



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Escola Superior d'Enginyeries Industrial,
Aeroespacial i Audiovisual de Terrassa

Trabajo Final de Grado

Grado en Ingeniería Electrónica Industrial y Automática

Vizcaíno Cabo, Raúl

ESTUDIO DEL PROCESO DE AUTOMATIZACIÓN DE UNA LÍNEA DE PRODUCCIÓN INDUSTRIAL

Anexo I

Director: Delgado Prieto, Miguel

Codirector: Fernández Sobrino, Ángel

Convocatoria: Enero 2022



ÍNDICE

1. PROGRAMA LÍNEA CAN	6
1.1 Variables del programa.....	6
1.2 Estructuras	10
1.3 Bloques de funciones (FB).....	11
1.3.1 Plataforma	11
1.4 Bloques de programa	14
2. PROGRAMA LÍNEA MODBUS	30
2.1 Declaración de variables.....	30
2.1.1 Variables e instancias globales.....	30
2.1.2 Variables de entradas físicas.....	31
2.1.3 Variables de salidas físicas.....	32
2.1.4 Variables declaradas para el tratamiento de sensores.....	33
2.2 Estructuras	33
2.3 Bloques de funciones (FB).....	35
2.3.1 Control_Plataforma.....	35
2.3.2 PROCESO	39
2.4 Bloques de programa	41

ÍNDICE DE ILUSTRACIONES Y TABLAS

Figura 1 : Variables globales con dirección física.....	6
Figura 3 : Variables con dirección de memoria para uso externo y entradas	7
Figura 4 : Variables globales (1)	8
Figura 6 : Variables globales (2)	9
Figura 7 : Variables derivadas	10
Figura 8 : Estructura de "Bandeja" y "DatosEstado".....	10
Figura 9: Estructura del FB "Plataforma".....	11
Figura 10 : Rutina "Control" del FB "Plataforma" (1).....	12
Figura 11 : Rutina "Control" del FB "Plataforma" (2).....	12
Figura 12 : Rutina "Control" del FB "Plataforma" (3).....	13
Figura 13 : Rutina "Seguimiento" del FB "Plataforma"	13
Figura 14 : Rutina "Setandreset" del FB "Plataforma"	14
Figura 15 : Bloque de programa "ENTRADAS"	14
Figura 16 : Bloque de programa "EMERGENCY"	14
Figura 17 : Bloque de programa "LINEA" (1).....	15
Figura 18 : Bloque de programa "LINEA" (2).....	16
Figura 19 : Bloque de programa "LINEA" (3).....	17
Figura 20 : Bloque de programa "LINEA" (4).....	18
Figura 21 : Bloque de programa "LINEA" (5).....	19
Figura 22 : Bloque de programa "SALIDAS" (1).....	20
Figura 23 : Bloque de programa "SALIDAS" (2).....	20
Figura 24 : Bloque de programa "SALIDAS" (3).....	21
Figura 25 : Bloque de programa "SALIDAS" (4).....	21
Figura 26 : Bloque de programa "INICIO"	21
Figura 27 : Bloque de programa "PT04".....	22
Figura 28 : Bloque de programa "PT05"	23
Figura 29 : Bloque de programa "PT06".....	23
Figura 30 : Bloque de programa "PT15".....	24
Figura 31 : Bloque de programa "PT16".....	24
Figura 32 : Bloque de programa "COMUNICACION_PULMON"	25
Figura 33 : Bloque de programa "COMM_PULMON" (1).....	26
Figura 34 : Bloque de programa "COMM_PULMON" (2).....	26
Figura 35 : Bloque de programa "COMM_PULMON" (3).....	27
Figura 36 : Bloque de programa "COMUNICACION_MODBUS"	27
Figura 37 : Bloque de programa "COMM_MODBUS" (1)	28
Figura 38 : Bloque de programa "COMM_MODBUS" (2)	28
Figura 42 : Orden de ejecución de los bloques de programa de la línea CAN	29
Figura 44 : Variables e instancias globales línea Modbus (1).....	30
Figura 45 : Variables entradas físicas línea Modbus.....	31
Figura 46 : Variables salidas físicas línea Modbus.....	32
Figura 47 : Variables declaradas para el tratamiento de sensores.....	33
Figura 48 : Estructura de "PRODUCTO"	33
Figura 49 : Estructura de "Datos_Estado"	33
Figura 50 : Estructura de "PLATAFORMA"	34
Figura 51 : Estructura del FB "Control_Plataforma" (1).....	35
Figura 52 : Estructura del FB "Control_Plataforma" (2)	36
Figura 53 : Rutina del FB "Control_Plataforma" (1).....	36

Figura 54 : Rutina del FB "Control_Plataforma" (2)	37
Figura 55 : Rutina del FB "Control_Plataforma" (3)	37
Figura 56 : Rutina del FB "Control_Plataforma" (4)	38
Figura 57 : Subrutina "Seguimiento_Producto" del FB "Control_Plataforma"	38
Figura 58 : Subrutina "SetandReset" del FB "Control_Plataforma"	39
Figura 59 : Estructura del FB "PROCESO"	39
Figura 60 : Rutina del FB "PROCESO"	40
Figura 61 : Subrutina "ESTADO_PROD" del FB "PROCESO"	40
Figura 62 : Subrutina "RESULTADO" del FB "PROCESO"	41
Figura 63 : Variables del bloque de programa "PREPROCESADO_SENSORES"	41
Figura 64 : Bloque de programa "PREPROCESADO_SENSORES" (1)	42
Figura 65 : Bloque de programa "PREPROCESADO_SENSORES" (2)	43
Figura 66 : Variables del bloque de programa "EM_REARME"	43
Figura 67 : Bloque de programa "EM_REARME"	43
Figura 68 : Variables del bloque de programa "ESTACIONES_TRABAJO"	44
Figura 69 : Bloque de programa "ESTACIONES_TRABAJO"	44
Figura 70 : Variables del bloque de programa "MOVIMIENTOS_LINEA"	45
Figura 71 : Bloque de programa "MOVIMIENTOS_LINEA" (1)	46
Figura 72 : Bloque de programa "MOVIMIENTOS_LINEA" (2)	47
Figura 73 : Bloque de programa "MOVIMIENTOS_LINEA" (3)	48
Figura 74 : Bloque de programa "MOVIMIENTOS_LINEA" (4)	49
Figura 75 : Bloque de programa "MOVIMIENTOS_LINEA" (5)	50
Figura 76 : Bloque de programa "MOVIMIENTOS_LINEA" (6)	51
Figura 77 : Bloque de programa "MOVIMIENTOS_LINEA" (7)	52
Figura 78 : Bloque de programa "MOVIMIENTOS_LINEA" (8)	53
Figura 79 : Bloque de programa "MOVIMIENTOS_LINEA" (9)	54
Figura 80 : Variables del bloque de programa "COMM_CAN"	54
Figura 81 : Bloque de programa "COMM_CAN"	55
Figura 82 : Variables del bloque de programa "COMM_WR_RD_CAN"	55
Figura 83 : Bloque de programa "COMM_WR_RD_CAN"	55
Figura 84 : Orden de ejecución de los bloques de programas de la línea Modbus.....	56

1. Programa línea CAN

1.1 Variables del programa

Nombre	Tipo	Comentario	Dirección
DIR20_down	BOOL		%QW\3.3\0.0.0.134.0
BAJA_PT16	BOOL		%QW\3.3\0.0.0.133.5
SUBE_PT16	BOOL		%QW\3.3\0.0.0.133.4
SUBE_PT15	BOOL		%QW\3.3\0.0.0.133.3
SUBE_PT04	BOOL		%QW\3.3\0.0.0.133.2
DIR07_down	BOOL		%QW\3.3\0.0.0.133.1
DIR03_down	BOOL		%QW\3.3\0.0.0.133.0
CINTACENTRAL	BOOL		%QW\3.2\0.0.0.134.5
CINTAIZQUIERDA	BOOL		%QW\3.2\0.0.0.134.4
SUBE_PT17	BOOL		%QW\3.2\0.0.0.134.3
SUBE_PT07	BOOL		%QW\3.2\0.0.0.134.2
SUBE_PT06	BOOL		%QW\3.2\0.0.0.134.1
BAJA_PT05	BOOL		%QW\3.2\0.0.0.134.0
SUBE_PT05	BOOL		%QW\3.2\0.0.0.133.5
DIR04_down	BOOL		%QW\3.2\0.0.0.133.4
DIR21_down	BOOL		%QW\3.2\0.0.0.133.3
DIR09_down	BOOL		%QW\3.2\0.0.0.133.2
DIR08_down	BOOL		%QW\3.2\0.0.0.133.1
DIR05_down	BOOL		%QW\3.2\0.0.0.133.0
LUZ_ROJA_BALIZA	EBOOL		%Q0.3.20
M12	EBOOL		%Q0.3.16
M11	EBOOL		%Q0.2.23
M9	EBOOL		%Q0.2.22
M8	EBOOL		%Q0.2.21
M7	EBOOL		%Q0.2.20
M5	EBOOL		%Q0.2.19
M3	EBOOL		%Q0.2.18
M2	EBOOL		%Q0.2.17

Figura 1 : Variables globales con dirección física

Nombre	Tipo	Comentario	Dirección
ComunicacionPulmonLectura	ARRAY[1..30] OF INT		%MW730
ComunicacionPulmonEscritura	ARRAY[1..30] OF INT		%MW700
MODBUS_RD_Buffer	ARRAY[1..10] OF INT		%MW610
MODBUS_WR_Buffer	ARRAY[1..10] OF INT		%MW600
EMERGENCY_LOCK	BOOL	%MW1234	%MW470
DIR03	BOOL	%IW\3.3\0.0.0.169.4	%MW447.4
DIR20	BOOL	%IW\3.3\0.0.0.169.2	%MW447.2
DIB01	BOOL	%IW\3.3\0.0.0.167.5	%MW445.5
DIR07	BOOL	%IW\3.3\0.0.0.167.3	%MW445.3
DI11	BOOL	%IW\3.3\0.0.0.167.2	%MW445.2
DIB06	BOOL	%IW\3.3\0.0.0.167.0	%MW445.0
DIB07	BOOL	%IW\3.2\0.0.0.169.2	%MW169.2
DIR21	BOOL	%IW\3.2\0.0.0.169.1	%MW169.1
DI07	BOOL	%IW\3.2\0.0.0.169.0	%MW169.0
DIR09	BOOL	%IW\3.2\0.0.0.167.5	%MW167.5
DI06	BOOL	%IW\3.2\0.0.0.167.4	%MW167.4
DIR08	BOOL	%IW\3.2\0.0.0.167.3	%MW167.3
DIR05	BOOL	%IW\3.2\0.0.0.167.2	%MW167.2
DI05	BOOL	%IW\3.2\0.0.0.167.1	%MW167.1
DIR04	BOOL	%IW\3.2\0.0.0.167.0	%MW167.0
EMERGENCY_LOCK_PULMON	BOOL		%MW39.6
REARME	EBOOL		%I0.1.6
PARO_EMERGENCIA	EBOOL		%I0.1.4

Figura 2 : Variables con dirección de memoria para uso externo y entradas

Nombre	Tipo	Comentario
TipoProducto_Random	INT	
Sensor_PT17	BOOL	
Sensor_PT16	BOOL	
Sensor_PT15	BOOL	
Sensor_PT07	BOOL	
Sensor_PT06	BOOL	
Sensor_PT05	BOOL	
Sensor_PT04	BOOL	
Sensor_DIR21	BOOL	
Sensor_DIR20	BOOL	
Sensor_DIR09	BOOL	
Sensor_DIR08	BOOL	
Sensor_DIR07	BOOL	
Sensor_DIR05	BOOL	
Sensor_DIR04	BOOL	
Sensor_DIR03_pulso	BOOL	
Sensor_DIR03	BOOL	
RET_PT17	BOOL	
RET_PT16_2	BOOL	
RET_PT16_1	BOOL	
RET_PT15	BOOL	
RET_PT07	BOOL	
RET_PT06	BOOL	
RET_PT05_2	BOOL	
RET_PT05_1	BOOL	
RET_PT04	BOOL	
RET_DIR21	BOOL	
RET_DIR20	BOOL	
RET_DIR09	BOOL	
RET_DIR08	BOOL	
RET_DIR07	BOOL	
RET_DIR05	BOOL	
RET_DIR04	BOOL	
RET_DIR03	BOOL	
Reposo_Pulmon_DIR06	BOOL	Indica reposo del Pulmon y del DIR06 (lectura al PLC Pulmón)
Pulmon_NecesitaAccion	BOOL	
PT15_Reposo_Previo	BOOL	
PT15_Avance_Previo	BOOL	
PT08_Acuse_Previo	BOOL	
PT06_Reposo_Previo	BOOL	
MODBUS_WR_Done	BOOL	
MODBUS_RD_Done	BOOL	

Figura 3 : Variables globales (1)

Nombre	Tipo	Comentario
PR5_Tiempo	TIME	
PR5_Resultado	INT	
PR5_OK	BOOL	
ErrorExcesoPulmon	BOOL	
EMERGENCIA_PULMON	BOOL	Emergencia Pulmon (Lectura del Pulmon)
EM	EBOOL	
DIR06_Acuse_Previo	BOOL	
DIR05_Reposo_Previo	BOOL	
Datos_GEST_2	ARRAY[1..2] OF INT	
Datos_GEST_1	ARRAY[1..4] OF INT	
Datos_GEST_0	ARRAY[1..4] OF INT	
Datos_GEST	ARRAY[1..4] OF INT	
ControlCalidad_Tiempo	TIME	
ControlCalidad_Resultado	INT	
ControlCalidad_OK	BOOL	
ActivarEscrituraPulmon	BOOL	
Activar_Lectura_Pulmon	BOOL	
ActivaMotor_PT17	BOOL	
ActivaMotor_PT16_22	BOOL	
ActivaMotor_PT16_21	BOOL	
ActivaMotor_PT16_11	BOOL	
ActivaMotor_PT15	BOOL	
ActivaMotor_PT07	BOOL	
ActivaMotor_PT06_12	BOOL	
ActivaMotor_PT06_11	BOOL	
ActivaMotor_PT05_22	BOOL	
ActivaMotor_PT05_21	BOOL	
ActivaMotor_PT05_11	BOOL	
ActivaMotor_PT04	BOOL	
ActivaMotor_DIR21	BOOL	
ActivaMotor_DIR20	BOOL	
ActivaMotor_DIR09	BOOL	
ActivaMotor_DIR08_12	BOOL	
ActivaMotor_DIR08_11	BOOL	
ActivaMotor_DIR07_12	BOOL	
ActivaMotor_DIR07_11	BOOL	
ActivaMotor_DIR06	BOOL	
ActivaMotor_DIR05	BOOL	
ActivaMotor_DIR04	BOOL	
ActivaMotor_DIR03	BOOL	

Figura 4 : Variables globales (2)

Nombre	Tipo
+ Dirección_Comunicacion_Pulmon	ADDM_TYPE
+ Dirección_Comunicacion	ADDM_TYPE
+ DatosEstado_PT04	DatosEstado
+ DatosEstado_DIR03	DatosEstado
+ Datos_PT17	DatosEstado
+ Datos_PT16	DatosEstado
+ Datos_PT15	DatosEstado
+ Datos_PT07	DatosEstado
+ Datos_PT06	DatosEstado
+ Datos_PT05	DatosEstado
+ Datos_PT04	DatosEstado
+ Datos_DIR21	DatosEstado
+ Datos_DIR20	DatosEstado
+ Datos_DIR09	DatosEstado
+ Datos_DIR08	DatosEstado
+ Datos_DIR07	DatosEstado
+ Datos_DIR05	DatosEstado
+ Datos_DIR04	DatosEstado
+ Datos_DIR03	DatosEstado
+ BandejaVacía2	Bandeja
+ BandejaVacía	Bandeja
+ Bandeja_PT08_Previa	Bandeja
+ Bandeja_DIR06_Previa	Bandeja

Figura 5 : Variables derivadas

1.2 Estructuras

Nombre	Tipo
Bandeja	<Estruct.>
IDPRODUCTO	INT
TIPOPRODUCTO	INT
ESTADO	INT
DatosEstado	<Estruct.>
BandejaEntrada	Bandeja
IDPRODUCTO	INT
TIPOPRODUCTO	INT
ESTADO	INT
BandejaSalida1	Bandeja
IDPRODUCTO	INT
TIPOPRODUCTO	INT
ESTADO	INT
BandejaSalida2	Bandeja
IDPRODUCTO	INT
TIPOPRODUCTO	INT
ESTADO	INT
Estado	INT

Figura 6 : Estructura de "Bandeja" y "DatosEstado"

1.3 Bloques de funciones (FB)

1.3.1 Plataforma

Nombre	Nº	Comentario	
Plataforma	<DFB>		
<entradas>			
ReposoDestino1	1	BOOL	Estado de Reposo del Destino en la dirección 1
ReposoDestino2	2	BOOL	Estado de Reposo del Destino en la dirección 2
AvanceAux1	3	BOOL	Evitar el avance simultaneo hacia una plataforma con dos entradas, conectar al estado de avance del otro acceso a la plataforma
AvanceAux2	4	BOOL	Evitar el avance simultaneo hacia una plataforma con dos entradas, conectar al estado de avance del otro acceso a la plataforma
PresenciaBandeja	5	BOOL	Sensor de la sección correspondiente que indicará si hay una bandeja
BandejaVacía	6	Bandeja	Variable para asignar el estado de la Bandeja de Entrada a 0
Tiempo_ACK	10	TIME	Tiempo a esperar el ack cuando el sensor detecta (una vez sobrepasado se activará el avance aun y no haber obtenido un ack)
Emergencia	11	BOOL	Emergencia del sistema
Rearme	12	BOOL	Para poner a zero todos los valores internos y volver al estado de reposo
Especial1	13	BOOL	Bit para las plataformas que envían una bandeja lateralmente en el destino 1
Especial2	14	BOOL	Bit para las plataformas que envían una bandeja lateralmente en el destino 2
<salidas>			
Retenedor1	1	BOOL	Retenedor o Plataforma a activar durante el avance al destino 1
Retenedor2	2	BOOL	Retenedor o Plataforma a activar durante el avance al destino 2
Motor1_1	3	BOOL	Activación del motor en Avance 1
Motor1_2	4	BOOL	Activación del motor en Avance 1
Motor2_1	5	BOOL	Activación del motor en Avance 2
Motor2_2	6	BOOL	Activación del motor en Avance 2
<entradas/salidas>			
DatosEstado	7	Datos...	Variable sobre la cual se van a reescribir ciertos datos internos para su accesibilidad des del exterior del bloque
BandejaDestino1	8	Bandeja	Bandeja a enviar al destino 1
BandejaDestino2	9	Bandeja	Bandeja a enviar al destino 2
ConfirmacionEnvioInfo1	15	BOOL	Bit para confirmar al Destino 1 que ya se le ha hecho la transmisión de datos acerca de la bandeja
ConfirmacionEnvioInfo2	16	BOOL	Bit para confirmar al Destino 2 que ya se le ha hecho la transmisión de datos acerca de la bandeja
<público>			
BandejaEntrada		Bandeja	Variable sobre la cual el retenedor sobrescribirá los valores de la bandeja que le envíe
BandejaSalida1		Bandeja	Datos de la Bandeja que iran a la Salida 1
BandejaSalida2		Bandeja	Datos de la Bandeja que iran a la Salida 2
Reposo		BOOL	Estado base de la Plataforma
Peticion		BOOL	Estado cuando se tiene una bandeja y se espera poder enviarla a la siguiente plataforma
Avance1		BOOL	Movimiento en la dirección 1
Avance2		BOOL	Movimiento en la dirección 2
Bloqueo1		BOOL	Variable para bloqueo del Avance 1
Bloqueo2		BOOL	Variable para bloqueo del Avance 2
AcuseEnvioInfo		BOOL	Bit para la confirmación de que se ha recibido la información de la bandeja
<privado>			
FReposo		BOOL	
FAvance1		BOOL	
FAvance2		BOOL	
FILTRO_SENSOR		TON	
FBI_0		TOF	
FBI_1		TOF	
Bloqueo_ACK_temp		BOOL	
FILTRO_SENSOR_0		TON	
<secciones>			
Control		<LD>	
Seguimiento		<ST>	
Almacenaje		<ST>	
Setandreset		<ST>	

