	Estado Inicial y estado Modo Prueba “S0 y S100”	75
5.1.2.3	Estado PA3 “S14”	75
5.1.2.4	Estado DRV3 “S20”	76
5.1.2.5	Estado Polling “S24”	76
5.1.2.6	Estado Modbus “S26”	76
5.2	SOFTWARE DEL SISTEMA SCADA	76
5.2.1	PROGRAMA ORIGINAL	77
5.2.1.1	Pantalla principal “Sala de Transmisores”	77
5.2.1.2	Pantallas de habilitación de transmisores	78
5.2.1.3	Pantalla “Sistema de Enfriamiento”	79
5.2.1.4	Pantalla “Fuente de Alta Tensión”	80
5.2.1.5	Pantalla “RF Permit”	81
5.2.1.6	Pantalla “Ventiladores del Túnel”	81
5.2.1.7	Pantalla “Bombas Auxiliares”	82
5.2.1.8	Pantalla “Estados y Alarmas del sistema”	82
5.2.2	MODIFICACIONES	83
5.2.2.1	Pantalla principal “Transmitters Room”	83
5.2.2.2	Pantalla de habilitación de transmisores “Transmitter #”	84
5.2.2.3	Pantallas “Driver TX#”	84
5.2.2.4	Pantallas “PA #”	85
5.2.2.5	Pantalla “System parameters”	85
CAPÍTULO 6:	ESULTADOS	87
6.1	CONTROL DEL ENCENDIDO DE LOS TRANSMISORES	87
6.2	CONTROL DE LA TENSIÓN DE PANTALLA	87
6.3	ANÁLISIS ECONÓMICO	89
CONCLUSIONES		91
REFERENCIAS		93
ANEXOS		95

INDICE DE FIGURAS

Figura 1.1: Diagrama de bloques del radar de Jicamarca.	3
Figura 1.2: Antena principal del radar de Jicamarca	4
Figura 1.3: Etapas de amplificación de una unidad de transmisión	5
Figura 1.4: Primera fase de automatización del sistema de transmisores	6
Figura 1.5: Arranque de los equipos del sistema de enfriamiento	7
Figura 1.6: Segunda fase de automatización del sistema de transmisores	10
Figura 1.7: Diagrama de bloques de una estación remota	11
Figura 1.8: Módulo 914MB de Acromag	12
Figura 2.1: Red RS485 en una configuración a 2-Hilos	14
Figura 2.2: Red RS485 en una configuración a 4-Hilos	15
Figura 2.3: Transmisor en una línea balanceada	16
Figura 2.4: Receptor diferencial en una línea balanceada	16
Figura 2.5: Unidad de datos de la aplicación en Modbus	19
Figura 2.6: Direcciones utilizadas en Modbus	19
Figura 2.7: Secuencia de bits en modo RTU sin paridad	19
Figura 2.8: Mensaje ó trama Modbus en modo RTU	19
Figura 2.9: Mensajes y retardos en modo RTU	20
Figura 2.10: Mensaje Modbus en modo ASCII	20
Figura 2.11: Unidad de datos del protocolo	21
Figura 2.12: Solicitud Modbus realizada sin error	21
Figura 2.13: Excepción Modbus	22
Figura 2.14: Categorías de códigos de función Modbus	23
Figura 2.15: Configuración del segundo puerto de comunicaciones del PLC	24
Figura 2.16: Conexión del PLC a la red RS485	25
Figura 2.17: Conexión de los módulos Acromag a la red RS485	27
Figura 2.18: Conexión de una estación remota al bus RS485	28
Figura 3.1: Diagrama de bloques del control de la tensión de filamento de la etapa <i>PA</i>	31
Figura 3.2: Diagrama de bloques de una estación remota	32
Figura 3.3: Esquemático del adaptador de comunicaciones	33
Figura 3.4: Esquemático de las etapas de potencia	36
Figura 3.5: Diagrama de bloques de la fuente de alimentación	37
Figura 3.6: Tipos de motores paso a paso	38
Figura 4.1: Ubicación de los relés en los paneles de control de <i>PA</i> y <i>Driver</i> .	42
Figura 4.2: Cableado de las señales de control e indicación al PLC.	43
Figura 4.3: Prototipo de la estación remota.	44
Figura 4.4: Memoria de programa y stack del PIC16F873A.	46

Figura 4.5: Rutina de reset.	48
Figura 4.6: Subrutinas de reset de los motores paso a paso.	48
Figura 4.7: Rutina principal.	49
Figura 4.8: Subrutinas de actualización de los motores paso a paso.	50
Figura 4.9: Rutina de interrupción	51
Figura 4.10: Subrutinas RX y CRC_RX	52
Figura 4.11: Subrutina para generación del CRC	53
Figura 4.12: Memoria EEPROM del microcontrolador PIC	53
Figura 4.13: Banco 0 de la memoria RAM del microcontrolador PIC	54
Figura 4.14: Rutina para la función 0x03	55
Figura 4.15: Rutina para la función 0x10	56
Figura 4.16: Mensaje de excepción Modbus	57
Figura 4.17: Rutina de excepción Modbus	58
Figura 4.18: Subrutina de transmisión	59
Figura 4.19: Montaje del motor paso a paso con el transformador variable de <i>PA</i>	60
Figura 4.20: Montaje del motor paso a paso con el transformador variable de <i>Driver</i>	61
Figura 4.21: Conexión de las entradas analógicas de los módulos Acromag	63
Figura 5.1: Tareas ejecutadas por el CPU	67
Figura 5.2: Símbolos utilizados en los diagramas de estados	68
Figura 5.3: Rutina principal	69
Figura 5.4: Estado inicial y estado prueba	70
Figura 5.5: Estado Cooling	71
Figura 5.6: Estados Blower_R y BLower_L	71
Figura 5.7: Estado LV	72
Figura 5.8: Estado RF Permit	72
Figura 5.9: Estado HV	73
Figura 5.10: Estado TX1	73
Figura 5.11: Estado Polling RX y Falla Poll RX	74
Figura 5.12: Rutina principal modificada	74
Figura 5.13: Estado Inicial modificado	75
Figura 5.14: Estado DRV3	76
Figura 5.15: Estado Polling	76
Figura 5.16: Pantalla principal “Sala de transmisores”	78
Figura 5.17: Pantalla de habilitación del transmisor N° 1 “PA Tx1”	79
Figura 5.18: Pantalla “Driver Tx1”	79
Figura 5.19: Pantalla “Sistema de Enfriamiento”	80
Figura 5.20: Pantalla “Fuente de Alta Tensión” bloqueada	80
Figura 5.21: Pantalla “Fuente de Alta Tensión” lista	80
Figura 5.22: Pantalla “RF Permit”	81
Figura 5.23: Pantalla “Ventiladores del Túnel”	81
Figura 5.24: Pantalla “Bombas Auxiliares”	82
Figura 5.25: Pantalla “Sistema de Alarmas”	82
Figura 5.26: Pantalla “Transmistters Room” modificada	84
Figura 5.27: Pantalla “Transmitter 3” modificada	84
Figura 5.28: Pantalla “Driver TX3” modificada	85
Figura 5.29: Pantalla nueva “PA TX3”	85
Figura 5.30: Pantalla nueva “System Parameters”	86
Figura 6.1: Software Modbus Tester.	88

INDICE DE TABLAS

Tabla 2.1: Modbus comparado con el modelo OSI	13
Tabla 2.2: Disposición de pines para Modbus a 2-Hilos	17
Tabla 2.3: Retardos en modo RTU	20
Tabla 2.4: Tablas primarias de datos Modbus	22
Tabla 2.5: Funciones Modbus del PLC en modo maestro	24
Tabla 2.6: Funciones Modbus soportadas por los módulos Acromag	26
Tabla 2.7: Parámetros de configuración de los módulos Acromag	26
Tabla 2.8: Funciones Modbus soportadas por las estaciones remotas	27
Tabla 2.9: Parámetros de configuración de las estaciones remotas	28
Tabla 3.1: Parámetros de operación de la etapa de salida de los transmisores	29
Tabla 3.2: Tabla de función del transmisor del integrado SN75176	33
Tabla 3.3: Tabla de función del receptor del integrado SN75176	34
Tabla 3.4: Consumo adaptador de comunicaciones	36
Tabla 3.5: Consumo unidad de control	36
Tabla 3.6: Consumo etapas de potencia	37
Tabla 3.7: Secuencias de control de un motor unipolar	38
Tabla 3.8: Secuencias de control de un motor bipolar	38
Tabla 4.1: Nuevas entradas y salidas	43
Tabla 4.2: Descripción de borneras de la estación remota	45
Tabla 4.3: Puertos utilizados en el microcontrolador PIC	47
Tabla 4.4: Registros utilizados de la memoria EEPROM del microcontrolador PIC	47
Tabla 4.5: Códigos Modbus implementados	51
Tabla 4.6: Mensajes de solicitud, respuesta y error para la función 0x03	54
Tabla 4.7: Mensajes de solicitud, respuesta y error para la función 0x10	57
Tabla 4.8: Posibles códigos de excepción Modbus	57
Tabla 4.9: Características de los motores paso a paso	60
Tabla 4.10: Características de los engranajes	60
Tabla 4.11: Corrientes leídas por los módulos Acromag	61
Tabla 4.12: Direcciones de esclavo de los módulos Acromag	62
Tabla 4.13: Escala de corrientes	62
Tabla 6.1: Pruebas de comunicación de la estación remota	88
Tabla 6.2: Materiales utilizados en la estación remota	89
Tabla 6.3: Materiales necesarios para controlar los motores paso a paso con el PLC	89

INTRODUCCIÓN

La automatización y control de procesos se ha vuelto hoy en día una necesidad para todo tipo de industria, de lo contrario corre el riesgo de quedar obsoleta. Automatizar un proceso es conseguir que este funcione con la mínima intervención humana y mínimas probabilidades de fallo; este concepto resulta muy claro, por ejemplo, en el caso de procesos continuos.

El Radio Observatorio de Jicamarca (ROJ) dada su naturaleza, siempre está en busca de nuevas tecnologías por lo que bajo este principio puso en marcha la primera fase del proyecto “Automatización del sistema de transmisores principales” la cual fue desarrollada junto con el departamento de electrónica y automática DEA de la UDEP, en la ejecución del proyecto se utilizó un PLC (*Programmer Logic Controller*) y un sistema SCADA (*Supervisory Control And Data Adquisition*) para controlar y supervisar respectivamente los procesos.

El sistema que fue desarrollado en dicha oportunidad permite el control y supervisión de los siguientes procesos: activación de la alimentación de baja tensión 120 V, encendido automático del sistema de enfriamiento, activación de la fuente de alta tensión 20 KV, activación del permiso de la señal de RF, supervisión de parámetros y sistema de alarmas.

El objetivo principal de la presente tesis es llevar a efecto la segunda fase del proyecto “Automatización del sistema de transmisores principales”, lo que significa automatizar todos los procesos involucrados con el encendido de los transmisores principales en cuanto a su etapa de salida (*Power amplifier*) y su etapa previa (*Driver*).

Los procesos a automatizar son, en lo que se refiere a la etapa de salida del transmisor el control remoto de los pulsadores de encendido y apagado, el control remoto de la tensión de filamento y el control remoto de la tensión de pantalla. En lo concerniente a la etapa previa: el control remoto de los pulsadores de encendido y apagado y el control remoto de la tensión de pantalla. Como tarea adicional se requiere el monitoreo de las corrientes de operación de los transmisores.

Para la ejecución de la mencionada segunda etapa se seguirá con la misma lógica empleada en la primera fase del proyecto, esto es utilizar el PLC y el sistema SCADA para controlar y supervisar respectivamente los nuevos procesos a automatizar. Además se diseñará y construirá una estación remota, basada en un microcontrolador PIC, la cual se encargará del control de la tensión de pantalla de ambas etapas del transmisor (etapa de salida y etapa

previa). Dicha estación remota debe ser capaz de comunicarse con el PLC a través de una red de comunicaciones RS485 bajo el protocolo Modbus.

Para el monitoreo de las corrientes de operación se utilizarán módulos ADC inteligentes de la marca Acromag, estos módulos para intercambiar información utilizan también el estándar RS485, protocolo Modbus.

La tesis está estructurada en seis capítulos y cuatro anexos en los cuales se muestran en detalle planos, circuitos y el código fuente de los programas desarrollados. Tras la presente introducción, en el capítulo 1 se describe de forma breve los componentes del radar de Jicamarca, también se describe de forma general el sistema de control de los transmisores del ROJ desarrollado en la primera fase del proyecto y finalmente se da una visión general del nuevo sistema a implementar. En este primer capítulo también se describen los componentes que formarán parte del nuevo sistema de control.

El capítulo 2 está dedicado al diseño de la red de comunicaciones utilizada por los diferentes dispositivos que se comunican con el PLC, la red de comunicación de datos se describe siguiendo el modelo de referencia para redes de comunicación OSI (*Open System Interconnection*). Al finalizar el capítulo se provee información de cada uno de los dispositivos que formarán parte de la red de comunicación, en cuanto a sus características y configuraciones.

En el capítulo 3 se analiza el modo en que se automatizarán los procesos involucrados con el encendido de los transmisores. Se analiza el control de los pulsadores de encendido y apagado, el control de la tensión de filamento de la etapa de salida de los transmisores y el control de la tensión de pantalla también de la etapa de salida y de su etapa previa.

Una vez analizados los procesos a automatizar el capítulo 4 está destinado a describir el prototipo del sistema desarrollado el cual ha sido implementado en el transmisor N° 3 del ROJ.

Como complemento al capítulo 4, el capítulo 5 describe el software del PLC y del sistema SCADA, primero se presenta un análisis de los programas originales (desarrollados en la primera fase del proyecto), para luego describir las modificaciones hechas a los mencionados programas.

En el capítulo 6 se presentan los resultados obtenidos durante el desarrollo del prototipo del sistema de control describiendo las pruebas realizadas y haciendo un análisis económico de la estación remota diseñada. Finalmente se presentan las conclusiones a las que se llegó tras la culminación del proyecto.

CAPÍTULO 1

DESCRIPCIÓN GENERAL DEL SISTEMA PROPUESTO

1.1 Introducción

El Radio Observatorio de Jicamarca (ROJ) forma parte de la red ecuatorial de radares de dispersión incoherente (cuyas siglas en inglés son ISR) y es uno de los más importantes en el mundo para el estudio de la ionósfera ecuatorial.

El ROJ pertenece al Instituto Geofísico del Perú y se ubica a media hora de viaje en automóvil hacia el este de Lima y a 10 km de la Carretera Central (latitud 11.95° Sur, longitud 76.87° Oeste). Gracias a su ubicación, el radar de gran capacidad con el que cuenta, permite por ejemplo, determinar con gran precisión la dirección del campo magnético terrestre, además de proporcionar datos invaluable para diversas disciplinas como: física ionosférica ecuatorial, física del plasma y dinámica atmosférica.

1.1.1 El radar de Jicamarca.

El radar de Jicamarca que opera a 50 MHz, es un radar mono-estático lo que significa que utiliza una misma antena para transmitir y recibir las señales, esto es posible gracias a un sistema de conmutación TR (*Transmitter/Receiver*) y ATR (*Anti-Transmitter/Receiver*) que no permite que la potencia de transmisión pase a los receptores.

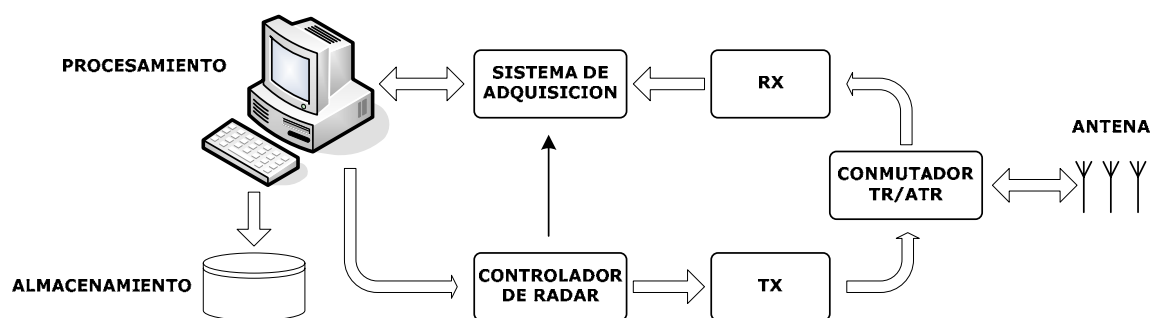


Figura 1.1: Diagrama de bloques del radar de Jicamarca.

1.1.1.1 La antena.

La antena principal del radar de Jicamarca está formada por dos arreglos cuadrados de dipolos horizontales que ocupan un área de aproximadamente 85,000 metros cuadrados; los dos arreglos se encuentran superpuestos y dispuestos de tal forma que los dipolos de uno forman un ángulo recto con los dipolos del otro. Cada arreglo está dividido en cuatro cuartos cada uno de los cuales es alimentado de forma independiente a través de líneas de transmisión. Cada cuarto de antena a su vez consta de 16 módulos de 144 dipolos, así la antena principal cuenta, en total, con 18,432 dipolos (9,216 dipolos del arreglo superior y 9,216 dipolos del arreglo inferior).

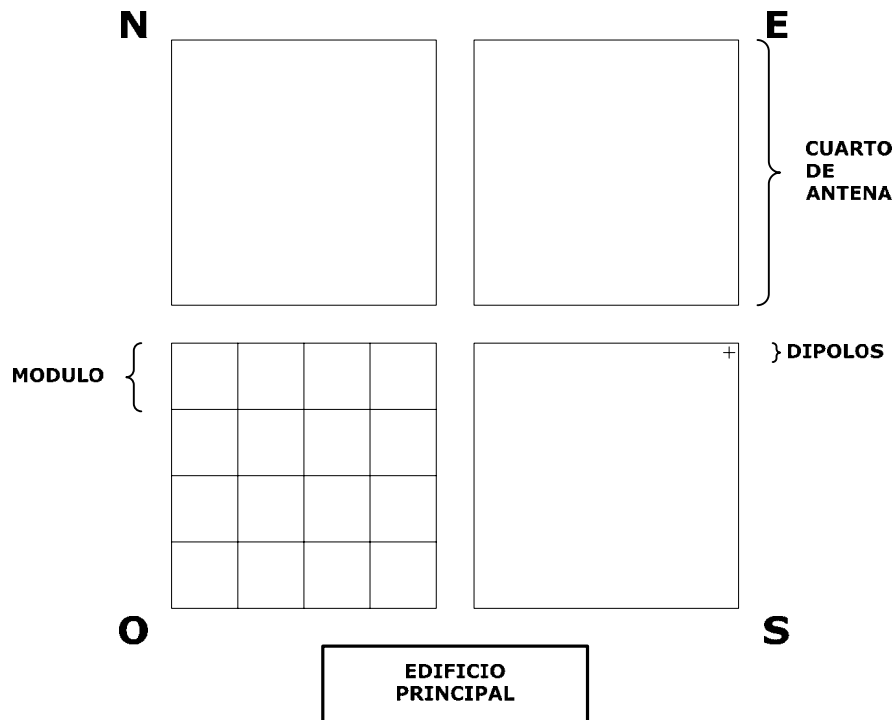


Figura 1.2: Antena principal del radar de Jicamarca

1.1.1.2 Transmisores y receptores

El sistema de transmisores cuenta con cuatro unidades de transmisión (tres en funcionamiento y una en construcción), cada una de las cuales puede entregar una potencia pico de 1.5 MW con un *duty cycle* del 6% (tiempo durante el cual el radar funciona como transmisor en todo un ciclo de funcionamiento), estas unidades de transmisión son capaces de operar de forma independiente o en conjunto.

En cada unidad de transmisión la señal que se envía a la antena es amplificada en varias etapas. La primera de ellas es una etapa de excitación que amplifica la señal hasta los 8 W de potencia, luego la señal pasa por los transmisores llamados MST (*Mesosphere Stratosphere Troposphere*), estos transmisores son conectados directamente a la antena cuando el radar funciona en el modo denominado JULIA (*Jicamarca Unattended Long-term Investigations of the Atmosphere*), los transmisores MST alcanzan una potencia pico de 15 KW. Finalmente cuando el radar opera a máxima potencia se utilizan dos etapas adicionales primero una etapa previa denominada “*Driver*” y finalmente la etapa de salida “*Power Amplifier*” comúnmente llamada “*PA*”. En la Figura 1.3 se muestran las etapas de

amplificación de un transmisor junto con los niveles de tensión que utilizan los respectivos tubos de vacío.

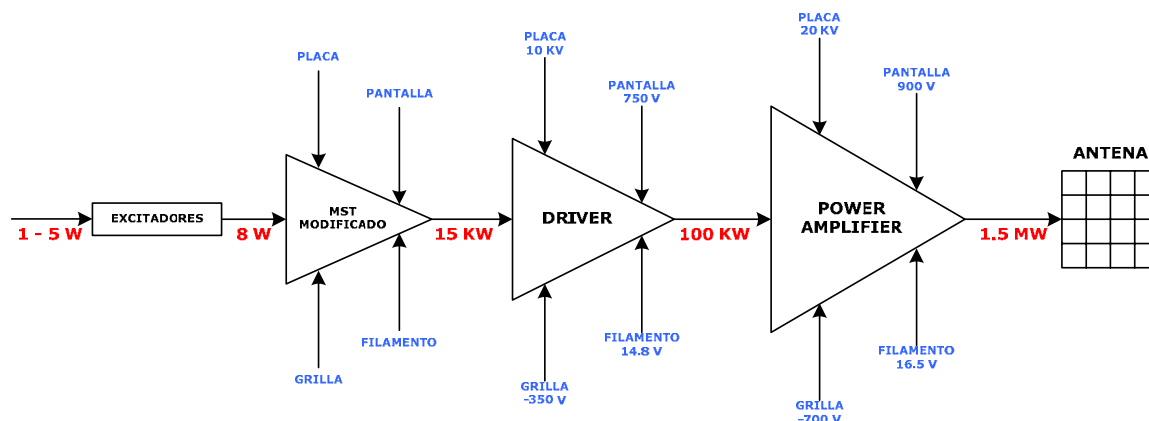


Figura 1.3: Etapas de amplificación de una unidad de transmisión

El sistema de recepción del radar de Jicamarca está formado básicamente por cuatro receptores, los cuales están conectados a la antena a través de pre-amplificadores de bajo ruido llamados *FRONT-ENDs* y el sistema de conmutación TR/ATR.

1.1.1.3 Controlador de radar

El controlador de radar es el que genera las señales o pulsos necesarios para controlar las secuencias de transmisión, recepción y adquisición de cada experimento. Algunos de los pulsos generados por el controlador de radar son:

- ✓ TR: Coloca a la antena en modo transmisor.
- ✓ Tx_Control: Comprende la señal de RF
- ✓ Código: la señal de RF puede ir codificada.
- ✓ Muestreo: Indica al sistema de adquisición cuando leer la señal proveniente de la antena.

1.1.1.4 Sistema de adquisición

El sistema de adquisición esta formado por conversores A/D (analógico/digital), tarjetas para procesamiento digital basadas en DSP (*Digital Signal Processing*) unidades de almacenamiento y presentación de resultados en una PC.

1.2 Sistema de control de los transmisores del ROJ

El sistema de transmisores como se explico anteriormente está formado por unidades de transmisión las cuales a su vez están formadas por etapas de amplificación. Las etapas de potencia *PA* y *Driver*, tienen además de las fuentes de tensión necesarias para su funcionamiento, un sistema de control y un sistema de enfriamiento el cual en el caso de la etapa de salida *PA* consta de circuitos de agua mientras que en la etapa previa *Driver* se utiliza ventilación forzada.

Tanto la etapa de salida *PA* como su correspondiente etapa previa *Driver*, cuentan con su sistema de control respectivo, el cual abarca principalmente los parámetros del sistema de enfriamiento de la etapa de salida (*interlock*), así como algunos otros parámetros eléctricos.

Cada etapa cuenta con su respectivo panel de control local, en el cual se encuentran los controles para su encendido y apagado así como medidores de tensión y corriente de los diferentes parámetros de funcionamiento. La lógica de control se realizaba utilizando relés y temporizadores.

Con el fin de controlar de forma remota (utilizando una PC) los diferentes procesos involucrados en la operación del sistema de transmisores (etapas de potencia), se puso en marcha la primera fase de automatización de dicho sistema, utilizando un PLC (*Programmer Logic Controller*) para el control de los procesos y un sistema SCADA (*Supervisory, Control And Data Adquisition*) para la supervisión de los mismos.

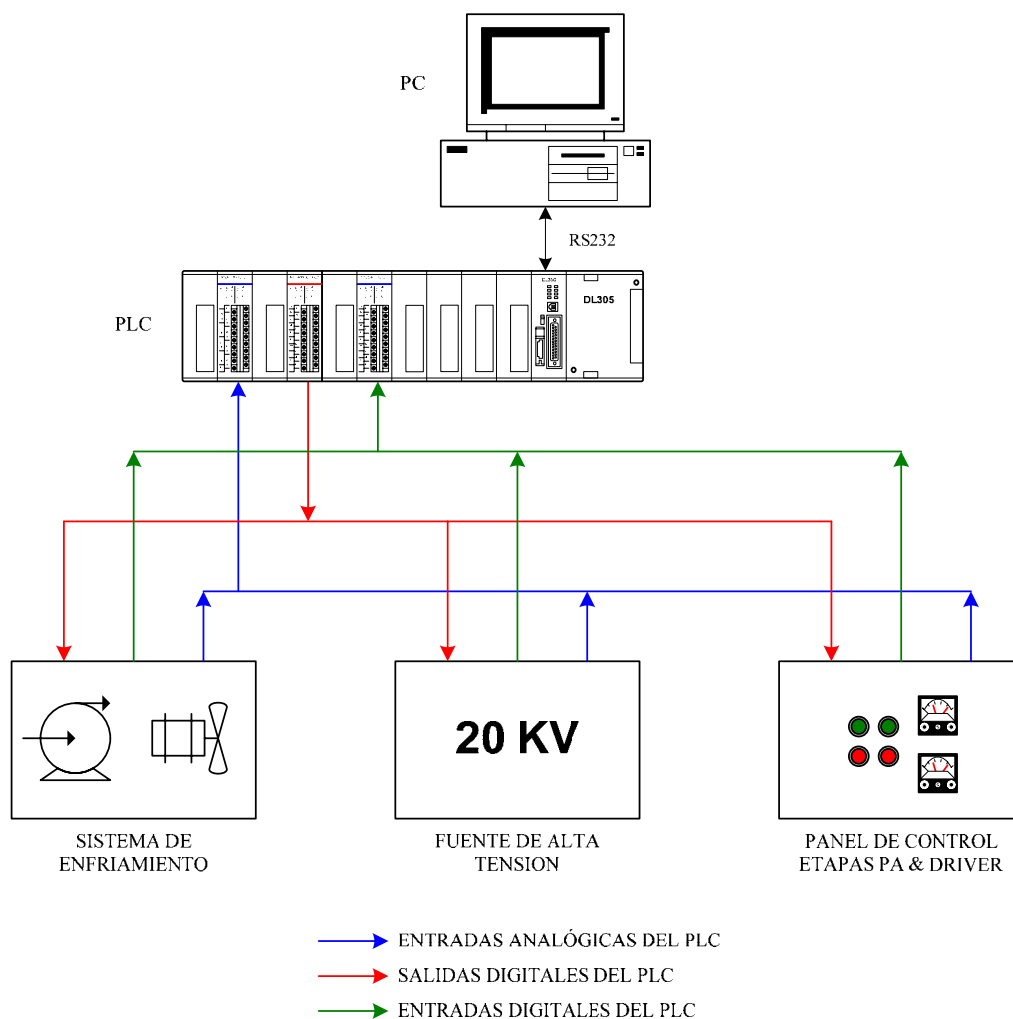


Figura 1.4: Primera fase de automatización del sistema de transmisores

Al finalizar la primera fase del proyecto quedó pendiente la automatización del encendido de la etapa de salida y de su etapa previa, cuyo desarrollo es objetivo del presente trabajo. A continuación se describen los principales procesos automatizados en la primera fase del proyecto la descripción de los programas del PLC y del sistema SCADA así como su funcionamiento se desarrolla en el CAPÍTULO 5.

1.2.1 Sistema de enfriamiento

La etapa de salida del sistema de transmisores cuenta con un sistema de enfriamiento el cual está formado por dos circuitos de agua uno de alta y otro de baja presión además de un circuito de enfriamiento secundario.

- ✓ Circuito de alta presión: es un circuito cerrado de agua destilada, cuenta con una electrobomba de 30 HP, un intercambiador de calor agua – agua y medidores de flujo a la entrada de cada transmisor.
- ✓ Circuito de baja presión: este circuito cuenta con dos electrobombas de 3 HP operando en paralelo, un intercambiador de calor agua – agua y medidores de flujo a la entrada de cada transmisor.
- ✓ Circuito secundario: El circuito secundario llamado también de *Raw Water*, se utiliza para enfriar el agua de los circuitos de alta y baja presión. Cuenta con una electrobomba de 30 HP dos intercambiadores de calor (los del circuito de alta y baja presión) y cuatro intercambiadores agua – aire.

La automatización del sistema de enfriamiento consiste en el arranque secuencial de cada uno de los equipos que forman parte del mencionado sistema y el monitoreo de los flujos de agua de cada uno de los circuitos los cuales están ligados al *interlock* de encendido de los transmisores. El PLC también permite determinar si en el arranque de alguno de los equipos ocurre una falla, para ello se utiliza un contacto NO (*Normally Open*) libre del contactor de potencia, el cual se conecta a un módulo de entradas digitales del PLC.

Para el arranque de los equipos el PLC utiliza módulos con salidas digitales las cuales activan relés auxiliares que a su vez activan los contactores de potencia.

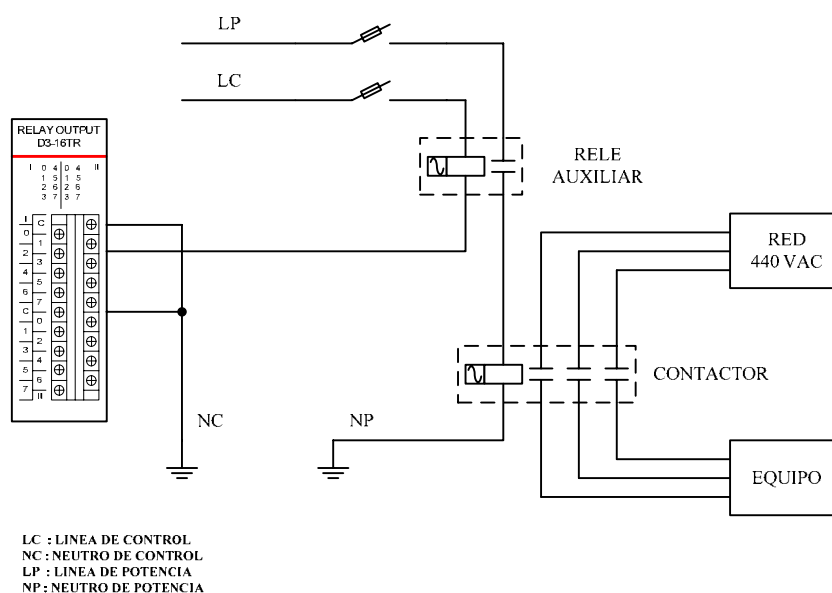


Figura 1.5: Arranque de los equipos del sistema de enfriamiento^[1]

^[1] Automatización de los transmisores del ROJ - Memoria Descriptiva

1.2.2 Habilitación de los transmisores

Para encender la etapa de salida de los transmisores *PA* es necesario que se cumplan ciertas condiciones, en primer lugar se debe activar la tensión primaria de baja tensión (120 VAC) la cual es necesaria para el funcionamiento de los circuitos de control del transmisor y en segundo lugar verificar que los niveles de flujo de los circuitos de alta y baja presión se encuentren dentro de los límites permitidos. Con la automatización del sistema el PLC se encarga de verificar estas condiciones y si se cumplen activa los relés necesarios que permiten el encendido de la etapa de salida. El cual se realiza de forma manual desde su panel de control local.

1.2.3 Activación de la fuente de alta tensión

La etapa de salida de los transmisores cuenta con el tubo de vacío mas potente del mundo, este tubo de vacío utiliza para alimentar uno de sus electrodos (la placa) una fuente de 20 KV. La activación de la fuente ahora la realiza el PLC siguiendo con la respectiva secuencia (las tareas realizadas por el PLC para activar la fuente de alta tensión se especifican en la sección 5.1.1.7).

El PLC también monitorea la señal analógica de la tensión y en caso de producirse alguna falla la fuente es desactivada.

1.2.4 Activación de la señal de radio frecuencia RF

Antes de que la señal de radio frecuencia RF pase al sistema de transmisión es necesario que se cumplan ciertas condiciones, el encendido de la fuente de 20 KV es una de ellas, es por ello que se utiliza un relé para permitir el paso de la señal de RF, este relé se encuentra en la sala de acondicionamiento. El PLC se encarga de verificar que se cumplan todas las condiciones para recién permitir la activación del relé. La cual se puede hacer desde el panel local o desde el sistema SCADA.

1.2.5 Monitoreo de parámetros y sistema de alarmas

La primera fase del proyecto también involucró el monitoreo desde el sistema SCADA de los diferentes parámetros del sistema de transmisores como son los flujos de de los circuitos de agua de alta y baja presión (*Flow Meters*), y las tensiones de operación de los electrodos de los tubos de vacío de las etapas de potencia. En todos los casos fue necesario acondicionar las señales ya que los módulos analógicos del PLC tienen entradas estándares configurables en los rangos $\pm 5\text{VDC}$ y $\pm 10\text{VDC}$.

Los circuitos acondicionadores utilizan puentes rectificadores, en el caso de señales AC, y divisores de tensión en el caso de señales DC, también se utilizan condensadores como filtros y diodos zener como protección por sobre tensiones. Todas las señales son acondicionadas al rango $0 - 5 \text{ VDC}^{[1]}$.

^[1] Automatización de los transmisores del ROJ - Memoria Descriptiva

1.3 Descripción del sistema propuesto

Siguiendo la misma lógica empleada en la primera fase del proyecto, el sistema propuesto utilizará el PLC y el sistema SCADA para controlar y supervisar respectivamente los procesos involucrados en el encendido de los transmisores, los cuales incluyen, en cuanto a la etapa de salida del transmisor, el control remoto de los pulsadores de encendido y apagado, el control remoto de la tensión de filamento y el control remoto de la tensión de pantalla. En cuanto a la etapa previa, comprende el control remoto de los pulsadores de encendido y apagado y el control remoto de la tensión de pantalla.

Adicionalmente al encendido de las etapas de transmisión la segunda fase del proyecto también incluye el monitoreo de las corrientes de operación de los transmisores para lo cual se utilizarán módulos inteligentes de la familia 900MB de la marca Acromag, estos módulos utilizan una red de comunicaciones RS485 para intercambiar información bajo el protocolo Modbus.

Para controlar la tensión de pantalla de cada una de las etapas de potencia del transmisor (etapa de salida y etapa previa), se realizará el diseño de una estación remota, la cual deberá ser capaz de comunicarse con el PLC. Para evitar el cableado adicional (del PLC a los transmisores, donde se ubicarán las estaciones remotas) y, además evitar el uso de salidas adicionales del PLC, la estación remota utilizará también la red de comunicaciones RS485 para intercambiar información con el PLC utilizando obligatoriamente el protocolo Modbus.

1.3.1 PC supervisora

La PC supervisora cuenta actualmente con un sistema SCADA, el cual fue desarrollado en la primera fase del proyecto utilizando el software LookoutDirect, este sistema SCADA cuenta con pantallas para el control y supervisión de los siguientes procesos:

- ✓ Arranque del sistema de enfriamiento, que incluye dos circuitos de agua uno de alta y otro de baja presión, cuatro intercambiadores de calor de agua-aire y 2 ventiladores auxiliares.
- ✓ Selección de los transmisores a utilizar.
- ✓ Activación de las fuentes de alto voltaje (20 KV) para las etapas PA del transmisor.
- ✓ Activación de la señal de radio frecuencia RF.
- ✓ Monitoreo de parámetros y alarmas del sistema
- ✓ Parada de emergencia.

Si bien desde el sistema SCADA se pueden enviar órdenes de parada y de marcha, la PC no realiza ninguna clase de control, el control radica únicamente en el PLC.

Para la implementación de la segunda fase del proyecto es necesario realizar la modificación de algunas pantallas existentes y agregar nuevas pantallas, con sus respectivos controles.

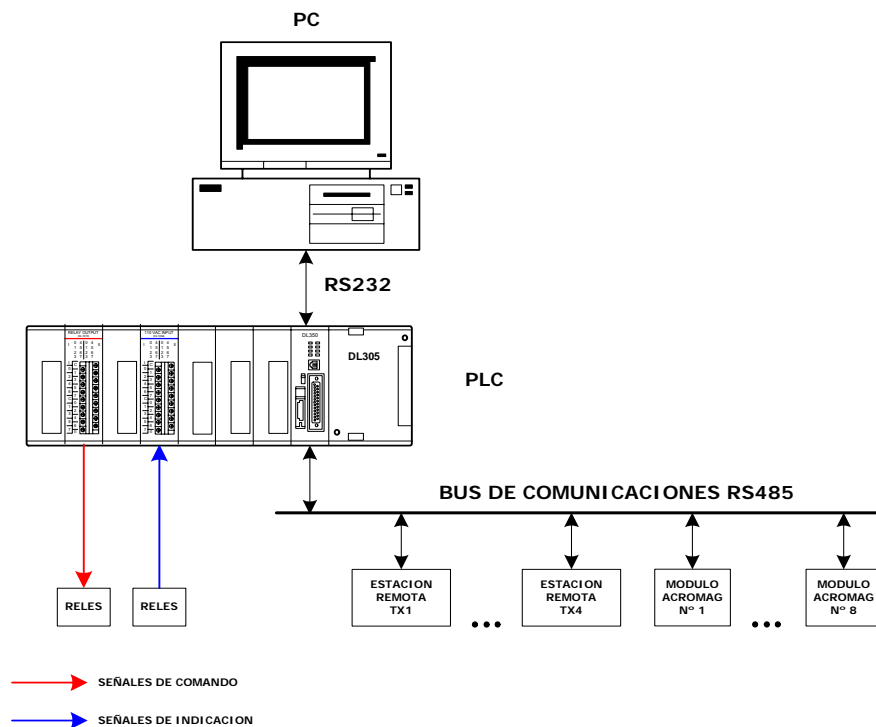


Figura 1.6: Segunda fase de automatización del sistema de transmisores

1.3.2 PLC

El PLC de la familia DirectLogic 305, es la unidad principal del sistema de control, este PLC cuenta con módulos ADC (*Analog to Digital Converter*) con los que se leen tensiones de operación y presiones de agua del sistema de enfriamiento, módulos con entradas digitales con los que se monitorean estados de tipo ON/OFF y módulos con salidas digitales tipo relé con los que se activan o desactivan los relés auxiliares de comando.

El PLC también cuenta con un módulo CPU el cual ejecuta el programa de aplicación, desarrollado también en la primera fase del proyecto, este modulo CPU cuenta además con dos puertos de comunicación el primero se utiliza para la comunicación serial con la PC supervisora y el segundo será utilizado para formar una red de comunicaciones RS-485 en donde el PLC actuará como maestro y las estaciones remotas junto con los módulos Acromag como esclavos.

Para incluir los nuevos procesos a automatizar es necesario realizar también modificaciones al programa original (*ladder diagram*) del PLC. Además será necesaria la instalación de módulos adicionales con salidas digitales tipo relé y entradas digitales a los cuales se conectarán los relés auxiliares de comando e indicación respectivamente.

1.3.3 Estaciones remotas

Cada transmisor contará con una estación remota, la cual se encargará de controlar la tensión de pantalla de las etapas de potencia de los transmisores. Para ello se utilizarán motores paso a paso, los cuales estarán acoplados a los auto – transformadores variables, los que finalmente permiten ajustar las respectivas tensiones de pantalla.

El control de la tensión de pantalla se hace de manera remota, lo cual significa que la estación remota estará a la escucha de los comandos que le envía el PLC, por medio de la red de comunicaciones RS485. El protocolo que utiliza el PLC es Modbus RTU por lo que este protocolo debe ser implementado en la estación remota.

Cada estación remota cuenta con los siguientes elementos: un adaptador de comunicaciones, una unidad de control, una etapa de potencia y una fuente de alimentación.

La unidad de control estará formada por el microcontrolador PIC16F873A el cual se encargará de recibir los comandos del PLC, procesarlos y generar los pulsos necesarios para hacer girar los motores paso a paso.

El adaptador de comunicaciones conecta la unidad de control y la red de comunicaciones mientras que la etapa de potencia es necesaria para suministrar la suficiente corriente a los motores paso a paso.

La fuente de alimentación de la estación remota debe contar con salidas para alimentar tanto la unidad de control, el adaptador de comunicaciones y los motores paso a paso.

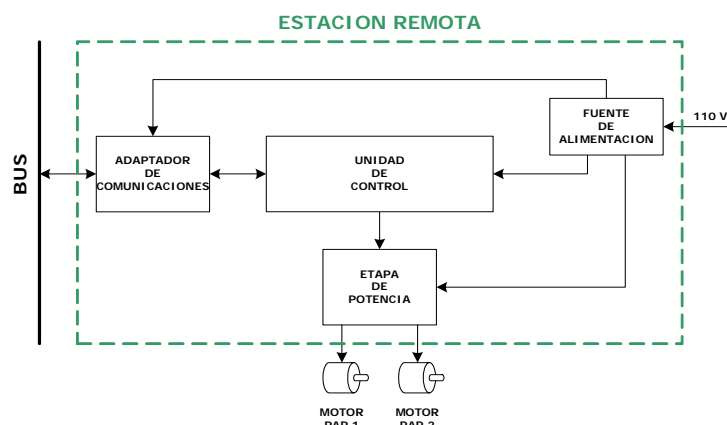


Figura 1.7: Diagrama de bloques de una estación remota

1.3.4 Módulos Acromag

La familia 900MB de Acromag es una línea de módulos I/O multi-función de alto rendimiento, que permiten una comunicación RS485 con una velocidad de hasta 115Kbps bajo el protocolo Modbus RTU.

El modelo utilizado es el 914MB, este acondicionador de señal es un módulo de entradas analógicas (cuatro canales) y cuatro salidas discretas. Proporciona aislamiento entre las entradas, las salidas, la fuente de alimentación y los circuitos de la red. Cuenta con una comunicación de red conforme al protocolo industrial Modbus RTU estándar RS485, y soporta fuentes de alimentación tanto AC como DC.

Las entradas analógicas de los módulos 914MB soportan diferentes rangos de VDC, y sus salidas discretas son bastante flexibles ya que funcionan como alarmas o como controles

ON/OFF. Las alarmas se pueden configurar para activarse cuando alguna entrada analógica se encuentra fuera de los límites permitidos^[2].



Figura 1.8: Módulo 914MB de Acromag

Características:

- ✓ Soporta el protocolo Modbus RTU estándar RS485 con una velocidad de hasta 115 Kbps.
- ✓ Cuatro canales analógicos de 16 bits que permiten una resolución de 0.1% del rango configurable en: $\pm 10\text{VDC}$, $\pm 5\text{VDC}$, $\pm 2.5\text{VDC}$, $\pm 1.25\text{VDC}$, $\pm 625\text{mVDC}$, $\pm 313\text{mVDC}$, $\pm 156\text{mVDC}$ y $\pm 78\text{mVDC}$.
- ✓ Cuatro salidas discretas configurables para funcionar como alarmas ó como controles ON/OFF. Cada salida cuenta con transistores mosfet que soportan corrientes de hasta 1A.
- ✓ Autocalibración que permite disminuir los costos de mantenimiento.
- ✓ Temporizadores “Watchdog” configurables que permiten detectar la perdida de comunicación con el dispositivo maestro.
- ✓ La fuente de alimentación, las entradas analógicas, las salidas discretas y los circuitos de la red se encuentran aislados.
- ✓ Autodiagnóstico que permite detectar fallas operacionales y ejecutar un reset para restaurar el sistema.

Se están utilizando acondicionadores de VDC para monitorear corrientes debido a que las fuentes de alimentación cuentan en su salida con resistencias en derivación (*Shunt*) en cuyos terminales se pueden conectar directamente voltímetros, haciendo la escala respectiva se puede obtener el valor de la corriente.

^[2] BusWorks® Model 913MB/914MB Network I/O Module User's Manual

CAPÍTULO 2

RED DE COMUNICACIÓN DE DATOS

Para cumplir con los requerimientos del nuevo sistema de control es necesario montar una red de comunicación de datos, a la cual estarán conectados el PLC los módulos 914MB de Acromag para la lectura de corrientes además de las estaciones remotas, los módulos Acromag cuentan con una interfase de comunicaciones acorde con el estándar RS485, bajo el protocolo Modbus RTU, mientras que en las estaciones el mencionado protocolo debe ser implementado. El módulo CPU del PLC también es capaz de comunicarse con una red bajo el protocolo Modbus.

El protocolo Modbus fue desarrollado por Modicon para el uso con sus PLC's. Modbus es un protocolo de comunicaciones estándar para la industria, puesto que representa uno de los medios más usados para conectar casi cualquier dispositivo electrónico industrial. Una de las principales razones del uso extenso de Modbus en vez de otros protocolos de comunicaciones, es el hecho de que es público.

Una red de comunicación de datos RS485, protocolo Modbus se adecua al modelo de referencia OSI (*Open System Interconnection*) por lo que la red a implementar se describirá siguiendo este modelo.

Tabla 2.1: Modbus comparado con el modelo OSI^[3]

Nivel	Modelo OSI	Modbus
7	Capa de aplicación	Protocolo de aplicación Modbus
6	Capa de presentación	Vacío
5	Capa de sesión	Vacío
4	Capa de transporte	Vacío
3	Capa de red	Vacío
2	Capa de enlace de datos	Modbus sobre una línea serial
1	Capa física	EIA/TIA-232 o EIA/TIA-485

Como podemos observar en la Tabla 2.1 el protocolo Modbus se ubica en el nivel 7 mientras que su implementación sobre una línea serial ocupa los niveles 1 y 2.

^[3] Specification & Implementation guide for MODBUS over serial line

2.1 Capa física

La capa física del modelo OSI es la que se encarga de las conexiones físicas entre los dispositivos y la red, tanto en lo que se refiere al medio y sus características así como la forma en la que se transmite la información.

2.1.1 Velocidad de transmisión

Un sistema Modbus debe permitir velocidades de 9.6 Kbps y 19.2 Kbps, adicionalmente se pueden implementar otras velocidades (1200, 2400,... 38400 bps, 56 kbps, 115K bps,...) la velocidad por defecto es de 19.2 Kbps^[3].

2.1.2 Interfase eléctrica RS485

El protocolo Modbus puede ser implementado utilizando una interfase eléctrica conforme al estándar EIA/TIA-485 también conocido como RS485, este estándar permite sistemas punto a punto (conexión entre dos dispositivos) y multipunto (conexión entre varios dispositivos). Como máximo la norma describe que se pueden conectar hasta 32 dispositivos transmisores/receptores (transceptores). El rango de voltaje para el modo común (V_{cm}), que pueden tolerar los transmisores y/o receptores, se extiende desde -7V hasta +12V^[4].

En la Figura 2.1 se muestra una red RS485 en una configuración a 2-Hilos, nótese que la línea de transmisión está terminada solo en sus extremos más no en los nodos intermedios.

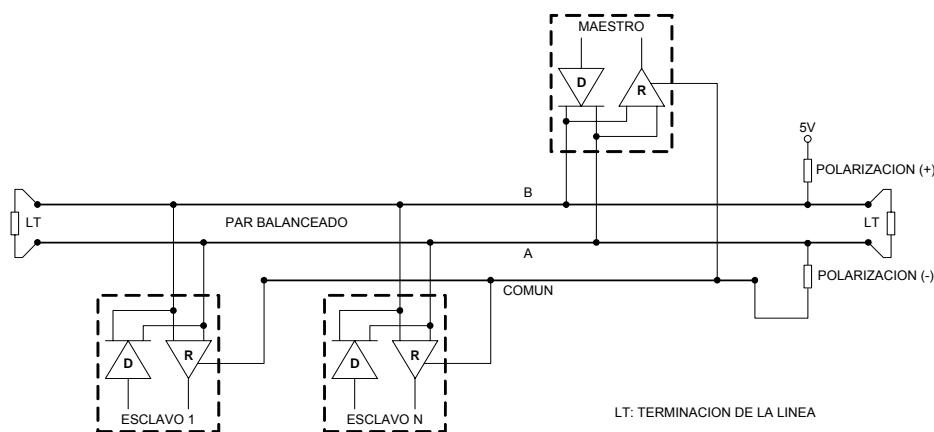


Figura 2.1: Red RS485 en una configuración a 2-Hilos

El estándar RS485 adicionalmente permite una configuración a 4-Hilos (un par maestro y un par esclavo) en la cual los datos sobre el par maestro solo son recibidos por los esclavos mientras que los datos sobre el par esclavo solo son recibidos por el maestro.

^[3] Specification & Implementation guide for MODBUS over serial line

^[4] RS 422/485 Application Note

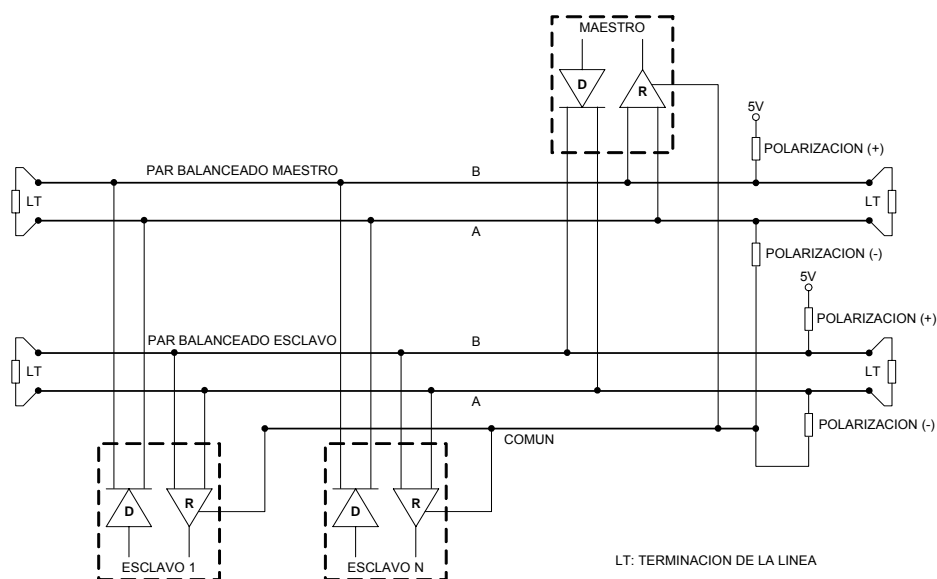


Figura 2.2: Red RS485 en una configuración a 4-Hilos^[3]

Tanto en la configuración a 2-Hilos como en la configuración a 4-Hilos, solo un dispositivo (a la vez) puede transmitir datos.

2.1.3 Cableado

Una red RS485 está constituida por tres conductores, dos de ellos (“A” y “B”) forman un par trenzado balanceado sobre el cual se transmiten y reciben los datos y el tercer conductor que interconecta también todos los dispositivos forma el común.

Se denomina sistema balanceado porque la señal en un conductor es exactamente opuesta a la señal en el otro conductor. En otras palabras, si uno de los cables está transmitiendo una señal de alto nivel (*High*), el otro cable debe estar transmitiendo una señal de bajo nivel (*Low*), y viceversa.

El transmisor en un sistema balanceado puede producir voltajes entre 2 y 6 voltios tanto en la parte positiva como en la parte negativa, aunque la señal de tierra es importante, esta no es utilizada por el receptor balanceado para determinar el estado lógico de la línea de datos. Algunos transmisores (*drivers*) de comunicación pueden también tener una señal de habilitación (*enable*)^[4].

^[3] Specification & Implementation guide for MODBUS over serial line

^[4] RS 422/485 Application Note

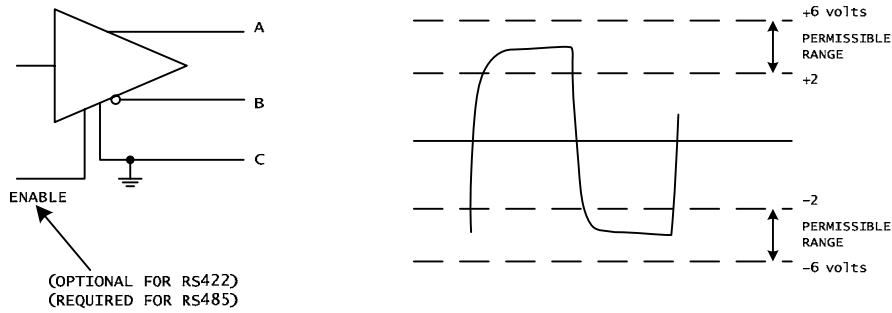


Figura 2.3: Transmisor en una línea balanceada^[4]

Como hemos indicado anteriormente el receptor no necesita, para determinar el estado lógico de la red, la señal de tierra (aunque es necesaria para un correcto diseño del sistema). En la figura 2.4 se muestra un diagrama esquemático de un receptor diferencial. En la figura 2.4 también se muestran los voltajes que son importantes para el receptor balanceado.

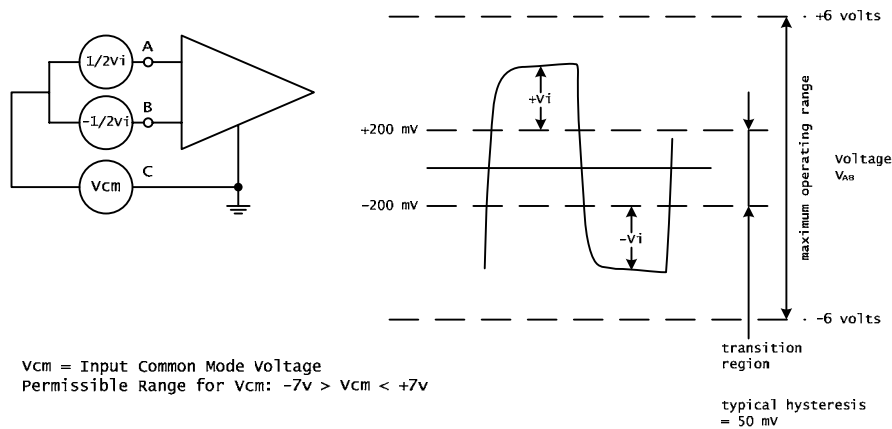


Figura 2.4: Receptor diferencial en una línea balanceada

La señal de un uno lógico (estado de reposo) aparece cuando el voltaje del conductor “A” respecto al del conductor “B” es mayor en 200mV y en el caso contrario, V_B mayor que V_A , tendremos un cero lógico que equivale a un estado activo de la red. La gama de voltajes entre 200mV y 6V es necesaria para evitar la atenuación a lo largo de la línea de transmisión.

2.1.4 Interfases mecánicas

Para la interfase con el bus se pueden utilizar borneras de tornillo, adicionalmente se pueden utilizar conectores del tipo RJ45 ó DB9. En todos los casos se debe proveer la información sobre la ubicación exacta de cada señal^[3].

^[4] RS 422/485 Application Note

^[3] Specification & Implementation guide for MODBUS over serial line

Tabla 2.2: Disposición de pines para Modbus a 2-Hilos^[3]

RJ45 pines	DB9 pines	Nivel de requerimiento	Señal	Descripción
4	5	Requerido	B	Voltaje V_B
5	9	Requerido	A	Voltaje V_A
7	2	Recomendado	Vp	Alimentación 5...24VDC
8	1	Requerido	C	Común

2.1.5 Requerimientos en un sistema multipunto Modbus

Para cualquier sistema multipunto RS485 ya sea en una configuración a 2-Hilos ó a 4-Hilos se deben tener en cuenta los siguientes requerimientos:

- ✓ Número máximo de dispositivos sin repetidor.- Modbus permite un máximo de 32 dispositivos sobre un sistema multipunto sin utilizar repetidor, sin embargo algunos sistemas permiten utilizar un número mayor de dispositivos.
- ✓ Topología.- Un sistema Modbus RS485 sin repetidor utiliza un solo cable principal (línea ó bus) sobre el cual se conectan todos los dispositivos, directamente (formando una *daisy chain*) ó utilizando cables cortos como derivación. El uso de repetidores entre dos sistemas Modbus sobrecargados también es posible.
- ✓ Longitud de la línea.- La longitud máxima de la línea depende de varios factores: la velocidad de transmisión, el cable utilizado (calibre, impedancia característica, capacitancia), el número de cargas sobre la línea y la configuración de la red (2-Hilos ó 4-Hilos). En un sistema trabajando a 9600 bps, con un cable calibre 26 AWG la máxima longitud es de 1000m para una configuración a 2-Hilos (para 4-Hilos la longitud se divide entre dos). La longitud de las derivaciones debe ser corta y en ningún caso mayor a 20 m.
- ✓ Conexión a tierra.- El común del circuito (común del bus y opcionalmente el común de la fuente de alimentación) se debe conectar directamente a la tierra de protección, preferiblemente en un solo punto del bus.
- ✓ Polarización de la línea.- Cuando no existe transmisión de datos en el bus su estado es indefinido y susceptible al ruido o interferencia externa. Esta situación permite que el bus tome cualquier valor, los cuales pueden ser interpretados por los receptores. Para asegurar que los receptores se encuentren en un estado constante cuando el bus se encuentra en reposo es conveniente polarizarlo, así la línea "A" debe estar a 5V y la línea "B" a tierra.
- ✓ Terminación de la línea.- Una línea balanceada utilizada para transmitir datos se comporta como una línea de transmisión por lo que para evitar reflexiones esta debe estar terminada cerca de sus extremos por su impedancia característica. Generalmente una resistencia de 120 Ω es la adecuada. Es importante que la línea esté terminada en ambos extremos dado que las reflexiones son bidireccionales

^[3] Specification & Implementation guide for MODBUS over serial line

pero no se permite colocar más de dos resistencias de terminación. Cada resistencia de terminación debe ser colocada entre los dos conductores “A” y “B”.

- ✓ Protecciones.- Dado que las redes de comunicación están muchas veces expuestas a inducciones del entorno, se pueden producir sobre tensiones que pueden dañar a los elementos de la red. La inmunidad al ruido, ofrecido por las redes RS485, en su esquema diferencial de funcionamiento puede no ser suficiente. La protección puede realizarse de varias formas, las más usuales son los sistemas que proporcionan aislamiento galvánico, como los opto aislados, y los sistemas que cuentan con circuitos limitadores de señal, realizados generalmente con diodos zener.

2.2 Capa enlace de datos

Esta capa se encarga de conseguir que la información fluya libre de errores sobre el medio físico. Para ello se tiene que montar bloques de información (tramas), gestionar la detección y corrección de errores, además del control del flujo de información entre los dispositivos.

Modbus es un protocolo Maestro/Esclavo, donde solo un maestro (a la vez) está conectado al bus y uno o varios dispositivos esclavos (hasta un máximo de 247) también están conectados al mismo bus. Una transacción Modbus siempre la inicia el maestro, mientras que los esclavos nunca transmiten sin antes recibir una solicitud del maestro. La comunicación entre esclavos no es posible y solo el maestro puede iniciar una transacción a la vez.

El maestro puede enviar una solicitud a un esclavo en dos modos diferentes^[3]:

- ✓ Modo *Unicast*.- En este modo el maestro envía una solicitud a un único esclavo, (identificado con una única dirección) el cual después de recibir y procesar la solicitud envía un mensaje de respuesta al maestro.
- ✓ Modo *Broadcast*.- El maestro envía una solicitud a todos esclavos, sin embargo, estos no envían una respuesta al maestro, generalmente se utiliza con funciones de escritura. Para identificar una solicitud en modo *broadcast* se utiliza la dirección cero.

2.2.1 Descripción de la trama Modbus

Cuando Modbus es implementado sobre una red o bus se introducen campos adicionales a la PDU (ver capa de aplicación) de esta manera se forma la unidad de datos de la aplicación ADU (*Application Data Unit*).

^[3] Specification & Implementation guide for MODBUS over serial line

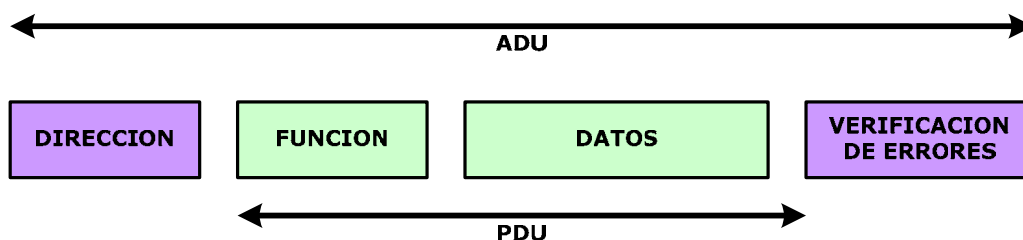


Figura 2.5: Unidad de datos de la aplicación en Modbus^[3]

En un mensaje de solicitud el maestro coloca en el campo dirección, la dirección del esclavo con el que quiere comunicarse, mientras que en el mensaje de respuesta el esclavo coloca su propia dirección para indicarle al maestro quien ha respondido.

0	1 - 247	248 - 255
DIRECCION BROADCAST	DIRECCIONES PARA ESCLAVOS	RESERVADOS

Figura 2.6: Direcciones utilizadas en Modbus

El campo de verificación de errores contiene un valor obtenido del calculo de redundancia (CRC ó LRC) sobre el contenido del mensaje, con este valor es posible determinar errores en la trama.

2.2.2 Modos de transmisión

El modo de transmisión serial por defecto es el RTU (*Remote Terminal Unit*) en este modo cada byte de 8 bits contiene dos caracteres hexadecimales de 4 bits. Cada byte de datos es enviado con un bit de arranque al inicio y dos bits de parada al final, el primer bit de parada se puede utilizar como bit de paridad. El tamaño máximo de un mensaje en modo RTU es de 256 bytes.

START	1	2	3	4	5	6	7	8	STOP	STOP
-------	---	---	---	---	---	---	---	---	------	------

Figura 2.7: Secuencia de bits en modo RTU sin paridad

DIRECCION	FUNCION	DATOS	CRC
1 BYTE	1 BYTE	0-252 BYTES	2 BYTES

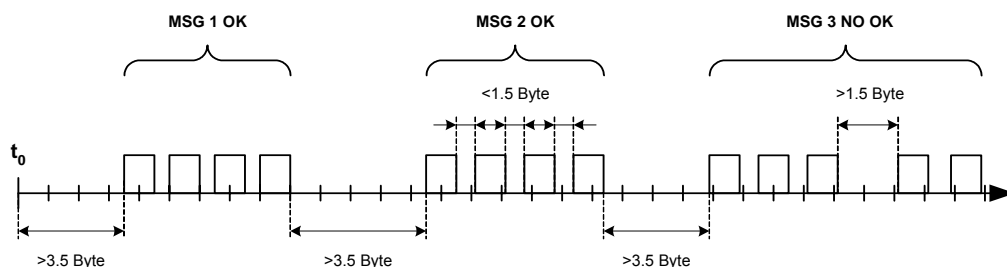
Figura 2.8: Mensaje ó trama Modbus en modo RTU

En modo RTU los mensajes deben estar separados por un intervalo de tiempo en silencio de al menos 3.5 veces el tamaño de un byte, de esta manera se puede detectar el inicio de un mensaje. Si en la transmisión de un mensaje ocurre un silencio, entre dos bytes, mayor a 1.5 veces el tamaño de un byte, el mensaje se declara incompleto y debe ser descartado.

^[3] Specification & Implementation guide for MODBUS over serial line

Tabla 2.3: Retardos en modo RTU

Velocidad de Transmisión [bps]	Retardo entre Mensajes recomendado [μs]	Retardo entre Bytes [μs]
4800	7000	2500
9600	3500	1250
19200	1750	625

Figura 2.9: Mensajes y retardos en modo RTU^[3]

El Modo RTU incluye comprobación de errores basado en el método CRC (*Cyclical Redundancy Check*) ejecutado sobre el contenido de todo el mensaje. El campo CRC contiene un valor de 16-Bit implementado en dos 8-Bit bytes los cuales son anexados al final de cada mensaje.

El otro modo de transmisión permitido por el protocolo Modbus es el modo ASCII (*American Standard Code for Information Interchange*), este modo opcional es utilizado cuando los dispositivos no permiten el modo RTU, en cuanto al control de los retardos por ejemplo. Este modo es menos eficiente que el modo RTU debido a que cada byte es enviado como dos caracteres ASCII por ejemplo: el byte 0x5B es enviado como dos caracteres 0x35 y 0x42 (0x35 = “5” y 0x42 = “B” en ASCII).

INICIO	DIRECCION	FUNCION	DATOS	LRC	FIN
1 CHR :	2 CHR	2 CHR	0-2X252 CHR	2 CHR	2 CHR CR, LF

Figura 2.10: Mensaje Modbus en modo ASCII

Para detectar el inicio y el fin de un mensaje en modo ASCII se utilizan caracteres especiales, los mensajes deben comenzar con el carácter “:” (ASCII 0x3A) y deben terminar con los caracteres “CRLF” (*carriage return – line feed* ASCII 0x0D y 0x0A), el método de comprobación de errores en este modo es LRC (*Longitudinal Redundancy Checking*).

2.3 Capa de aplicación

El protocolo Modbus proporciona una comunicación de tipo Maestro/Esclavo entre diferentes dispositivos conectados en una misma red o bus y se basa en el principio de

^[3] Specification & Implementation guide for MODBUS over serial line

solicitud/respuesta, que ofrecen diferentes servicios especificados por códigos de función, los cuales serán descritos más adelante.

2.3.1 Descripción del protocolo Modbus

El protocolo Modbus define una simple unidad de datos del protocolo o PDU (*Protocol Data Unit*) la cual es independiente de las capas de comunicación inferiores, la PDU consta de dos campos: el campo función y el campo de datos^[5].

