

Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías
Industriales

**INTEGRACIÓN DE ALIMENTADOR DE
BANDEJAS A CÉLULA DE FABRICACIÓN
FLEXIBLE CONTROLADA POR PLC**

Autor: Antonio Luis Velasco Martín

Tutor: Luis Fernando Castaño Castaño

**Dep. Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla**

Sevilla, 2015



Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías Industriales

INTEGRACIÓN DE ALIMENTADOR DE BANDEJAS A CÉLULA DE FABRICACIÓN FLEXIBLE CONTROLADA POR PLC

Autor:

Antonio Luis Velasco Martín

Tutor:

Luis Fernando Castaño Castaño

Profesor Contratado Doctor

Dep. Ingeniería de Sistemas y Automática

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2015

Trabajo Fin de Grado: INTEGRACIÓN DE ALIMENTADOR DE BANDEJAS A CÉLULA DE
FABRICACIÓN FLEXIBLE CONTROLADA POR PLC

Autor: Antonio Luis Velasco Martín
Tutor: Luis Fernando Castaño Castaño

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2015

El Secretario del Tribunal

Agradecimientos

Llegado este momento, me gustaría agradecer todo el apoyo recibido durante estos años, que han culminado en la realización del presente trabajo.

Especialmente, quiero agradecer a mis padres y hermana por la confianza que han depositado en mí.

Finalmente, agradecer a Luis Fernando Castaño Castaño, mi tutor, por haberme ayudado e ilustrado durante el desarrollo de este proyecto.

Resumen

En el siguiente proyecto se ha realizado la programación de la integración de una máquina auxiliar, un alimentador de bandejas, a una célula de fabricación que se encuentra en el Departamento de Ingeniería de Sistemas y Automática de la Universidad de Sevilla.

La célula de fabricación flexible consta de un almacén robotizado que gestiona una serie de piezas que serán tratadas en distintos puntos de manipulación. Las bandejas sirven como elemento de soporte de estas piezas para ser desplazadas a lo largo de circuito de transporte desde el almacén a las distintas zonas de trabajo.

Como dispositivo de control de la máquina se ha usado un PLC Modicon M340 de Schneider Electric, y se ha usado el software de programación comercial Unity Pro.

El planteamiento de integración de la máquina se ha realizado siguiendo un esquema de guía GEMMA, implementando mediante bloques de programación SFC (Sequential Function Chart), para crear una estructura maestro-esclavo entre diferentes GRAFCETs. Cabe destacar en este planteamiento la utilización de eventos de tiempo, lo que permite hacer un tratamiento muy eficaz de los posibles fallos que pudieran aparecer en la máquina a nivel de sensores o actuadores.

Debido a que la célula de fabricación flexible dispone de su propio PLC para el control del sistema de transporte de bandejas, se ha desarrollado una comunicación TCP/IP entre dos autómatas, de modo que el autómata principal pueda comandar con instrucciones de alto nivel al autómata que controla el alimentador, (básicamente instrucciones de servir bandejas y almacenar bandejas).

Por otra parte, al disponer la máquina de una pantalla de explotación (Magelis STU 650 de Schneider Electric), se ha desarrollado una aplicación con el software comercial Vijeo Designer. La aplicación consta de varias pantallas que permiten gestionar la máquina de forma sencilla y visualizar la evolución de los estados del almacén de bandejas mediante la guía GEMMA.

ÍNDICE GENERAL

Agradecimientos	viii
Resumen	x
Índice General	xii
Índice de Figuras	xv
Índice de Tablas.....	xix
Índice de la memoria	pág. 2
1.1. Introducción.....	pág. 4
1.2. Antecedentes.....	pág. 5
1.3. Objetivos.....	pág. 7
1.4. Definiciones y abreviaturas	pág. 8
1.5. Descripción general.....	pág. 9
1.5.1. Especificaciones de la instalación.....	pág. 13
1.5.2. Suposiciones de la instalación	pág. 15
1.5.3. Descripción del PLC.....	pág. 15
1.5.3.1. CPS 2020	pág. 16
1.5.3.2. P34 2020	pág. 16
1.5.3.3. DDI 1602	pág. 18
1.5.3.4. DDM 16022	pág. 19
1.5.4. HMI STU 655 MAGELIS.....	pág. 19
1.5.4.1. Manejo del panel LCD.....	pág. 19
1.5.4.2. Identificación y funciones de las piezas	pág. 20
1.6. GRAFCET.....	pág. 22
1.6.1. Red de Petri	pág. 22
1.6.1.1. Propiedades estáticas o estructurales	pág. 23
1.6.1.2. Propiedades dinámicas o del comportamiento	pág. 24
1.6.1.3. Reglas de reducción.....	pág. 24
1.6.2. Introducción histórica.	pág. 25
1.6.3. Principios del GRAFCET	pág. 26
1.6.4. Niveles del GRAFCET	pág. 27
1.6.4.1. Nivel 1: Descripción funcional.....	pág. 27
1.6.4.2. Nivel 2: Descripción tecnológica	pág. 27
1.6.4.3. Nivel 3: Descripción operativa	pág. 28
1.6.5. Estructuras lógicas fundamentales.....	pág. 29
1.6.5.1. Secuencia lineal.....	pág. 29
1.6.5.2. Bifurcación en O	pág. 30
1.6.5.3. Bifurcación en Y.....	pág. 32
1.6.6. Reglas de evolución	pág. 33

1.6.6.1. Inicialización	pág. 33
1.6.6.2. Evolución de las transiciones	pág. 34
1.6.6.3. Evolución de las etapas activas	pág. 34
1.6.6.4. Simultaneidad en el franqueamiento de las transiciones... ..	pág. 35
1.6.6.5. Prioridad de la activación	pág. 35
1.6.7. Macroetapas.....	pág. 35
1.6.8. GRAFCET jerarquizado	pág. 36
1.7. Guía GEMMA.....	pág. 38
1.7.1. Presentación de la Guía GEMMA	pág. 38
1.7.2. Descripción de la Guía GEMMA.....	pág. 38
1.7.2.1. Grupo F: Procedimientos de funcionamiento	pág. 40
1.7.2.2. Grupo A: Procedimiento de paradas y puestas en marcha. ..	pág. 41
1.7.2.3. Grupo D: Procedimientos de defecto.....	pág. 42
1.7.3. Método de aplicación	pág. 42
1.8. Unity Pro.....	pág. 44
1.8.1. Crear y configurar un nuevo proyecto.....	pág. 44
1.8.2. Interfaz de usuario	pág. 45
1.8.3. Variables.....	pág. 46
1.8.3.1. Variables elementales	pág. 47
1.8.3.2. Variables derivadas	pág. 48
1.8.3.3. Variables de E/S derivadas.....	pág. 48
1.8.3.4. Instancias FB elementales.....	pág. 48
1.8.3.5. Instancias FB derivadas.....	pág. 48
1.8.4. Estructura de la aplicación	pág. 49
1.8.4.1. Tarea maestra	pág. 49
1.8.4.2. Tarea rápida	pág. 50
1.8.4.3. Tarea auxiliar.....	pág. 50
1.8.4.4. Tarea de evento	pág. 50
1.8.4.4.1. Evento de temporizador	pág. 51
1.8.4.4.2. Evento de E/S	pág. 53
1.8.5. Lenguajes de programación.....	pág. 54
1.8.6. Configuración de una red Ethernet	pág. 55
1.9. Vijeo Designer	pág. 58
1.9.1. Crear un nuevo proyecto	pág. 58
1.9.2. Interfaz de usuario	pág. 60
1.9.3. Comunicaciones	pág. 61
1.9.4. Paneles.....	pág. 62
1.9.5. Variables....	pág. 62
1.9.6. Animación de gráficos	pág. 63
1.9.7. Visualización e introducción de datos	pág. 65
1.9.8. Ventanas emergentes.....	pág. 65

1.9.9. Alarmas.....	pág. 66
1.10. Bibliografía.....	pág. 68

Índice de los Anexos.....	pág. 70
---------------------------	---------

2.1. Guía GEMMA: Automatización del almacenador de bandejas	pág. 71
2.1.1. Grupo f: Procedimiento de funcionamiento	pág. 71
2.1.2. Grupo A: Procedimiento de paradas y puesta en marcha	pág. 72
2.1.3. Grupo D: Procedimiento de defecto.....	pág. 73
2.2. Programar PLC mediante el programa Unity Pro	pág. 74
2.2.1. Crear el proyecto.	pág. 74
2.2.2. Configurar el hardware.	pág. 74
2.2.3. Configurar la comunicación Ethernet	pág. 76
2.2.4. Lista de señales internas y externas	pág. 76
2.2.5. PE	pág. 79
2.2.6. GEMMA	pág. 82
2.2.7. MA.....	pág. 84
2.2.8. MP.....	pág. 91
2.2.9. PEI	pág. 95
2.2.10. PB	pág. 96
2.2.11. NBANDJ	pág. 99
2.2.12. FALLO	pág. 100
2.2.13. FCFRP Y FCFRE	pág. 101
2.2.14. AERP Y AERE	pág. 103
2.2.15. FCFCE Y FCFCG	pág. 104
2.2.16. AECE	pág. 106
2.2.17. AECG.....	pág. 107
2.2.18. ITP	pág. 109
2.2.19. COM_ETH	pág. 110
2.3. Programar Pantalla Táctil mediante el programa Vijeo Designer	pág. 112
2.3.1. Crear el proyecto	pág. 112
2.3.2. Comunicación con un PLC a través de TCP/IP	pág. 112
2.3.3. Lista de variables	pág. 113
2.3.4. Pantalla Principal	pág. 114
2.3.5. Pantalla Guía GEMMA	pág. 115
2.3.6. Pantalla de información del almacén	pág. 116
2.3.7. Pantalla panel de control	pág. 117
2.3.8. Pantalla de alarmas.....	pág. 118

ÍNDICE DE FIGURAS

Figura 1.1-1: Plano de la célula de fabricación	pág. 5
Figura 1.5-1: Alimentador de Bandejas.....	pág. 9
Figura 1.5-2: Panel de Control	pág. 11
Figura 1.5-3: Identificación de los elementos del panel de control.....	pág. 13
Figura 1.5.-4: Ilustración del módulo CPU BMX P34 2020	pág. 17
Figura 1.5-5: Pantalla de indicadores LED del módulo P34 2020	pág. 17
Figura 1.5-6: Conexión del módulo DDI 1602 a los sensores	pág. 18
Figura 1.5-7: Conexión del módulo DDM 16022 a sensores y pre-actuadores	pág. 19
Figura 1.5-8: Frontal del módulo de la Pantalla	pág. 20
Figura 1.5-9: Conectores parte inferior del dispositivo HMI	pág. 21
Figura 1.5-10: Conector lateral del dispositivo HMI	pág. 21
Figura 1.6-1: Red de Petri.....	pág. 22
Figura 1.6-2: Fusión de etapas en serie.....	pág. 25
Figura 1.6-3: Fusión de transiciones en serie	pág. 25
Figura 1.6-4: Fusión de transiciones paralelas	pág. 25
Figura 1.6-5: Fusión de etapas paralelas.....	pág. 25
Figura 1.6-6: Eliminación de auto bucles en etapas	pág. 25
Figura 1.6-7: Eliminación de auto bucles en transición	pág. 25
Figura 1.6-8: Elementos del GRAFCET.....	pág. 26
Figura 1.6-9: GRAFCET de nivel 1.....	pág. 27
Figura 1.6-10: GRAFCET de nivel 2.....	pág. 28
Figura 1.6-11: GRAFCET de nivel 3.....	pág. 28
Figura 1.6-12: Secuencia lineal	pág. 29
Figura 1.6-13: Divergencia en O	pág. 30
Figura 1.6-14: Modelado tradicional de exclusión lógica.....	pág. 30
Figura 1.6-15: Convergencia en O.....	pág. 31
Figura 1.6-16: Salto condicional	pág. 31
Figura 1.6-17: Repetición de secuencias.....	pág. 32
Figura 1.6-18: Divergencia en Y	pág. 32
Figura 1.6-19: Convergencia en Y	pág. 33
Figura 1.6-20: Posibles situaciones de una transición	pág. 34
Figura 1.6-21: Descomposición de GRAFCET en dos GRAFCET independientes	pág. 35
Figura 1.6-22: Ilustración de una macroetapa y su GRAFCET parcial expandido	pág. 36
Figura 1.6-23: Representación de acción de forzado.....	pág. 37
Figura 1.7-1: Modos de funcionamiento de la Guía GEMMA	pág. 39
Figura 1.7-2: Estados de la Guía GEMMA.....	pág. 39
Figura 1.8-1: Crear nuevo proyecto de Unity Pro	pág. 44
Figura 1.8-2: Seleccionar CPU.....	pág. 44
Figura 1.8-3: Configuración del PLC.....	pág. 45
Figura 1.8-4: Interfaz de Usuario de Unity Pro	pág. 45
Figura 1.8-5: Familias de tipos de datos de Unity Pro	pág. 46
Figura 1.8-6: Crear un bloque DFB.....	pág. 49
Figura 1.8-7: Nivel de prioridad de las tareas de Unity Pro.....	pág. 49