Figura 7: Estructura del FB "Plataforma"

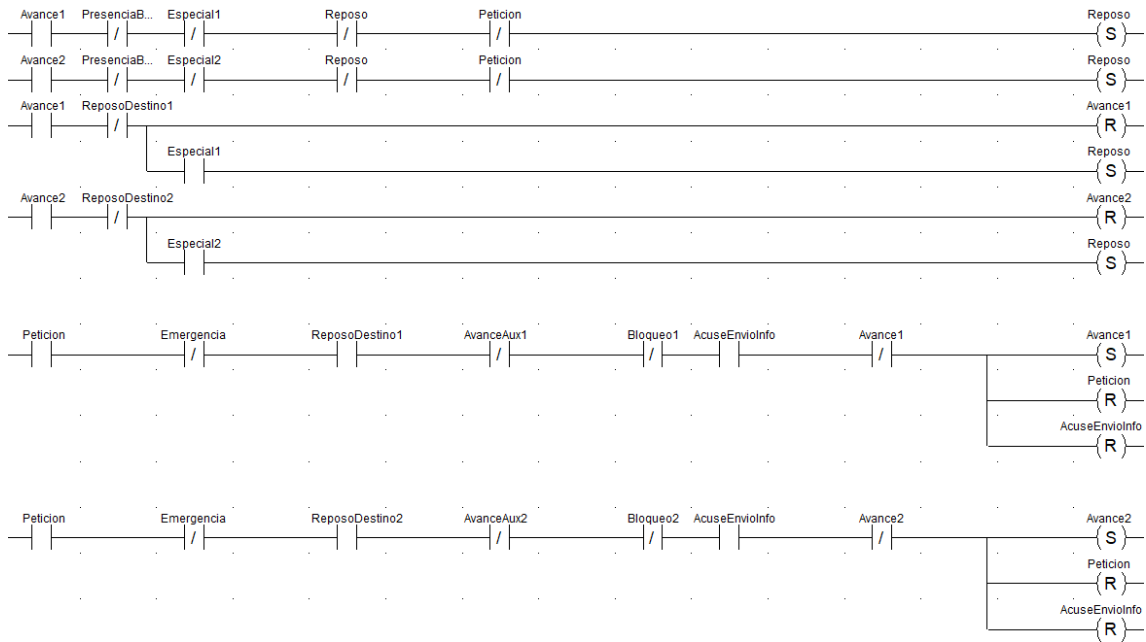


Figura 8 : Rutina "Control" del FB "Plataforma" (1)

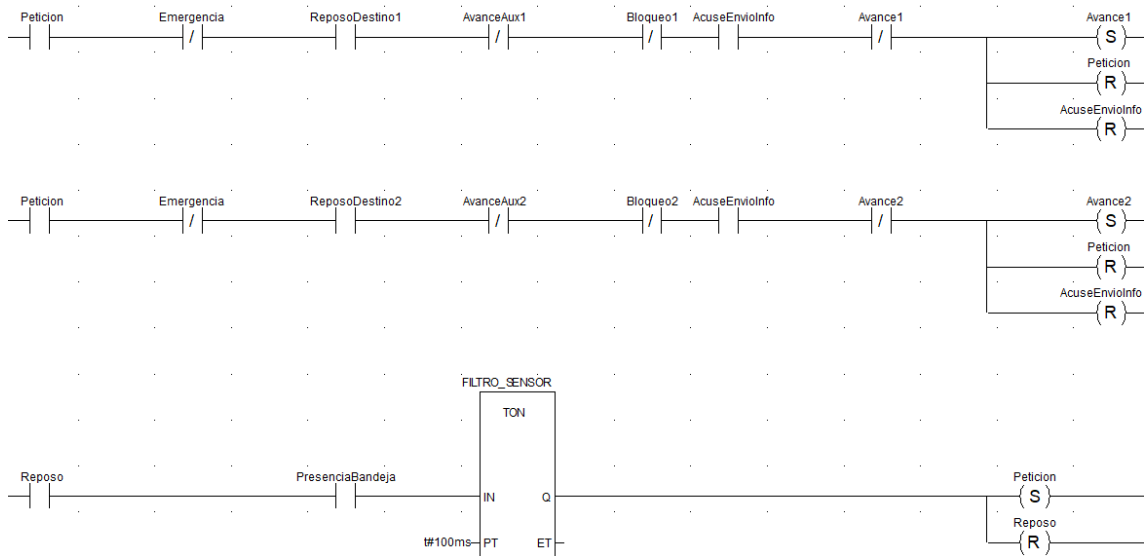


Figura 9 : Rutina "Control" del FB "Plataforma" (2)

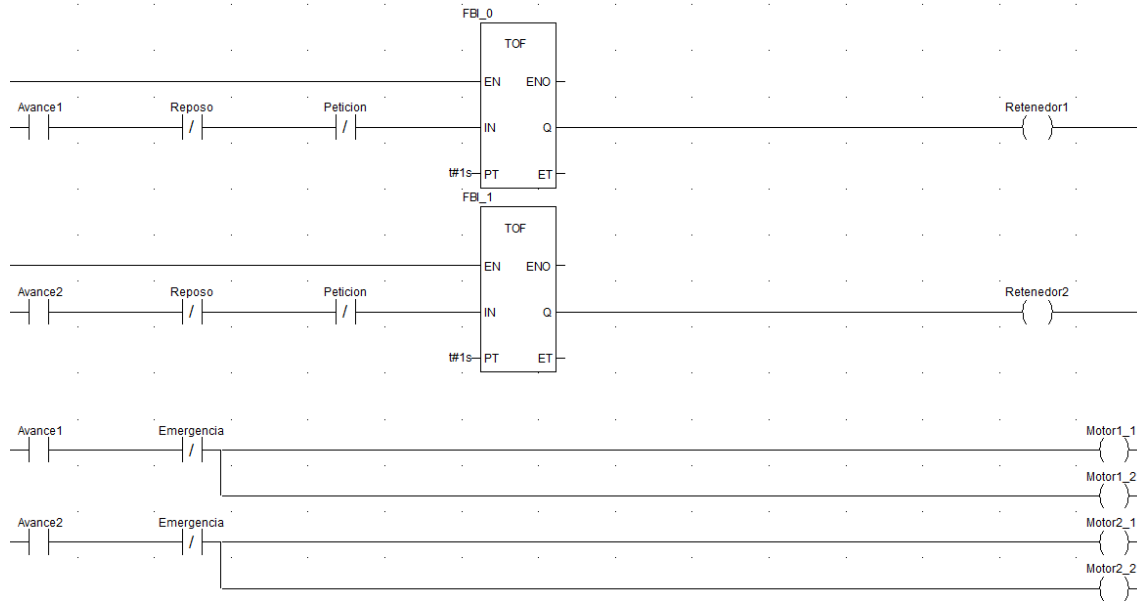


Figura 10 : Rutina "Control" del FB "Plataforma" (3)

```

IF Avance1 AND NOT FAvance1 THEN
    BandejaSalida1 := BandejaEntrada;
END_IF;

IF Avance2 AND NOT FAvance2 THEN
    BandejaSalida2 := BandejaEntrada;
END_IF;

IF FAvance1 AND NOT Avance1 THEN
    BandejaDestino1 := BandejaSalida1;
    ConfirmacionEnvioInfo1 := TRUE;
END_IF;

IF FAvance2 AND NOT Avance2 THEN
    BandejaDestino2 := BandejaSalida2;
    ConfirmacionEnvioInfo2 := TRUE;
END_IF;

IF Reposo AND NOT FReposo THEN
    BandejaEntrada := BandejaVacía;
END_IF;

FAvance1 := Avance1;
FAvance2 := Avance2;
FReposo := Reposo;

```

Figura 11 : Rutina "Seguimiento" del FB "Plataforma"

```

IF %S13 OR Rearme THEN
    Reposo := TRUE;
    Peticion := FALSE;
    Avance1 := FALSE;
    Avance2 := FALSE;
    Bloqueo1 := FALSE;
    Bloqueo2 := FALSE;
END_IF;

IF Rearme THEN
    BandejaEntrada := BandejaVacía;
END_IF;

```

Figura 12 : Rutina "Setandreset" del FB "Plataforma"

1.4 Bloques de programa

```

Sensor_DIR03 := NOT DIR03;
Sensor_PT04 := DIB01;
Sensor_DIR04 := DIR04;
Sensor_PT05 := DI05;
Sensor_DIR08 := DIR08;
Sensor_PT17 := NOT DIB07;
Sensor_DIR21 := DIR21;
Sensor_PT07 := DI07;
Sensor_DIR05 := DIR05;

Sensor_PT06 := DI06;
Sensor_DIR09 := DIR09;
Sensor_PT15 := DIB06;
Sensor_DIR20 := NOT DIR20;
Sensor_PT16 := NOT DI11;
Sensor_DIR07 := NOT DIR07;

```

Figura 13 : Bloque de programa "ENTRADAS"

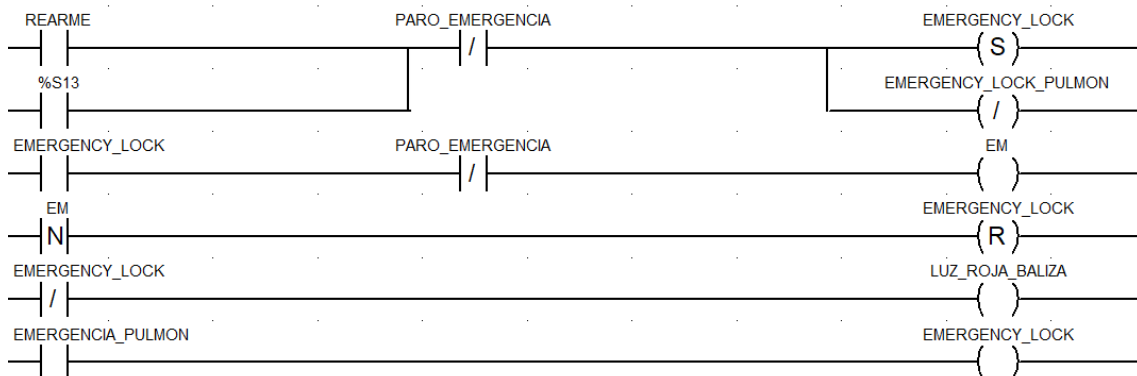


Figura 14 : Bloque de programa "EMERGENCY"

```

DIR03_FB (ReposoDestino1 := PT04_FB.Reposo,
AvanceAux1 := DIR07_FB.Avancel,
PresenciaBandeja := Sensor_DIR03,
BandejaVacía := BandejaVacía,
BandejaDestino1 := PT04_FB.BandejaEntrada,
Emergencia := EMERGENCY_LOCK,
Rearme := REARME,
ConfirmacionEnvioInfo1 := PT04_FB.AcuseEnvioInfo,
Retenedor1 => RET_DIR03,
Motor1_1 => ActivaMotor_DIR03);

PT04_FB (ReposoDestino1 := DIR04_FB.Reposo,
PresenciaBandeja := Sensor_PT04,
BandejaVacía := BandejaVacía,
BandejaDestino1 := DIR04_FB.BandejaEntrada,
Emergencia := EMERGENCY_LOCK,
Rearme := REARME,
ConfirmacionEnvioInfo1 := DIR04_FB.AcuseEnvioInfo,
Retenedor1 => RET_PT04,
Motor1_1 => ActivaMotor_PT04);

DIR04_FB (ReposoDestino1 := PT05_FB.Reposo,
PresenciaBandeja := Sensor_DIR04,
BandejaVacía := BandejaVacía,
BandejaDestino1 := PT05_FB.BandejaEntrada,
Emergencia := EMERGENCY_LOCK,
Rearme := REARME,
ConfirmacionEnvioInfo1 := PT05_FB.AcuseEnvioInfo,
Retenedor1 => RET_DIR04,
Motor1_1 => ActivaMotor_DIR04);

```

Figura 15 : Bloque de programa "LINEA" (1)

```

PT05_FB (ReposoDestino1 := DIR05_FB.Reposo,
ReposoDestino2 := DIR08_FB.Reposo,
PresenciaBandeja := Sensor_PT05,
BandejaVacía := BandejaVacía,
BandejaDestino1 := DIR05_FB.BandejaEntrada,
BandejaDestino2 := DIR08_FB.BandejaEntrada,
Emergencia := EMERGENCY_LOCK,
Rearme := REARME,
Especial1 := TRUE,
ConfirmacionEnvioInfo1 := DIR05_FB.AcuseEnvioInfo,
ConfirmacionEnvioInfo2 := DIR08_FB.AcuseEnvioInfo,
Retenedor1 => RET_PT05_1,
Retenedor2 => RET_PT05_2,
Motor1_1 => ActivaMotor_PT05_11,
Motor2_1 => ActivaMotor_PT05_21,
Motor2_2 => ActivaMotor_PT05_22);

DIR05_FB (ReposoDestino1 := Reposo_Pulmon_DIR06,
PresenciaBandeja := Sensor_DIR05,
BandejaVacía := BandejaVacía,
BandejaDestino1 := DIR06_FB.BandejaEntrada,
Emergencia := EMERGENCY_LOCK,
Rearme := REARME,
ConfirmacionEnvioInfo1 := DIR06_FB.AcuseEnvioInfo,
Retenedor1 => RET_DIR05,
Motor1_1 => ActivaMotor_DIR05);

PT06_FB (ReposoDestino1 := DIR09_FB.Reposo,
PresenciaBandeja := Sensor_DIR09,
BandejaVacía := BandejaVacía,
BandejaDestino1 := DIR09_FB.BandejaEntrada,
Emergencia := EMERGENCY_LOCK,
Rearme := REARME,
Especial1 := TRUE,
ConfirmacionEnvioInfo1 := DIR09_FB.AcuseEnvioInfo,
Retenedor1 => RET_PT06,
Motor1_1 => ActivaMotor_PT06_11,
Motor1_2 => ActivaMotor_PT06_12);

```

Figura 16 : Bloque de programa "LINEA" (2)


```

DIR07_FB (ReposoDestino1 := PT04_FB.Reposo,
AvanceAux1 := DIR03_FB.Avancel,
PresenciaBandeja := Sensor_DIR07,
BandejaVacía := BandejaVacía,
BandejaDestino1 := PT04_FB.BandejaEntrada,
Emergencia := EMERGENCY_LOCK,
Rearme := REARME,
ConfirmacionEnvioInfo1 := PT04_FB.AcuseEnvioInfo,
Retenedor1 => RET_DIR07,
Motor1_1 => ActivaMotor_DIR07_11,
Motor1_2 => ActivaMotor_DIR07_12);

DIR08_FB (ReposoDestino1 := PT17_FB.Reposo,
AvanceAux1 := PT16_FB.Avancel,
PresenciaBandeja := Sensor_DIR08,
BandejaVacía := BandejaVacía,
BandejaDestino1 := PT17_FB.BandejaEntrada,
Emergencia := EMERGENCY_LOCK,
Rearme := REARME,
ConfirmacionEnvioInfo1 := PT17_FB.AcuseEnvioInfo,
Retenedor1 => RET_DIR08,
Motor1_1 => ActivaMotor_DIR08_11,
Motor1_2 => ActivaMotor_DIR08_12);

DIR09_FB (ReposoDestino1 := PT07_FB.Reposo,
AvanceAux1 := DIR21_FB.Avancel,
PresenciaBandeja := Sensor_DIR09,
BandejaVacía := BandejaVacía,
BandejaDestino1 := PT07_FB.BandejaEntrada,
Emergencia := EMERGENCY_LOCK,
Rearme := REARME,
Especial1 := TRUE,
ConfirmacionEnvioInfo1 := PT07_FB.AcuseEnvioInfo,
Retenedor1 => RET_DIR09,
Motor1_1 => ActivaMotor_DIR09);

```

Figura 17: Bloque de programa "LINEA" (3)

```

PT07_FB (ReposoDestino1 := PT08_FB.Reposo,
         PresenciaBandeja := Sensor_PT07,
         BandejaVacía := BandejaVacía,
         BandejaDestino1 := PT08_FB.BandejaEntrada,
         Emergencia := EMERGENCY_LOCK,
         Rearme := REARME,
         Especial1 := TRUE,
         ConfirmacionEnvioInfo1 := PT08_FB.AcuseEnvioInfo,
         Retenedor1 => RET_PT07,
         Motor1_1 => ActivaMotor_PT07);

PT15_FB (ReposoDestino1 := DIR20_FB.Reposo,
         PresenciaBandeja := Sensor_PT15,
         BandejaVacía := BandejaVacía,
         BandejaDestino1 := DIR20_FB.BandejaEntrada,
         Emergencia := EMERGENCY_LOCK,
         Rearme := REARME,
         ConfirmacionEnvioInfo1 := DIR20_FB.AcuseEnvioInfo,
         Retenedor1 => RET_PT15,
         Motor1_1 => ActivaMotor_PT15);

DIR20_FB (ReposoDestino1 := PT16_FB.Reposo,
         PresenciaBandeja := Sensor_DIR20,
         BandejaVacía := BandejaVacía,
         BandejaDestino1 := PT16_FB.BandejaEntrada,
         Emergencia := EMERGENCY_LOCK,
         Rearme := REARME,
         ConfirmacionEnvioInfo1 := PT16_FB.AcuseEnvioInfo,
         Retenedor1 => RET_DIR20,
         Motor1_1 => ActivaMotor_DIR20);