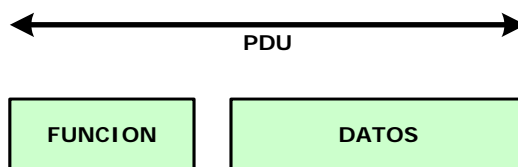


Figura 2.11: Unidad de datos del protocolo

Para iniciar una transacción Modbus el dispositivo maestro construye una PDU (mensaje de solicitud) que envía al dispositivo esclavo el cual a su vez construye una nueva PDU (mensaje de respuesta), que contiene los datos solicitados, y la envía al dispositivo maestro. Si el dispositivo esclavo no puede ejecutar la acción requerida por el dispositivo maestro este envía una PDU denominada mensaje de excepción.

El campo función de la PDU es codificado en un byte y puede tomar valores en el rango de 1 a 255 en decimal, cuando un mensaje es enviado a un dispositivo esclavo el campo función le indica el tipo de acción a ejecutar.

El campo datos contiene información adicional que el dispositivo esclavo necesita para ejecutar la acción que indica la función Modbus. En algunos tipos de solicitud el campo datos es omitido en el mensaje, lo que indica que la función en si es suficiente para realizar la acción.

Si ningún error relacionado con la ejecución de la solicitud requerida por el dispositivo maestro ocurre, la PDU que envía como respuesta el dispositivo esclavo contiene: en el campo función, la especificada por el dispositivo maestro y en el campo datos la información requerida.

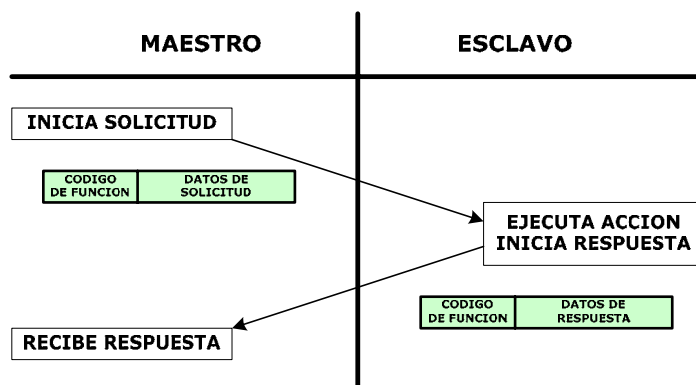


Figura 2.12: Solicitud Modbus realizada sin error

^[5] MODBUS Application Protocol Specification

En el caso contrario el campo función contiene: el código de error (función más el valor 0x80) que le indica al dispositivo maestro que ha ocurrido algún error, y el campo datos contiene el código de excepción que indica el tipo de error ocurrido.

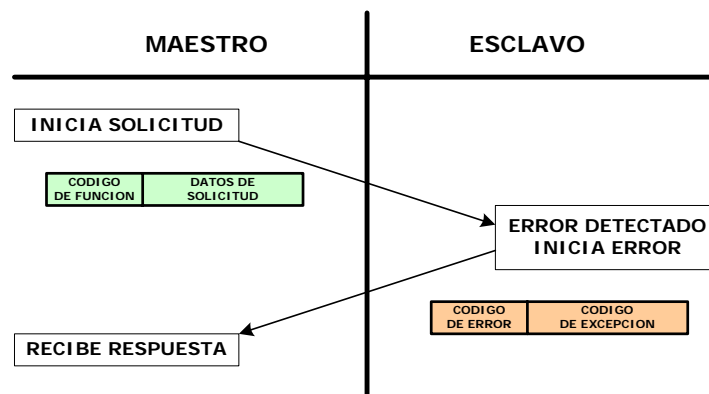


Figura 2.13: Excepción Modbus^[5]

2.3.2 Tipos de datos en Modbus

El tipo de datos utilizados en Modbus hacen referencia principalmente a bits y registros los cuales deben estar mapeados en la memoria del dispositivo al cual se quiere acceder, en general el tipo de datos en Modbus se basa en una serie de tablas con diferentes características. Las cuatro tablas primarias se muestran en la Tabla 2.2.

Tabla 2.4: Tablas primarias de datos Modbus

Tabla	Tipo de dato	Acceso
<i>Discretes Input</i>	Un solo bit	Lectura
<i>Coils</i>	Un solo bit	Lectura/Escritura
<i>Input Registers</i>	Palabra de 16 bits	Lectura
<i>Holding Registers</i>	Palabra de 16 bits	Lectura/Escritura

2.3.3 Códigos de función Modbus

Los códigos de función Modbus pueden tomar valores en el rango de 1 a 127 en decimal y están divididos en tres categorías:

- ✓ Códigos públicos, son los códigos validados por Modbus, además de ser únicos se encuentran amplia y públicamente documentados.
- ✓ Códigos definidos por el usuario, como su nombre lo indica estos códigos son utilizados para implementar funciones no definidas por Modbus, por lo que no hay garantía de que el código sea único (lo pueden utilizar diferentes usuarios para diferentes propósitos)
- ✓ Códigos reservados, utilizados por algunas compañías y no se encuentran disponibles al público.

^[5] MODBUS Application Protocol Specification

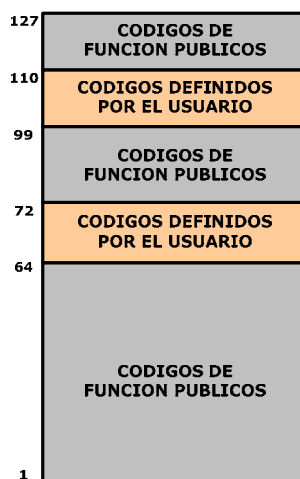


Figura 2.14: Categorías de códigos de función Modbus^[5]

2.4 Dispositivos de red

2.4.1 PLC

El PLC cuenta con un modulo CPU DL350, el cual puede comunicarse con una red Modbus a través de su segundo puerto de comunicaciones. Este puerto de comunicaciones soporta los estándares RS-232C y RS422C por lo que para poder comunicarse con la red RS485 es necesario utilizar un adaptador. El adaptador a utilizar es el FA-ISONET, el cual cuenta con las siguientes características^[6]:

- ✓ Convierte la mayoría de interfases de datos RS232 en una interfase de datos aislada RS422 (4-Hilos) ó RS485 (2-Hilos). Los requerimientos mínimos de la interfase RS232 son las salidas TXD y RTS y la entrada RXD, RS485 requiere que el dispositivo RS232 controle correctamente la señal de RTS.
- ✓ El transmisor del adaptador está protegido contra corto circuitos y sobre cargas.
- ✓ El adaptador cuenta con una protección transitoria por sobre tensiones.
- ✓ La fuente de alimentación de 5V para el RS232 se puede tomar directamente del cable de comunicaciones ó usar una fuente externa de 24VDC.
- ✓ Las resistencias de terminación requeridas en los extremos del bus están incorporadas.
- ✓ La polarización del bus tanto positiva como negativa también están incorporadas
- ✓ La transmisión y recepción cuentan con led's indicadores para simplificar la detección de errores.

En la red Modbus el PLC actuará como maestro por lo que este inicia las transacciones para recibir/enviar información a los esclavos, el PLC en modo maestro cuenta con las siguientes funciones Modbus para el manejo de variables:

^[5] MODBUS Application Protocol Specification

^[6] Universal Isolate Network Adapter Data Sheet

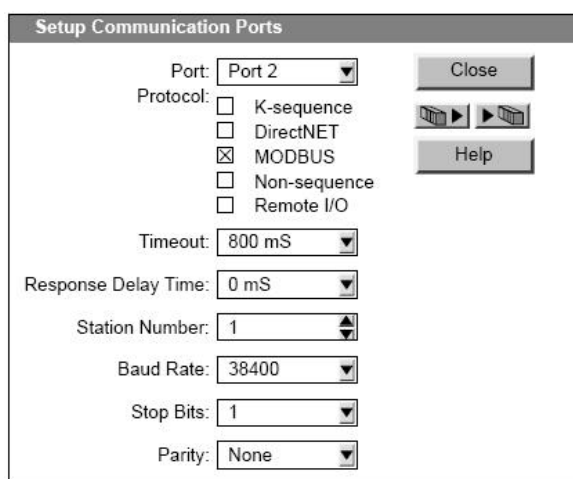
Tabla 2.5: Funciones Modbus del PLC en modo maestro^[6]

Código (hex)	Función	Tipo de dato	Referencia Modbus
0x03	<i>Read holding registers</i>	Palabra de 16 bits	4xxxx
0x04	<i>Read input registers</i>	Palabra de 16 bits	3xxxx
0x10	<i>Write multiple holding registers</i>	Palabra de 16 bits	4xxxx

Para configurar el segundo puerto de comunicaciones del PLC como maestro Modbus se debe acceder al software DirectSoft Programming elegir el menú PLC, luego Setup, luego “Secondary Comm Port”

- ✓ Port: elegir “Port 2” de la lista.
- ✓ Protocol: seleccionar “MODBUS” en la lista de verificación, el cuadro de dialogo aparecerá como en la figura 2.15.
- ✓ Timeout: cantidad de tiempo que el puerto espera después de enviar un mensaje para recibir una respuesta antes de registrar un error.
- ✓ Response Delay Time: cantidad de tiempo entre la habilitacion de la señal RTS y el envío de datos.
- ✓ Station Number: dirección de esclavo Modbus (1 – 90 BCD), si el PLC actúa como maestro seleccionar “1”.
- ✓ Baud Rate: los posibles valores son 300, 600, 900, 2400, 4800, 9600, 19200, y 38400 bps. Todos los dispositivos conectados a la red deben operar con el mismo valor.
- ✓ Stop Bits: escoger entre 1 ó 2 bits de parada para usar en el protocolo (por defecto 2 bits).
- ✓ Parity: elegir entre par, impar ó ninguna paridad para la detección de errores (por defecto ninguna).

Finalmente enviar la configuración al PLC con el comando respectivo.

Figura 2.15: Configuración del segundo puerto de comunicaciones del PLC^[6]

^[6] Universal Isolate Network Adapter Data Sheet

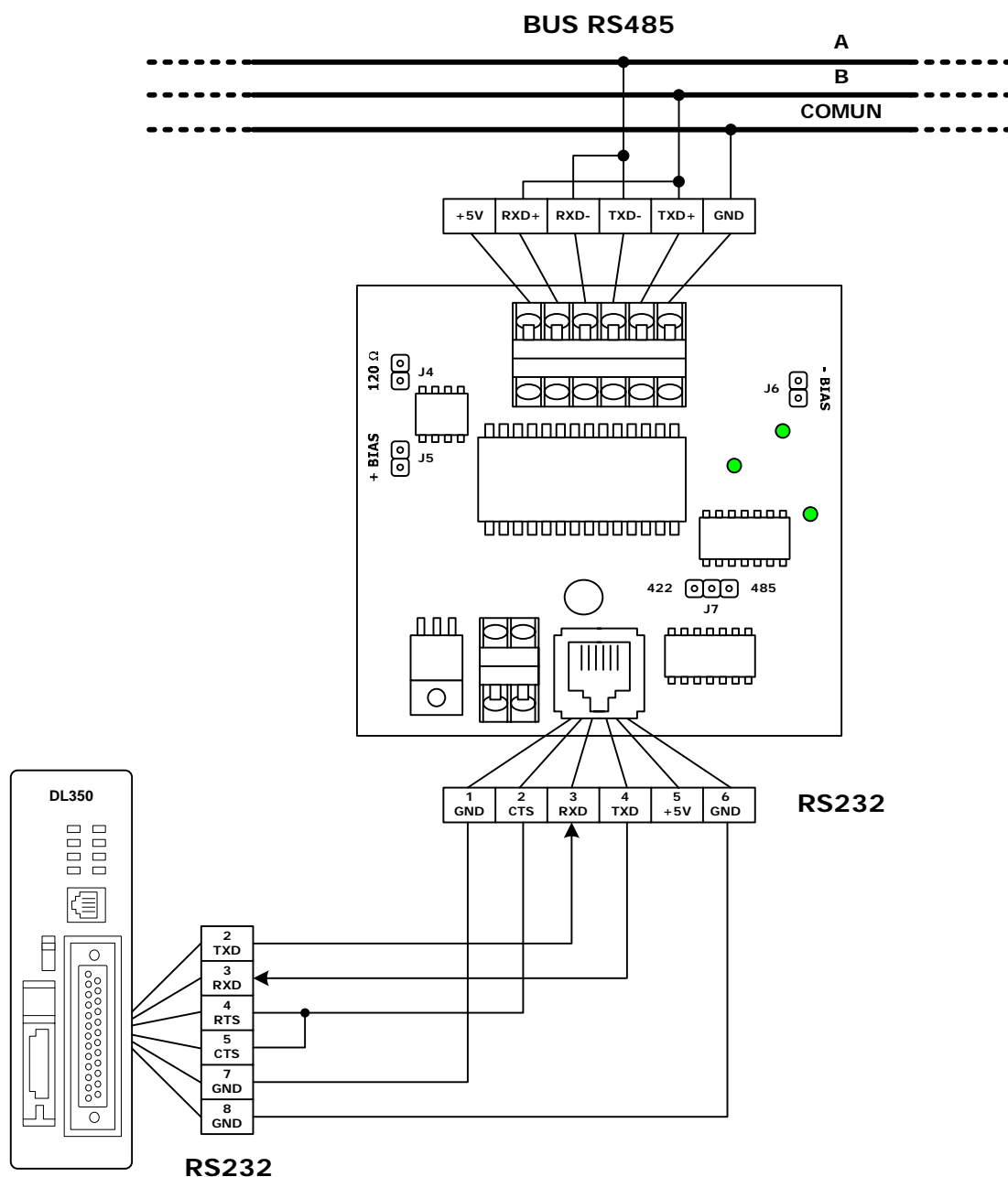


Figura 2.16: Conexión del PLC a la red RS485

2.4.2 Módulos Acromag

Estos acondicionadores de señal cuentan con una interfase de comunicaciones conforme al estándar RS485 (configuración a 2-Hilos) de alta velocidad bajo el protocolo Modbus RTU, las funciones que soportan se muestran en la Tabla 2.6.

Tabla 2.6: Funciones Modbus soportadas por los módulos Acromag^[2]

Código (hex)	Función	Tipo de dato	Referencia Modbus
0x01	<i>Read coil (output status)</i>	Bit	0xxxx
0x03	<i>Read holding registers</i>	Palabra de 16 bits	4xxxx
0x04	<i>Read input registers</i>	Palabra de 16 bits	3xxxx
0x05	<i>Force single coil (output)</i>	Bit	0xxxx
0x06	<i>Write single holding registers</i>	Palabra de 16 bits	4xxxx
0x08	<i>Reset slave</i>	-	Oculto
0x0F	<i>Force Multiple coils (outputs)</i>	Bit	0xxxx
0x10	<i>Write multiple holding registers</i>	Palabra de 16 bits	4xxxx
0x11	<i>Report slave ID</i>	-	Oculto

Los módulos Acromag cuentan con un pulsador (DFT) para cargar los parámetros de configuración por defecto sin embargo estos parámetros pueden ser modificados según se requiera, para ello se cuenta con el software Series900MB.exe. Adicionalmente la memoria que contiene los parámetros de configuración es accesible vía Modbus por lo que se puede utilizar cualquier estación maestra para configurarlos.

Tabla 2.7: Parámetros de configuración de los módulos Acromag

Registro	Referencia	Descripción	Valores
0	40001	Dirección de esclavo Default: 247	1 – 247
1	40002	<i>Baud Rate</i> Default: 2, 9600 bps	0 = 2400bps 1 = 4800bps 2 = 9600bps 3 = 14400bps 4 = 19200bps 5 = 28800bps 6 = 38400bps 7 = 57600bps 8 = 76800bps 9 = 115200bps
2	40003	Paridad Default: 0, ninguna	0 = ninguna, 2 stop bits 1 = impar, 1 stop bit 2 = par, 1 stop bit
3	40004	Tiempo del Watchdog Default: 0, deshabilitado	0 – 65534 (segundos)
4	40005	Habilitación de salidas digitales cuando se alcanza el tiempo del Watchdog Default: 65535, deshabilitado	Ch0 = bit 0 Ch2 = bit 1 Ch3 = bit 2 Ch4 = bit 3
5	40006	Retardo de respuesta Default: 0, sin retardo	0 – 65500 ticks (1 tick = 1.085 µs)
6	40007	Configuración de entradas analógicas Default: 0, ±10V	0 = ±10V 1 = ±5V 2 = ±2.5V 3 = ±1.25V 4 = ±625mV 5 = 313mV 6 = ±156mV 7 = ±78mV

Cuando el módulo está configurado con los parámetros por defecto el led status (ST) parpadea, si se cambia alguno de los parámetros de configuración es necesario resetear el módulo para que estos tengan efecto.

^[2] BusWorks® Model 913MB/914MB Network I/O Module User's Manual

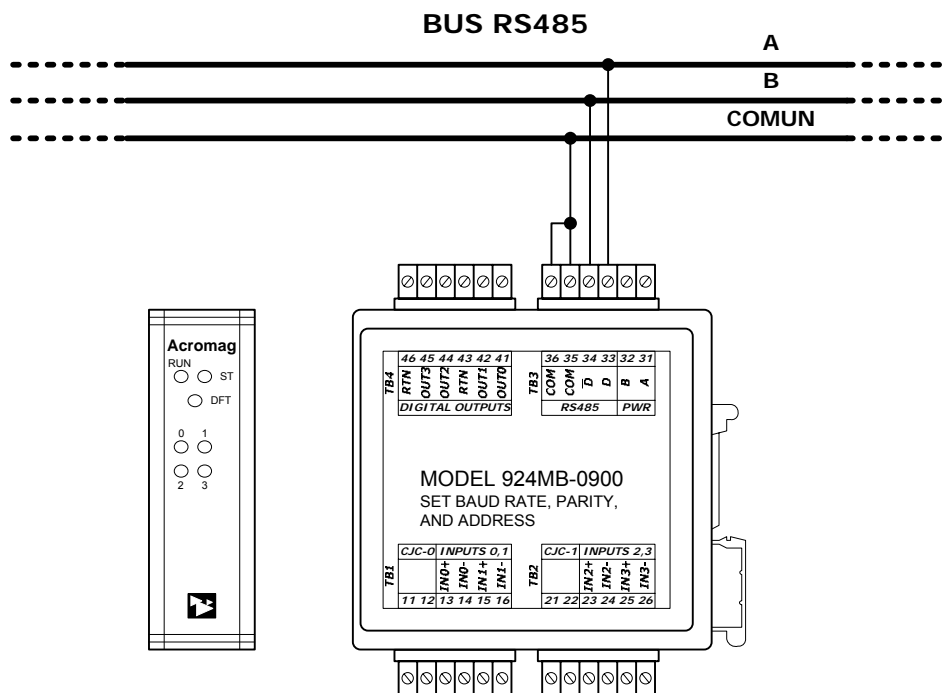


Figura 2.17: Conexión de los módulos Acromag a la red RS485

2.4.3 Estaciones remotas

La estación remota cuenta con una unidad de control la cual está formada por un microcontrolador PIC16F873A, en el cual se ha implementado el protocolo Modbus. Las funciones soportadas por las estaciones remotas se muestran en la tabla 2.8, con estas funciones es posible leer y escribir en las memorias de datos (EEPROM y RAM) del microcontrolador PIC.

Tabla 2.8: Funciones Modbus soportadas por las estaciones remotas

Código (hex)	Función	Tipo de dato	Referencia Modbus
0x03	<i>Read holding registers</i>	Palabra de 16 bits	4xxxx
0x10	<i>Write multiple holding registers</i>	Palabra de 16 bits	4xxxx

El microcontrolador PIC cuenta con un puerto de comunicaciones USART (*Universal Synchronous Asynchronous Receiver Transmitter*) el cual opera con niveles TTL (*Transistor Transistor Logic*) por lo que para poder conectarse al bus RS485 es necesario adaptar las señales TTL a las utilizadas en este estándar. Para ello se utiliza un adaptador de comunicaciones. Para mas detalle sobre el adaptador de comunicaciones de la estación remota referirse a la sección 3.5.1.1.

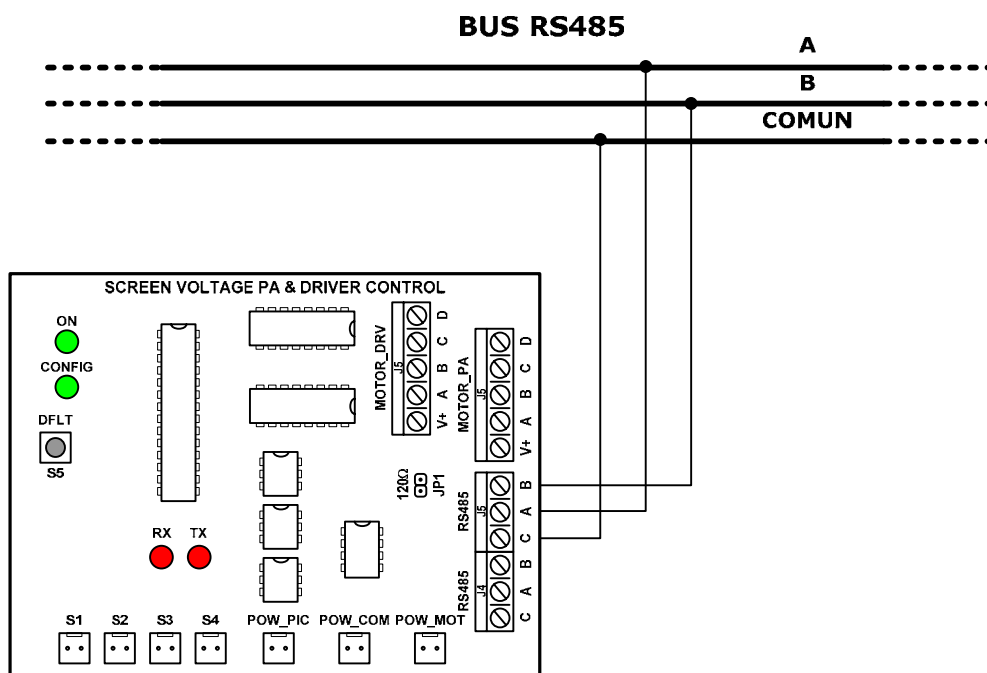


Figura 2.18: Conexión de una estación remota al bus RS485

Al igual que el PLC y los módulos Acromag las estaciones remotas tienen que ser configuradas antes de poder comunicarse a través de la red RS485. Los parámetros de configuración se guardan en la memoria EEPROM del microcontrolador PIC, estos parámetros pueden ser modificados utilizando el modo configuración (*switch* S4) de la estación remota y el protocolo Modbus. Sin embargo cuando los parámetros de configuración son incorrectos se cargan los valores por defecto, estos valores por defecto también se pueden cargar utilizando el pulsador S5.

Tabla 2.9: Parámetros de configuración de las estaciones remotas

Registro	Referencia	Descripción	Valores
0x00	40001	Dirección de esclavo Default: 127	1 – 127
0x01	40002	Baud Rate Default: 1, 9600 bps	0 = 4800 bps 1 = 9600 bps 2 = 19200 bps
0x02	40006	Retardo de respuesta Default: 0, sin retardo	0 – 200 ticks (1 tick = 10 μ s)
0x03	40007	Variable de reset Default: 0	0 = normal 255 = reset

CAPÍTULO 3

CONTROL DE LOS PROCESOS INVOLUCRADOS CON EL ENCENDIDO DE LOS TRANSMISORES

3.1 Introducción

La etapa de salida de los transmisores *PA* tiene como elemento principal un tubo de vacío de alta potencia tetrodo modelo 8973 de la marca Eimac, el cual es usado como amplificador de RF, este tubo cuenta con cuatro electrodos: el filamento (cátodo), la placa (ánodo), la grilla de control y la grilla pantalla.

Los parámetros controlados en esta etapa corresponden a la tensión de filamento, tensión de grilla, tensión de pantalla y fuente de alta tensión de placa. De los cuales la tensión de filamento y la tensión de pantalla son variables. En cuanto las dos restantes (tensiones de grilla y de placa) son fijas^[7].

Tabla 3.1: Parámetros de operación de la etapa de salida de los transmisores^[7]

Parámetro	Rango de operación
Tensión de filamento	0 – 16.5 VDC
Tensión de pantalla	0 – 900 VDC
Tensión de grilla	-700 VDC
Tensión de placa	20 KV

La etapa previa de los transmisores *Driver* también cuenta con un tubo de vacío el 4CX40,000G. El cual también posee cuatro electrodos: el filamento ó cátodo, la placa o ánodo, la grilla de control y la grilla pantalla.

La operación de la etapa *Driver* está definida por la tensión de pantalla, la cual dependiendo del nivel de potencia que se quiera transmitir se ajusta entre los valores 0 y 750 VDC como máximo^[7].

Ambas etapas *PA* y *Driver* cuentan cada una con un panel de control local en el cual se encuentran los controles para su encendido y medidores para visualizar tanto las tensiones como las corrientes de funcionamiento.

^[7] PA & DRIVER Control and Operation

El nuevo sistema de control permitirá controlar de manera remota, utilizando el software SCADA de la PC, los pulsadores de encendido y apagado de estas etapas, asimismo ajustar la tensión de filamento de la etapa *PA* y la tensión de pantalla de ambas etapas.

3.2 Encendido y apagado de la etapa de salida *PA*

El panel de control local de esta etapa cuenta con dos juegos de pulsadores ON/OFF, el primero para el encendido y apagado del sistema y el segundo para el encendido y apagado de la tensión de pantalla del tubo de radiofrecuencia.

Para controlar de manera remota estos pulsadores es necesario instalar, en el panel de control de esta etapa, relés auxiliares los cuales serán controlados por un módulo de salidas tipo relé del PLC.

Es necesario también utilizar relés auxiliares para conmutar los nuevos circuitos con los circuitos existentes, esto para que sea posible la operación de manera local y sin utilizar el PLC. Estos relés de conmutación serán activados por un selector *Local/Remote* ubicado en el mismo panel de control local.

3.3 Encendido y apagado de la etapa previa *Driver*

Al igual que la etapa *PA* el panel de control local de esta etapa cuenta con dos juegos de pulsadores ON/OFF el primero para el encendido y apagado del sistema y el segundo para el encendido y apagado de la tensión de pantalla. Para controlar de manera remota estos pulsadores se utilizarán relés auxiliares los cuales serán activados por un modulo de salidas tipo relé del PLC. También se utilizarán relés para conmutar los circuitos nuevos de los existentes, esto por la misma razón que en la etapa *PA*.

3.4 Control de la tensión de filamento de la etapa de salida *PA*

La tensión de filamento de la etapa *PA* es proporcionada por una fuente DC, la cual cuenta con un auto – transformador variable que tiene como tensión de entrada 440 V trifásicos y como salida una tensión que puede ir de 0 V a 20 V, también trifásicos. Esta tensión de salida es rectificada y conectada al cátodo (filamento) del tubo de vacío de la etapa de salida *PA*.

El transformador variable de la fuente de la tensión de filamento está acoplado a un motor síncrono de control bidireccional, con el cual es posible ajustar la tensión de filamento haciéndolo girar en un sentido ó en otro.

El control del giro del motor síncrono se realiza utilizando dos selectores del panel de control. Un primer selector de dos posiciones *Auto/Manual* es utilizado para incrementar de forma automática la tensión de filamento hasta alcanzar un límite prefijado. El segundo selector de tres posiciones *Raise/ Neuter /Lower*, es utilizado para incrementar ó disminuir la tensión de filamento, cuando el selector *Auto/Manual* se encuentra en la posición *Manual*.

El control de la tensión de filamento se basa en la activación de unos relés que no permiten que la tensión de filamento alcance unos límites preestablecidos, quitando luego la alimentación del motor síncrono.

Siguiendo con la misma lógica, con el nuevo sistema de control, la tensión de filamento será ahora controlada por el PLC, utilizando el valor digital de la tensión de filamento para fijar los límites, y relés auxiliares para alimentar el motor síncrono. El diagrama de bloques del nuevo sistema de control se muestra en la siguiente figura.

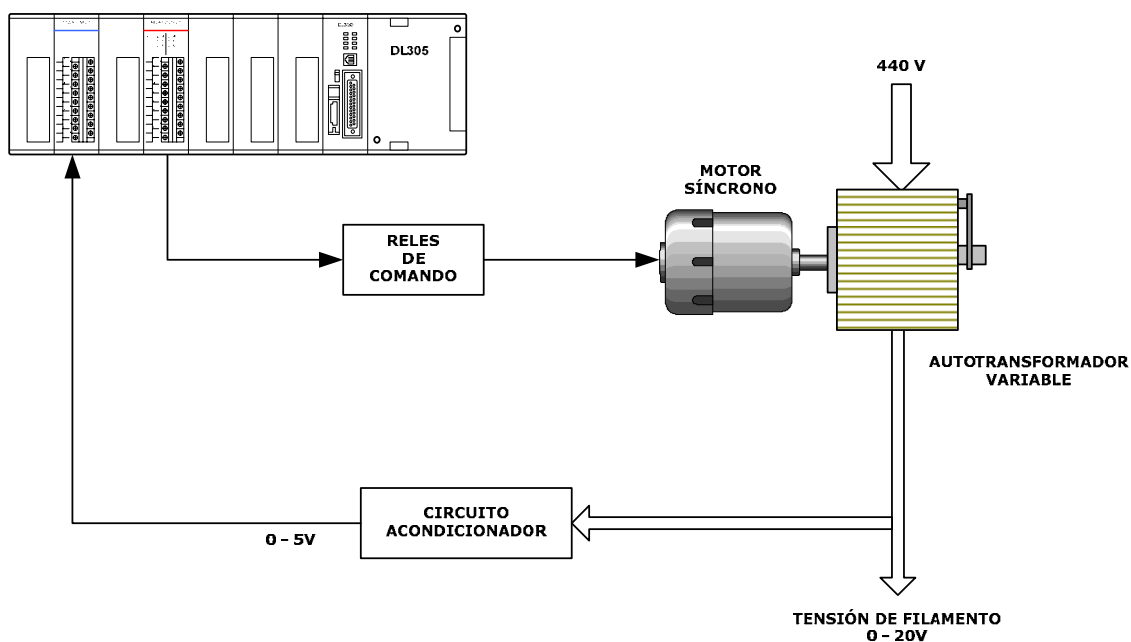


Figura 3.1: Diagrama de bloques del control de la tensión de filamento de la etapa *PA*

Es necesario que el sistema también permita la operación de forma local por lo que se utilizan relés auxiliares para conmutar los circuitos nuevos de los existentes, estos relés auxiliares serán activados por el selector *Local/Remote* del panel de control.

3.5 Control de la tensión de pantalla de ambas etapas *PA* y *Driver*

La tensión de pantalla de la etapa de salida *PA* es proporcionada por una fuente DC, la cual entrega una tensión variable entre 0 VDC y 1350 VDC esta tensión es ajustada utilizando un auto – transformador variable. El valor nominal de operación de la tensión de pantalla en esta etapa es de 900 V^[7].

La tensión de pantalla en la etapa previa *Driver*, también es proporcionada por una fuente DC, que entrega una tensión variable entre 0 VDC y 1200 VDC esta tensión también es ajustada usando un transformador variable. Los valores de operación de la tensión de pantalla en la etapa *Driver* pueden variar en el rango de 500 VDC a 750 VDC dependiendo de la potencia que se quiera transmitir^[7].

^[7] PA & DRIVER Control and Operation

Los elementos a controlar son los auto – transformadores variables a los cuales se acoplarán motores paso a paso, los que a su vez serán controlados por una estación remota. El PLC se comunicará con la estación remota enviándole los datos necesarios para hacer girar los motores paso a paso, la comunicación entre el PLC y estaciones remotas se basará en el estándar RS485, protocolo Modbus RTU.

3.5.1 Estación remota

La estación remota se compone de una unidad de control (elemento principal), un adaptador de comunicaciones y dos etapas de potencia (arreglo de transistores en *Darlington*), que permiten alimentar los motores paso a paso. Dicha estación incluye una fuente de alimentación con salidas múltiples. El diagrama de bloques de la estación remota se muestra en la Figura 3.2.

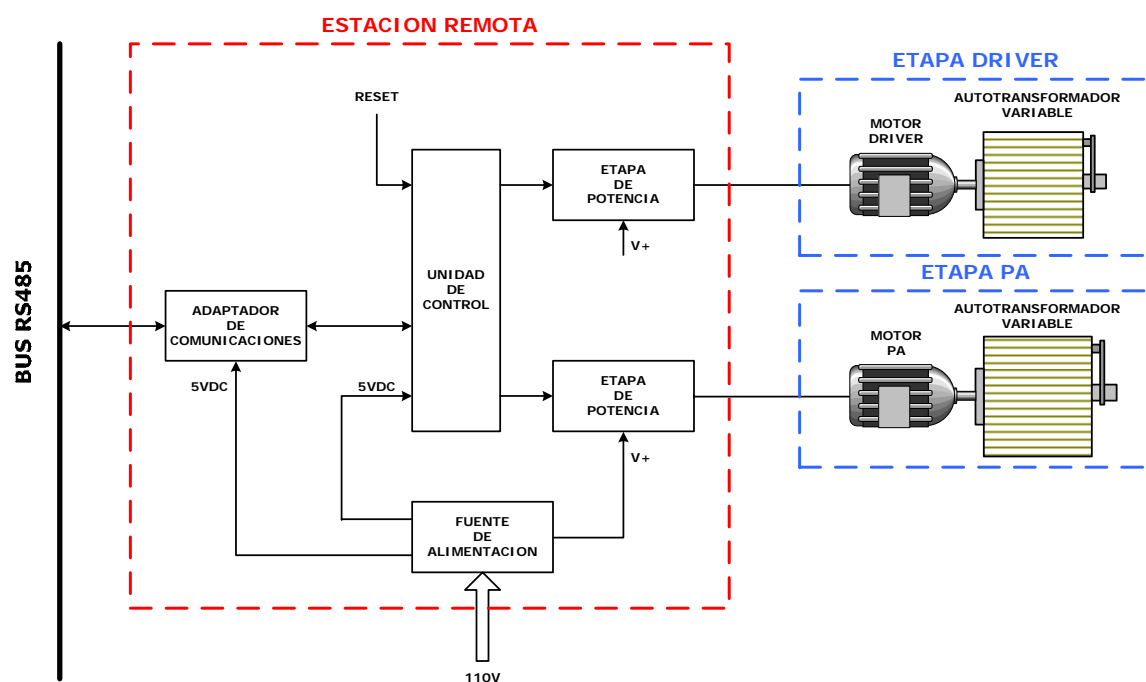


Figura 3.2: Diagrama de bloques de una estación remota

3.5.1.1 Adaptador de comunicaciones

El adaptador de comunicaciones es la interfase entre la red o bus y la unidad de control, su función es la de adaptar las señales del estándar RS485 a señales compatibles que puedan ser interpretadas por la unidad de control.

La unidad de control cuenta con una interfase para comunicaciones seriales USART (*Universal Synchronous Asynchronous Receiver Transmitter*) que opera con niveles TTL por lo que para poder conectar la unidad de control al bus de comunicaciones RS485 es necesario adaptar las señales TTL a las utilizadas en el estándar RS485, para ello se utiliza un transceptor RS485 como el circuito integrado SN75176. Este integrado contiene un receptor diferencial con histéresis y un transmisor, también diferencial, que puede ser puesto en estado de alta impedancia. Esto permite que varios transceptores estén conectados al mismo bus.

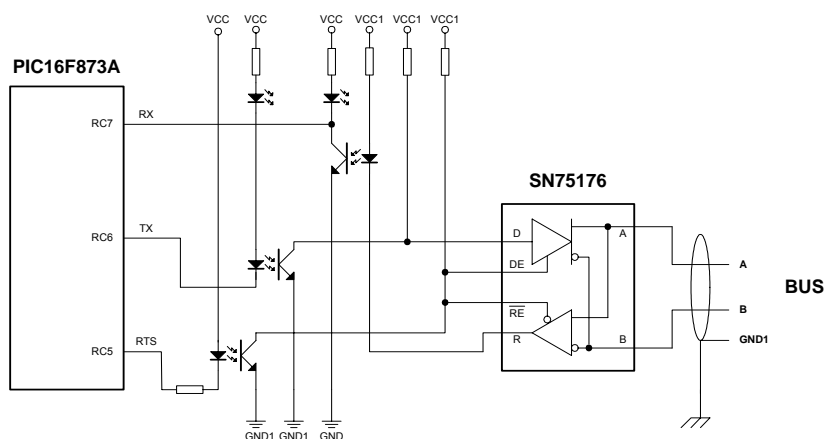


Figura 3.3: Esquemático del adaptador de comunicaciones

Para aislar el circuito del adaptador de comunicaciones, este cuenta con su propia fuente de alimentación (VCC1 y GND1) y para aislar las señales de datos se utilizan optoacopladores.

La señal “RX” de la unidad de control sigue la señal “R” del transceptor, mientras que la señal “D” del transceptor sigue la señal de “TX” de la unidad de control. Tal y como está diseñado el adaptador no hay inversión en las señales de datos transmitidos ó recibidos entre cada lado de los optoacopladores.

Para habilitar la transmisión/recepción se utiliza la señal RTS de tal forma que cuando no hay actividad en el bus (ningún transmisor activo) se coloca la señal RTS en nivel bajo, este nivel bajo excita el optoacoplador que a su vez en el lado del bus pone a cero las entradas DE (*Driver Enable*) y RE’ (*Receiver Enable*) del transceptor habilitando la recepción (al mismo tiempo que se deshabilita la transmisión). En el caso contrario colocando la señal RTS en nivel alto se habilita la transmisión y se deshabilita la recepción.

Existen dos diodos LED para monitorear la actividad del bus los cuales siguen las señales RX y TX de la unidad de control, el conjunto LED con su resistencia se utilizan también para polarizar estas dos líneas.

Tabla 3.2: Tabla de función del transmisor del integrado SN75176^[8]

Entradas		Salidas	
D	DE	A	B
1	1	1	0
0	1	0	1
X	0	Alta impedancia	Alta impedancia

^[8] SN75176A Differential Bus Transceiver – Data Sheet

Tabla 3.3: Tabla de función del receptor del integrado SN75176^[8]

Entradas		Salidas
A-B	RE'	R
$V_{AB} > 0.2V$	0	1
$-0.2V < V_{AB} < 0.2V$	0	?
$V_{AB} < -0.2V$	0	0
X	1	Alta impedancia
Abierto	0	?

3.5.1.2 Unidad de control

La unidad de control la formará un microcontrolador PIC, esta familia de microcontroladores ofrece una gran variedad de dispositivos con diversas características, tanto en velocidad de operación, capacidad de memoria, diversos periféricos y puertos de entrada y/o salida.

La unidad de control debe ser capaz de generar la lógica necesaria para el control de los motores paso a paso y a la vez comunicarse con el PLC utilizando el protocolo Modbus sobre un sistema multipunto RS485. El microcontrolador seleccionado para cubrir estos requerimientos es el PIC16F873A. Las principales características de este microcontrolador se mencionan a continuación^[9]:

- ✓ Procesador RISC (*Reduce Instruction Set Computer*)
- ✓ Velocidad de operación, dependiendo del cristal usado, hasta 20 MHz.
- ✓ Velocidad de ejecución de instrucciones hasta 5MHz (la cuarta parte de la frecuencia del cristal usado)
- ✓ Memoria Flash programable 4096 palabras de 14 bits.
- ✓ Memoria de datos RAM de 192 bytes.
- ✓ Memoria de datos EEPROM de 128 bytes.
- ✓ Capacidad de manejo de 14 interrupciones.
- ✓ Programación serial mediante dos pines.
- ✓ Protección de código programable.

Además cuenta con los siguientes periféricos:

- ✓ Veintidós puertos de entrada y salida.
- ✓ Tres temporizadores, uno de 8 bits (TRM0) y dos de 16 bits (TMR1 y TMR2)
- ✓ Puerto de comunicación serial MSSP y USART.
- ✓ Cinco canales ADC de 10 bits.
- ✓ Dos módulos de captura, comparación y PWM.

El software que se ejecuta en el microcontrolador PIC se encargará de hacer girar los motores paso a paso, los cuales a su vez ajustan las tensiones de pantalla, según se requiera. La posición de los motores es enviada por el PLC a través de la red de comunicaciones. Si bien es cierto el microcontrolador cuenta con un puerto de comunicaciones seriales USART, es necesario implementar en software funciones del protocolo Modbus de acuerdo a nuestros requerimientos. Las funciones a implementar son las siguientes:

^[8] SN75176A Differential Bus Transceiver – Data Sheet

^[9] PIC16F87XA Data Sheet

- ✓ Función 0x03 “*Read Holding Registers*”, será utilizada para leer uno o varios registros (de la memoria RAM o de la memoria EEPROM) del microcontrolador PIC.
- ✓ Función 0x10 “*Write Multiple Registers*”, será utilizada para escribir en uno o en varios registros (de la memoria RAM o de la memoria EEPROM) del microcontrolador PIC.

La estación remota contará con dos modos de operación:

- ✓ Modo normal, en este modo el software ejecuta un lazo cerrado que compara la posición actual del motor paso a paso con la posición enviada por el PLC. En caso de ser diferentes, la posición del motor es actualizada. En el modo normal las funciones Modbus se ejecutan sobre la memoria RAM del microcontrolador PIC.
- ✓ Modo configuración, este modo se utiliza para configurar los parámetros de comunicación de la estación remota, en este modo las funciones Modbus se ejecutan sobre la memoria de datos EEPROM del microcontrolador PIC.

Independientemente del modo de operación se debe mantener atención a la actividad en la red de comunicaciones, esto se realiza por medio de una rutina de interrupción, si se recibe algún mensaje se procede a analizarlo, ejecutarlo y responderlo.

La unidad de control contará además con un pulsador de reset para restaurar el sistema, dos entradas digitales para determinar cuando los motores alcanzan la posición cero, una entrada digital para determinar el modo de operación y finalmente dos indicadores (led's) el primero utilizado para indicar que la estación remota se encuentra funcionando normalmente y el segundo para indicar el modo de operación de la misma.

3.5.1.3 Etapas de potencia

El microcontrolador PIC16F873A de la unidad de control tiene salidas con niveles TTL que pueden proporcionar una corriente de hasta 20mA, esta corriente es insuficiente para alimentar los motores paso a paso, por lo que es necesario utilizar una etapa de potencia. Para ello se utiliza el circuito integrado ULN2064, el cual posee cuatro arreglos de transistores en Darlington capaces de entregar una corriente de hasta 1.5A por salida. Este integrado cuenta además con entradas compatibles TTL y diodos supresores para cargas inductivas^[10].

^[10] ULN2064B Quad Darlington switches – Data Sheet

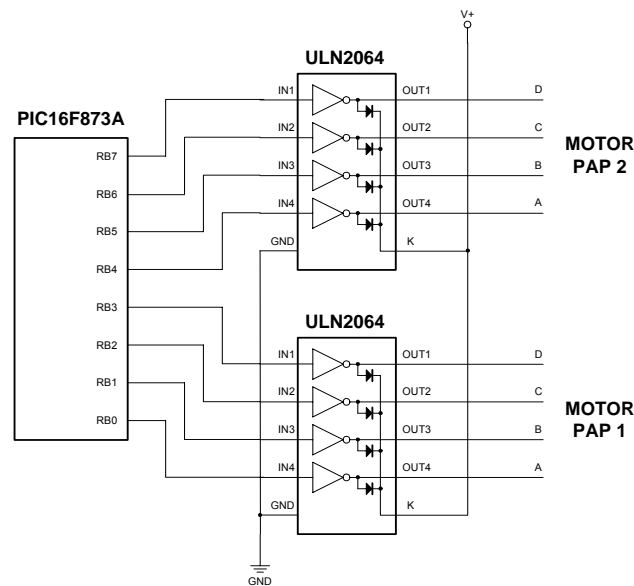


Figura 3.4: Esquemático de las etapas de potencia

3.5.1.4 Fuente de alimentación

La fuente de alimentación de la estación remota alimenta la unidad de control, el adaptador de comunicaciones y las etapas de potencia de los motores paso a paso. Los requerimientos de voltaje y corriente de cada elemento de la estación remota se han obtenido de las hojas de datos de cada uno de los componentes del circuito.

Tabla 3.4: Consumo adaptador de comunicaciones

Elemento	Cantidad	Voltaje [V]	Corriente total [mA]
SN75176	1	5	150
Optoacopladores (Input)	2	5	40
Optoacopladores (Output)	1	5	35
Adicional	-	-	30
Total			255

Tabla 3.5: Consumo unidad de control

Elemento	Cantidad	Voltaje [V]	Corriente total [mA]
PIC16F873A	1	5	100
Optoacopladores (Input)	1	5	20
Optoacopladores (Output)	2	5	70
Adicional	-	-	50
Total			240

Tabla 3.6: Consumo etapas de potencia

Elemento	Cantidad	Voltaje [V]	Corriente total [mA]
ULN2064	2	7	400
Motores PAP	2	7	2800
Adicional	-	-	300
Total			3500

Para cubrir estos requerimientos se optó por utilizar una doble fuente regulada. La primera fuente cuenta con dos salidas: una fija de 5V que alimenta a la unidad de control y la otra variable de 1.25V – 16V que alimenta a los motores paso a paso. Adicionalmente esta salida cuenta con un transistor que incrementa la corriente que brinda el regulador. La segunda fuente también de 5V es para uso exclusivo del adaptador de comunicaciones.

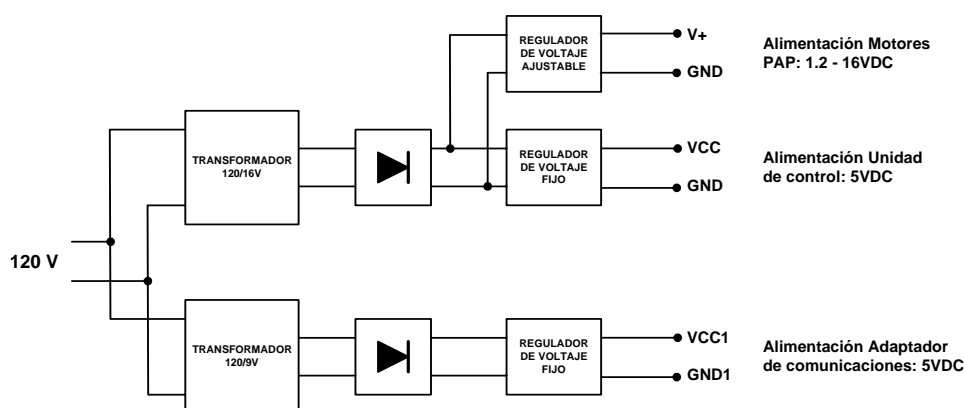


Figura 3.5: Diagrama de bloques de la fuente de alimentación

3.5.2 Motores paso a paso

Un motor paso a paso se diferencia de un motor convencional en que en este, se puede posicionar su eje en posiciones fijas o pasos, pudiendo mantener la posición. Esta peculiaridad es debida a la construcción del motor en sí, teniendo por un lado el rotor constituido por un imán permanente y por el otro el estator construido por bobinas, al alimentar estas bobinas se atraerá el polo del magnético opuesto del rotor con respecto al polo generado por la bobina y este permanecerá en esta posición atraído por el campo magnético de la bobina hasta que esta deje de generar el campo magnético y se active otra bobina haciendo avanzar o retroceder el rotor variando los campos magnéticos en torno al eje del motor y haciendo que este gire.

Los motores paso a paso pueden moverse confiablemente a la posición deseada siempre y cuando se le envíe el número de pulsos en la secuencia adecuada. Existen dos tipos de motores a paso a paso los unipolares y los bipolares.

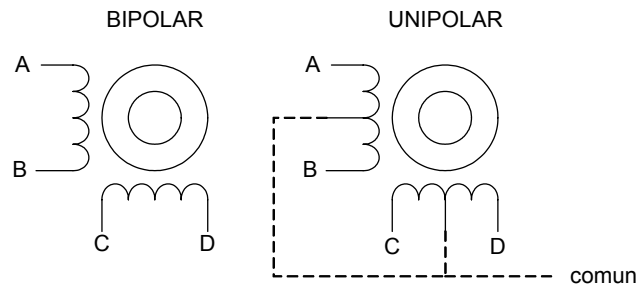


Figura 3.6: Tipos de motores paso a paso

3.5.2.1 Motores paso a paso unipolares

El motor unipolar normalmente dispone de 5 o 6 cables dependiendo si el común está unido internamente o no, para controlar este tipo de motores existen dos métodos con sus correspondientes secuencias de encendido de bobinas, el común irá conectado a V+ ó a tierra según el circuito de control usado y luego tan solo se tiene que alimentar la bobina correcta para que avance o retroceda el motor según avancemos o retrocedamos en la secuencia

Tabla 3.7: Secuencias de control de un motor unipolar

Paso Simple	A	B	C	D
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1

Medio Paso	A	B	C	D
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1

3.5.2.2 Motores paso a paso bipolares

Este tipo de motor lleva dos bobinados independientes el uno del otro, para controlar este motor se necesita invertir la polaridad de cada una de las bobinas en la secuencia adecuada.

Tabla 3.8: Secuencias de control de un motor bipolar

Paso Simple	A	B	C	D
1	V+	Gnd	V+	Gnd
2	V+	Gnd	Gnd	V+
3	Gnd	V+	Gnd	V+
4	Gnd	V+	V+	Gnd

3.5.2.3 Control de motores paso a paso

Para hacer girar los transformadores variables que ajustan las tensiones de pantalla se utilizan motores del tipo unipolar esto debido a la simplicidad de su control y además de poder conseguirse en el mercado nacional.

Como se menciono antes el control de los motores paso a paso lo realiza el PIC16F873A el cual genera la secuencia de pulsos necesarios para hacer girar los motores. La secuencia a utilizar será la de medio paso.

CAPÍTULO 4

PROTOTIPO DEL SISTEMA

El nuevo sistema de control ha sido instalado en el transmisor número tres del ROJ y se encuentra actualmente en operación. El prototipo desarrollado involucra la instalación de relés auxiliares para el control de los pulsadores ON/OFF de las etapas *PA* y *Driver*, además de la instalación de una estación remota para el control de la tensión de pantalla. Los módulos Acromag también han sido instalados en los paneles de control de las etapas *PA* de los tres transmisores.

4.1 Control de los pulsadores ON/OFF

4.1.1 Instalación de relés

Los relés de comando, indicación y conmutación de circuitos se han instalado, sobre rieles DIN tipo “T”, en el interior de los paneles de control de las etapas *PA* y *Driver*. Los planos actualizados que incluyen la instalación de todos los relés así como su conexión con el PLC se muestran en el Anexo A.

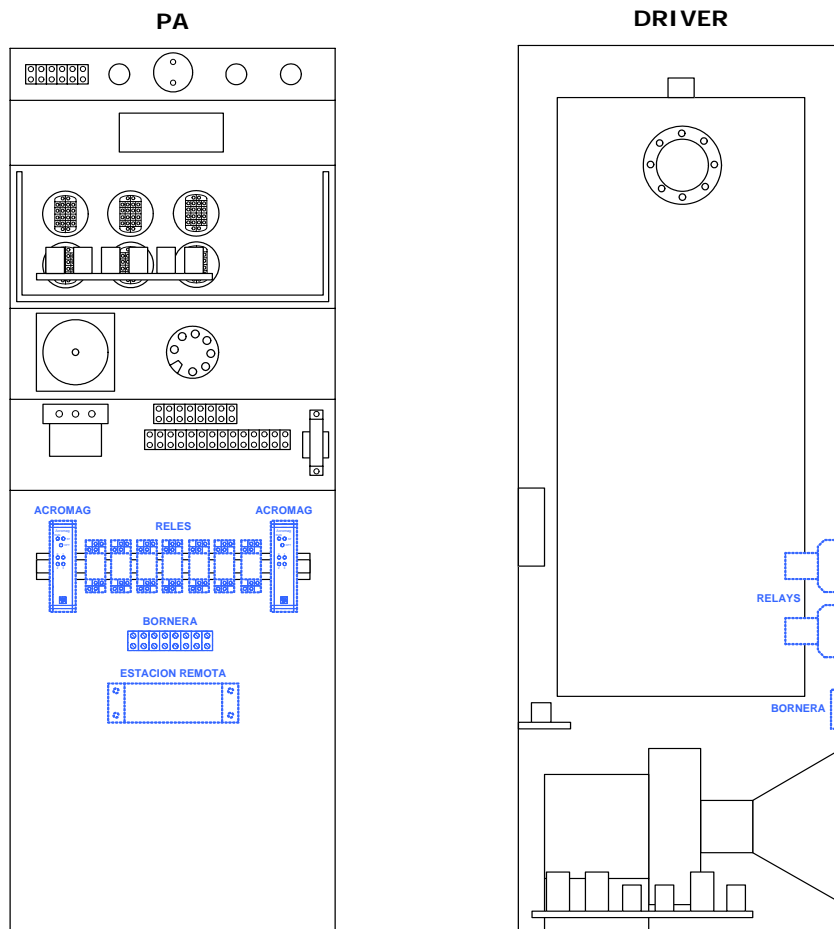


Figura 4.1: Ubicación de los relés en los paneles de control de *PA* y *Driver*.

La lista de las nuevas entradas y salidas del sistema se muestra en la Tabla 4.1. Estas señales se cablearon desde los paneles de control de ambas etapas hasta el PLC, los módulos del PLC utilizados fueron el D3-16TR (módulo de salidas tipo relé) y el D3-16NA (módulo de entradas de 110 VAC)

Tabla 4.1: Nuevas entradas y salidas

Tipo	Etiqueta	Descripción	Señal de campo
Salidas	PA3_on	Encendido de la etapa de salida (PA)	K32
	SCRPA3_on	Encendido de la tensión de pantalla de PA	K34
	SCRPA3_off	Apagado de la tensión de pantalla de PA	K35
	FIL_up	Incremento de la tensión de filamento	K36
	FIL_down	Decremento de la tensión de filamento	K37
	DRV3_on	Encendido de la etapa previa (Driver)	K41
	DRV3_off	Apagado de la etapa previa (Driver)	K42
	SCRDRV3_on	Encendido de la tensión de pantalla de Driver	K43
	SCRDRV3_off	Apagado de la tensión de pantalla de Driver	K44
Entradas	PA3_ok	Señal de encendido de PA	K38
	SCRPA3_ok	Señal de encendido de la tensión de pantalla de PA	K39
	DRV3_ok	Señal de encendido de Driver	K45
	SCRDRV3_ok	Señal de encendido de la tensión de pantalla de Driver	K46
	PA3_loc/rem	Posición del selector local/remoto de PA	K30
	DRV3_loc/rem	Posición del selector local/remoto de Driver	K40

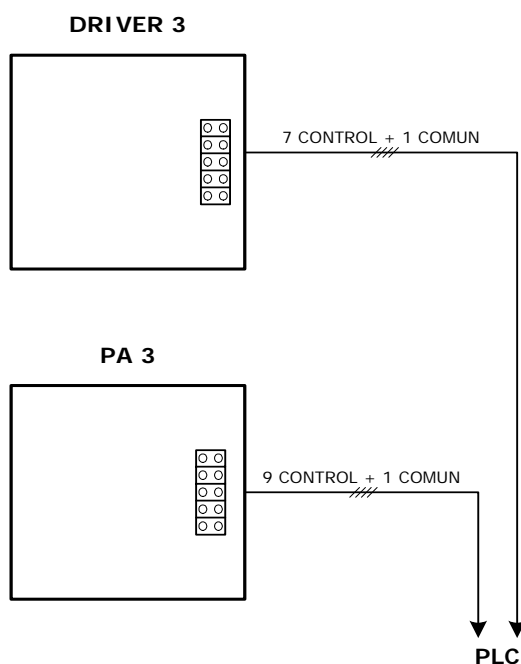


Figura 4.2: Cableado de las señales de control e indicación al PLC.

4.2 Prototipo de la estación remota

4.2.1 Diseño de los circuitos

La estación remota se ubica dentro del panel de control de la etapa *PA* del transmisor N° 3, como se mencionó en el capítulo tres la estación remota está formada por cuatro elementos de los cuales el adaptador de comunicaciones, las etapas de potencia y la unidad de control forman un solo circuito impreso mientras que la fuente de alimentación otro.

Debemos tener en cuenta que la alimentación del adaptador de comunicaciones está aislada de la alimentación del resto del circuito la razón de esto es por que la alimentación de la etapa de potencia es ruidosa debido a que está sujeta a caídas de voltaje producidas por la carga de los motores. Por otro lado el protocolo Modbus recomienda que el común del circuito de red se encuentre aislado y conectado a tierra en un solo punto del bus.

Los circuitos esquemáticos y los circuitos impresos de la estación remota se muestran en el Anexo B.

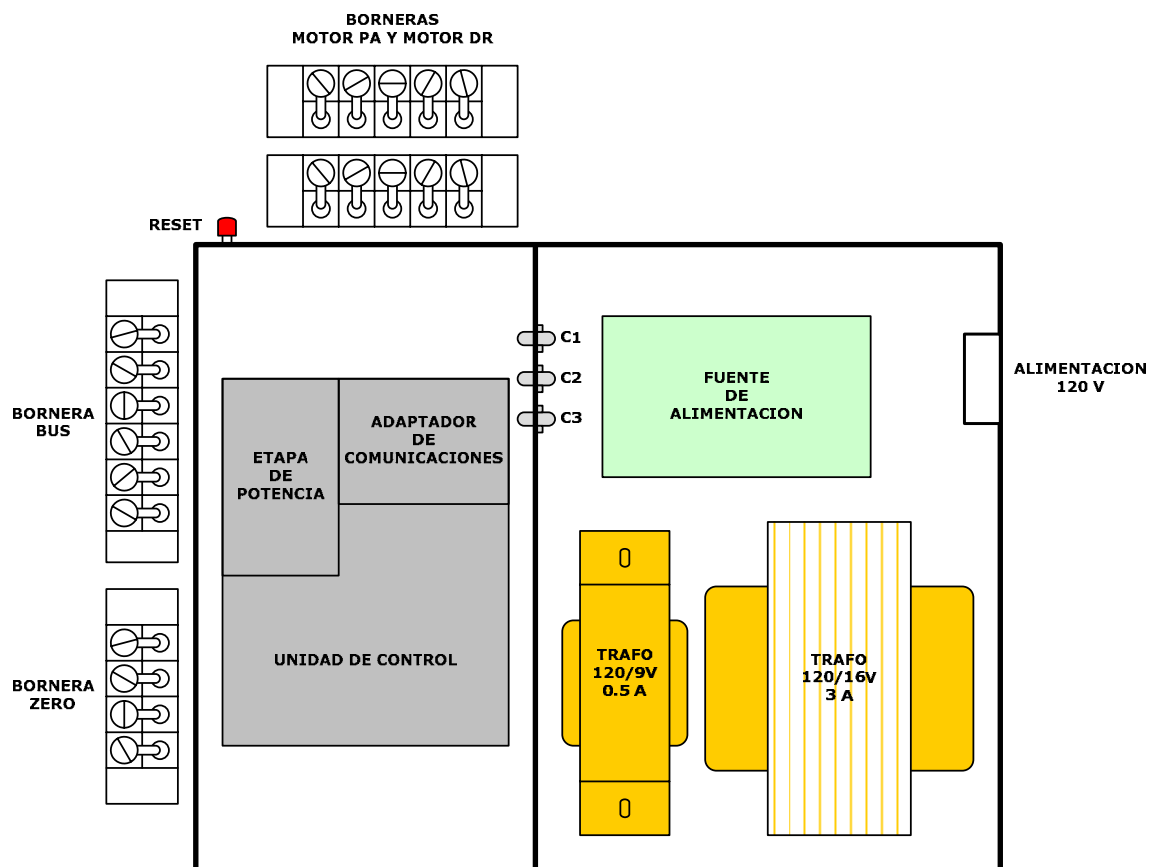


Figura 4.3: Prototipo de la estación remota.

4.2.2 Alojamiento mecánico y conexiones

Los circuitos impresos junto con los transformadores de la fuente de alimentación se han colocado en un alojamiento mecánico (caja de aluminio) para su protección. La conexión entre el circuito impreso de la fuente de alimentación y el otro circuito impreso se hace por medio de condensadores de paso, los que a su vez funcionan como filtros, y conectores del tipo molex.

La estación remota cuenta con un jack de alimentación estándar a la red eléctrica (120 VAC) y cuatro borneras. Además de un pulsador de reset para la reposición de la misma.

Tabla 4.2: Descripción de borneras de la estación remota

Bornera	Pines	Descripción
MOTOR PA	V+	Común del motor 1
	A	Bobina A del motor 1
	B	Bobina B del motor 1
	C	Bobina C del motor 1
	D	Bobina D del motor 1
MOTOR DR	V+	Común del motor 2
	A	Bobina A del motor 2
	B	Bobina B del motor 2
	C	Bobina C del motor 2
	D	Bobina D del motor 2
BUS	B	Línea B del bus RS485
	A	Línea A del bus RS485
	COM	Común del bus RS485
	B	Línea B del bus RS485
	A	Línea A del bus RS485
	COM	Común del bus RS485
ZERO	PA_ZERO 1	Indicación de motor 1 en posición cero
	PA_ZERO 2	
	DR_ZERO 1	Indicación de motor 2 en posición cero
	DR_ZERO 2	

4.2.3 Software del microcontrolador PIC

El microcontrolador PIC 16F873A cuenta con un contador de programa de 13 bits capaz de direccionar las 4096 palabras de 14 bits, de memoria de programa *flash* con la que cuenta. Este microcontrolador cuenta con un vector de reset y un vector de interrupción ubicados en las direcciones 0x00 y 0x04 (de la memoria de programa) respectivamente así cuando se produce un reset ó una interrupción el contador de programa apunta a la dirección respectiva y se ejecuta la instrucción ubicada en la misma^[9].

^[9] PIC16F87XA Data Sheet

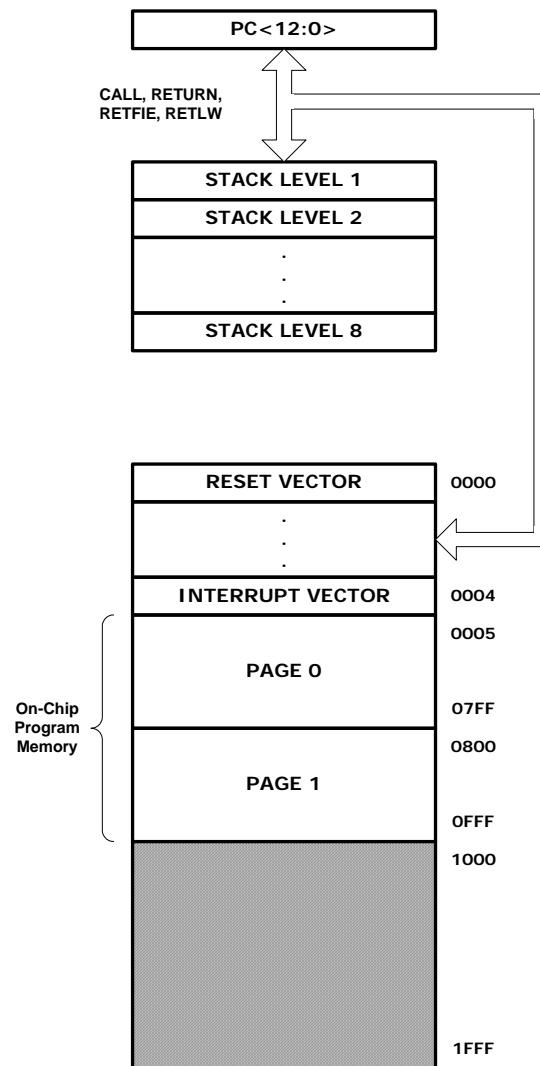


Figura 4.4: Memoria de programa y stack del PIC16F873A^[9].

La rutina principal de la estación remota consiste en un lazo cerrado, el cual primero determina el modo de operación de la estación remota (modo normal o configuración) y dependiendo de este se ejecutan las rutinas respectivas.

Se ha habilitado la interrupción por recepción en la USART, de esta manera cuando el microcontrolador PIC recibe un mensaje del PLC, el programa atiende la rutina de interrupción. Esta rutina se encarga de recibir los mensajes del PLC y verificar que se cumplan las condiciones necesarias para declarar que un mensaje es válido.

Si la dirección es correcta y no hay errores en la comunicación se procede a analizar el mensaje verificar y ejecutar, si es válida, la función MODBUS recibida. Finalmente un mensaje de respuesta es enviado al PLC. Cuando la dirección no es válida o se verifica que hay errores en el mensaje no se envía respuesta y se incrementa un contador de errores.

A continuación se describen las rutinas y subrutinas de las que se compone el programa así como sus respectivos diagramas de flujo, el código fuente del programa se encuentra en el Anexo C

^[9] PIC16F87XA Data Sheet

4.2.3.1 Rutina de Inicio “RESET”

La rutina de reset se encarga de configurar los puertos del microcontrolador como entrada ó salida, leer parámetros de configuración de la memoria EEPROM como son: la dirección de esclavo de la estación remota, la velocidad de la comunicación y el retardo de respuesta. Este último es un retardo adicional que la estación remota espera antes de enviar un mensaje de respuesta.

Tabla 4.3: Puertos utilizados en el microcontrolador PIC

Puerto	Pin	Tipo	Descripción
Puerto A	RA0	Salida	Indicador de PIC funcionando, led ON
	RA1	Entrada	Indicador de motor PA en posición cero
	RA2	Entrada	Indicador de motor DRV en posición cero
	RA3	Entrada	Interruptor para determinar modo de operación
	RA4	Entrada	Pulsador para cargar datos por defecto, DFLT
	RA5	Salida	Indicador del modo de operación, led MODE
Puerto B	RB0	Salida	Alimentación de motor PA
	RB1	Salida	Alimentación de motor PA
	RB2	Salida	Alimentación de motor PA
	RB3	Salida	Alimentación de motor PA
	RB4	Salida	Alimentación de motor DRV
	RB5	Salida	Alimentación de motor DRV
	RB6	Salida	Alimentación de motor DRV
	RB7	Salida	Alimentación de motor DRV
Puerto C	RC5	Salida	Habilitación de la transmisión de datos por el bus RS485
	RC6	Salida	Pin de transmisión de datos
	RC7	Entrada	Pin de recepción de datos

Tabla 4.4: Registros utilizados de la memoria EEPROM del microcontrolador PIC

Registro	Descripción	Valor
0x00	Dirección de esclavo de la estación remota Default = 127	1 – 127
0x01	Baud Rate Default = 1, 9600 bps	0 = 4800 bps 1 = 9600 bps 2 = 19200 bps
0x02	Retardo de respuesta Default = 0, Sin retardo	0 – 200 ticks (1 tick \approx 10 μ s)
0x03	Variable de Reset Default = 0x00	0x00 – 0xFF

Una vez leídos los datos de la memoria EEPROM se procede a configurar la USART y el TIMER0 el cual se utiliza, para calcular el retardo entre los mensajes recibidos (3.5 veces la longitud de un byte). En la rutina de reset también se habilita la interrupción por recepción en la USART y se asignan los valores iniciales a las variables del programa.