Figura 1.8-8: Crear evento de temporizador	pág. 51
Figura 1.8-9: Configuración del temporizador del evento de tiempo	pág. 51
Figura 1.8-10: Representación FBD de la función ITCNTRL.....	pág. 52
Figura 1.8-11: Tendencias de funcionamiento normal de eventos de temporizador ..	pág. 53
Figura 1.8-12: Crear Evento E/S.....	pág. 53
Figura 1.8-13: Configurar Evento de E/S	pág. 54
Figura 1.8-14: Crear nueva sección.....	pág. 54
Figura 1.8-15: Configurar nueva sección.....	pág. 55
Figura 1.8-16: Orden de ejecución de las secciones	pág. 55
Figura 1.8-17: Crear red Ethernet.....	pág. 56
Figura 1.8-18: Vincular red Ethernet al módulo de comunicación	pág. 56
Figura 1.8-19: Bloque de función ADDM.....	pág. 56
Figura 1.8-20: Bloque de función READ_VAR.....	pág. 57
Figura 1.9-1: Crear un nuevo proyecto de Vijeo Designer	pág. 58
Figura 1.9-2: Configuración del proyecto de Vijeo Designer	pág. 59
Figura 1.9-3: Configuración del proyecto de Vijeo Designer (continuación)	pág. 59
Figura 1.9-4: Interfaz de usuario de Vijeo Designer.....	pág. 60
Figura 1.9-5: Añadir un nuevo controlador	pág. 61
Figura 1.9-6: Crear nuevo panel en Vijeo Designer	pág. 62
Figura 1.9-7: Ventana Inspector de propiedades	pág. 62
Figura 1.9-8: Animación de color.....	pág. 64
Figura 1.9-9: Animación de relleno.....	pág. 64
Figura 1.9.10: Animación de valor	pág. 64
Figura 1.9.11: Visualización de datos en Runtime	pág. 65
Figura 1.9-12: Componentes de las ventanas emergentes	pág. 65
Figura 1.9.13: Visualización de Resumen de alarmas	pág. 66
Figura 1.9.14: Aviso de alarma	pág. 67
Figura 1.9.15: Alarma de sonido.....	pág. 67
Figura 2.1-1: Guía GEMMA del sistema	pág. 71
Figura 2.2-1: Creación del proyecto.....	pág. 74
Figura 2.2-2: Ventana de configuración del PLC.....	pág. 75
Figura 2.2-3: Ventana Nuevo dispositivo	pág. 75
Figura 2.2-4: Composición del PLC.....	pág. 75
Figura 2.2-5: SECCIÓN SFC (A) Y GRAFCET PE (B).....	pág. 79
Figura 2.2-6: Transición T40 de la sección PE	pág. 80
Figura 2.2-7: Bloque de función CLEARCHART	pág. 80
Figura 2.2-8: Transición T41 de la sección PE.....	pág. 80
Figura 2.2-9: Bloque de función INITCHART	pág. 81
Figura 2.2-10: Bloque de función SETSTEP	pág. 81
Figura 2.2-11: SECCIÓN GEMMA	pág. 82
Figura 2.2-12: GRAFCET GEMMA	pág. 82
Figura 2.2-13: Transición T30 de la sección GEMMA.....	pág. 83
Figura 2.2-14: Transición T32 de la sección GEMMA.....	pág. 83
Figura 2.2-15: Transición T35 de la sección GEMMA.....	pág. 84
Figura 2.2-16: SECCIÓN MA.....	pág. 85

Figura 2.2-17: GRAFCET MA	pág. 85
Figura 2.2-18: Transición T1 de la sección MA	pág. 85
Figura 2.2-19: Transición T2 de la sección MA	pág. 86
Figura 2.2-20: Transiciones T3 y T12 de la sección MA.....	pág. 86
Figura 2.2-21: Transiciones T4 y T13 de la sección MA.....	pág. 86
Figura 2.2-20: Transiciones T7 y T16 de la sección MA.....	pág. 85
Figura 2.2-21: Transiciones T6 y T15 de la sección MA.....	pág. 87
Figura 2.2-22: Transiciones T5 y T14 de la sección MA.....	pág. 87
Figura 2.2-23: Transiciones T8 y T17 de la sección MA.....	pág. 88
Figura 2.2-24: Transición T66 de la sección MA	pág. 88
Figura 2.2-25: Transición T11 de la sección MA	pág. 88
Figura 2.2-26: Transición T18 de la sección MA	pág. 88
Figura 2.2-27: Transición T19 de la sección MA	pág. 89
Figura 2.2-28: Transición T20 de la sección MA	pág. 89
Figura 2.2-29: Transición T21 de la sección MA	pág. 89
Figura 2.2-30: Transición T115 de la sección MA	pág. 89
Figura 2.2-31: SECCIÓN MP	pág. 91
Figura 2.2-32: GRAFCET MP	pág. 92
Figura 2.2-33: Obtención de duración de activación de una etapa	pág. 94
Figura 2.2-34: Incrementar variables TIME	pág. 94
Figura 2.2-35: SECCIÓN PEI.....	pág. 95
Figura 2.2-36: GRAFCET PEI	pág. 95
Figura 2.2-37: SECCIÓN PB	pág. 96
Figura 2.2-38: GRAFCET PB.....	pág. 96
Figura 2.2-39: Transición T43 de la sección PB.....	pág. 97
Figura 2.2-40: Transición T67 de la sección PB.....	pág. 97
Figura 2.2-41: Transición T48 de la sección PB.....	pág. 98
Figura 2.2-42: Transición T70 de la sección PB.....	pág. 98
Figura 2.2-43: Transición T155 de la sección PB.....	pág. 98
Figura 2.2-44: Transición T157 de la sección PB	pág.98
Figura 2.2-45: SECCIÓN NBANDEJ.....	pág. 99
Figura 2.2-46: GRAFCET NBANDEJ	pág.99
Figura 2.2-47: SECCIÓN FALLO.....	pág. 100
Figura 2.2-48: GRAFCET FALLO	pág. 100
Figura 2.2-49: Transición T72 de la sección FALLO	pág. 100
Figura 2.2-50: Transición T73 de la sección FALLO	pág. 101
Figura 2.2-51: SECCIONES FCFRP y FCFRE	pág. 102
Figura 2.2-52: GRAFCETs FCFRP Y FCFRE	pág. 102
Figura 2.2-53: Ejemplo función ITCNTRL.....	pág. 103
Figura 2.2-53: SECCIONES AERP y AERE	pág. 103
Figura 2.2-54: GRAFCETs AERP y AERE.....	pág. 103
Figura 2.2-55: Sección del evento temporizador.....	pág. 104
Figura 2.2-56: SECCIONES FCFCE y FCFCG.....	pág. 104
Figura 2.2-57: GRAFCETs FCFCE y FCFCG	pág. 105

Figura 2.2-58: SECCIÓN AECE	pág. 106
Figura 2.2-59: GRAFCET AECE	pág. 106
Figura 2.2-60: SECCIÓN AECG.....	pág. 107
Figura 2.2-61: GRAFCET AECG	pág. 108
Figura 2.2-62: SECCIÓN ITP	pág. 109
Figura 2.2-63: GRAFCET ITP	pág. 109
Figura 2.2-64: Diagrama LD de lectura de variables remota.....	pág. 110
Figura 2.3-1: Selección del dispositivo HMI.....	pág. 112
Figura 2.3-2: Añadir un nuevo controlador	pág. 113
Figura 2.3-3: Selección del controlador de comunicación	pág. 113
Figura 2.3-3: Selección del controlador de comunicación	pág. 114
Figura 2.3-4: Pantalla Principal.....	pág. 109
Figura 2.3-5: Crear botón en la pantalla	pág. 114
Figura 2.3-6: Configurar botón	pág. 115
Figura 2.3-7: Pantalla Guía GEMMA	pág. 115
Figura 2.3-8: Animación de relleno.....	pág. 116
Figura 2.3-9: Pantalla Información del Almacén.....	pág. 116
Figura 2.3-10: Configurar visualizador numérico.....	pág. 117
Figura 2.3-11: Configurar gráfica de barra	pág. 117
Figura 2.3-12: Pantalla Panel de Control.....	pág. 118
Figura 2.3-13: Pantalla de alarmas	pág. 118
Figura 2.3-14: Añadir variable a grupo de alarma desde el editor de variables.....	pág. 119
Figura 2.3-15: Agregar variable a grupo de variable desde la ventana de alarmas.....	pág. 119

ÍNDICE DE TABLAS

Tabla 1.5-1: Actuadores del sistema.....	pág. 9
Tabla 1.5-2: Sensores del sistema.....	pág. 10
Tabla 1.5-3: Función LEDs del módulo P34 2020.....	pág. 18
Tabla 1.8-1: Ejemplos valor de variable tipo Time.....	pág. 47
Tabla 1.8-2: Parámetros de entrada de la función ITCNTRL.....	pág. 52
Tabla 1.8-3: Parámetros de salida de la función ITCNTRL.....	pág. 53
Tabla 1.9-1: Variables elementales de Vijeo Designer.....	pág. 63
Tabla 2.2-1: Señales del módulo DDI 1602.....	pág. 77
Tabla 2.2-2: Señales del módulo DDM 16022	pág. 77
Tabla 2.2-3: Señales internas del programa Unity Pro.....	pág. 78
Tabla 2.2-4: Señales internas del programa Unity Pro (continuación)	pág. 78
Tabla 2.2-5: Señales internas del programa Unity Pro (continuación 2)	pág. 79
Tabla 2.2-6: Tiempos de los actuadores	pág. 93

Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías
Industriales

MEMORIA

Autor: Antonio Luis Velasco Martín

Tutor: Luis Fernando Castaño Castaño

Dep. Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla



Sevilla, 2015



ÍNDICE DE LA MEMORIA

1.1. Introducción.....	pág. 4
1.2. Antecedentes.....	pág. 5
1.3. Objetivos.....	pág. 7
1.4. Definiciones y abreviaturas	pág. 8
1.5. Descripción general.....	pág. 9
1.5.1. Especificaciones de la instalación.....	pág. 13
1.5.2. Suposiciones de la instalación	pág. 15
1.5.3. Descripción del PLC.....	pág. 15
1.5.3.1. CPS 2020	pág. 16
1.5.3.2. P34 2020	pág. 16
1.5.3.3. DDI 1602	pág. 18
1.5.3.4. DDM 16022	pág. 19
1.5.4. HMI STU 655 MAGELIS.....	pág. 19
1.5.4.1. Manejo del panel LCD.....	pág. 19
1.5.4.2. Identificación y funciones de las piezas	pág. 20
1.6. GRAFCET.....	pág. 22
1.6.1. Red de Petri	pág. 22
1.6.1.1. Propiedades estáticas o estructurales	pág. 23
1.6.1.2. Propiedades dinámicas o del comportamiento	pág. 24
1.6.1.3. Reglas de reducción.....	pág. 24
1.6.2. Introducción histórica.	pág. 25
1.6.3. Principios del GRAFCET	pág. 26
1.6.4. Niveles del GRAFCET	pág. 27
1.6.4.1. Nivel 1: Descripción funcional.....	pág. 27
1.6.4.2. Nivel 2: Descripción tecnológica	pág. 27
1.6.4.3. Nivel 3: Descripción operativa	pág. 28
1.6.5. Estructuras lógicas fundamentales.....	pág. 29
1.6.5.1. Secuencia lineal.....	pág. 29
1.6.5.2. Bifurcación en O	pág. 30
1.6.5.3. Bifurcación en Y.....	pág. 32
1.6.6. Reglas de evolución	pág. 33
1.6.6.1. Inicialización	pág. 33
1.6.6.2. Evolución de las transiciones	pág. 34
1.6.6.3. Evolución de las etapas activas.....	pág. 34
1.6.6.4. Simultaneidad en el franqueamiento de las transiciones. ...	pág. 35
1.6.6.5. Prioridad de la activación	pág. 35
1.6.7. Macroetapas.....	pág. 35
1.6.8. GRAFCET jerarquizado	pág. 36
1.7. Guía GEMMA.....	pág. 38