```

Figura 18: Bloque de programa "LINEA" (4)

```

PT16_FB (ReposoDestino1 := PT17_FB.Reposo,
         ReposoDestino2 := DIR07_FB.Reposo,
         AvanceAux1 := DIR08_FB.Avancel,
         PresenciaBandeja := Sensor_PT16,
         BandejaVacía := BandejaVacía,
         BandejaDestino1 := PT17_FB.BandejaEntrada,
         BandejaDestino2 := DIR07_FB.BandejaEntrada,
         Emergencia := EMERGENCY_LOCK,
         Rearme := REARME,
         Especial2 := TRUE,
         ConfirmacionEnvioInfo1 := PT17_FB.AcuseEnvioInfo,
         ConfirmacionEnvioInfo2 := DIR07_FB.AcuseEnvioInfo,
         Retenedor1 => RET_PT16_1,
         Retenedor2 => RET_PT16_2,
         Motor1_1 => ActivaMotor_PT16_11,
         Motor2_1 => ActivaMotor_PT16_21,
         Motor2_2 => ActivaMotor_PT16_22);

PT17_FB (ReposoDestino1 := DIR21_FB.Reposo,
         PresenciaBandeja := Sensor_PT17,
         BandejaVacía := BandejaVacía,
         BandejaDestino1 := DIR21_FB.BandejaEntrada,
         Emergencia := EMERGENCY_LOCK,
         Rearme := REARME,
         ConfirmacionEnvioInfo1 := DIR21_FB.AcuseEnvioInfo,
         Retenedor1 => RET_PT17,
         Motor1_1 => ActivaMotor_PT17);

DIR21_FB (ReposoDestino1 := PT07_FB.Reposo,
         AvanceAux1 := DIR09_FB.Avancel,
         PresenciaBandeja := Sensor_DIR21,
         BandejaVacía := BandejaVacía,
         BandejaDestino1 := PT07_FB.BandejaEntrada,
         Emergencia := EMERGENCY_LOCK,
         Rearme := REARME,
         ConfirmacionEnvioInfo1 := PT07_FB.AcuseEnvioInfo,
         Retenedor1 => RET_DIR21,
         Motor1_1 => ActivaMotor_DIR21);

```

Figura 19: Bloque de programa "LINEA" (5)



Figura 20 : Bloque de programa "SALIDAS" (1)



Figura 21 : Bloque de programa "SALIDAS" (2)



Figura 22 : Bloque de programa "SALIDAS" (3)

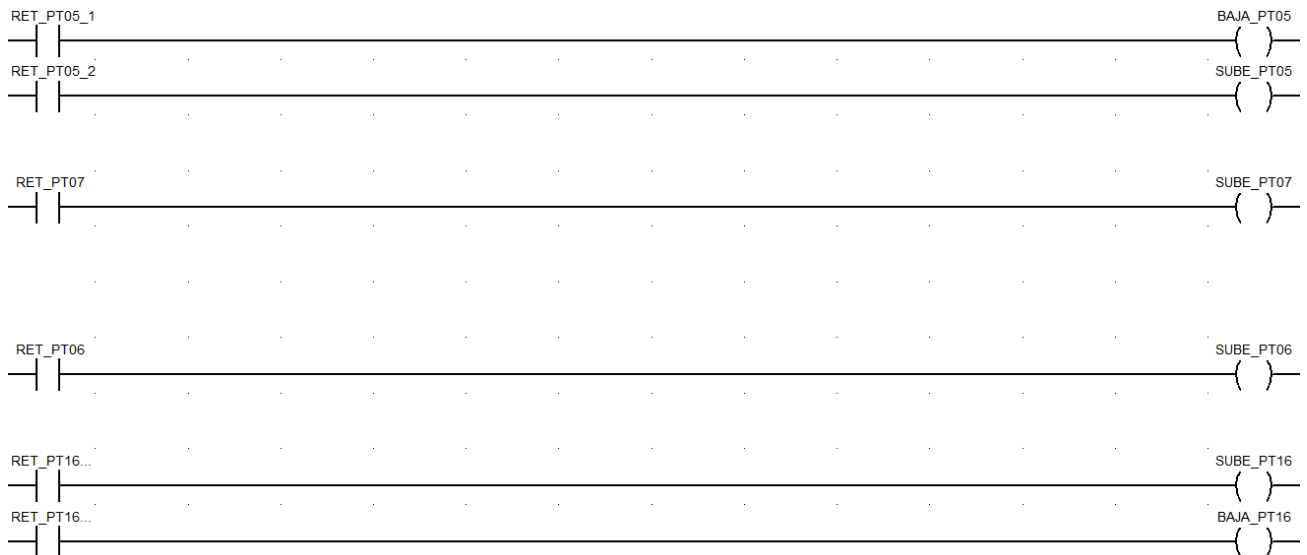


Figura 23 : Bloque de programa "SALIDAS" (4)

```

R_TRIG_0 (CLK := Sensor_DIR03,
          Q => Sensor_DIR03_pulso);

IF Sensor_DIR03_pulso THEN

    RAND_0 (TB := TRUE,
           HL := 3,
           LL := 1,
           OY_INT => TipoProducto_Random);

    DIR03_FB.BandejaEntrada.IDPRODUCTO := DIR03_FB.BandejaEntrada.IDPRODUCTO + 1;
    DIR03_FB.BandejaEntrada.TIPOPRODUCTO := TipoProducto_Random;
    DIR03_FB.BandejaEntrada.ESTADO := 0;

    DIR03_FB.AcuseEnvioInfo := TRUE;

END IF;

```

Figura 24 : Bloque de programa "INICIO"

```
(* CONTROL CALIDAD

Hay una probabilidad del 25% de que el producto no pase el control de calidad.
Si ControlCalidad_Resultado <= 3, el producto debe volver a pasar de nuevo.
Si ControlCalidad_Resultado = 1, el producto debe pasar por control de calidad.

0 = Sin procesar
1, 2 = No acabado
3 = Acabado
4 = Defectuoso -> Avanza por DIR04 para retirar en PT06
5 = Procesado correctamente -> Avanza por DIR04 para retirar en PT05
*)

IF PT04_FB.BandejaEntrada.ESTADC = 3 THEN

    PT04_FB.Bloqueo1 := TRUE;

    TON_ControlCalidad (IN := DIB01,
                        PT := t#15s,
                        Q => ControlCalidad_OK,
                        ET => ControlCalidad_Tiempo);

END_IF;

IF ControlCalidad_OK THEN

    RAND_1 (TB := TRUE,
           HL := 4,
           LL := 1,
           OY_INT => ControlCalidad_Resultado);

    IF ControlCalidad_Resultado = 1 THEN
        PT04_FB.BandejaEntrada.ESTADC := 4;
    ELSE
        PT04_FB.BandejaEntrada.ESTADC := 5;
    END_IF;

    PT04_FB.Bloqueo1 := FALSE;

END_IF;
```

Figura 25 : Bloque de programa "PT04"

```
(* RECOGIDA DEL PRODUCTO *)

(* 0 = Sin procesar -> Avanza por RET5 o RET8 según disponibilidad
1, 2 = No acabado -> Pasar por PR5, avanza por RET5 o RET8 según disponibilidad
3 = Acabado
4 = Defectuoso -> Avanza por DIR05 para retirar en PT06
5 = Procesado correctamente
*)

IF PT05_FB.BandejaEntrada.ESTADC = 5 THEN

    PT05_FB.Bloqueo1:= TRUE;
    PT05_FB.Bloqueo2 := TRUE;

ELSE

    IF Sensor_DIR08 OR PT05_FB.BandejaEntrada.ESTADC = 4 THEN
        PT05_FB.Bloqueo2 := TRUE;
        PT05_FB.Bloqueo1 := FALSE;

    ELSE
        PT05_FB.Bloqueo1 := TRUE;
        PT05_FB.Bloqueo2 := FALSE;
    END_IF;

END_IF;

IF NOT Sensor_PT05 THEN

    PT05_FB.Bloqueo1:= FALSE;
    PT05_FB.Bloqueo2 := FALSE;

END_IF;
```

Figura 26 : Bloque de programa "PT05"

```
(* RECOGIDA DEL PRODUCTO DEFECTUOSO

0 = Sin procesar
1, 2 = No acabado
4 = Defectuoso -> Retirar
*)

IF PT06_FB.BandejaEntrada.ESTADC = 4 THEN

    PT06_FB.Bloqueo1:= TRUE;

ELSE IF PT06_FB.BandejaEntrada.ESTADC < 4 OR NOT Sensor_PT06 THEN

    PT06_FB.Bloqueo1:= FALSE;

END_IF;
END_IF;
```

Figura 27 : Bloque de programa "PT06"

```
(* ESTACIÓN DE TRABAJO 5 (P5) - ACABADO

El PR5 consiste en el acabado de cualquier tipo de producto.
Hay una probabilidad del 33% de que el acabado sea NO satisfactorio.
Si PR5_Resultado <= 2, el producto debe volver a pasar de nuevo.
Si PR5_Resultado = 3, el producto debe pasar por control de calidad.

1, 2 = No acabado
3 = Acabado
*)

IF PT15_FB.BandejaEntrada.ESTADC = 2 THEN

    PT15_FB.Bloqueo1 := TRUE;

    TON_PR5 (IN := DIB06,
             PT := t#10s,
             Q => PR5_OK,
             ET => PR5_Tiempo);

END_IF;

IF PR5_OK THEN

    RAND_1 (TB := TRUE,
           HL := 3,
           LL := 1,
           OY_INT => PR5_Resultado);

    IF PR5_Resultado <> 1 THEN
        PT15_FB.BandejaEntrada.ESTADC := 3;
    END_IF;

    PT15_FB.Bloqueo1 := FALSE;

END_IF;
```

Figura 28 : Bloque de programa "PT15"

```
(* CONTROL DE PIEZAS ACABADAS

0, 1, 2 = No acabado -> Avanzar hacia PT17
3 = Acabado -> Avanzar por DIR07
*)

IF PT16_FB.BandejaEntrada.ESTADC < 3 THEN

    PT16_FB.Bloqueo1:= FALSE;
    PT16_FB.Bloqueo2:= TRUE;

ELSE

    PT16_FB.Bloqueo1:= TRUE;
    PT16_FB.Bloqueo2:= FALSE;

END_IF;
```

Figura 29 : Bloque de programa "PT16"


```
(*COMUNICACIONES CON EL PULMÓN, QUE SOLO FUNCIONARÁ COMO UN RETENEDOR MÁS*)

(*PULMON -> CAN
Esta parte se debería hacer des del Pulmón, pero debido a problemas de comunicaciones
se hace directamente des de la CAN
*)
Pulmon_NecesitaAccion      := ComunicacionPulmonLectura[1].0;
ActivaMotor_DIR06         := ComunicacionPulmonLectura[1].1;
Reposo_Pulmon_DIR06      := ComunicacionPulmonLectura[1].2;
PT06_FB.AcuseEnvioInfo   := ComunicacionPulmonLectura[1].3;
ErrorExcesoPulmon       := ComunicacionPulmonLectura[1].4;
PT06_FB.BandejaEntrada.IDPRODUCTO      := ComunicacionPulmonLectura[2];
PT06_FB.BandejaEntrada.TIPOPRODUCTO   := ComunicacionPulmonLectura[3];
PT06_FB.BandejaEntrada.ESTADO         := ComunicacionPulmonLectura[4];

(*CAN -> PULMON*)

IF PT06_Reposo_Previo <> PT06_FB.Reposo
OR DIR06_Acuse_Previo <> DIR06_FB.AcuseEnvioInfo OR DIR05_Reposo_Previo <> DIR05_FB.Reposo
OR Bandeja_DIR06_Previa.IDPRODUCTO <> DIR06_FB.BandejaEntrada.IDPRODUCTO
OR Bandeja_DIR06_Previa.TIPOPRODUCTO <> DIR06_FB.BandejaEntrada.TIPOPRODUCTO
OR Bandeja_DIR06_Previa.ESTADO <> DIR06_FB.BandejaEntrada.ESTADO THEN

    ComunicacionPulmonEscritura[1].0 := PT06_FB.Reposo;
    ComunicacionPulmonEscritura[1].4 := DIR06_FB.AcuseEnvioInfo;
    ComunicacionPulmonEscritura[1].5 := DIR05_FB.Reposo;
    ComunicacionPulmonEscritura[2]  := DIR06_FB.BandejaEntrada.IDPRODUCTO;
    ComunicacionPulmonEscritura[3]  := DIR06_FB.BandejaEntrada.TIPOPRODUCTO;
    ComunicacionPulmonEscritura[4]  := DIR06_FB.BandejaEntrada.ESTADO;
    ActivarEscrituraPulmon := TRUE;

END_IF;

Bandeja_DIR06_Previa      := DIR06_FB.BandejaEntrada;
DIR05_Reposo_Previo      := DIR05_FB.Reposo;
DIR06_Acuse_Previo       := DIR06_FB.AcuseEnvioInfo;
PT06_Reposo_Previo       := PT06_FB.Reposo;
```

Figura 30 : Bloque de programa "COMUNICACION_PULMON"

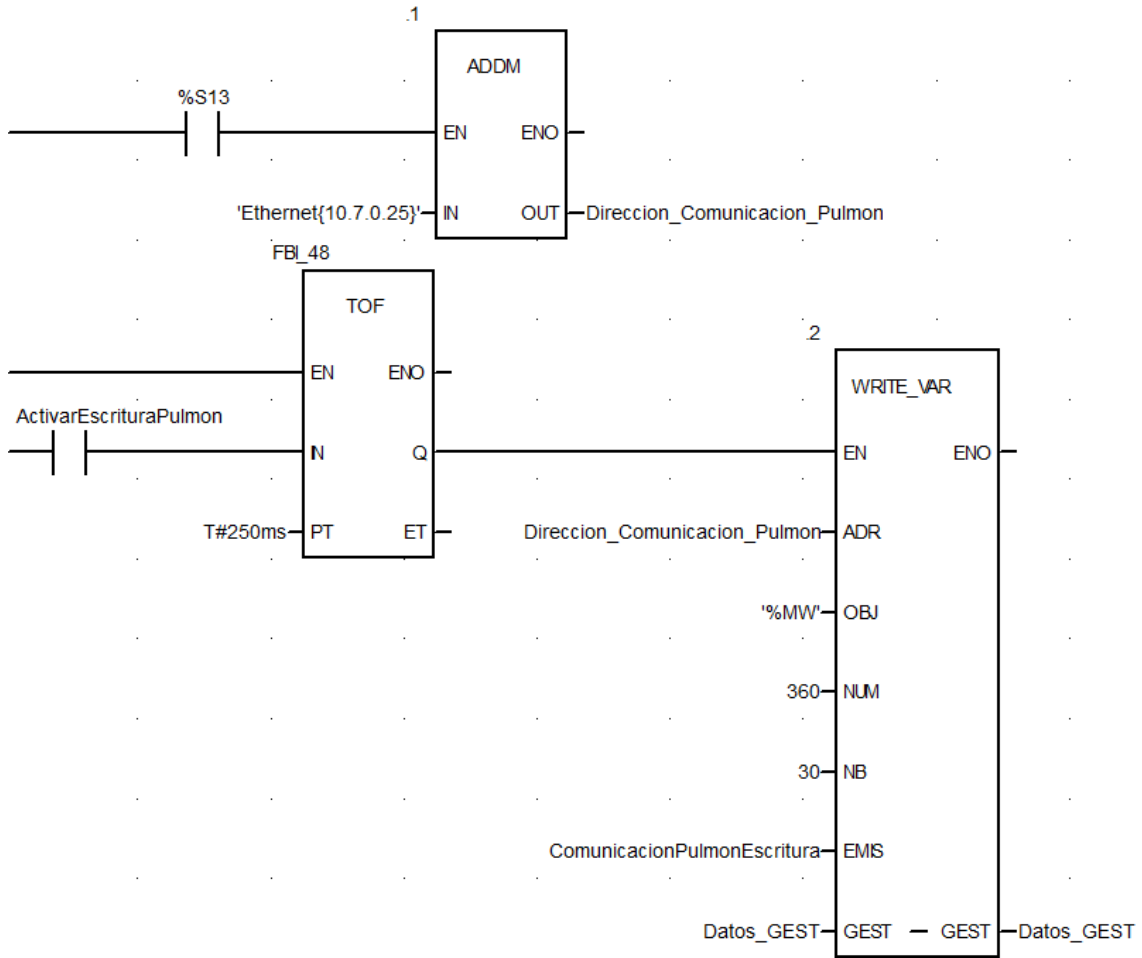


Figura 31 : Bloque de programa "COMM_PULMON" (1)

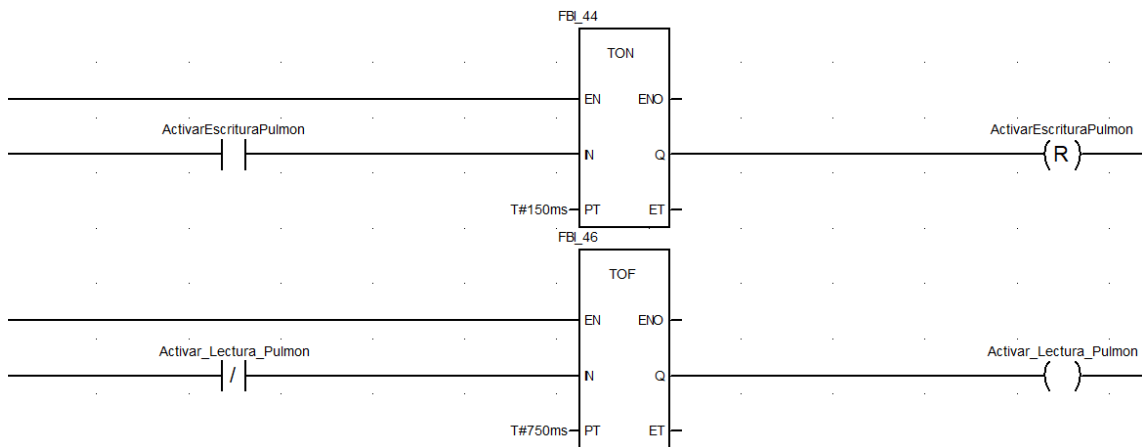


Figura 32 : Bloque de programa "COMM_PULMON" (2)