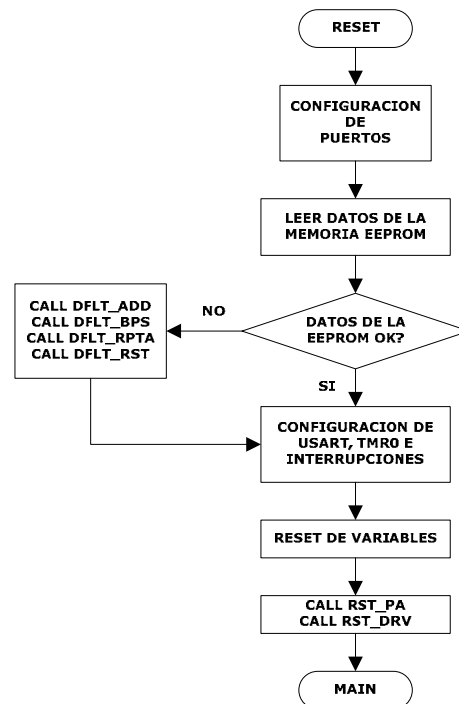


Figura 4.5: Rutina de reset.

Cuando los datos de la EEPROM se encuentran fuera de los rangos permitidos se llama a las subrutinas “DFLT_ADD”, “DFLT_BPS”, “DFLT_RPTA” y “DFLT_RST”, las cuales se encargan de cargar los valores por defecto de la dirección de esclavo, la velocidad de comunicación, retardo de respuesta y variable de reset respectivamente.

Finalmente una vez concluida la etapa de configuración se ejecutan dos subrutinas que verifican la posición física de los motores paso a paso, por medio de los pines 1 y 2 del puerto “A”, y los hace girar hasta la posición cero. Por seguridad estas subrutinas sólo pueden regresar a cada motor 160 pasos, después de los cuales sale de la subrutina

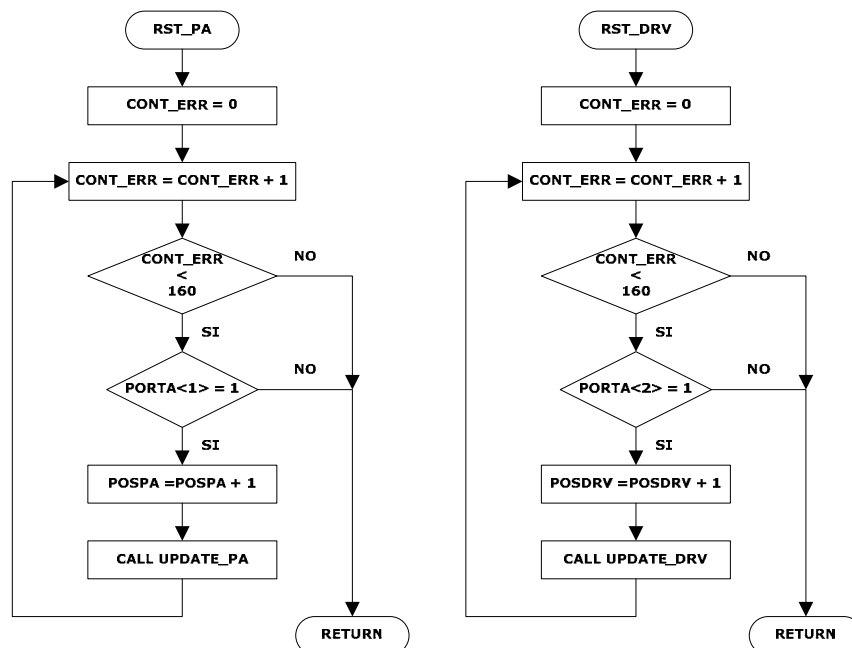


Figura 4.6: Subrutinas de reset de los motores paso a paso.

4.2.3.2 Rutina principal “MAIN”

La rutina principal consiste en el lazo cerrado mencionado anteriormente, primero se verifica si es que se ha pulsado el pulsador “DFLT” si es así se llaman a las subrutinas que cargan los parámetros de configuración por defecto y se salta a la rutina de reset. De lo contrario se salta a la rutina principal para luego determinar el modo de operación por medio del pin 3 del puerto “A”.

En modo normal se apaga el led “MODE” y el bucle compara la posición actual de cada uno de los motores paso a paso con la respectiva posición enviada por el PLC. En caso de que las posiciones sean diferentes se procede a ejecutar las subrutinas de actualización de la posición de cada motor. En este modo las funciones Modbus se ejecutan sobre la memoria RAM del microcontrolador PIC.

En modo configuración el lazo de la rutina principal enciende el led “MODE” (para indicar el modo configuración) y se activa una bandera para indicar que las funciones Modbus se ejecuten sobre la memoria EEPROM del microcontrolador PIC.

En ambos modos de operación se verifica que el valor de la variable RESET_PIC (mapeada en la RAM y la EEPROM del PIC) coincida con el valor cero, de lo contrario se ejecutará la rutina de RESET.

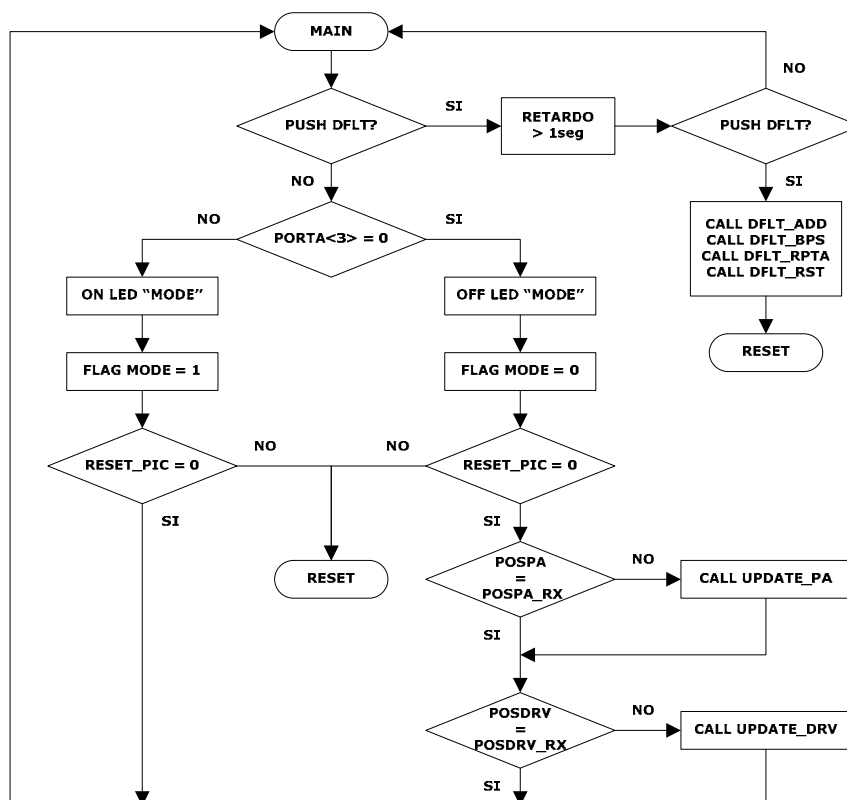


Figura 4.7: Rutina principal.

En la rutina principal se llaman a las subrutinas “UPDATE_PA” y “UPDATE_DRV” estas subrutinas se encargan de generar los pulsos necesarios para hacer girar los motores paso a paso y así alcanzar la posición deseada. La posición de cada motor se guarda en dos variables POSPA Y POSDRV, que son las que utiliza el programa principal para

determinar si la posición actual de cada motor corresponde a las posiciones enviadas por el PLC las cuales se encuentran en las variables POSPA_RX y POSDRV_RX.

El control que se realiza sobre los motores es a medio paso por lo que se tienen ocho secuencias de pulsos o pasos, estas secuencias de pulsos se encuentran en dos tablas de bytes una para el motor de la etapa de salida *PA* y otra para el motor de la etapa previa *Driver*.

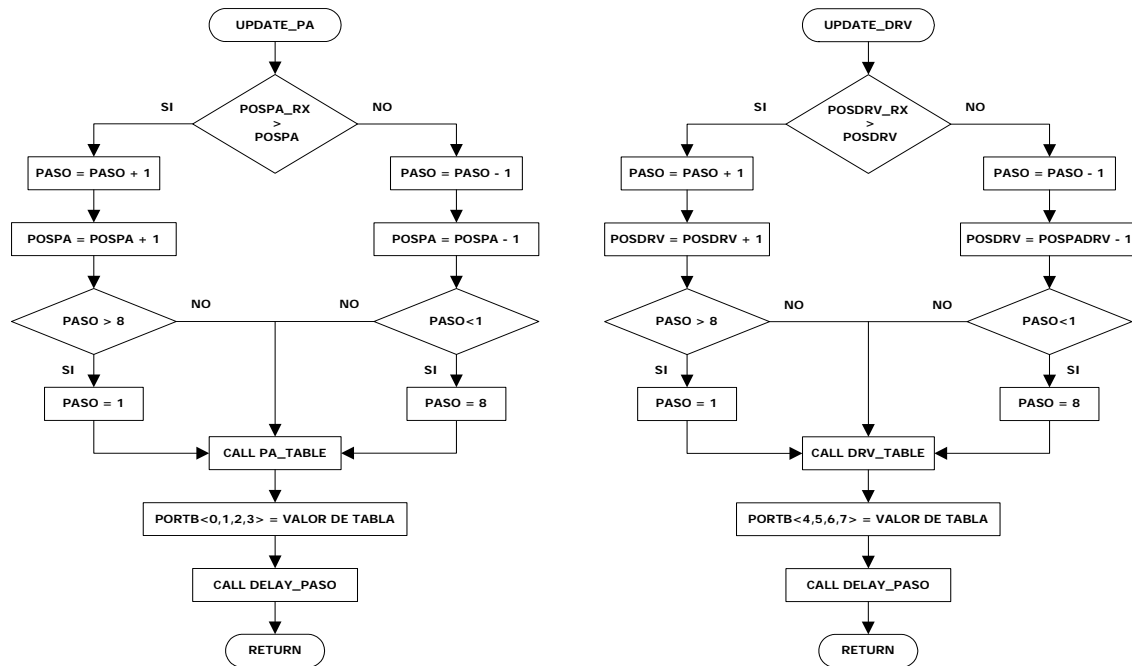


Figura 4.8: Subrutinas de actualización de los motores paso a paso.

4.2.3.3 Rutina de interrupción “INTERRUPT”

Como se menciono antes se ha habilitado la interrupción por recepción en la USART, por lo que cada vez que se recibe un byte el contador de programa apunta al vector de interrupción y desde aquí se salta a la rutina de interrupción por medio de la instrucción GOTO INTERRUPT.

La rutina de interrupción primero debe determinar las condiciones necesarias para declarar que un mensaje es valido estas condiciones son las siguientes:

- ✓ Comprobar que el tiempo transcurrido desde que se recibió el último mensaje sea de al menos 3.5 veces el tamaño de un byte esto se hace verificando si ha ocurrido un desborde en el TMR0.
- ✓ Leer el primer byte del mensaje y verificar que se trata de su dirección de esclavo.
- ✓ Determinar si hay errores en la comunicación calculando el CRC del mensaje.

Si no se cumple cualquiera de las condiciones anteriores el mensaje se descarta, no se envía ninguna respuesta y se retorna a la rutina principal. Si las condiciones se cumplen entonces se procede a analizar el mensaje y verificar que los datos Modbus están soportados por la estación remota en cuyo caso se procede a ejecutar la función Modbus y enviar una respuesta.

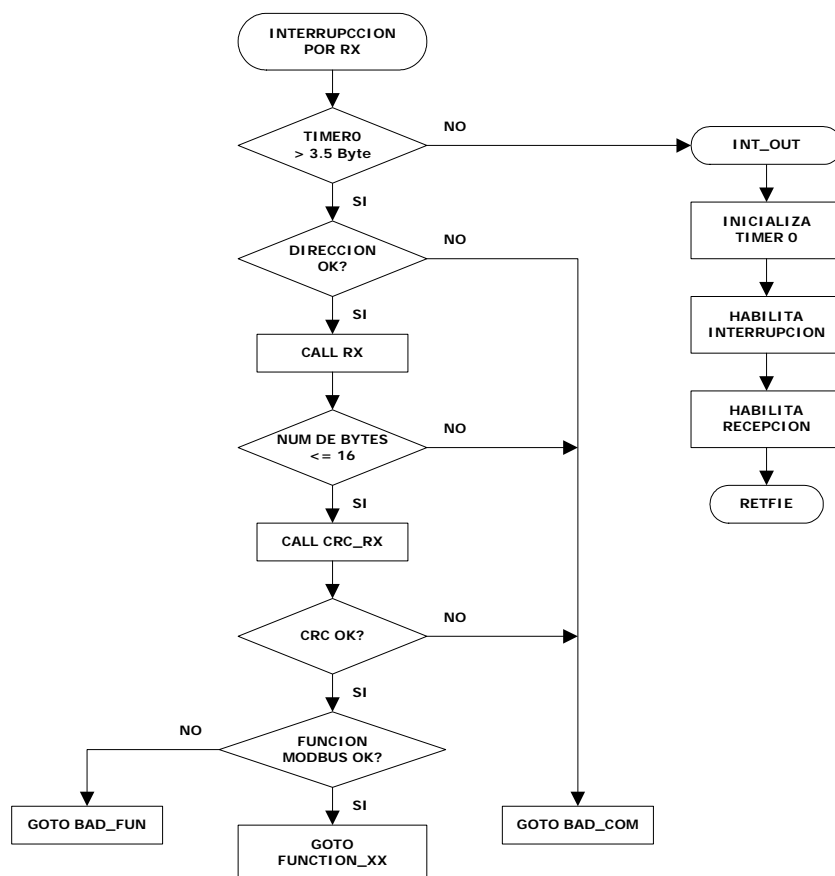


Figura 4.9: Rutina de interrupción

Tabla 4.5: Códigos Modbus implementados

Código	Función	Referencia	Función en el PIC	Registros en el PIC
03 (0x03)	<i>Read Holding Registers</i>	40001 – 40128	Leer uno ó varios registros de la EEPROM	0x00 – 0x7F
		40001 - 40078	Leer uno ó varios registros de la RAM	0x50 – 0x7F
16 (0x10)	<i>Write Multiple Registers</i>	40001 – 40128	Escribir en uno ó varios registros de la EEPROM	0x00 – 0x7F
		40001 - 40078	Escribir en uno ó varios registros de la RAM	0x50 – 0x7F

La subrutina “RX” se utiliza para recibir el mensaje byte a byte a intervalos fijos de tiempo (retardo entre bytes igual a 1.5 veces la longitud de un byte) cada byte se guarda en una variable de la memoria de datos RAM, se ha previsto un buffer de 16 bytes en el microcontrolador PIC por lo que si el mensaje contiene más de 16 bytes este es descartado, se incrementa el contador de mensajes con errores en la comunicación y se vuelve al programa principal sin enviar ninguna respuesta.

Dado que el microcontrolador PIC cuenta con una interfase de comunicaciones USART implementada en hardware la recepción de bytes es bastante sencilla simplemente se espera la activación de una bandera (RCIF del registro PIR1) y se lee el registro RCREG el cual contiene el byte recibido^[11].

^[11] Asynchronous Communications with the PICmicro® USART

Luego de recibir el mensaje se calcula el CRC del mensaje recibido por medio de la subrutina “CRC_RX”. El CRC contiene un valor binario de 16 bits implementado en dos bytes de 8 bits: CRC_Hi y CRC_Lo. Los bytes del CRC calculados se comparan con los bytes del CRC del mensaje recibido; de no ser iguales, el mensaje se descarta y el contador de mensajes con errores en la comunicación se incrementa. En esta situación no se envía ningún mensaje de respuesta y se regresa a la rutina principal.

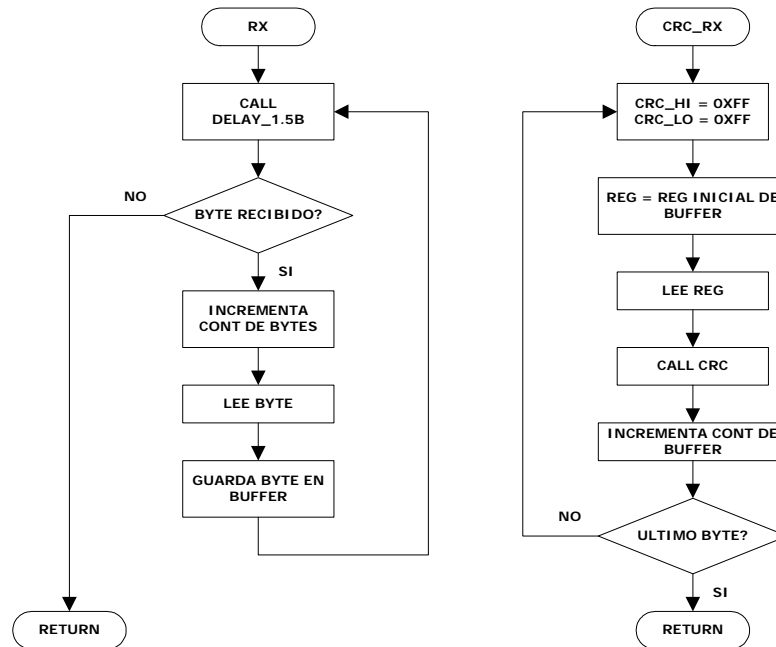


Figura 4.10: Subrutinas RX y CRC_RX

El CRC también se calcula sobre los mensajes de respuesta que genera la estación remota, en esta situación los bytes del CRC son anexados al final del mensaje enviando primero el byte menos significativo (CRC_Lo) seguido por el byte más significativo (CRC_Hi).

Para calcular el CRC de un mensaje primero se pone a uno los bytes de las variables CRC_HI y CRC_LO luego se utiliza la subrutina “CRC” con cada uno de los bytes que conforman el mensaje. En la subrutina “CRC” cada byte de 8 bits es operado con los registros del CRC (CRC_HI y CRC_LO) mediante una operación de OR exclusivo, luego el resultado es desplazado en la dirección del bit menos significativo (LSB). La posición del bit más significativo (MSB) es llenada con un cero y el LSB extraído es analizado, si es un uno el resultado del desplazamiento es operado con un valor fijo, mediante una operación de OR exclusivo. Si el LSB es un cero no se realiza ninguna operación. El proceso se repite hasta completar ocho desplazamientos^[3]

El contenido final de CRC_HI y CRC_LO después de ejecutar la subrutina “CRC” con todos los bytes del mensaje, contiene el valor del CRC del mensaje.

^[3] Specification & Implementation guide for MODBUS over serial line

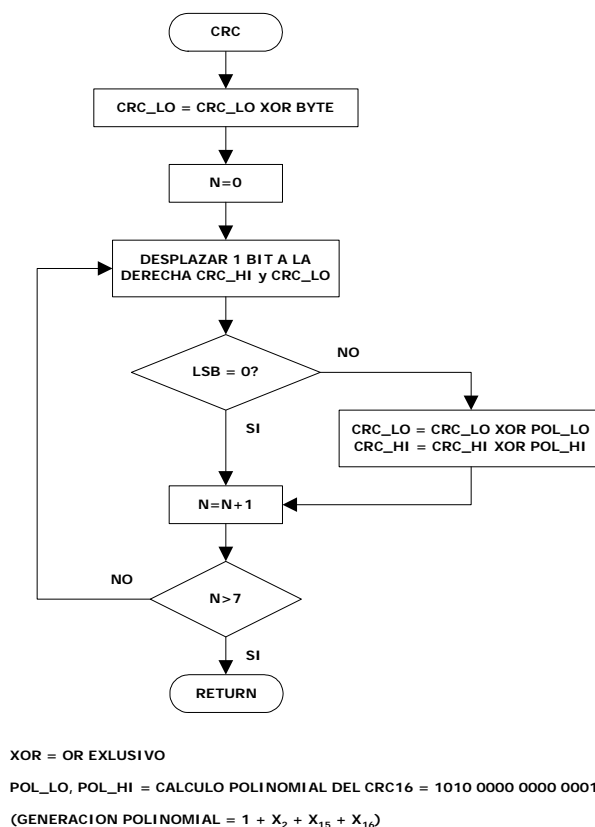


Figura 4.11: Subrutina para generación del CRC

4.2.3.4 Rutina para la función 0x03 “FUNCT_03”

Esta función se utiliza para leer el contenido de las memorias de datos del microcontrolador PIC, la solicitud de esta función especifica el registro inicial y la cantidad de registros a ser leídos, debido a que el protocolo Modbus utiliza registros de 16 bits el byte menos significativo lo forma el registro del microcontrolador PIC y el más significativo siempre es un 0x00.

Cuando la estación remota opera en modo configuración, la función 0x03 lee los registros de la memoria EEPROM del microcontrolador PIC, el cual está formado por un solo banco de 128 bytes

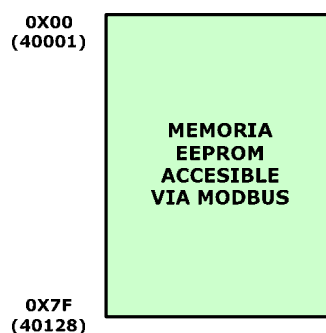


Figura 4.12: Memoria EEPROM del microcontrolador PIC

En modo normal la función 0x03 lee el contenido de la memoria RAM del microcontrolador PIC, la cual está formada por cuatro bancos de 128 *bytes*, estos a su vez contienen a los registros de funciones especiales (SFR) utilizados para la configuración del

microcontrolador PIC y los registros de propósito general (GPR), los cuales se utilizan para almacenar variables en el desarrollo del programa.

La función 0x03 puede leer los registros entre la dirección 0x0050 y 0x007F pertenecientes al grupo de registros de propósito general del banco 0 de la memoria RAM, por lo que la referencia Modbus 40001 es direccionada internamente al registro 0x0050 del microcontrolador PIC.

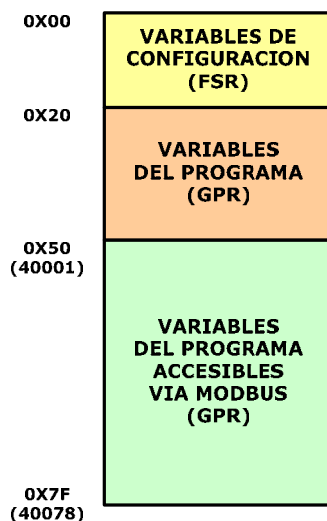


Figura 4.13: Banco 0 de la memoria RAM del microcontrolador PIC

Tabla 4.6: Mensajes de solicitud, respuesta y error para la función 0x03

Función	1 byte	0x03
Registro inicial	2 bytes	0x0000 – 0x007F (EEPROM) 0x0000 – 0x004D (RAM)
Cantidad de registros (N)	2 bytes	1 – 16

Función	1 byte	0x03
Contador de registros	1 bytes	2N
Valor(es) del(os) registro(s)	2N bytes	

Función de error	1 byte	0x83
Código de error	1 byte	0x01 ó 0x02 ó 0x03

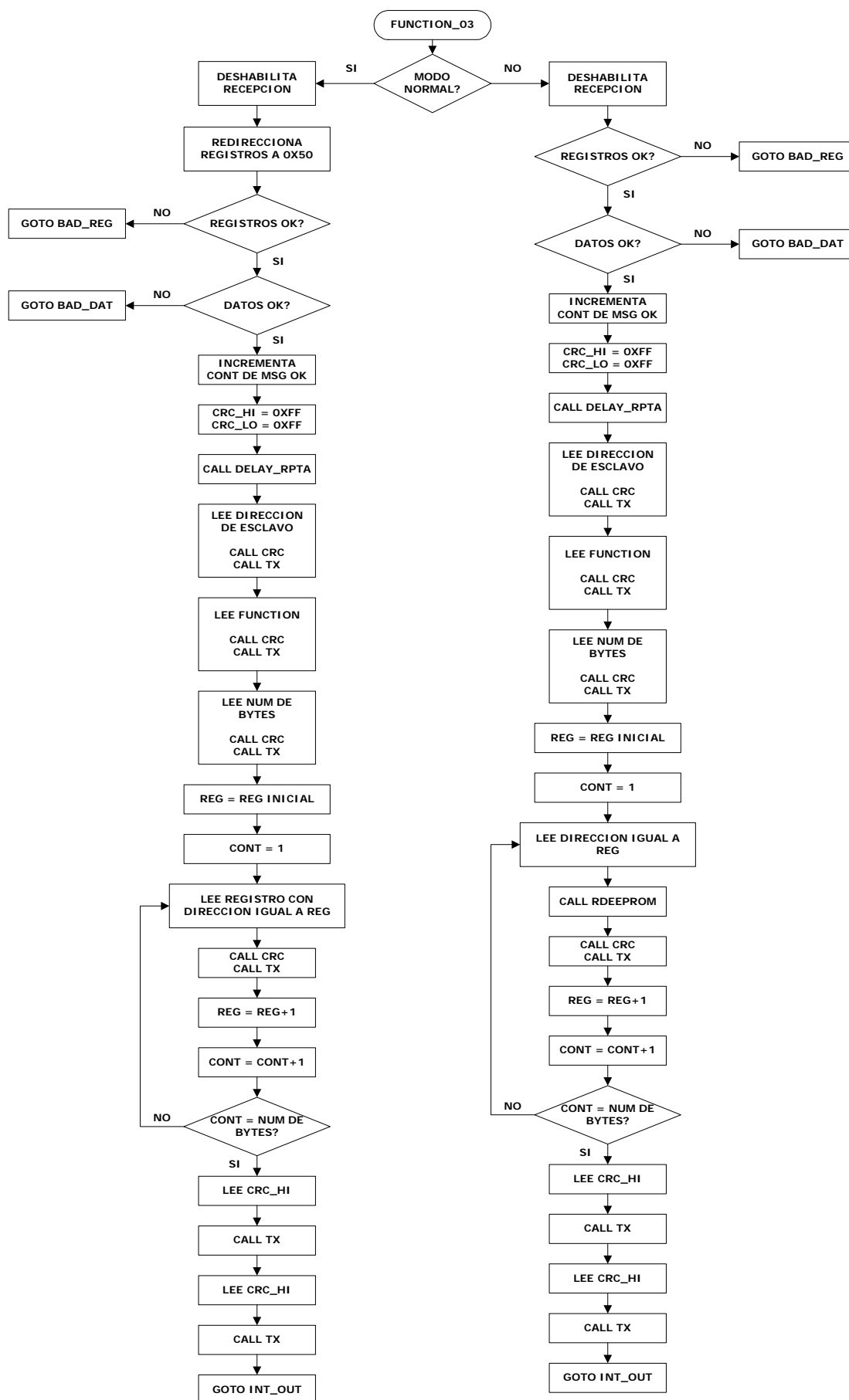


Figura 4.14: Rutina para la función 0x03

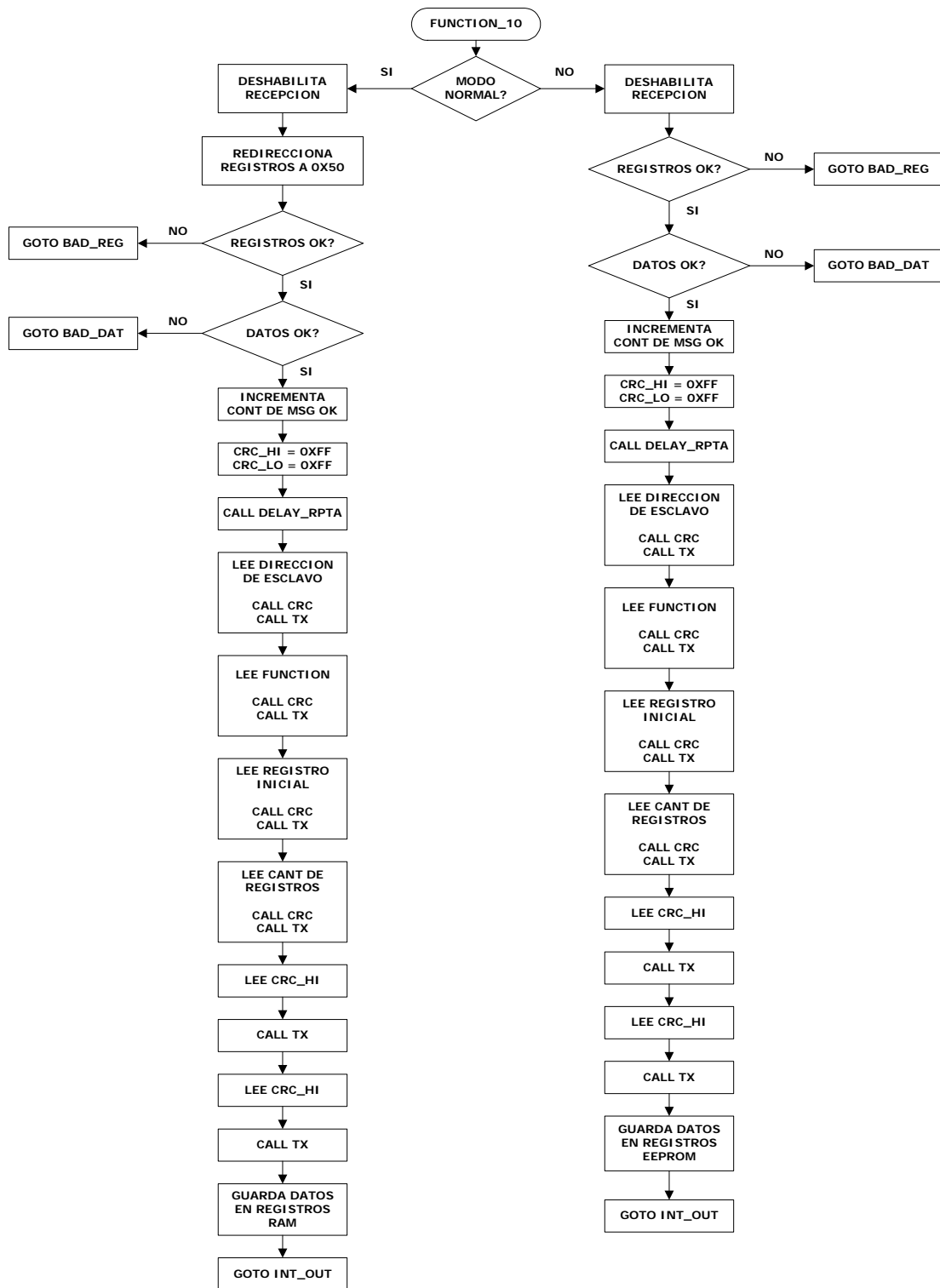


Figura 4.15: Rutina para la función 0x10

4.2.3.5 Rutina para la función 0x10 “FUNCT_10”

Cuando la estación remota se encuentra en modo configuración la función 0x10 se utiliza para escribir en una o más variables de la memoria EEPROM, mientras que en modo normal, se utiliza para escribir en la memoria RAM del microcontrolador PIC. El mensaje de solicitud de esta función especifica el registro inicial desde donde se va a escribir, un

contador de bytes y los valores de los registros a escribir. Los valores de cada registro son enviados en dos bytes, pero el PIC solo utiliza el byte menos significativo.

En modo normal al igual que en la función 0x03 solo los registros entre 0x0050 y 0x007F del grupo de registros de propósito general del banco cero de la memoria RAM, se pueden escribir vía Modbus.

Tabla 4.7: Mensajes de solicitud, respuesta y error para la función 0x10

Función	1 byte	0x10
Registro inicial	2 bytes	0x0000 – 0x004D
Cantidad de registros (N)	2 bytes	1 – 4
Contador de bytes	1 byte	2N
Valor(es) del(os) registro(s)	2N bytes	

Función	1 byte	0x10
Registro inicial	2 bytes	0x0000 – 0x004D
Cantidad de registros (N)	2 bytes	1 – 4

Función de error	1 byte	0x90
Código de error	1 byte	0x01 ó 0x02 ó 0x03

4.2.3.6 Rutina de respuesta de excepción Modbus “EXCEPTION”

Cuando la estación maestra (PLC) envía un mensaje de solicitud a la estación remota, espera un mensaje de respuesta normal, pero cualquiera de las siguientes situaciones es posible:

- ✓ Si la estación remota recibe un mensaje de solicitud sin errores en la comunicación y puede manejar la solicitud normalmente, entonces se envía un mensaje de respuesta normal.
- ✓ Si la estación remota recibe un mensaje de solicitud con errores en la comunicación (CRC diferente) entonces no se envía ninguna respuesta.
- ✓ Si la estación remota recibe un mensaje de solicitud sin errores en la comunicación pero no puede manejar la solicitud (por ejemplo la solicitud pide leer un registro no valido) entonces se envía un mensaje de excepción Modbus y se incrementa el contador de mensajes Modbus inválidos.

DIRECCION	FUNCION+0x80	CODIGO DE EXCEPCION	CRC
1 BYTE	1 BYTE	1 BYTE	2 BYTES

Figura 4.16: Mensaje de excepción Modbus

Tabla 4.8: Posibles códigos de excepción Modbus

Código	Nombre
01	Función inválida
02	Dirección de registro inválido
03	Valores de registros inválidos

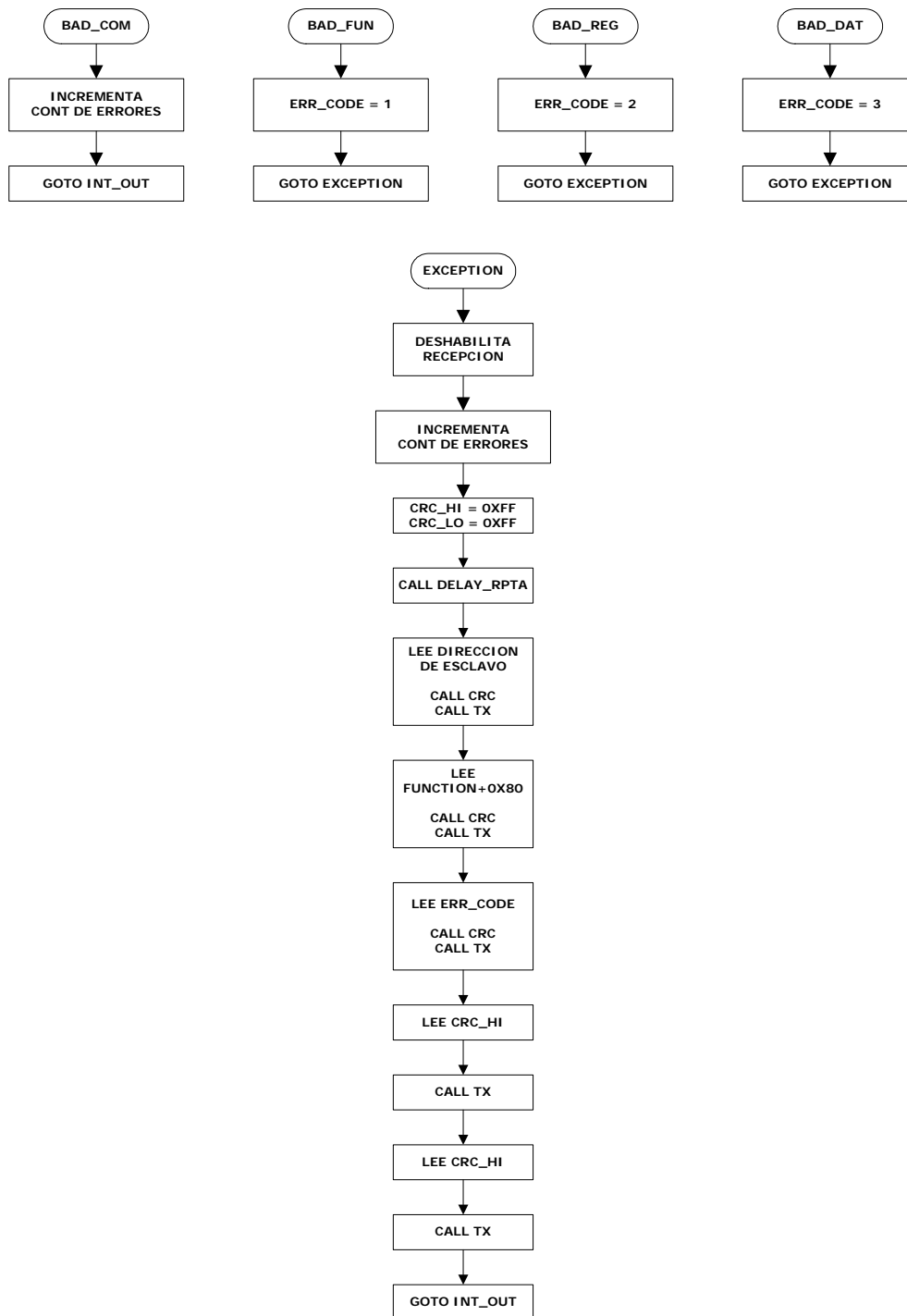


Figura 4.17: Rutina de excepción Modbus

Tanto las rutinas de las funciones Modbus como la de excepción utilizan la subrutina de transmisión TX la cual se encarga de enviar un byte por la interfase USART. Al igual que la subrutina de recepción la transmisión de un byte es sencilla basta con cargar el byte en el registro TXREG y esperar a que se active una bandera (TRMR del registro TXSTA) que indica que el byte ha sido enviado^[11].

^[11] Asynchronous Communications with the PICmicro® USART

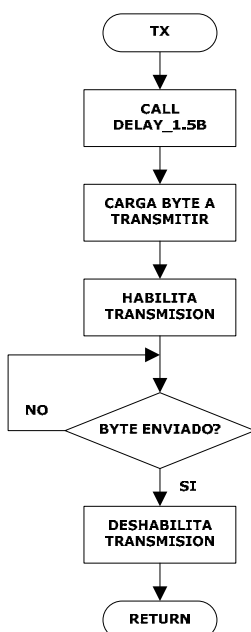


Figura 4.18: Subrutina de transmisión

4.2.3.7 Otras subrutinas

En el desarrollo del programa se hacen uso de otras subrutinas las cuales se mencionan a continuación:

- ✓ RDEEPROM: subrutina utilizada para leer un registro de la memoria de datos EEPROM.
- ✓ WREEPROM: subrutina utilizada para escribir en un registro de la memoria de datos EEPROM.
- ✓ DELAY_1.5B: retardo de 1.5 veces un byte, utilizado en la recepción y transmisión de un mensaje.
- ✓ DELAY_RPTA: retardo adicional que la estación remota espera antes de enviar un mensaje de respuesta. Es variable dependiendo del dato guardado en el registro 0x03 de la memoria EEPROM.
- ✓ DELAY_100: retardo de 100 μ s utilizado para compensar el tiempo consumido en el cálculo del CRC.
- ✓ DELAY_STEP: retardo de 100 ms utilizado en las rutinas de actualización de los motores a pasos.

4.3 Montaje de los motores paso a paso

Para transmitir el movimiento de los motores paso a paso a los transformadores variables que ajustan las tensiones de pantalla se utilizan juegos de engranajes, para el montaje se utilizaron bases de aluminio a las cuales se fijan los motores. Un esquema del montaje se muestra en las figuras 4.19 y 4.20.

Tabla 4.9: Características de los motores paso a paso

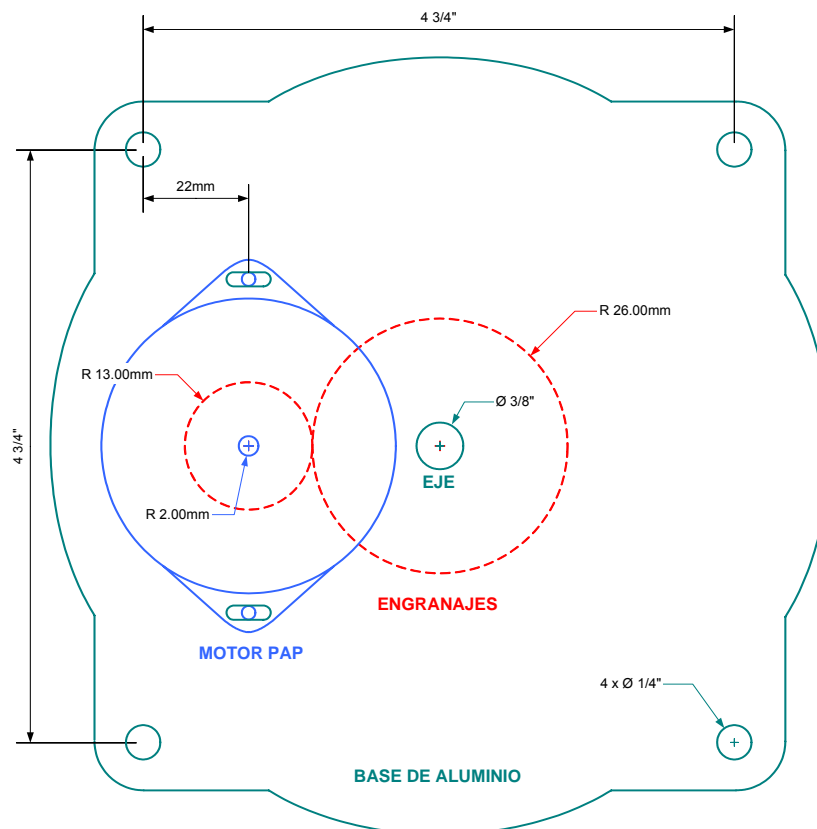
Tipo	Unipolar
Angulo de paso	7.5°
Numero de pasos por vuelta	48
Resistencia	5.1 Ω
Tensión	3.7VDC – 7.5 VDC
Torque	0.15 N.m

Para mejorar la resolución de los motores los juegos de engranajes tienen una relación de velocidad de 2:1 con ello se logra un ángulo de paso de 3.75° ó lo que es lo mismo 96 pasos por vuelta. Dado que el tipo de control es a medio paso finalmente se obtiene una resolución de 192 pasos por vuelta.

Tabla 4.10: Características de los engranajes

Juego N° 1 (PA)	Módulo	1
	N° de dientes del piñón	25
	N° de dientes de la rueda	50
Juego N° 2 (Driver)	Módulo	1
	N° de dientes del piñón	34
	N° de dientes de la rueda	68

Fue necesario en ambos casos (etapa *PA* y etapa *Driver*) cambiar los ejes de los transformadores variables por unos más largos en los cuales se puedan fijar los engranajes, la fijación de los engranajes a los ejes respectivos se hace con prisioneros.

Figura 4.19: Montaje del motor paso a paso con el transformador variable de *PA*

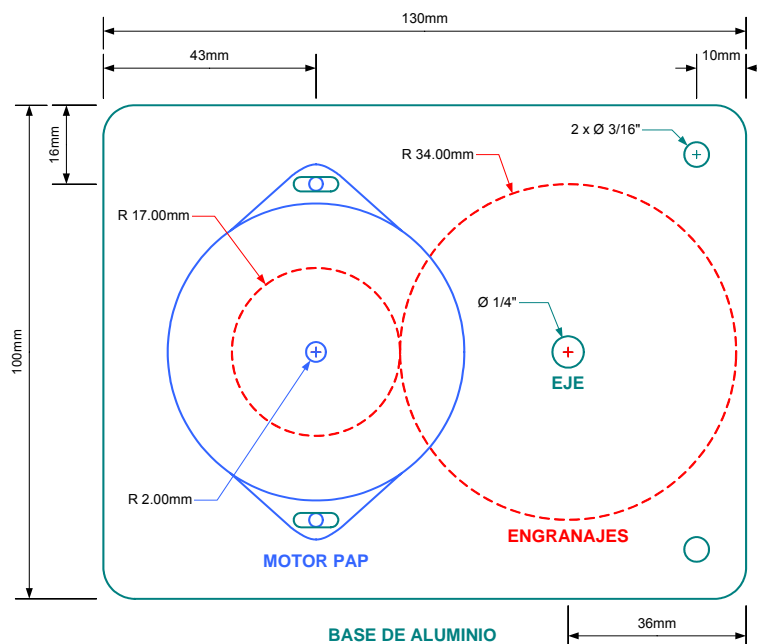


Figura 4.20: Montaje del motor paso a paso con el transformador variable de *Driver*

4.4 Instalación de los módulos Acromag

Para monitorear las corrientes de operación del transmisor se utilizó el modelo 914MB de la marca *Acromag* estos módulos cuentan con cuatro canales analógicos configurables a distintos rangos.

Estos módulos cuentan con sujetadores para su montaje en rieles DIN tipo “T” por lo que han sido instalados junto con los relés auxiliares, en los paneles de control de las etapas de salida *PA* de los tres transmisores que se encuentran actualmente en operación. Los módulos también han sido conectados a la red de comunicaciones RS485 y a una fuente de alimentación de 24 VDC ubicada en el rack del PLC.

Tabla 4.11: Corrientes leídas por los módulos Acromag

Etapa	Corriente	Unidad
PA	VAC ION	[μ A]
	Bias	[A]
	Pantalla	[A]
Driver	Bias	[A]
	Cátodo	[mA]
	Pantalla	[mA]

Se han utilizado un total de dos módulos por transmisor uno para monitorear las corrientes de la etapa de salida *PA* y el otro para las corrientes de la etapa previa *Driver*, las direcciones de esclavo correspondientes a cada módulo dependen del transmisor al que pertenece.

Tabla 4.12: Direcciones de esclavo de los módulos Acromag

Transmisor	Etapas	Dirección de esclavo
TX1	PA	02
	Driver	03
TX2	PA	04
	Driver	05
TX3	PA	06
	Driver	07
TX4 (*)	PA	08
	Driver	09

* Transmisor en construcción

Las señales de las corrientes de ambas etapas se han tomado directamente de los medidores ubicados en los paneles de control ya que estos están conectados a los terminales de las resistencias en derivación (*Shunt*) con las que cuentan las respectivas fuentes de alimentación.

Para no cargar el medidor analógico (y forzarlo a mostrar una lectura errónea) es necesario que la impedancia de entrada del instrumento colocado en los terminales de la resistencia en derivación sea alta. En el caso de los módulos 914MB de Acromag las entradas analógicas cuentan con una impedancia de entrada de $110\text{K}\Omega$ ^[2] por lo que comparada con las resistencias *shunt* de cada una de las fuentes no habría problema.

Tabla 4.13: Escala de corrientes

Etapas	Corriente	R _{SHUNT}	Rango de corriente	Rango de voltaje
PA	Vac Ion	51.00 Ω	0 – 5 μA	0 – 0.00025 V
	Bias	0.100 Ω	0 – 0.5 A	0 – 0.05 V
	Pantalla	0.050 Ω	0 – 1.0 A	0 – 0.05 V
Driver	Bias	0.250 Ω	0 – 0.2 A	0 – 0.05 V
	Cátodo	0.025 Ω	0 – 2.0 A	0 – 0.05 V
	Pantalla	0.250 Ω	0 – 0.2 A	0 – 0.05 V

En la tabla 4.13 se muestran los rangos de voltaje necesarios para la lectura de corrientes por lo que los módulos Acromag han sido configurados en el rango de 0 – 78 mV con el cual se obtiene una resolución de 3.9 μV .

^[2] BusWorks® Model 913MB/914MB Network I/O Module User's Manual

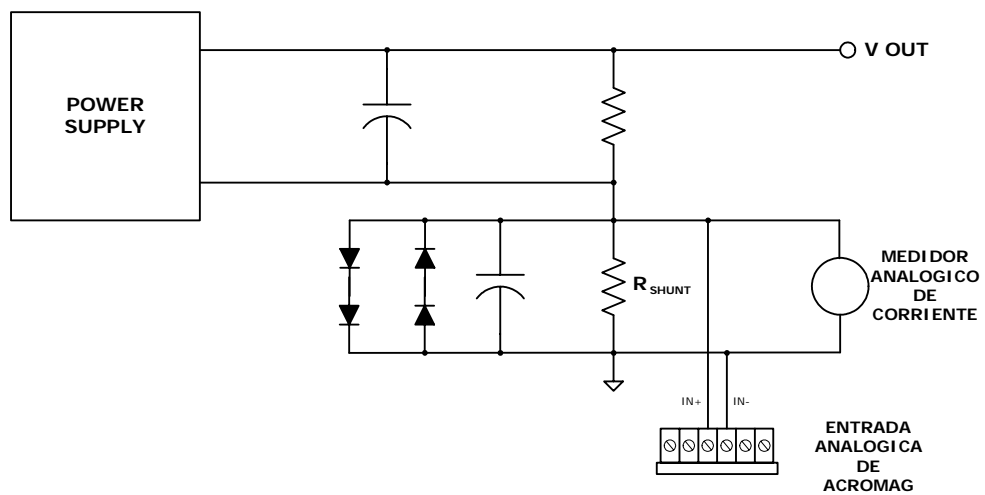


Figura 4.21: Conexión de las entradas analógicas de los módulos Acromag

CAPÍTULO 5

SOFTWARE DEL PLC Y DEL SISTEMA SCADA

Como se menciona en el capítulo uno el sistema de control de los transmisores del radar del ROJ cuenta con un PLC como elemento principal de control al cual se conectan todas las señales (entradas y salidas del sistema). Además cuenta con una PC como elemento auxiliar del sistema mediante el se supervisa y comanda el funcionamiento del sistema a través del sistema SCADA.

El sistema cuenta también con un panel de control local conformado por pulsadores e indicadores luminosos, este panel sirve como interfase local para los operadores, a través del cual se dan las órdenes de inicio y parada del sistema. Este panel permite seleccionar el modo de funcionamiento del sistema: Local y Remoto:

- ✓ Modo Local.- En este modo el comando del sistema lo hacemos desde el panel de control local, en este modo la PC sólo es un elemento de supervisión monitoreando los estados de los parámetros de funcionamiento del sistema y la medición de señales analógicas.
- ✓ Modo Remoto.- En este modo el comando del sistema lo hacemos desde la PC el cual se comporta también como un elemento de supervisión. En este modo, el sistema SCADA, permite a su vez trabajar en Modo Operación o en Modo Prueba, el primero es el modo de funcionamiento normal del sistema y el segundo permite trabajar independientemente cada sistema con el propósito de hacer pruebas.

Si bien desde la PC se pueden enviar órdenes de parada y de marcha, la PC no realiza ninguna clase de control, el control radica únicamente en el PLC.

5.1 Software del PLC

El PLC cuenta con el CPU DL350 este módulo ejecuta el programa de aplicación, la programación del CPU se puede realizar utilizando dos métodos el RLL (*Relay Ladder Logic*) y el RLL^{PLUS} (*Stage Programming*) para ello se cuenta con el software DirectSOFT32 Programming para Windows.

Al energizar el PLC (*Power up*), el CPU inicializa el hardware electrónico interno, la inicialización de la memoria comienza con examinar los ajustes de la memoria retentiva (variables, temporizadores, relés de control, contadores y estados). En general, el contenido de la memoria retentiva se preserva, y la memoria no-retentiva se inicializa en cero (salvo que se especifique lo contrario).

Después de las tareas del *power up* (las cuales solo se ejecutan una vez), el CPU comienza una actividad de exploración cíclica (*cyclical scan activity*) La figura 5.1 muestra cómo se diferencian las tareas, dependiendo del modo de operación del CPU y la existencia de algún error.

Como podemos observar el CPU siempre está leyendo las entradas, incluso durante el modo *program*, esto permite que las herramientas de programación supervisen el estado de las entradas en cualquier momento. Las salidas se actualizan solamente en modo *run*, en modo *program*, estas están siempre apagadas.

En modo *run*, el CPU ejecuta el programa de aplicación. Luego, se ejecutan los lazos PID si es que están configurados y finalmente el CPU escribe los resultados de las salidas de estas dos tareas. La detección de errores tiene dos niveles cuando ocurre un error no fatal este se reporta y el CPU permanece en su modo actual. Sin embargo si ocurre algún error fatal, el CPU es forzado a cambiar al modo *program* y las salidas se apagan.

El CPU DL350 cuenta con tres modos de operación:

- ✓ Modo programación (*Program*).- en modo programación el CPU no ejecuta el programa de aplicación ni tampoco actualiza las salidas. Este modo es utilizado para ingresar cambios al programa de aplicación y también para ajustar los parámetros del CPU.
- ✓ Modo funcionamiento (*Run*).- en modo funcionamiento, el CPU ejecuta el programa de aplicación, realiza los cálculos PID para los lazos configurados y actualiza las entradas y salidas. Durante el modo *run* es posible realizar las siguientes operaciones: monitoreo y cambio de los estados de las entradas y salidas, actualizar los valores de temporizadores y contadores y actualizar la ubicación de las variables de memoria.
- ✓ Modo prueba (*Test*).- el modo prueba permite iniciar el modo *Test – Program* ingresar al modo *Test – Run*, ejecutar un número fijo de *scans* y retornar al modo *Test – Program*. El modo *Test* permite mantener el estado de las salidas mientras se cambia entre los modos *Test – Program* y *Test – Run*.

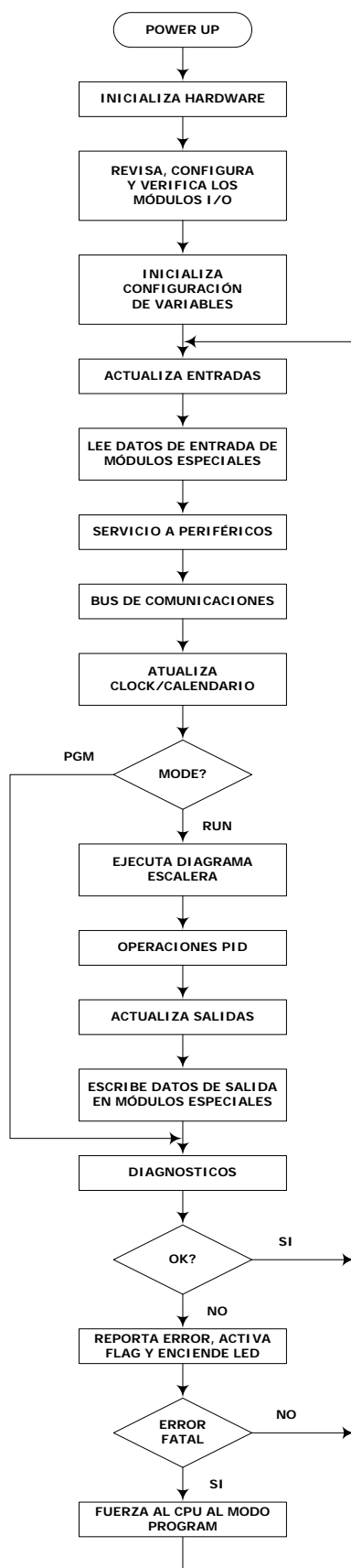


Figura 5.1: Tareas ejecutadas por el CPU del PLC^[12]

^[12] D3-350 CPU User Manual

5.1.1 Programa original

El programa de aplicación original del PLC fue desarrollado en la primera etapa del proyecto por el departamento de electrónica y automática DEA de la UDEP, este programa tiene el control de los siguientes procesos principales:

- ✓ Activación del sistema de baja tensión.
- ✓ Encendido de los ventiladores del túnel.
- ✓ Arranque del sistema de enfriamiento.
- ✓ Habilidad de los transmisores.
- ✓ Activación de la fuente de alta tensión (HV).
- ✓ Permiso de RF.

Adicionalmente el programa cuenta con instrucciones para: leer y almacenar las medidas analógicas (presiones de agua y tensiones de operación de los transmisores) y actualizar el sistema de alarmas.

El programa original se desarrollo utilizando el método RLL^{PLUS}, por lo que el programa está estructurado en estados. A continuación se describen cada uno de los estados que componen le programa, también se presentan los diagramas de estados los cuales permiten ver el programa (*ladder diagram*) como un diagrama de flujo, la figura 5.2 muestra los símbolos y convenciones utilizadas en los diagramas de estados.

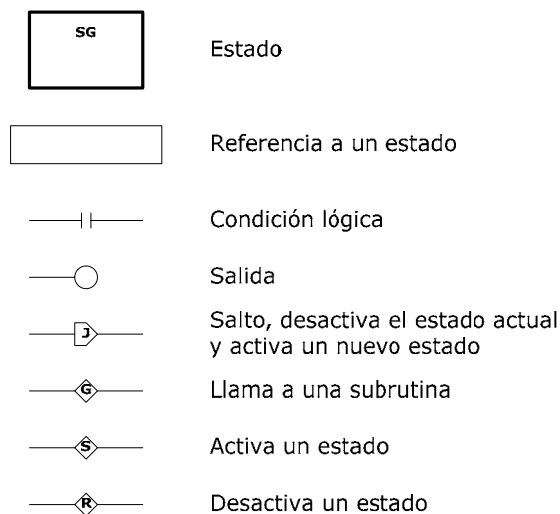


Figura 5.2: Símbolos utilizados en los diagramas de estados^[12]

5.1.1.1 Rutina principal “MAIN”

En la rutina principal se realizan las siguientes tareas: primero se llama a la subrutina K2 la cual se encarga de poner a cero los contadores y temporizadores, además de cargar los valores iniciales de las variables del programa. Luego de ejecutar la subrutina K2 se procede a activar los estados correspondientes a las medidas analógicas y la comunicación con la red Modbus. En la rutina principal también se presta atención al sistema de alarmas y se controla el apagado remoto de las etapas *PA* de los transmisores. La rutina principal se ejecuta siempre.

^[12] D3-350 CPU User Manual

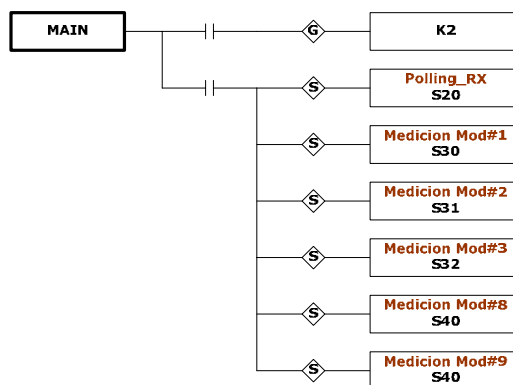


Figura 5.3: Rutina principal

5.1.1.2 Estado Inicial y estado Modo Prueba “S0 y S100”

En el estado inicial se determina las condiciones necesarias para activar los estados de los procesos principales. En este estado también se determina el modo de operación (local ó remoto) y en modo remoto se determina también si el sistema funcionara en modo operación ó en modo prueba.

Cuando el sistema opera en modo local el estado inicial siempre esta activado, mientras que en modo remoto, el estado inicial solo esta activado cuando el sistema funciona en modo operación. En modo prueba se desactiva el estado inicial S0 y se activa el estado Modo Prueba S100.

Las condiciones para activar cada uno de los estados de cada proceso son las siguientes:

- ✓ Estado LV.- en modo local activar el control respectivo del panel local y que no hay falla en los fusibles. En modo remoto cuando se activa el sistema de enfriamiento desde el sistema SCADA y no hay alarmas por fusibles o en el mismo sistema de enfriamiento.
- ✓ Estados TX1, TX2, TX3 y TX4.- activar el control respectivo de cada transmisor (desde el panel local ó desde el sistema SCADA) y que el estado LV se encuentre activo, el estado HV desactivado y finalmente que no existan alarmas por fusibles o en el mismo estado.
- ✓ Estados Blower_R y Blower_L.- activar el control respectivo de cada ventilador (desde el panel local ó desde el sistema SCADA) y que no existan alarmas por fusibles o en el mismo estado (Blower_R ó Blower_L).
- ✓ Estados Cooling.- activar el control respectivo del sistema de enfriamiento (desde el panel local ó desde el sistema SCADA) y que no existan alarmas por fusibles o en el mismo estado del sistema de enfriamiento.
- ✓ Estado HV.- activar el control respectivo de la fuente de alta tensión (desde el panel local ó desde el sistema SCADA), que al menos un transmisor se encuentre encendido, que el estado LV se encuentre activo, que el sistema de enfriamiento se encuentre funcionando y finalmente que no existan alarmas en el mismo estado HV.
- ✓ Estado RF Permit.- activar el control respectivo (desde el panel local ó desde el sistema SCADA), que al menos un transmisor se encuentre encendido y con su tensión de pantalla en el valor adecuado, que el estado LV se encuentre activo, que

el estado HV se encuentre activo, que el sistema de enfriamiento se encuentre funcionando y finalmente que no existan alarmas por fusibles ó en el mismo estado RF Permit.

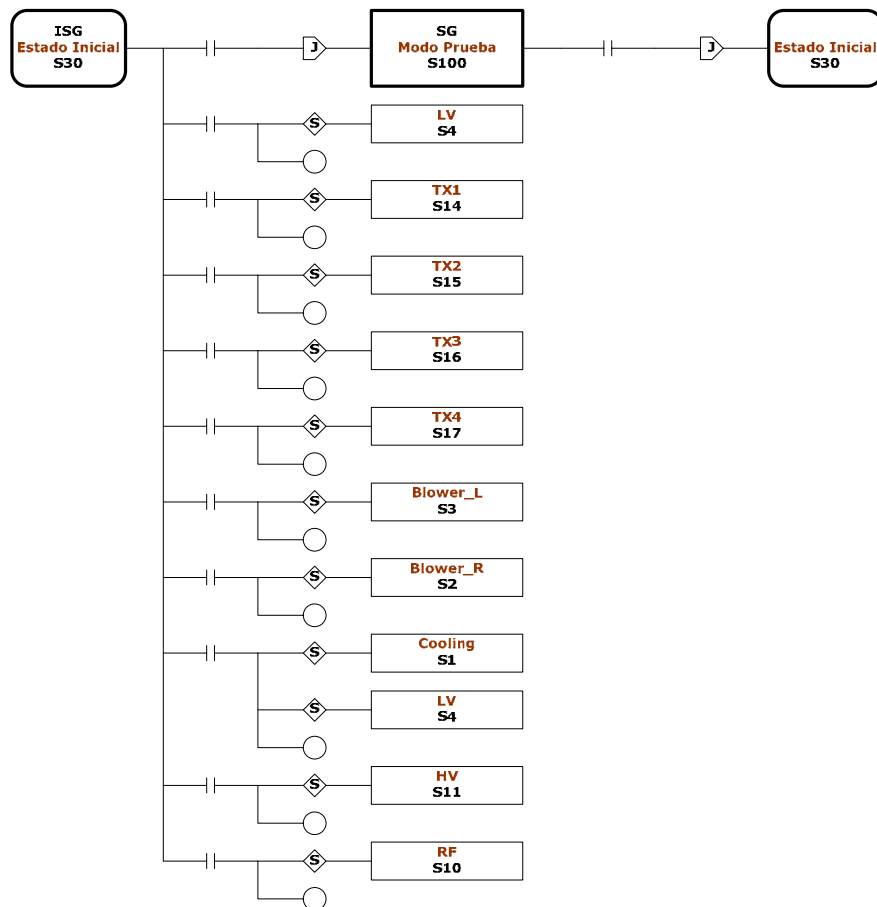


Figura 5.4: Estado inicial y estado prueba

En el estado Modo Prueba es posible hacer funcionar cada uno de los procesos principales de forma independiente. Esto con el propósito de hacer pruebas y detectar posibles fallas del sistema, para ingresar y salir del modo prueba se utiliza el control ubicado en la pantalla “Sistema de transmisores” del sistema SCADA. Al salir del modo prueba se desactiva este estado y se activa el estado inicial.

5.1.1.3 Estado Cooling “S1”

El estado S1 se encarga de poner en funcionamiento el sistema de enfriamiento del sistema de transmisores el cual incluye tres subsistemas. El primero es un circuito de agua de baja presión que cuenta con dos electrobombas. El segundo también es un circuito de agua pero de alta presión este circuito cuenta solo con una electrobomba y finalmente el tercero es un sistema secundario utilizado para enfriar el agua de los dos circuitos anteriores por medio de intercambiadores de calor agua – agua y aire – agua, este sistema cuenta con una bomba.

El arranque de cada sub-sistema se realiza de manera secuencial, primero se arrancan los ventiladores de los intercambiadores agua – aire, luego las bombas del circuito de baja

presión, seguido por la bomba del circuito de alta presión y finalmente la bomba del circuito secundario. En total el arranque del sistema de enfriamiento dura 83 segundos.

En el diagrama de estados observamos que el estado *cooling* se desactiva por cuatro razones, la primera cuando falla cualquiera de las dos electrobombas del sistema de baja presión, la segunda cuando falla la electrobomba del sistema de alta presión, la tercera cuando falla la electrobomba del sistema de enfriamiento secundario y finalmente cuando el sistema de enfriamiento se apaga de forma normal.

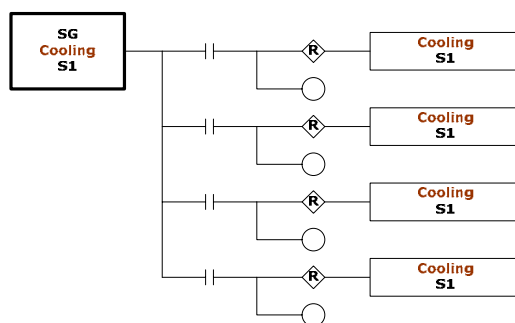


Figura 5.5: Estado Cooling

5.1.1.4 Estados Blower_R y Blower_L “S2 y S3”

Los estados S2 y S3 se encargan de controlar el arranque de los ventiladores del túnel siempre y cuando no exista ninguna alarma, en cada estado se verifica el encendido de los ventiladores de lo contrario se activa un bit de alarma. Los estados S2 y S3 se desactivan en tres ocasiones: cuando ocurre una alarma, cuando fallan los fusibles de la línea de alimentación del sistema de control y finalmente cuando se apagan de forma normal.