1.7.1.	Presentación de la Guía GEMMA	pág. 38
1.7.2.	Descripción de la Guía GEMMA.....	pág. 38
1.7.2.1.	Grupo F: Procedimientos de funcionamiento	pág. 40
1.7.2.2.	Grupo A: Procedimiento de paradas y puestas en marcha.	pág. 41
1.7.2.3.	Grupo D: Procedimientos de defecto.....	pág. 42
1.7.3.	Método de aplicación	pág. 42
1.8.	Unity Pro.....	pág. 44
1.8.1.	Crear y configurar un nuevo proyecto.....	pág. 44
1.8.2.	Interfaz de usuario	pág. 45
1.8.3.	Variables.....	pág. 46
1.8.3.1.	Variables elementales	pág. 47
1.8.3.2.	Variables derivadas	pág. 48
1.8.3.3.	Variables de E/S derivadas.....	pág. 48
1.8.3.4.	Instancias FB elementales.....	pág. 48
1.8.3.5.	Instancias FB derivadas.....	pág. 48
1.8.4.	Estructura de la aplicación	pág. 49
1.8.4.1.	Tarea maestra	pág. 49
1.8.4.2.	Tarea rápida	pág. 50
1.8.4.3.	Tarea auxiliar.....	pág. 50
1.8.4.4.	Tarea de evento	pág. 50
1.8.4.4.1.	Evento de temporizador	pág. 51
1.8.4.4.2.	Evento de E/S	pág. 53
1.8.5.	Lenguajes de programación.....	pág. 54
1.8.6.	Configuración de una red Ethernet	pág. 55
1.9.	Vijeo Designer	pág. 58
1.9.1.	Crear un nuevo proyecto	pág. 58
1.9.2.	Interfaz de usuario	pág. 60
1.9.3.	Comunicaciones	pág. 61
1.9.4.	Paneles.....	pág. 62
1.9.5.	Variables.....	pág. 62
1.9.6.	Animación de gráficos	pág. 63
1.9.7.	Visualización e introducción de datos.....	pág. 65
1.9.8.	Ventanas emergentes.....	pág. 65
1.9.9.	Alarmas.....	pág. 66
1.10.	Bibliografía.....	pág. 68

1.1. Introducción

En este proyecto, “Integración de alimentador de bandejas a célula de fabricación flexible controlada por PLC”, se realiza la automatización del almacenador de bandejas, que forma parte de una célula de fabricación ubicada en el departamento de Ingeniería de Sistemas y Automática de la Universidad de Sevilla, usando el programa Unity Pro de Schneider Electric. En este programa se hará uso de eventos y GRAFCETs jerarquizados, además se cuenta con una pantalla táctil, MAGELIS STU 655, que se programará usando Vijeo Designer, con la que el operador podrá interactuar con la máquina y conocer el estado en la que esta se encuentra.

1.2. Antecedentes

Como se menciona en la introducción, en el departamento de Ingeniería de Sistemas y Automática se ubica una célula de fabricación flexible que constaba con cuatro cintas transportadoras que permiten transportar bandejas. Dichas bandejas pueden portar pallets con piezas que pueden ser tratadas por diferentes máquinas en diferentes puntos de trabajo alrededor del circuito. Los trabajos realizados sobre las piezas pueden ser muy distintos y abarcan desde la clasificación de piezas, el ensamblado, control de calidad....

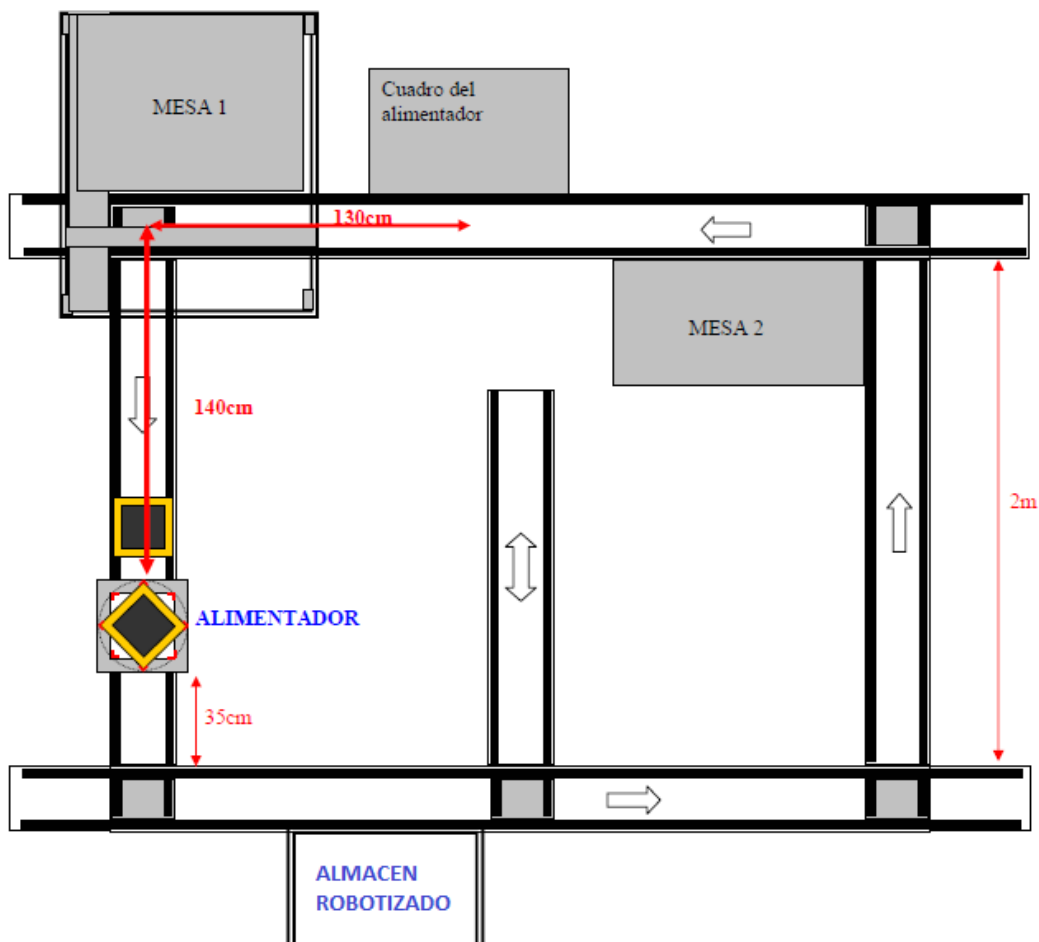


Figura 1.1-1: Plano de la célula de fabricación

Para el almacenamiento de los pallets, la célula contaba con un almacén robotizado de pallets con dos matrices de celdas de almacenaje. Sin embargo, no se disponía de ninguna maquina equivalente para las bandejas, por lo que la retirada y colocación de bandejas se hacía de forma manual, según las necesidades de la aplicación que se estuviera ejecutando en ese momento. Con el fin de evitar la manipulación manual se diseña el alimentador de bandejas, cuya función principal es

realizar una gestión automática de las bandejas, sirviéndolas o almacenándolas según las necesidades de la instalación principal.

Esta nueva máquina aporta una serie de objetivos didácticos que no se contemplaban en el diseño inicial, como:

1. **Cuadro de maniobras más complejo:** incluye selección de modo manual/automático, semiautomático, seta de emergencia, forzar fallo en sensores y actuadores, leds para indicación de los sensores y actuadores activos...
2. **Pantalla táctil:** que permite conocer otro dispositivo HMI.
3. **Instalación de seguridad:** esta máquina consta de mamparas de seguridad móviles, dispositivo anti-aprisionamiento y setas de emergencia a pie de máquina y en el panel de control.
4. **Gestión de fallos:** permite forzar fallos en las electroválvulas y posee sensores para poder confirmar su correcto funcionamiento.

1.3. Objetivos

Siguiendo la filosofía con la que se diseñó la máquina de la que se va a realizar la automatización, este proyecto tendrá un carácter principalmente didáctico, pudiéndose distinguir dos partes fundamentales: una primera parte en la que se hará una descripción general de las principales herramientas que se han usado para la realización de este proyecto, y una segunda parte en la que se aplican estos conocimientos al caso que nos ocupa. En él se aplicarán algunos conceptos vistos durante el desarrollo de los cursos, pero se hará especial hincapié en aspectos relacionados con el desarrollo de un automatismo que no han sido tratados.

Entre los aspectos que se tratarán en este proyecto destacan:

- La utilización de eventos para detectar el correcto funcionamiento de los sensores y actuadores de la máquina.
- La utilización de GRAFCET jerarquizados que nos permite forzar los estados activos de un GRAFCET a partir de un tercero.
- Estudiar la comunicación entre autómatas Modicon M340.
- Conocer un método de comunicación HMI a través del desarrollo de una aplicación para una pantalla táctil.

1.4. Definiciones y abreviaturas

- Actuador: Dispositivo encargado de realizar el control de algún elemento del sistema. Por ejemplo, motores, electroválvulas, variadores de frecuencia...
- Sensor: Dispositivo encargado de captar información del sistema. Por ejemplo, detectores fotoeléctricos, finales de carrera, etc.
- Alarma: Dispositivo o función que indica la existencia de una condición anormal.
- PLC: Controlador Lógico Programable. Es un controlador dotado normalmente de múltiples entradas y salidas, el cual tiene un programa que puede ser modificado.
- Grafcet: Graphe de Comanden Etape-Transition. Método gráfico que permite el estudio de un automatismo de forma rigurosa y fácil de aplicar a la industria.
- Guía Gemma: Guide de Etude des Modes des Marches et de Arrêts. Complemento organizativo del Grafcet. Guía de estudio de los modos de marcha y parada.
- Red de Petri pura: se dice que una red es pura cuando no tiene auto bucles.
- Auto Bucle: está definido por el conjunto de una etapa y una transición, donde la etapa es simultáneamente etapa de entrada y salida de la transición.
- CPU: Unidad Central de Procesos
- TFT: Tecnología Thin-Film Transistor

1.5. Descripción General

El alimentador de bandejas está compuesto de un total de 4 cilindros y 11 sensores.

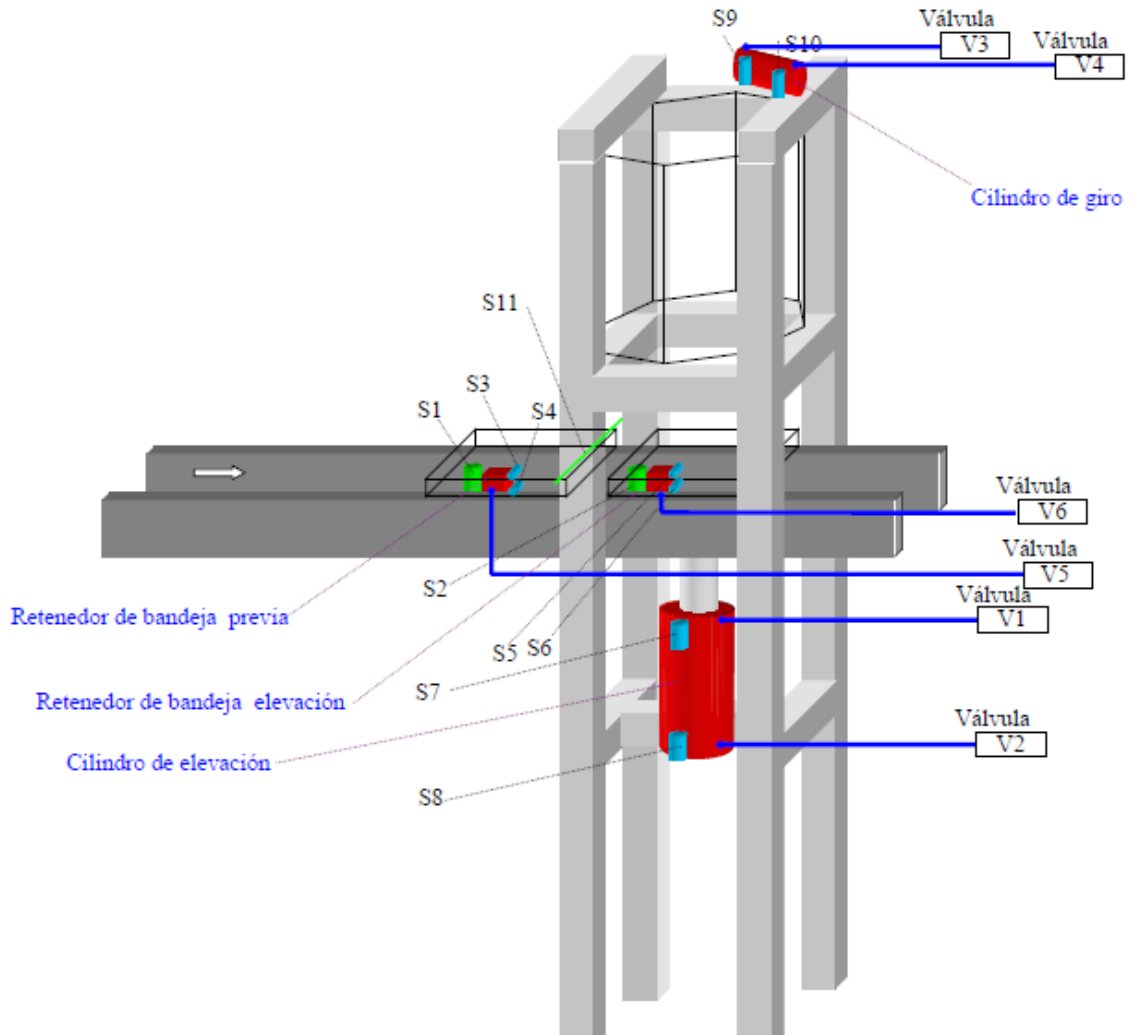


Figura 1.5-1: Alimentador de Bandejas

1	BAJAR CILINDRO ELEVADOR	V1
2	SUBIR CILINDRO ELEVADOR	V2
3	ROTACIÓN SENTIDO HORARIO CILINDRO GIRO	V3
4	ROTACIÓN SENTIDO ANTIHORARIO CILINDRO GIRO	V4
5	RETENEDOR BANDEJA PREVIA	V5
6	RETENEDOR BANDEJA ELEVACIÓN	V6