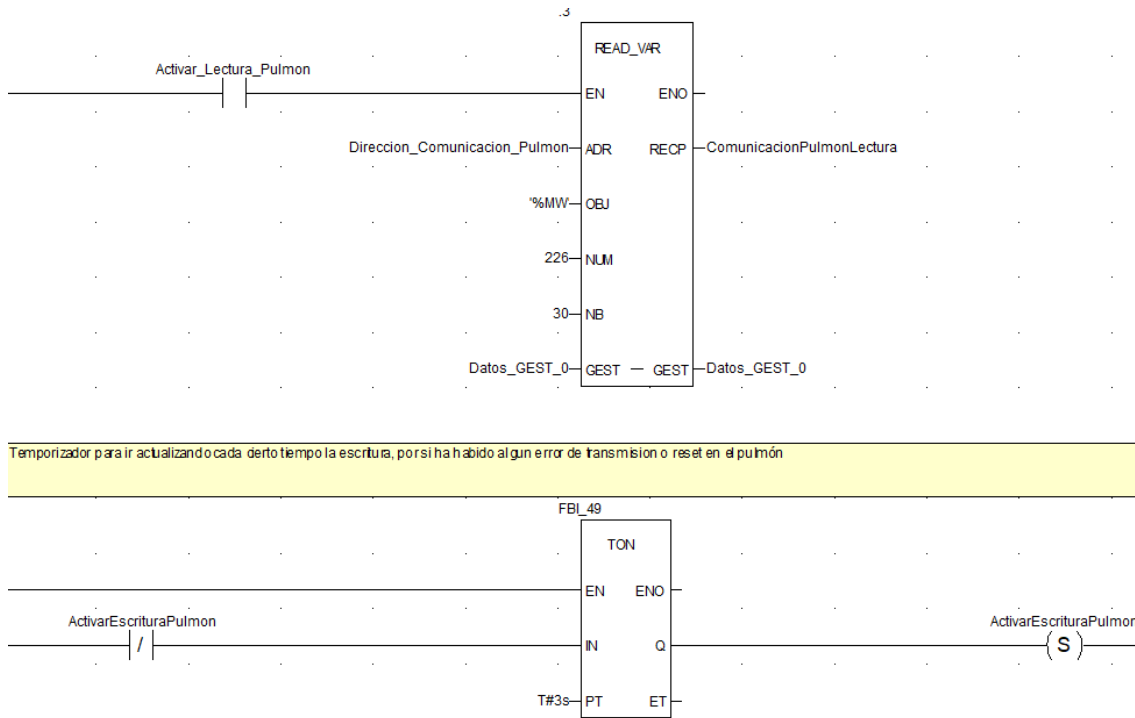


Figura 33 : Bloque de programa "COMM_PULMON" (3)

```

PT08_FB.Reposo                := MODBUS_RD_Buffer[1].0;
PT08_FB.Avance1               := MODBUS_RD_Buffer[1].1;
PT15_FB.AcuseEnvioInfo        := MODBUS_RD_Buffer[1].2;
PT15_FB.BandejaEntrada.IDPRODUCTO := MODBUS_RD_Buffer[2];
PT15_FB.BandejaEntrada.TIPOPDUCTO := MODBUS_RD_Buffer[3];
PT15_FB.BandejaEntrada.ESTADO  := MODBUS_RD_Buffer[4];
MODBUS_RD_Done := TRUE;

IF Bandeja_PT08_Previa.ESTADO <> PT08_FB.BandejaEntrada.ESTADO
OR Bandeja_PT08_Previa.IDPRODUCTO <> PT08_FB.BandejaEntrada.IDPRODUCTO
OR Bandeja_PT08_Previa.TIPOPDUCTO <> PT08_FB.BandejaEntrada.TIPOPDUCTO
OR PT08_Acuse_Previo <> PT08_FB.AcuseEnvioInfo
OR PT15_Reposo_Previo <> PT15_FB.Reposo
OR PT15_Avance_Previo <> PT15_FB.Avance1 THEN

    MODBUS_WR_Buffer[1].0 := PT15_FB.Reposo;
    MODBUS_WR_Buffer[1].1 := PT15_FB.Avance1;
    MODBUS_WR_Buffer[1].2 := PT08_FB.AcuseEnvioInfo;
    MODBUS_WR_Buffer[2] := PT08_FB.BandejaEntrada.IDPRODUCTO;
    MODBUS_WR_Buffer[3] := PT08_FB.BandejaEntrada.TIPOPDUCTO;
    MODBUS_WR_Buffer[4] := PT08_FB.BandejaEntrada.ESTADO;
    MODBUS_WR_Done := TRUE;

END_IF;

PT08_Acuse_Previo := PT08_FB.AcuseEnvioInfo;
Bandeja_PT08_Previa := PT08_FB.BandejaEntrada;
PT15_Reposo_Previo := PT15_FB.Reposo;
PT15_Avance_Previo := PT15_FB.Avance1;

```

Figura 34 : Bloque de programa "COMUNICACION_MODBUS"

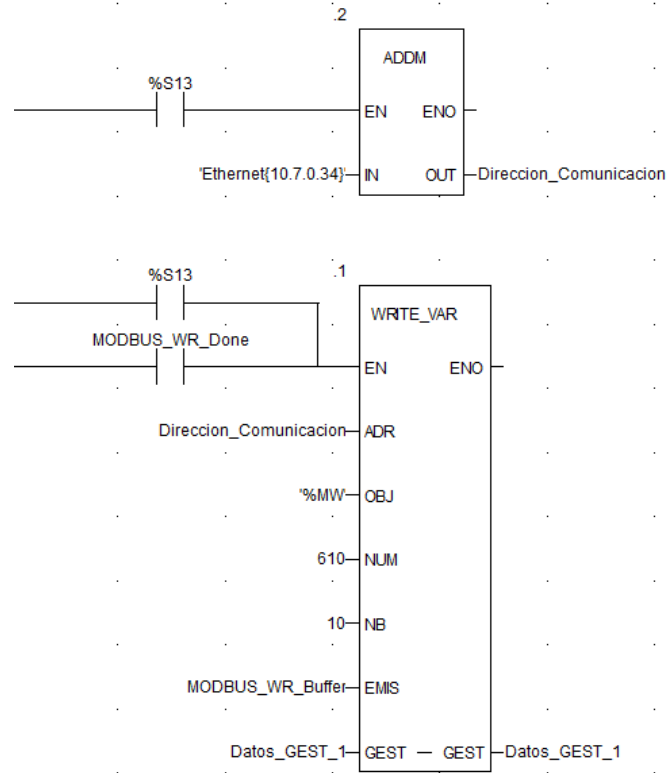


Figura 35 : Bloque de programa "COMM_MODBUS" (1)

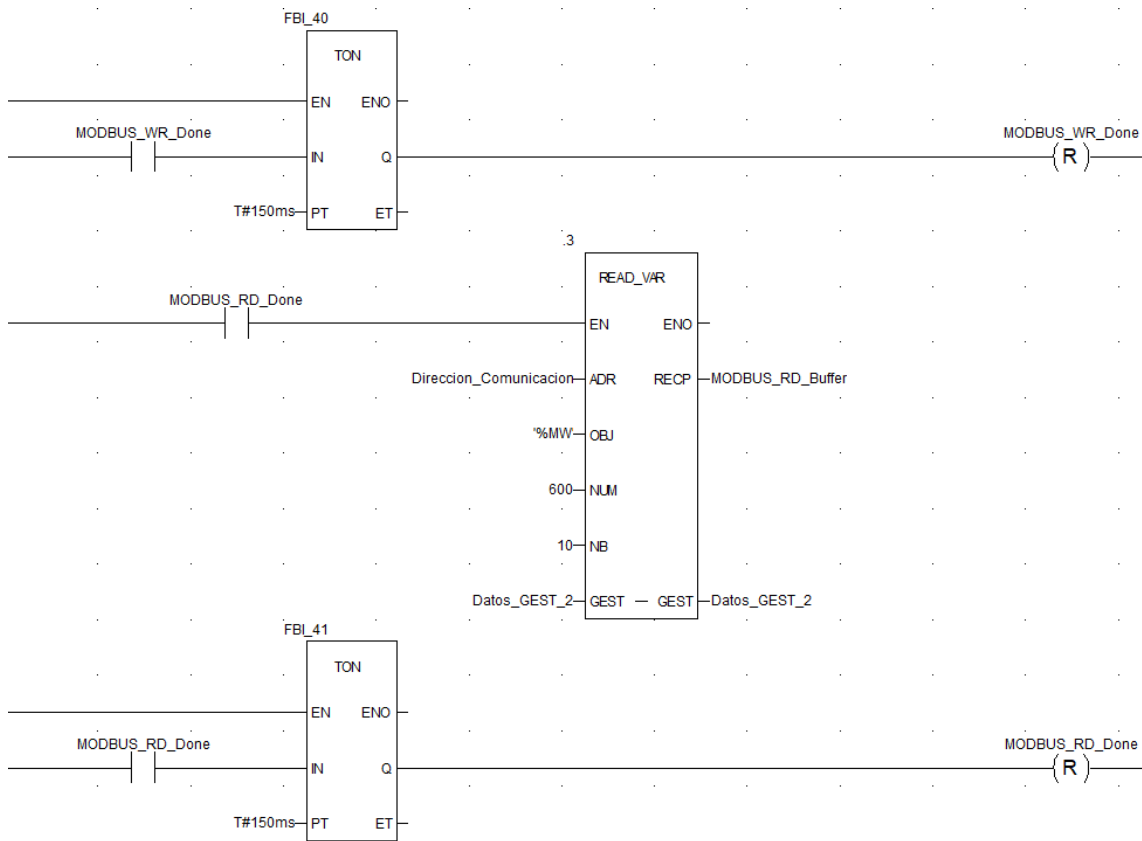


Figura 36 : Bloque de programa "COMM_MODBUS" (2)

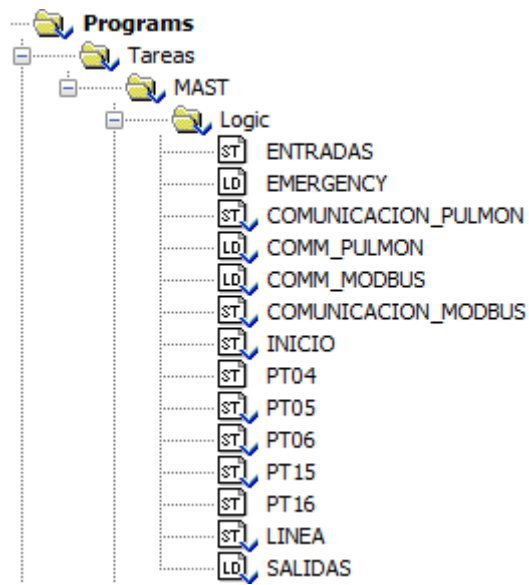


Figura 37 : Orden de ejecución de los bloques de programa de la línea CAN

2. Programa línea Modbus

2.1 Declaración de variables

2.1.1 Variables e instancias globales

```

1  VAR_GLOBAL
2      EMERGENCIA: BOOL;
3      PRODUCT_0: PRODUCTO;
4      REARME: BOOL := 0;
5      PARO_EMERGENCIA: BOOL;
6      EMERGENCY_LOCK: BOOL;
7      LUZ_ROJA_BALIZA: BOOL;
8
9      PT07_FB: Control_Plataforma;
10     PT08_FB: Control_Plataforma;
11     PT09_FB: Control_Plataforma;
12     PT10_FB: Control_Plataforma;
13     PT11_FB: Control_Plataforma;
14     PT12_FB: Control_Plataforma;
15     PT13_FB: Control_Plataforma;
16     PT14_FB: Control_Plataforma;
17     PT15_FB: Control_Plataforma;
18     DIR10_FB: Control_Plataforma;
19     DIR11_FB: Control_Plataforma;
20     DIR12_FB: Control_Plataforma;
21     DIR13_FB: Control_Plataforma;
22     DIR14_FB: Control_Plataforma;
23     DIR15_FB: Control_Plataforma;
24     DIR16_FB: Control_Plataforma;
25     DIR17_FB: Control_Plataforma;
26     DIR18_FB: Control_Plataforma;
27     DIR19_FB: Control_Plataforma;
28
29     PR_ET1: PROCESO;
30     PR_ET2: PROCESO;
31     PR_ET3: PROCESO;
32     PR_ET4: PROCESO;
33
34     CAN_ADDRESS: ADDRESS;
35     CAN_WR_Buffer AT %MW600 : ARRAY[0..10] OF INT;
36     CAN_RD_Buffer AT %MW610 : ARRAY[0..10] OF INT;
37     CAN_RD_Done: BOOL;
38     CAN_WR_Done: BOOL;

```

Figura 38 : Variables e instancias globales línea Modbus (1)

2.1.2 Variables de entradas físicas

```

1  VAR_GLOBAL
2  (*Variables existentes previas no usadas
3  DC00  AT  %IX0.1: BOOL;
4  DC02  AT  %IX9.2: BOOL;
5  E6    AT  %IX1.4: BOOL;
6  E7 AT  %IX9.5:   BOOL;
7  P_ALL_Plates2 AT  %IX9.4: BOOL;
8  P_ONE_Plate1  AT  %IX1.5: BOOL;
9  P_ONE_Plate2  AT  %IX9.3: BOOL;
10 PRESOSTATO_SEGURIDAD_MINIMA AT  %IX9.1: BOOL;
11 PULSADOR_REAL_MOTOR_CENTR  AT  %IX3.0: BOOL;
12 PULSADOR_REAL_MOTOR_DER AT  %IX3.1: BOOL; *)
13
14
15 DC01  AT  %IX38.5:   BOOL;
16 DFT3  AT  %IX30.2:   BOOL;
17 DFT4  AT  %IX34.1:   BOOL;
18 DI08  AT  %IX30.3:   BOOL;
19 DI09  AT  %IX34.0:   BOOL;
20 DI10  AT  %IX42.0:   BOOL;
21 DIB03  AT  %IX34.3:   BOOL;
22 DIB04  AT  %IX38.3:   BOOL;
23 DIB05  AT  %IX46.0:   BOOL;
24
25 EM    AT  %IX34.4:   BOOL; (*Variable para pulsador de emergencia lateral izquierdo*)
26
27 EM2   AT  %IX42.5:   BOOL; (*Pulsador de Emergencia del medio, al lado de PT12*)
28
29 P_ALL_Plates1 AT  %IX38.0:   BOOL;
30 P01    AT  %IX34.5:   BOOL; (*Pulsador 1 lateral izquierdo*)
31 P02    AT  %IX34.6:   BOOL; (*Pulsador 2 lateral izquierdo*)
32 P03    AT  %IX42.3:   BOOL; (*Pulsador 3, al lado de la emergencia 2*)
33 P04    AT  %IX42.4:   BOOL;
34
35 S_DIR10 AT  %IX54.0:   BOOL;
36 S_DIR11 AT  %IX30.4:   BOOL;
37 S_DIR12 AT  %IX30.5:   BOOL;
38 S_DIR13 AT  %IX46.1:   BOOL;
39 S_DIR14 AT  %IX34.2:   BOOL;
40 S_DIR15 AT  %IX38.4:   BOOL;
41 S_DIR16 AT  %IX38.1:   BOOL;
42 S_DIR17 AT  %IX38.2:   BOOL;
43 S_DIR18 AT  %IX46.3:   BOOL;
44 S_DIR19 AT  %IX46.2:   BOOL;
45 DIB02  AT  %IX30.0:   BOOL;
46
47
48 END_VAR

```

Figura 39 : Variables entradas físicas línea Modbus

2.1.3 Variables de salidas físicas

```

1  VAR_GLOBAL
2
3  DIR10_BAJA AT %QX22.2:  BOOL;
4  DIR11_BAJA AT %QX22.0:  BOOL;
5  DIR12_BAJA AT %QX22.1:  BOOL;
6  DIR13_BAJA AT %QX30.2:  BOOL;
7  DIR14_BAJA AT %QX20.2:  BOOL;
8  DIR15_BAJA AT %QX28.2:  BOOL;
9  DIR16_BAJA AT %QX20.4:  BOOL;
10 DIR17_BAJA AT %QX20.5:  BOOL;
11 DIR18_BAJA AT %QX28.4:  BOOL;
12 DIR19_BAJA AT %QX30.0:  BOOL;
13
14 M16      AT %QX34.1:  BOOL;  (* motor PT09 to DIR14 *)
15 M17      AT %QX34.2:  BOOL;  (* motor PT10 to DIR15 *)
16 M19      AT %QX34.3:  BOOL;
17 M20      AT %QX34.4:  BOOL;
18 M20_reverse AT %QX34.5:  BOOL;
19 M21      AT %QX34.0:  BOOL;
20 M22      AT %QX36.0:  BOOL;
21
22 MOTOR_CENTRALBAND AT %QX22.4:  BOOL;
23 MOTOR_RIGHTBAND  AT %QX22.5:  BOOL;
24
25 PT08_SUBE AT %QX30.3:  BOOL;
26 PT09_BAJA AT %QX20.1:  BOOL;
27 PT09_SUBE AT %QX20.0:  BOOL;
28 PT10_BAJA AT %QX28.1:  BOOL;
29 PT10_SUBE AT %QX28.0:  BOOL;
30 PT11_SUBE AT %QX20.3:  BOOL;
31 PT12_SUBE AT %QX28.3:  BOOL;
32 PT13_SUBE AT %QX28.5:  BOOL;
33 PT14_SUBE AT %QX30.1:  BOOL;
34
35 END_VAR

```

Figura 40 : Variables salidas físicas línea Modbus

2.1.4 Variables declaradas para el tratamiento de sensores

```

1  VAR_GLOBAL
2      Sensor_DIR10: BOOL;
3      Sensor_DIR11: BOOL;
4      Sensor_DIR12: BOOL;
5      Sensor_DIR13: BOOL;
6      Sensor_DIR14: BOOL;
7      Sensor_DIR15: BOOL;
8      Sensor_DIR16: BOOL;
9      Sensor_DIR17: BOOL;
10     Sensor_DIR18: BOOL;
11     Sensor_DIR19: BOOL;
12
13     Sensor_DIB02: BOOL;
14     Sensor_DIB03: BOOL;
15     Sensor_DIB04: BOOL;
16     Sensor_DIB05: BOOL;
17
18     Sensor_DI08: BOOL;
19     Sensor_DI09: BOOL;
20     Sensor_DI10: BOOL;
21 END_VAR