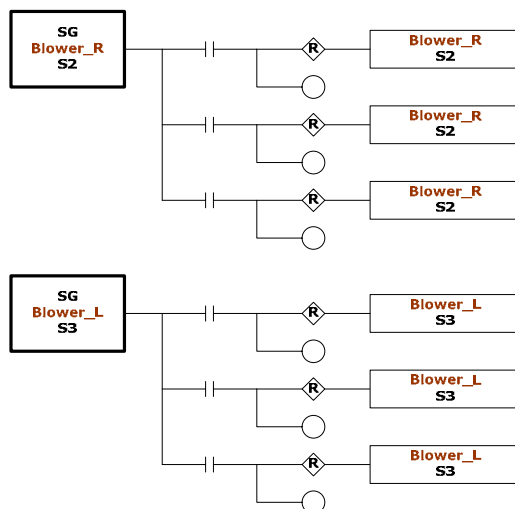


Figura 5.6: Estados Blower_R y BLower_L

5.1.1.5 Estado LV “S4”

Este estado se encarga de activar los relés que permiten alimentar el sistema de transmisores y otros sistemas con baja tensión (120V). En modo remoto este estado se activa automáticamente al encender el sistema de enfriamiento y en modo local se debe

utilizar el control respectivo del panel local. El estado LV se desactiva al apagar el sistema de enfriamiento (en modo remoto) ó al utilizar el control del panel local en modo local.

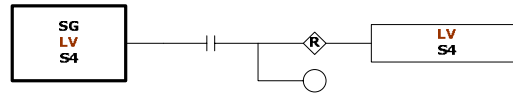


Figura 5.7: Estado LV

5.1.1.6 Estado RF Permit “S10”

En este estado se activa un relé auxiliar que permite el paso de la señal de RF al sistema de transmisores. Este estado se desactiva por tres razones: cuando ocurre alguna falla en cualquiera de los transmisores, por fallas en otras partes del sistema (bombas del sistema de enfriamiento, fuente de alta tensión, fusibles) y cuando se desactiva de forma normal.

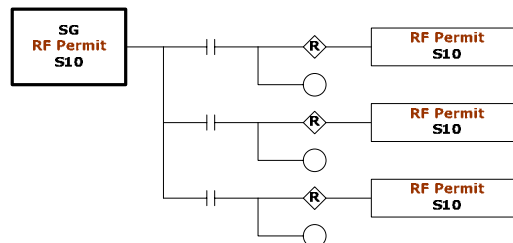


Figura 5.8: Estado RF Permit

5.1.1.7 Estado HV “S11”

El estado HV se encarga de iniciar la secuencia de encendido de la fuente de alta tensión (20kV) que alimenta la placa de los tubos de vacío de la etapa de salida de los transmisores. Para encender la fuente de alta tensión el programa ejecuta las siguientes tareas:

- ✓ Desconecta las resistencias de descarga y conecta el relé auxiliar de apagado que esta normalmente cerrado.
- ✓ Se esperan 5 segundos después de desconectar las resistencias de descarga y se da un pulso de un segundo a la bobina del interruptor de encendido de la fuente.
- ✓ Se espera la señal de encendido del interruptor y luego de 5 segundos se cortocircuitan las resistencias que están en serie con el primario del transformador de la fuente.
- ✓ Luego de 15 segundos de recibir la señal de encendido del interruptor se conectan las resistencias de carga de la fuente.
- ✓ Finalmente luego de cumplida la secuencia de encendido se activa un bit de confirmación.

Dentro de este estado también se verifica las posibles fallas de la fuente así este estado se desactiva por cinco motivos: puertas abiertas del túnel, pulsador remoto de apagado, por sobrecorriente en la fuente, por fallas en otras partes del sistema (transmisores, bombas del sistema de enfriamiento, fusibles) y finalmente se desactiva cuando la fuente se apaga de modo normal.

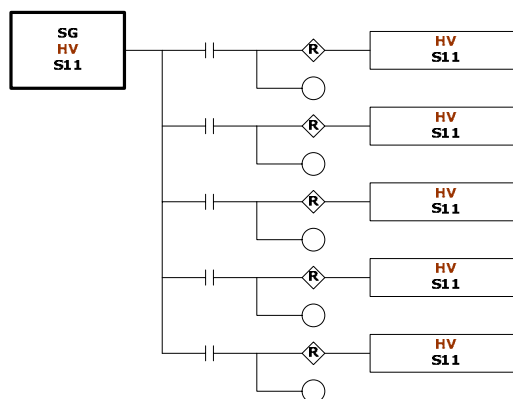


Figura 5.9: Estado HV

5.1.1.8 Estados TX1, TX2, TX3 y TX4 “S14, S15, S16 y S17”

Cada uno de estos estados se encargan de habilitar el funcionamiento de cada uno de los transmisores, primero se comprueba que los flujos de agua tanto del sistema de baja presión como del sistema de alta presión se encuentren por encima de los límites permitidos una vez que se cumple esta condición se activan los relés que permiten el funcionamiento de cada transmisor. El encendido y la operación de cada transmisor se realizan desde su panel local.

El estado de cada transmisor se desactiva en cinco ocasiones: cuando el flujo del sistema de baja presión se encuentra por debajo del límite permitido, cuando el flujo del sistema de alta presión se encuentra por debajo del límite permitido, cuando falla el transmisor, por falla en los fusible y finalmente cuando el transmisor se desactiva de forma normal.

En la figura 5.9 se muestra el diagrama de estados correspondiente a la habilitación del transmisor N° 1, los diagramas de estados de los transmisores restantes son análogos.

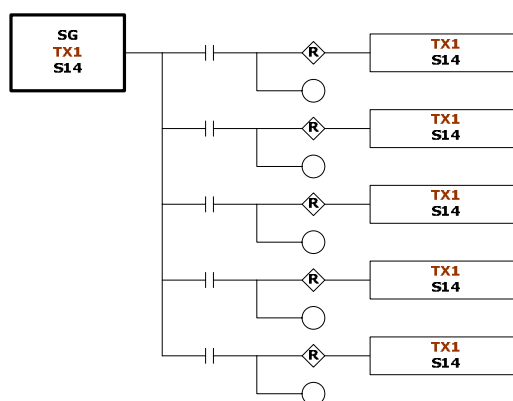


Figura 5.10: Estado TX1

5.1.1.9 Estado Polling RX y Falla Poll RX “S20 y S21”

El estado Polling RX se encarga de iniciar la comunicación con los dispositivos esclavos conectados a la red de comunicaciones Modbus, se utiliza la técnica denominada *Polling* en la cual cada dispositivo esclavo se interroga de manera secuencial cada cierto tiempo de

esta manera se cuenta con información actualizada de los dispositivos esclavos todo el tiempo.

El *Polling* se realiza cada 100ms después de los cuales se llama a la subrutina K3 encargada de leer las variables de los dispositivos esclavos si ocurre algún error en la comunicación se desactiva el estado S20 y se activa el estado S21 el cual espera un tiempo y regresa al estado S20.

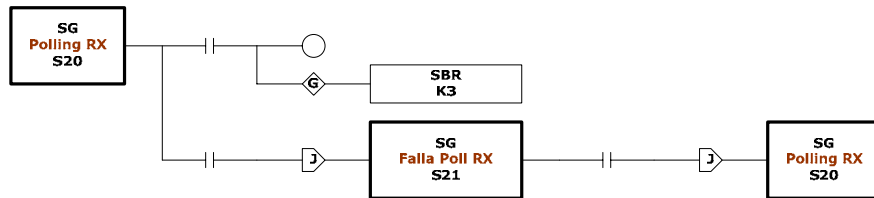


Figura 5.11: Estado Polling RX y Falla Poll RX

5.1.2 Modificaciones

Para incluir los cambios del nuevo sistema de control se han realizado modificaciones al programa original del PLC. Estas modificaciones incluyen cambios en algunos estados y la adición de nuevos estados.

Como se menciona en el capítulo cuatro el nuevo sistema de control se ha instalado en el transmisor número tres del ROJ, por lo que de los cuatro estados que controlan el funcionamiento de los transmisores solo se ha modificado el correspondiente al transmisor número tres.

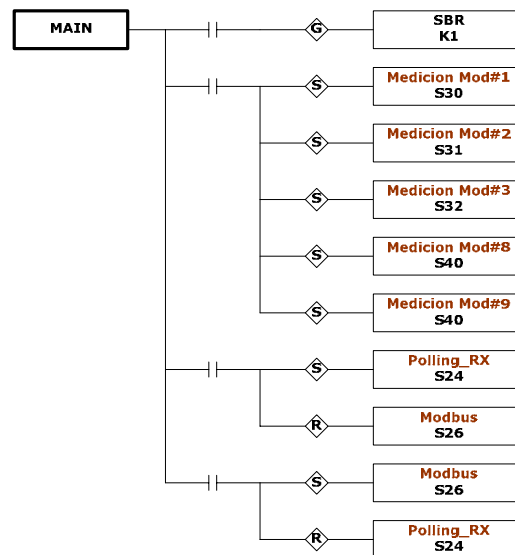


Figura 5.12: Rutina principal modificada

5.1.2.1 Rutina principal “MAIN”

En la rutina principal se ha cambiado el nombre de la subrutina K2 por K1, adicionalmente la activación del estado Polling (antes Polling RX) se realiza utilizando un control de la pantalla “Slave’s Modbus Configuration” del sistema SCADA la razón de esto es que se ha

agregado un nuevo estado llamado Modbus el cual también se utiliza para comunicarse con los dispositivos conectados a la red Modbus. Dado que ambos estados utilizan el mismo *bus* no pueden estar activados al mismo tiempo. Por defecto se encuentra activado el estado Polling.

Adicionalmente al control del apagado remoto de las etapas *PA* de los transmisores se han agregado instrucciones para también controlar el apagado remoto de las etapas *Driver* de los transmisores

5.1.2.2 Estado Inicial y estado Modo Prueba “S0 y S100”

En el estado inicial se ha agregado la activación del estado DRV3 el cual se activa junto con el estado PA3 (antes TX3) las condiciones de activación de cada uno de los estados no se han modificado.

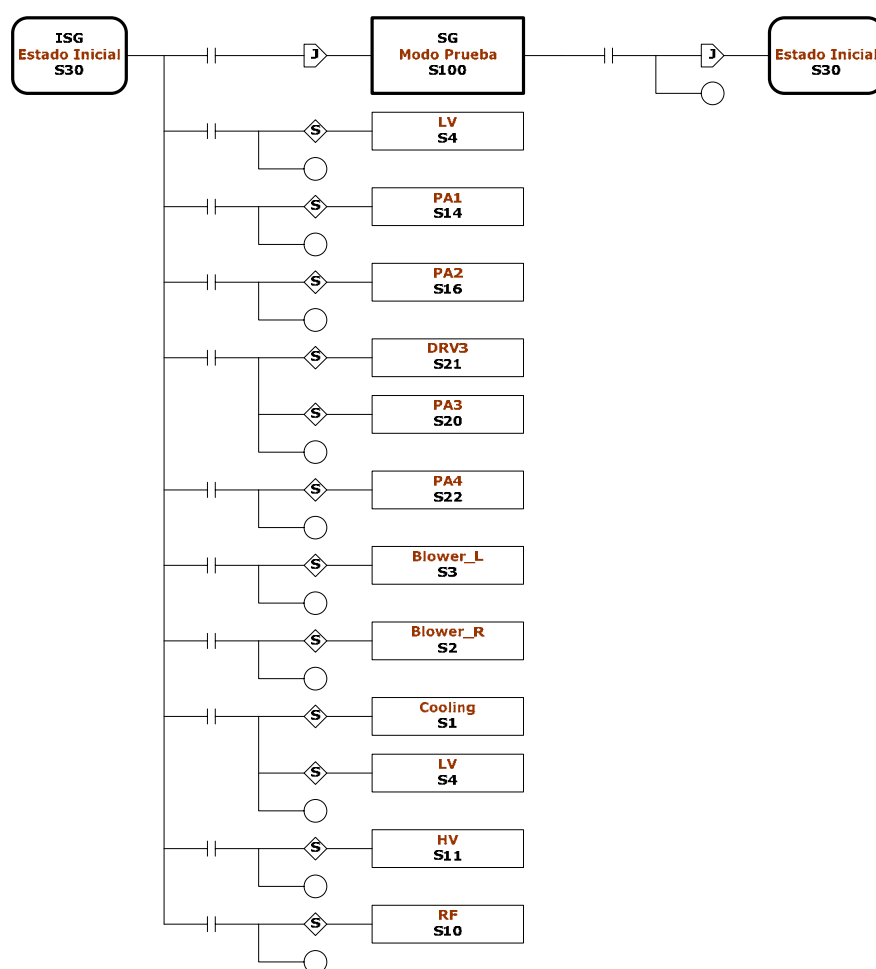


Figura 5.13: Estado Inicial modificado

5.1.2.3 Estado PA3 “S14”

Los estados TX1, TX2, TX3 y TX4 han sido renombrados con las etiquetas PA1, PA2, PA3 y PA4 respectivamente. En el caso del estado PA3 se han agregado las instrucciones necesarias para controlar los siguientes procesos:

- ✓ Encendido remoto tanto del sistema como de la tensión de pantalla.
- ✓ Apagado remoto de la tensión de pantalla
- ✓ Ajuste remoto de la tensión de filamento.

El nuevo diagrama del estado PA3 no cambia respecto al original TX3, esto debido a que en los diagramas de estados solo se muestran las relaciones entre estados.

5.1.2.4 Estado DRV3 “S20”

Este nuevo estado se encarga de controlar los procesos involucrados con el encendido remoto de la etapa *Driver* del transmisor número tres, estos procesos son el encendido y apagado del sistema y de la tensión de pantalla. Este estado se desactiva solo cuando se apaga normalmente.

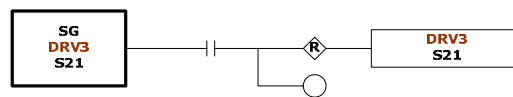


Figura 5.14: Estado DRV3

5.1.2.5 Estado Polling “S24”

El estado Polling ha sido modificado para incluir la comunicación con las estaciones remotas. Para la lectura se ha modificado la subrutina K2 agregándole nuevas instrucciones ya que el número de variables leídas de las estaciones remotas es mayor a las leídas de los módulos Acromag además las direcciones de las variables también son diferentes. Para la escritura se ha creado una nueva subrutina K3 la cual se encarga de escribir los nuevos valores de las variables (de la estación remota) que contienen la posición de los motores paso a paso, cuando se cambian desde el sistema SCADA.

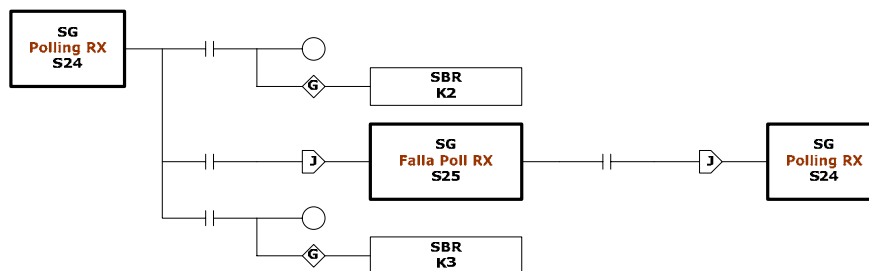


Figura 5.15: Estado Polling

5.1.2.6 Estado Modbus “S26”

Este nuevo estado ha sido creado para configurar los dispositivos conectados a la red Modbus desde el sistema SCADA. Especificando la dirección del esclavo se puede leer/escribir las variables con referencia Modbus 4XXXX.

5.2 Software del sistema SCADA

Los sistemas SCADA forman parte importante de la automatización ya que conforman la interfase entre los operarios y el sistema de control esto normalmente se denomina HMI (*Human Machine Interface*). La herramienta de desarrollo utilizada es LookoutDirect.

LookoutDirect opera bajo el entorno Windows y es capaz de comunicarse con diferentes dispositivos de campo como módulos I/O, controladores lógicos programables (PLC's), unidades de terminales remotos (RTU's) etc. LookoutDirect es un paquete configurable orientado a objetos y activado por eventos, que no requiere ninguna programación, sino mas bien se utiliza en su lugar, representaciones gráficas de dispositivos del mundo real como interruptores, potenciómetros, registradores, botones, perillas, medidores, los cuales ligán sus imágenes a instrumentos reales de campo usando PLCs, RTUs, tableros de adquisición de datos, u otros dispositivos de I/O.

La arquitectura de LookoutDirect consiste en objetos y sus miembros de datos (*Data Members*), conexiones y servicios. El desarrollo de una aplicación en LookoutDirect consiste en la creación, configuración y conexión de objetos. Los objetos son representaciones del software de todos los dispositivos conectados físicamente a la PC que tiene instalado el LookoutDirect mientras que los servicios de LookoutDirect son los que manejan las conexiones entre la PC y los diferentes dispositivos conectados a la misma^[13].

5.2.1 Programa original

El sistema SCADA original fue desarrollado junto con el programa original del PLC en la primera etapa del proyecto por el DEA de la UDEP el sistema desarrollado en dicha oportunidad se encarga de la supervisión y control de los siguientes procesos:

- ✓ Arranque del sistema de enfriamiento, que incluye dos circuitos de agua uno de alta y otro de baja presión, cuatro intercambiadores de calor de agua-aire y 2 ventiladores auxiliares.
- ✓ Selección de los transmisores a utilizar.
- ✓ Activación de las fuentes de alto voltaje (20 KV) para las etapas PA del transmisor.
- ✓ Activación de la señal de radio frecuencia RF.
- ✓ Monitoreo de parámetros y alarmas del sistema
- ✓ Parada de emergencia.

Cada uno de los procesos se encuentran organizados en pantallas las cuales cuentan con los respectivos objetos para su supervisión y control, a continuación se describen las pantallas mas relevantes del sistema SCADA.

5.2.1.1 Pantalla principal “Sala de Transmisores”

Esta es la pantalla principal del sistema desde aquí en primer lugar se puede seleccionar el modo de funcionamiento esto es entre el modo operación y el modo prueba dependiendo de este modo de operación el aspecto de la pantalla “Sala de transmisores” cambia ya que habilita y rehabilita el acceso a los diferentes procesos del sistema. Desde esta pantalla se pueden realizar las siguientes acciones, dependiendo del modo de funcionamiento:

Modo Operación:

- ✓ Acceder a las pantallas de habilitación de los cuatro transmisores.
- ✓ Seleccionar el modo Prueba.

^[13] Lookout Direct – Getting starting guide

- ✓ Acceder a la pantalla del sistema de enfriamiento.
- ✓ Acceder a la pantalla de activación de la fuente de alta tensión.
- ✓ Acceder a la pantalla del permiso de RF.
- ✓ Acceder a la pantalla de los ventiladores del túnel.
- ✓ Acceder a la pantalla del sistema de bombas auxiliares.
- ✓ Acceder a la pantalla de estado y alarmas del sistema.
- ✓ Acceder a la pantalla para iniciar la grabación de datos.

Modo Prueba:

- ✓ Acceder a las pantallas de habilitación de los cuatro transmisores.
- ✓ Seleccionar el modo Operación.
- ✓ Acceder a la pantalla de los ventiladores del túnel.
- ✓ Acceder a la pantalla del sistema de bombas auxiliares.
- ✓ Acceder a la pantalla de estado y alarmas del sistema.
- ✓ Acceder a la pantalla de activación de la fuente de alta tensión.
- ✓ Acceder a la pantalla del sistema de baja presión.
- ✓ Acceder a la pantalla del sistema de alta presión.
- ✓ Acceder a la pantalla del sistema de enfriamiento secundario.

Adicionalmente en esta pantalla se presenta el estado de los fusibles de las líneas que alimentan el sistema de control. También se cuenta con un pulsador de reset para las alarmas.

Cuando el sistema se encuentra en modo operación los pulsadores de acceso a las pantallas del sistema de enfriamiento, activación de la fuente de alta tensión y permiso de RF se encuentran desactivados ya que primero se deben cumplir las condiciones necesarias antes de activarlos.



Figura 5.16: Pantalla principal “Sala de transmisores”

5.2.1.2 Pantallas de habilitación de transmisores

En las pantallas de habilitación de transmisores se cuenta con un interruptor para habilitar o deshabilitar el transmisor, en esta pantalla también se cuenta con medidores para la supervisión de los flujos de agua (alta y baja presión) y las tensiones de operación del transmisor. Desde estas pantallas se puede acceder a dos pantallas adicionales la primera

“Configuración de Medidores” utilizada para establecer los valores límites mínimos y máximos de los diferentes parámetros del transmisor, es necesario aclarar que estos valores no son enviados al PLC y solo se utilizan para su visualización en el sistema SCADA. La segunda pantalla dependiendo del transmisor muestra los medidores correspondientes a las tensiones de operación de la etapa *Driver*.

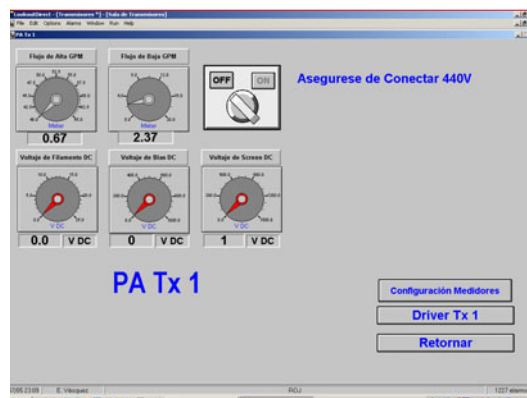


Figura 5.17: Pantalla de habilitación del transmisor N° 1 “PA Tx1”

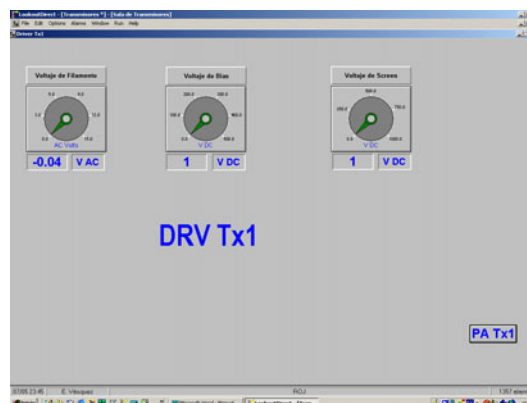


Figura 5.18: Pantalla “Driver Tx1”

5.2.1.3 Pantalla “Sistema de Enfriamiento”

El acceso a esta pantalla está disponible una vez que se ha habilitado al menos un transmisor, en esta pantalla se cuenta con un interruptor para iniciar el arranque del sistema de enfriamiento una vez iniciado el arranque se habilitan tres pulsadores que permiten acceder a los subsistemas del sistema de enfriamiento (sistema de baja presión, sistema de alta presión y enfriamiento secundario).

Cuando el sistema se encuentra en modo prueba el acceso a las pantallas de los subsistemas del sistema de enfriamiento se puede hacer directamente de la pantalla principal “Sala de Transmisores”.

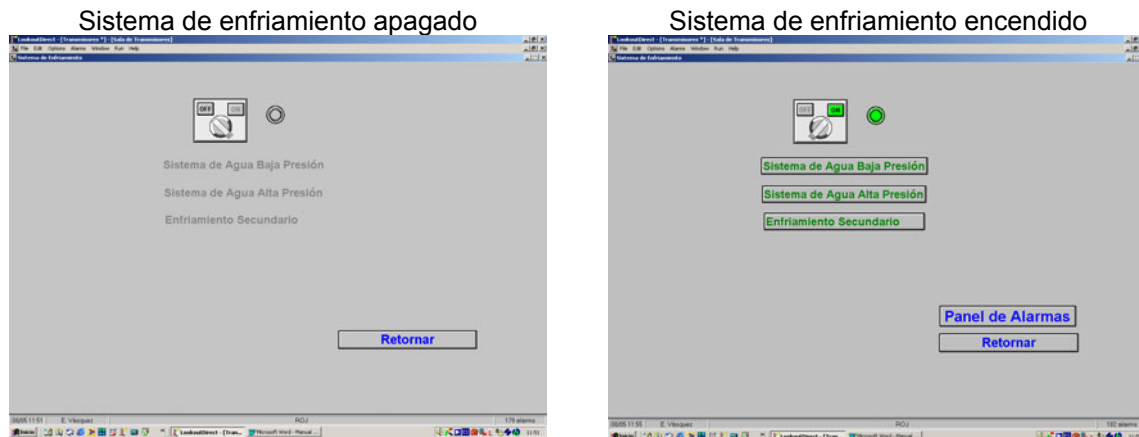


Figura 5.19: Pantalla “Sistema de Enfriamiento”

5.2.1.4 Pantalla “Fuente de Alta Tensión”

En modo operación el acceso a esta pantalla es posible después de que el sistema de enfriamiento se encuentre funcionando. La pantalla de la fuente de alta tensión cuenta con un interruptor para activar la fuente y un medidor para visualizar el valor, en voltios, de la fuente. Adicionalmente se puede verificar el estado de los transmisores (apagado, alimentado ó activado) y se cuenta con un indicador del estado de las puertas del túnel.



Figura 5.20: Pantalla “Fuente de Alta Tensión” bloqueada



Figura 5.21: Pantalla “Fuente de Alta Tensión” lista

Cuando las condiciones no están dadas la activación de la fuente de alta tensión no es posible y la pantalla se muestra como en la figura 5.20, cuando las condiciones estén dadas se puede utilizar el interruptor para activar la fuente.

En modo prueba el acceso a esta pantalla está permitido sin necesidad de que se cumplan las condiciones necesarias, por lo que se tiene que tener mucho cuidado si se quiere operar la fuente de alta tensión en modo prueba.

5.2.1.5 Pantalla “RF Permit”

El acceso a esta pantalla solo es posible en modo operación, esta pantalla se utiliza para permitir el paso de la señal de RF al sistema de transmisores. Al igual que la pantalla de la fuente de alta tensión es necesario que se cumplan las condiciones respectivas para poder accionar el interruptor y tener acceso a la pantalla.

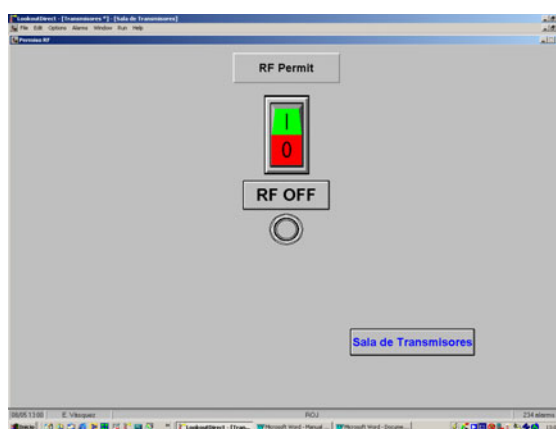


Figura 5.22: Pantalla “RF Permit”

5.2.1.6 Pantalla “Ventiladores del Túnel”

Esta pantalla es utilizada para encender los ventiladores del túnel por medio de dos interruptores uno para el ventilador izquierdo y otro para el ventilador derecho. El acceso a esta pantalla y el encendido de los ventiladores no tienen restricción tanto en modo operación como en modo prueba. El único requisito es que la media tensión este conectada (440 V).

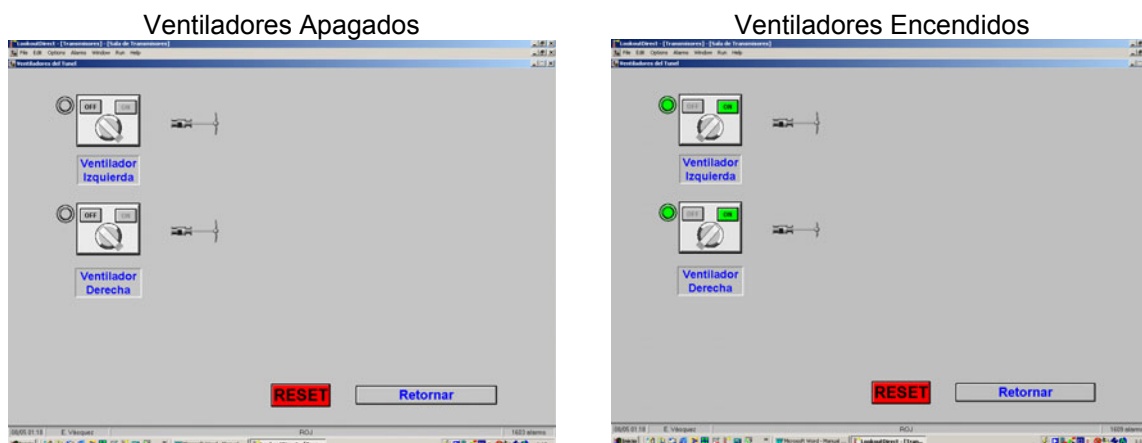


Figura 5.23: Pantalla “Ventiladores del Túnel”

5.2.1.7 Pantalla “Bombas Auxiliares”

La pantalla bombas auxiliares es accesible tanto en modo prueba como en modo operación y se utiliza para poner en marcha las bombas que alimentan los tanques de reserva de los subsistemas del sistema de enfriamiento. Así la pantalla cuenta con tres interruptores, el primero para encender la bomba del tanque de reserva del sistema de enfriamiento secundario, el segundo para encender la bomba del tanque de reserva del sistema de alta presión y el tercero para encender la bomba del tanque de reserva del sistema de baja presión.



Figura 5.24: Pantalla “Bombas Auxiliares”

5.2.1.8 Pantalla “Estados y Alarmas del sistema”

Esta pantalla se utiliza para monitorear el estado de cada uno de los equipos del sistema de transmisores para ello se utilizan indicadores luminosos, el color plomo indica que el equipo ó sistema se encuentra apagado, el color verde indica que el equipo ó sistema se encuentra en funcionamiento normal y finalmente el color rojo que indica la ocurrencia de alguna falla. Existe un color adicional el naranja que se utiliza para indicar que el equipo o sistema se encuentra listo para entrar en funcionamiento.



Figura 5.25: Pantalla “Sistema de Alarmas”

5.2.2 Modificaciones

Al igual que el programa del PLC fue necesario realizar modificaciones al sistema SCADA para incluir los cambios realizados en el sistema de control. En primer lugar se realizaron modificaciones que involucran a todas las pantallas del sistema SCADA estas modificaciones tienen que ver con la presentación de las pantallas así se realizaron los siguientes cambios:

- ✓ Se ha cambiado el tipo y tamaño de letra.
- ✓ Se han uniformizado el tamaño de todos los pulsadores y en algunos casos estos han sido reubicados dentro de la pantalla.
- ✓ El idioma de todas las etiquetas se ha cambiado al inglés, la razón de esto fue para uniformizar todo el sistema.
- ✓ Se ha mejorado el aspecto de los interruptores y de las imágenes utilizadas, esto para que el sistema SCADA tenga la apariencia del sistema real.

Los cambios concernientes al nuevo sistema de control involucran el cambio de algunas pantallas y la adición de nuevas pantallas.

Pantallas modificadas:

- ✓ Pantalla principal “Transmitters Room”.
- ✓ Pantalla de habilitación de transmisores “Transmitter #”
- ✓ Pantalla “Driver #”.

Pantallas nuevas:

- ✓ Pantalla de control de la etapa de salida “PA #”.
- ✓ Pantalla de supervisión de parámetros “System Parameters”.
- ✓ Pantalla “Modbus Slave’s Configuration”

En los siguientes apartados se describen estas pantallas.

5.2.2.1 Pantalla principal “Transmitters Room”

En la pantalla principal se ha agregado un pulsador para acceder a la pantalla “System parameters” se ha reorganizado la ubicación de los diferentes pulsadores y se ha cambiado el interruptor con el que se cambia el modo de funcionamiento (operación/prueba). Existe un pulsador llamado “RF Power” con el cual se accede a la pantalla que será utilizada para monitorear la potencia transmitida y/o reflejada de los transmisores.

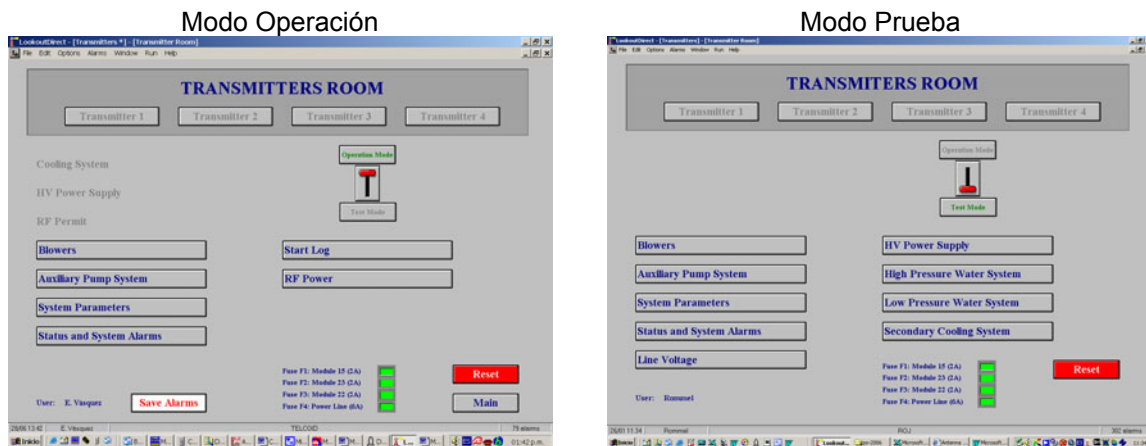


Figura 5.26: Pantalla “Transmitters Room” modificada

5.2.2.2 Pantalla de habilitación de transmisores “Transmitter #”

En estas pantallas (cuatro en total) se han agregado dos pulsadores para acceder a las pantallas que controlan y supervisan la operación de la etapa de salida *PA* y de su etapa previa *Driver*. Los medidores de flujo se mantienen y los medidores de las tensiones han sido reubicados en la respectiva pantalla (*PA TX#*). En esta pantalla se han agregado también controles para monitorear la comunicación y el estado de la estación remota respectiva.

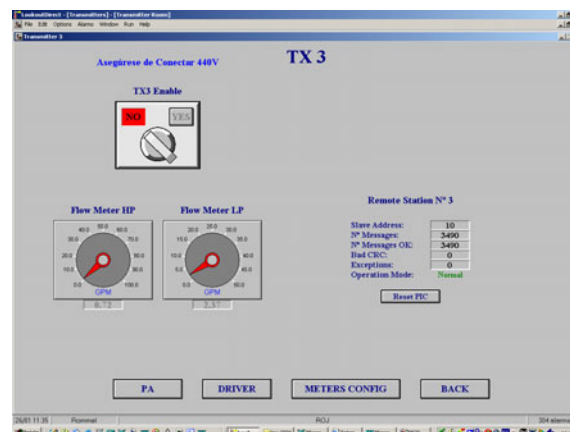


Figura 5.27: Pantalla “Transmitter 3” modificada

5.2.2.3 Pantallas “Driver TX#”

En el sistema original estas pantallas estaban destinadas únicamente al monitoreo de las tensiones de operación de estas etapas. Ahora con el nuevo sistema se han agregado los pulsadores/indicadores respectivos para el encendido y apagado remoto del sistema y de la tensión de pantalla además se han agregado medidores para la supervisión de las corrientes de operación y finalmente se han agregado los controles para ajustar la tensión de pantalla.

En esta pantalla también se muestra el modo de operación de esta etapa *Local/Remote*, en modo *Local* se desactivan los pulsadores de encendido y los controles que ajustan la tensión de pantalla.

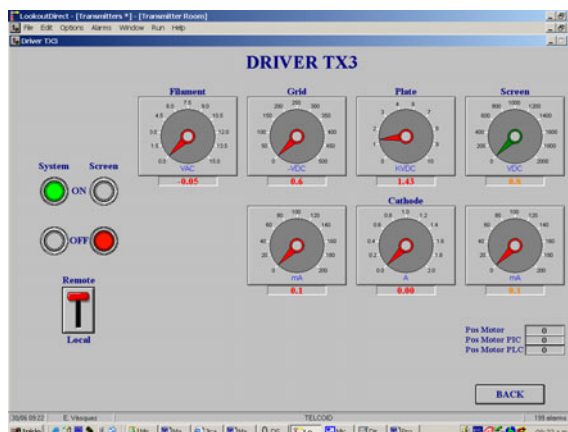


Figura 5.28: Pantalla “Driver TX3” modificada

5.2.2.4 Pantallas “PA #”

Estas pantallas se utilizan para controlar y supervisar de forma remota las etapas de salida de cada transmisor, cuentan con pulsadores/indicadores para el encendido y apagado remoto del sistema y de la tensión de pantalla así como con controles para ajustar tanto la tensión de filamento como la tensión de pantalla.

Al igual que las pantallas “Driver TX#” estas pantallas cuentan con medidores para la supervisión de las corrientes de operación así como con un indicador que muestra el modo de operación de estas etapas *Local/Remote*. En modo *Local* los pulsadores de encendido y apagado junto con los controles que ajustan la tensión de filamento y la tensión de pantalla se desactivan.

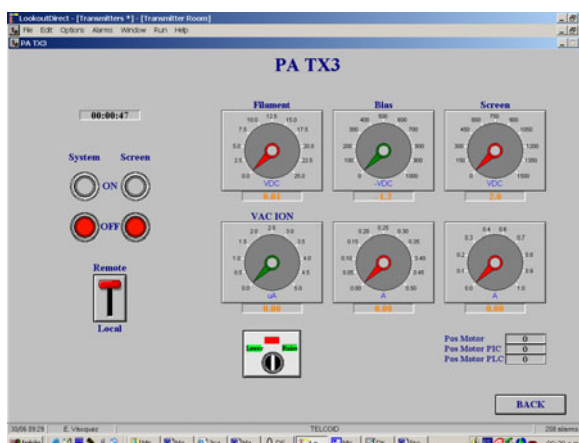


Figura 5.29: Pantalla nueva “PA TX3”

5.2.2.5 Pantalla “System parameters”

Esta pantalla accesible desde la pantalla “Transmitters Room” se utiliza para monitorear todos los parámetros de operación de los cuatro transmisores de manera simultanea. Para la visualización de los mismos se utilizan valores digitales que cambian de color dependiendo del rango en que se encuentran así: el color plomo indica que el transmisor se encuentra apagado, el color verde indica que el valor se encuentra dentro de los límites permitidos y el color rojo indica que se encuentran fuera de los límites permitidos. Un cuarto color, el naranja, es utilizado para indicar que el transmisor se encuentra listo para ser encendido.



Figura 5.30: Pantalla nueva “System Parameters”

CAPÍTULO 6

RESULTADOS

En el presente capítulo se presentan los resultados obtenidos en el desarrollo del proyecto, como se mencionó en el capítulo 4, el prototipo del sistema ha sido implementado en el transmisor N° 3 por lo que todos los resultados están referidos a dicho prototipo.

6.1 Control del encendido de los transmisores

El sistema ha sido probado tanto en modo local como en modo remoto. Con el sistema operando en modo local se pudo comprobar que no existiera ningún conflicto entre la instalación antigua y la nueva instalación, tampoco hubo problemas utilizando los programas del PLC y del sistema SCADA originales.

En modo remoto se realizaron pruebas de encendido y apagado, también se comprobó el buen funcionamiento del control de la tensión de filamento de la etapa *PA*; las pruebas fueron satisfactorias.

6.2 Control de la tensión de pantalla

El prototipo de la estación remota se montó en el panel de control del transmisor y se probó su operación. En cuanto al control de los motores paso a paso fue necesario ajustar el voltaje de alimentación de los mismos, hasta lograr el torque apropiado para mover el transformador variable de la etapa *PA* (el transformador de la etapa *Driver* no presentó problemas ya que cuenta con una inercia mucho menor).

De esta manera se alimentaron los motores con 6.5 VDC, con un consumo de corriente de 0.7 A por fase. En el peor de los casos, esto es cuando se alimentan dos fases de los dos motores, la corriente total consumida por los motores será de 2.8 A. Si bien es cierto que, la fuente fue diseñada para entregar 4 A, fue necesario colocar un disipador en el transistor de la fuente ya que la potencia disipada por el mismo puede alcanzar 27 W en el peor de los casos. Es necesario aclarar que el transistor solo trabaja cuando los motores están en movimiento.

La comunicación de la estación remota utilizando una red de comunicaciones RS485, protocolo Modbus, fue probada, utilizando una PC como estación maestra, para ello se utilizó el software Modbus Tester V3.00 en su versión *freeware*. Con dicho software fue posible determinar la confiabilidad de la comunicación, así la tabla 5.1 muestra algunos de los resultados obtenidos para diferentes periodos de muestreo.

Tabla 6.1: Pruebas de comunicación de la estación remota

Scan Rate [ms]	Polls	Valid Responses	Error [%]
50	282633	282632	0.000
50	280514	280512	0.001
100	220496	220496	0.000
100	223654	223651	0.001
200	118425	118425	0.000
200	118647	118645	0.002
500	60477	60477	0.000
500	60852	60852	0.000
1000	89284	89283	0.001
1000	34621	34620	0.003

Al finalizar las pruebas se pudo determinar que la tasa de errores se encuentra dentro de límites razonables, considerando el entorno con alta EMI (*Electro – Magnetic Interference*), los temas relacionados con la interferencia de la EMI en el sistema podrán ser estudiados en otras tesis.

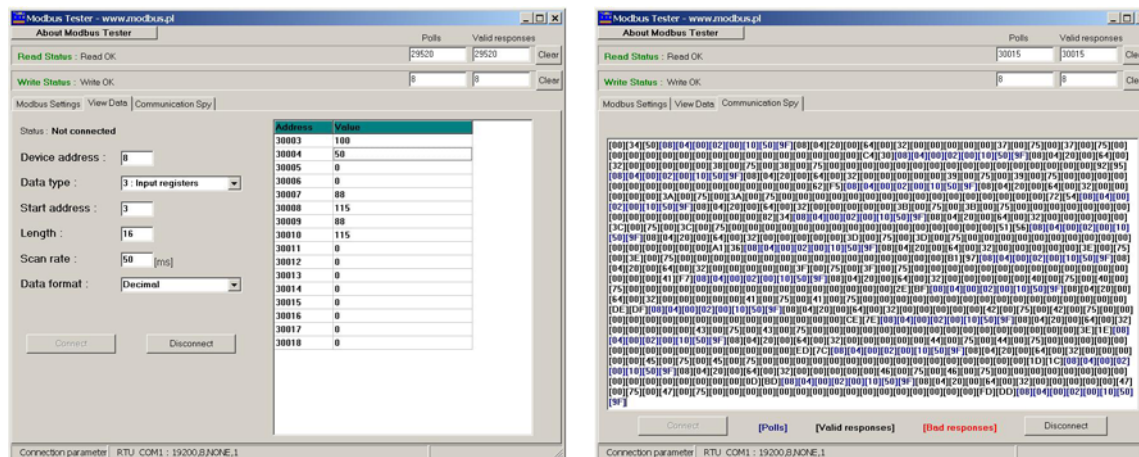


Figura 6.1: Software Modbus Tester.

La comunicación de la estación remota con el PLC también fue probada. El comportamiento fue el mismo que con la PC, habiéndose usado en este caso como referencia los contadores de mensajes implementados en el microcontrolador.

6.3 Análisis económico

El análisis económico efectuado, corresponde al desarrollo de la estación remota, el cual se compara luego con sistemas de control para motores paso a paso (*stepper systems*) comerciales.

Tabla 6.2: Materiales utilizados en la estación remota

Item	Cant.	Unidad	Descripcion	P. Unitario US\$	P. Total US\$
1	1	Und.	Microcontrolador PIC16F873	6.80	6.80
2	1	Und.	Transceptor diferencial SN75176	0.80	0.80
3	3	Und.	Optoacopladores 4N35	0.56	1.68
4	2	Und.	Darlington switch ULN2064B	0.57	1.14
5	1	Und.	Cristal de 4MHz	0.38	0.38
6	1	Und.	Transformador 120/12 VAC, 4A	3.22	3.22
7	1	Und.	Transformador 120/9 VAC, 0.5A	1.71	1.71
8	2	Und.	Regulador de voltaje L7805	0.85	1.70
9	1	Und.	Regulador de voltaje LM317	0.85	0.85
10	1	Und.	Transistor 2N2955	0.62	0.62
11	1	Und.	Puente rectificador de 4A	1.14	1.14
12	1	Und.	Puente rectificador de 1A	0.48	0.48
13	1	Und.	Diodo 1N4002	0.06	0.06
14	3	Und.	Diodo LED	0.06	0.18
15	6	Und.	Condensadores electroliticos	0.11	0.66
16	3	Und.	Condensadores ceramicos	0.03	0.09
17	1	Und.	Potenciometro de 2.2K	0.36	0.36
18	14	Und.	Resistencias varias	0.02	0.28
19	16	Und.	Conectores Varios	0.14	2.24
20	6	pulg2	Fabricacion de circuitos impresos	0.72	4.32

Total US\$	28.71
-------------------	--------------

En el mercado existen diferentes tipos de sistemas de control para motores paso a paso, incluso el fabricante del PLC utilizado en el ROJ ofrece sistemas de este tipo. Así, el PLC necesitaría los siguientes componentes para trabajar con un sistema de motores paso a paso: un modulo contador ó un modulo de salidas digitales para generar los pulsos, cable de control para llevar las señales, un driver y una fuente de alimentación. La generación de pulsos tendría que programarse en el PLC.

Tabla 6.3: Materiales necesarios para controlar los motores paso a paso con el PLC

Item	Cant.	Unidad	Descripcion	P. Unitario US\$	P. Total US\$
1	1	Und.	Stepper Drive 3.5A 32VDC Bipolar 2 Phase	149.00	149.00
2	1	Und.	Power Supply Stepper 32VDC 4A	99.00	99.00
2	1	Und.	Module D3-08TD2 8PT 5-24VDC Output	73.00	73.00
4	40	m	Cable de control de 4 hilos	0.72	28.80

Total US\$	349.80
-------------------	---------------

A simple vista observamos la gran diferencia en el costo. Otra opción pudo haber sido utilizar circuitos integrados como el L297 ó el NJM3517 (controladores de motores paso a

paso) pero igual es necesario utilizar un microcontrolador o un modulo del PLC para generar los pulsos.

CONCLUSIONES

El sistema desarrollado constituye un prototipo y, las pruebas realizadas con él, indican que los objetivos del presente trabajo de tesis fueron cumplidos en forma satisfactoria. Como resultado, se cuenta con la facilidad de encender operar y supervisar de forma remota el Transmisor N° 3 del radar principal del ROJ. Para extender el nuevo sistema de control al resto de transmisores solo se necesita copiar lo realizado en el transmisor N° 3.

Las comunicaciones entre el sistema SCADA, el PLC y el resto de periféricos (módulos Acromag y estaciones remotas) son de vital importancia ya que permiten al usuario tener un control y acceso remoto de las diferentes variables del sistema de transmisores.

El protocolo Modbus, en una red de control distribuido puede ser usado para comunicarse con una serie de dispositivos, distribuidos alrededor de la planta. Esto permite a una sola persona supervisar remotamente diversos procesos simultáneamente desde una única posición. Además de permitir el monitoreo de variables (como las corrientes de operación del transmisor) y el ajuste de parámetros operativos (como las tensiones de pantalla de las etapas *PA* y *Driver*) cuando el sistema lo requiere.

Con la implementación del protocolo Modbus en el microcontrolador PIC, se evidenció la facilidad y flexibilidad de este protocolo y por ende la razón de su alta difusión en entornos industriales.

La utilización de componentes de bajo costo, en este caso el microcontrolador PIC permitió ver el alcance de estos componentes en la solución de problemas específicos de ingeniería que integren hardware y software. Es decir, partiendo de una tecnología comercial se ha obtenido, en nuestro caso, un valor agregado con ventajas de costo o facilidad de uso sobre sistemas comerciales equivalentes. Se pudo constatar la versatilidad y flexibilidad de estos componentes en tareas como el control de motores paso a paso y comunicación serial, sin explotar aún toda su capacidad.

En el caso de la estación remota solo se implementaron los códigos de función Modbus necesarios en la aplicación de este proyecto (dos códigos de función públicos), sin embargo, la estación remota funciona de manera adecuada ante mensajes con códigos de función no implementados, es decir, responde con una respuesta de excepción a la estación maestra indicando que no soporta un código de función no definido.

Por otra parte, la estación remota puede emplearse en aplicaciones distintas en las que se requiera: transmisión serial, almacenamiento y lectura de datos en la memoria interna (EEPROM y RAM), transacciones Modbus diferentes a las desarrolladas, etc. Las pruebas realizadas indican un alto grado de confiabilidad del sistema

En contraste con los servo – motores, los motores paso a paso pueden ser usados a lazo abierto, sin la necesidad de costosos *Encoders* para comprobar su posición. Los motores de pasos son mucho más rentables que los servo sistemas debido a su control simplificado y circuito de alimentación (*Drivers*). En un motor a pasos no existen escobillas por lo que se elimina la necesidad del mantenimiento. Aunque un sistema con motor paso a paso no puede alcanzar la velocidad de un sistema con servo motor, su facilidad de empleo permite que sean la solución preferida para muchos de los sistemas de control de movimiento automatizado, motivo por el cual se ha usado en el presente trabajo.

Se puede continuar con el desarrollo exportando los datos del sistema a un servidor de Internet, con lo cual será posible visualizar la operación del sistema desde cualquier lugar del mundo.

REFERENCIAS

- [1] Universidad de Piura – Departamento de Electrónica y Automática, “Automatización del sistema de transmisores del ROJ – Memoria Descriptiva”, 2003.
- [2] Acromag Inc, “BusWorks® Model 913MB/914MB Network I/O Module User’s Manual”, 2000.
- [3] Modbus.org, “Specification & Implementation for MODBUS over Serial Line” V1.0, 2003, <http://www.modbus.org>.
- [4] B&B Electronics, “RS422/485 Application Note”, 1997.
- [5] Modbus.org, “MODBUS Application Protocol Specification” V1.1, 2003, <http://www.modbus.org>.
- [6] FACTS Engineering, “Universal Isolate Network Adapter Data Sheet”, 2000.
- [7] Radio observatorio de Jicamarca, “PA & DRIVER Control and Operation”.
- [8] Texas Instruments Incorporated, “SN75176A Differential Bus Transceiver – Data Sheet” 1998.
- [9] Microchip Technology, “PIC16F87XA Data Sheet”, 2003.
- [10] STMicroelectronics, “ULN2064B Quad Darlington switches – Data Sheet”, 2003.
- [11] Mike Garbutt, “Asynchronous Communications with the PICmicro® USART”, Microchip Technology, AN774, 2003.
- [12] Automationdirect.com, “D3-350 CPU User Manual”, 2002.
- [13] Automationdirect.com, “Lookout Direct – Getting starting guide”, First Edition, Rev A, 2002.

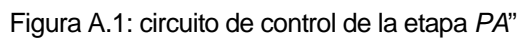
ANEXOS

- A Planos modificados del circuito de control de las etapas *PA* y *Driver*.
- B Circuitos de la estación remota.
- C Programa del microcontrolador PIC.
- D Programa del PLC.

ANEXO A

Planos modificados del circuito de control de las etapas *PA* y *Driver*

Las figuras A.1 y A.2 muestran respectivamente los circuitos de control modificados de las etapas *PA* y *Driver* del transmisor N° 3 del ROJ, en donde se muestran los relés instalados y su conexión con los módulos del PLC.



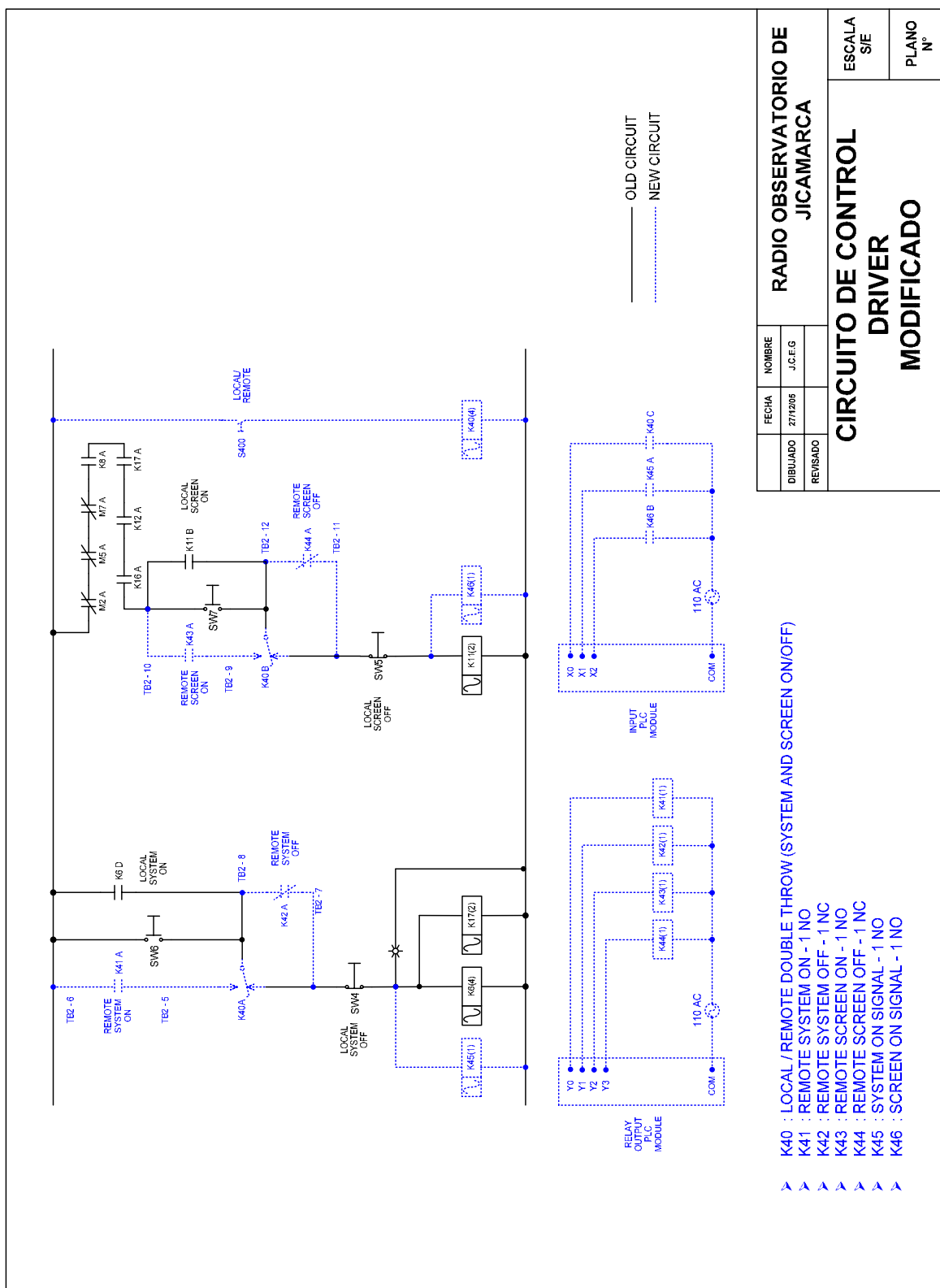


Figura A.2: Circuito de control de la etapa *Driver*

ANEXO B

Circuitos de la estación remota

A continuación se muestran los circuitos esquemáticos de la estación remota y los respectivos circuitos impresos. Al final se muestra una foto real del prototipo de la estación remota.

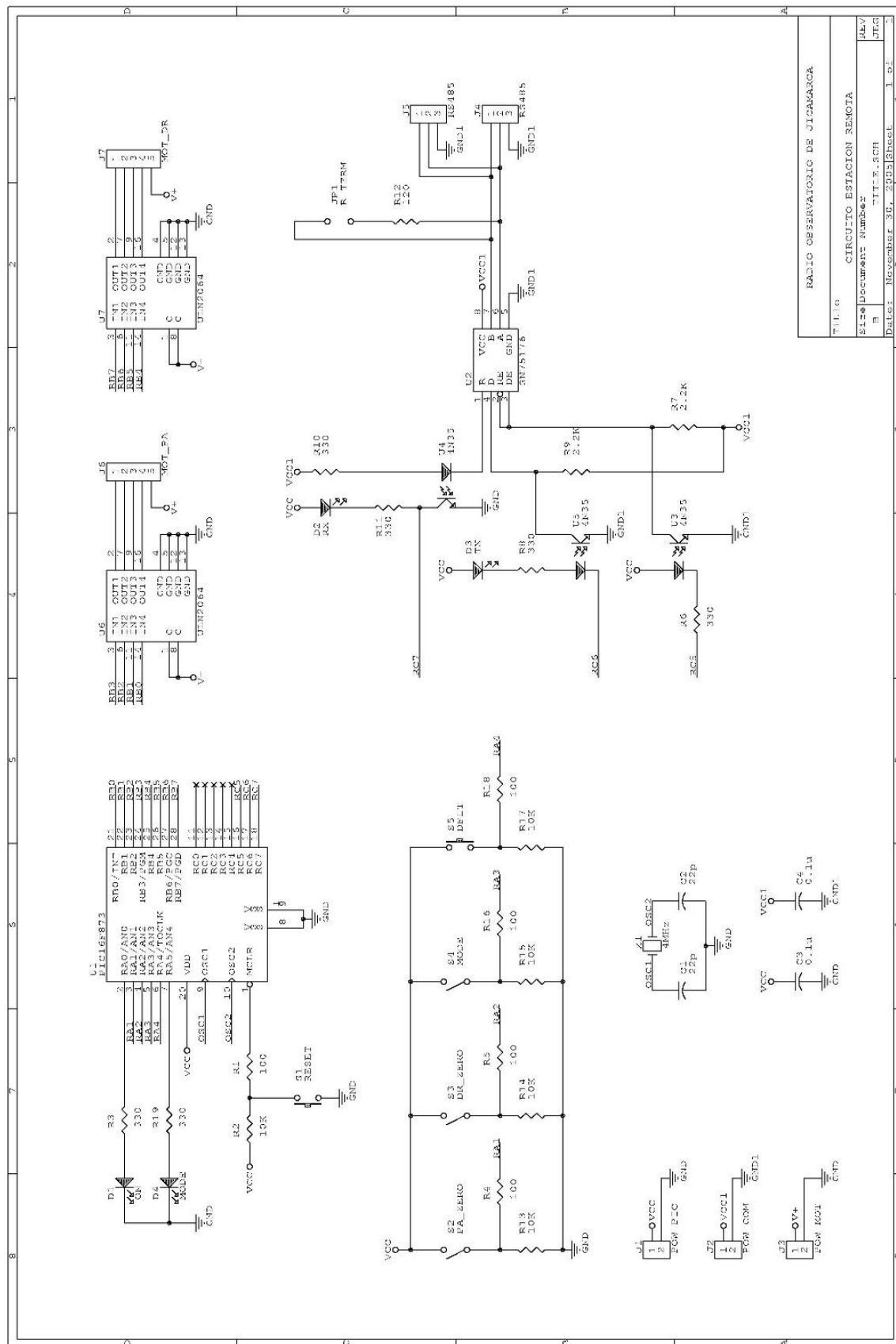


Figura B.1: Esquemático estación remota”

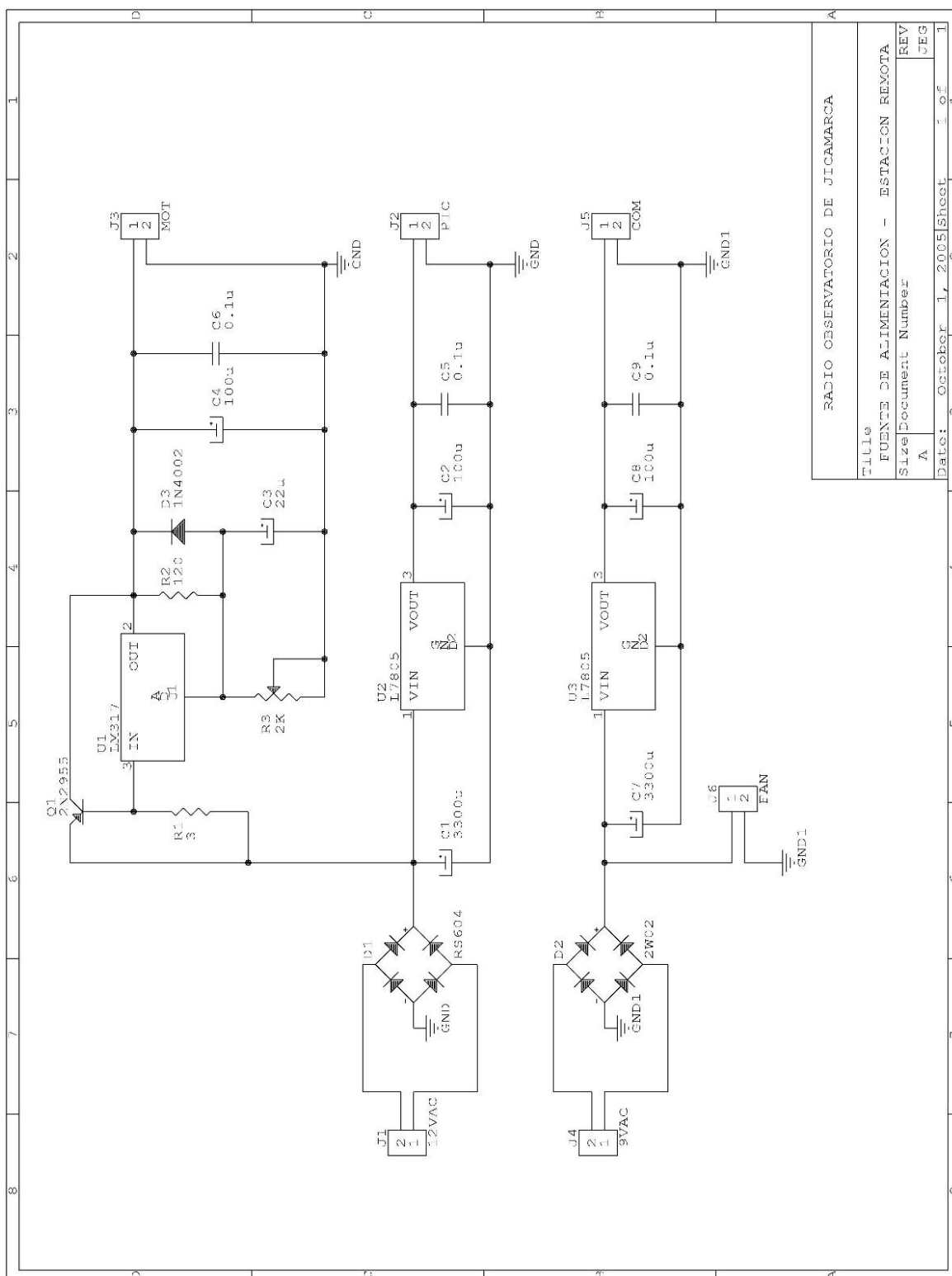


Figura B.2: Esquemático de la fuente de la estación remota"

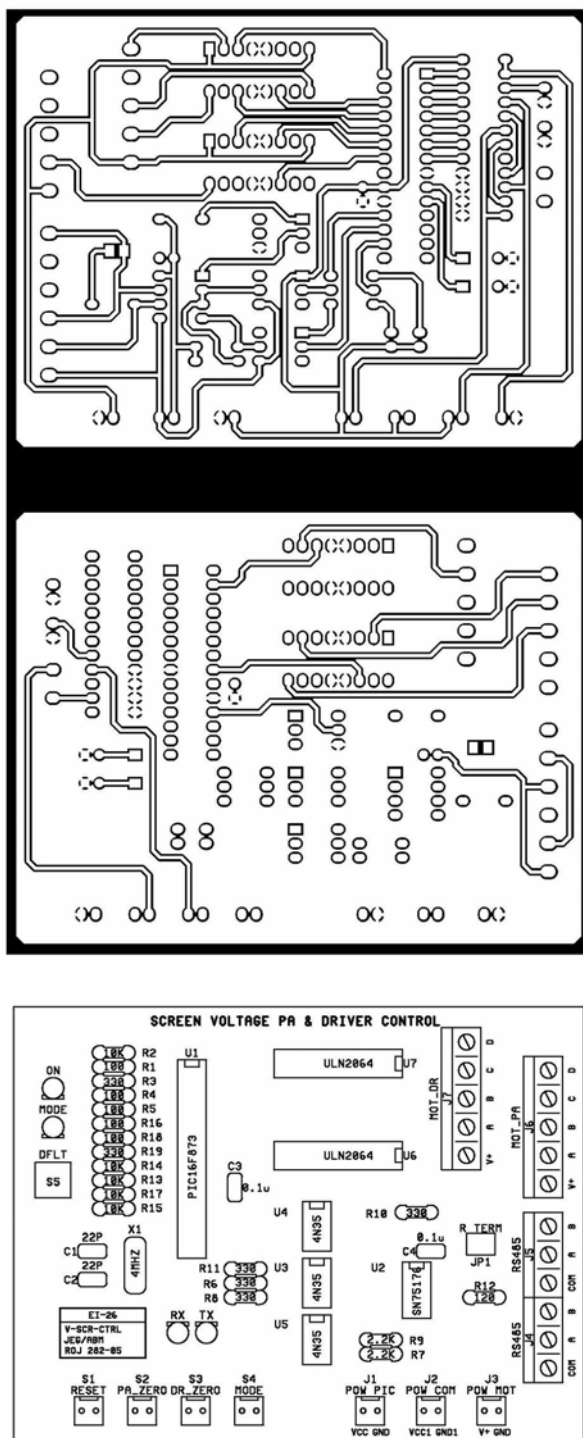


Figura B.3: Circuito impreso de la estación remota



Figura B.4: Prototipo de la estación remota (vista superior)



Figura B.5: Prototipo de la estación remota (vista frontal)