Tabla 1.5-1: Actuadores del sistema

Cuenta con una electroválvula neumática de dos posiciones que mueve el cilindro encargado de elevar las bandejas vacías hasta el alimentador (V2) o depositarlas en la cinta para que comience el recorrido por el circuito (V1). También hay instalado otra electroválvula neumática de dos posiciones que hace girar el alimentador para dejar la bandeja que se sube almacenada o liberar una bandeja que ya se encuentra almacenada y poder servirla (V3 Y V4), en este último caso es necesario que el cilindro elevador esté en la posición superior para evitar que la bandeja caiga.

Además de estos dos actuadores, también cuenta con otras dos electroválvulas neumáticas de una posición encargadas de activar los retenedores de bandeja que se aproximan al alimentador. El primer retenedor (V5) parará la bandeja, ya esté vacía o no, en el caso de que el cilindro elevador no esté en la posición baja para evitar que puedan amontonarse las bandejas. Mientras que el segundo retenedor (V6) sólo se activará si la bandeja se tiene que almacenar o para retener una bandeja que se está sirviendo hasta que el cilindro retenedor esté totalmente bajado.

1	SENSOR DE BANDEJA PREVIA	S1
2	SENSOR DE BANDEJA DE ELEVACIÓN	S2
3	SENSOR ALTO RETENEDOR PREVIO	S3
4	SENSOR BAJO RETENEDOR PREVIO	S4
5	SENSOR ALTO RETENEDOR ELEVACIÓN	S5
6	SENSOR BAJO RETENEDOR ELEVACIÓN	S6
7	SENSOR SUPERIOR CILINDRO ELEVADOR	S7
8	SENSOR INFERIOR CILINDRO ELEVACIÓN	S8
9	SENSOR IZQUIERDO CILINDRO GIRO	S9
10	SENSOR DERECHO CILINDRO GIRO	S10
11	SENSOR DE BARRERA	S11

Tabla 1.5-2: Sensores del sistema

La máquina también posee 8 sensores (S3-S10), dos por cada actuador, para detectar en qué posición se encuentran, alta o baja, dos sensores inductivos (S1 y S2) que se activan cuando una bandeja pasa por la posición en la que están ubicadas, ya que estas bandejas tienen una pieza metálica, y un último sensor de barrera (S11) que se activa si la bandeja porta un pallet y por tanto no se podría almacenar.

Como se menciona en el apartado 1.2, antecedentes, esta máquina cuenta con un panel de control más complejo que los que se encontraban en la célula hasta el momento. En el que se encuentra una pantalla táctil, varios pulsadores, interruptores e indicaciones luminosas que se detallarán a continuación.

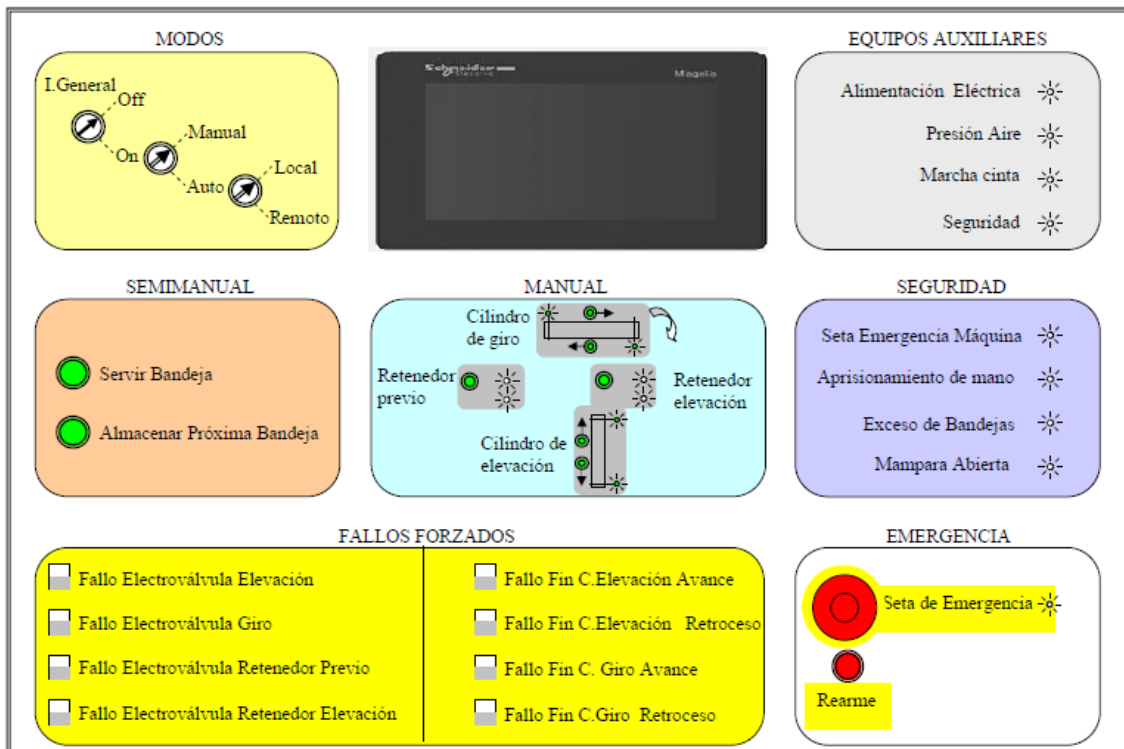


Figura 1.5-2: Panel de Control

Como se ve en la figura 1.5-3, en este panel de control podemos encontrar tres conjuntos de leds. Un grupo indica al operario si los equipos auxiliares están conectados correctamente (L1 aL4), un segundo grupo de leds se activan en caso de fallo de seguridad para que el operador vea de que se trata y pueda solucionarlo (L5 a L9) y un tercer grupo que muestra en qué posición se encuentra cada electroválvula (L10 a L17).

Dentro del primer grupo de indicadores luminosos tenemos cuatro leds. El primero nos indica si la máquina tiene alimentación eléctrica (L1), el segundo indica si hay presión de aire (L2) en el circuito neumático en el que están conectadas las electroválvulas, el tercer led muestra si el motor que pone las cintas transportadoras en marcha está funcionando (L3) y el último led de este grupo nos muestra que hay algún problema de seguridad (L4) que impide poner en marcha la máquina.

El segundo grupo de leds indican el motivo de que el led L4 esté encendido, es decir, el motivo del fallo de seguridad. Este fallo puede producirse por varios motivos: porque se haya pulsado la seta de emergencia ubicada en el alimentador de bandejas (L5), en este caso el operario, una vez que se ha asegurado que se ha eliminado el problema que provocó que se pulsara la seta de emergencia, desactivará la seta; también puede producirse un problema de seguridad si el pulsador anti-

aprisionamiento, que es un dispositivo que se encuentra en el recorrido del cilindro elevador a la altura del almacén y evita que algún operario pueda quedar atrapado, se activa (L6), en este caso debe retirarse el objeto o la mano del operario que lo ha provocado; otro motivo sería que se pulsara el pulsador que está ubicado en el techo del almacén indicando un exceso de bandejas (L7), en el almacén cabe un máximo de 10 bandejas y si se intenta introducir una cantidad mayor contamos con este dispositivo de seguridad para no dañar la máquina, la solución a este problema radica en que un operador retire una bandeja del almacén de forma manual; si la mampara está abierta (L8) no funcionará la máquina para evitar que alguien pueda quedar atrapado y sufrir daños; y por último se puede producir porque alguien pulse la seta de emergencia del panel de control (L9), esto se solucionará de forma análoga al caso de la seta de emergencia que está en la máquina.

El último grupo de leds indica que sensores de las electroválvulas están activos, es decir, en qué posición está cada cilindro. Esta información es esencial para que el operario sepa que movimientos tiene que mandar a cada electroválvula en el funcionamiento manual de la máquina, esto resulta especialmente útil para el cilindro de giro que hace rotar el almacén, liberando o reteniendo una bandeja, ya que aunque la instalación se puede ver desde el lugar en el que está el panel de control, no se puede saber a simple vista en qué posición está la electroválvula, y por tanto en qué sentido hay que girar. Los dos sensores inductivos, que indican si hay una bandeja en su ubicación, y el sensor de barrera, que indica al sistema si la bandeja está vacía, no tienen ninguna señal luminosa en el panel de control ya que, como se ha mencionado, desde este el operario puede ver si una bandeja está vacía o no y en qué lugar de la cinta se encuentra.

También posee tres selectores. El primero, y más importante, enciende la máquina (P0); el segundo sirve para elegir el modo de funcionamiento de la máquina (P1), modo manual o automático; y el tercero selecciona entre control local o remoto (P2), en el control remoto el PLC se conecta a través de Ethernet con el PLC principal, que le indica si debe servir una bandeja, mientras que en el control local el PLC funciona independientemente del resto de las máquinas.

Luego nos encontramos con los pulsadores del modo semiautomático y el manual. Hay un pulsador que realiza el ciclo completo de almacenar una bandeja (P4) y otro que realiza el ciclo completo de servir una bandeja (P5), estos dos pulsadores controlan el modo de funcionamiento de la máquina. También hay un pulsador para cada posición de todas las electroválvulas (Z1 a Z6), que son los que se usan en el modo manual para dar las órdenes a los actuadores.

Para mayor seguridad de la instalación y las personas que pueden interferir con esta, hay una seta de emergencia (P3). Una vez despejado la situación que provocó que se pulsara la seta de emergencia y esta deja de estar activa, hay un pulsador de rearme

(P6) en el panel de control que es necesario pulsar para que la máquina salga del estado de emergencia en el que se encuentra.

Por último nos encontramos con ocho interruptores para forzar fallos. Hay cuatro para forzar el fallo en las electroválvulas (F1 aF4) y otras cuatro para forzar el fallo en los finales de carrera del cilindro de elevación y del cilindro de giro.

Además de lo expuesto, el panel de control cuenta con una pantalla táctil que permite conocer una forma de interactuar entre un operario y el sistema que se está controlando, tanto modificando parámetros del sistema, como viendo el estado en el que este se encuentra o recibiendo mensajes que nos pueden indicar posibles error que servirán como apoyo al operario para realizar las reparaciones necesarias.