```

Figura 41 : Variables declaradas para el tratamiento de sensores

2.2 Estructuras

```

1  TYPE PRODUCTO :
2  STRUCT
3      IDPRODUCTO: INT;
4      TIPOPRODUCTO: INT;
5      ESTADO: INT;
6  END_STRUCT
7  END_TYPE

```

Figura 42 : Estructura de "PRODUCTO"

```

1  TYPE Datos_Estado :
2  STRUCT
3      BandejaEntrada: PRODUCTO;
4      BandejaSalidal: PRODUCTO;
5      BandejaSalida2: PRODUCTO;
6      Estado: INT;
7  END_STRUCT
8  END_TYPE

```

Figura 43 : Estructura de "Datos_Estado"

```

1  TYPE PLATAFORMA :
2  STRUCT
3      Reposo: BOOL;
4      Peticion:  BOOL := 0;
5      Avancel:  BOOL := 0;
6      Avance2:  BOOL := 0;
7      Bloqueo1:  BOOL := 0;
8      Bloqueo2:  BOOL := 0;
9      AVAN1:  BOOL := 0;
10     AVAN2:  BOOL := 0;
11     AcuseEnvioInfo:  BOOL := 0;
12     BandejaEntrada: PRODUCTO;
13     PROD_OUT1_TEMP: PRODUCTO;
14     PROD_OUT2_TEMP: PRODUCTO;
15     PROD_OUT1: PRODUCTO;
16     PROD_OUT2: PRODUCTO;
17     Bandeja_Vacia: PRODUCTO;
18 END_STRUCT
19 END_TYPE

```

Figura 44 : Estructura de "PLATAFORMA"

2.3 Bloques de funciones (FB)

2.3.1 Control_Plataforma

```

1  FUNCTION_BLOCK Control_Plataforma
2  VAR_INPUT
3      PresenciaBandeja: BOOL;
4  END_VAR
5  VAR_IN_OUT
6      BandejaDestino1: PRODUCTO;
7      BandejaDestino2: PRODUCTO;
8      ConfirmacionInfo1: BOOL;
9      ConfirmacionInfo2: BOOL;
10 END_VAR
11 VAR_INPUT
12     ReposoDestino1: BOOL;
13     ReposoDestino2: BOOL;
14     AvanceAux1: BOOL;
15     AvanceAux2: BOOL;
16     Emergency: BOOL;
17     Especial1: BOOL;
18     Especial2: BOOL;
19     ProductoVacio: PRODUCTO;
20 END_VAR
21 VAR_OUTPUT
22     Motor1_1: BOOL;
23     Motor1_2: BOOL;
24     Motor2_1: BOOL;
25     Motor2_2: BOOL;
26     Retenedor1: BOOL;
27     Retenedor2: BOOL;
28 END_VAR
29 VAR_INPUT
30     REARME: BOOL;
31 END_VAR
32 VAR
33     TOF_Retenedor2: TOF;
34     TOF_Retenedor1: TOF;
35     FILTRO_SENSOR: TON;
36     ET_Filtro: TIME;
37     ET_Retenedor2: TIME;
38     ET_Retenedor1: TIME;
39 END_VAR
40 VAR_INPUT
41     Tiempo_ACK: TIME;
42 END_VAR

```

Figura 45 : Estructura del FB "Control_Plataforma" (1)

```

43 VAR
44     ET_Retenedor3: TIME;
45     BandejaSalidal: PRODUCTO;
46     FReposo: BOOL;
47     FAvance2: BOOL;
48     FAvance1: BOOL;
49     BandejaSalida2: PRODUCTO;
50     Bloqueo: BOOL;
51     Acuse_Timer: TON;
52 END_VAR
53 VAR_INPUT
54     Reposo: BOOL;
55     Avance2: BOOL;
56     Avance1: BOOL;
57     BandejaEntrada: PRODUCTO;
58     Peticion: BOOL;
59     Bloqueo2: BOOL;
60     Bloqueo1: BOOL;
61     AcuseEnvioInfo: BOOL;
62     ActivacionSalida2: BOOL;
63     ActivacionSalidal: BOOL;
64 END_VAR

```

Figura 46 : Estructura del FB "Control_Plataforma" (2)

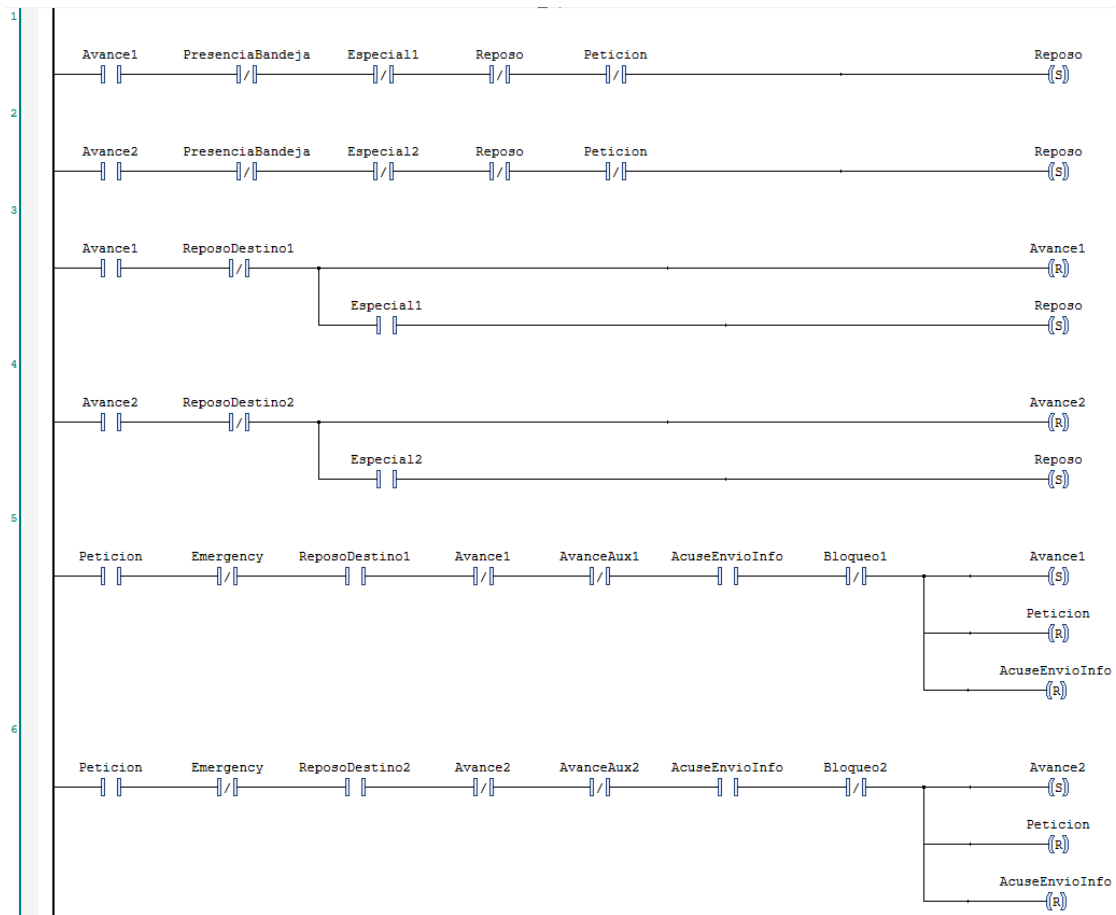


Figura 47 : Rutina del FB "Control_Plataforma" (1)

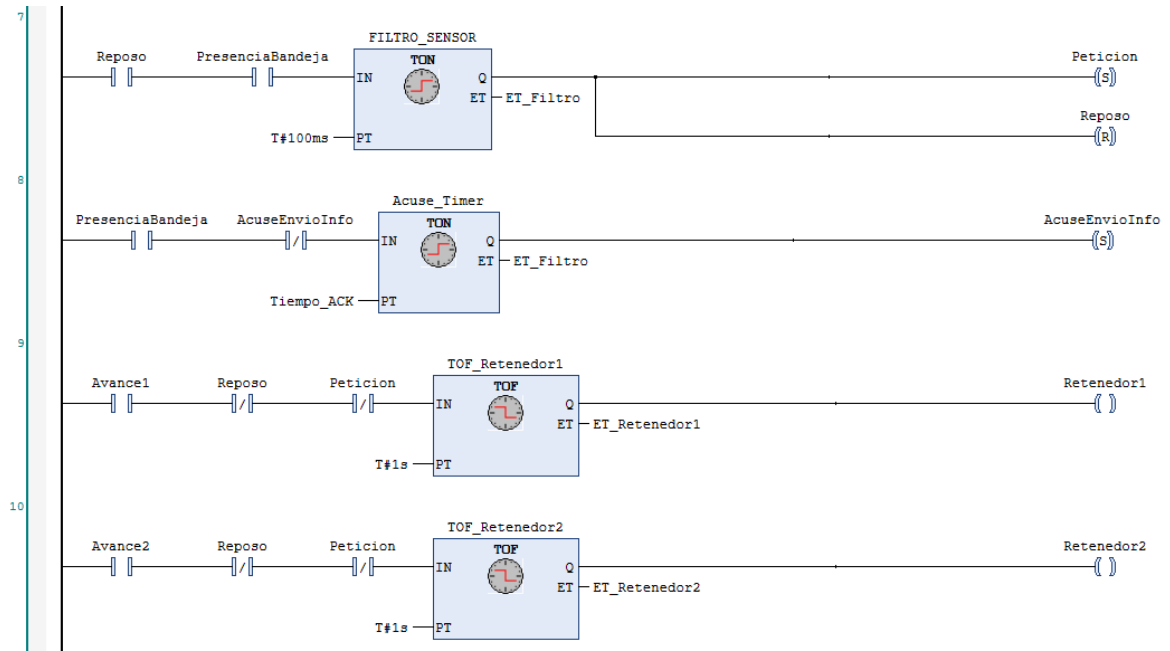


Figura 48 : Rutina del FB "Control_Plataforma" (2)



Figura 49 : Rutina del FB "Control_Plataforma" (3)

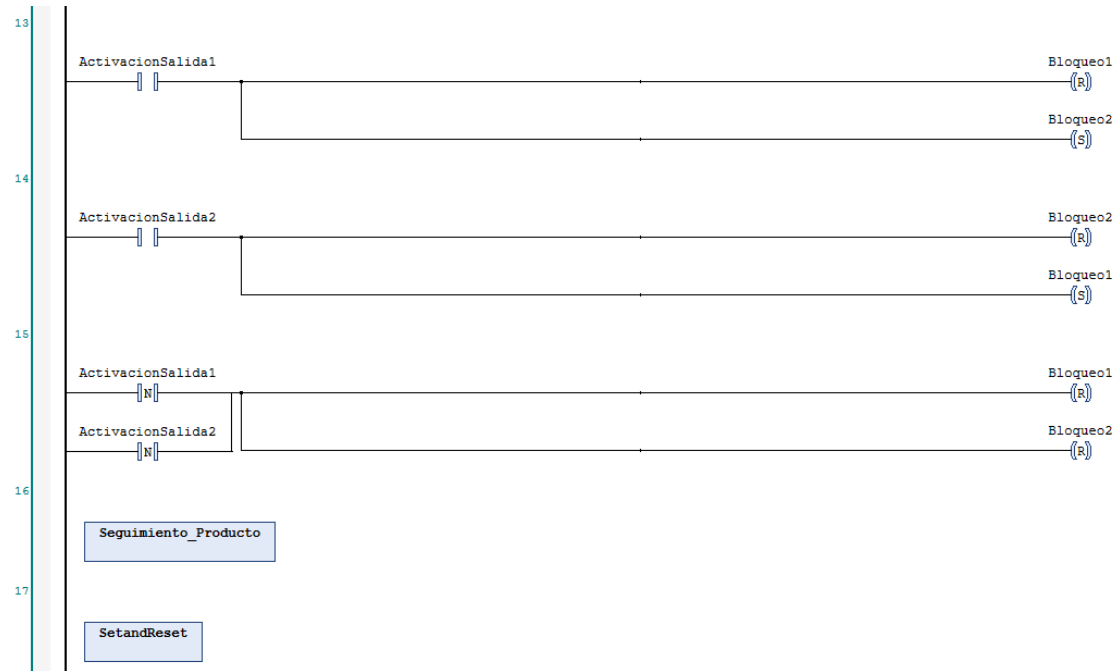


Figura 50 : Rutina del FB "Control_Plataforma" (4)

```

1 //Gestión del paso de datos del producto entre la plataforma actual y las que la siguen
2
3 //Variable Plat_IN1 es un booleano que tiene como funcion saber si por la via 1 se le está enviando o no producto
4 IF Avancel AND NOT FAvancel THEN
5     //En caso de que si, se actualiza el producto de la plataforma actual
6     BandejaSalida1:=BandejaEntrada;
7 END_IF
8
9 IF Avance2 AND NOT FAvance2 THEN
10    BandejaSalida2:=BandejaEntrada;
11 END_IF
12
13 IF NOT Avancel AND FAvancel THEN
14    BandejaDestino1:=BandejaSalida1;
15    ConfirmacionInfo1 := TRUE;
16 END_IF
17
18 IF NOT Avance2 AND FAvance2 THEN
19    BandejaDestino2:=BandejaSalida2;
20    ConfirmacionInfo2 := TRUE;
21 END_IF
22
23 //Puesta a 0 de la información sobre la bandeja cuando la celda esta en Reposo
24 IF Reposo AND NOT FReposo THEN
25    BandejaEntrada:=ProductoVacio;
26 END_IF
27
28 FAvancel:=Avancel;
29 FAvance2:=Avance2;
30 FReposo:=Reposo;

```

Figura 51 : Subrutina "Seguimiento_Producto" del FB "Control_Plataforma"

```

1  IF S13 OR REARME THEN
2      Reposo:=TRUE;
3      Avancel:=FALSE;
4      Avance2:=FALSE;
5      Peticion:=FALSE;
6  END_IF

```

Figura 52 : Subrutina "SetandReset" del FB "Control_Plataforma"

2.3.2 PROCESO

```

1  FUNCTION_BLOCK PROCESO
2  VAR_INPUT
3      PresenciaBandeja: BOOL;
4      Num_PT: INT;
5      PR_Time: TIME;
6  END_VAR
7  VAR_OUTPUT
8      Bloqueo1: BOOL;
9      Bloqueo2: BOOL;
10 END_VAR
11 VAR_IN_OUT
12     BandejaEntrada: PRODUCTO;
13 END_VAR
14 VAR
15     PR_No_Aplica: BOOL;
16     Time_IN: BOOL;
17     TON_PR: TON;
18     PR_RndNum: INT;
19     PR_END: BOOL;
20     PR_Time_OK: BOOL;
21     PR_Succes: BOOL;
22     Desvio_DIR1: BOOL;
23     Desvio_DIR2: BOOL;
24 END_VAR

```

Figura 53 : Estructura del FB "PROCESO"

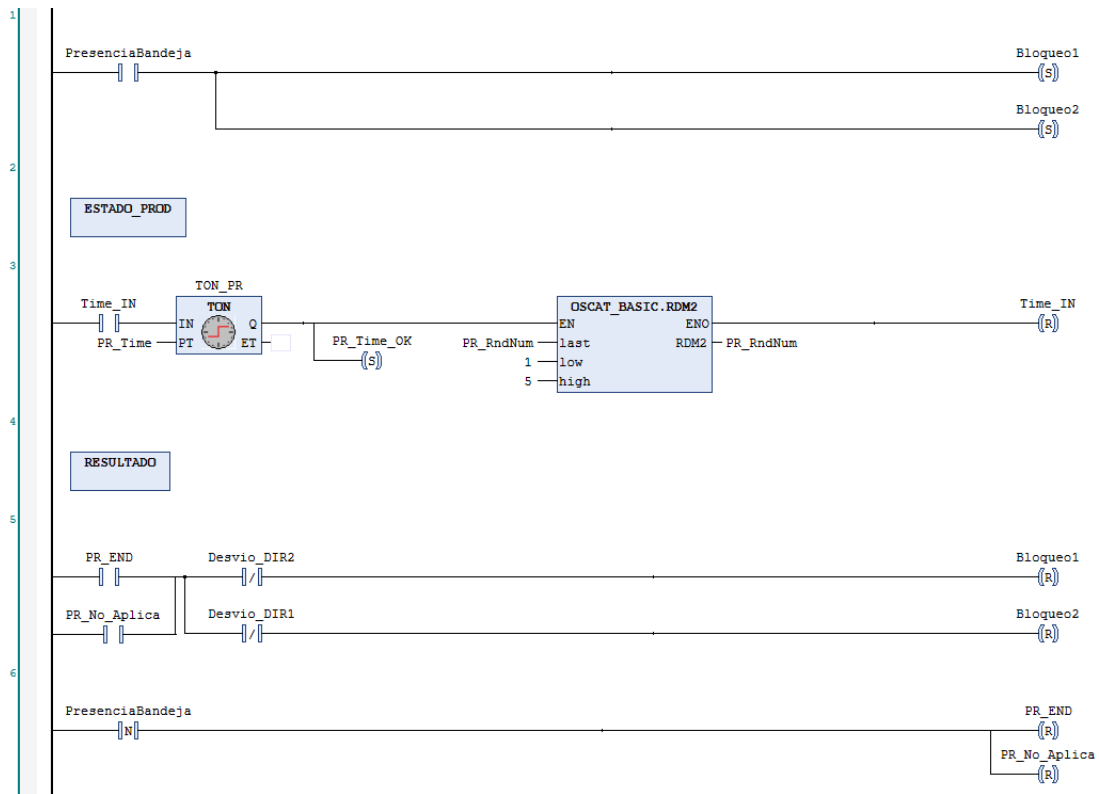


Figura 54 : Rutina del FB "PROCESO"

```

1
2 IF NOT PR_END AND PresenciaBandeja AND (
3   (BandejaEntrada.TIPOPUESTO = 1 AND BandejaEntrada.ESTADO = 0 AND Num_PT = 9) OR
4   (BandejaEntrada.TIPOPUESTO = 1 AND BandejaEntrada.ESTADO = 1 AND Num_PT = 10) OR
5   (BandejaEntrada.TIPOPUESTO = 2 AND BandejaEntrada.ESTADO = 0 AND Num_PT = 11) OR
6   (BandejaEntrada.TIPOPUESTO = 2 AND BandejaEntrada.ESTADO = 1 AND Num_PT = 12) OR
7   (BandejaEntrada.TIPOPUESTO = 3 AND BandejaEntrada.ESTADO = 0 AND Num_PT = 10) OR
8   (BandejaEntrada.TIPOPUESTO = 3 AND BandejaEntrada.ESTADO = 1 AND Num_PT = 12)) THEN
9
10  PR_No_Aplica := FALSE;
11  Time_IN := TRUE;
12
13 ELSE
14
15  PR_No_Aplica := TRUE;
16
17 END_IF