ANEXO C

Programa del microcontrolador PIC

El código fuente que se lista a continuación corresponde al programa del microcontrolador PIC 16F873A, el programa ha sido desarrollado utilizando lenguaje maquina (*Assembler*). La herramienta utilizada para editar depurar y compilar el programa fue MPLAB V7.00 for Windows este software es proporcionado por Microchip (fabricante de los microcontroladores PIC) de forma gratuita.

```

;*****
;PROGRAMA PARA EL CONTROL DE LOS MOTORES PASO A PASO QUE AJUSTAN *
;LAS TENSIONES DE PANTALLA DE LAS ETAPAS PA Y DRIVER DEL TRANSMISOR *
;*****

LIST          P=16F873A          ;PIC UTILIZADO
__CONFIG      _XT_OSC & _WDT_OFF & _BODEN_OFF & _LVP_OFF

#include       "P16F873A.INC"          ;ENCABEZADO NECESARIO

;*****
;CONSTANTES DEL PROGRAMA *
;*****

#define BANK0      BCF      STATUS,RP0      ;SELECCIONA EL BANK0
#define BANK1      BSF      STATUS,RP0      ;SELECCIONA EL BANK1
#define BANK2      BSF      STATUS,RP1      ;SELECCIONA EL BANK2
#define CLR_CARRY  BCF      STATUS,C
#define PA_ZERO    PORTA,1                ;ENTRADA DIGITAL
#define DRV_ZERO   PORTA,2                ;ENTRADA DIGITAL
#define RXON       BSF      RCSTA,CREN      ;HABILITA RECEPCION POR LA USART
#define RXOFF      BCF      RCSTA,CREN      ;DESHABILITA RECEPCION POR LA USART
#define TXON       BSF      TXSTA,TXEN      ;HABILITA TRANSMISION POR LA USART
#define TXOFF      BCF      TXSTA,TXEN      ;DESHABILITA TRANSMISION POR LA USART
#define RTSON      BSF      PORTC,5         ;HABILITA TRANSMISION AL BUSRS485
#define RTSOFF     BCF      PORTC,5         ;DESHABILITA TRANSMISION AL BUS RS485

;*****
;VARIABLES DEL PROGRAMA *
;*****

CBLOCK          0X20
STEP_PA                      ;PASO ACTUAL DEL MOTOR PA
STEP_DRV                     ;PASO ACTUAL DEL MOTOR DRIVER
PA_TABLE_TMP                 ;VARIABLE TEMPORAL
DRV_TABLE_TMP                ;VARIABLE TEMPORAL
CRC_HI                       ;BYTE MAS SIGNIFICATIVO DEL CRC
CRC_LO                       ;BYTE MENOS SIGNIFICATIVO DEL CRC
CONT_CRC                     ;CONTADOR PARA EL CALCULO DEL CRC
W_TEMP                      ;VARIABLE TEMPORAL PARA GUARDAR W
ST_TEMP                     ;VARIABLE TEMPORAL PARA GUARDAR STATUS
CONT_1                      ;VARIABLE PARA DELAYS
CONT_2                      ;VARIABLE PARA DELAYS
CONT_3                      ;VARIABLE PARA DELAYS
CONT_4                      ;VARIABLE PARA DELAYS
CONT_F                      ;CONTADOR PARA FUNCIONES MODBUS
CONFIGG                     ;VARIABLE DE CONFIGGURACION
CONT_DELAY1.5B              ;VARIABLE EN DELAY1.5B
RPTACONT                    ;CONTADOR DEL DELAY DE RESPUESTA
ERR_CODE                    ;CODIGO DE EXCEPTION MODBUS
REG_TEMP                   ;VARIABLE TEMPORAL PARA FUNCIONES MODBUS
CONT_RX                    ;CONTADOR DEL NUMERO DE BYTES RECIBIDOS
CONT_BUFFER                ;CONTADOR DEL BUFFER
CONT_F10                   ;CONTADOR PARA FUNCION MODBUS H'10'
CONT_ERR                   ;CONTADOR DE PASOS POR ERROR
ENDC

CBLOCK          0X40
ADDRESS_RX                 ;VARIABLES UTILIZADAS COMO BUFFER EN LA
FUNCTION                   ;RECEPCION DE UN MENSAJE, EL BUFFER ESTA
BYTE1                     ;LIMITADO HA RECIBIR UN MAXIMO DE 16 BYTES
BYTE2                     ;POR MENSAJE
BYTE3
BYTE4
BYTE5
BYTE6
BYTE7
BYTE8
BYTE9
BYTE10
BYTE11
BYTE12
BYTE13
BYTE14
ENDC

```

```

CBLOCK      0X50                                ;VARIABLES QUE PUEDEN SER LEIDAS Y ESCRITAS
                                                ;VIA MODBUS

NOUSE:8
POSPA_RX                                ;REF MODBUS:X0009, POSICION DEL MOTOR PA rx
POSDRV_RX                                ;REF MODBUS:X0010, POSICION DEL MOTOR DRIVER rx
RESET_PIC                                ;REF MODBUS:X0011, RESET REMOTO DE LOS MOTORES
POSPA                                ;REF MODBUS:X0012, POSICION ACTUAL DEL MOTOR PA
POSDRV                                ;REF MODBUS:X0013, POSICION ACTUAL DEL MOTOR DRIVER
ADDRESS                                ;REF MODBUS:X0014, DIRECCION DE LA ESTACION REMOTA
BADCOM                                ;REF MODBUS:X0015, CONT DE MENSAJES CON CRC ERRONEO
BADMSG                                ;REF MODBUS:X0016, CONT DE EXCEPTIONS MODBUS
MSGs_LO                                ;REF MODBUS:X0017, CONT DE MENSAJES RECIBIDOS
MSGs_HI                                ;REF MODBUS:X0018, CONT DE MENSAJES RECIBIDOS
MSGsOK_LO                                ;REF MODBUS:X0019, CONT DE MENSAJES RECIBIDOS OK
MSGsOK_HI                                ;REF MODBUS:X0020, CONT DE MENSAJES RECIBIDOS OK
MODE                                ;REF MODBUS:x0021, INDICA MODO DE OPERACION
ENDC

;*****
;VECTOR DE RESET *
;*****

                ORG 0X00
                CLRF      PCLATH
                GOTO      RESET

;*****
;VECTOR DE INTERRUPCION *
;*****

                ORG 0X04
                GOTO      INTERRUPT

;*****
;TABLAS DE PULSOS A MEDIO PASO PARA LOS MOTORES PA Y DRIVER *
;*****
;REQUIERE      :POSICION DE LA TABLA EN W                                *
;RETORNA      :VALOR DE LA TABLA EN W                                *
;*****

PA_TABLE      ADDWF      PCL,F
              NOP
              RETLW      B'00000001'
              RETLW      B'00000011'
              RETLW      B'00000010'
              RETLW      B'00000110'
              RETLW      B'00000100'
              RETLW      B'00001100'
              RETLW      B'00001000'
              RETLW      B'00001001'

DRV_TABLE      ADDWF      PCL,F
              NOP
              RETLW      B'10000000'
              RETLW      B'11000000'
              RETLW      B'01000000'
              RETLW      B'01100000'
              RETLW      B'00100000'
              RETLW      B'00110000'
              RETLW      B'00010000'
              RETLW      B'10010000'

;*****
;TABLAS DE CONFIGURACION *
;*****
;REQUIERE      :POSICION DE LA TABLA EN W                                *
;RETORNA      :VALOR DE LA TABLA EN W                                *
;*****

BPS_TABLE      ADDWF      PCL,F
              RETLW      D'51'                                ;VALOR PARA 4800BPS
              RETLW      D'25'                                ;VALOR PARA 9600BPS
              RETLW      D'12'                                ;VALOR PARA 19200BPS

PRESCALER      ADDWF      PCL,F
              RETLW      B'10000100'                        ;PRESCALER DEL TMR0 EN 1:32
              RETLW      B'10000011'                        ;PRESCALER DEL TMR0 EN 1:16

```



```

        RETLW    B'10000010'          ;PRESCALER DEL TMR0 EN 1:8

CONT1_TABLE ADDWF    PCL,F
        RETLW    D'46'                ;VALOR PARA 4800BPS
        RETLW    D'22'                ;VALOR PARA 9600BPS
        RETLW    D'10'                ;VALOR PARA 19200BPS

;*****
;RUTINA DE INTERRUPCION POR RECEPCION EN LA USART *
;*****

INTERRUPT MOVWF    W_TEMP              ;GUARDA EL REGISTRO W
        MOVF     STATUS,W              ;LEE EL REGISTRO STATUS
        MOVWF    ST_TEMP              ;GUARDA EL REGISTRO STATUS
        MOVF     RCREG,W              ;LEE EL BYTE RECIBIDO (DIRECCION DEL ESCLAVO)
        MOVWF    ADDRESS_RX           ;GUARDA EL BYTE RECIBIDO
        BTFSS    INTCON,T0IF          ;OVERFLOW DEL TMR0? (OCURRE CADA 3.5 BYTE)
        GOTO     INT_OUT              ;NO: SALE DE LA INTERRUPCION
        MOVF     ADDRESS_RX,W         ;SI: LEE LA DIRECCION RECIBIDA ADDRESS_RX
        XORWF    ADDRESS,W            ;COMPARA ADDRESS_RX CON ADDRESS
        BTFSS    STATUS,Z             ;DIRECCION OK?
        GOTO     INT_OUT              ;NO: SALE DE LA INTERRUPCION
        INCF     MSGS_LO,F            ;SI: INCREMENTA CONTADOR DE MENSAJES (BYTE BAJO)
        BTFSC    STATUS,Z             ;VALOR DE MSGS_LO = 0?
        INCF     MSGS_HI,F            ;SI: INCREMENTA CONTADOR DE MENSAJES (BYTE ALTO)
        MOVLW    H'40'                ;NO: LEE EL VALOR 0X40
        MOVWF    CONT_RX              ;CONT_RX = 0X40 (INICIO DEL BUFFER)
        CALL     RX                   ;SUBROUTINA DE RECEPCION DEL MENSAJE
        MOVF     CONT_RX,W            ;LEER CONTADOR DE BYTES
        XORLW    H'50'                ;COMPARA CON EL VALOR 0X50
        BTFSC    STATUS,Z             ;NUMERO DE BYTES <16
        GOTO     BAD_COM              ;NO: IR A RUTINA DE MENSAJE ERRONEO
        CALL     CRC_RX               ;SUBROUTINA QUE CALCULA EL CRC DEL MSG RECIBIDO
        DECF     CONT_RX,W            ;SI: W = CONT_RX - 1
        MOVWF    FSR                 ;USO DE DIRECCIONAMIENTO INDIRECTO
        MOVF     INDF,W              ;LEE PENULTIMO BYTE (BYTE BAJO DEL CRC)
        XORWF    CRC_LO,W            ;COMPARA PENULTIMO BYTE RECIBIDO CON CRC_LO
        BTFSS    STATUS,Z             ;REGISTROS DIFERENTES??
        GOTO     BAD_COM              ;SI: IR A RUTINA DE MENSAJE ERRONEO
        MOVF     CONT_RX,W            ;NO: LEE CONTADOR DE MENSAJE
        MOVWF    FSR                 ;USO DE DIRECCIONAMIENTO INDIRECTO
        MOVF     INDF,W              ;LEE ULTIMO BYTE (CORRESPONDE AL BYTE ALTO DEL CRC)
        XORWF    CRC_HI,W            ;COMPARA ULTIMO BYTE RECIBIDO CON CRC_HI
        BTFSS    STATUS,Z             ;REGISTROS DIFERENTES?
        GOTO     BAD_COM              ;SI: IR A RUTINA DE MENSAJE ERRONEO
        MOVF     FUNCTION,W          ;NO: LEE SEGUNDO BYTE RECIBIDO (FUNCION)
        XORLW    H'03'                ;COMPARA FUNCION CON VALOR 0X03
        BTFSC    STATUS,Z             ;REGISTROS IGUALES?
        GOTO     FUNCTION_03         ;SI: IR A RUTINA DE LA FUNCION 0X03
        MOVF     FUNCTION,W          ;NO: LEE SEGUNDO BYTE RECIBIDO (FUNCION)
        XORLW    H'10'                ;COMPARA FUNCION CON VALOR 0X10
        BTFSC    STATUS,Z             ;REGISTROS IGUALES?
        GOTO     FUNCTION_10         ;SI: IR A RUTINA DE LA FUNCION 0X10
        GOTO     BAD_FUN              ;NO: IR A RUTINA DE FUNCION ERRONEA

;*****
;RUTINA PARA SALIR DE LA INTERRUPCION *
;*****

INT_OUT  MOVLW    D'36'                ;LEE VALOR 36
        MOVWF    TMR0                 ;OVERFLOW DE TMR0 EN (255-36)*PRESCALER = 3.5BYTE
        MOVF     ST_TEMP,W            ;LEE REGISTRO STATUS GUARDADO
        MOVWF    STATUS               ;RESTAURA REGISTRO STATUS
        MOVF     W_TEMP,W            ;RESTAURA REGISTRO W
        BCF      PIR1,RCIF            ;HABILITA INTERRUPCION POR RX
        BCF      INTCON,T0IF          ;LIMPIA BANDERA DE OVERFLOW EN TMR0
        RXON                      ;HABILITA RECEPCION POR LA USART
        RETFIE                       ;SALE DE LA INTERRUPCION

;*****
;SUBROUTINA PARA RECEPCION DE UN MENSAJE *
;*****
;REQUIERE    :NINGUNO *
;RETORNA     :NUMERO DE BYTES EN CONT_RX *
;*****

RX        CALL     DELAY_1.5B         ;SUBROUTINA RETARDO DE 1.5B

```

```

CALL    DELAY_100                ;SUBROUTINA RETARDO DE 100us
BTFSS   PIR1,RCIF                ;BYTE RECIBIDO EN LA USART?
RETURN
INCF    CONT_RX,F                ;SI: INCREMENTA CONTADOR DE BYTES RECIBIDOS
MOVF    CONT_RX,W                ;LEE CONTADOR DE BYTES
MOVWF   FSR                      ;USO DIRECCIONAMIENTO INDIRECTO
MOVF    RCREG,W                  ;LEER BYTE RECIBIDO
MOVWF   INDF                     ;GUARDAR BYTE RECIBIDO
MOVF    CONT_RX,W                ;LEER CONTADOR DE BYTES
XORLW   H'50'                    ;COMPARA CON EL VALOR 0X50
BTFSC   STATUS,Z                 ;NUMERO DE BYTES <16
RETURN
GOTO    RX                       ;SI: ESPERA SIGUIENTE BYTE

;*****
;SUBROUTINA PARA EL CALCULO DEL CRC DEL MENSAJE RECIBIDO *
;*****

CRC_RX   MOVLW   H'FF'            ;LEE 0XFF (VALOR INICIAL PARA REGISTROS DEL CRC)
         MOVWF   CRC_HI           ;
         MOVWF   CRC_LO           ;
         MOVLW   H'40'            ;LEE VALOR 0X40 (INICIO DEL BUFFER)
         MOVWF   CONT_BUFFER      ;CONT_BUFFER = 0X40
LOOP_RX  MOVF    CONT_BUFFER,W    ;LEE CONT_BUFFER
         MOVWF   FSR              ;USO DE DIRECCIONAMIENTO INDIRECTO
         MOVF    INDF,W           ;LEE BYTE DEL BUFFER
         CALL    CRC              ;SUBROUTINA CRC
         INCF    CONT_BUFFER,F    ;INCREMENTA CONTADOR DE BUFFER
         MOVF    CONT_BUFFER,W    ;LEE CONTADOR DE BUFFER
         SUBWF   CONT_RX,W        ;W = CONT_RX - CONT_BUFFER
         XORLW   H'01'            ;
         BTFSS   STATUS,Z         ;ULTIMO BYTE DEL MENSAJE??
         GOTO    LOOP_RX         ;NO: SIGUIENTE BYTE
         RETURN

;*****
;ROUTINA PARA LA FUNCION MODBUS 0X03 "READ HOLDING REGISTER *
;*****

FUNCTION_03 MOVF   MODE,W          ;LEE MODO DEL PIC
            BTFSS  STATUS,Z
            GOTO   F03_EEPROM

F03_RAM    RXOFF
            MOVLW  H'50'            ;LEE VALOR 0X50
            ADDWF  BYTE2,F          ;BYTE2 = BYTE2 + 0X50
            CLR    CARRY
            MOVF   BYTE1,W          ;LEE BYTE1 DEL MENSAJE
            XORLW  H'00'            ;COMPARA BYTE1 CON 0X00
            BTFSS  STATUS,Z         ;IGUALES??
            GOTO   BAD_REG          ;NO: IR A RUTINA DE REGISTRO ERRONEO
            MOVF   BYTE2,W          ;SI: LEE BYTE2 DEL MENSAJE
            ADDWF  BYTE4,W          ;W = BYTE2 + BYTE4
            BTFSC  STATUS,C         ;W > 255?
            GOTO   BAD_REG          ;NO: IR A RUTINA DE REGISTRO ERRONEO
            MOVF   BYTE3,W          ;SI: LEE BYTE3 DEL MENSAJE
            XORLW  H'00'            ;COMPARA BYTE3 CON EL VALOR 0X00
            BTFSS  STATUS,Z         ;IGUALES?
            GOTO   BAD_DAT          ;NO: IR A RUTINA DE REGISTRO ERRONEO
            MOVF   BYTE4,W          ;SI: LEE BYTE4 DEL MENSAJE
            SUBLW  H'10'            ;COMPARA BYTE4 CON VALOR 0X10 (NUMERO DE BYTES)
            BTFSS  STATUS,C         ;MENOR A H'10'?
            GOTO   BAD_REG          ;NO: IR A RUTINA DE REGISTRO ERRONEO
            INCF   MSGSOK_LO,F      ;SI: INCREMENTAR BYTE BAJO DEL CONT DE MENSAJES OK
            BTFSC  STATUS,Z         ;CONTADOR IGUAL A CERO?
            INCF   MSGSOK_HI,F      ;SI: INCREMENTAR BYTE ALTO DEL CONT DE MENSAJES OK
            CLRF   CONT_F           ;NO: LIMPIA CONT_F
            CALL   DELAY_RPTA       ;RETARDO DE RESPUESTA
            MOVLW  H'FF'            ;LEE EL VALOR 0XFF
            MOVWF  CRC_HI           ;
            MOVWF  CRC_LO           ;
            MOVF   ADDRESS,W        ;LEER DIRECCION DE LA ESTACION REMOTA
            CALL   CRC              ;ACTUALIZA BYTES DEL CRC
            MOVF   ADDRESS,W        ;LEER DIRECCION DE LA ESTACION REMOTA
            CALL   TX               ;ENVIA DIRECCION DE LA ESTACION REMOTA
            MOVF   FUNCTION,W       ;LEE FUNCION MODBUS
            CALL   CRC              ;ACTUALIZA BYTES DEL CRC

```

```

MOVWF    FUNCTION,W      ;LEE FUNCION MODBUS
CALL     TX              ;ENVIA FUNCION MODBUS
MOVWF    BYTE4,W         ;LEE BYTE 4 DEL MENSAJE
ADDWF    BYTE4,W         ;W = BYTE4 + BYTE4
CALL     CRC             ;ACTUALIZA BYTES DEL CRC
MOVWF    BYTE4,W         ;LEE BYTE4 DEL MENSAJE
ADDWF    BYTE4,W         ;W = BYTE4 + BYTE4
CALL     TX              ;ENVIA NUMERO DE REGISTROS
LOOP_F3RAM MOVWF    H'00'  ;LEE VALOR 0X00
CALL     CRC             ;ACTUALIZA BYTES DEL CRC
MOVWF    H'00'          ;LEE VALOR 0X00
CALL     TX              ;ENVIA VALOR 0X00
MOVWF    BYTE2,W         ;LEE BYTE 2 DEL MENSAJE
ADDWF    CONT_F,W        ;W = CONT_F + BYTE2
MOVWF    FSR             ;UTILIZA DIRECCIONAMIENTO INDIRECTO
MOVWF    INDF,W          ;LEE VALOR DEL REGISTRO
MOVWF    REG_TEMP        ;GUARDA VALOR DEL REGISTRO EN REG_TEMP
CALL     CRC             ;ACTUALIZA BYTES DEL CRC
MOVWF    REG_TEMP,W      ;LEE VALOR DE REG_TEMP
CALL     TX              ;ENVIA VALOR DEL REGISTRO
INCF     CONT_F,F        ;INCREMENTA CONTADOR DE BYTES
MOVWF    CONT_F,W        ;LEE CONTADOR DE BYTES
XORWF    BYTE4,W         ;COMPARA CONTADOR DE BYTES CON BYTE4
BTFSS    STATUS,Z        ;IGUALES?
GOTO     LOOP_F3RAM      ;NO: SIGUIENTE REGISTRO
CALL     DELAY_100        ;SI: RETARDO DE 100us
MOVWF    CRC_LO,W        ;LEE BYTE BAJO DEL CRC
CALL     TX              ;ENVIA BYTE BAJO DEL CRC
CALL     DELAY_100        ;RETARDO DE 100us
MOVWF    CRC_HI,W        ;LEE BYTE ALTO DEL CRC
CALL     TX              ;ENVIA BYTE ALTO DEL CRC
GOTO     INT_OUT         ;SALE DE LA INTERRUPCION

F03_EEPROM RXOFF
CLR_CARRY
;
MOVWF    BYTE1,W         ;LEE BYTE1 DEL MENSAJE
XORLW    H'00'          ;COMPARA BYTE1 CON 0X00
BTFSS    STATUS,Z        ;Z = 1?
GOTO     BAD_REG         ;NO: IR A RUTINA DE REGISTRO ERRONEO
MOVWF    BYTE2,W         ;SI: LEE BYTE2 DEL MENSAJE
ADDWF    BYTE4,W         ;W = BYTE2+BYTE4
SUBLW    D'127'          ;W = 127 - W
BTFSS    STATUS,C        ;CARRY = 1?
GOTO     BAD_REG         ;SI: IR A RUTINA DE REGISTRO ERRONEO
MOVWF    BYTE3,W         ;NO: LEE BYTE3 DEL MENSAJE
XORLW    H'00'          ;COMPARA BYTE3 CON 0X00
BTFSS    STATUS,Z        ;Z = 1?
GOTO     BAD_REG         ;NO: IR A RUTINA DE REGISTRO ERRONEO
MOVWF    BYTE4,W         ;SI: LEE BYTE4
SUBLW    H'10'          ;COMPARA BYTE4 CON 0X10
BTFSS    STATUS,C        ;CARRY = 1?
GOTO     BAD_REG         ;SI: IR A RUTINA DE REGISTRO ERRONEO
INCF     MSGSOK_LO,F      ;NO: INCREMENTA BYTE BAJO DEL CONT DE MENSAJES OK
BTFSC    STATUS,Z        ;MSGSOK_LO = 0?
INCF     MSGSOK_HI,F      ;SI: INCREMENTA BYTE ALTO DEL CONT DE MENSAJES OK
CLRF     CONT_F          ;CONT_F = 0
CALL     DELAY_RPTA      ;SUBROUTINA DE RETARDO DE RESPUESTA
MOVWF    H'FF'          ;LEE VALOR 0XFF (VALOR INICIAL DEL CRC)
MOVWF    CRC_HI
MOVWF    CRC_LO
;
MOVWF    ADDRESS,W       ;LEE DIRECCION DE LA ESTACION REMOTA
CALL     CRC             ;ACTUALIZA BYTES DEL CRC
MOVWF    ADDRESS,W       ;LEE DIRECCION DE LA ESTACION REMOTA
CALL     TX              ;ENVIA DIRECCION DE LA ESTACION REMOTA
MOVWF    FUNCTION,W      ;LEE BYTE DE FUNCION
CALL     CRC             ;ACTUALIZA BYTES DEL CRC
MOVWF    FUNCTION,W      ;LEE BYTE DE FUNCION
CALL     TX              ;ENVIA VALOR DE LA FUNCION MODBUS
MOVWF    BYTE4,W         ;LEE BYTE4 DEL MENSAJE
ADDWF    BYTE4,W         ;W = BYTE4 + BYTE4
CALL     CRC             ;ACTUALIZA BYTES DEL CRC
MOVWF    BYTE4,W         ;LEE BYTE4 DEL MENSAJE
ADDWF    BYTE4,W         ;W = BYTE4 + BYTE4
CALL     TX              ;ENVIA NUMERO DE BYTES A TRANSMITIR
LOOP_F3EE  MOVWF    H'00'  ;LEE VALOR 0X00
CALL     CRC             ;ACTUALIZA BYTES DEL CRC
MOVWF    H'00'          ;LEE VALOR 0X00

```

```

CALL    TX                                ;ENVIA EL VALOR 0X00
MOVF    BYTE2,W                          ;LEE BYTE2 DEL MENSAJE
ADDWF   CONT_F,W                         ;W = BYTE2 + CONT_F
CALL    RDEEPROM                         ;LEE EL REGISTRO DE LA EEPROM
MOVWF   REG_TEMP                         ;GUARDA REGISTRO EN REG_TEMP
CALL    CRC                             ;ACTUALIZA BYTES DEL CRC
MOVF    REG_TEMP,W                      ;LEE REG_TEMP
CALL    TX                                ;ENVIA REGISTRO LEIDO DE LA EEPROM
INCF    CONT_F,F                         ;CONT_F = CONT_F+1
MOVF    CONT_F,W                         ;LEE CONT_F
XORWF   BYTE4,W                         ;COMPARA CONT_F CON BYTE4
BTFSS   STATUS,Z                        ;Z = 1?
GOTO    LOOP_F3EE                       ;NO: SIGUIENTE BYTE
CALL    DELAY_100                       ;SI: RETARDO DE 100us
MOVF    CRC_LO,W                        ;LEE BYTE BAJO DEL CRC CALCULADO
CALL    TX                                ;ENVIA BYTE BAJO DEL CRC CALCULADO
CALL    DELAY_100                       ;RETARDO DE 100us
MOVF    CRC_HI,W                        ;LEE BYTE ALTO DEL CRC CALCULADO
CALL    TX                                ;ENVIA BYTE ALTO DEL CRC CALCULADO
GOTO    INT_OUT                          ;SALE DE INTERRUPTACION

;*****
;RUTINA PARA LA FUNCION MODBUS 0X10 "WRITE MULTIPLE REGISTER" *
;*****

FUNCTION_10 MOVF    MODE,W                ;LEE MODO DEL PIC
            BTFSS   STATUS,Z
            GOTO    F10_EEPROM

F10_RAM     RXOFF
            MOVLW   H'50'                  ;LEE VALOR 0X50
            ADDWF   BYTE2,F                ;BYTE2 = BYTE2+0X50
            CLRC    CARRY                  ;
            MOVF    BYTE1,W                ;LEE BYTE1 DEL MENSAJE
            XORLW   H'00'                  ;COMPARA CON H'00'
            BTFSS   STATUS,Z                ;IGUALES?
            GOTO    BAD_REG                ;NO: IR A RUTINA DE REGISTRO ERRONEO
            MOVLW   H'50'                  ;SI: LEE H'50'
            SUBWF   BYTE2,W                ;COMPARA BYTE2 CON H'50' (BYTE2 = REGISTRO INICIAL)
            BTFSS   STATUS,C                ;BYTE2 < H'50'
            GOTO    BAD_REG                ;SI: IR A RUTINA DE REGISTRO ERRONEO
            MOVF    BYTE2,W                ;NO: LEE BYTE2
            ADDWF   BYTE4,W                ;BYTE4 = BYTE4 + BYTE2
            SUBLW   D'127'                 ;COMPARA CON 127
            BTFSS   STATUS,C                ;MAYOR A 127
            GOTO    BAD_REG                ;SI: IR A RUTINA DE REGISTRO ERRONEO
            MOVF    BYTE5,W                ;NO: LEE BYTE5
            SUBLW   H'08'                  ;COMPARA CON H'08'
            BTFSS   STATUS,C                ;MAYOR A 8
            GOTO    BAD_REG                ;SI: IR A RUTINA DE REGISTRO ERRONEO
            MOVLW   H'50'                  ;NO: LEE H'50'
            SUBWF   BYTE2,F                ;REGRESA AL VALOR DE BYTE2 INICIAL
            INCF    MSGSOK_LO,F            ;INCREMENTA CONTADOR DE MENSAJES OK
            BTFSC   STATUS,Z                ;
            INCF    MSGSOK_HI,F            ;INCREMENTA CONTADOR DE MENSAJES OK
            CALL    DELAY_RPTA             ;RETARDO DE RESPUESTA
            MOVLW   H'FF'                  ;LEE H'FF'
            MOVWF   CRC_HI                 ;VALOR INICIAL PARA CRC_HI
            MOVWF   CRC_LO                 ;VALOR INICIAL PARA CRC_LO
            MOVF    ADDRESS,W              ;LEE DIRECCION DE ESCLAVO
            CALL    CRC                    ;CALCULA CRC
            MOVF    ADDRESS,W              ;LEE DIRECCION DE ESTACION REMOTA
            CALL    TX                      ;ENVIA DIRECCION DE ESTACION REMOTA
            MOVF    FUNCTION,W             ;LEE BYTE DE FUNCION
            CALL    CRC                    ;CALCULA CRC
            MOVF    FUNCTION,W             ;LEE BYTE DE FUNCION
            CALL    TX                      ;ENVIA BYTE DE FUNCION
            MOVF    BYTE1,W                ;LEE BYTE1 DEL MENSAJE
            CALL    CRC                    ;CALCULA CRC
            MOVF    BYTE1,W                ;LEE BYTE1
            CALL    TX                      ;ENVIA BYTE 1 DEL MENSAJE
            MOVF    BYTE2,W                ;LEE BYTE2 DEL MENSAJE
            CALL    CRC                    ;CALCULA CRC
            MOVF    BYTE2,W                ;LEE BYTE 2 DEL MENSAJE
            CALL    TX                      ;ENVIA BYTE2 DE LMENSAJE
            MOVF    BYTE3,W                ;LEE BYTE 3 DEL MENSAJE
            CALL    CRC                    ;CALCULA CRC

```

```

MOVF    BYTE3,W           ;LEE BYTE 3 DEL MENSAJE
CALL    TX                ;ENVIA BYTE 3 DE MENSAJE
MOVF    BYTE4,W           ;LEE BYTE4 DEL MENSAJE
CALL    CRC                ;CALCULA CRC
MOVF    BYTE4,W           ;LEE BYTE 4 DEL MENSAJE
CALL    TX                ;ENVIA BYTE 4 DEL MENSAJE
CALL    DELAY_100         ;RETARDO DE 100us
MOVF    CRC_LO,W          ;LEE BYTE BAJO DEL CRC
CALL    TX                ;ENVIA BYTE BAJO DEL CRC
CALL    DELAY_100         ;RETARDO DE 100us
MOVF    CRC_HI,W          ;LEE BYTE ALTO DEL CRC
CALL    TX                ;ENVIA BYTE ALTO DEL CRC
CLRF    CONT_F            ;LIMPIA CONTADOR F
CLRF    CONT_F10          ;LIMPIA CONTADOR F
MOVLW   H'50'             ;LEE H'50'
ADDWF   BYTE2,F           ;REDIRECCIONA REGISTRO INICIAL
LOOP_F10RAM MOVLW   H'48'   ;LEE VALOR H'48' (VALOR INICIAL DE BUFFER)
ADDWF   CONT_F10,W        ;CONTADOR F = VALOR INICIAL (48) + 2*N
MOVWF   FSR               ;DIRECCIONAMIENTO INDIRECTO
MOVF    INDF,W            ;LEE BYTE
MOVWF   REG_TEMP          ;GUARDA EN REG TEMPORAL
MOVF    BYTE2,W           ;LEE BYTE2 DEL MENSAJE
ADDWF   CONT_F,W         ;W = CONTADOR F + BYTE2
MOVWF   FSR               ;DIRECCIONAMIENTO INDIRECTO
MOVF    REG_TEMP,W        ;LEE DATO TEMPORAL
MOVWF   INDF              ;GUARDA DATO
MOVLW   H'02'             ;LEE H'02'
ADDWF   CONT_F10,F        ;ACTUALIZA CONTADOR F
INCF    CONT_F,F          ;ACTUALIZA CONTADOR F10
MOVF    BYTE5,W           ;LEE BYTE5 DEL MENSAJE (NUMERO DE BYTES)
XORWF   CONT_F10,W        ;COMPARA CON CONTADOR F10
BTFSS   STATUS,Z          ;IGUALES
GOTO    LOOP_F10RAM       ;NO: SIGUE EN LAZO
GOTO    INT_OUT           ;SI: SALE DE INTERRUPCION

F10_EEPROM RXOFF
CLR_CARRY                ;
MOVF    BYTE1,W           ;LEE BYTE1 DEL MENSAJE
XORLW   H'00'             ;COMPARA CON H'00'
BTFSS   STATUS,Z          ;IGUALES?
GOTO    BAD_REG           ;NO: IR A RUTINA DE REGISTRO ERRONEO
MOVF    BYTE2,W           ;NO: LEE BYTE2
ADDWF   BYTE4,W           ;BYTE4 = BYTE4 + BYTE2
SUBLW   D'127'            ;COMPARA CON 127
BTFSS   STATUS,C          ;MAYOR A 127
GOTO    BAD_REG           ;SI: IR A RUTINA DE REGISTRO ERRONEO
MOVF    BYTE5,W           ;NO: LEE BYTE5
SUBLW   H'08'             ;COMPARA CON H'08'
BTFSS   STATUS,C          ;MAYOR A 8
GOTO    BAD_REG           ;SI: IR A RUTINA DE REGISTRO ERRONEO
INCF    MSGSOK_LO,F       ;NO: INCREMENTA CONTADOR DE MENSAJES OK
BTFSC   STATUS,Z          ;
INCF    MSGSOK_HI,F       ;INCREMENTA CONTADOR DE MENSAJES OK
CALL    DELAY_RPTA        ;RETARDO DE RESPUESTA
MOVLW   H'FF'             ;LEE H'FF'
MOVWF   CRC_HI            ;VALOR INICIAL PARA CRC_HI
MOVWF   CRC_LO            ;VALOR INICIAL PARA CRC_LO
MOVF    ADDRESS,W         ;LEE DIRECCION DE ESCLAVO
CALL    CRC                ;CALCULA CRC
MOVF    ADDRESS,W         ;LEE DIRECCION DE ESTACION REMOTA
CALL    TX                ;ENVIA DIRECCION DE ESTACION REMOTA
MOVF    FUNCTION,W        ;LEE BYTE DE FUNCION
CALL    CRC                ;CALCULA CRC
MOVF    FUNCTION,W        ;LEE BYTE DE FUNCION
CALL    TX                ;ENVIA BYTE DE FUNCION
MOVF    BYTE1,W           ;LEE BYTE1 DEL MENSAJE
CALL    CRC                ;CALCULA CRC
MOVF    BYTE1,W           ;LEE BYTE1
CALL    TX                ;ENVIA BYTE 1 DEL MENSAJE
MOVF    BYTE2,W           ;LEE BYTE2 DEL MENSAJE
CALL    CRC                ;CALCULA CRC
MOVF    BYTE2,W           ;LEE BYTE 2 DEL MENSAJE
CALL    TX                ;ENVIA BYTE2 DE L MENSAJE
MOVF    BYTE3,W           ;LEE BYTE 3 DEL MENSAJE
CALL    CRC                ;CALCULA CRC
MOVF    BYTE3,W           ;LEE BYTE 3 DEL MENSAJE
CALL    TX                ;ENVIA BYTE 3 DE MENSAJE

```

```

        MOVF    BYTE4,W           ;LEE BYTE4 DEL MENSAJE
        CALL    CRC               ;CALCULA CRC
        MOVF    BYTE4,W           ;LEE BYTE 4 DEL MENSAJE
        CALL    TX               ;ENVIA BYTE 4 DEL MENSAJE
        CALL    DELAY_100         ;RETARDO DE 100us
        MOVF    CRC_LO,W         ;LEE BYTE BAJO DEL CRC
        CALL    TX               ;ENVIA BYTE BAJO DEL CRC
        CALL    DELAY_100         ;RETARDO DE 100us
        MOVF    CRC_HI,W         ;LEE BYTE ALTO DEL CRC
        CALL    TX               ;ENVIA BYTE ALTO DEL CRC
        CLRF    CONT_F           ;LIMPIA CONTADOR F
        CLRF    CONT_F10         ;LIMPIA CONTADOR F
LOOP_F10EE MOV LW    H'48'         ;LEE VALOR H'48'
        ADDWF   CONT_F10,W        ;CONTADOR F = VALOR INICIAL (48) + 2*N
        MOVWF   FSR              ;DIRECCIONAMIENTO INDIRECTO
        MOVF    INDF,W           ;LEE BYTE
        MOVWF   REG_TEMP         ;GUARDA EN REG TEMPORAL
        MOVF    BYTE2,W          ;LEE BYTE2 DEL MENSAJE
        ADDWF   CONT_F,W         ;W = CONTADOR F + BYTE2
        CALL    WREEPROM         ;GUARDA DATO
        MOV LW   H'02'          ;LEE H'02'
        ADDWF   CONT_F10,F        ;ACTUALIZA CONTADOR F
        INCF    CONT_F,F         ;ACTUALIZA CONTADOR F10
        MOVF    BYTE5,W          ;LEE BYTE5 DEL MENSAJE (NUMERO DE BYTES)
        XORWF   CONT_F10,W        ;COMPARA CON CONTADOR F10
        BTFS    STATUS,Z         ;IGUALES
        GOTO    LOOP_F10EE       ;NO: SIGUE EN LAZO
        GOTO    INT_OUT          ;SI: SALE DE INT

;*****
;RUTINA PARA ENVIAR UNA EXCEPTION MODBUS *
;*****

EXCEPTION RXOFF
        INCF    BADMSG,F         ;INCREMENTA CONTADOR DE MENSAJES ERRONEOS
        CALL    DELAY_RPTA       ;RETARDO DE RESPUESTA
        MOV LW   H'FF'          ;LEE VALOR H'FF'
        MOVWF   CRC_HI           ;CARGA VALOR DEL CRC
        MOVWF   CRC_LO           ;CARGA VALOR DEL CRC
        MOVF    ADDRESS,W        ;LEE DIRECCION DE ESTACION REMOTA
        CALL    CRC               ;CALCULA CRC
        MOVF    ADDRESS,W        ;LEE DIRECCION DE ESTACION REMOTA
        CALL    TX               ;ENVIA DIRECCION DE ESTACION REMOTA
        MOVF    FUNCTION,W       ;LEE BYTE DE FUNCION
        ADD LW   H'80'           ;LE SUMA 80 (INDICA ERROR)
        CALL    CRC               ;CALCULA CRC
        MOVF    FUNCTION,W       ;LEE BYTE DE FUNCION
        ADD LW   H'80'           ;LE SUMA 80
        CALL    TX               ;ENVIA BYTE DE FUNCION + 80
        MOVF    ERR_CODE,W       ;LEE CODIGO DE ERROR
        CALL    CRC               ;CALCULA CRC
        MOVF    ERR_CODE,W       ;LEE CODIGO DE ERROR
        CALL    TX               ;ENVIA CODIGO DE ERROR
        CALL    DELAY_100         ;RETARDO DE 100us
        MOVF    CRC_LO,W         ;LEE BYTE BAJO DEL CRC
        CALL    TX               ;ENVIA BYTE BAJO DEL CRC
        CALL    DELAY_100         ;RETARDO DE 100 us
        MOVF    CRC_HI,W         ;LEE BYTE ALTO DEL CRC
        CALL    TX               ;ENVIA BYTE ALTO DEL CRC
        GOTO    INT_OUT          ;IR A RUTINA PARA SALIR DE INTERRUPCION

;*****
;RUTINA DE ERROR POR CRC ERRONEO *
;*****

BAD_COM    INCF    BADCOM,F      ;INCREMENTA CONTADOR DE ERRORES
        GOTO    INT_OUT          ;IR A RUTINA PARA SALIR DE INTERRUPCION

;*****
;RUTINA DE ERROR POR FUNCION NO SOPORTADA *
;*****

BAD_FUN    MOV LW   H'01'        ;LEE H'01' CODIGO DE ERROR
        MOVWF   ERR_CODE         ;GUARDA CODIGO DE ERRPR
        GOTO    EXCEPTION        ;IR A GENERACION DE EXCEPTION MODBUS

```

```

;*****
;RUTINA DE ERROR POR REGSITRO ERRONEO *
;*****

BAD_REG      MOVLW    H'02'          ;LEE H'02' CODIGO DE ERROR
              MOVWF    ERR_CODE      ;GUARDA CODIGO DE ERROR
              GOTO     EXCEPTION     ;IR A GENERACION DE EXCEPTION MODBUS

;*****
;RUTINA DE ERROR POR DATO ERRONEO *
;*****

BAD_DAT      MOVLW    H'03'          ;LEE H'03' CODIGO DE ERROR
              MOVWF    ERR_CODE      ;GUARDA CODIGO DE ERROR
              GOTO     EXCEPTION     ;IR A GENERACION DE EXCEPTION MODBUS

;*****
;SUBROUTINA "TX" TRANSMISION DE UN BYTE POR LA USART *
;*****
;REQUIERE     :BYTE A TRANSMITIR EN W          *
;RETORNA      :NADA                          *
;*****

TX            RTSON
              MOVWF    TXREG          ;GUARDA BYTE A TRANSMITIR
              BANK1
              TXON
TX_WAIT       BTFSS    TXSTA,TRMT      ;TRANSMISION TERMINADA?
              GOTO     TX_WAIT        ;NO: SIGUE EN LAZO
              TXOFF
              BANK0
              RTSOFF
              CALL     DELAY_1.5B      ;RETARDO DE 1.5 BYTE
              RETURN

;*****
;SUBROUTINA PARA CALCULAR EL CRC16 DE UN BYTE *
;*****
;REQUIERE     :BYTE EN W          *
;RETORNA      :CRC_LO Y CRC_HI ACTUALIZADOS *
;*****

CRC           XORWF    CRC_LO,F        ;XOR CON EL BYTE
              CLRF     CONT_CRC        ;LIMPIA CONTADOR
CRC_LOOP      CLR_CARRY
              RRF       CRC_HI,F        ;DESPLAZAMIENTO A DERECHA CON CARRY
              RRF       CRC_LO,F        ;DESPLAZAMIENTO A DERECHA CON CARRY
              BTFSC     STATUS,C        ;CARRY=0??
              GOTO     XORPOLY         ;NO: CALCULA POLINOMIO
              INCF      CONT_CRC,F      ;SI: ACTUALIZA CONTADOR
              MOVF      CONT_CRC,W      ;LEE CONTADOR
              XORLW     H'08'          ;COMPARA CON H'08'
              BTFSS     STATUS,Z        ;IGUALES??
              GOTO     CRC_LOOP        ;NO: SIGUE EN LAZO
              RETURN                  ;SI: SALE DE SUBROUTINA
XORPOLY       MOVLW     B'10100000'    ;LEE VALOR 1
              XORWF     CRC_HI,F        ;XOR CON VALOR 1
              MOVLW     B'00000001'    ;LEE VALOR 2
              XORWF     CRC_LO,F        ;XOR CON VALOR 2
              INCF      CONT_CRC,F      ;ACTUALIZA CONTADOR
              MOVF      CONT_CRC,W      ;LEE CONTADOR
              XORLW     H'08'          ;COMPARA CON H'08'
              BTFSS     STATUS,Z        ;IGUALES??
              GOTO     CRC_LOOP        ;NO: SIGUE EN LAZO
              RETURN                  ;SI: SALE DE SUBROUTINA

;*****
;SUBROUTINA PARA LEER UN REGISTRO DE LA MEMORIA EEPROM*
;*****
;REQUIERE     :DIRECCION DEL REGISTRO EN W          *
;RETORNA      :VALOR DEL REGISTRO EN W          *
;*****

RDEEPROM      BANK2
              MOVWF     EEADR          ;GUARDA REGISTRO
              BANK1
              BCF       EECON1,EEPGD   ;SELECCIONA BANCO 3
              ;PUNTERO A MEMORIA EEPROM

```

```

        BSF      EECON1, RD          ;LEE EEPROM
        BANK0
        MOVF     EEDATA, W           ;SELECCIONA BANCO 2
        BCF      STATUS, RP1        ;GUARDA DATO EN W
        RETURN   ;SELECCIONA BANCO 0
        ;SALE DE SUBROUTINA

;*****
;SUBROUTINA PARA ESCRIBIR EN UN REGISTRO DE LA MEMORIA EEPROM*
;*****
;REQUIERE      :DIRECCION DEL REGISTRO EN W      *
;              :VALOR DEL REGISTRO EN REG_TEMP  *
;RETORNA       :NADA                            *
;*****

WREEPROM      BANK2
               BANK1                ;SELECCIONA BANCO3
LOOP_WR        BTFSC    EECON1, WR    ;ESCRITURA EN PROCESO
               GOTO     LOOP_WR      ;SI: ESPERA
               BANK0                ;NO: CONTINUA
               MOVWF     EEADR        ;GUARDA REGISTRO
               BCF      STATUS, RP1   ;ACTIVA BANCO 0
               MOVF     REG_TEMP, W   ;DATO A W
               BANK2
               MOVWF     EEDATA       ;GUARDO DATO EN EEPROM
               BANK1                ;ACTIVO BANCO 3
               BCF      EECON1, EEPGD ;PUNTERO A MEMORIA EEPROM
               BSF      EECON1, WREN   ;HABILITA ESCRITURA
               BCF      INTCON, GIE    ;DESABILITA INTERRUPTOS GLOBALES
               MOVLW    H'55'         ;
               MOVWF     EECON2       ;SECUENCIA
               MOVLW    H'AA'         ;REQUERIDA
               MOVWF     EECON2       ;
               BSF      EECON1, WR    ;COMIENZA ESCRITURA
               BSF      INTCON, GIE    ;HABILITA INTERRUPTOS GLOBALES
               BCF      EECON1, WREN   ;DESHABILITA ESCRITURA
               BCF      STATUS, RP1   ;SELECCIONA BANCO 1
               BANK0
               RETURN   ;SALE DE SUBROUTINA

;*****
;RETARDO ENTRE BYTES *
;*****

DELAY_1.5B     MOVF      CONT_DELAY1.5B, W
               MOVWF     CONT_1
LOOP_1.5B2      MOVLW    H'10'
               MOVWF     CONT_2
LOOP_1.5B1      DECFSZ   CONT_2, F
               GOTO     LOOP_1.5B1
               DECFSZ   CONT_1, F
               GOTO     LOOP_1.5B2
               RETURN

;*****
;RETARDO DE RESPUESTA *
;*****

DELAY_RPTA     MOVF      RPTACONT, W
               BTFSC    STATUS, Z
               RETURN
               MOVWF     CONT_1
LOOP_RPTA2      MOVLW    H'02'
               MOVWF     CONT_2
LOOP_RPTA1      DECFSZ   CONT_2, F
               GOTO     LOOP_RPTA1
               DECFSZ   CONT_1, F
               GOTO     LOOP_RPTA2
               RETURN

;*****
;RETAROD DE 100us *
;*****

DELAY_100      MOVLW    H'21'
               MOVWF     CONT_1
LOOP_100        DECFSZ   CONT_1, F
               GOTO     LOOP_100

```



```

RETURN

;*****
;PARPADEO DE LED CADA 500us *
;*****

LED_ONOFF    MOVLW    H'01'
              XORWF    PORTA,F          ;ACTIVA/DESACTIVA PORTA<0>
              CALL     DELAY_STEP        ;RETARDO DE 100ms
              CALL     DELAY_STEP        ;RETARDO DE 100ms
              CALL     DELAY_STEP        ;RETARDO DE 100ms
              CALL     DELAY_STEP        ;RETARDO DE 100ms
              CALL     DELAY_STEP        ;RETARDO DE 100ms
              RETURN

;*****
;CONFIGURACION DE USART *
;*****

USART_CFG     MOVLW    B'10110000'      ;SPEN=1(SERIAL PORT ENABLE),RX9=1(9BIT RX ENABLE)
              ;CREN=1(CONT RX ENABLE)
              MOVWF    RCSTA             ;CONFIGURACION DE RECEPCION
              BANK1
              MOVLW    B'01000101'      ;TX9=1(9BIT), BRGH=1(HIGH SPEED), 9BIT=1
              MOVWF    TXSTA            ;CONFIGURACION DE TRANSMISION
              BANK0
              RETURN

;*****
;SUBROUTINAS PARA REGRESAR MOTORES A POSICION CERO *
;*****

RST_PA        INCF     CONT_ERR,F        ;CUENTA PASOS PARA DETECTAR ERROR
              MOVF     CONT_ERR,W        ;LEE CONTADOR DE PASOS
              XORLW    D'160'            ;COMPARA CON 180
              BTFSC    STATUS,Z          ;IGUALES??
              RETURN                    ;SI: SALE
              BTFSC    PA_ZERO           ;NO: MOTOR PA EN POSICION CERO?
              RETURN                    ;SI: RETORNA
              INCF     POSPA,F           ;NO: INCREMENTA POSPA
              CALL     UPDATE_PA         ;SUBROUTINA DE ACTUALIZACION DEL MOTOR PA
              CALL     DELAY_STEP
              GOTO     RST_PA            ;SIGUE EN LAZO

RST_DRV        INCF     CONT_ERR,F        ;CUENTA PASOS PARA DETECTAR ERROR
              MOVF     CONT_ERR,W        ;LEE CONTADOR DE PASOS
              XORLW    D'160'            ;COMPARA CON 180
              BTFSC    STATUS,Z          ;IGUALES??
              RETURN                    ;SI: SALE
              BTFSC    DRV_ZERO          ;NO: MOTOR DRIVER EN POSICION CERO?
              RETURN                    ;SI: RETORNA
              INCF     POSDRV,F          ;NO: INCREMENTA POSPA
              CALL     UPDATE_DRV        ;SUBROUTINA DE ACTUALIZACION DEL MOTOR DRIVER
              CALL     DELAY_STEP
              GOTO     RST_DRV           ;SIGUE EN LAZO

;*****
;SUBROUTINAS PARA CARGAR VALORES POR DEFECTO *
;*****

DFLT_ADD       MOVLW    D'127'           ;VALOR POR DEFECTO PARA DIRECCION DE ESTACION
              MOVWF    ADDRESS           ;GUARDA VALOR
              MOVWF    REG_TEMP          ;GUARDA VALOR
              MOVLW    H'00'            ;REGISTRO DE LA EEPROM
              GOTO     WREEPROM          ;SUBROUTINA PARA ESCRIBIR EN LA EEPROM

DFLT_BPS       MOVLW    H'01'           ;VALOR POR DEFECTO PARA CONFIGG
              MOVWF    CONFIGG          ;GUARDA VALOR
              MOVWF    REG_TEMP          ;GUARDA VALOR
              MOVLW    H'01'            ;REGISTRO DE LA EEPROM
              GOTO     WREEPROM          ;SUBROUTINA PARA ESCRIBIR EN LA EEPROM

DFLT_RPTA      CLRF     REG_TEMP         ;VALOR POR DEFECTO PARA RETARDO DE RESPUESTA
              CLRF     RPTACONT          ;LIMPIA CONTADOR
              MOVLW    H'02'            ;REGISTRO DE LA EEPROM
              GOTO     WREEPROM          ;SUBROUTINA PARA ESCRIBIR EN LA EEPROM

```

```

DFLT_RST    CLRF    REG_TEMP        ;VALOR POR DEFECTO PARA VAR DE RESET
            MOVLW   H'03'           ;REGISTRO DE LA EEPROM
            GOTO    WREEPROM        ;SUBROUTINA PARA ESCRIBIR EN LA EEPROM

;*****
;RUTINA DE RESET *
;*****

RESET       CLRF    PORTA            ;LIMPIA PUERTO A
            CLRF    PORTB            ;LIMPIA PUERTO B
            CLRF    PORTC            ;LIMPIA PUERTO C
            CLR_CARRY                ;
            BANK1                    ;
            MOVLW   B'00011110'      ;
            MOVWF   TRISA             ;RA0,RA5 SALIDAS; RA1, RA2, RA3, RA4 ENTRADAS
            MOVLW   B'00000110'      ;
            MOVWF   ADCON1            ;PORTA DIGITAL
            CLRF    TRISB             ;PORTB COMO SALIDA
            MOVLW   B'10000000'      ;RC5, RC6 COMO SALIDA; RC7 COMO ENTRADA
            MOVWF   TRISC             ;
            BSF     PIE1,RCIE         ;INTERRUP POR RECEPCION HABILITADA
            MOVLW   B'01000101'      ;TX9=1(9BIT), BRGH=1(HIGH SPEED), 9BIT=1
            MOVWF   TXSTA             ;CONFIGURACION DE TRANSMISION
            BANK0                    ;
            MOVLW   H'01'            ;LEE VALOR 01
            CALL    RDEEPROM          ;LEE DIRECCION 01 DE LA EEPROM (BAUD RATE)
            MOVWF   CONFIGG           ;
            SUBLW   H'02'            ;
            BTFSS   STATUS,C          ;CONFIGG>02??
            CALL    DFLT_BPS          ;SI: DEFAULT
            MOVF    CONFIGG,W         ;NO: CARGA CONFIGG
            CALL    BPS_TABLE         ;LEE TABLA CON BPS
            BANK1                    ;
            MOVWF   SPBRG             ;GUARDA BAUD RATE
            BANK0                    ;
            MOVF    CONFIGG,W         ;LEE CONFIGG
            CALL    PRESCALER         ;LEE TABLA CON PRESCALER
            BANK1                    ;
            MOVWF   OPTION_REG        ;GUARDA PRESCALER
            BANK0                    ;
            CLR_CARRY                ;
            MOVLW   H'00'            ;LEE VALOR 00
            CALL    RDEEPROM          ;LEE DIRECCION 00 DE LA EEPROM
            MOVWF   ADDRESS           ;GUARDA DIRECCION DE ESCLAVO
            SUBLW   D'127'           ;
            BTFSS   STATUS,C          ;ADDRESS >= 127
            CALL    DFLT_ADD          ;SI: CARGA VALOR POR DEFECTO
            MOVF    CONFIGG,W         ;NO: LEE CONFIGG
            CALL    CONT1_TABLE       ;LEE TABLA DE RETARDO ENTRE BYTES
            MOVWF   CONT_DELAY1.5B    ;GUARDA VALOR
            CLR_CARRY                ;
            MOVLW   H'02'            ;LEE VALOR 02
            CALL    RDEEPROM          ;LEE DIRECCION 02 DE LA EEPROM
            MOVWF   RPTACONT          ;GUARDA RETARDO DE RESPUESTA
            SUBLW   D'200'           ;
            BTFSS   STATUS,C          ;TICKS > 200
            CALL    DFLT_RPTA        ;CARGA VALOR POR DEFECTO
            MOVLW   H'03'            ;LEE VALOR 03
            CALL    RDEEPROM          ;LEE DIRECCION 03 DE LA EEPROM
            BTFSS   STATUS,Z          ;VALOR = 0?
            CALL    DFLT_RST          ;NO: CARGA VALOR POR DEFECTO
            MOVLW   H'01'            ;SI: LEE VALOR 01
            MOVWF   STEP_PA           ;PASO INICIAL DE MOTOR PA
            MOVWF   STEP_DRV          ;PASO INICIAL DE MOTOR DRIVER
            CLRF    MODE              ;VAR = 0
            CLRF    BADCOM            ;VAR = 0
            CLRF    BADMSG            ;VAR = 0
            CLRF    MSGS_LO           ;VAR = 0
            CLRF    MSGS_HI           ;VAR = 0
            CLRF    MSGSOK_LO        ;VAR = 0
            CLRF    MSGSOK_HI        ;VAR = 0
            CLRF    RESET_PIC        ;VAR = 0
            CLRF    POSPA             ;VAR = 0
            CLRF    POSDRV            ;VAR = 0
            CLRF    POSPA_RX          ;VAR = 0
            CLRF    POSDRV_RX         ;VAR = 0

```

	CLRF	CONT_ERR	;VAR = 0
	CALL	LED_ONOFF	;
	CALL	LED_ONOFF	;
	CALL	LED_ONOFF	;
	CALL	LED_ONOFF	;
	CALL	LED_ONOFF	;
	BCF	INTCON,T0IF	;RST FLAG DE TMR0
	MOVLW	D'36'	;LEE VALOR 36
	MOVWF	TMR0	;VALOR INICIAL DE TMR0 (255-36)*PRESCALER = 3.5 CHR
	CALL	USART_CFG	;CONFIGURA LA USART
	BSF	INTCON,PEIE	;HABILITA INTERRUPCIONES GENERALES
	BSF	INTCON,GIE	;HABILITA INTERRUPCIONES DE PERIFERICOS
	CALL	RST_PA	;PONE MOTOR PA EN POSICION CERO
	CLRF	CONT_ERR	;VAR = 0
	CALL	RST_DRV	;PONE MOTOR DRIVER EN POSICION CERO
	BSF	PORTA,0	;ENCIENDE LED ON
;*****			
;RUTINA PRINCIPAL DEL PROGRAMA *			
;*****			
MAIN	BTFSC	PORTA,4	;CARGAR VALORES POR DEFECTO
	GOTO	DEFAULT	;SI: IR A SUBROUTINA
	BTFSC	PORTA,3	;MODE
	GOTO	MODE_CFG	;NO: IR A MODO CONFIGURACION
MODE_STD	BCF	PORTA,5	;LED MODE OFF
	CLRF	MODE	;VAR = 0
	MOVF	RESET_PIC,W	;LEE VARIABLE DE RESET
	BTFSS	STATUS,Z	;VARIABLE IGUAL A CERO??
	GOTO	RESET	;NO: RESET AL PIC
	MOVF	POSPA_RX,W	;SI: LEE POSICION RECIBIDA DEL MOTOR PA
	XORWF	POSPA,W	;COMPARA CON POSICION ACTUAL
	BTFSS	STATUS,Z	;SON IGUALES??
	CALL	UPDATE_PA	;NO: ACTUALIZA POSICION
	MOVF	POSDRV_RX,W	;LEE POSICION RECIBIDA DEL MOTOR PA
	XORWF	POSDRV,W	;COMPARA CON POSICION ACTUAL
	BTFSS	STATUS,Z	;SON IGUALES??
	CALL	UPDATE_DRV	;NO: ACTUALIZA POSICION
	GOTO	MAIN	;SI: IR A RUTINA PRINCIPAL
MODE_CFG	BSF	PORTA,5	;LED STATUS ON
	MOVLW	H'01'	;LEE VALOR 01
	MOVWF	MODE	;GUARDA 01 EN MODE
;	MOVLW	H'03'	;LEE VALOR 03
;	CALL	RDEEPROM	;LEE DIRECCION 03 DE LA EEPROM
;	BTFSS	STATUS,Z	;VARIABLE IGUAL A CERO??
;	GOTO	RESET	;NO: RESET AL PIC
	GOTO	MAIN	;SI: IR A RUTINA PRINCIPAL
DEFAULT	CALL	DELAY_STEP	;RETARDO DE 100ms
	BTFSS	PORTA,4	;SEGURO DE CARGAR VALORES POR DEFECTO
	GOTO	MAIN	;NO: IR A RUTINA PRINCIPAL
	BTFSS	PORTA,4	;SEGURO DE CARGAR VALORES POR DEFECTO
	GOTO	MAIN	;NO: IR A RUTINA PRINCIPAL
	CALL	DELAY_STEP	;RETARDO DE 100ms
	BTFSS	PORTA,4	;SEGURO DE CARGAR VALORES POR DEFECTO
	GOTO	MAIN	;NO: IR A RUTINA PRINCIPAL
	CALL	DELAY_STEP	;RETARDO DE 100ms
	BTFSS	PORTA,4	;SEGURO DE CARGAR VALORES POR DEFECTO
	GOTO	MAIN	;NO: IR A RUTINA PRINCIPAL
	CALL	DELAY_STEP	;RETARDO DE 100ms
	BTFSS	PORTA,4	;SEGURO DE CARGAR VALORES POR DEFECTO
	GOTO	MAIN	;NO: IR A RUTINA PRINCIPAL
	CALL	DELAY_STEP	;RETARDO DE 100ms
	BTFSS	PORTA,4	;SEGURO DE CARGAR VALORES POR DEFECTO
	GOTO	MAIN	;NO: IR A RUTINA PRINCIPAL
	CALL	DELAY_STEP	;RETARDO DE 100ms
	BTFSS	PORTA,4	;SEGURO DE CARGAR VALORES POR DEFECTO
	GOTO	MAIN	;NO: IR A RUTINA PRINCIPAL
	CALL	DELAY_STEP	;RETARDO DE 100ms
	BTFSS	PORTA,4	;SEGURO DE CARGAR VALORES POR DEFECTO
	GOTO	MAIN	;NO: IR A RUTINA PRINCIPAL
	CALL	DELAY_STEP	;RETARDO DE 100ms
	BTFSS	PORTA,4	;SEGURO DE CARGAR VALORES POR DEFECTO
	GOTO	MAIN	;NO: IR A RUTINA PRINCIPAL
	CALL	DELAY_STEP	;RETARDO DE 100ms
	BTFSS	PORTA,4	;SEGURO DE CARGAR VALORES POR DEFECTO
	GOTO	MAIN	;NO: IR A RUTINA PRINCIPAL
	CALL	DELAY_STEP	;RETARDO DE 100ms
	BTFSS	PORTA,4	;SEGURO DE CARGAR VALORES POR DEFECTO
	GOTO	MAIN	;NO: IR A RUTINA PRINCIPAL
	CALL	DELAY_STEP	;RETARDO DE 100ms
	BTFSS	PORTA,4	;SEGURO DE CARGAR VALORES POR DEFECTO
	GOTO	MAIN	;NO: IR A RUTINA PRINCIPAL
	CALL	DELAY_STEP	;RETARDO DE 100ms
	BTFSS	PORTA,4	;SEGURO DE CARGAR VALORES POR DEFECTO
	GOTO	MAIN	;NO: IR A RUTINA PRINCIPAL
	CALL	DELAY_STEP	;RETARDO DE 100ms
	BTFSS	PORTA,4	;SEGURO DE CARGAR VALORES POR DEFECTO
	GOTO	MAIN	;NO: IR A RUTINA PRINCIPAL
	CALL	DELAY_STEP	;RETARDO DE 100ms
	BTFSS	PORTA,4	;SEGURO DE CARGAR VALORES POR DEFECTO
	GOTO	MAIN	;NO: IR A RUTINA PRINCIPAL
	CALL	DELAY_STEP	;RETARDO DE 100ms
	BTFSS	PORTA,4	;SEGURO DE CARGAR VALORES POR DEFECTO
	GOTO	MAIN	;NO: IR A RUTINA PRINCIPAL
	CALL	DELAY_STEP	;RETARDO DE 100ms
	BTFSS	PORTA,4	;SEGURO DE CARGAR VALORES POR DEFECTO
	GOTO	MAIN	;NO: IR A RUTINA PRINCIPAL
	CALL	DELAY_STEP	;RETARDO DE 100ms
	BTFSS	PORTA,4	;SEGURO DE CARGAR VALORES POR DEFECTO
	GOTO	MAIN	;NO: IR A RUTINA PRINCIPAL
	CALL	DELAY_STEP	;RETARDO DE 100ms
	BTFSS	PORTA,4	;SEGURO DE CARGAR VALORES POR DEFECTO
	GOTO	MAIN	;NO: IR A RUTINA PRINCIPAL
	CALL	DELAY_STEP	;RETARDO DE 100ms
	BTFSS	PORTA,4	;SEGURO DE CARGAR VALORES POR DEFECTO
	GOTO	MAIN	;NO: IR A RUTINA PRINCIPAL
	CALL	DELAY_STEP	;RETARDO DE 100ms
	BTFSS	PORTA,4	;SEGURO DE CARGAR VALORES POR DEFECTO
	GOTO	MAIN	;NO: IR A RUTINA PRINCIPAL
	CALL	DELAY_STEP	;RETARDO DE 100ms
	BTFSS	PORTA,4	;SEGURO DE C

```

        GOTO    MAIN                ;NO: IR A RUTINA PRINCIPAL
        CALL    DELAY_STEP          ;RETARDO DE 100ms
        BTFSS   PORTA,4             ;SEGURO DE CARGAR VALORES POR DEFECTO
        GOTO    MAIN                ;NO: IR A RUTINA PRINCIPAL
        CALL    DFLT_ADD            ;SI: CARGA DIRECC POR DEFECTO
        CALL    DFLT_BPS            ;CARGA BPS POR DEFECTO
        CALL    DFLT_RPTA          ;CARGA RETARDO POR DEFECTO
        CALL    DFLT_RST            ;CARGA VARIABLE DE RESET
        GOTO    RESET               ;RESET AL PIC

;*****
;ACTUALIZA MOTOR PA *
;*****

UPDATE_PA  MOVLW   H'01'
           XORWF   PORTA,F
           CLR_CARRY
           MOVF    POSPA_RX,W       ;LEE POS DE MOTOR PA
           SUBWF   POSPA,W          ;COMPARA POSPA_RX CON POSPA
           BTFSC   STATUS,C         ;POSPA_RX > POSPA??
           GOTO    PA_BACK          ;NO: IR A PASO HACIA ATRAZ
           GOTO    PA_FWD           ;SI: IR A PASO HACIA ADELANTE
PA_FWD     INCF    STEP_PA,F        ;INCREMENTA PASO
           MOVF    STEP_PA,W        ;LEE PASO
           XORLW   H'09'            ;COMPARA CON VALOR 09
           BTFSC   STATUS,Z         ;STEP = 09??
           GOTO    RST_PAFWD        ;SI: RST PASO
           MOVF    STEP_PA,W        ;NO: LEE PASO
           CALL    PA_TABLE         ;LEE TABLA CON PASOS
           MOVWF   PA_TABLE_TMP     ;GUARDA PASO
           MOVF    PORTB,W          ;LEE PUERTO B
           ANDLW   H'F0'            ;MASK DATOS DEL MOTOR DE DRIVER
           ADDWF   PA_TABLE_TMP,W   ;SUMA PASOS A PUERTB MASK
           MOVWF   PORTB            ;VALOR AL PORTB
           INCF    POSPA,F          ;INCREMENTA POSICION
           CALL    DELAY_STEP       ;RETARDO DE 100ms
           GOTO    OUT_PA           ;SALE DE RUTINA
RST_PAFWD  CLRF    STEP_PA          ;VAR = 0
           GOTO    PA_FWD           ;SIGUIENTE PASO

PA_BACK    DECFSZ  STEP_PA,F        ;DECREMENTA PASO SKIP SI ES CERO
           GOTO    PA_BACK2         ;NO: SIGUE
           GOTO    RST_BACK         ;SI: RST PASO
PA_BACK2   MOVF    STEP_PA,W        ;LEE PASO
           CALL    PA_TABLE         ;LEE TABLA CON PASOS
           MOVWF   PA_TABLE_TMP     ;GUARDA PASO
           MOVF    PORTB,W          ;LEE PORTB
           ANDLW   H'F0'            ;MASK PULSOS DEL MOTOR DE DRIVER
           ADDWF   PA_TABLE_TMP,W   ;SUMA PASOS A PORTB MASK
           MOVWF   PORTB            ;VALOR AL PORTB
           DECF    POSPA,F          ;DECREMENTA POSICION
           CALL    DELAY_STEP       ;RETARDO DE 100ms
           GOTO    OUT_PA           ;SALIR DE RUTINA
RST_BACK   MOVLW   H'09'            ;LEE VALOR 09
           MOVWF   STEP_PA          ;GUARDA NUEVO PASO
           GOTO    PA_BACK          ;SIGUIENTE PASO

OUT_PA     MOVF    POSPA_RX,W       ;LEE POSICION RECIBIDA
           XORWF   POSPA,W          ;COMPARA CON POSICION ACTUAL
           BTFSS   STATUS,Z         ;IGUALES??
           RETURN                  ;NO: REGRESA A RUTINA PRINCIPAL
           MOVF    PORTB,W          ;SI: LEE PORTB
           ANDLW   H'F0'            ;MASK PORTB
           MOVWF   PORTB            ;VALOR AL PORTB
           BSF     PORTA,0
           RETURN                    ;REGRESA RUTINA PRINCIPAL

;*****
;ACTUALIZA MOTOR DRV *
;*****

UPDATE_DRV MOVLW   H'01'
           XORWF   PORTA,F
           CLR_CARRY
           MOVF    POSDRV_RX,W      ;LEE POSICION RECIBIDA DE DRV
           SUBWF   POSDRV,W         ;COMPARA CON POSICION DE DRV ACTUAL
           BTFSC   STATUS,C         ;POSDRV_RX > POSDRV??