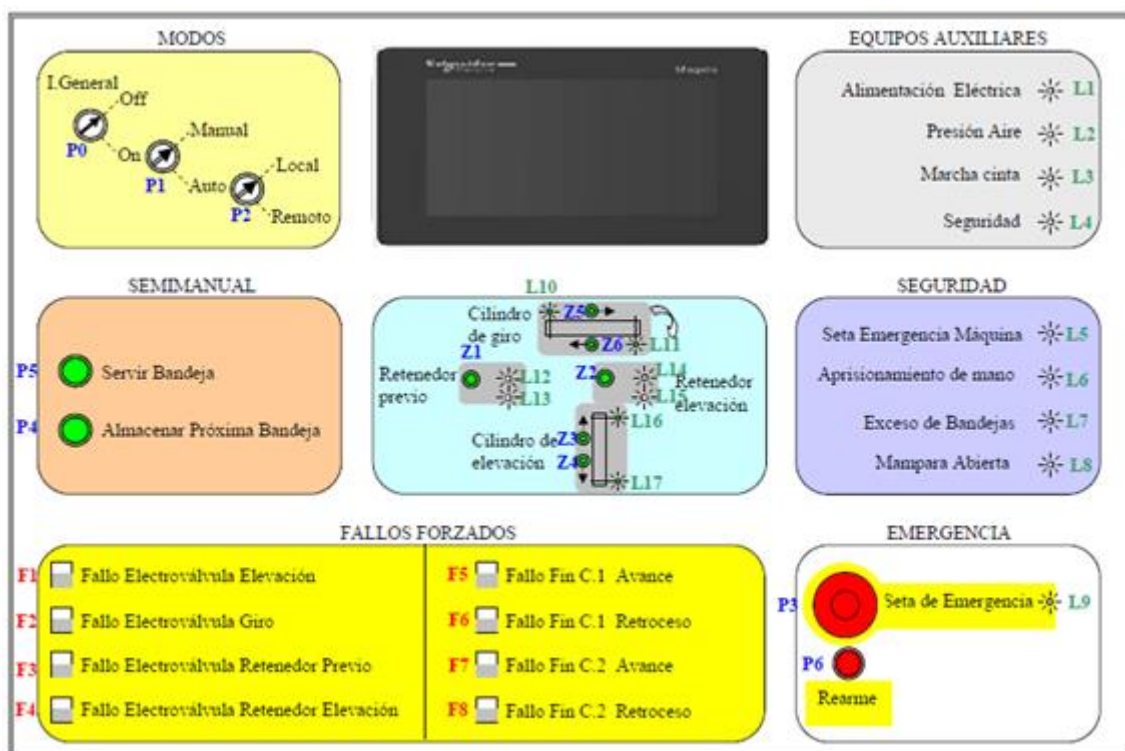


Figura 1.5-3: Identificación de los elementos del panel de control

1.5.1. Especificaciones de la instalación

A continuación se expondrán las especificaciones que deben cumplir el programa encargado de controlar el funcionamiento del almacén y la aplicación de la pantalla táctil:

- Cuando se apague la máquina, en modo automático, deben quedar todas las bandejas recogidas. En este caso no se aceptarán más peticiones de bandejas pero se seguirá operando hasta que todas hayan sido satisfecha.

- En el proceso de parada no será necesario que el autómatas principal de la orden de recoger bandejas. Y si habiendo peticiones pendientes de servir bandeja llega una al almacén que podría ser recogida, se dejará pasar asumiendo que se ha satisfecho una petición de servir bandeja pendiente.
- Si se apaga en modo manual, será el operario que esté manipulando la máquina el que deberá encargarse de que esta quede en un estado adecuado para posteriores usos.
- Toda petición de servir o almacenar una bandeja que se reciba durante el modo de funcionamiento automático debe ser cumplida, esto implica que si se solicita que se sirva o almacene una bandeja y en ese momento, por la razón que sea, no se puede satisfacer la demanda, ésta se llevará a cabo tan pronto como sea posible.
- Cada vez que se reciba la orden de almacenar la próxima bandeja, cuando una bandeja llegue al almacén, si está vacía y el almacén no está completa, esta debe ser recogida, sino no se hará nada para evitar errores de almacenamiento.
- Los pulsadores de recoger (P4) o servir (P5) una bandeja sustituirán las instrucciones del PLC principal, es decir, funcionarán como peticiones de servir bandeja o recoger próxima bandeja si el selector P1 está en la posición automática.
- En el modo manual no se atenderán peticiones de servir bandeja, si se desea bajar más de una bandeja, deberá realizarlo un operario con los controles independientes de cada actuador.
- En el modo manual, los pulsadores que realizan un ciclo completo, no tendrán ninguna función, ya que la máquina está cableada de forma que si el selector P1 está en posición manual, las señales de salida del PLC no llegarán a las electroválvulas.
- En la pantalla debe verse en qué estado de la Guía GEMMA se encuentra la ejecución del sistema en cada instante.
- Cuando se pase de modo de funcionamiento automático a manual, todas las peticiones pendientes de servir una bandeja se supondrán canceladas.

- El número de bandejas que se encuentran almacenada al comienzo de la ejecución serán indicadas al sistema a través de la aplicación de la pantalla táctil.
- En la pantalla se podrá ver el número de bandejas total que hay en el sistema, así como cuantas están almacenadas, además también se verá el número de peticiones pendientes que hay.
- Al comienzo de la ejecución automática, todas las bandejas deben estar almacenadas. Si posteriormente se quiere añadir o retirar una, se indicará a través de la pantalla táctil. Las bandejas incluidas durante la ejecución se colocarán en cualquier punto de la cinta fuera de la zona donde se encuentran los sensores del almacén.
- Cuando se produzca algún error, la pantalla mostrará un mensaje con el error detectado para ayudar a los operarios a resolverlo.

1.5.2. Suposición de la instalación

En este apartado se darán a conocer la hipótesis que se ha tenido en cuenta a la hora de desarrollar el programa de control:

- Cuando se produzca algún problema este será subsanado inmediatamente, por tanto se supone que no se producirá más de un fallo de sensores o actuadores simultáneamente.
- Durante el funcionamiento de la máquina se supone que no se modificarán las condiciones de presión del aire, a menos que haya algún fallo en el sistema neumático y provoque una caída de presión.

1.5.3. Descripción del PLC

Un autómata programable es una máquina electrónica preparada para realizar automatismos combinativos y secuenciales en tiempo real. Consta de tres partes fundamentales: la unidad de memoria, la unidad de control y elementos de entrada/salida.

La memoria sirve para almacenar el programa y los datos del proceso. Tiene una parte fija que es la del sistema operativo, encargado de leer las entradas y salidas, ejecutar el programa paso a paso..., viene programada de serie. La otra parte de la memoria puede modificarse en función del programa que se va a almacenar.

La unidad de control o CPU ejecuta las instrucciones del programa de forma cíclica. Primero lee las entradas, luego ejecuta el programa y modifica el valor de las salidas si es necesario, y vuelve a comenzar un nuevo ciclo.

Los elementos de entradas y salidas permiten comunicar el autómata con el proceso a controlar y con el operador. Las entradas dan información del estado del proceso y las salidas actúan sobre este.

1.5.3.1. CPS 2000

Los procesadores de plataforma automatizados M340 de Modicon gestionan toda la estación PLC, compuesta por módulos de E/S binarias y analógicas, módulos de conteo, módulos de comunicación... Cada bastidor debe tener un módulo de alimentación, en nuestro caso el módulo BMX CPS 2000, y si hay más de un bastidor, es en el bastidor principal donde se aloja la CPU.

El módulo de alimentación BMX CPS 2000 dispone en la parte frontal de un bloque de visualización, que cuenta con un piloto OK (encendido si hay tensión en los racks y es correcta) y un piloto de 24 V (encendido cuando la fuente suministra los 24 VCC), y un pulsador RESET, que provoca un arranque en frío (apagar/encender) de la aplicación.

1.5.3.2. P34 2020

El modulo CPU BMX P34 2020 es un procesador que se debe instalar siempre en la dirección 0 (ranura marcada como 00) de un bastidor BMX XBP. Este dispositivo cuenta con varios tipos de puertos, que serán detallados a continuación, y nos permite comunicar el PLC con otro PLC, ordenador o dispositivo.

La figura 1.5-4 muestra una representación de la CPU BMX P34 2020.

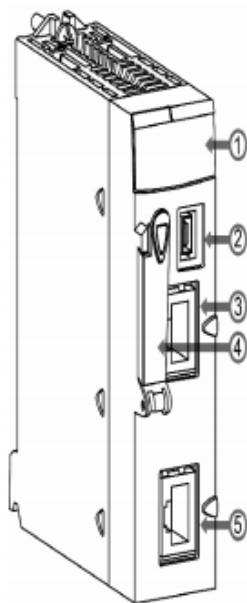


Figura 1.5-4: Ilustración del módulo CPU BMX P34 2020

- 1: Pantalla de indicadores LED. Los colores y los modelos de intermitencia de los LEDs indican el estado y las condiciones de funcionamiento de las comunicaciones Ethernet en el módulo. La figura 1.5-5 muestra los indicadores LEDs de la pantalla, hay dos versiones en función de si se usa la V1 o V2 (superior) del firmware del procesador, y en la tabla 1.5-3 se indica el significado de cada uno de los LEDs.

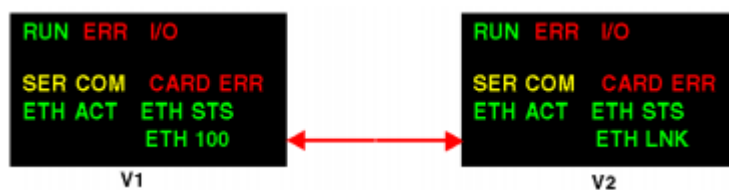


Figura 1.5-5: Pantalla de indicadores LED del módulo P34 2020

Etiqueta	Modelo	Indicación
RUN (verde): estado operativo	Encendido	<ul style="list-style-type: none"> Funcionamiento normal del hardware y software del PLC. El módulo se encuentra en estado RUN.
	Parpadeando	<ul style="list-style-type: none"> El PLC se encuentra en modalidad STOP o se ha detectado un error de bloqueo en la aplicación. El procesador está configurado pero no se encuentra en estado RUN.
	Apagado	PLC no configurado (aplicación ausente, no válida o incompatible).
ERR (rojo): error detectado	Encendido	Error detectado de procesador, sistema o configuración.
	Parpadeando	<ul style="list-style-type: none"> PLC no configurado (aplicación ausente, no válida o incompatible). El PLC se encuentra en modalidad STOP o se ha detectado un error de bloqueo en la aplicación.
	Apagado	Normal (sin errores detectados).
ETH STS (verde): estado de comunicación Ethernet	Encendido	Comunicación correcta.
	2 intermitencias	Dirección MAC no válida.
	3 intermitencias	Enlace sin conexión.
	4 intermitencias	Dirección IP duplicada.
	5 intermitencias	En espera de una dirección IP del servidor.
	6 intermitencias	Modalidad segura (con una dirección IP predeterminada).
	7 intermitencias	Conflicto de configuración entre los conmutadores rotatorios y la configuración interna.
	Apagado	Normal (sin errores detectados).
CARDERR (rojo): error detectado de tarjeta de memoria	Encendido	<ul style="list-style-type: none"> Falta la tarjeta de memoria. Tarjeta de memoria no utilizable (formato incorrecto, tipo no reconocido). Contenido de la tarjeta de memoria incoherente con la aplicación RAM interna.
	Apagado	<ul style="list-style-type: none"> Tarjeta de memoria detectada y válida. Aplicación de la tarjeta coherente con la aplicación RAM interna.
I/O (rojo): estado de entrada/salida	Encendido	<ul style="list-style-type: none"> Error detectado en un módulo configurado o en un canal de la CPU. Discrepancia en la configuración con la aplicación (módulo ausente...).
	Apagado	Normal (sin errores detectados).
SER COM (amarillo): estado de los datos serie	Parpadeando	Intercambio de datos (enviar/recibir) en curso en la conexión serie.
	Apagado	No se están intercambiando datos en la conexión serie.

Tabla 1.5-3: Función LEDs del módulo P34 2020

- 2: Puerto USB
- 3: Puerto Ethernet
- 4: Ranura de la tarjeta de memoria. Se utiliza para almacenar archivos como páginas web, archivos de registro código de copia de seguridad y aplicación de copia de seguridad.
- 5: Puerto serie Modbus

1.5.3.3. DDI 1602

El módulo BMX DDI 1602 es un módulo binario de 24 VCC a través de un bloque de terminales de 20 pins, de los cuales 16 son canales de entrada y los 4 restantes son de alimentación.

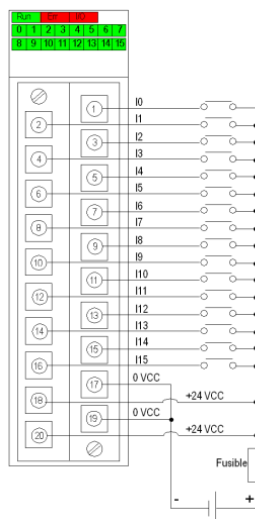


Figura 1.5-6: Conexión del módulo DDI 1602 a los sensores

1.5.3.4. DDM 16022

El módulo BMX DDM 16022 es un módulo binario de 24 VCC conectados a través de un bloque de terminales de 20 pins, de los cuales 8 son canales de entrada, 8 canales de salidas, y los 4 restantes son de alimentación.