```

Figura 55 : Subrutina "ESTADO_PROD" del FB "PROCESO"


```

1
2 IF PR_Time_OK AND PR_RndNum <5 THEN
3
4     BandejaEntrada.ESTADO := BandejaEntrada.ESTADO + 1;
5
6     PR_Succes := TRUE;
7     PR_Time_OK := FALSE;
8
9     IF (BandejaEntrada.TIPOPDUCTO = 1 AND Num_PT = 9) OR
10        (BandejaEntrada.TIPOPDUCTO = 1 AND Num_PT = 10) THEN
11
12         Desvio_DIR1 :=TRUE;
13
14     ELSIF (BandejaEntrada.TIPOPDUCTO = 3 AND Num_PT = 10) OR
15          (BandejaEntrada.TIPOPDUCTO = 2 AND BandejaEntrada.ESTADO <3 AND Num_PT = 9) THEN
16
17         Desvio_DIR2 :=TRUE;
18
19     END_IF;
20
21     PR_END :=TRUE;
22
23 ELSIF PR_Time_OK AND PR_RndNum = 5 THEN
24
25     BandejaEntrada.ESTADO := BandejaEntrada.ESTADO;
26
27     PR_Time_OK := FALSE;
28     PR_END := TRUE;
29
30 END_IF;

```

Figura 56 : Subrutina "RESULTADO" del FB "PROCESO"

2.4 Bloques de programa

```

1 PROGRAM PREPROCESADO_SENSORES
2 VAR
3 END_VAR

```

Figura 57 : Variables del bloque de programa "PREPROCESADO_SENSORES"

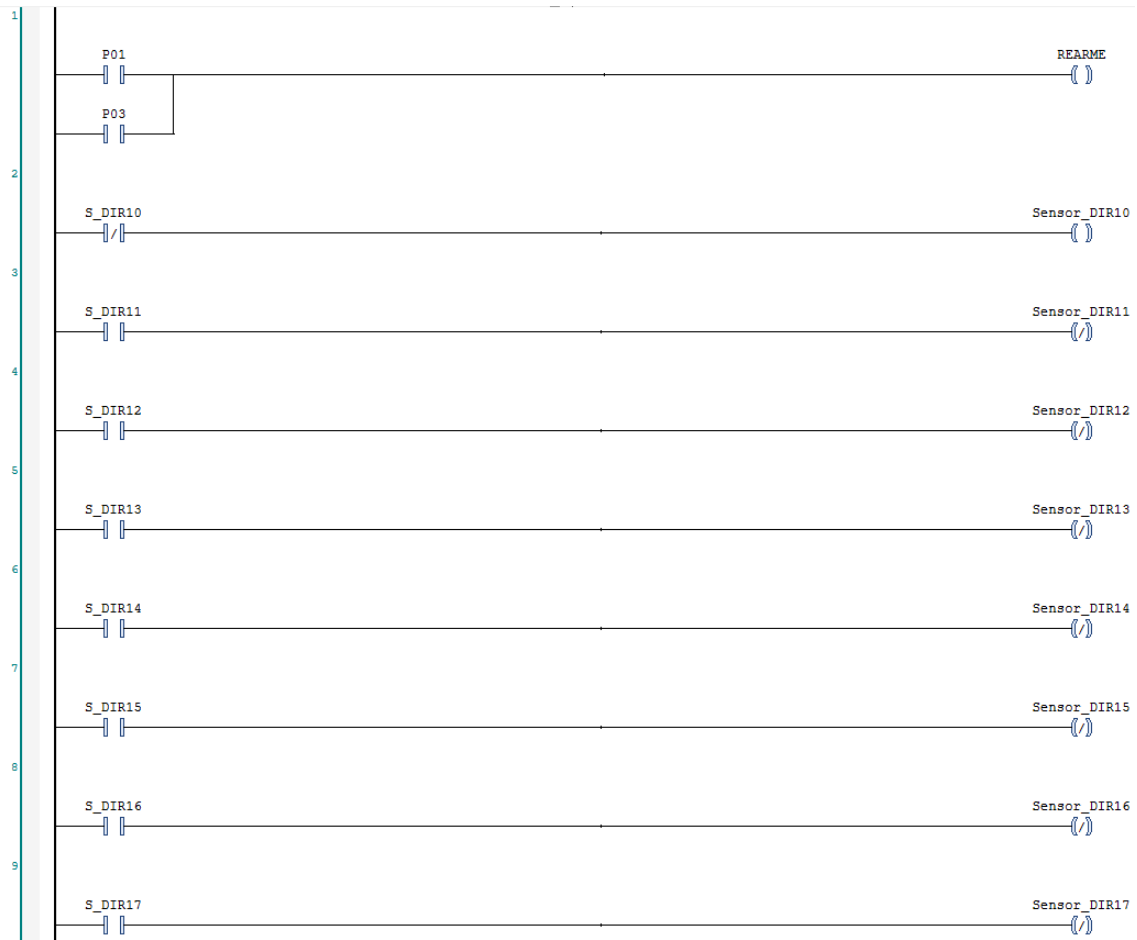


Figura 58 : Bloque de programa "PREPROCESADO_SENSORES" (1)

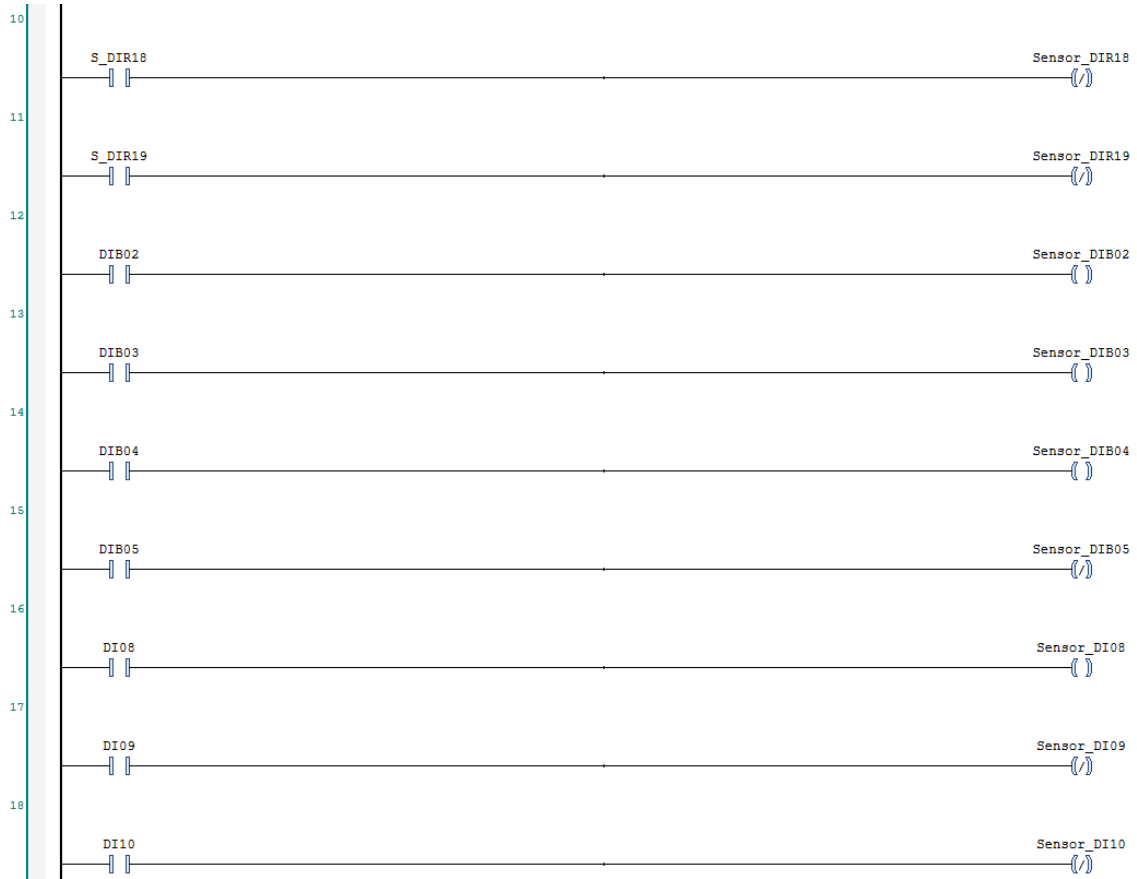


Figura 59 : Bloque de programa "PREPROCESADO_SENSORES" (2)

```

1 | PROGRAM EM_REARME
2 | VAR
3 | END_VAR

```

Figura 60 : Variables del bloque de programa "EM_REARME"

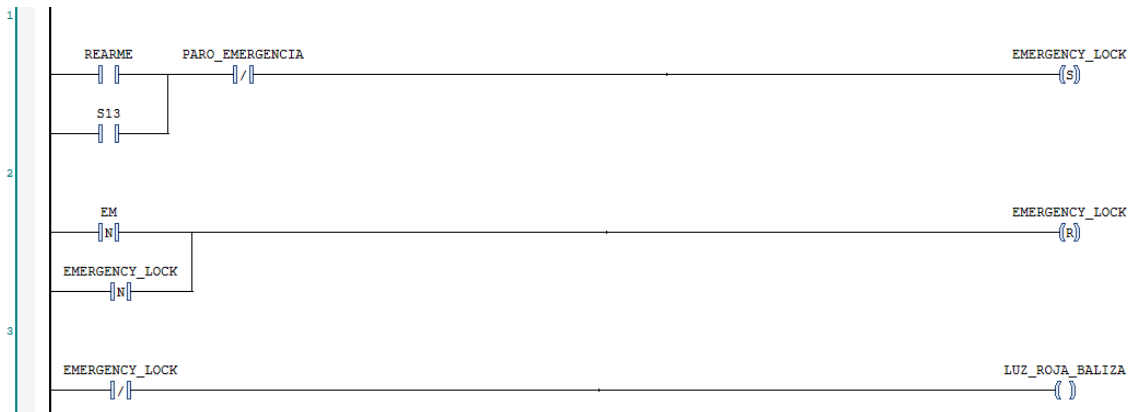


Figura 61 : Bloque de programa "EM_REARME"

```

1 | PROGRAM ESTACIONES_TRABAJO
2 | VAR
3 | END_VAR

```

Figura 62 : Variables del bloque de programa "ESTACIONES_TRABAJO"

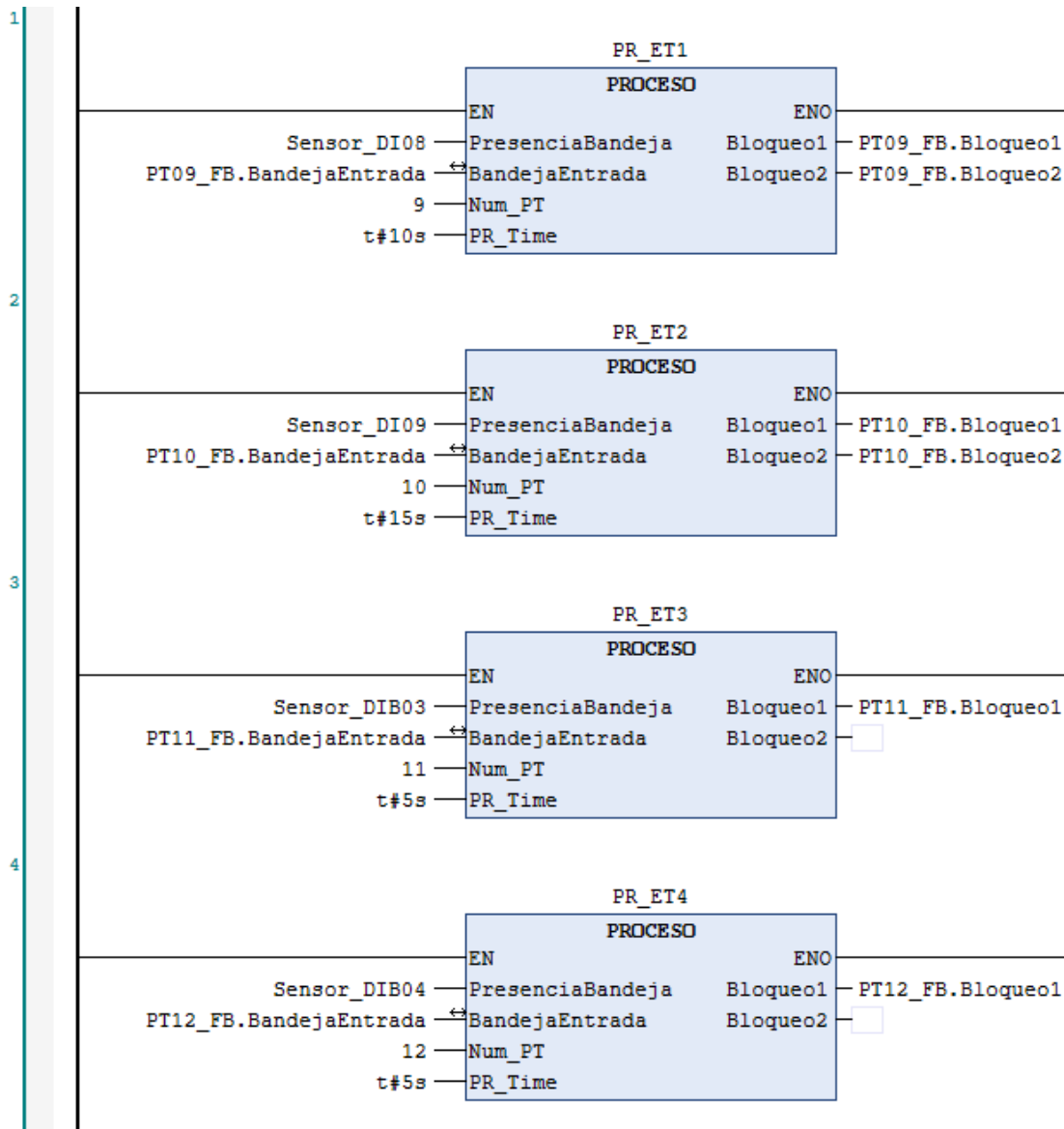


Figura 63 : Bloque de programa "ESTACIONES_TRABAJO"

```
1 PROGRAM MOVIMIENTOS_LINEA
2 VAR
3     Bandeja_NULL: PRODUCTO;
4     TON_P14_P15: TON;
5     TON_DIR19_P14_P15: TON;
6     V_ET : Time;
7     R_ET: Time;
8     T_Peticion_R8: TON;
9     T_Peticion: TON;
10    Rm : Time;
11    Confirmacion_NULL: BOOL;
12 END VAR
```

Figura 64 : Variables del bloque de programa "MOVIMIENTOS_LINEA"

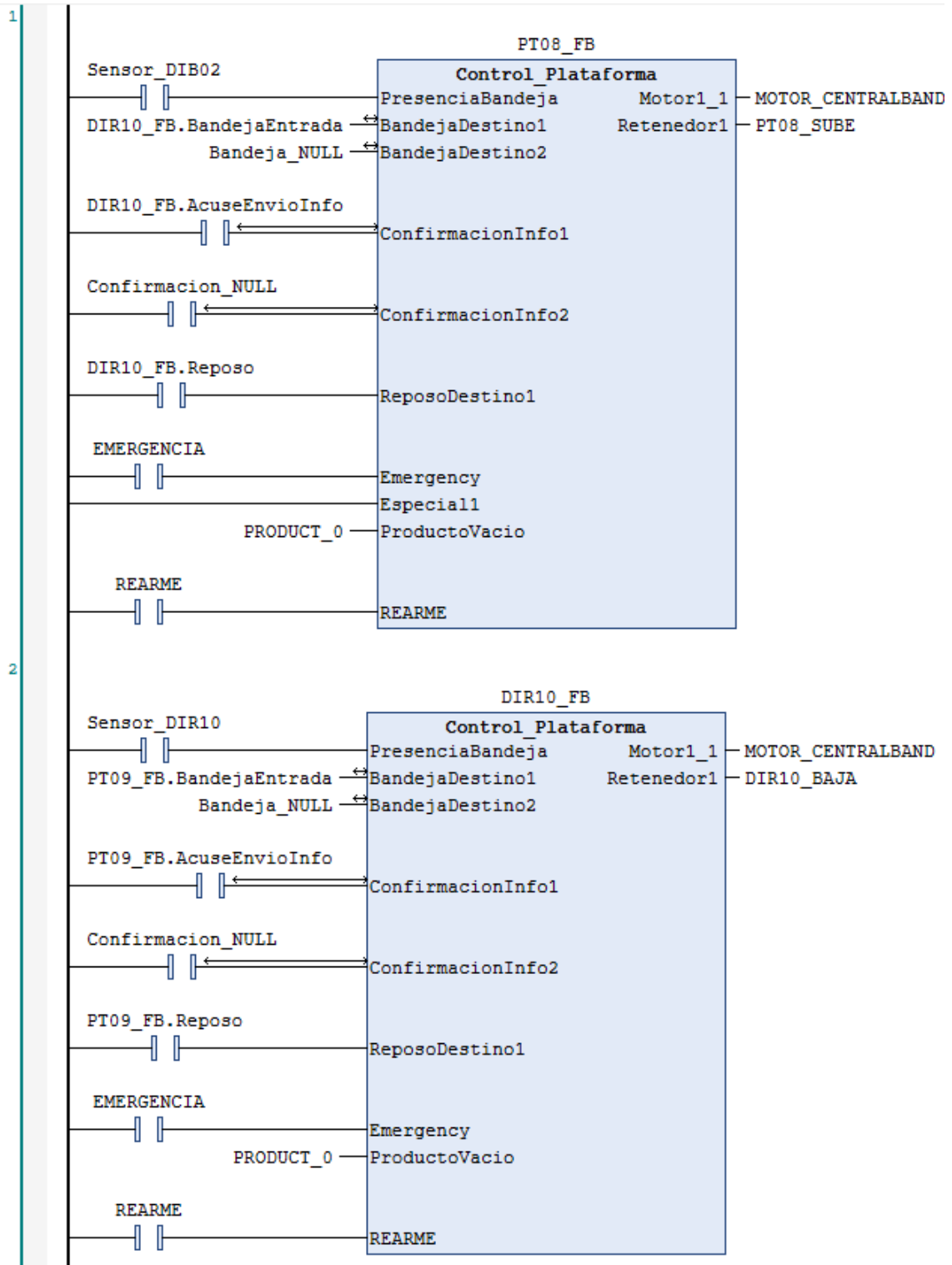


Figura 65 : Bloque de programa "MOVIMIENTOS_LINEA" (1)