```

```

DRV_FWD      GOTO    DRV_BACK      ;NO: PASO HACIA ATRAZ
              GOTO    DRV_FWD      ;SI: PASO HACIA ADELANTE
              INCF     STEP_DRV,F    ;INCREMENTA PASO
              MOVF     STEP_DRV,W    ;LEE PASO
              XORLW    H'09'         ;COMPARA CON VALOR 09
              BTFSC    STATUS,Z      ;STEP = 09??
              GOTO    RST_DRVFWDD   ;SI: RST PASO
              MOVF     STEP_DRV,W    ;NO: LEE PASO
              CALL     DRV_TABLE     ;LEE TABLA CON PULSOS
              MOVWF    DRV_TABLE_TMP ;GUARDA PULSOS
              MOVF     PORTB,W       ;LEE PORTB
              ANDLW    H'0F'         ;MASK DATOS DEL MOTOR DE PA
              ADDWF    DRV_TABLE_TMP,W ;SUMA PULSOS CON PORTB MASK
              MOVWF    PORTB         ;VALOR AL PORTB
              INCF     POSDRV,F      ;INCREMENTA POSICION
              CALL     DELAY_STEP    ;RETARDO DE 100ms
              GOTO    OUT_DRV        ;SALE DE RUTINA
RST_DRVFWDD   CLRF     STEP_DRV      ;RST PASO
              GOTO    DRV_FWD        ;SIGUIENTE PASO

DRV_BACK      DECFSZ   STEP_DRV,F    ;DECREMENTA PASO SKIP SI ES CERO
              GOTO    DRV_BACK2      ;NO: SIGUE
              GOTO    RST_DRVBACK    ;SI: RESET PASO
DRV_BACK2     MOVF     STEP_DRV,W    ;LEE PASO
              CALL     DRV_TABLE     ;LEE TABLA CON PULSOS
              MOVWF    DRV_TABLE_TMP ;GUARDA PULSOS
              MOVF     PORTB,W       ;LEE PORTB
              ANDLW    H'0F'         ;MASK PULSOS DEL MOTOR DE PA
              ADDWF    DRV_TABLE_TMP,W ;SUMO PULSOS DEL MOTOR DRV
              MOVWF    PORTB         ;VALOR AL PORTB
              DECF     POSDRV,F      ;DECREMENTEO POSICION
              CALL     DELAY_STEP    ;RETARDO DE 100ms
              GOTO    OUT_DRV        ;SALE DE RUTINA
RST_DRVBACK   MOVLW    H'09'         ;LEO VALOR 09
              MOVWF    STEP_DRV      ;STEP=09
              GOTO    DRV_BACK        ;SIGUIENTE PASO

OUT_DRV       MOVF     POSDRV_RX,W    ;LEO POS DE DRV RECIBIDA
              XORWF    POSDRV,W      ;COMPARO CON POSICION ACTUAL
              BTFSS    STATUS,Z      ;IGUALES?
              RETURN                ;NO: REGRESA A RUTINA PRINCIPAL
              MOVF     PORTB,W       ;SI: LEO PORTB
              ANDLW    H'0F'         ;MASK PULSOS DE MOTOR PA
              MOVWF    PORTB         ;VALOR AL PORTB
              BSF      PORTA,0       ;
              RETURN                ;REGRESA A RUTINA PRINCIPAL

```

```

;*****
;RETARDO ENTRE PASOS IGUAL A DE 100 ms *
;*****

```

```

DELAY_STEP   MOVLW    H'82'
              MOVWF    CONT_4
              MOVLW    H'FF'
              MOVWF    CONT_3
LOOP_STEP     DECFSZ   CONT_3,F
              GOTO    LOOP_STEP
              DECFSZ   CONT_4,F
              GOTO    LOOP_STEP
              RETURN

```

```

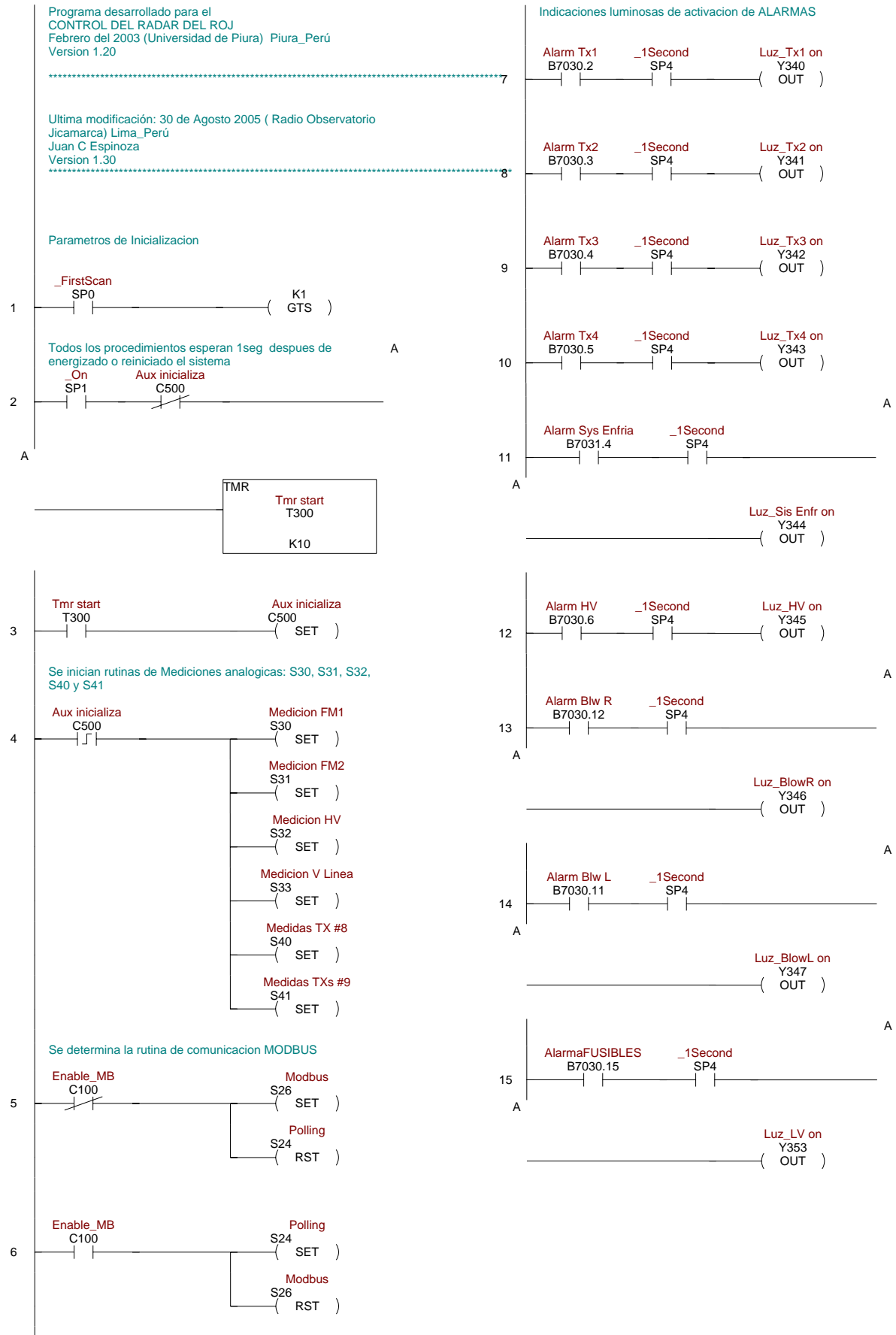
END

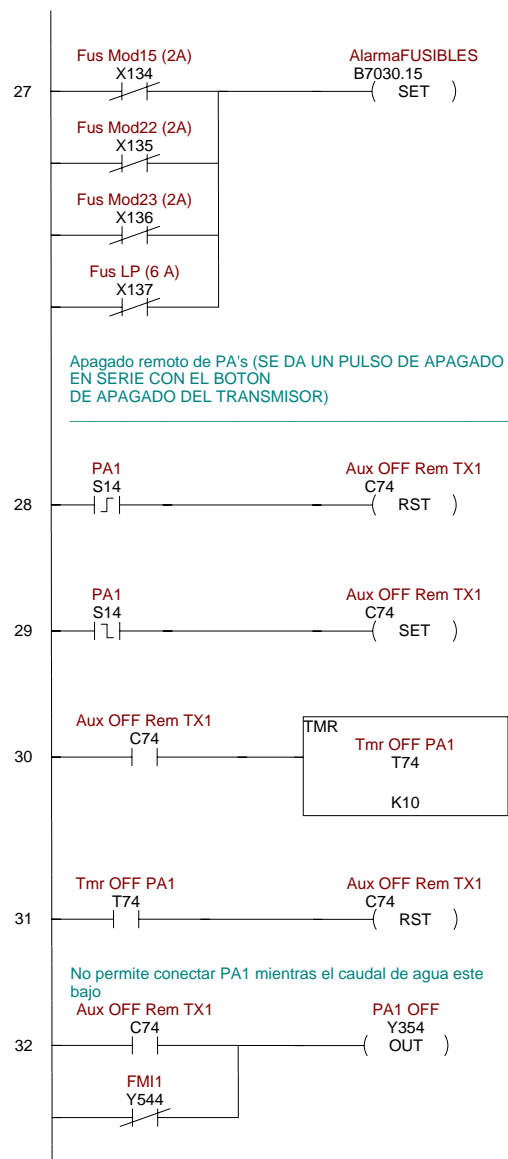
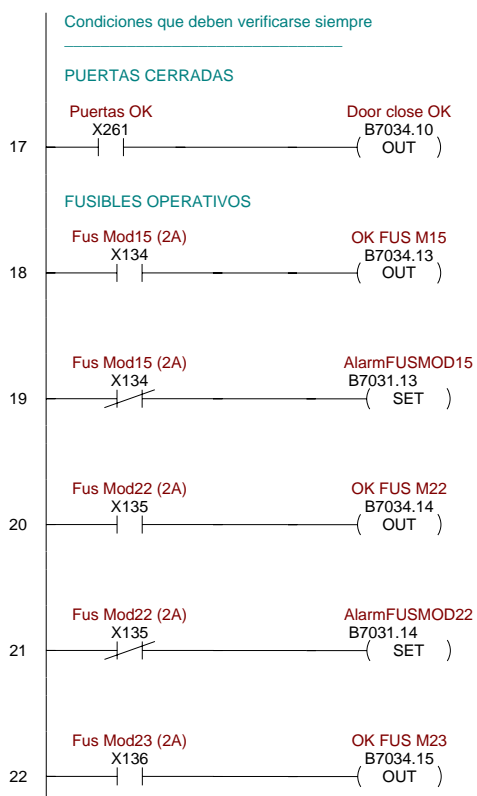
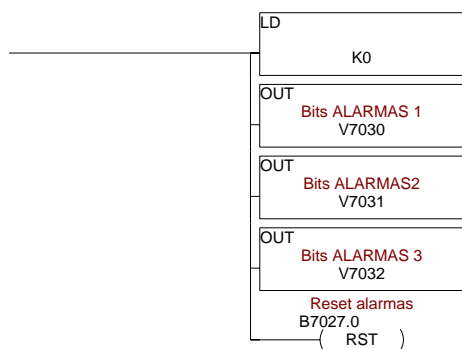
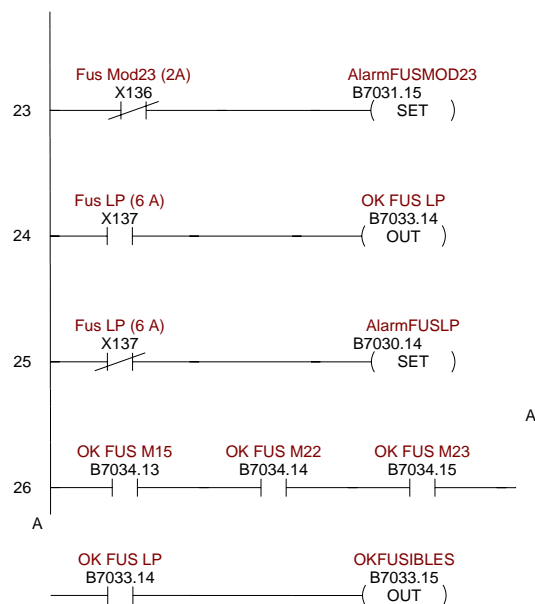
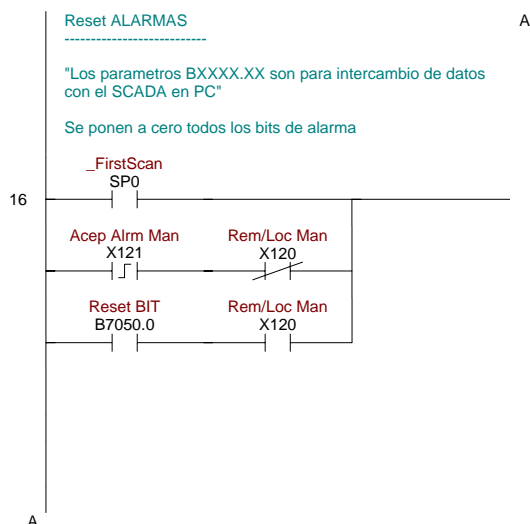
```

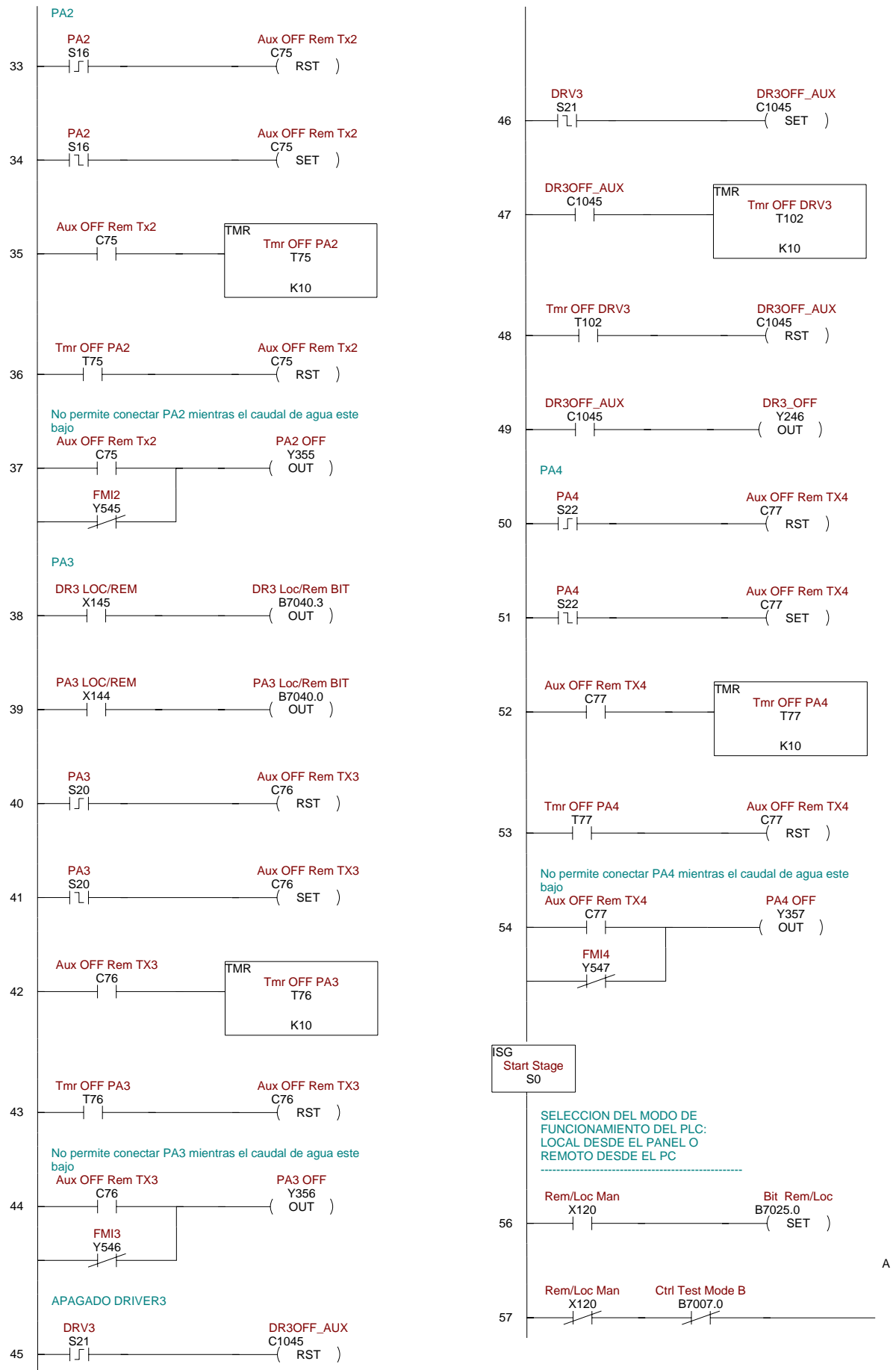
ANEXO D

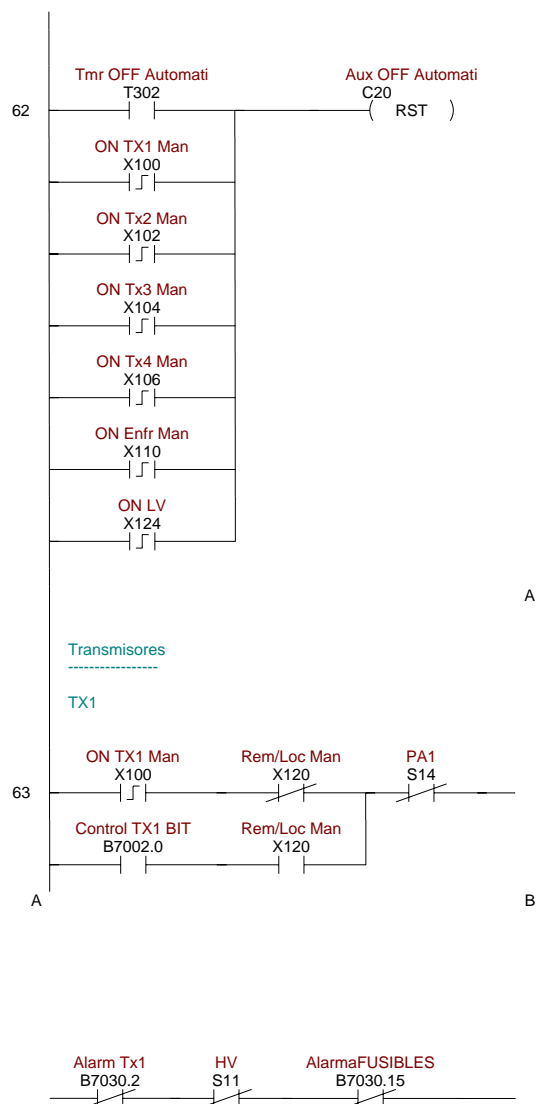
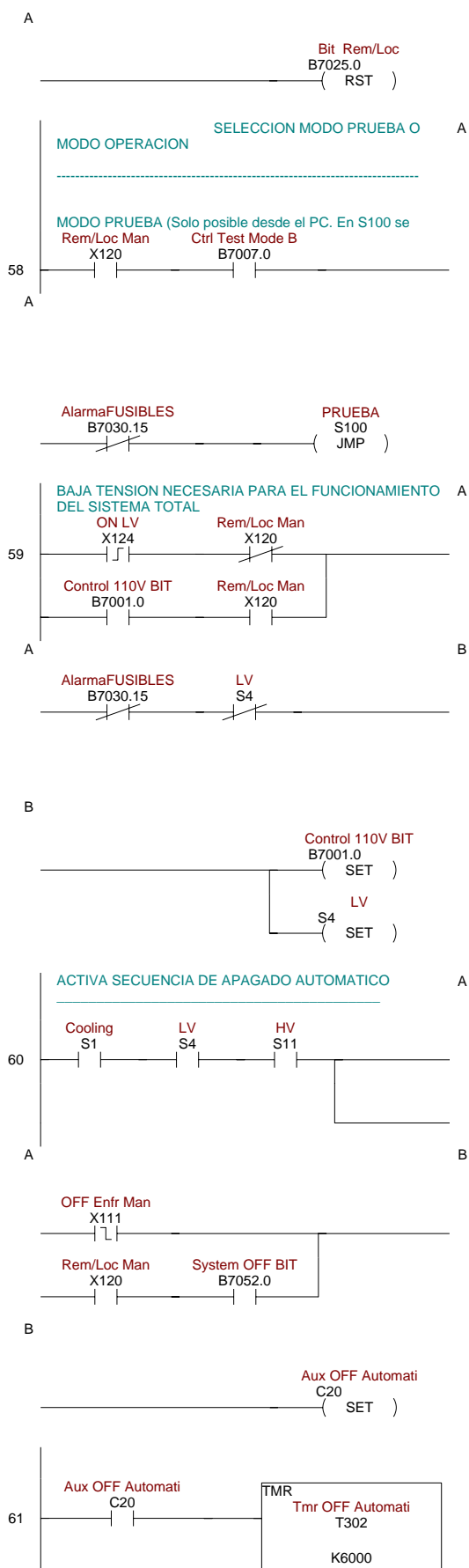
Programa del PLC

En este anexo se muestra el diagrama escalera del programa de aplicación modificado, el cual ha sido escrito con el metodo RLL^{PLUS} (*Stage Programming*) utilizando el software DirectSOFT32 Programming para Windows.

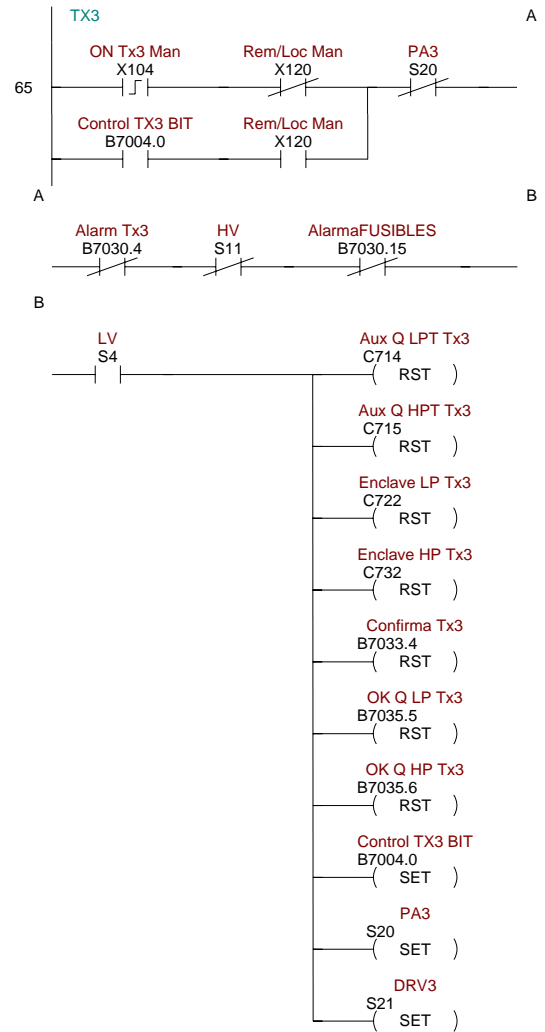
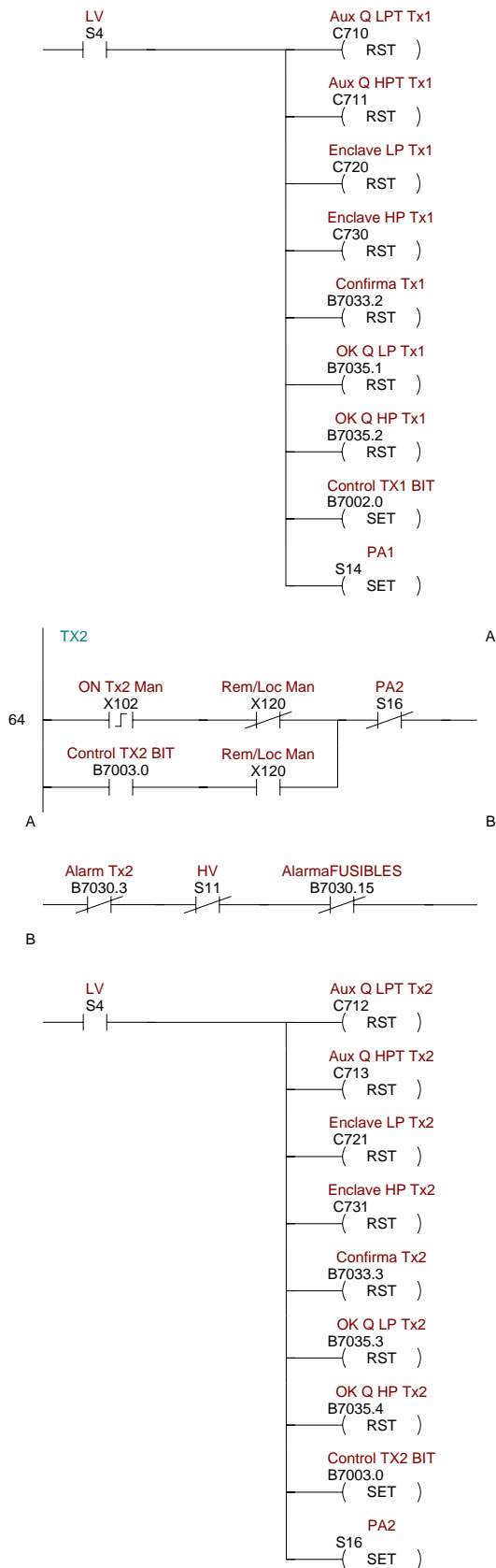


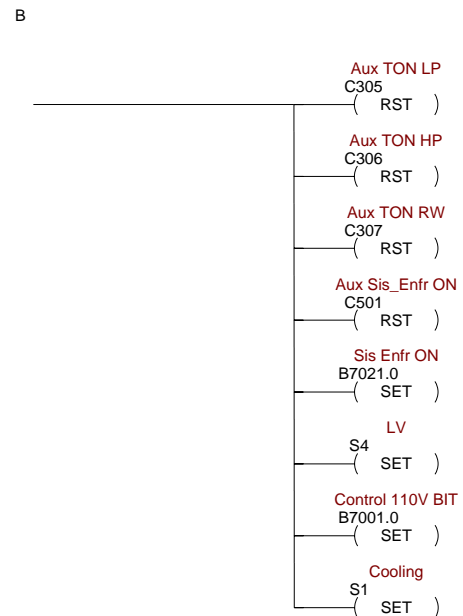
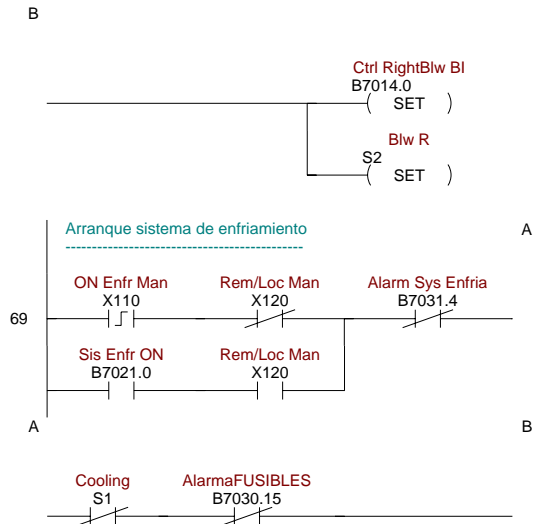
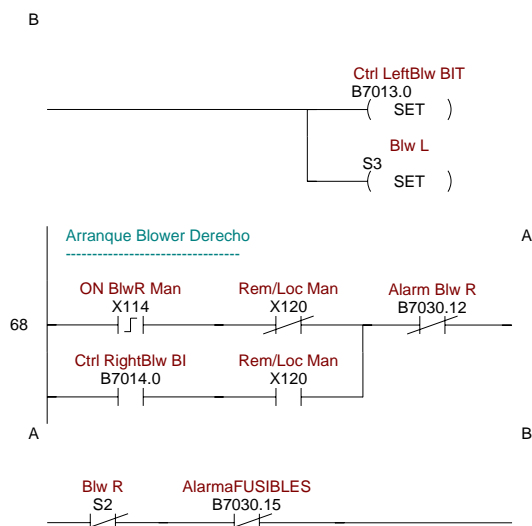
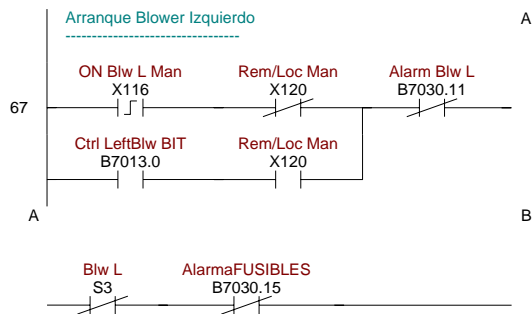
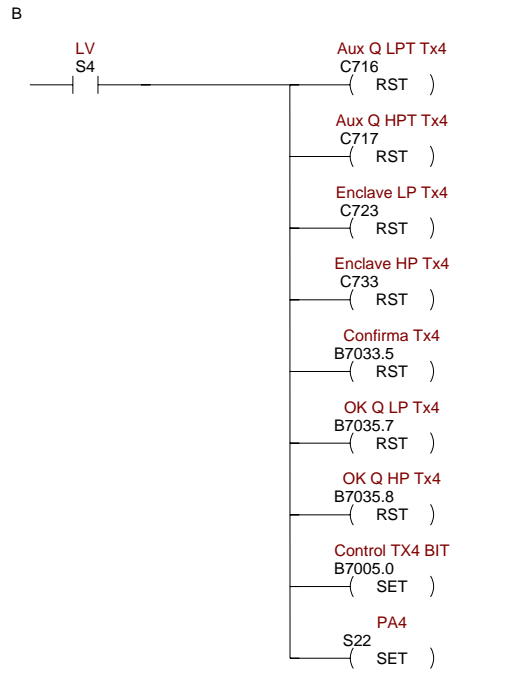


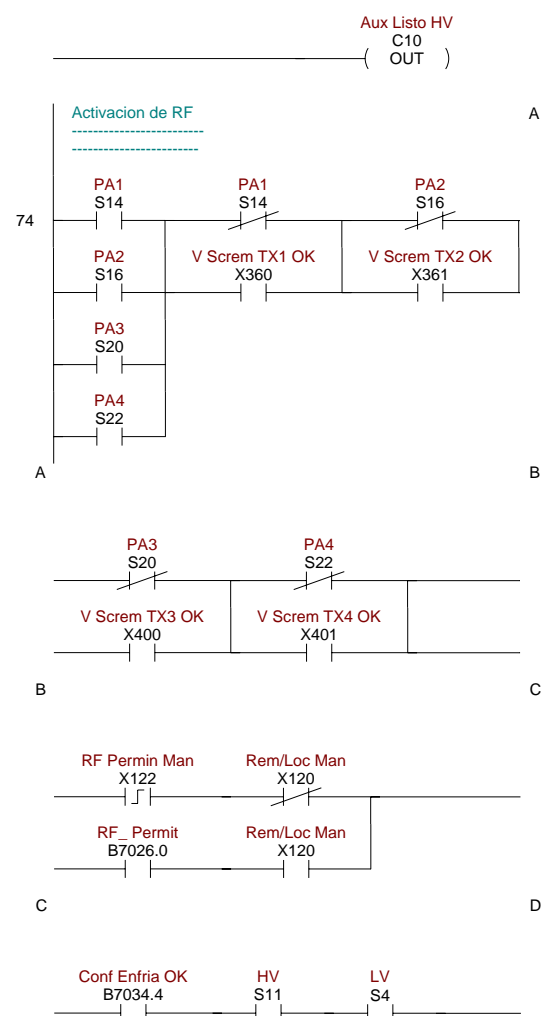
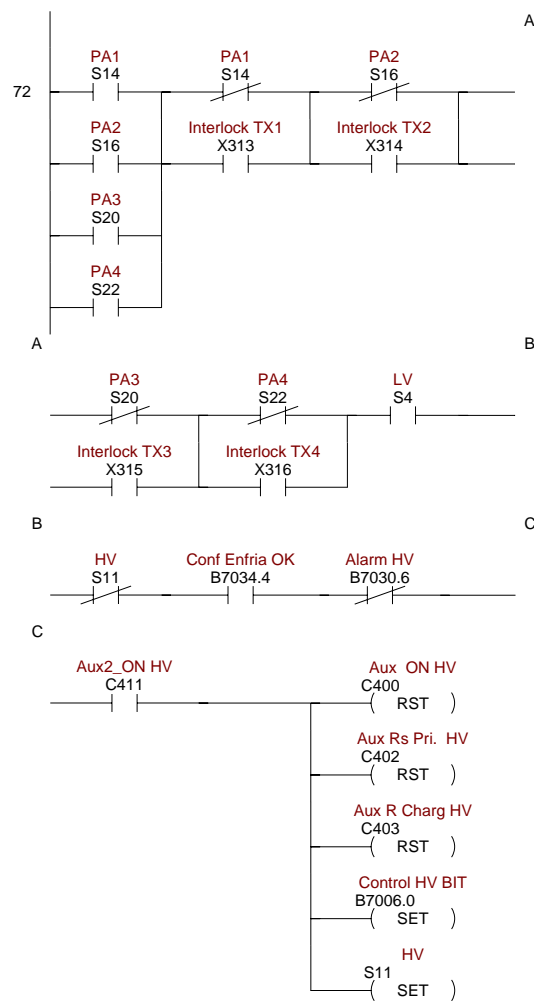
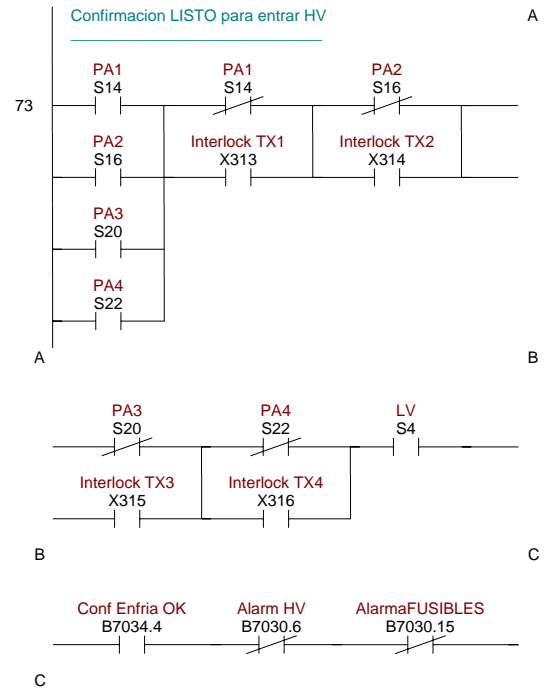
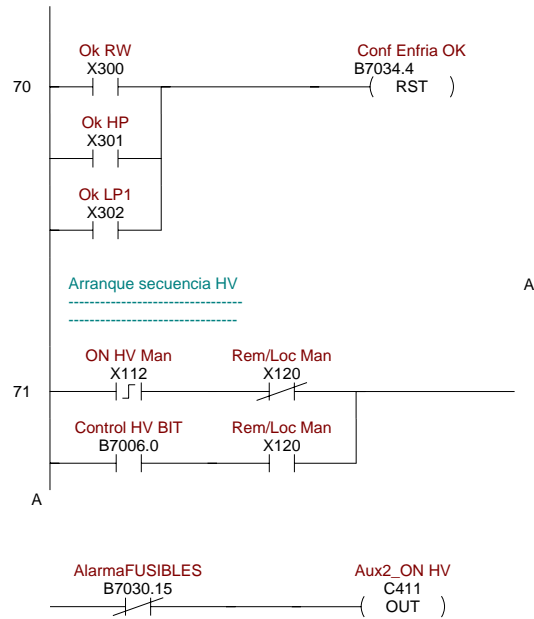


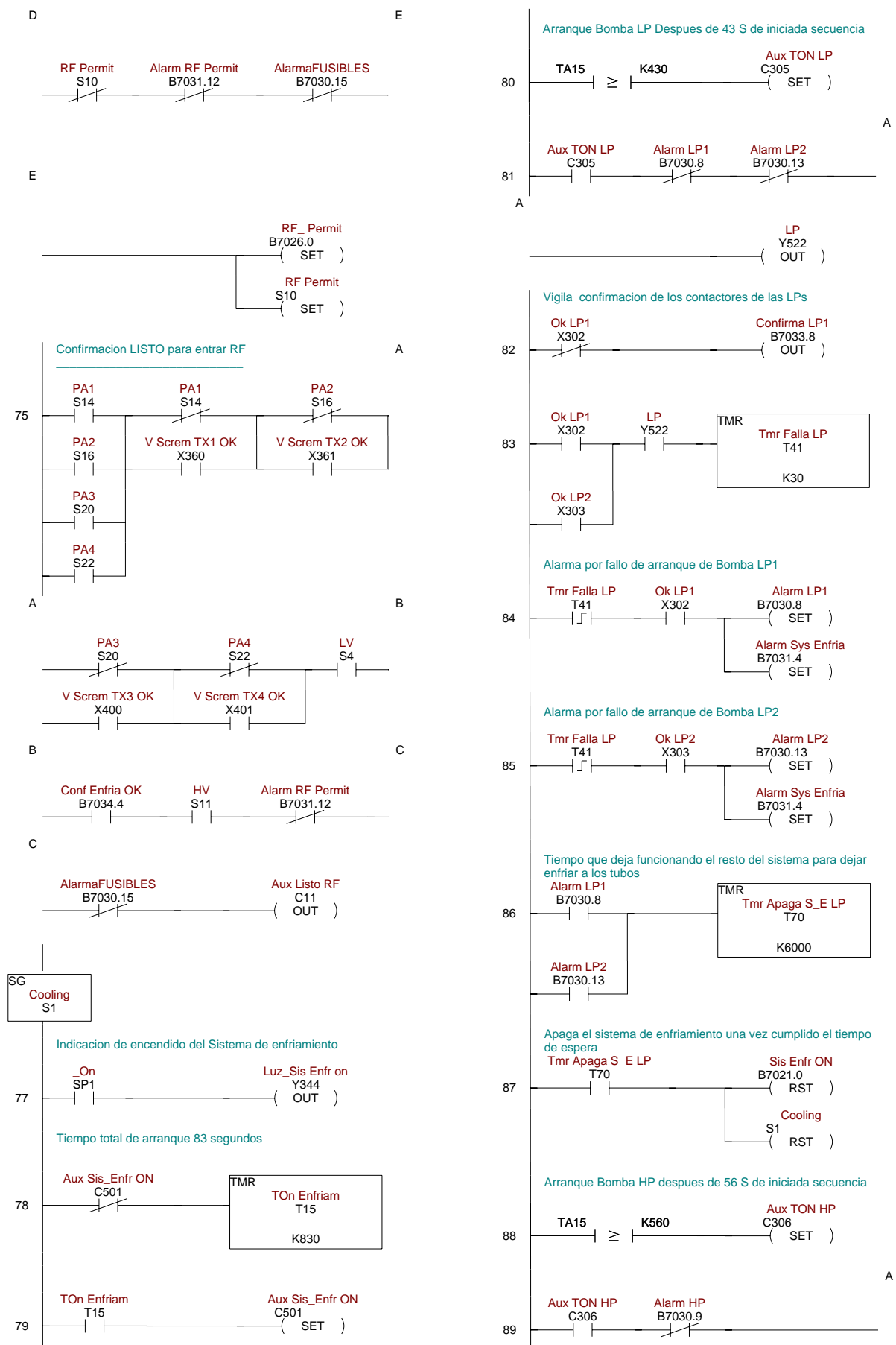


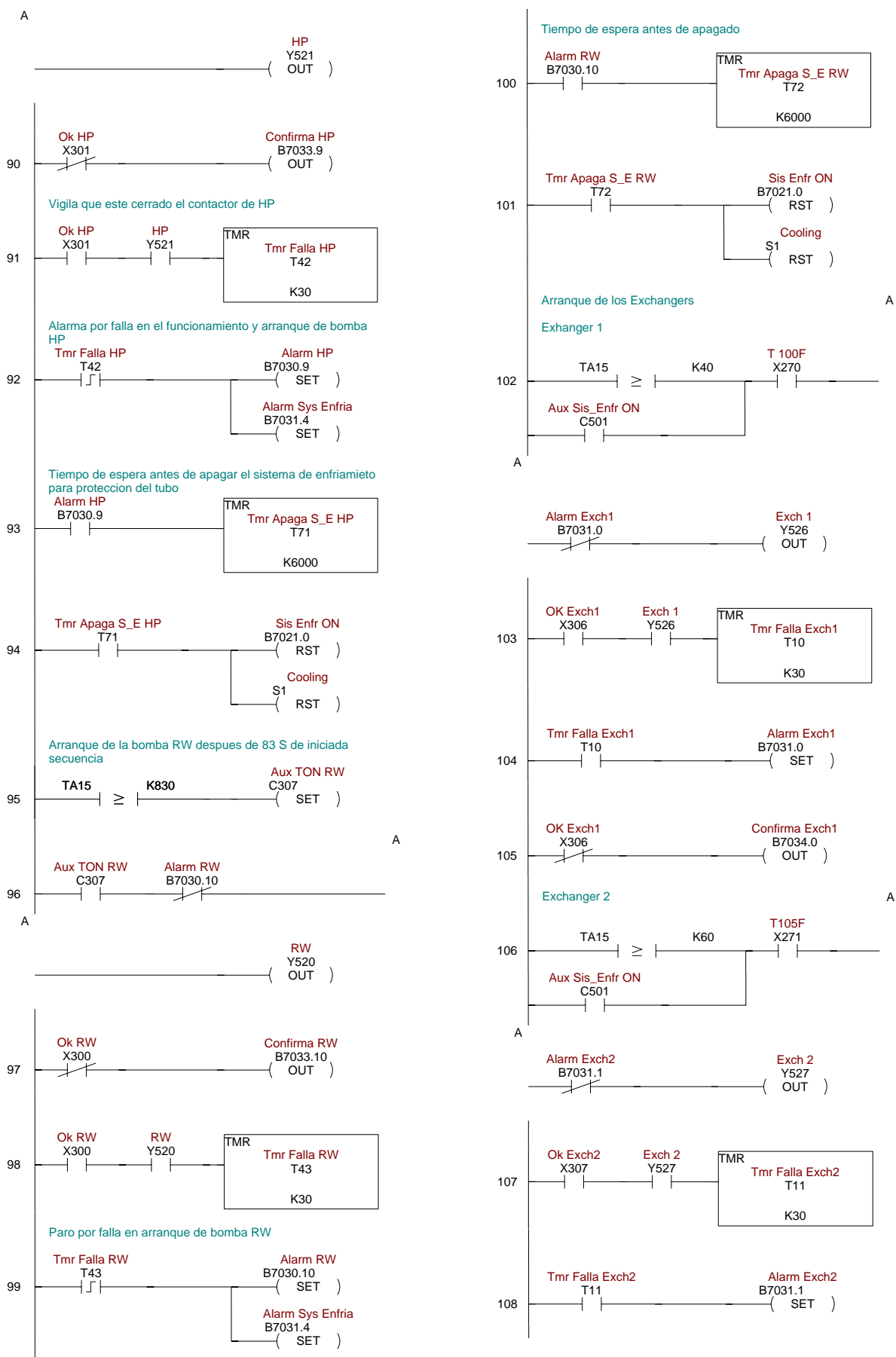
B

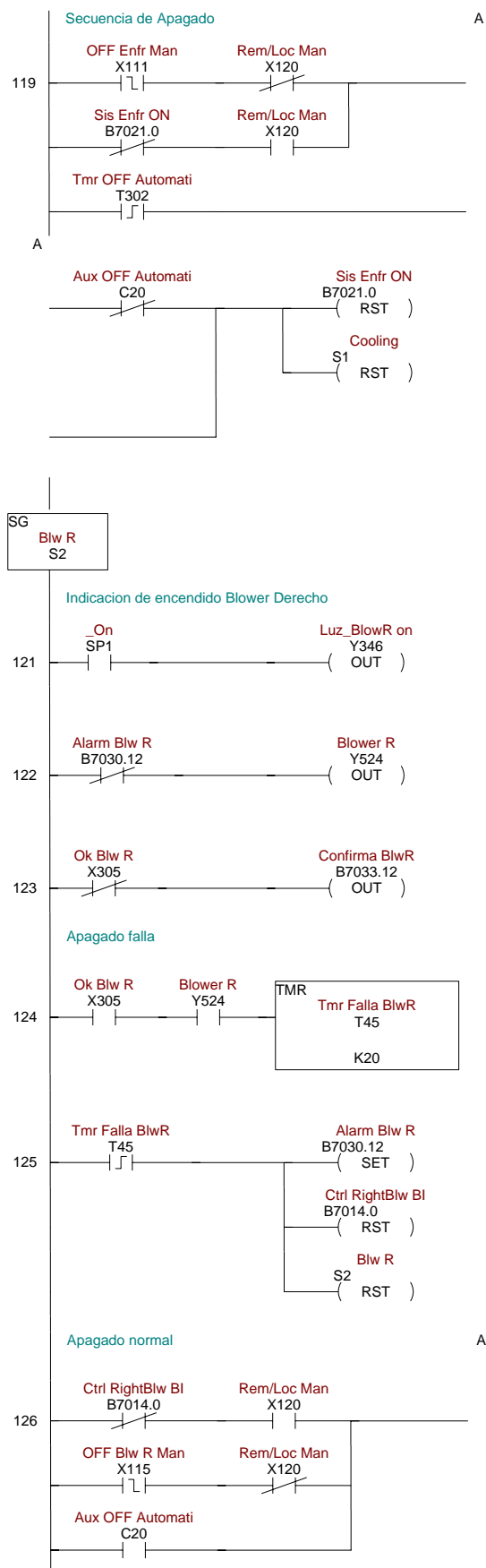
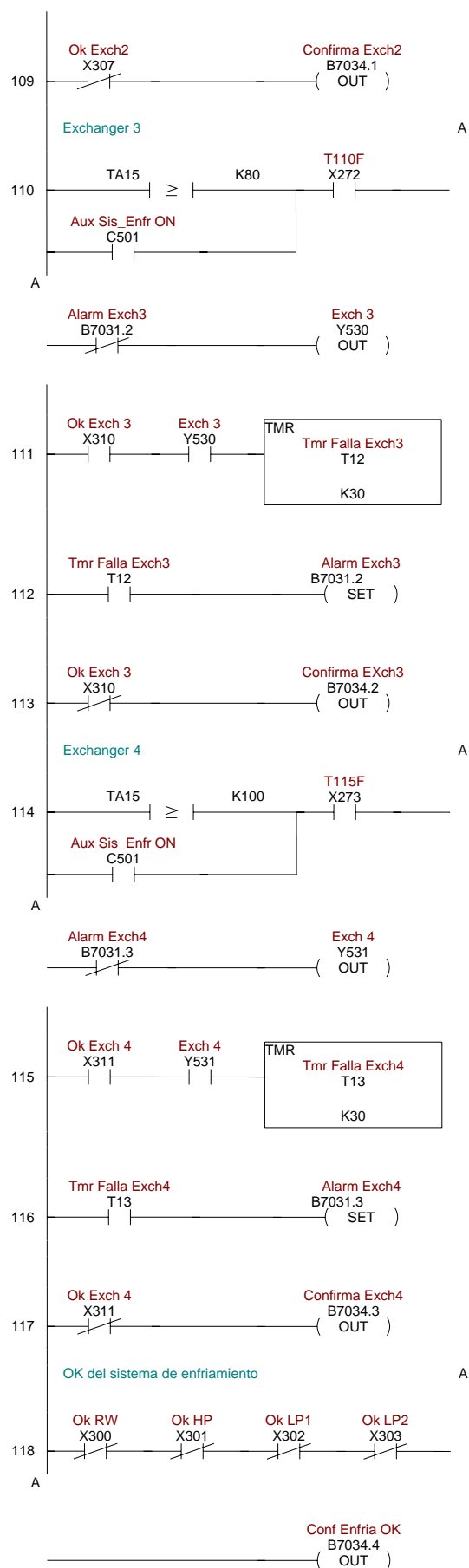


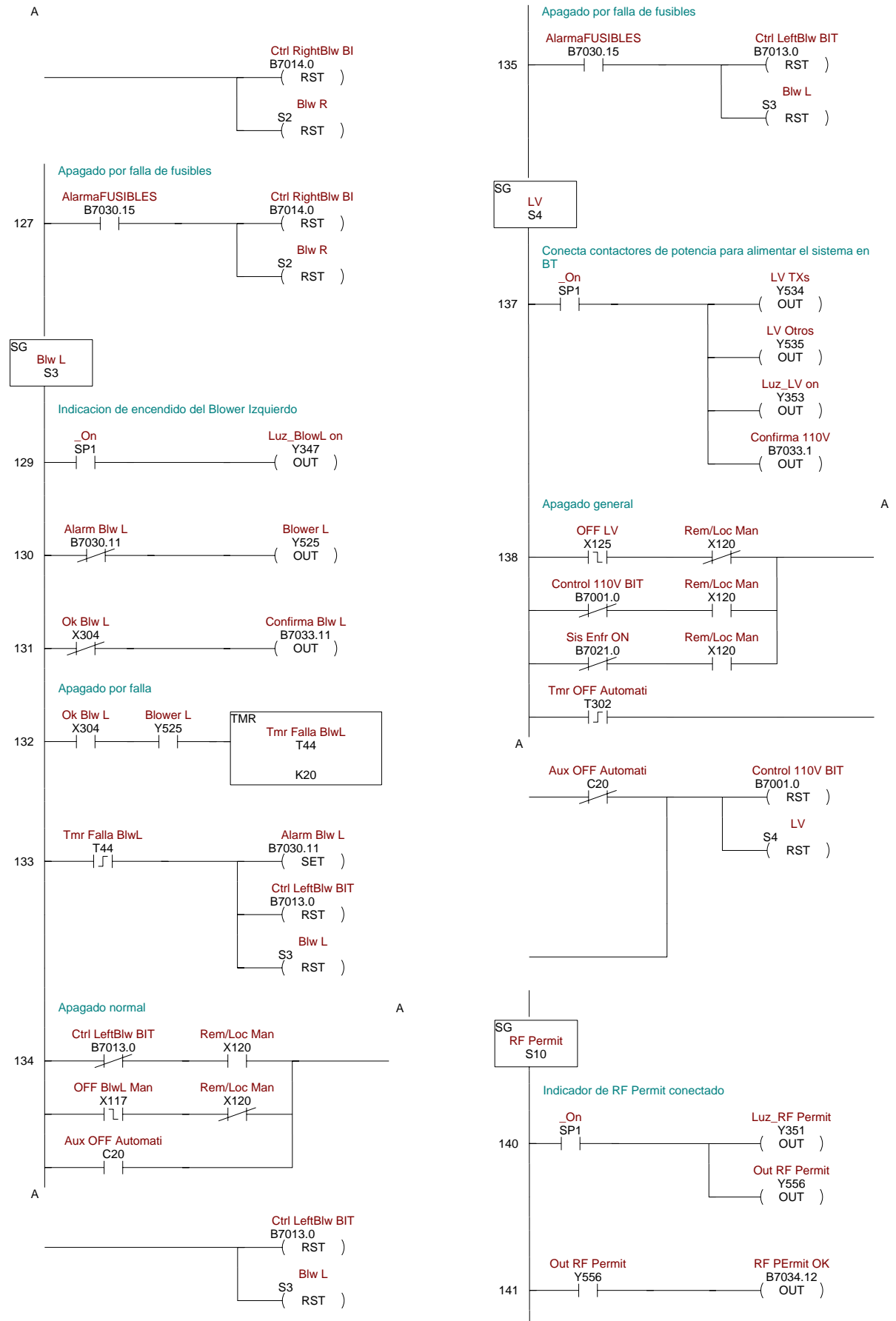


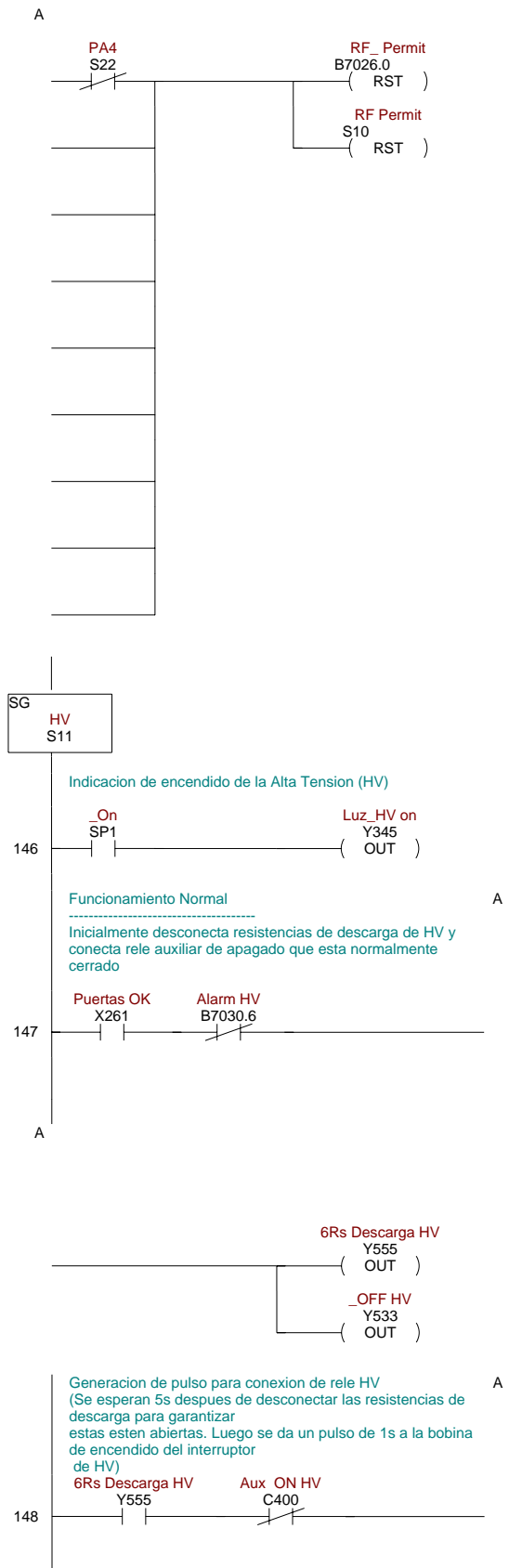
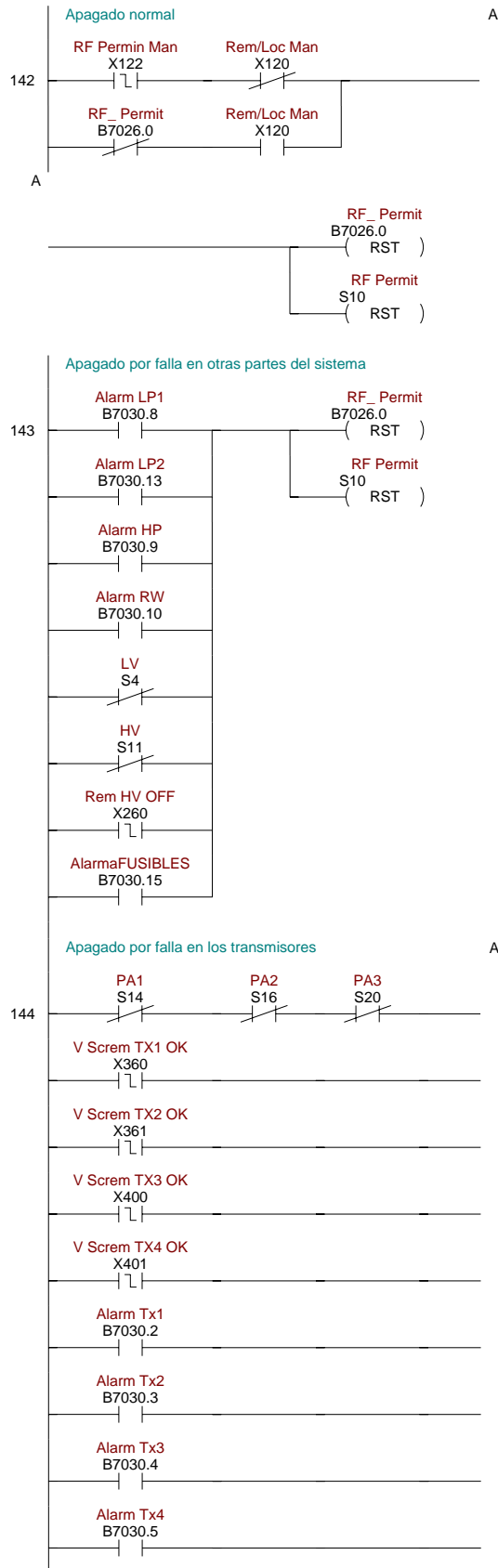




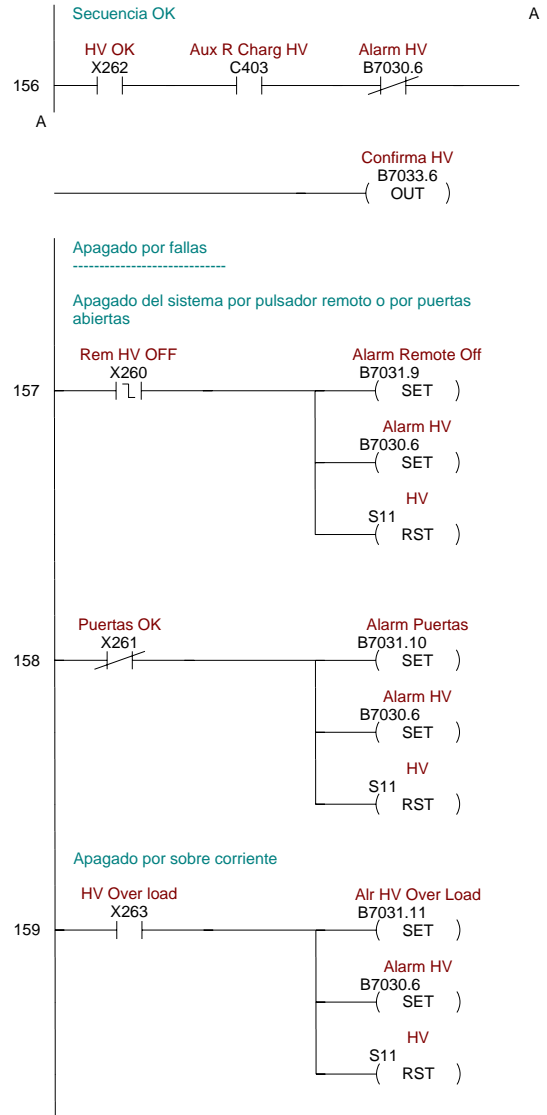
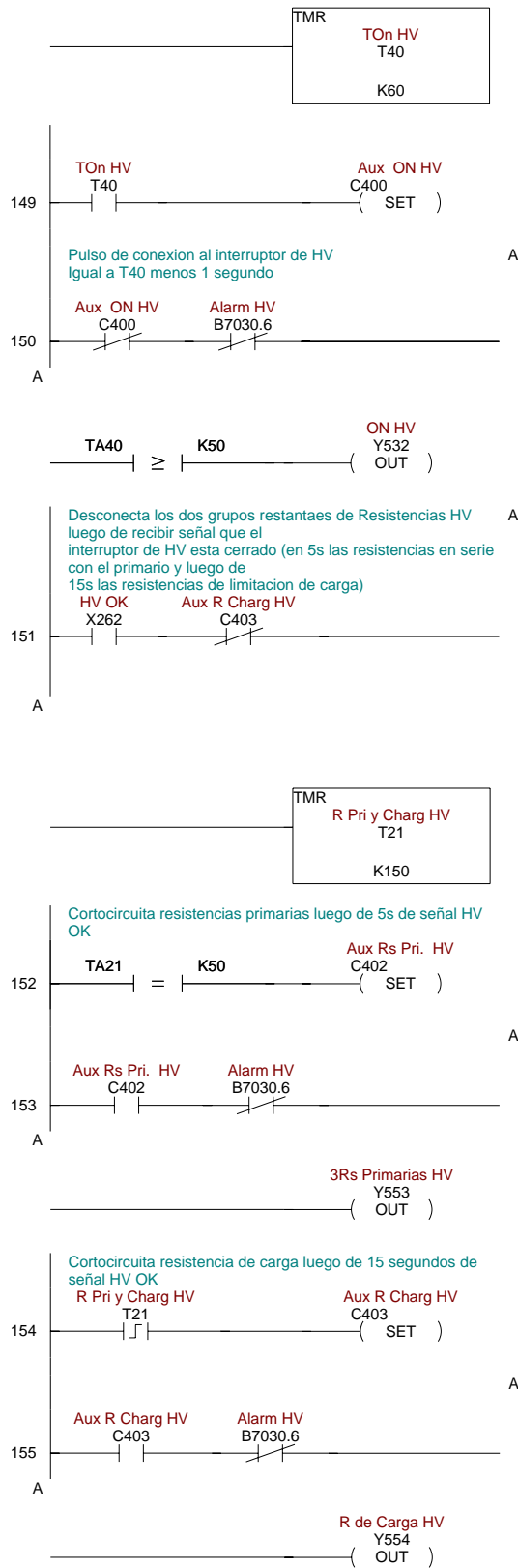


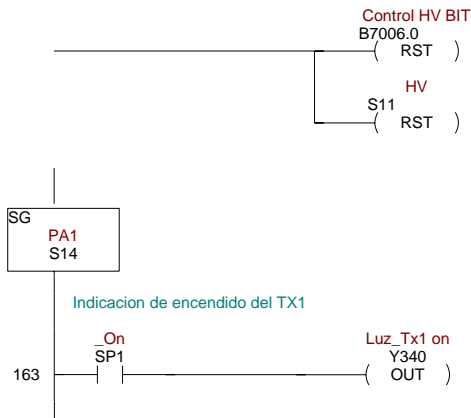
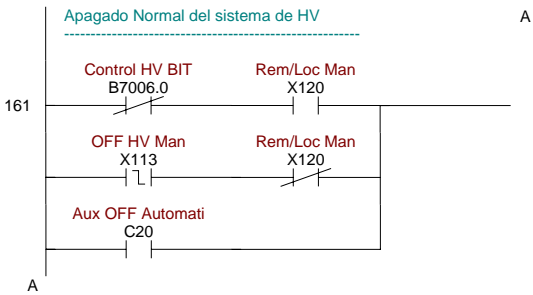
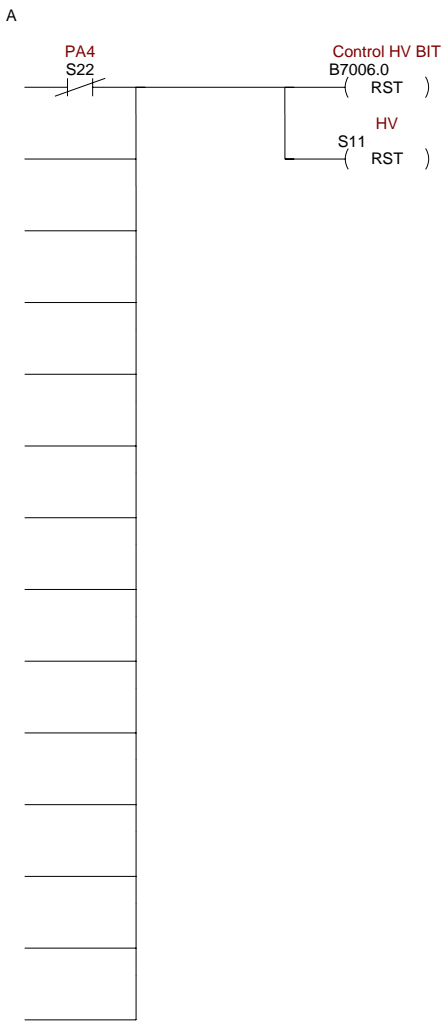
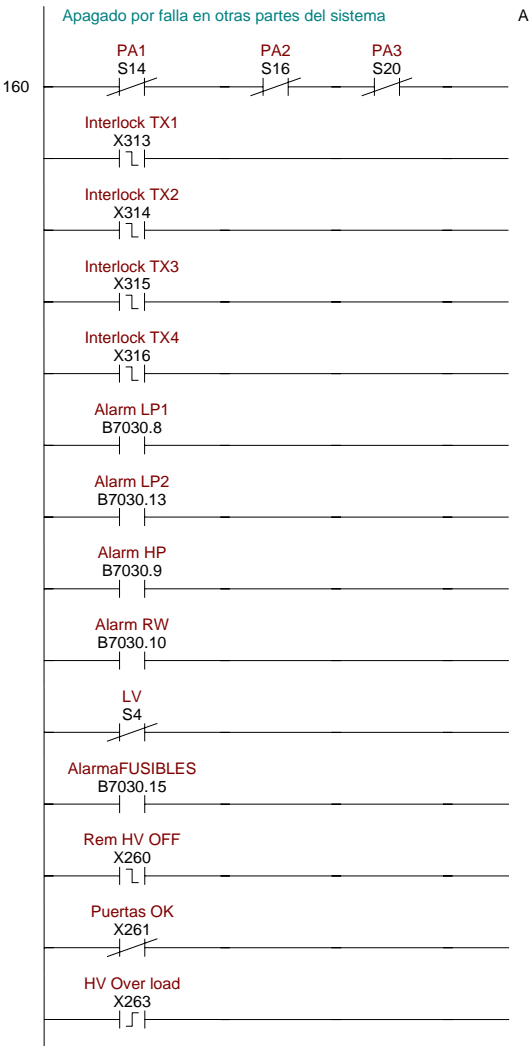