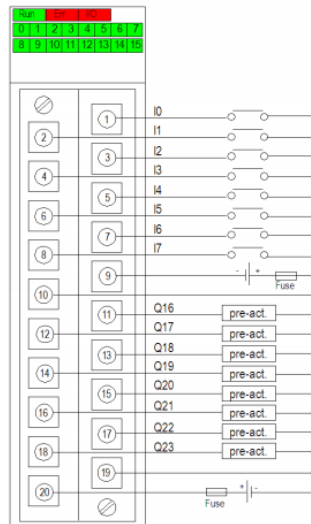


Figura 1.5-7: Conexión del módulo DDM 16022 a sensores y pre-actuadores

1.5.4. HMI STU 655 MAGELIS

La unidad HMI STU 655 MAGELIS es una interfaz hombre-máquina que tiene una tensión de funcionamiento de 24 VCC. Este dispositivo cuenta con una pantalla con tecnología TFT de 3,5 pulgadas y resolución de pantalla de 320x240 píxeles. Además cuenta con un puerto serie y un puerto Ethernet.

Para programar una aplicación en este dispositivo se usa el programa Vijeo Designer en una versión 5.1 o superior.

1.5.4.1. Manejo del panel LCD

A la hora de usar el panel LCD hay que tener en cuenta una serie de características que pueden aparecer durante el funcionamiento normal del dispositivo:

- Maneje el panel táctil HMI STU 655 sólo con un dedo para seleccionar un objeto, ya que si ejerce presión en dos o más puntos simultáneamente, es posible que se seleccione un objeto que no sea el que se desea.

- Los píxeles de la pantalla LCD pueden contener manchas blancas y negras y también es posible que parezca que el color ha cambiado.
- Puede parecer que las imágenes que se muestran en la pantalla LCD son diferentes si se mira desde fuera del ángulo de visión especificado. También pueden aparecer sombras o brillos irregulares en las imágenes.
- Si una misma imagen se muestra durante un largo periodo de tiempo en la pantalla, puede que al cambiar a otra imagen siga viéndose la imagen anterior. Cuando esto ocurra apague y encienda la unidad, pero es importante esperar al menos 10 segundos antes de reiniciarla ya que si se apaga y enciende el dispositivo rápidamente, este se puede dañar.

1.5.4.2. Identificación y funciones de las piezas

En este apartado veremos los diferentes elementos que componen el dispositivo, su ubicación y una breve descripción de ellos.

Parte Frontal del módulo de la Pantalla

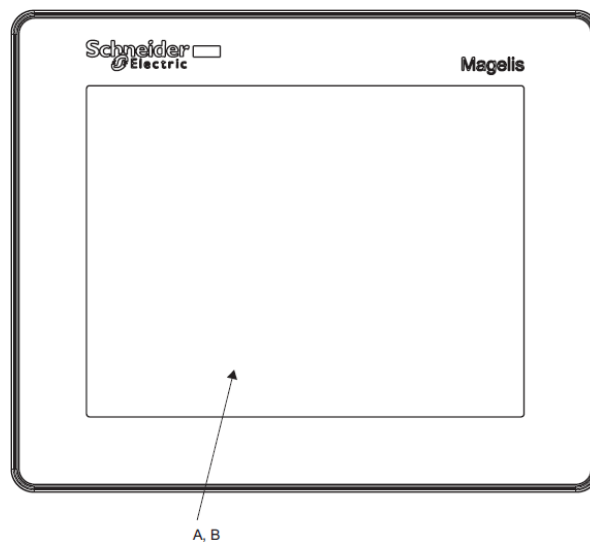


Figura 1.5-8: Frontal del módulo de la Pantalla

- **A** Pantalla: muestra las pantallas creadas por el usuario y las variables de equipo remoto.
- **B** Pantalla táctil: lleva a cabo las operaciones de cambio de pantalla y envía datos al host (PLC)

Conectores de la parte inferior

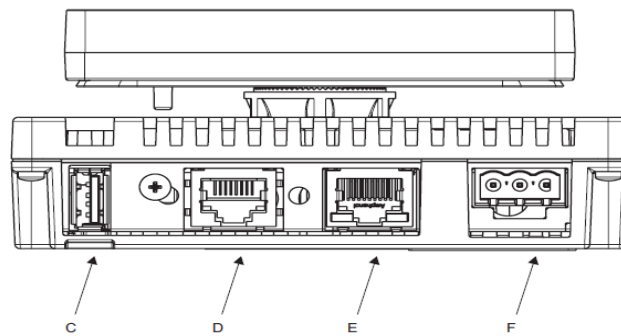


Figura 1.5-9: Conectores parte inferior del dispositivo HMI

- **C** Conector del puerto USB A estándar: conecta el cable de transferencia de datos o tarjeta de memoria a la unidad.
- **D** Interfaz serie: conecta un cable serie (RS-232C o RS485) desde el host/PLC a la unidad.
- **E** Interfaz Ethernet (LAN): conecta un cable Ethernet desde el host/PLC a la unidad.
- **F** Bloque de terminales de entrada de alimentación: conecta la entrada de alimentación y los conductores de tierra a la unidad

Conector lateral

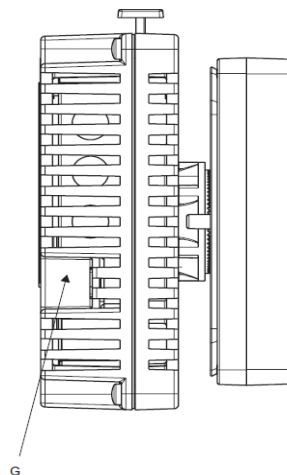


Figura 1.5-10: Conector lateral del dispositivo HMI

- **G** Conector de puerto USB mini-B: conecta el cable de PC de transferencia de datos a la unidad.

1.6. GRAFCET

1.6.1. Redes de Petri

Las Redes de Petri clásicas son un grafo dirigido que posee dos tipos de nodos principales: las etapas representadas por círculos y las transiciones representadas por líneas horizontales. Las transiciones con las etapas y viceversa se unen a través de arcos dirigidos ubicados entre ellos, pero no se pueden unir etapas con etapas ni transiciones con transiciones, sino que deben unirse de manera alterna. Cada arco dirigido tiene un número que indica su peso. El peso indica el número de marcas que se quitan de una etapa o se depositan en una etapa. Si los arcos no tienen ningún número se supone que tienen peso 1.

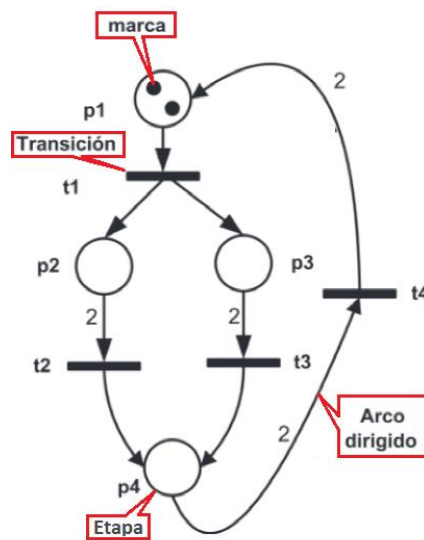


Figura 1.6-1: Red de Petri

Las marcas se representan como puntos negros dentro de cada etapa. El marcado inicial de una Red de Petri son las marcas que hay en cada etapa red al comienzo.

El número de marcas que hay en cada etapa de la red va cambiando a medida que se disparan las transiciones. Estos cambios siguen las reglas expuestas a continuación:

- Se dice que una transición está habilitada cuando la o las etapas que la preceden tienen la misma cantidad de marcas que el peso que tiene el arco que la une con la transición.
- Una transición habilitada puede o no ser franqueada.

- El franqueo de una transición habilitada quita un número de marcas igual al peso del arco que une la etapa o etapas de entrada con la transición y pone un número de marcas igual al peso del arco que une la etapa o etapas de salida con la transición.

Una Red de Petri es ordinaria cuando el peso de todos sus arcos es uno. Se dice que una Red de Petri es pura si no existen auto bucles, que están definidos como el conjunto de una etapa y una transición en el que la etapa es a la vez etapa de entrada y de salida de la transición. Existen dos tipos más de transiciones: la transición fuente que es la que no posee etapas de entrada y la transición sumidero que es la que no tiene etapa de salida.

Las Redes de Petri tienen una serie de propiedades que se pueden dividir en dos grandes grupos: las propiedades estáticas o estructurales, que dependen de la topología de la red, es decir, son independientes al marcado inicial; y las propiedades dinámicas o del comportamiento que son las que dependen de la ejecución y evolución de las marcas en la red, es decir, las que dependen del marcado inicial.

1.6.1.1. Propiedades estáticas o estructurales

Las propiedades estáticas que se exponen a continuación sólo son aplicables a Redes de Petri ordinarias y puras. Estas propiedades son:

- Vivacidad Estructural: una Red de Petri es estructuralmente viva si tiene alguna marca inicial.
- Controlabilidad: una Red de Petri es completamente controlable si cualquier marcado es alcanzable desde cualquier otro, es decir, la red puede evolucionar desde una etapa a otra etapa cualesquiera.
- Limitación o acotado estructural: una Red de Petri es limitada estructuralmente si es limitada para cualquier conjunto finito de marcas iniciales.
- Conservabilidad: una Red de Petri es conservativa si el número total de marcas en toda la red es constante a lo largo de su ejecución.
- Repetibilidad: una Red de Petri es repetible si existe una marca inicial y una secuencia de disparo desde dicha marca tal que las transiciones se disparan infinitamente en la secuencia de disparo.

- Consistencia: una Red de Petri es consistente si existe una marca inicial y una secuencia de disparos reversible desde la marca inicial hacia ella misma, de forma que cada transición haya sido disparada al menos una vez.

1.6.1.2. Propiedades dinámicas o del comportamiento

Las principales propiedades dinámicas son:

- Alcanzabilidad: es la principal propiedad dinámica. Consiste en la existencia de una secuencia de disparos, tal que, desde el marcado inicial se llegue a marcar cualquier estado.
- Limitable o acotada: una Red de Petri está limitada cuando el número de marcas máximo posible en cada etapa de la red alcanzable desde el marcado inicial es finito e igual para todas. Si ninguna etapa de la red puede tener más de una marca simultáneamente, entonces se dice que la red es segura.
- Vivacidad: una red es viva si desde cualquier etapa que haya sido alcanzada existe una secuencia de disparos, tal que, se alcance otra. Esta es una propiedad deseable en la ejecución de programas.
- Reversibilidad y estado inicial. Una Red de Petri es reversible si para cualquier marca distinta de la marca inicial, la marca inicial es alcanzable desde dicha etapa, es decir, una red es reversible siempre que se pueda alcanzar nuevamente el estado inicial del sistema.
- Persistencia: una Red de Petri es persistente si para más de una transición habilitada, el flaqueo de una no deshabilita a las otras.

1.6.1.3. Reglas de reducción

Las reglas de reducción permiten convertir sistemas complejos en sistemas más simples, con menos etapas y transiciones, pero manteniendo las propiedades originales. Las reglas de reducción son:

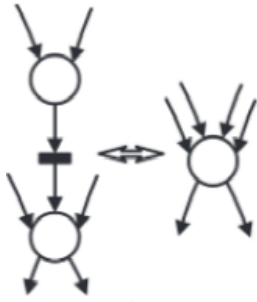


Figura 1.6-2: Fusión de etapas en serie

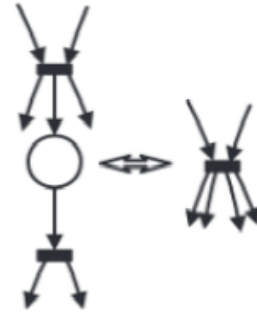


Figura 1.6-3: Fusión de transiciones en serie

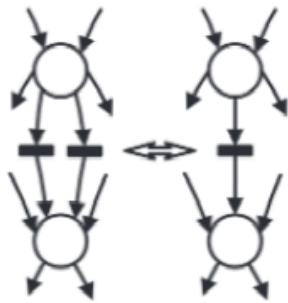


Figura 1.6-4: Fusión de transiciones paralelas

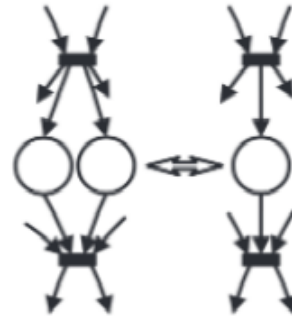


Figura 1.6-5: Fusión de etapas paralelas



Figura 1.6-6: Eliminación de auto bucles en etapas

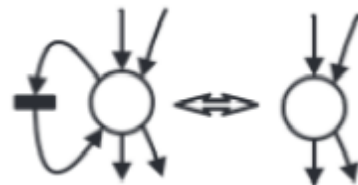


Figura 1.6-7: Eliminación de auto bucles en transición

1.6.2. Introducción histórica

El GRAFCET (GRAPhe Fonctionnel de Commande Etapes-Transitions), como resultado de los trabajos de la AFCET (Association Française pour la Cybernétique Economique et Technique) en 1977.