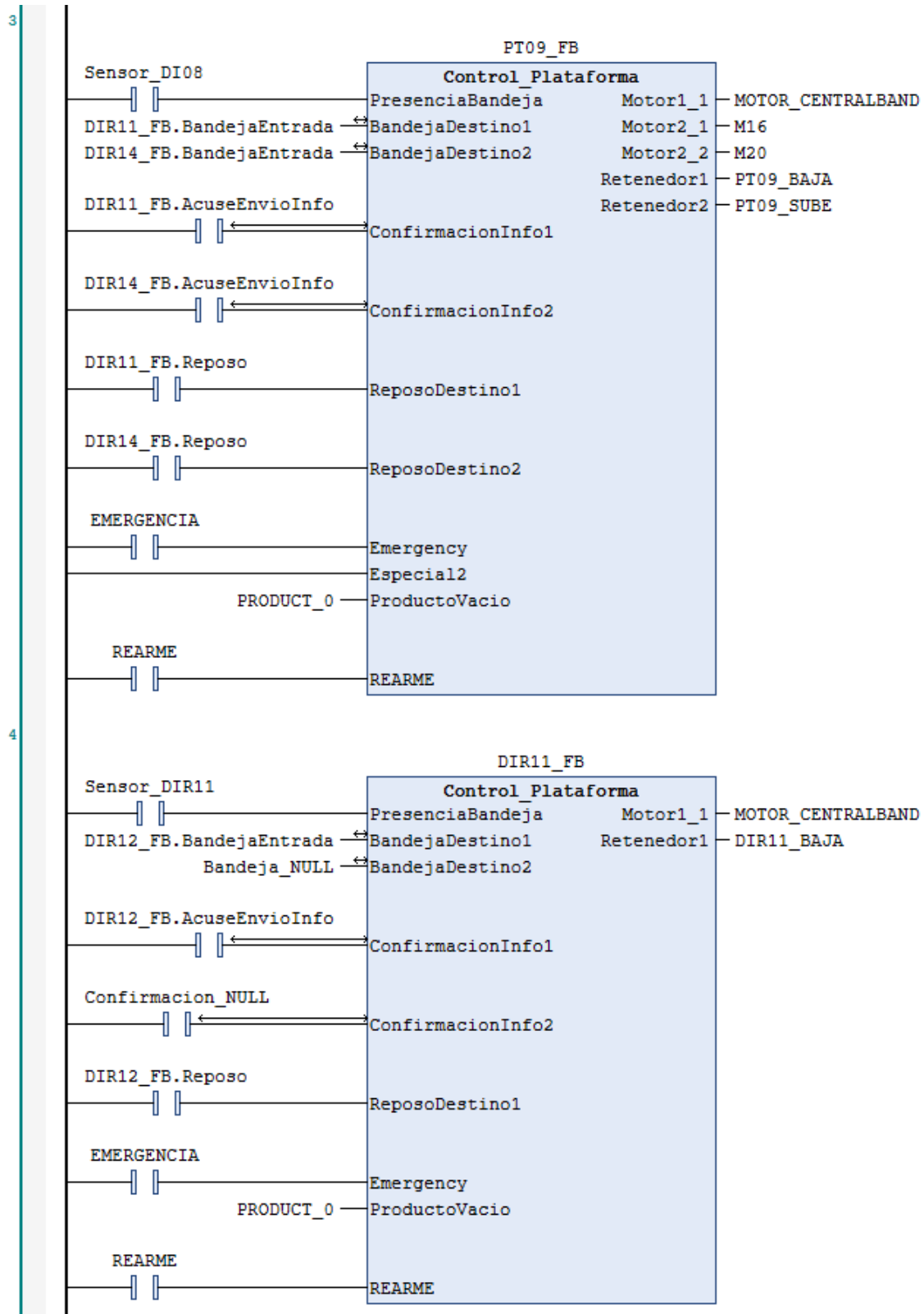


Figura 66 : Bloque de programa "MOVIMIENTOS_LINEA" (2)

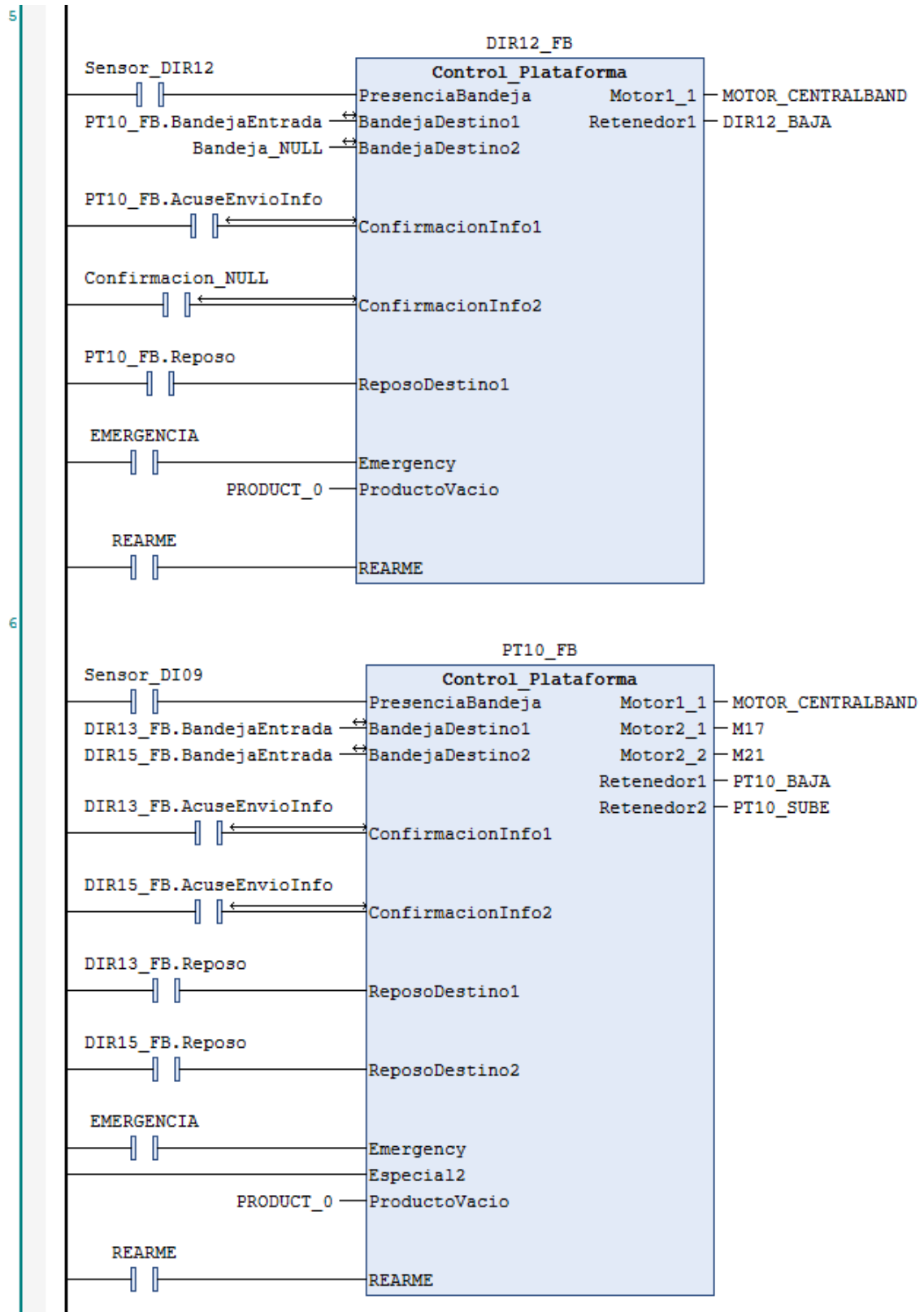


Figura 67 : Bloque de programa "MOVIMIENTOS_LINEA" (3)

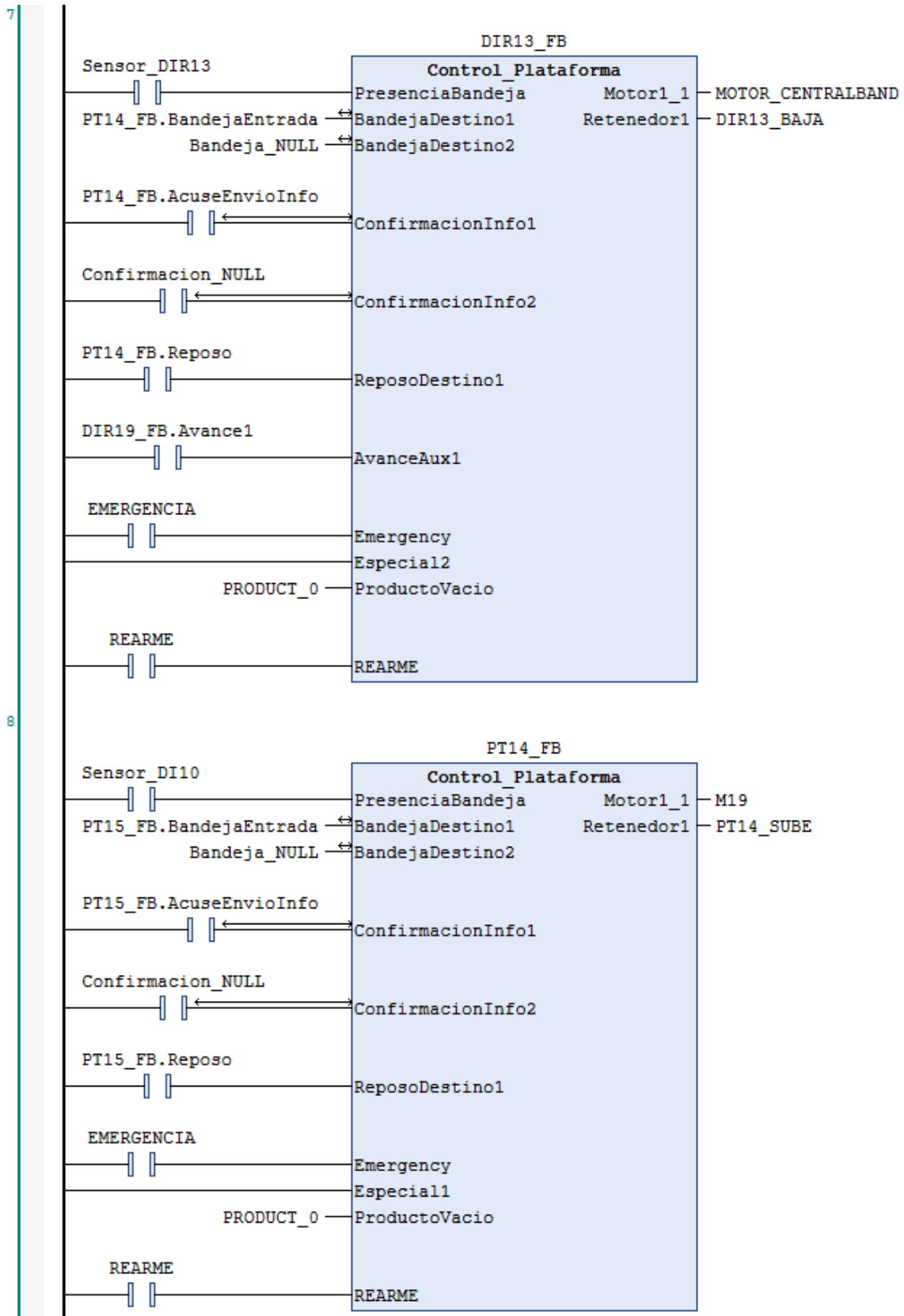


Figura 68 : Bloque de programa "MOVIMIENTOS_LINEA" (4)

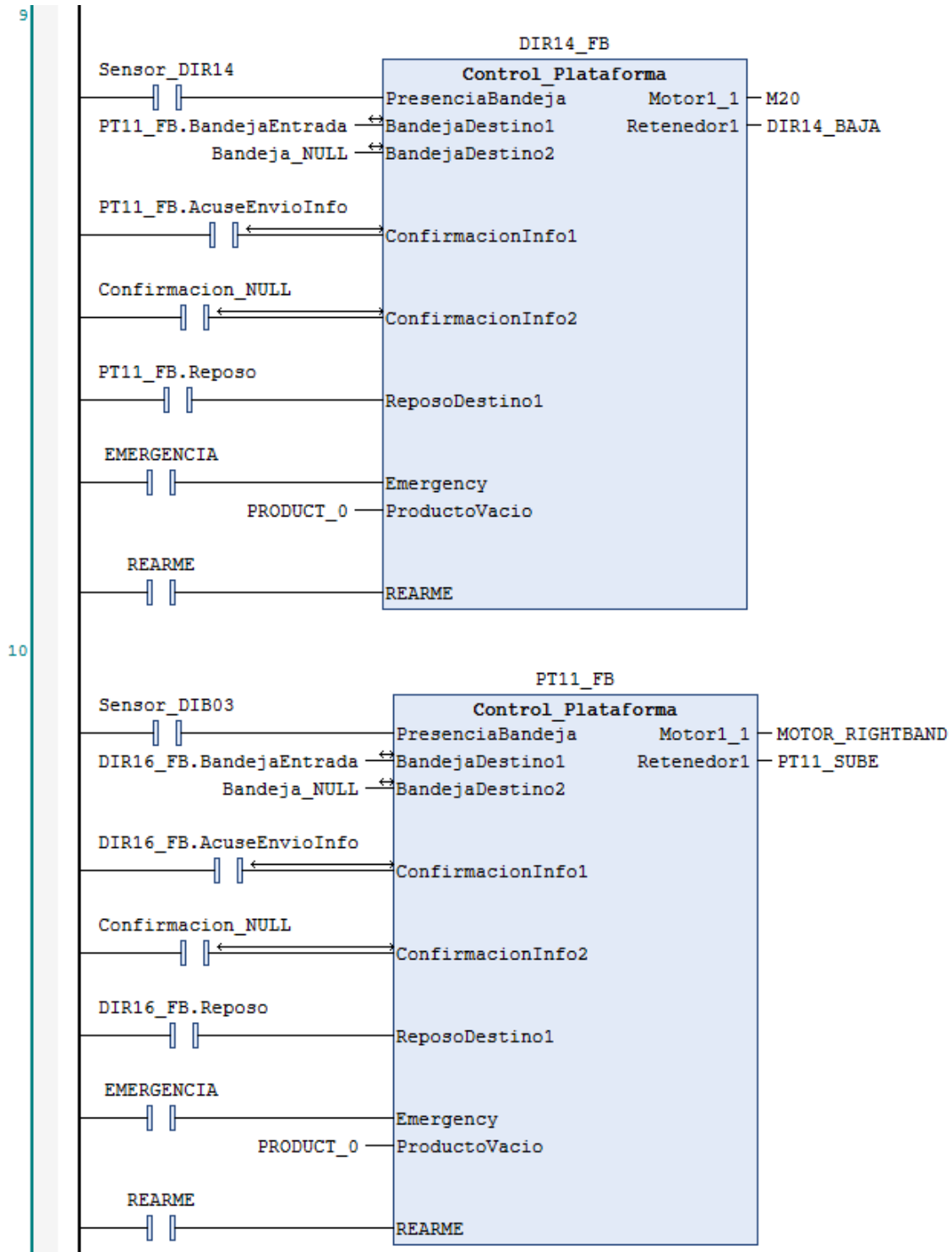


Figura 69 : Bloque de programa "MOVIMIENTOS_LINEA" (5)

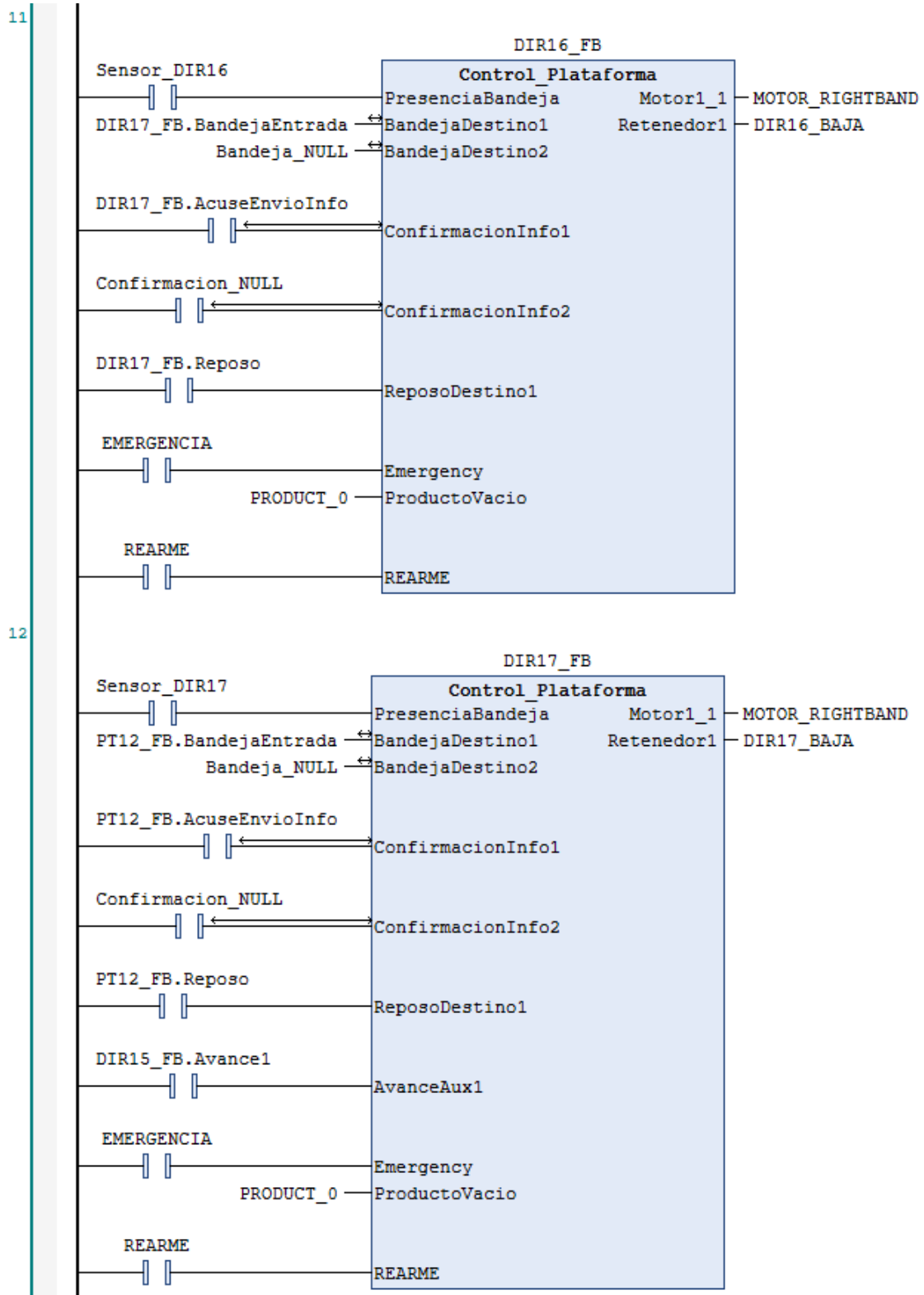


Figura 70 : Bloque de programa "MOVIMIENTOS_LINEA" (6)