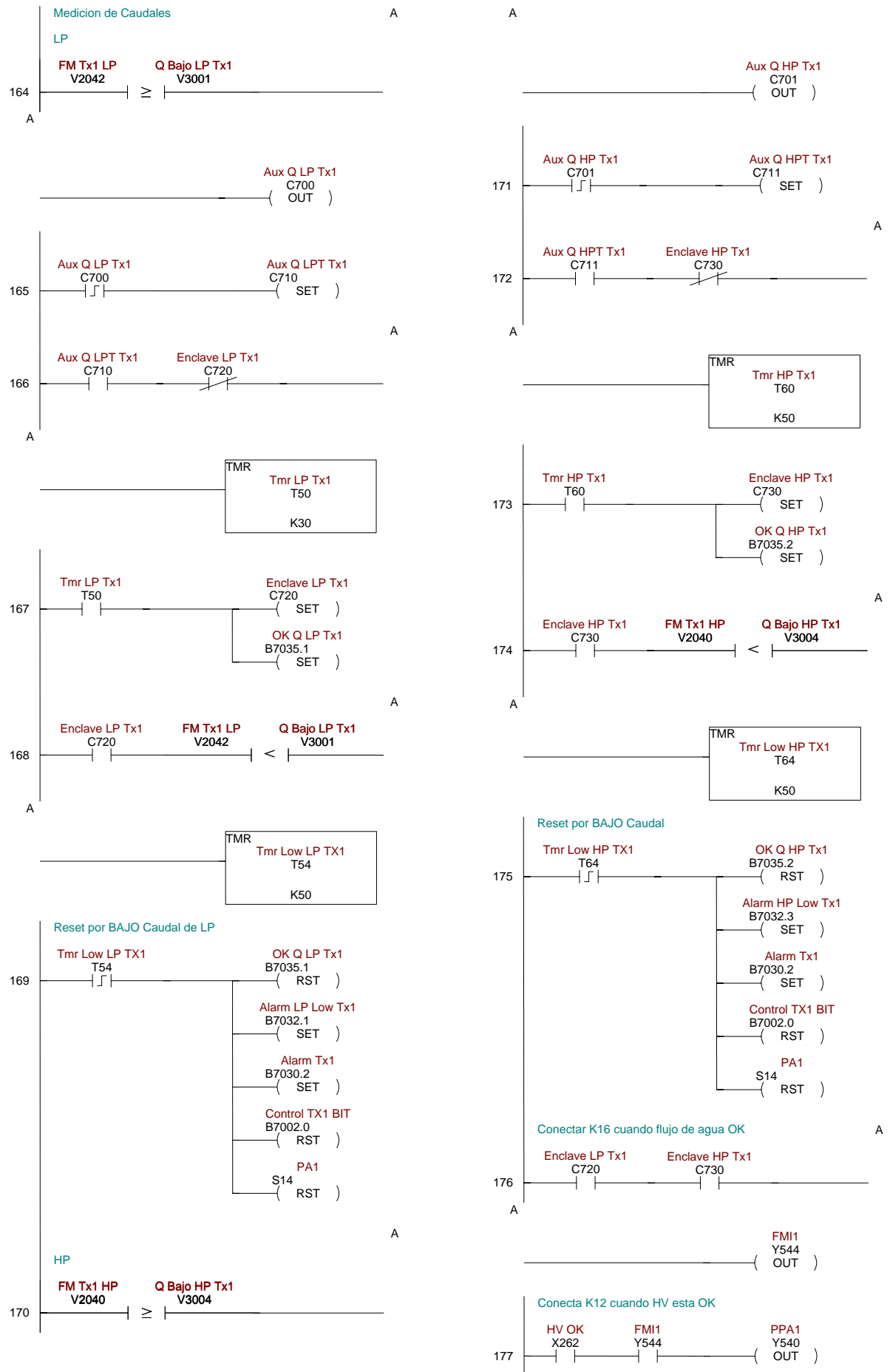


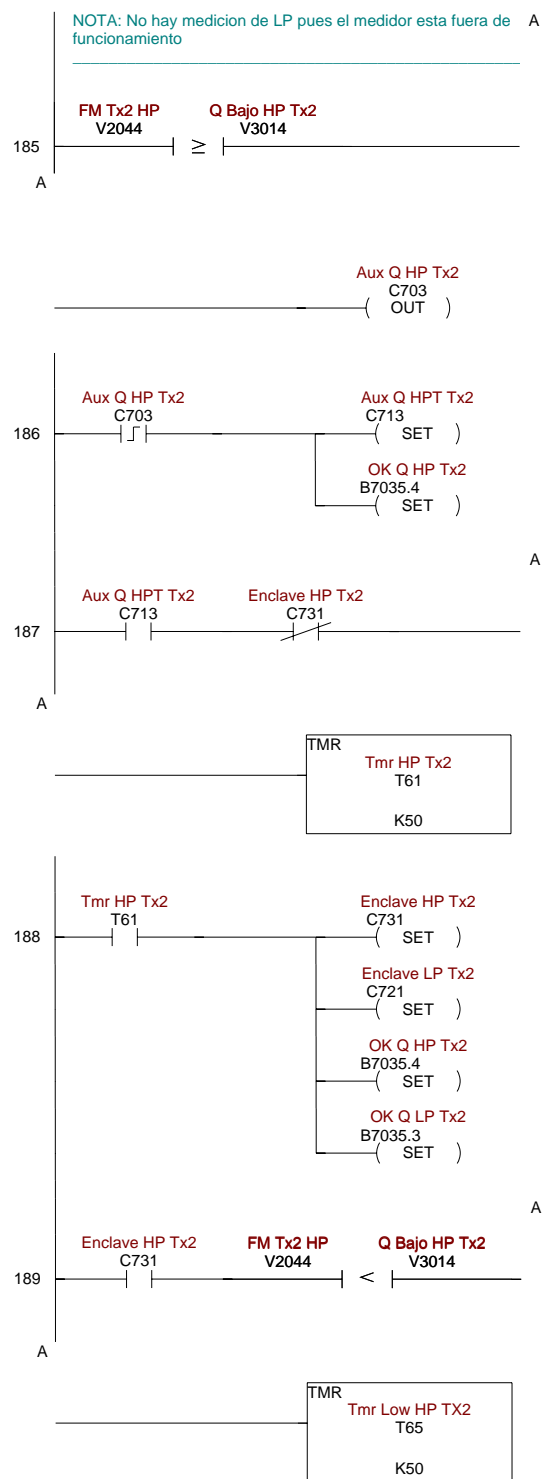
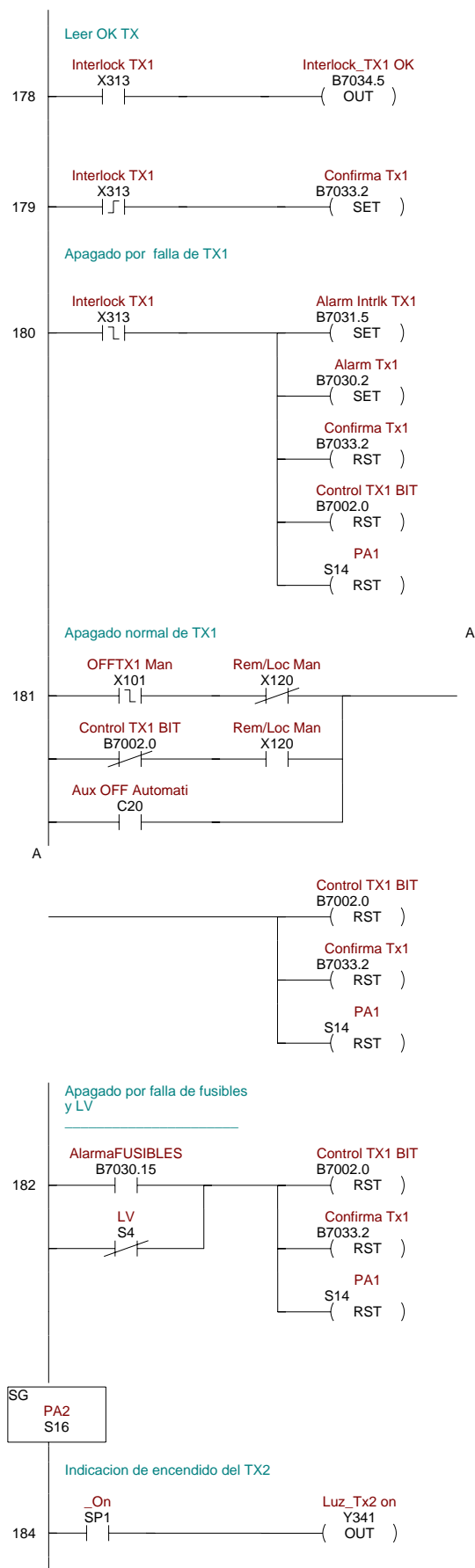


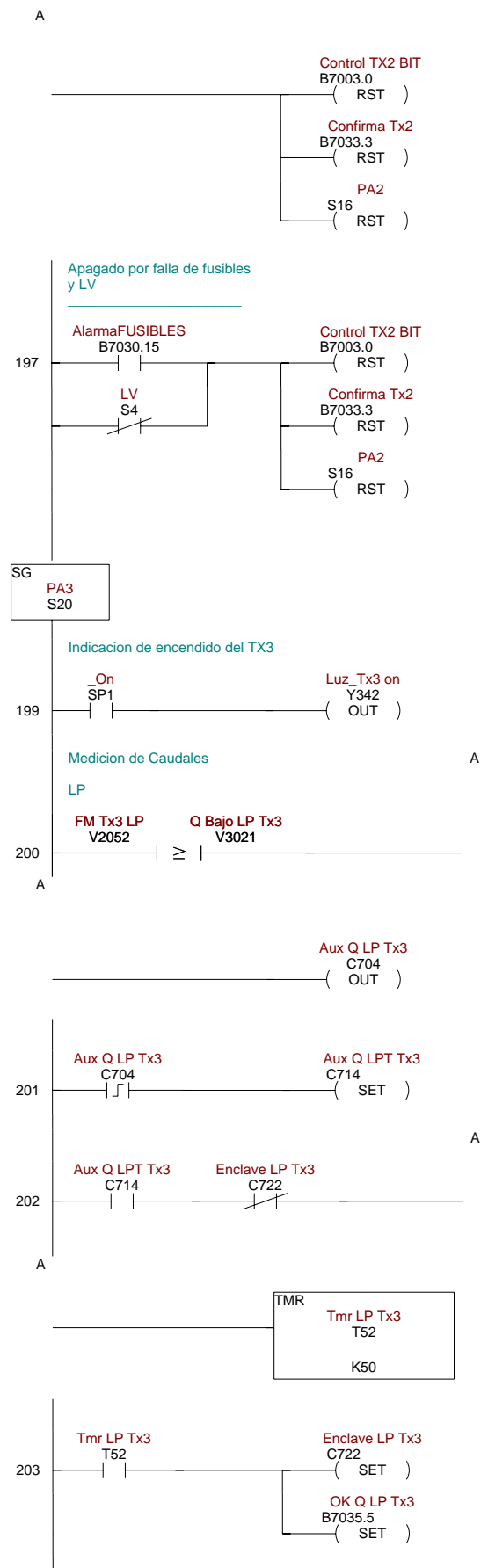
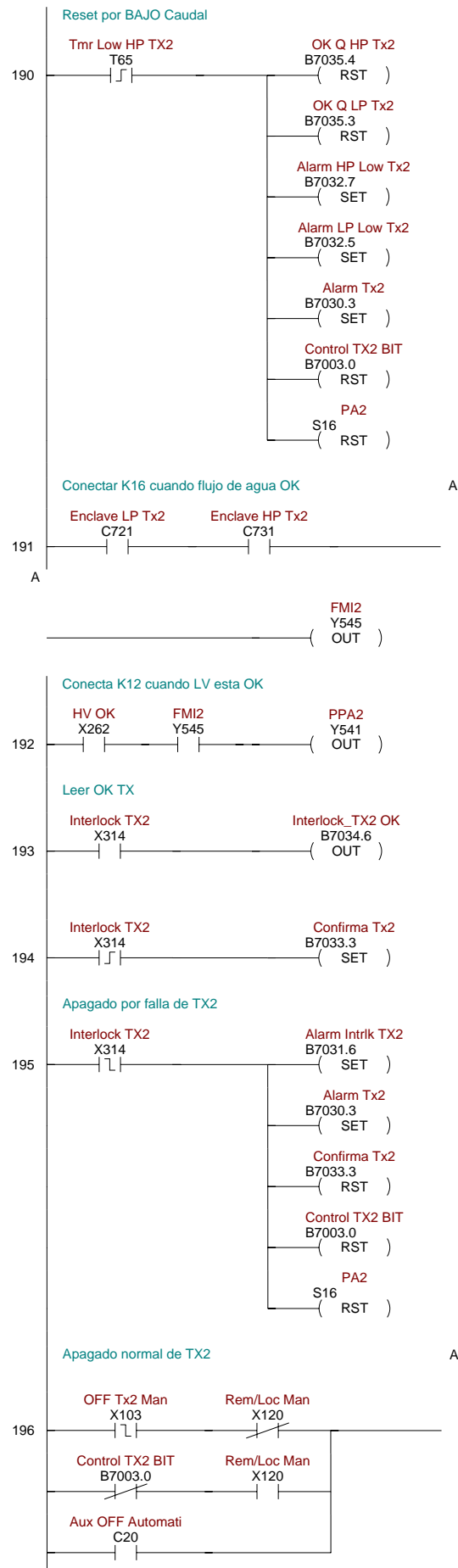
A

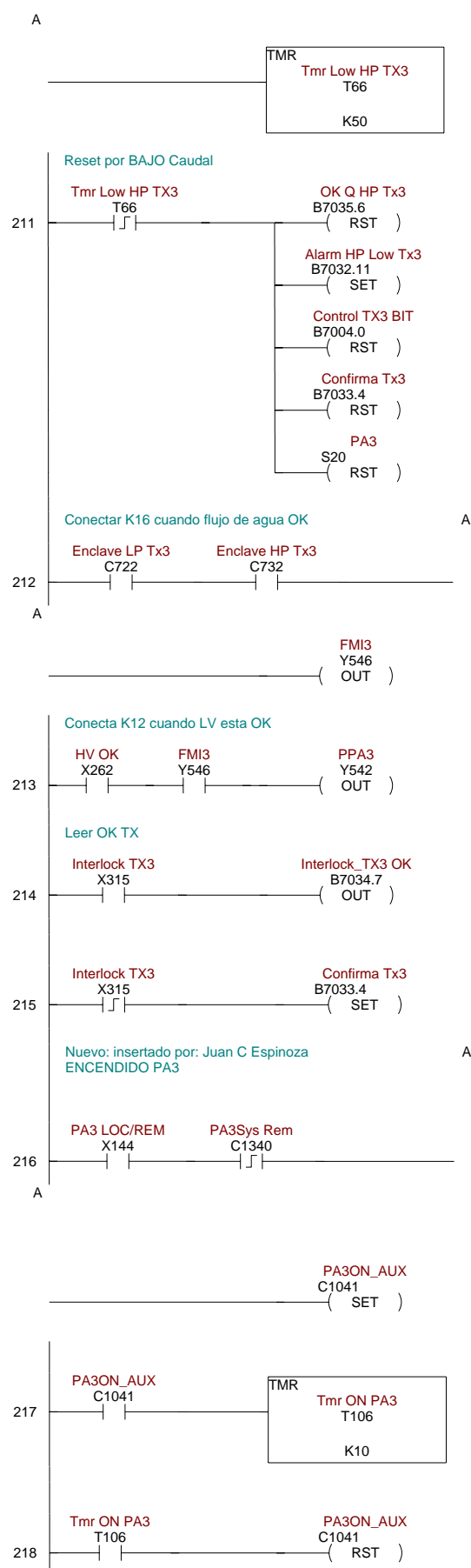
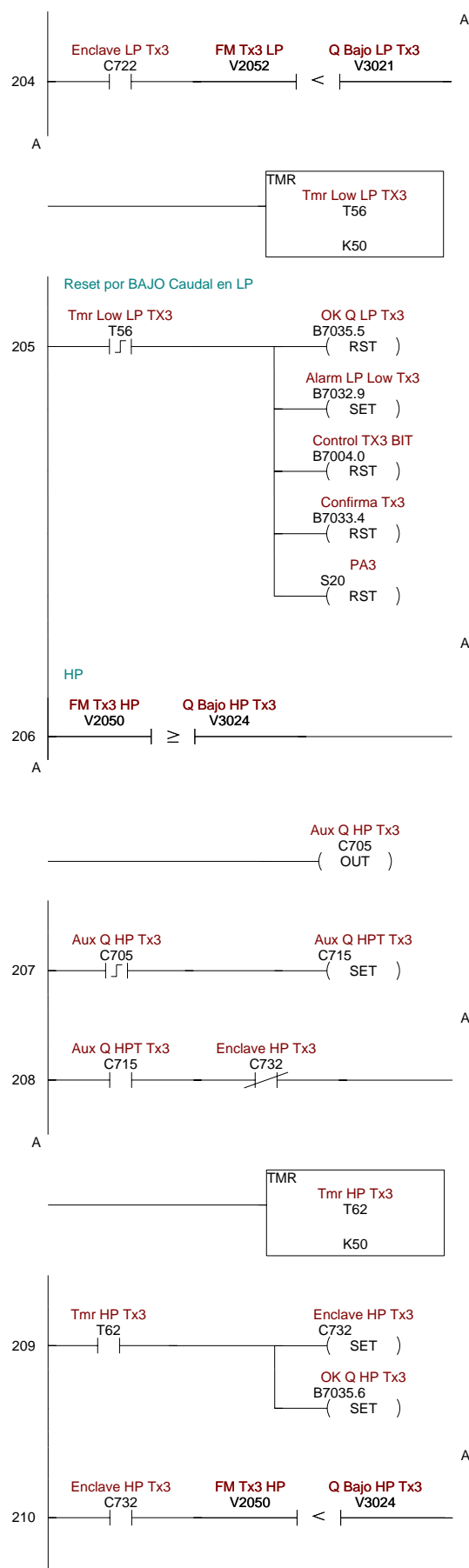


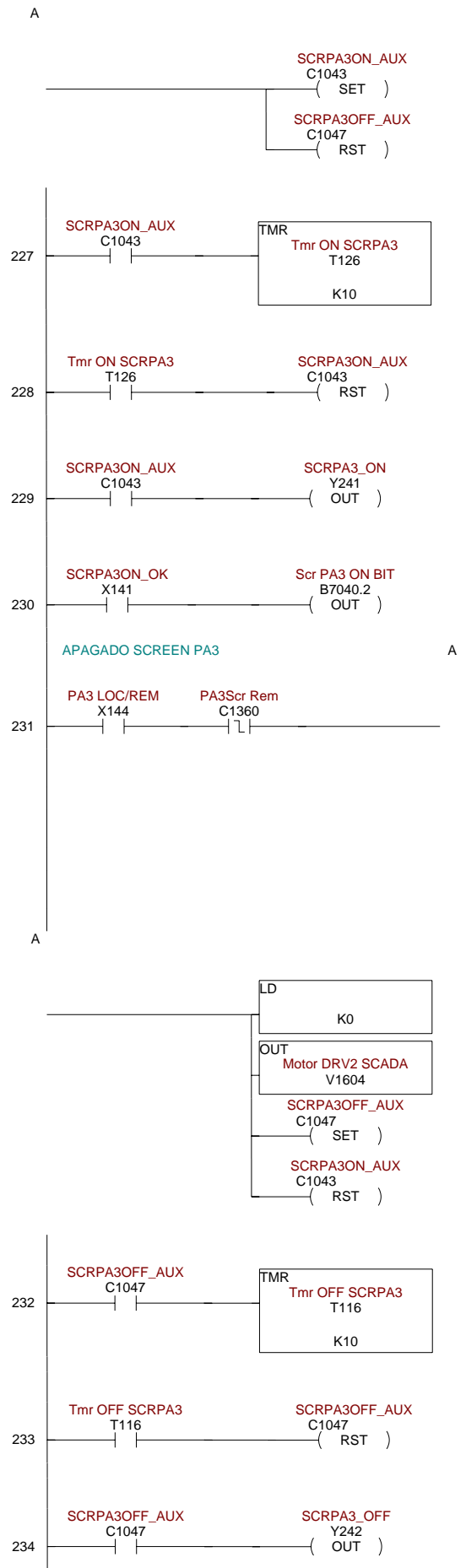
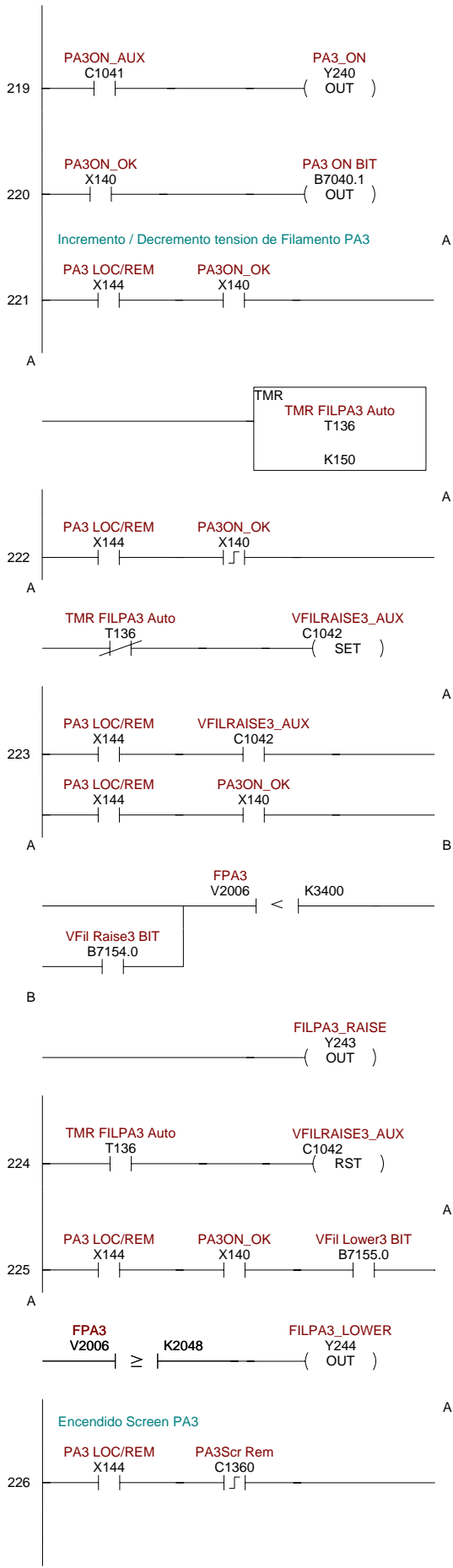


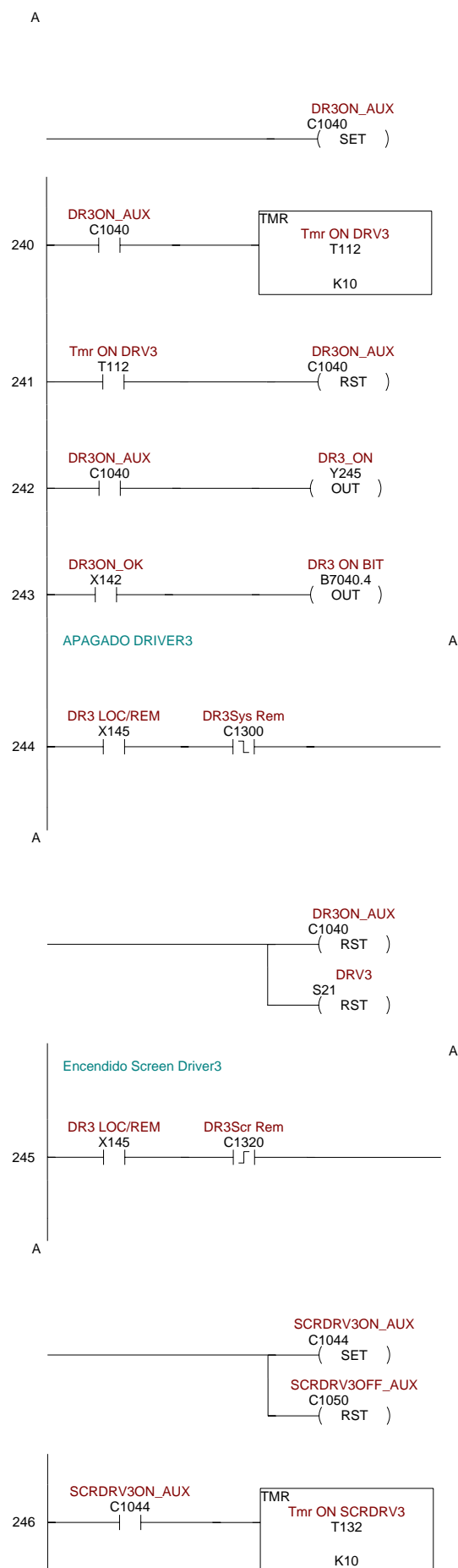
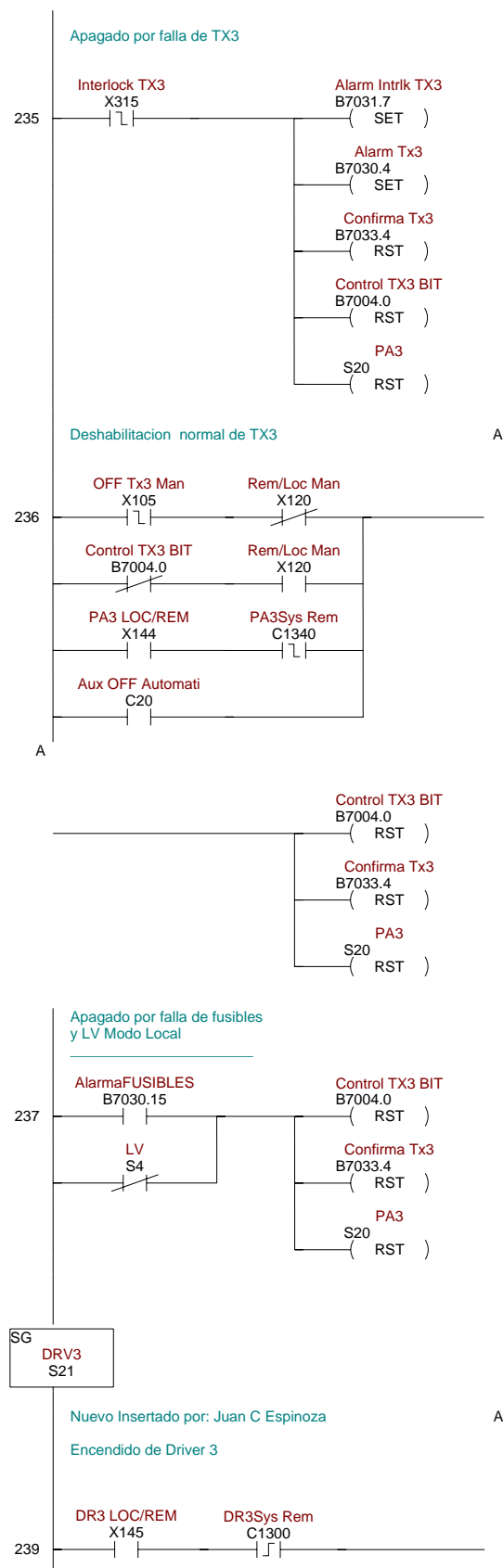


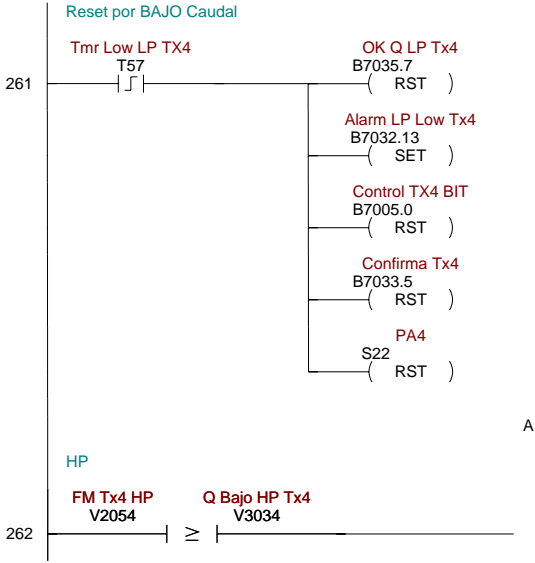
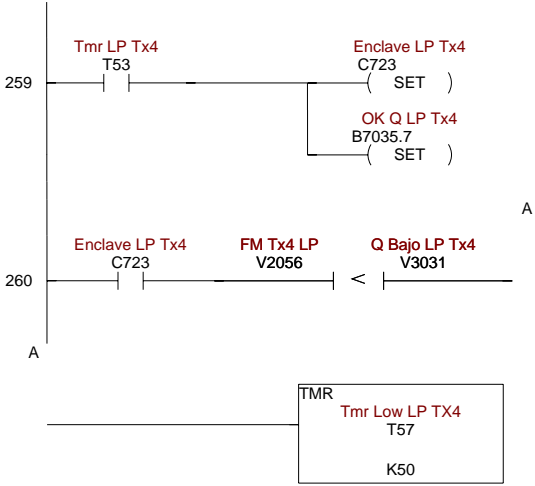
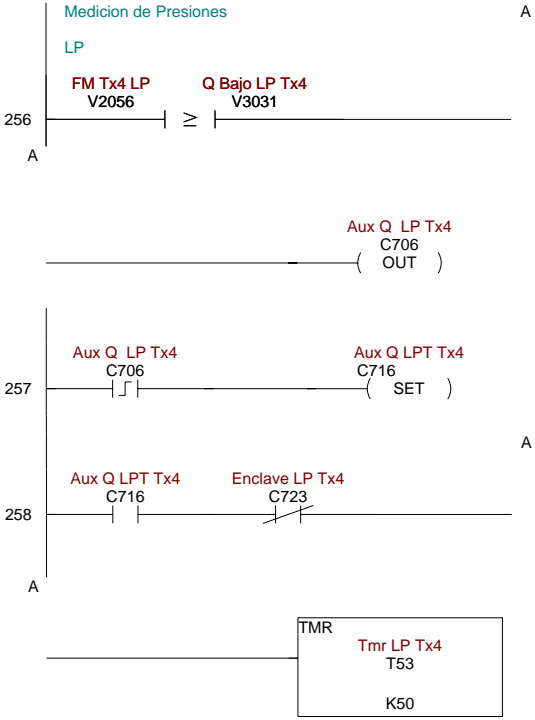
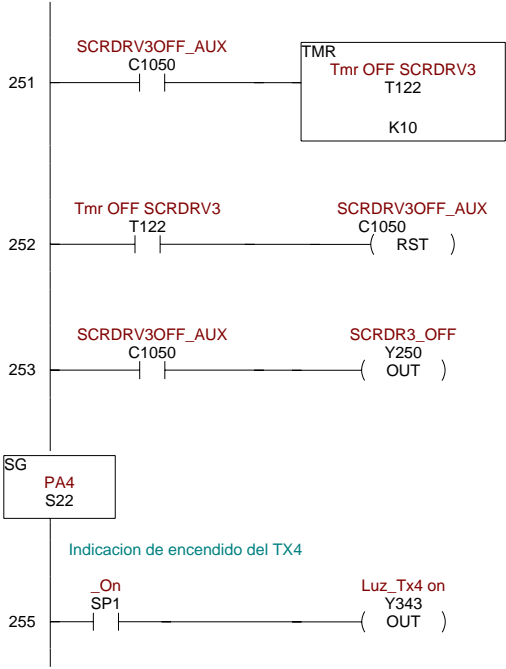
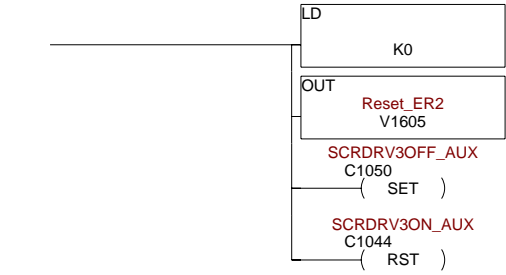
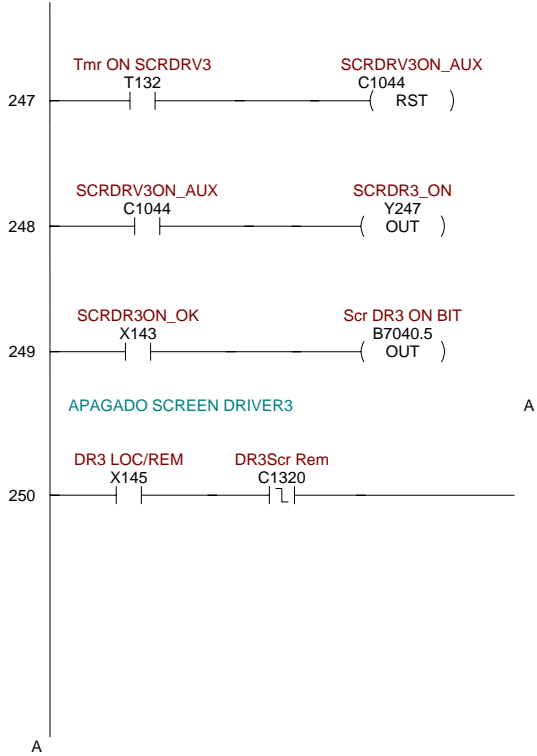


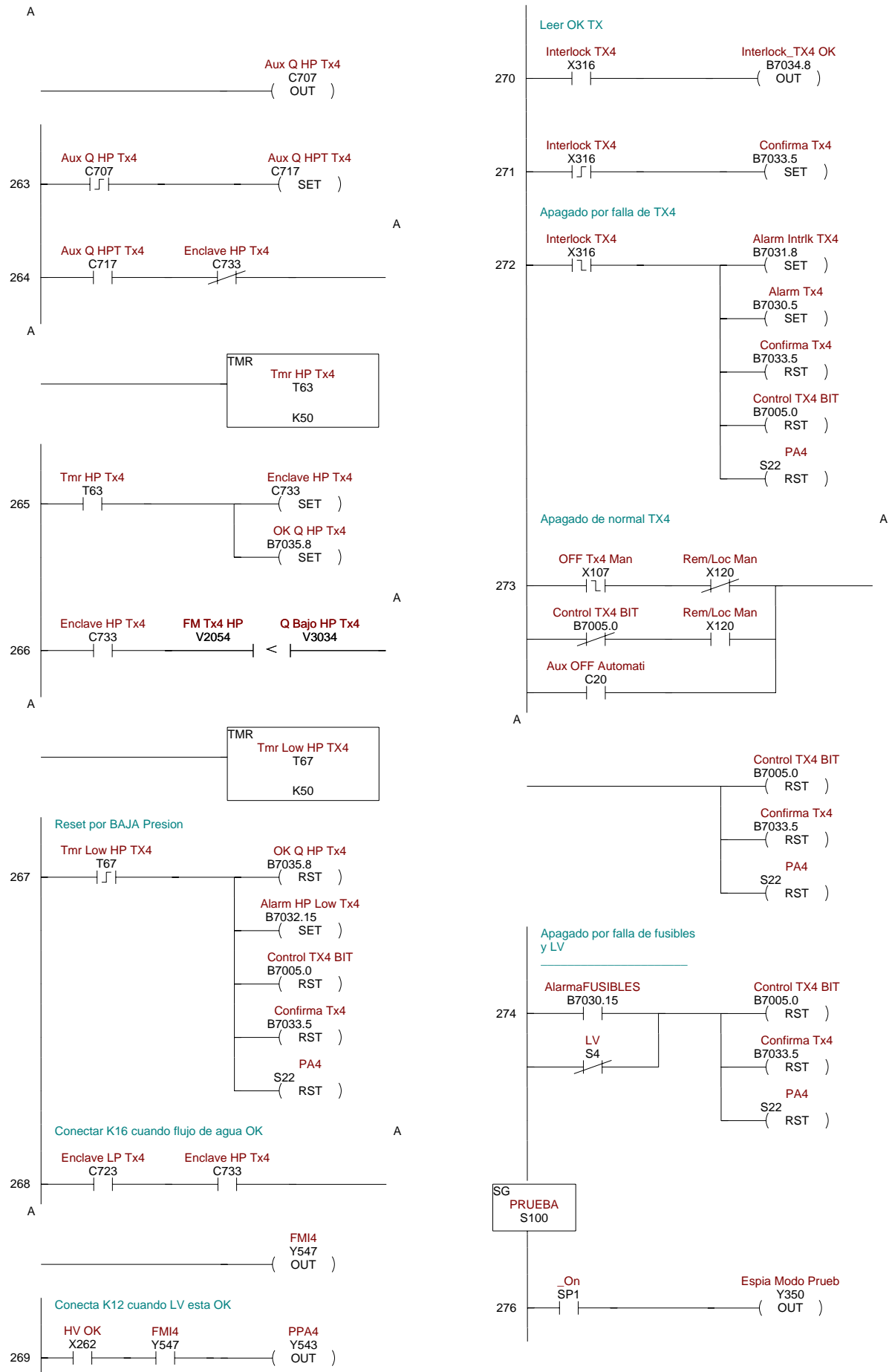


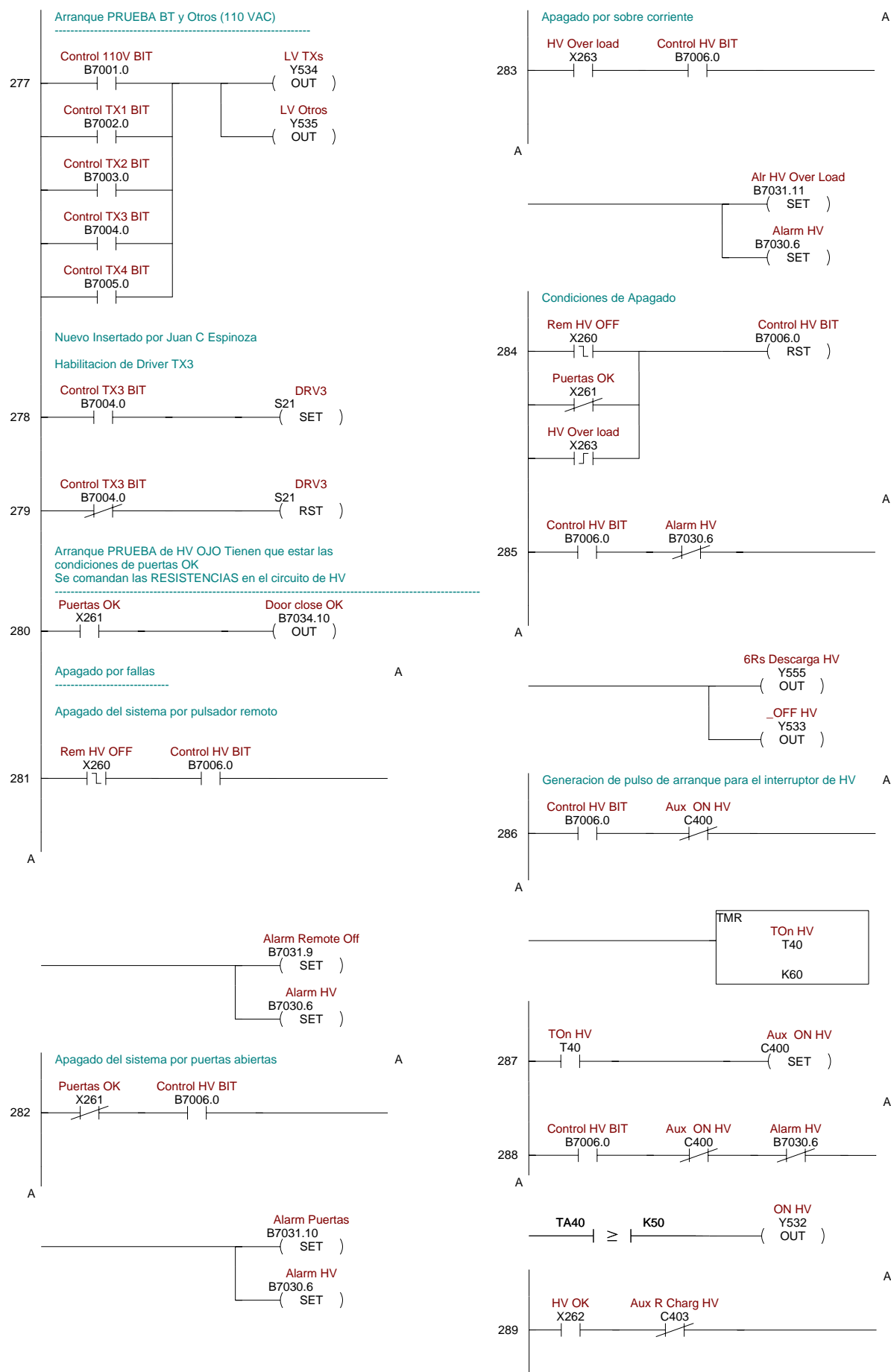


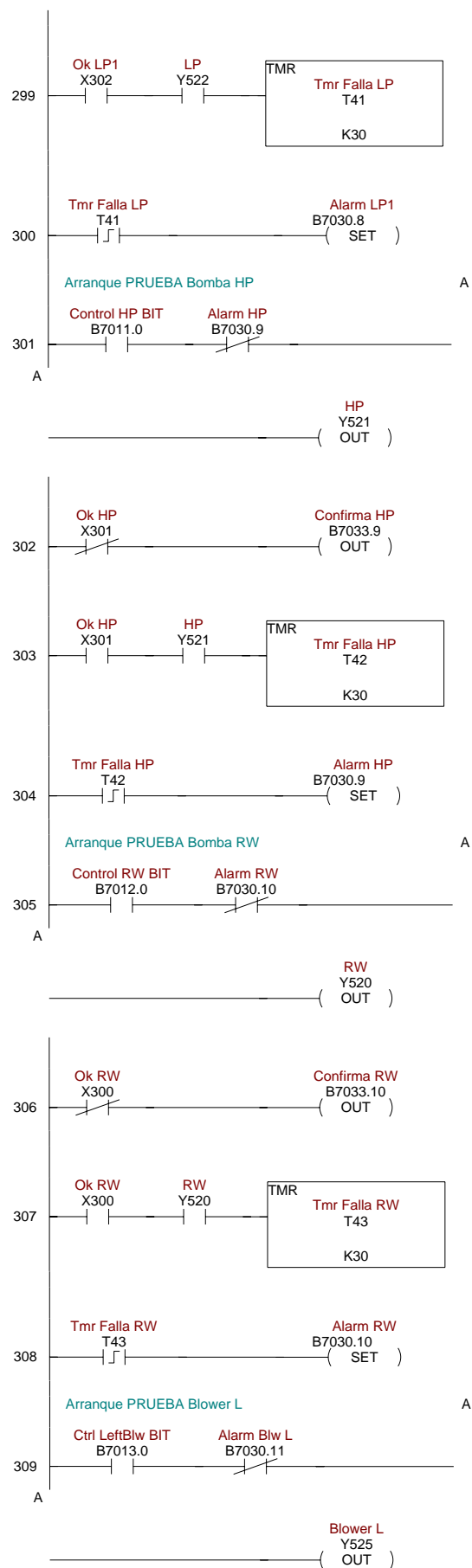
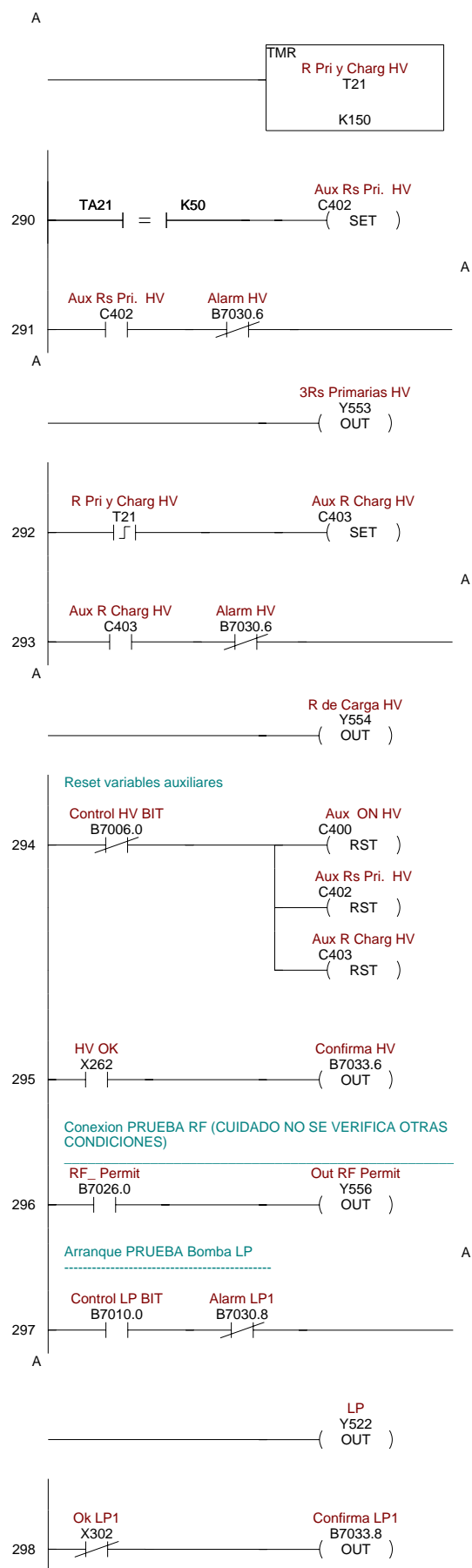


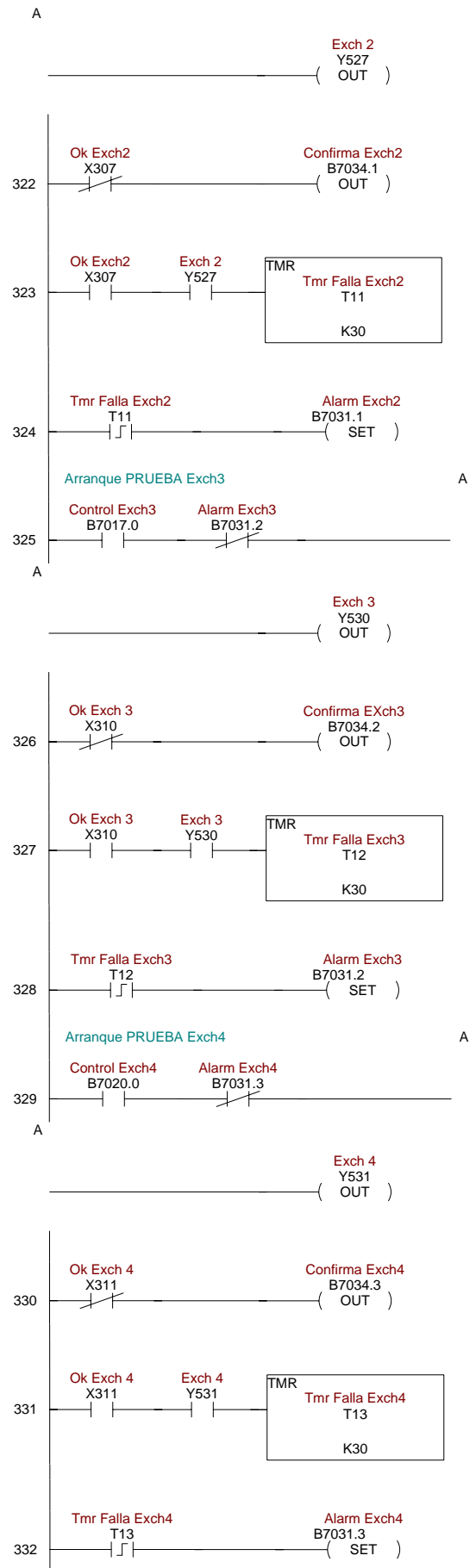
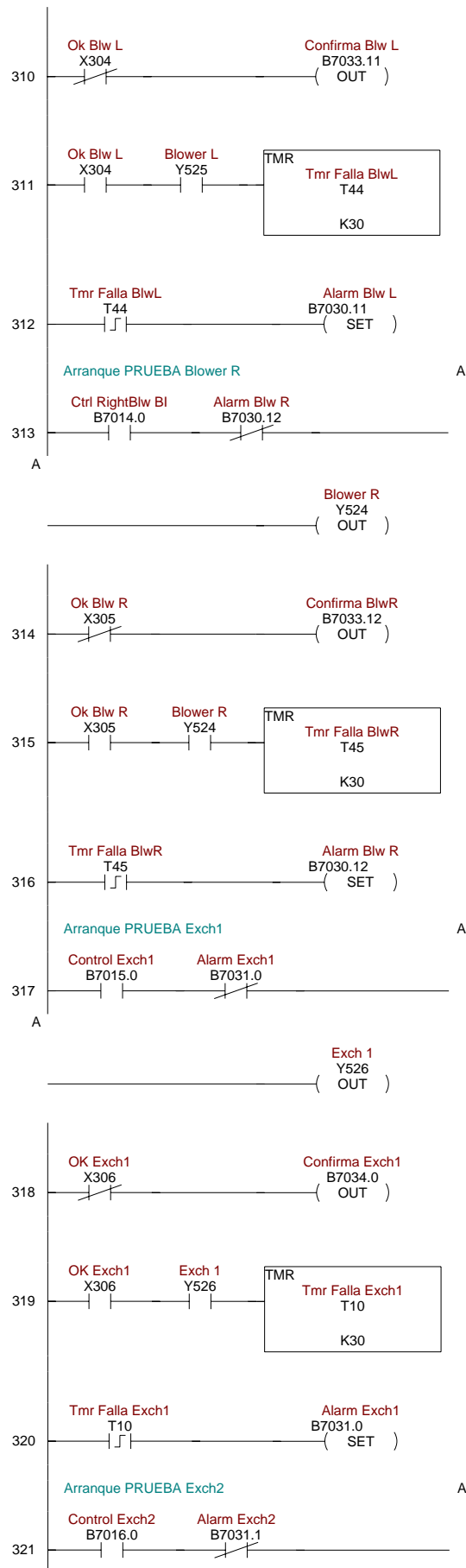


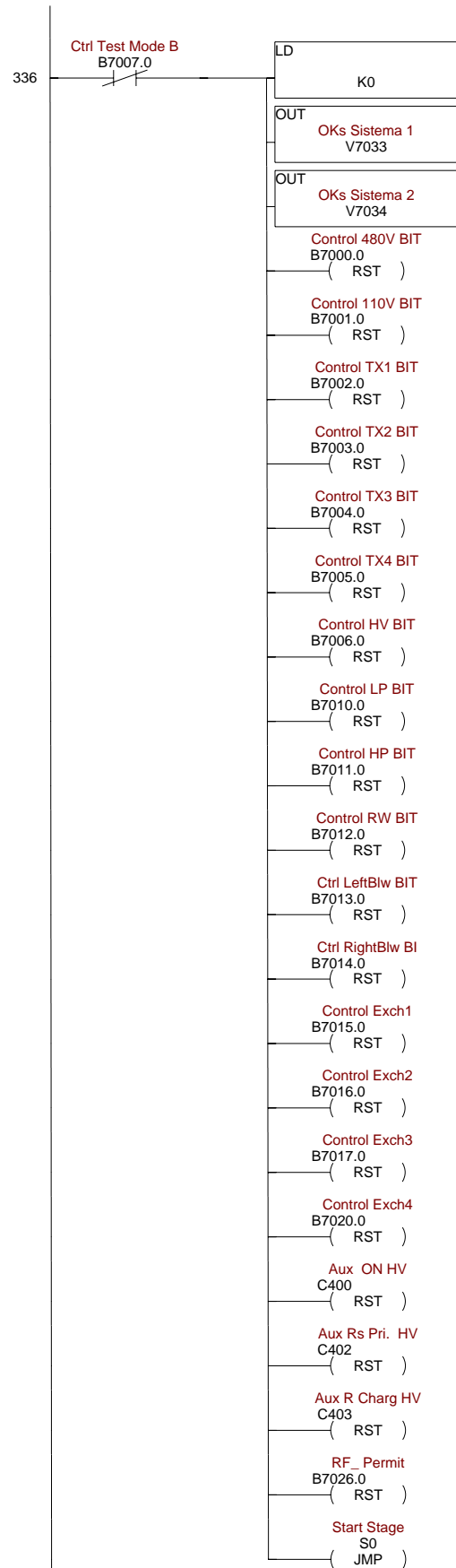
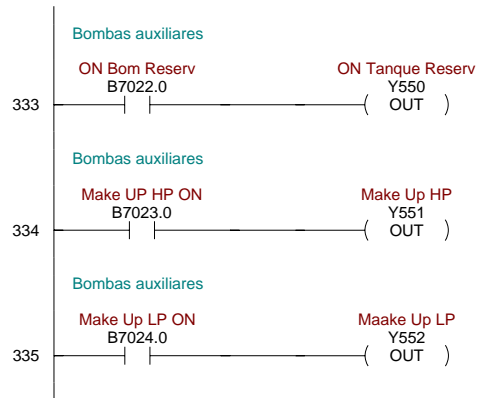


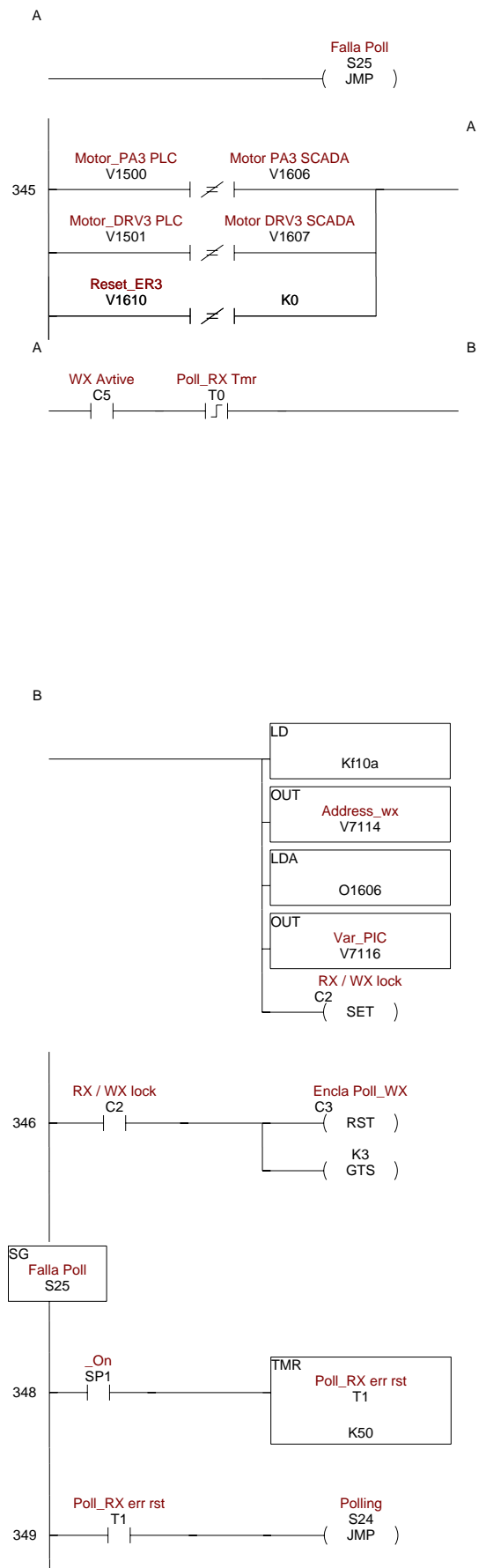
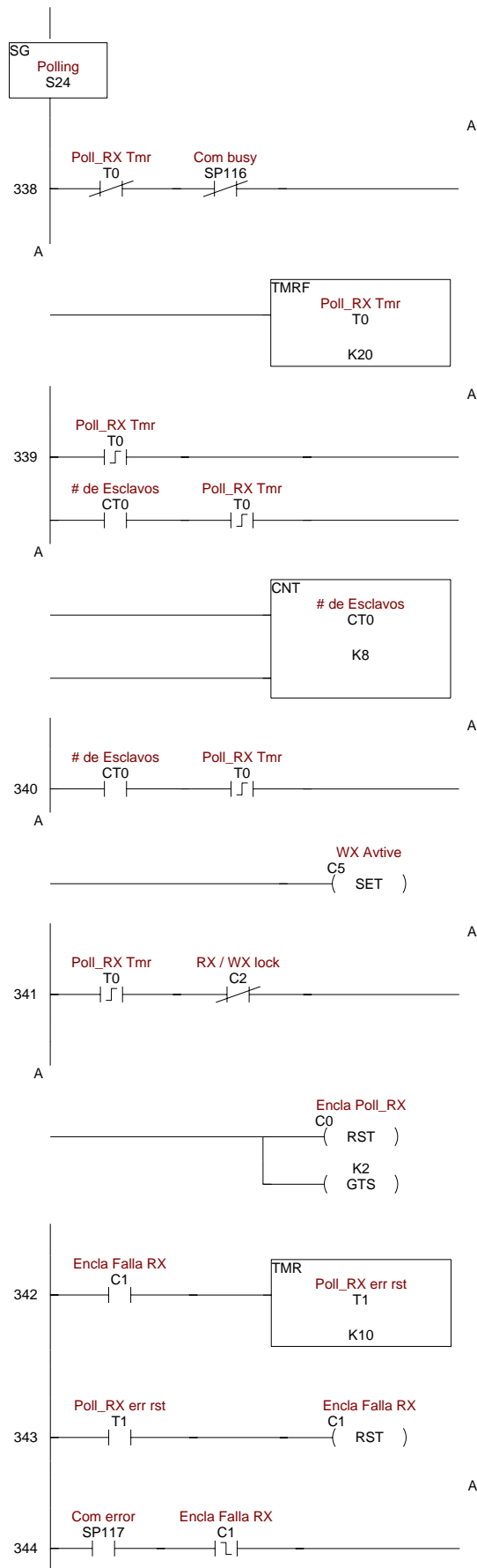


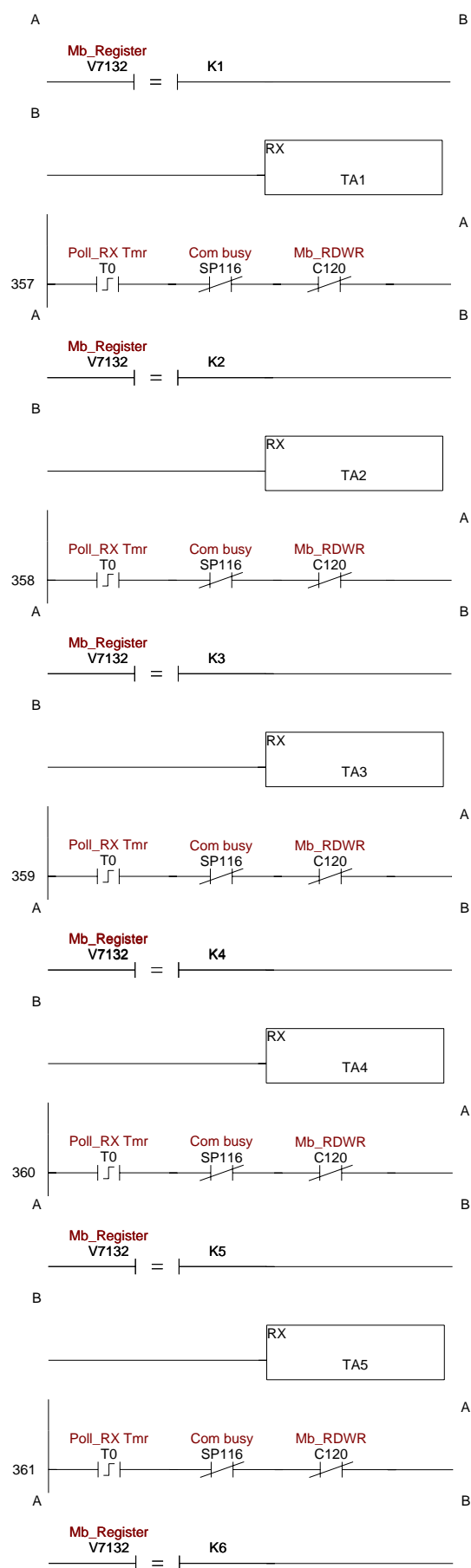
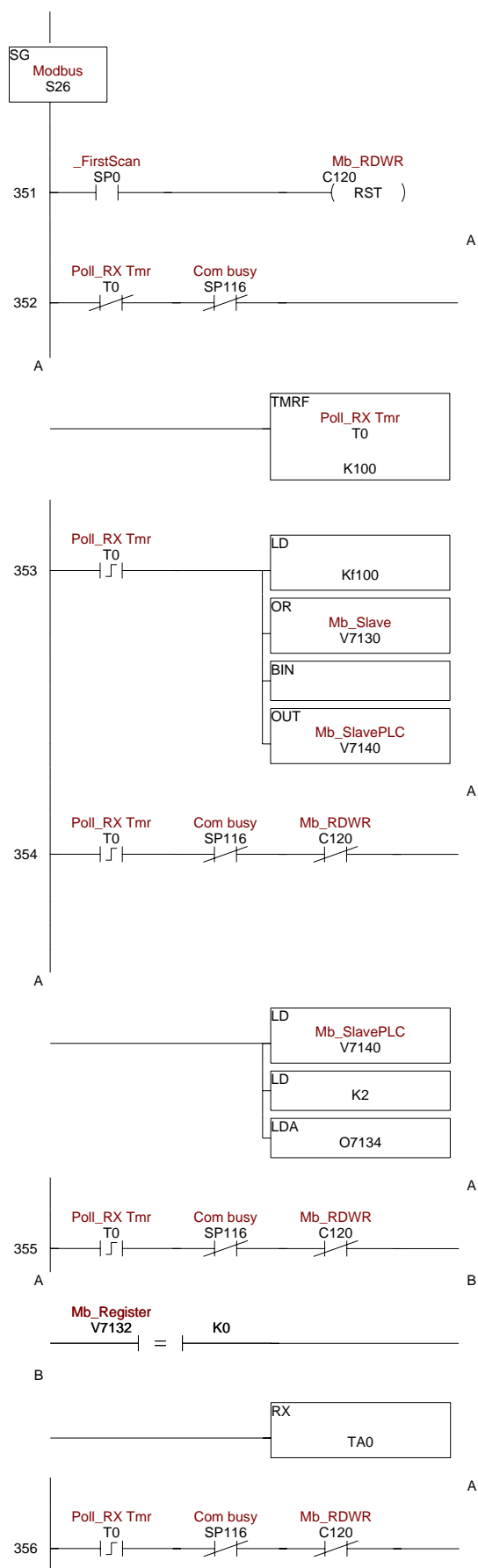


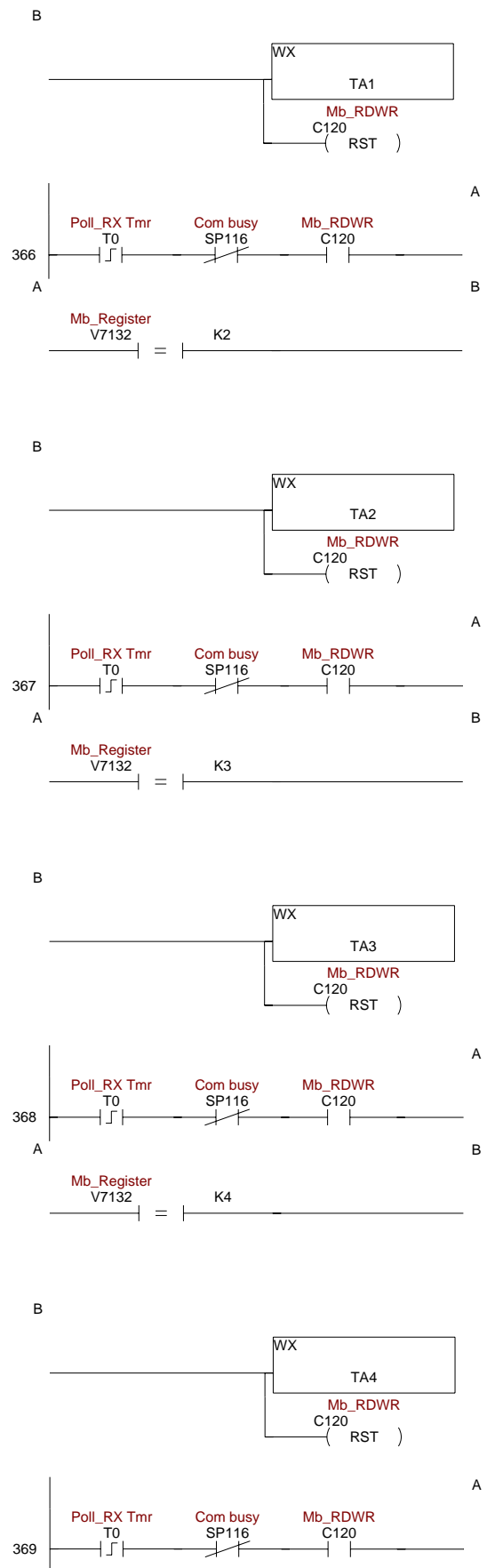
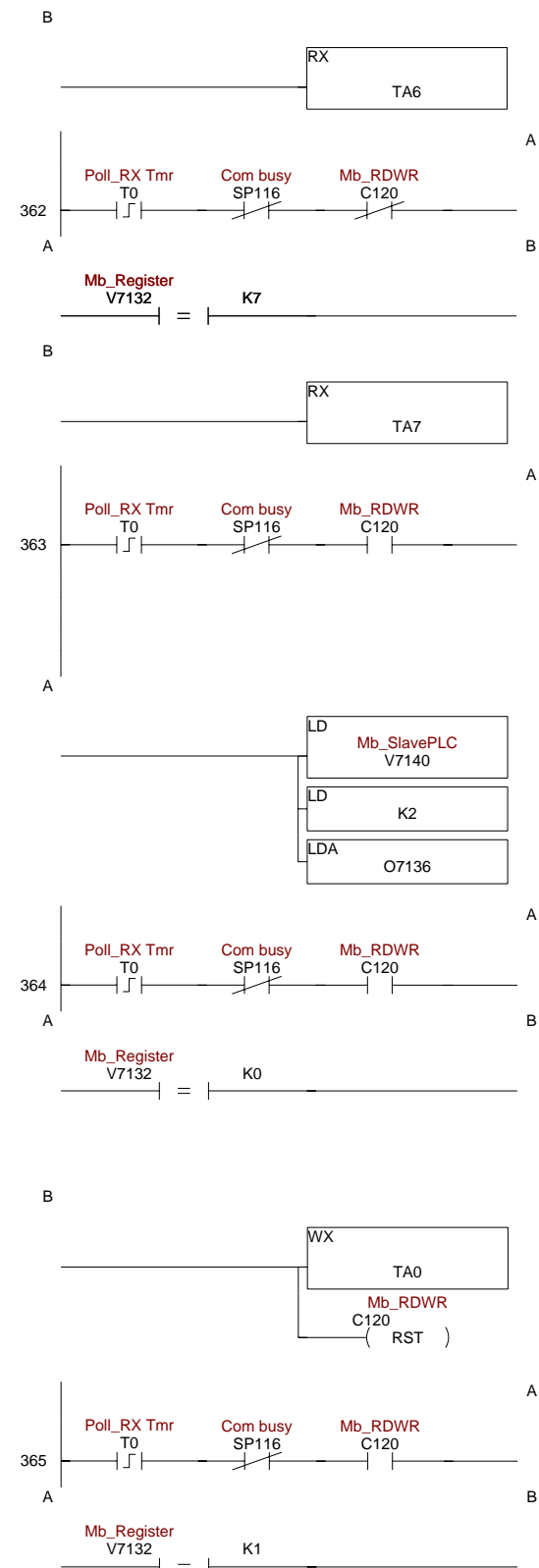