El principal objetivo fue el desarrollo de un método de descripción de procesos, independiente de la tecnología empleada, mediante un gráfico funcional que pudiera ser interpretado fácilmente por personas que no estuvieran especializados en automatización.

En 1982 un grupo de trabajo de AENOR dio lugar a la publicación de la Norma NF C03-1904. Posteriormente, en 1988, dicha norma sería reconocida a nivel internacional, cuando IEC desarrollo la norma IEC-848 que se titula “Établissement des diagrammes fonctionnels pour systèmes de commande”.

Actualmente, muchos fabricantes de autómatas programables incorporan instrucciones de programación que permiten introducir un grafo GRAFCET directamente. Además, existen softwares que son capaces de compilar un grafo GRAFCET a lenguaje del autómatas, pudiéndose usar este método para prácticamente cualquier autómatas, aunque el autómatas no tuviera esta opción. Aportando una gran ventaja en las fases de diseño, verificación, explotación e incluso en una posible modificación del automatismo. Sin embargo, no hay que confundir al GRAFCET con un lenguaje de programación.

1.6.3. Principios del GRAFCET

El GRAFCET es un método gráfico que permite representar sistemas secuenciales, y que está basado en las Redes de Petri.

Por tanto, como vimos en el apartado 1.6.1., es una sucesión de etapas y transiciones unidas mediante arcos. Con la diferencia de que en este caso las etapas se representan mediante rectángulos, en lugar de círculos.

Cada etapa tiene una serie de acciones asociadas, y cuando la etapa está activa dichas acciones son realizadas. Las etapas iniciales, que se activen con la puesta en marcha, se representan con un rectángulo doble.

Entre dos etapas hay una transición, y a cada una le corresponde una receptividad. La receptividad de la transición es la condición que se debe cumplir para poder pasarla. Una transición es válida cuando la etapa anterior a ella está activa. Si la transición es válida y su receptividad se verifica, entonces se dice que la transición es franqueable. Al franquear una transición se desactivan sus etapas anteriores y se activan las etapas que hay justo después de la transición.

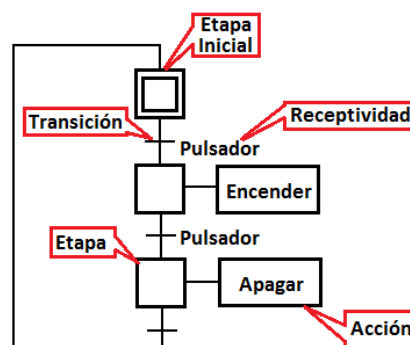


Figura 1.6-8: Elementos del GRAFCET

1.6.4. Niveles del GRAFCET

El GRAFCET se puede usar para describir los tres niveles de especificaciones de un automatismo. Estos tres niveles de descripción son: funcional, tecnológica y operativa. A continuación, se desarrollará cada uno de estos niveles y se incluirá un ejemplo de ellos.

1.6.4.1. Nivel 1: Descripción funcional

El primer nivel de descripción se corresponde con una descripción global sin muchos detalles sobre el automatismo, pero permite comprender rápidamente su funcionamiento general. Es el tipo de descripción que haríamos para poder explicarle a la persona que va a diseñar la máquina lo que queremos que esta haga, o para justificar la viabilidad de esa máquina a una persona con poder de dirección o para buscar financiación para un proyecto. Se emplea un lenguaje natural para describir las acciones y las transiciones y no se hace referencia a la tecnología que se va a emplear. Para este nivel se suele emplear la norma IEC 60848.

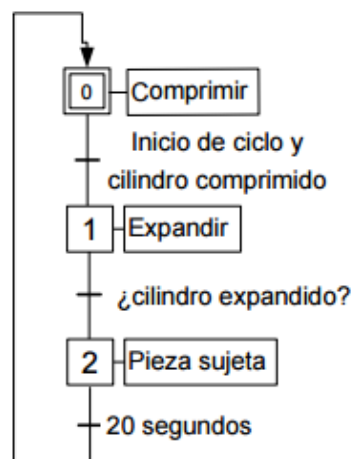


Figura 1.6-9: GRAFCET de nivel 1

1.6.4.2. Nivel 2: Descripción tecnológica

En este nivel se debe hacer una descripción del automatismo de forma que quede perfectamente definida la tecnología que se va a usar (relé, neumática, pulsadores, contactores...). Al igual que en el nivel anterior, en este también se usa la norma IEC 60848.

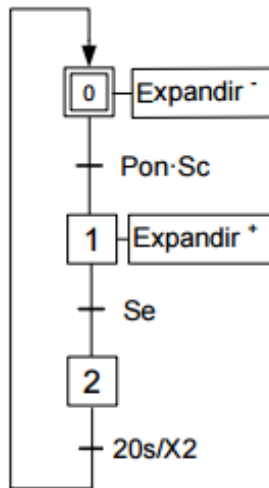


Figura 1.6-10: GRAFCET de nivel 2

1.6.4.3. Nivel 3: Descripción operativa

En el último nivel se realiza la implementación del automatismo. El GRAFCET definirá la secuencia de actuaciones que realizará el automatismo, y en caso de ser un autómatas programable, también definirá la activación de las salidas en función de la evolución de las entradas. Para este nivel se emplea la norma IEC 61131-3, que especifica el diagrama SFC, un lenguaje de programación gráfico completo.

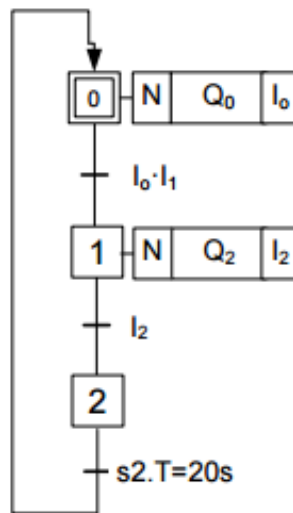


Figura 1.6-11: GRAFCET de nivel 3

1.6.5. Estructuras lógicas fundamentales

En lógica combinacional se demuestra que cualquier función lógica puede expresarse mediante combinación de los operadores “Y”, “O” y “NO”. Por tanto, en los sistemas combinacionales siempre se pueden expresar en GRAFCET gráficos que sólo incluyan tres estructuras básicas, por muy complejo que este sea. Estas estructuras son:

- Secuencia lineal
- Bifurcación en O
- Bifurcación en Y

1.6.5.1. Secuencia lineal

La secuencia lineal es la estructura más simple posible. Consiste en una sucesión alterna de etapas y transiciones en la que las acciones se van produciendo una tras otra, es decir, cuando una acción finaliza, se desactiva la etapa a la que está asociada, y comienza la siguiente etapa a ejecutar su acción. Una secuencia está activa cuando alguna de sus etapas lo está, y está desactiva cuando no hay ninguna etapa de la secuencia activa. En una secuencia sólo debería estar activa una etapa cada vez, ya que, aunque no lo impiden las reglas del GRAFCET, podría crear incoherencias que haga que el sistema no funcione correctamente en algunos momentos, por ejemplo, desactivándose y activándose la misma etapa simultáneamente o dando dos órdenes que sean físicamente opuesta, como subir y bajar. Es común ver este tipo de estructuras en el nivel descriptivo junto con las macroetapas, que se verán posteriormente, para describir la instalación.

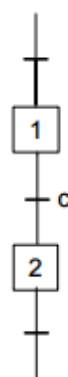


Figura 1.6-12: Secuencia lineal

1.6.5.2. Bifurcación en O

Las estructuras O expresan secuencias alternativas de control en función de variables. Una bifurcación en O está compuesta por dos elementos, una divergencia en O y una convergencia en O.

Divergencia en O

La divergencia en O es el comienzo de varios caminos o procesos alternativos posibles. No es necesario que todos los subprocesos que pueden salir de la divergencia tengan el mismo número de elementos. Informáticamente sería equivalente a la instrucción "IF....THEN....ELSE".

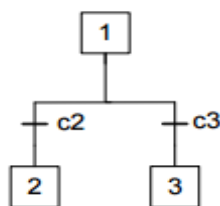


Figura 1.6-13: Divergencia en O

La divergencia en O presenta un problema de indeterminación cuando se verifican varias receptividades simultáneamente. Por ejemplo, en la figura 1.6-13, si se verifican a la vez las receptividades c_2 y c_3 , en este caso no se sabe a priori qué transición se va a franquear y cual no. Esto es algo que hay que solucionar, bien sea por que por las condiciones de la instalación sea físicamente imposible que ambas receptividades se verifiquen simultáneamente o bien porque se use algún tipo de lógica que resuelva el problema.

Algunos ejemplos de modelado de prioridad tradicional por exclusión lógica (norma IEC 60848) se pueden ver en la figura 1.6-14. En el caso A, si se verifican c_2 y c_3 , se franqueará la transición asociada a c_3 , análogamente se puede hacer para franquear la transición asociada a c_2 . Mientras que en el caso B no se franqueará ninguna transición si ambas están verificadas.

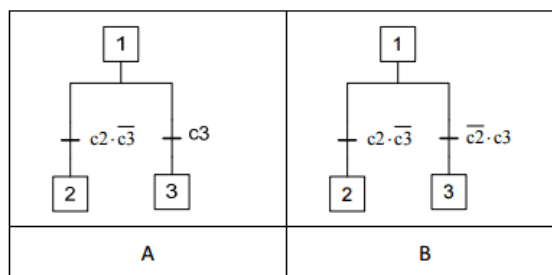


Figura 1.6-14: Modelado tradicional de exclusión lógica

Convergencia en O

La convergencia en O finaliza la secuencia de control alternativo. En este caso no da lugar a una situación de incertidumbre como el caso anterior, ya que sólo una etapa de puede estar activa al provenir de la divergencia en O.

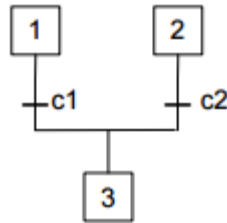


Figura 1.6-15: Convergencia en O

Saltos condicionales

El salto condicional es un caso particular de la divergencia en O. Consiste en saltarse una serie de instrucciones si se verifica una condición.

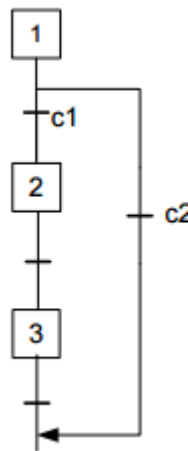


Figura 1.6-16: Salto condicional

Repetición de secuencia

La repetición de secuencia es otro caso particular de la divergencia en O. Consiste en repetir una secuencia de instrucciones que ya han sido ejecutadas si se verifica una condición. Para esto hay que unir la etapa en la que se produce la bifurcación con la etapa anterior donde se quiere empezar a repetir la secuencia, al ser el camino ascendente es necesario indicarlo explícitamente con una flecha hacia arriba. Informáticamente se puede asociar a la instrucción "DO...WHILE".

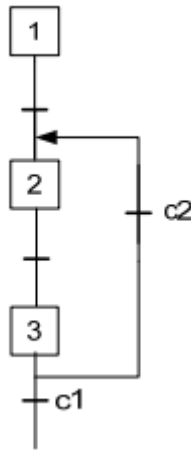


Figura 1.6-17: Repetición de secuencias

1.6.5.3. Bifurcación en Y

Las estructuras Y expresan concurrencia y se representan con dos barras horizontales. Una bifurcación en Y está compuesta por dos elementos, una divergencia en Y y una convergencia en Y.

Divergencia en Y

La divergencia en Y comienza un punto de sincronismo y el inicio de varias estructuras series que serán ejecutadas en paralelo, es decir, cuando se cumpla la condición que da comienzo a la divergencia en Y, se activarán todas las etapas que estén conectadas a la doble barra horizontal. No es necesario que todos los caminos tengan el mismo número de elementos. En el ejemplo de la figura 1.6-18, cuando se valide la especificación c, se activarán las etapas 2 y 3.

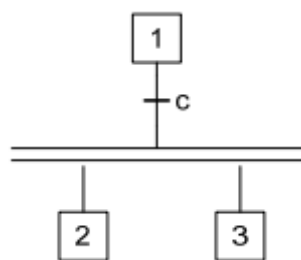


Figura 1.6-18: Divergencia en Y

Convergencia en Y

La convergencia en Y indica un punto de sincronismo y finaliza la ejecución de las estructuras que se estaban ejecutando paralelamente. Para que esto ocurra es necesario que estén activas todas las etapas que llegan hasta la doble barra horizontal, para que se habilite la transición, y así cuando se cumpla la condición asociada a la

transición, esta podrá ser flanqueada. En la figura 1.6-19, hasta que no estén activos las etapas 2 y 3, aunque se verifique la receptividad c, la transición no será franqueada.