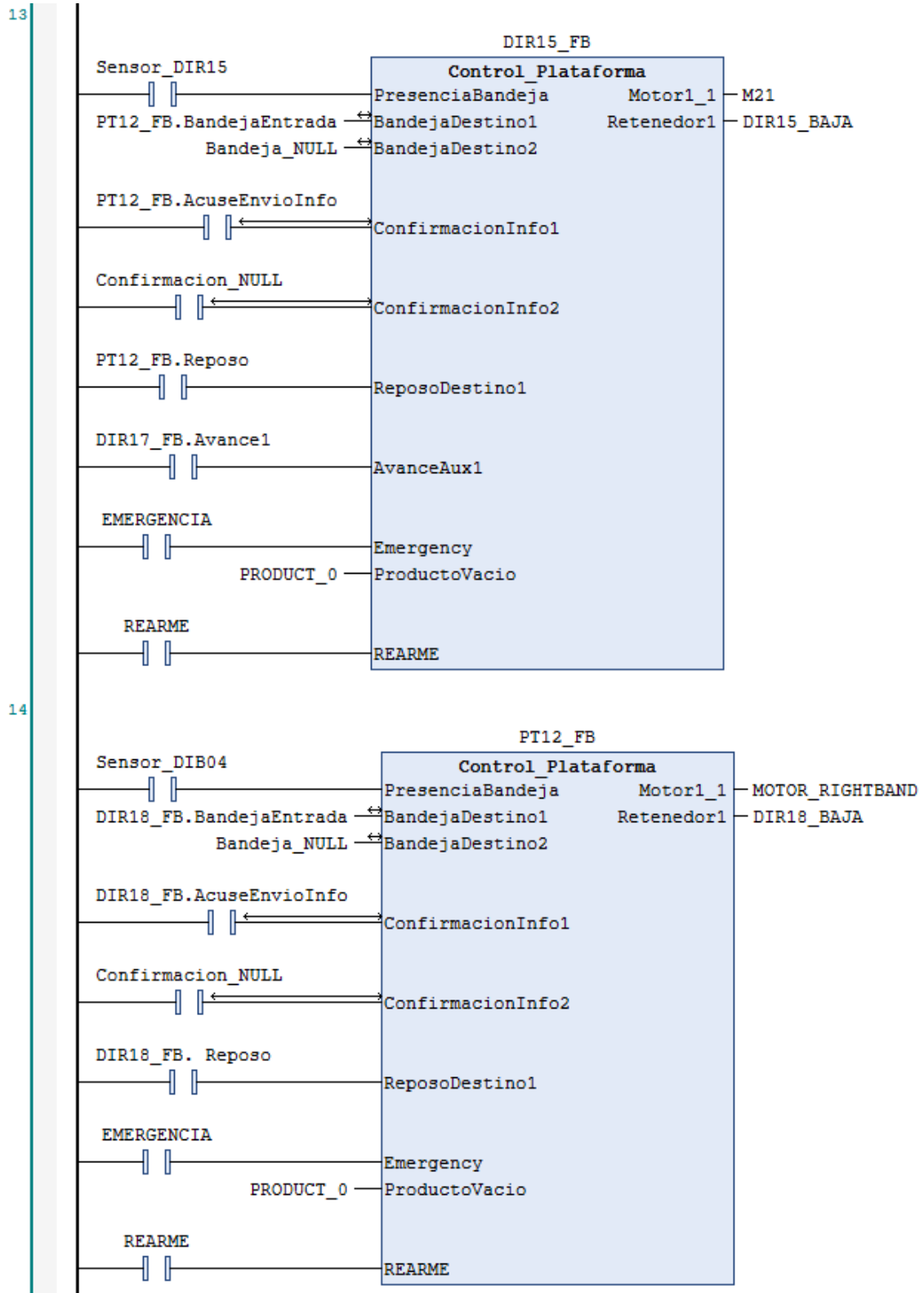


Figura 71 : Bloque de programa "MOVIMIENTOS_LINEA" (7)

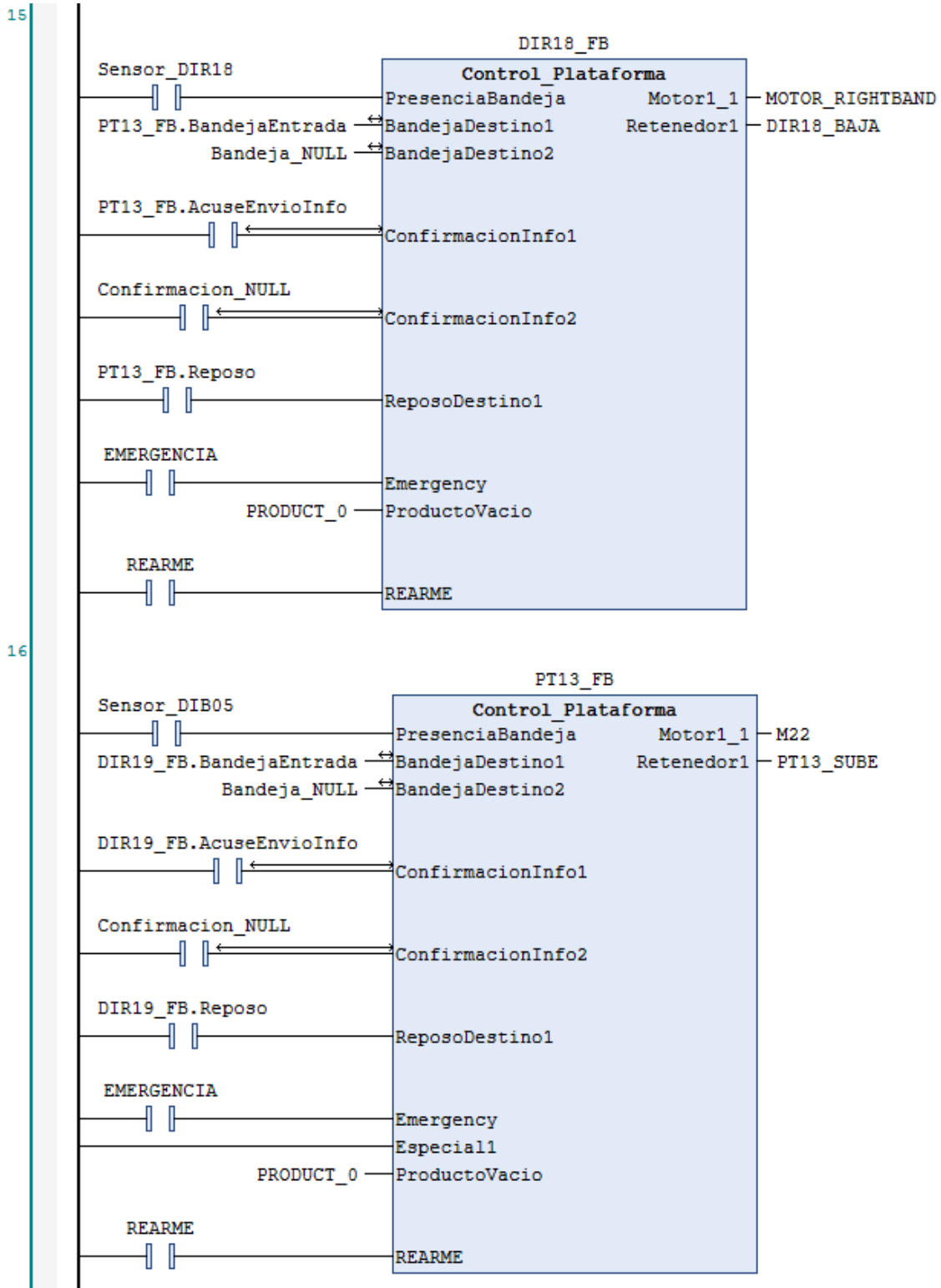


Figura 72 : Bloque de programa "MOVIMIENTOS_LINEA" (8)

17

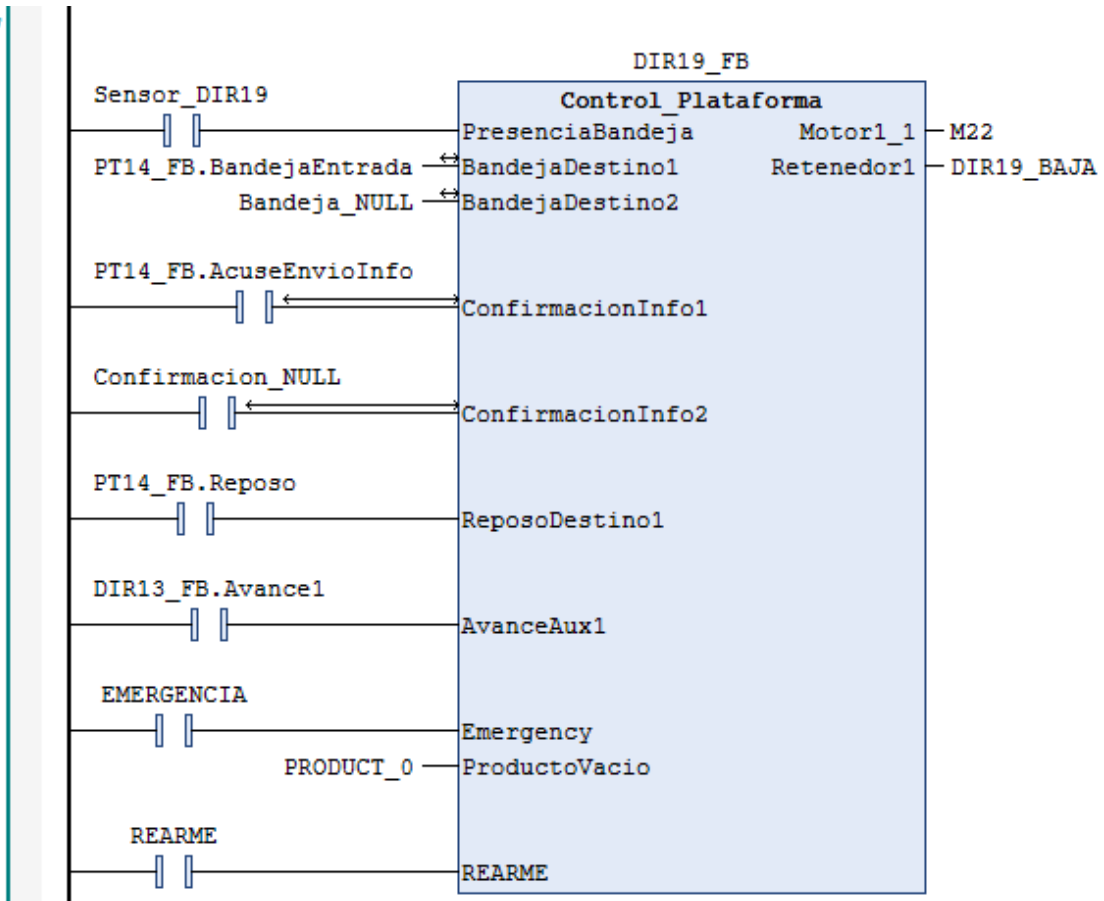


Figura 73 : Bloque de programa "MOVIMIENTOS_LINEA" (9)

```

1  PROGRAM COMM_CAN
2  VAR
3      Comm_CAN : ADDM;
4      CAN_Connected: BOOL;
5      CAN_WR_Buffer: POINTER TO BYTE;
6      CAN_Aborted: BOOL;
7      CAN_Error: BOOL;
8      CAN_CommError: BYTE;
9      CAN_OperError: DWORD;
10     CAN_Busy: BOOL;
11     CAN_RD_Buffer: POINTER TO BYTE;
12     CAN_WR_RD: WRITE_READ_VAR;
13     CAN_Comm_Done: BOOL;
14     CAN_Comm_Redund: BOOL;
15     CAN_TON_REDUND: TON;
16     CAN_TON_WR: TON;
17     CAN_Comm_Start: BOOL;
18 END_VAR

```

Figura 74 : Variables del bloque de programa "COMM_CAN"

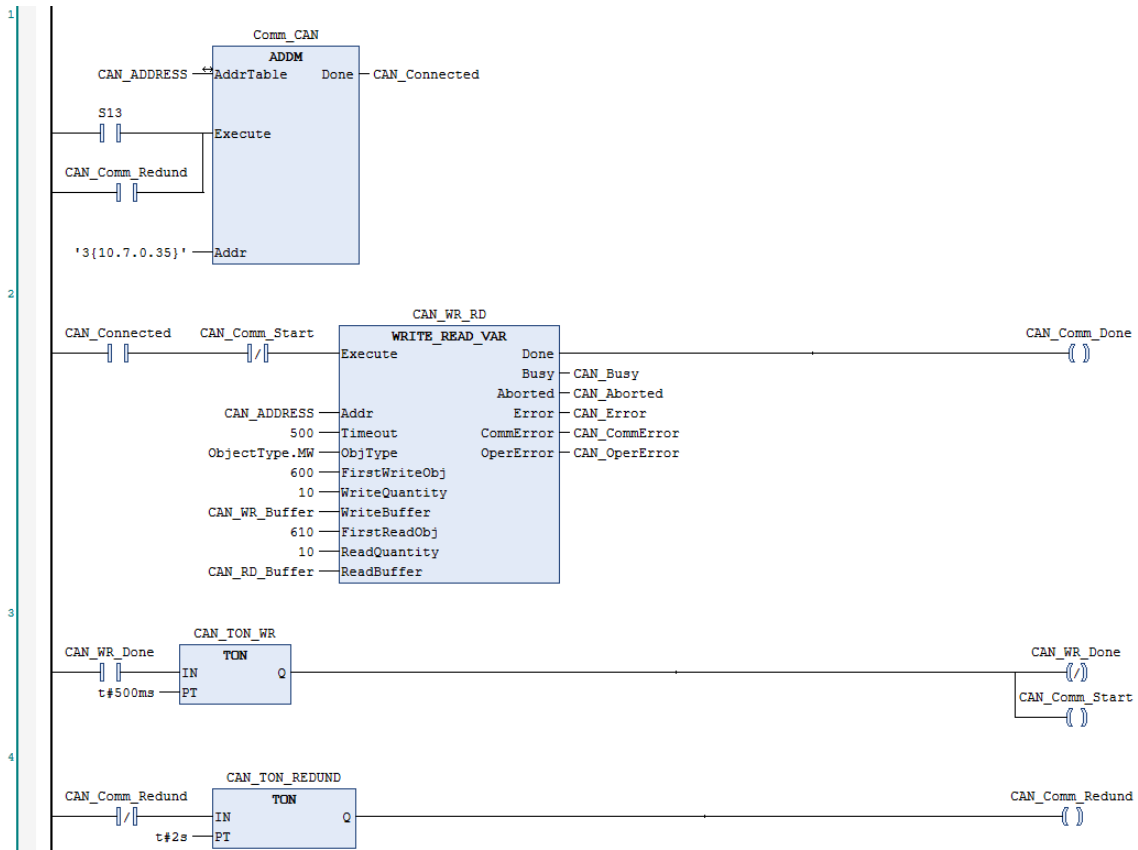


Figura 75 : Bloque de programa "COMM_CAN"

```

1 PROGRAM COMM_WR_RD_CAN
2 VAR
3 END_VAR

```

Figura 76 : Variables del bloque de programa "COMM_WR_RD_CAN"

```

1 PT15_FB.Reposo := CAN_RD_Buffer[1].0;
2 PT15_FB.Avancel := CAN_RD_Buffer[1].1;
3 PT08_FB.AcuseEnvioInfo := CAN_RD_Buffer[1].2;
4 PT08_FB.BandejaEntrada.IDPRODUCTO := CAN_RD_Buffer[2];
5 PT08_FB.BandejaEntrada.TIPOPRODUCTO := CAN_RD_Buffer[3];
6 PT08_FB.BandejaEntrada.ESTADO := CAN_RD_Buffer[4];
7 CAN_RD_Done := TRUE;
8
9 CAN_WR_Buffer[1].0 := PT08_FB.Reposo;
10 CAN_WR_Buffer[1].1 := PT08_FB.Avancel;
11 CAN_WR_Buffer[1].2 := PT15_FB.AcuseEnvioInfo;
12 CAN_WR_Buffer[2] := PT15_FB.BandejaEntrada.IDPRODUCTO;
13 CAN_WR_Buffer[3] := PT15_FB.BandejaEntrada.TIPOPRODUCTO;
14 CAN_WR_Buffer[4] := PT15_FB.BandejaEntrada.ESTADO;
15
16 CAN_WR_Done := TRUE;

```

Figura 77 : Bloque de programa "COMM_WR_RD_CAN"

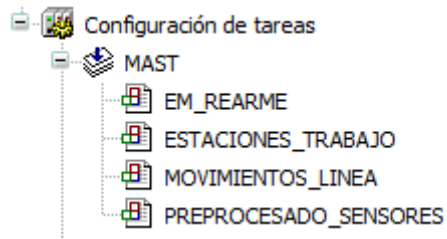


Figura 78 : Orden de ejecución de los bloques de programas de la línea Modbus