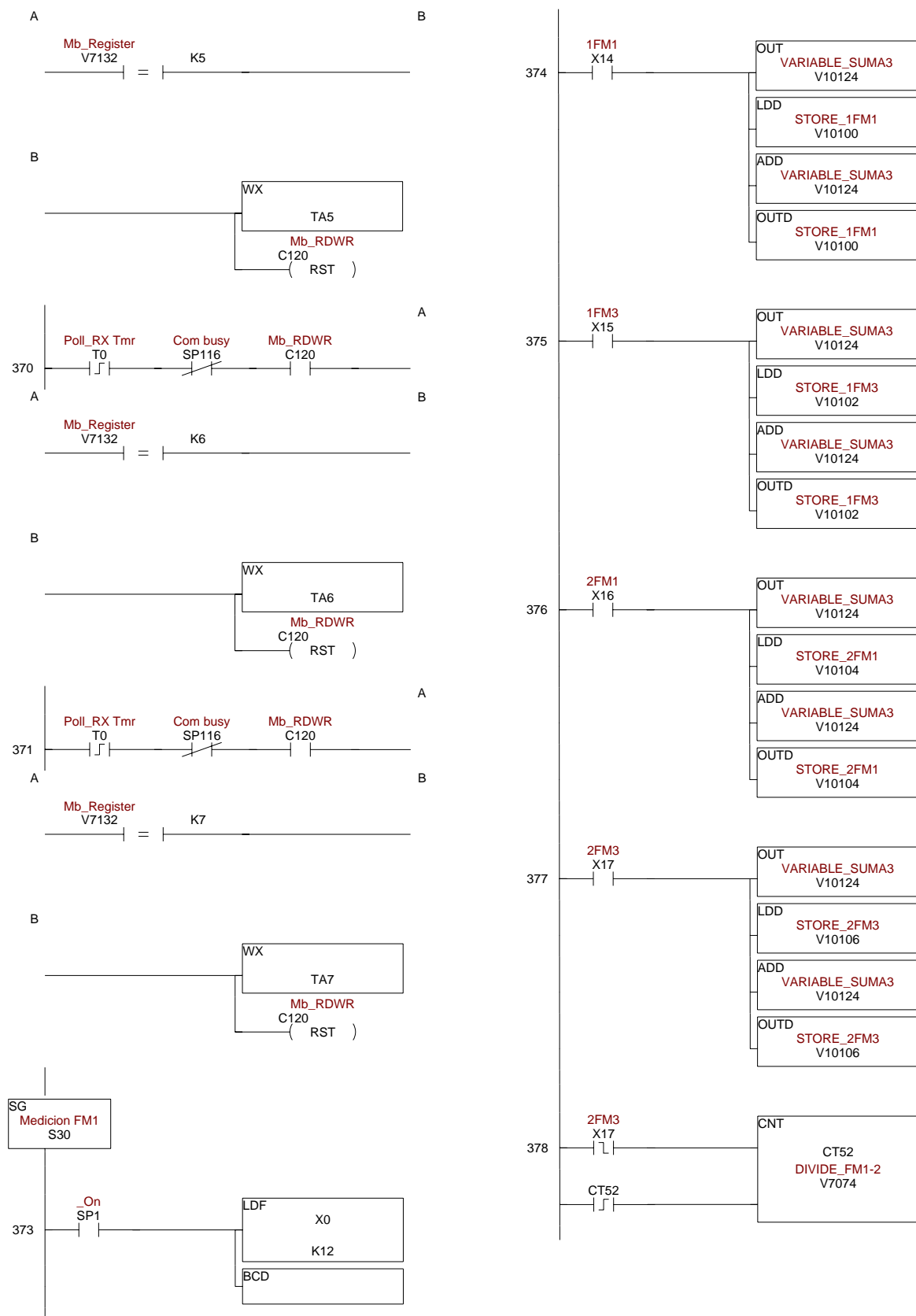


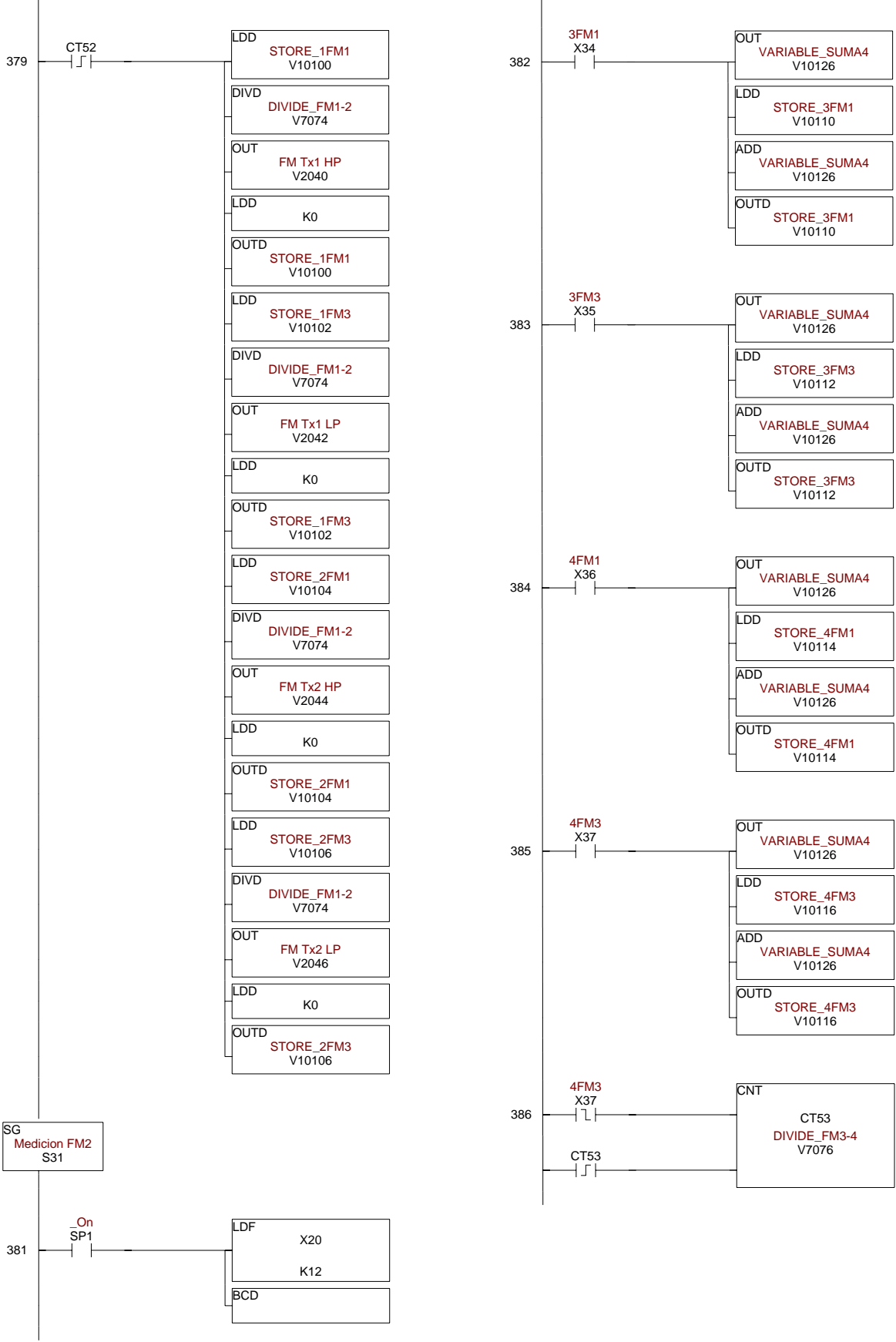


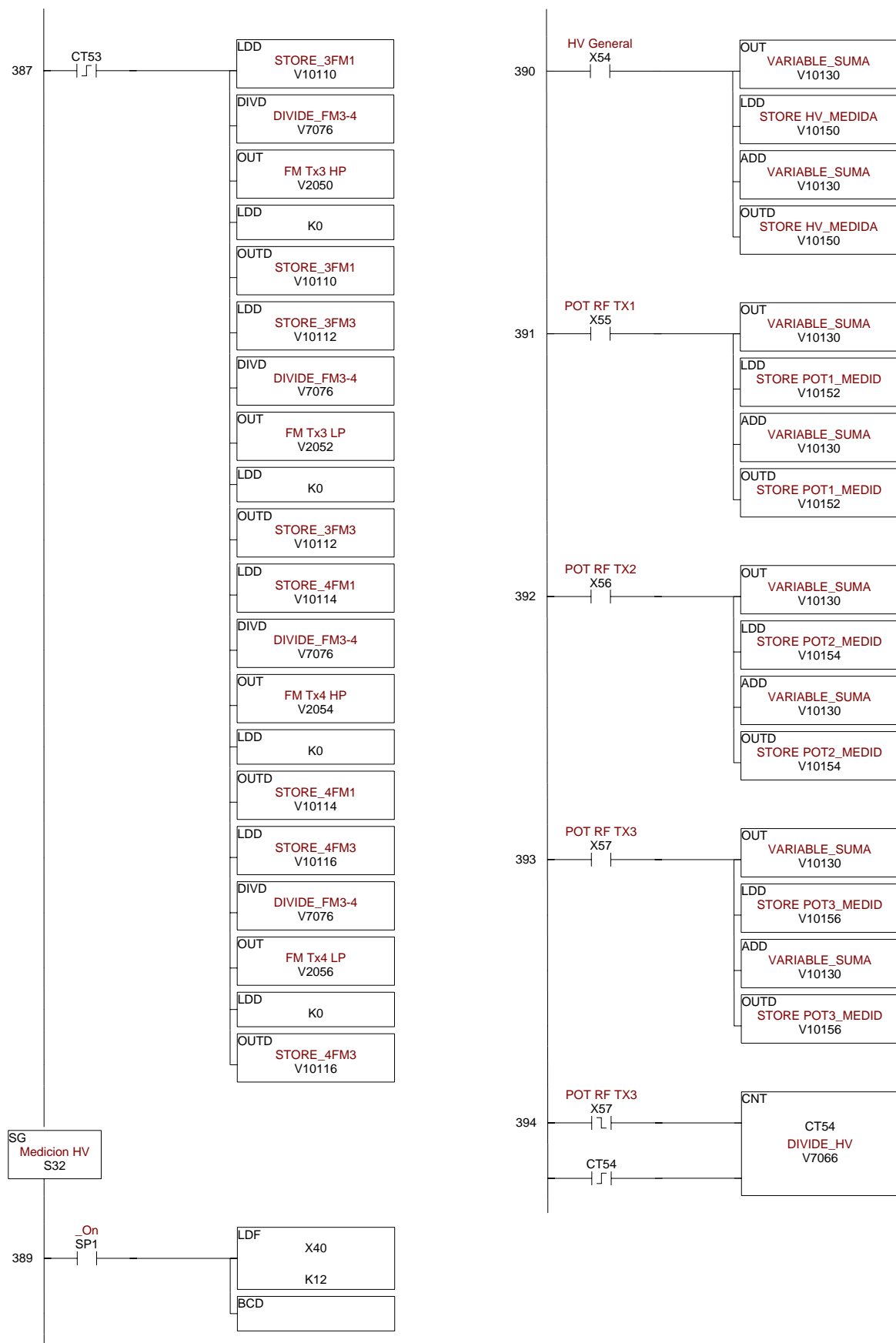


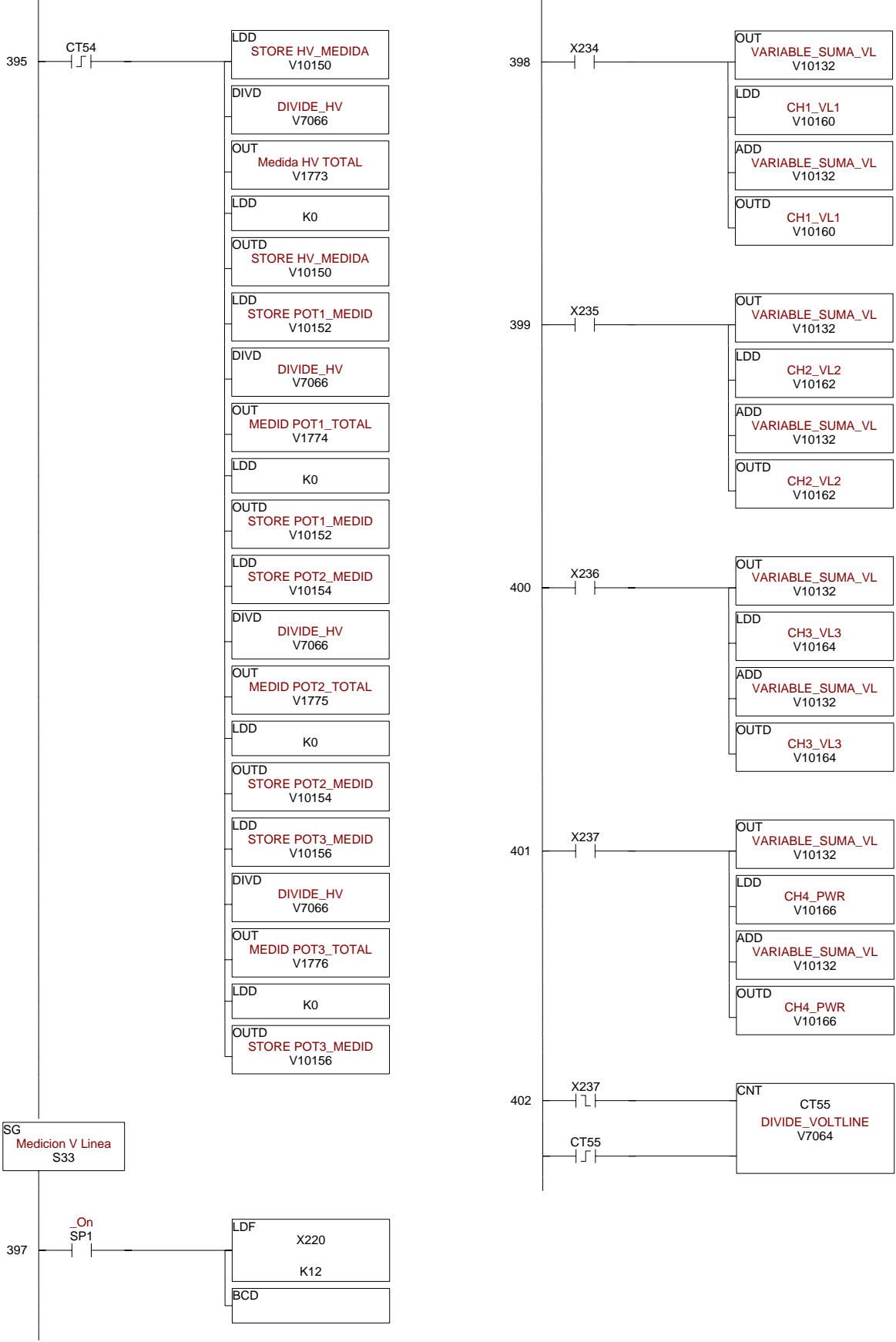


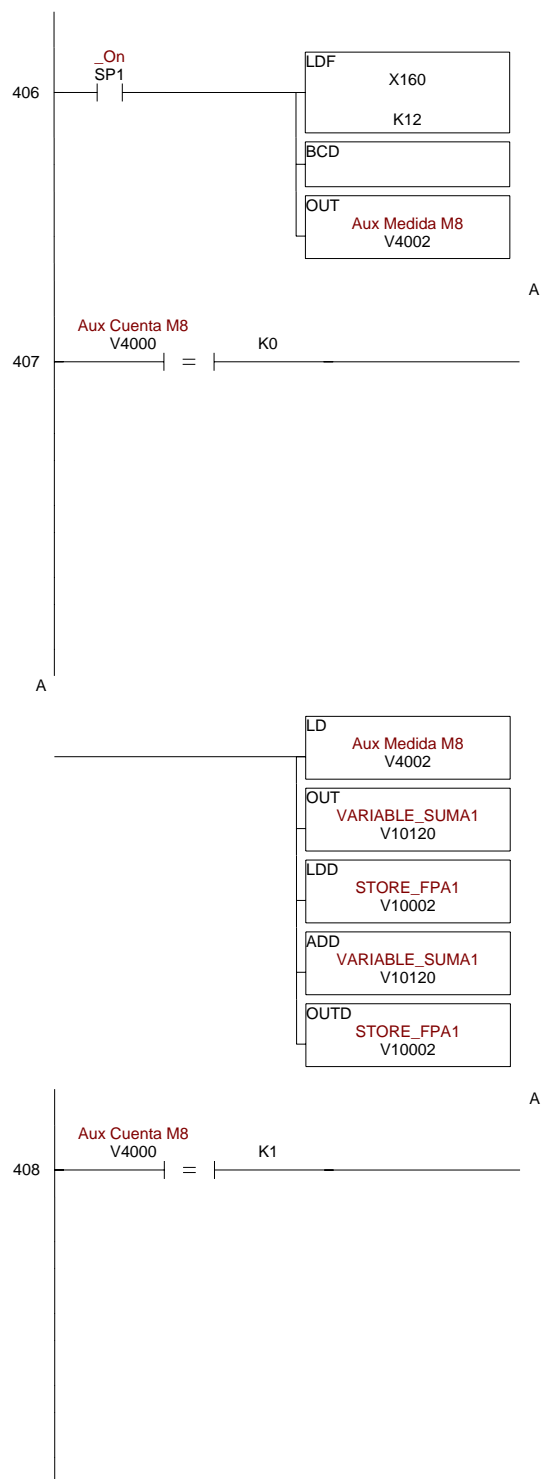
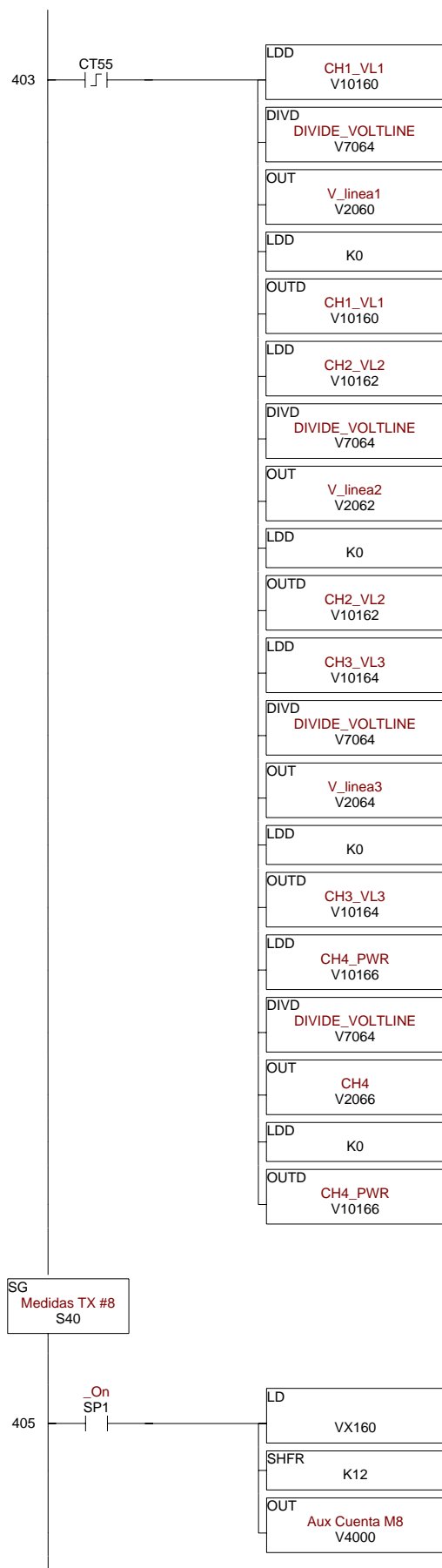


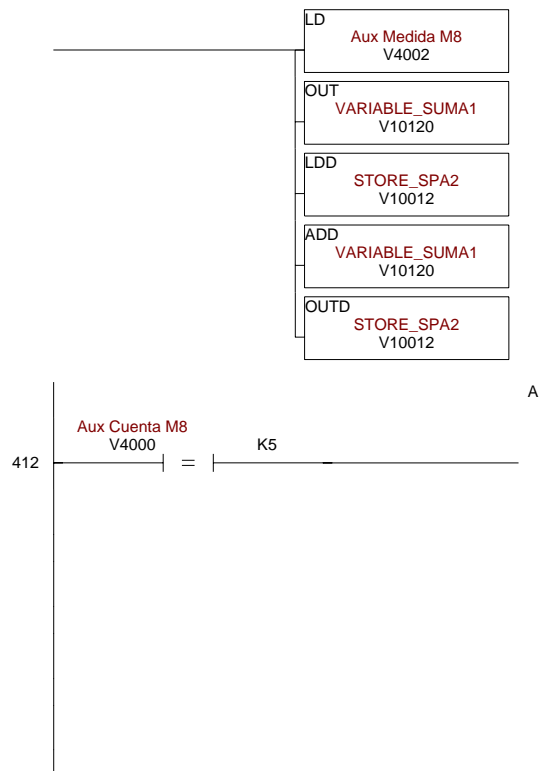
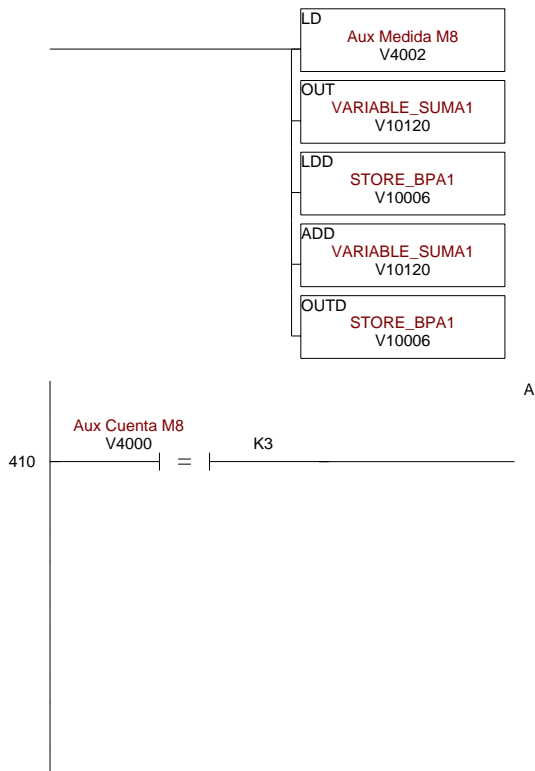
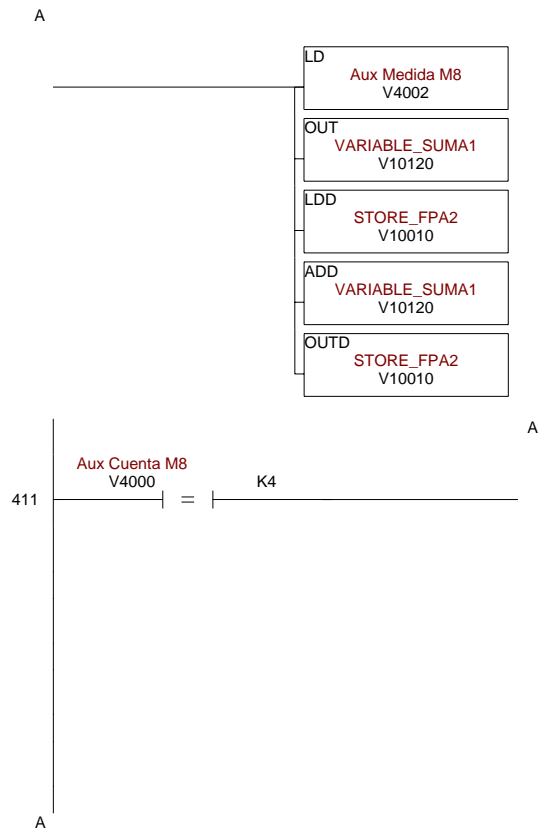
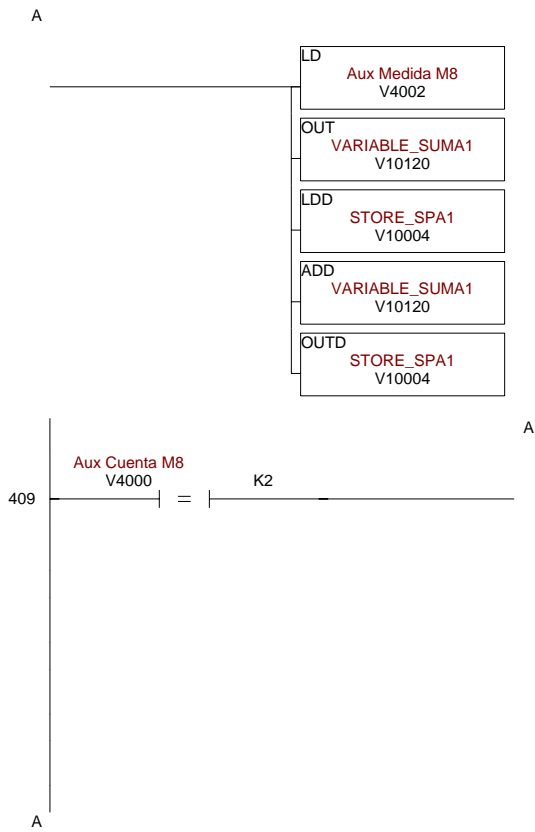


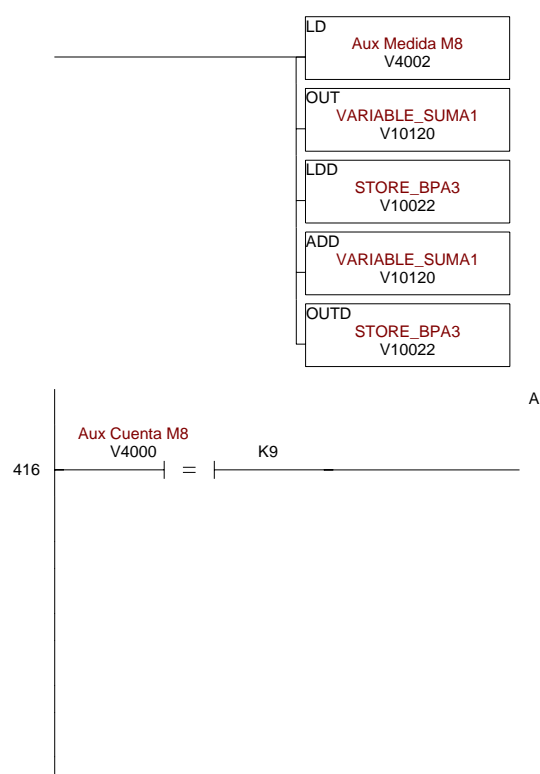
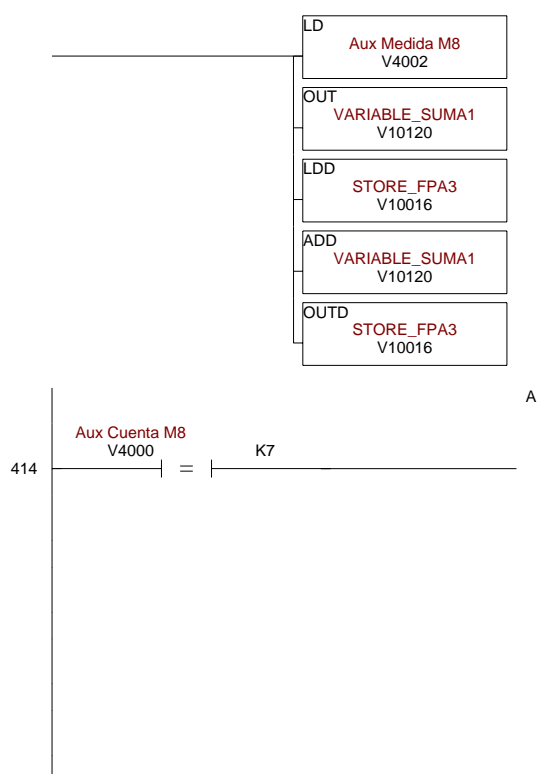
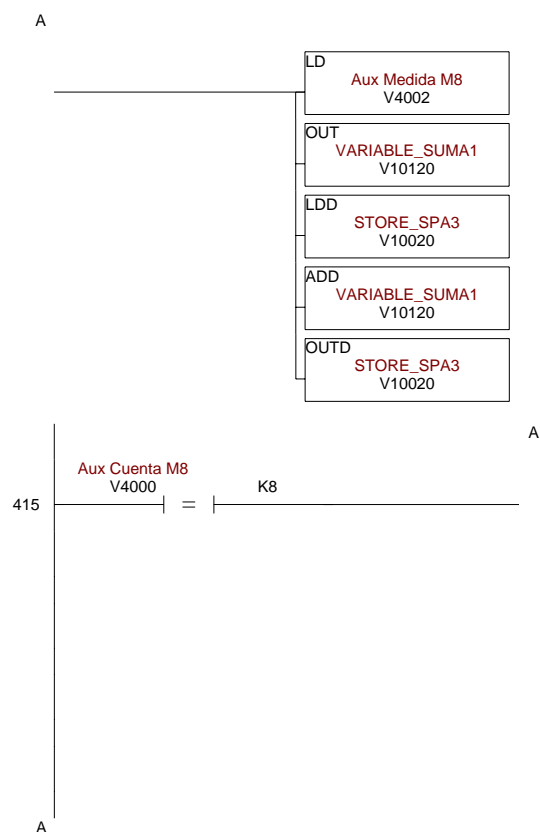
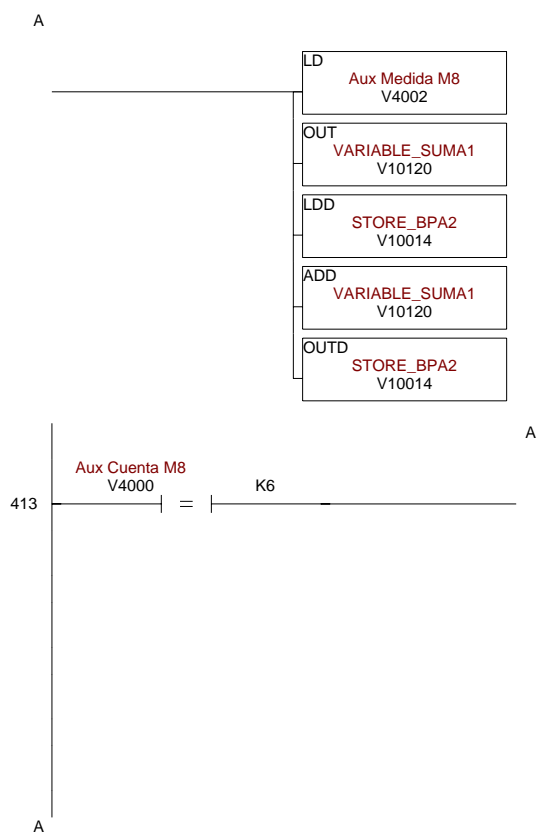


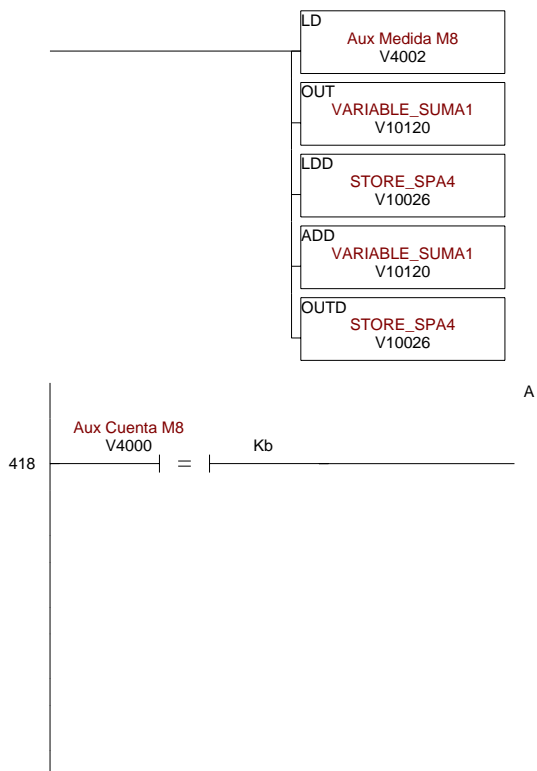
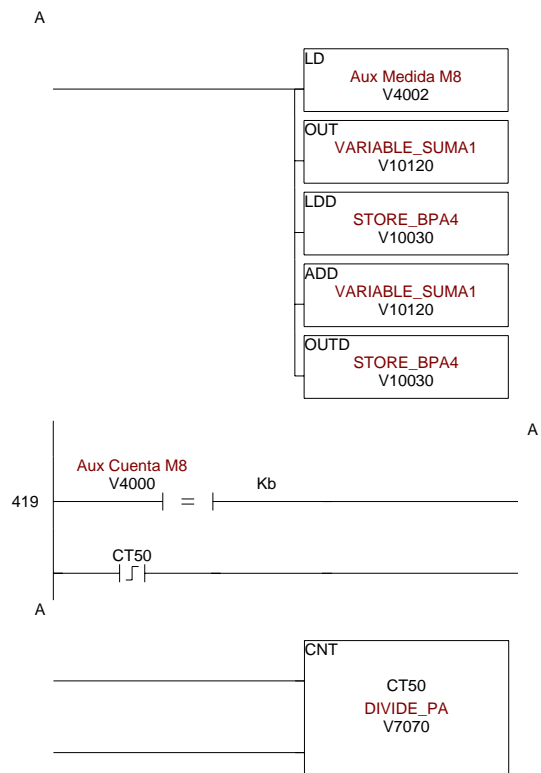
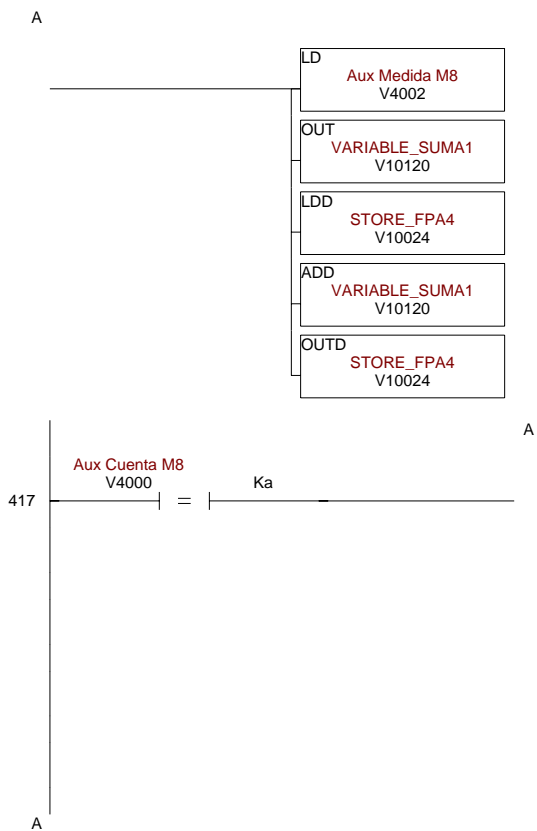






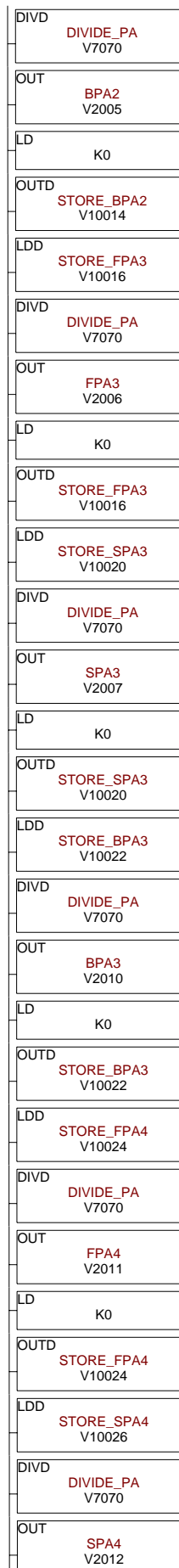
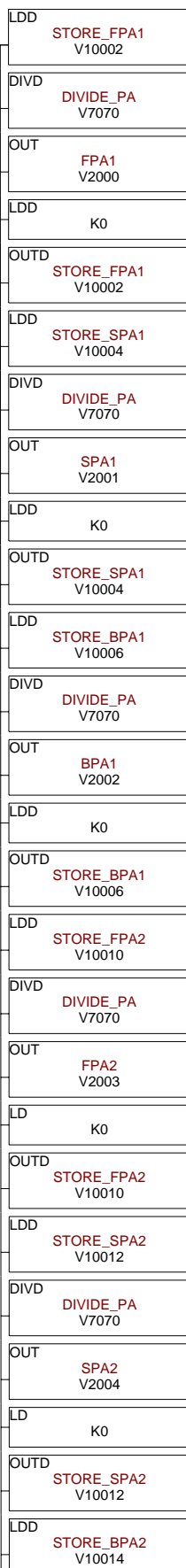


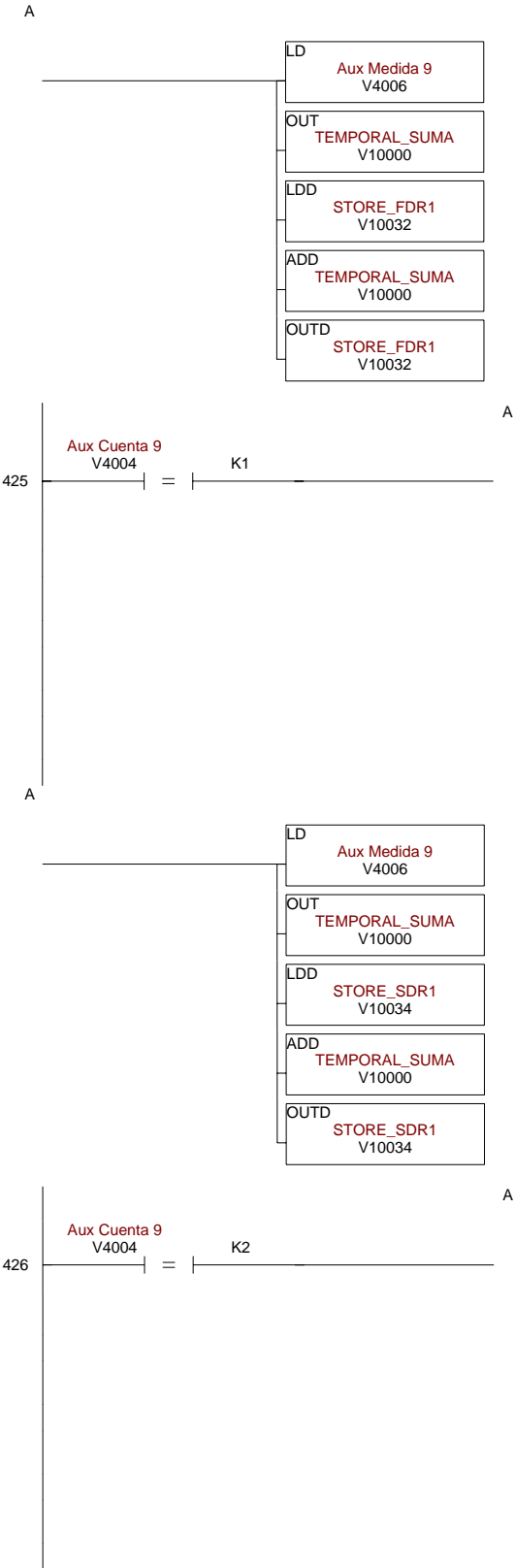
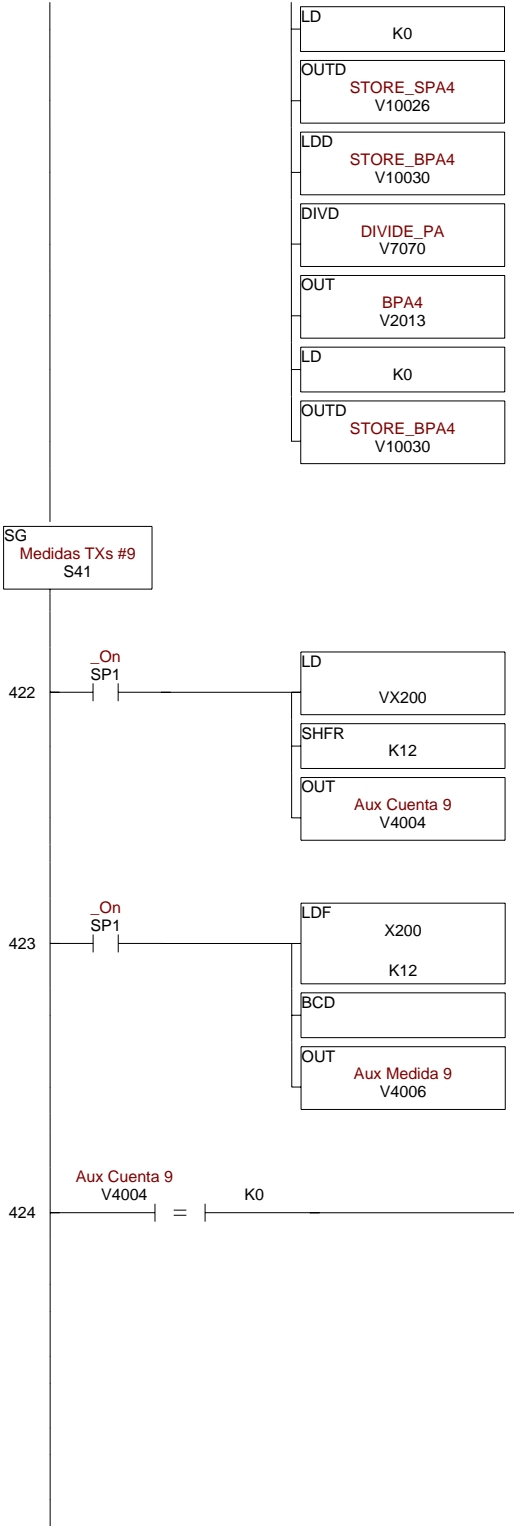


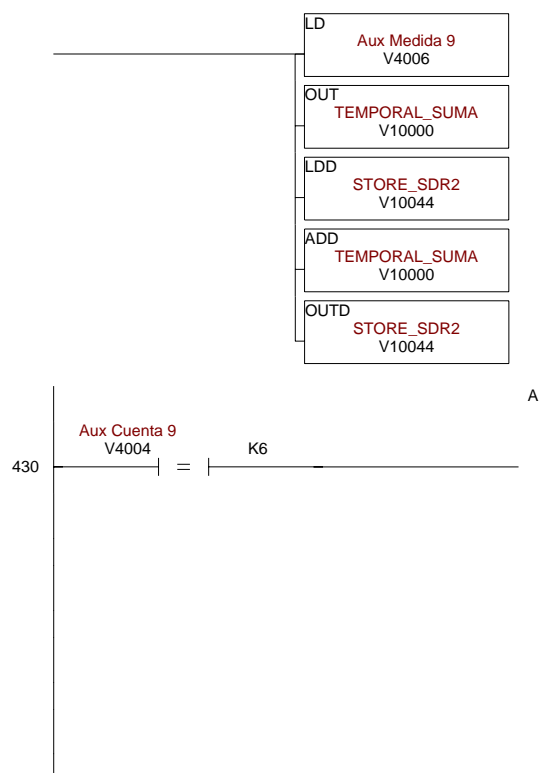
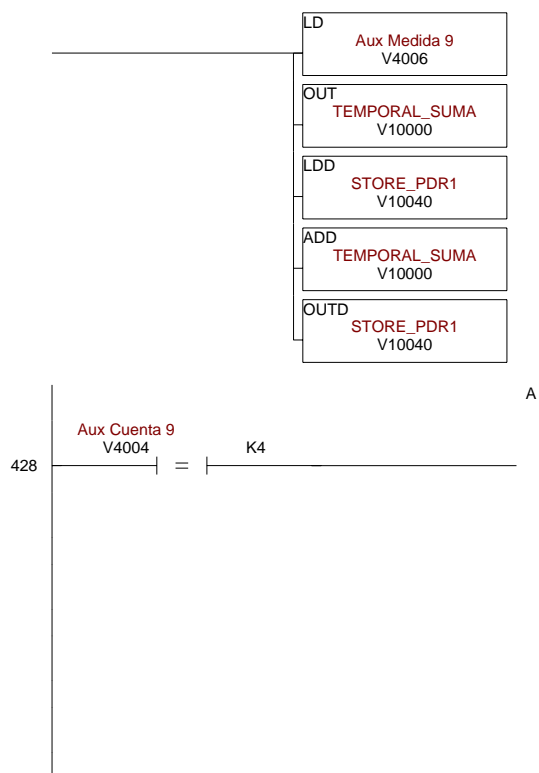
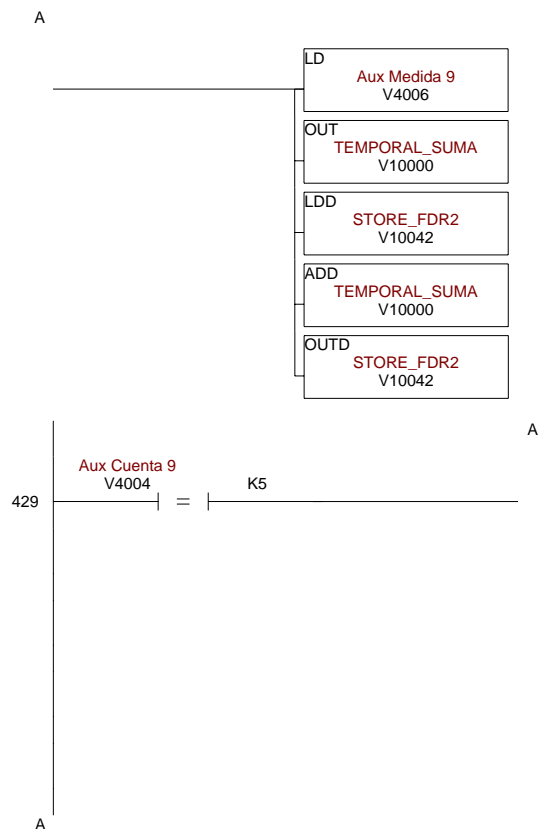
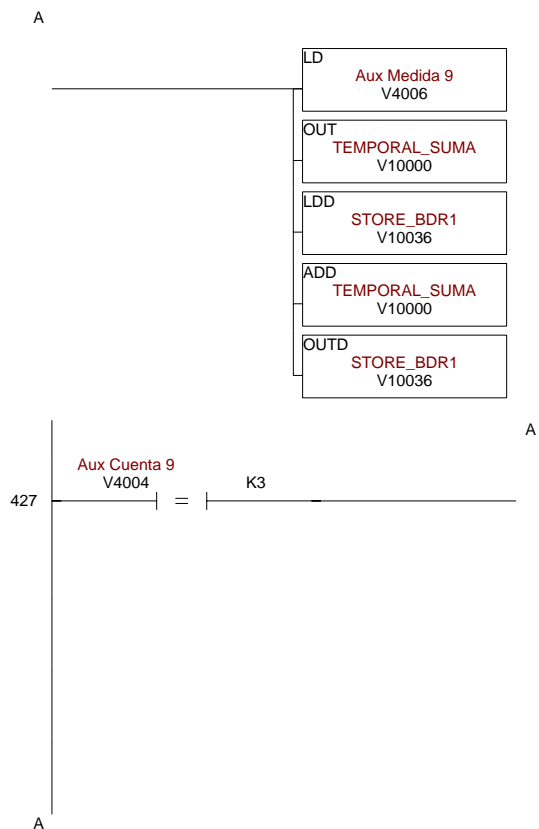


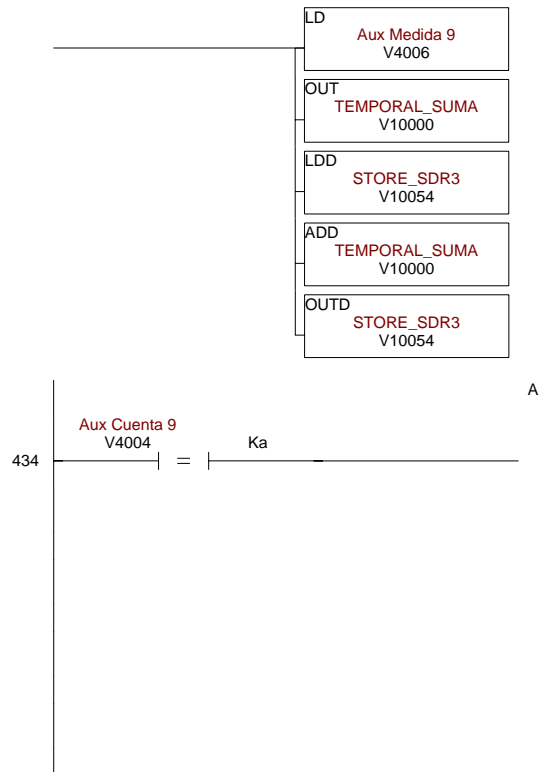
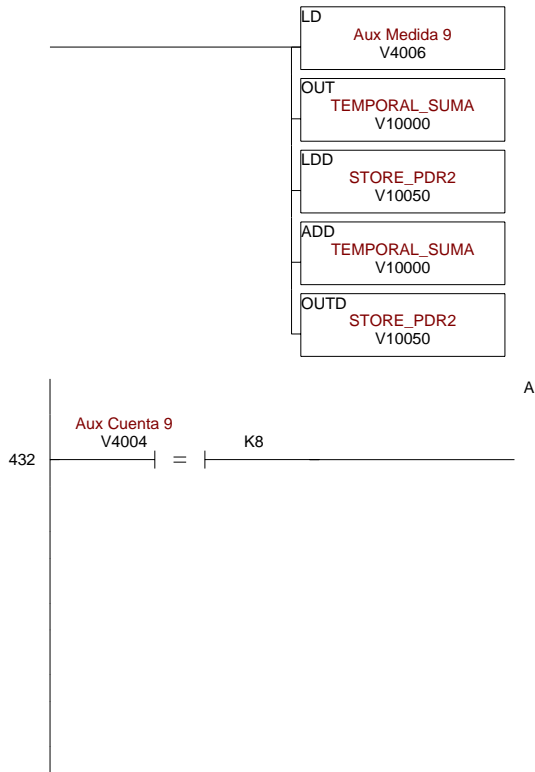
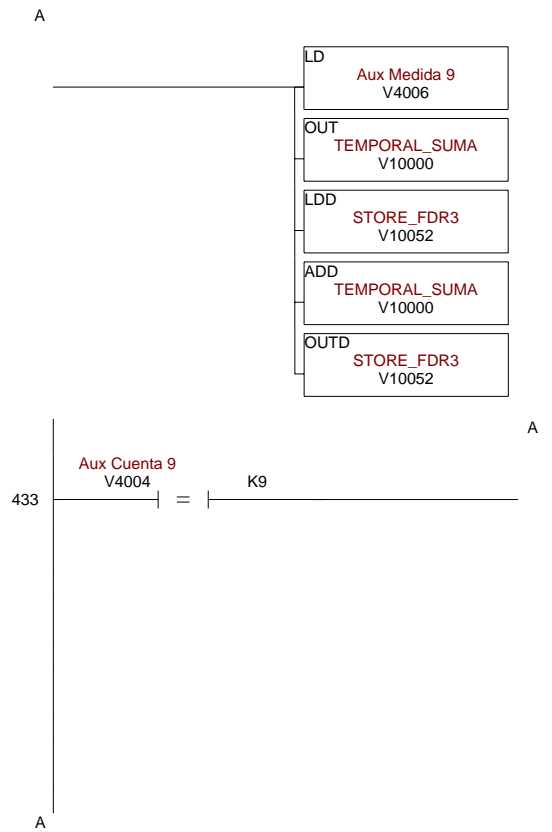
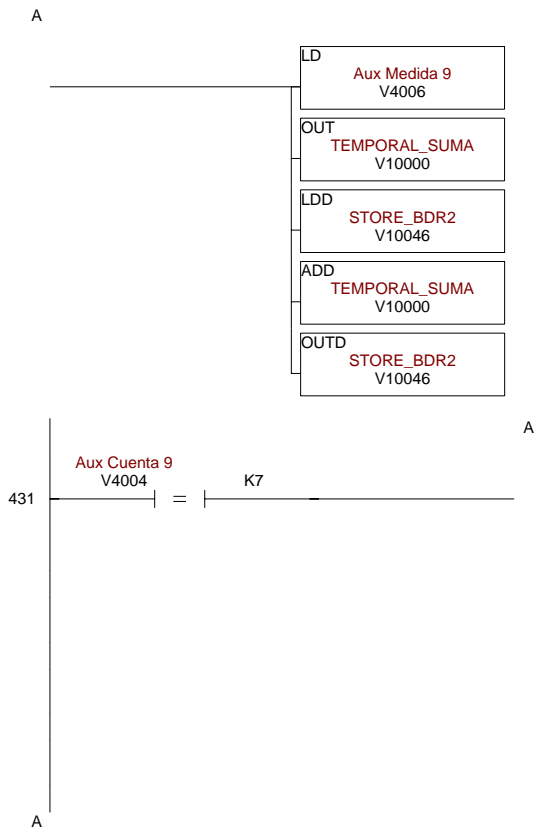
420

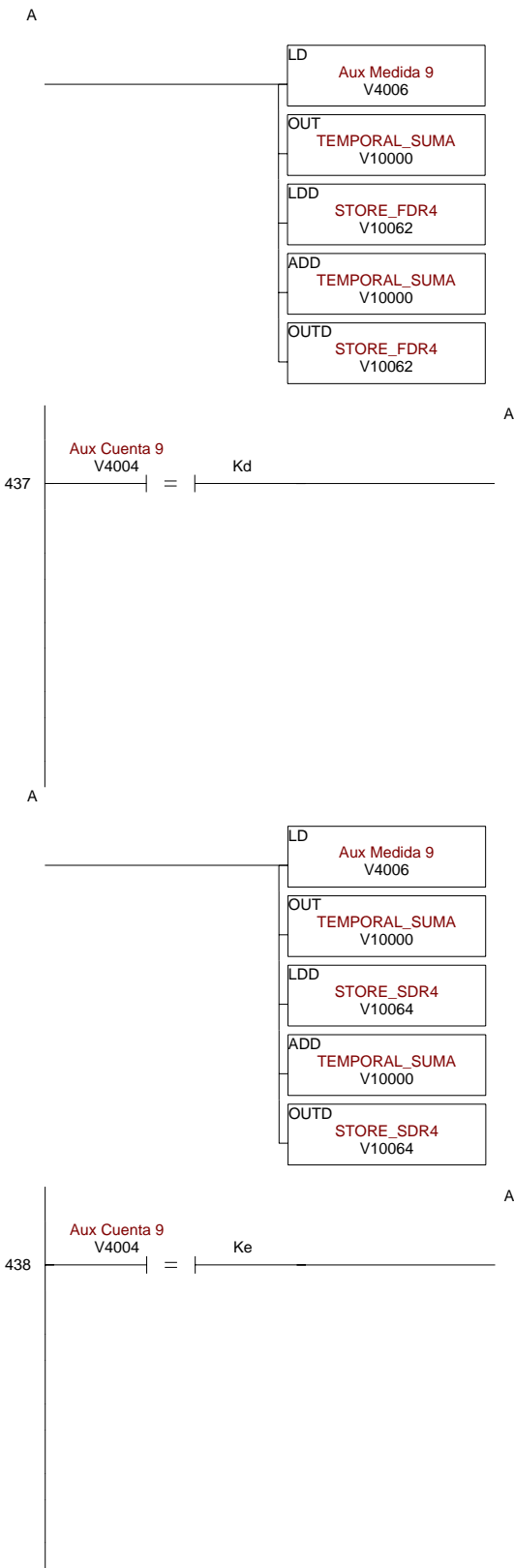
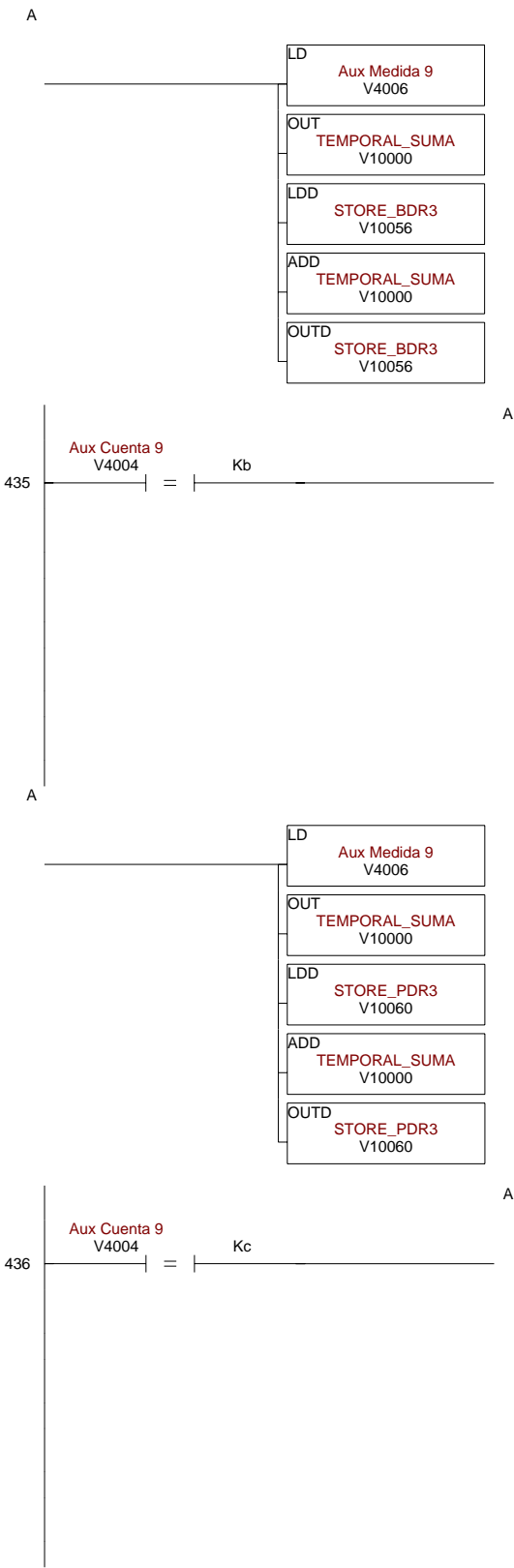
CT50

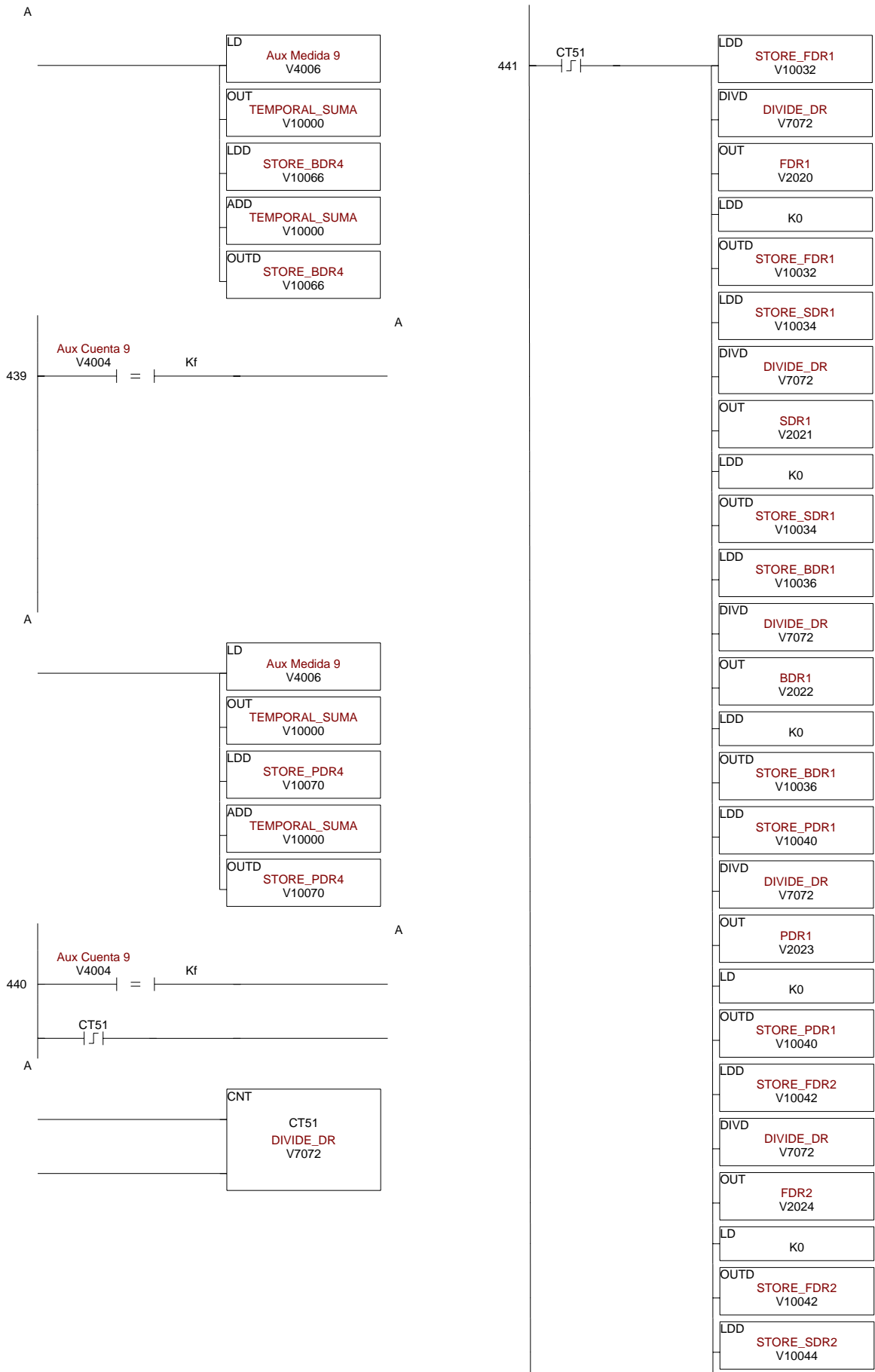












DIVD	DIVIDE_DR V7072
OUT	SDR2 V2025
LD	K0
OUTD	STORE_SDR2 V10044
LDD	STORE_BDR2 V10046
DIVD	DIVIDE_DR V7072
OUT	BDR2 V2026
LD	K0
OUTD	STORE_BDR2 V10046
LDD	STORE_PDR2 V10050
DIVD	DIVIDE_DR V7072
OUT	PDR2 V2027
LD	K0
OUTD	STORE_PDR2 V10050
LDD	STORE_FDR3 V10052
DIVD	DIVIDE_DR V7072
OUT	FDR3 V2030
LD	K0
OUTD	STORE_FDR3 V10052
LDD	STORE_SDR3 V10054
DIVD	DIVIDE_DR V7072
OUT	SDR3 V2031
LD	K0
OUTD	STORE_SDR3 V10054
LDD	STORE_BDR3 V10056
DIVD	DIVIDE_DR V7072
OUT	BDR3 V2032

LD	K0
OUTD	STORE_BDR3 V10056
LDD	STORE_PDR3 V10060
DIVD	DIVIDE_DR V7072
OUT	PDR3 V2033
LD	K0
OUTD	STORE_PDR3 V10060
LDD	STORE_FDR4 V10062
DIVD	DIVIDE_DR V7072
OUT	FDR4 V2034
LD	K0
OUTD	STORE_FDR4 V10062
LDD	STORE_SDR4 V10064
DIVD	DIVIDE_DR V7072
OUT	SDR4 V2035
LD	K0
OUTD	STORE_SDR4 V10064
LDD	STORE_BDR4 V10066
DIVD	DIVIDE_DR V7072
OUT	BDR4 V2036
LD	K0
OUTD	STORE_BDR4 V10066
LDD	STORE_PDR4 V10070
DIVD	DIVIDE_DR V7072
OUT	PDR4 V2037
LD	K0
OUTD	STORE_PDR4 V10070