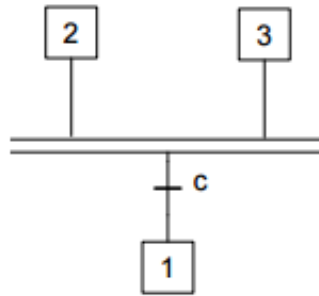


Figura 1.6-19: Convergencia en Y

Una vez analizadas las bifurcaciones en O y en Y, podemos ver que cuando se comienza con una divergencia en O no se puede finalizar con una divergencia en Y, y viceversa. Esto es debido a que con la divergencia en O sólo se activa uno de los posibles caminos, por tanto, sólo habrá una etapa activa antes de la convergencia en Y, y el programa quedará bloqueado. Mientras que en el caso contrario, cuando se comienza por una divergencia en Y se activan todas las etapas que estén después de esta, por lo tanto, cuando finalicemos con una convergencia en O estarán activas varias etapas inmediatamente anteriores, esto provocará que aparezca el problema de incertidumbre en la transición que no teníamos, además, lo que puede ser peor, que como sólo es necesario una transición activa para franquear, las marcas se irán multiplicando, pudiéndose llegar a saturar el programa si se está ejecutando durante mucho tiempo.

1.6.6. Reglas de evolución

Cuando se describe un automatismo o cualquier conjunto de sucesos condicionales y secuenciales. Al diseñar el GRAFCET se deben representar las cinco reglas de evolución del GRAFCET, ya que si no, el funcionamiento no sería el esperado del GRAFCET implementado.

A continuación se expondrán estas cinco reglas, junto con algún ejemplo si se considera necesario.

1.6.6.1. Inicialización

En la inicialización del sistema se deben activar todas las etapas iniciales. Normalmente la situación inicial del GRAFCET corresponde a una situación de reposo o parada segura, el estado en el que se pretende que esté la máquina al comenzar a funcionar.

1.6.6.2. Evolución de la transición

Una transición puede estar en cuatro situaciones posibles durante la ejecución de un programa. Estas situaciones son:

- No validada: La etapa o etapas inmediatamente anteriores no están activas.
- Validada: La etapa o etapas inmediatamente anteriores si están activas, pero no se cumple la condición lógica de transición.
- Franqueable: La etapa o etapas inmediatamente anteriores están activas y además se cumple la condición lógica de transición. Esta situación es transitoria, ya que, como se ve más adelante, dicha transición será automáticamente franqueada.
- Franqueada: Se ha activado la etapa o etapas inmediatamente siguientes y se ha desactivado la etapa o etapas inmediatamente anteriores a la transición.

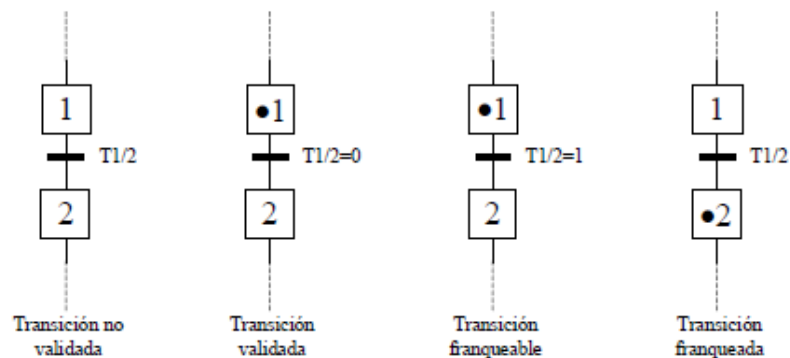


Figura 1.6-20: Posibles situaciones de una transición

Ahora que hemos visto las posibles situaciones en la que se puede encontrar una transición, veremos cómo actúa en esos casos.

- Sólo es posible franquear una transición si ésta está previamente validada.
- Cualquier transición franqueable será inmediatamente franqueada.
- Cuando existan varias transiciones franqueables simultáneamente, todas ellas serán franqueadas simultáneamente.

1.6.6.3. Evolución de las etapas activas

El franqueo de una transición implica la activación simultánea de todas las etapas inmediatamente posteriores y desactivación simultánea de todas las etapas inmediatamente anteriores. La evolución de las etapas activas sigue el camino de los arcos orientados cuando se franquea una transición.

1.6.6.4. Simultaneidad en el franqueamiento de las transiciones

Las transiciones simultáneas franqueables han de ser simultáneamente franqueadas.

La existencia de esta cuarta regla nos permite la descomposición de un GRAFCET complejo en dos más sencillos. En la figura 1.6-21, tenemos un GRAFCET con paralelismo estructural (izquierda) y lo descomponemos en dos GRAFCETs independientes (derecha), teniendo en cuenta que la receptividad de cada una de las dos transiciones obtenidas ha de considerar la activación de la etapa correspondiente del otro GRAFCET, ya que, en caso contrario, el funcionamiento de las dos estructuras no sería el mismo.

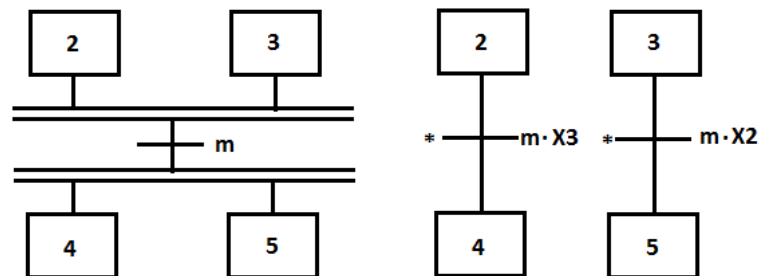


Figura 1.6-21: Descomposición de GRAFCET en dos GRAFCET independientes

A menudo es conveniente señalar con un (*) aquellas transiciones en las que el cumplimiento de la cuarta regla es imprescindible para el correcto funcionamiento, tal como se ve en los GRAFCETs de la derecha de la figura 1.6-21.

1.6.6.5. Prioridad de la activación

Si durante el funcionamiento de un automatismo, una etapa debe ser simultáneamente activada y desactivada, dicha etapa permanecerá activada. Esta regla es un convencionalismo para resolver casos de indeterminación.

1.6.7. Macroetapas

Una macroetapa es una representación simbólica de una parte de un GRAFCET, esa parte se conoce como expansión de la macroetapa. Se usa para representar procesos complejos de forma más simple, para que resulte sencilla su interpretación y distribución.

Se representa como una etapa con dos líneas horizontales. En su interior se escribe un identificador que empieza por la letra “M” seguida de un número identificativo. El GRAFCET parcial expandido comienza con una etapa de entrada, con el nombre “E” seguido por el número identificativo de la macroetapa, y finaliza con otra etapa cuyo nombre es la letra “S” seguida del número identificativo de la macroetapa, como puede verse en la figura 1.6-20.

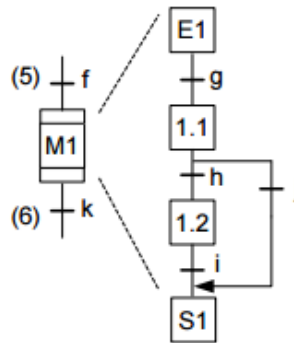


Figura 1.6-22: Ilustración de una macroetapa y su GRAFCET parcial expandido

La evolución de una macroetapa sigue las mismas reglas que hemos visto en los apartados anteriores, es decir, para franquear una transición del GRAFCET parcial expandido, es necesario que esté activa la etapa anterior y que se verifique la receptividad.

Hay que tener en cuenta las consideraciones recogidas en la norma IEC 60848 sobre las macroetapas:

- Se admite que el GRAFCET parcial expandido de una macroetapa tenga más de una etapa fuente inicial.
- Se admiten anidamiento de macroetapas sin ninguna restricción en cuanto a número.

1.6.8. GRAFCET jerarquizado

Cuando un sistema está compuesto por varios GRAFCETs parciales, es posible que un GRAFCET fuerce el estado de otro. El forzado de un GRAFCET parcial sobre otro establece una relación jerárquica de mando del primero sobre el segundo. El forzado de un estado de un GRAFCET se realiza a través de una acción interna. Su ejecución tiene prioridad sobre las reglas de evolución ordinarias que se vieron en apartados previos. Al GRAFCET que fuerza se le llama GRAFCET maestro y el que es forzado recibe el nombre de GRAFCET esclavo.

En el GRAFCET, el forzado se representa como una acción con doble recuadro donde la descripción de la acción emplea la siguiente sintaxis:

F/(nombre del GRAFCET esclavo):{etapa o etapas que se desea forzar}



Figura 1.6-23: Representación de acción de forzado

La acción de forzado es una acción continua convencional, por tanto, se mantiene un estado forzado mientras que la etapa a la que está asociado está activa, y deja de ejecutarse al desactivarse la etapa.

Por tanto, mientras está activa la etapa forzante, el GRAFCET esclavo no puede evolucionar, independientemente que cualquiera de sus transiciones sean franqueables.

Como resultado de la orden, el diagrama esclavo se sitúa en el estado de actividad de etapas descrito en la orden recibida.

Para evitar problemas con el funcionamiento del programa es conveniente cumplir una serie de reglas:

- Si un GRAFCET tiene posibilidad de forzar a otro, el GRAFCET que es forzado no podrá forzar al GRAFCET maestro .
- En cualquier momento, un GRAFCET sólo podrá ser forzado por un único GRAFCET. Se sigue una jerarquía en la que cada miembro sólo puede ser forzado por su superior inmediato.

Ejemplos de forzados:

F/G1:{ } -> desactivará todas las etapas del GRAFCET esclavo (G1)

F/G2:{1} -> desactivará todas las etapas del GRAFCET esclavo (G2) y activará la etapa inicial

F/G3:{3, 5, 9} -> desactivará todas las etapas del GRAFCET esclavo (G3) y activará las etapas 3, 5 y 9

F/G4:{*} -> congelará todas las etapas del GRAFCET esclavo (G4) en el estado en el que se encuentran

1.7. Guía GREMMA

1.7.1. Presentación de la Guía GEMMA

En un proceso productivo automatizado la máquina no está siempre funcionando en modo automático, como sería deseable, sino que con frecuencia aparecen contingencias que hace que el proceso tenga que detenerse, por ejemplo, averías, mantenimiento, falta de piezas, cambio del producto a realizar, etc., o simplemente que la empresa no esté trabajando 24 horas al día durante todos los días del año, y por tanto la máquina también debe dejar de producir.

En los automatismos modernos, estas contingencias son previsibles y el propio automatismo está preparado para prever todos los estados posibles: funcionamiento manual o semiautomático, paradas de emergencia, funcionamiento automático, puesta en marcha...; así como para detectar defectos y colaborar con el personal encargado de la supervisión o mantenimiento de la instalación en la puesta a punto, la representación y otras tareas no relacionadas con el proceso productivo habitual.

Los trabajos realizados por la ADEPA (Agence nationale pour le DEveloppement de la Productique Appliquée à l'industrie) para fijar una forma universal de denominar y definir los diferentes estados que puede tener un sistema, culminaron en la creación de la Guía GEMMA (Guide d'Etude des Modes de Marches et d'Arrets) en 1993.

La Guía GEMMA es una guía gráfica que permite presentar, de una forma sencilla y comprensible, los diferentes modos o estados de marcha y parada en los que puede estar un proceso productivo automatizado, así como las formas y condiciones que se deben cumplir para pasar de un estado a otro.

Bajo la norma nacional francesa UTE C 03-191, la Guía GEMMA se complementa con el GRAFCET, dando lugar a una metodología que incluye los modos de marcha y paro del control secuencial, el funcionamiento correcto del proceso controlado, junto con el funcionamiento deteriorado ante anomalías e incluso el tratamiento de situaciones de emergencia.

1.7.2. Descripción de la Guía GEMMA

Un automatismo consta de dos partes fundamentales: el sistema de producción y el control del mismo (ordenador, autómatas programables...). El control puede estar alimentado o sin alimentar.

Cuando el control está alimentado, el sistema puede encontrarse en tres situaciones diferentes: en funcionamiento, por lo tanto está produciendo; parado o en proceso de parada; y en defecto. Puede haber producción en cada uno de estas

situaciones. En funcionamiento, obviamente, pero también puede producir durante el proceso de parada, e incluso en situación de defecto, aunque en estas circunstancias puede darse el caso en el que, o bien el producto derivado no es aprovechable o es necesario una manipulación posterior de este para que los sea.

La Guía GEMMA representa estas cuatro situaciones (control sin alimentación, funcionamiento, parada y defecto) mediante rectángulos grises, además a través de un quinto rectángulo, marcado con líneas discontinuas que se intersecta con los tres rectángulos que representan las situaciones en las que puede estar el control alimentado, indica que el sistema está produciendo, tal como muestra la figura 1.7-1.

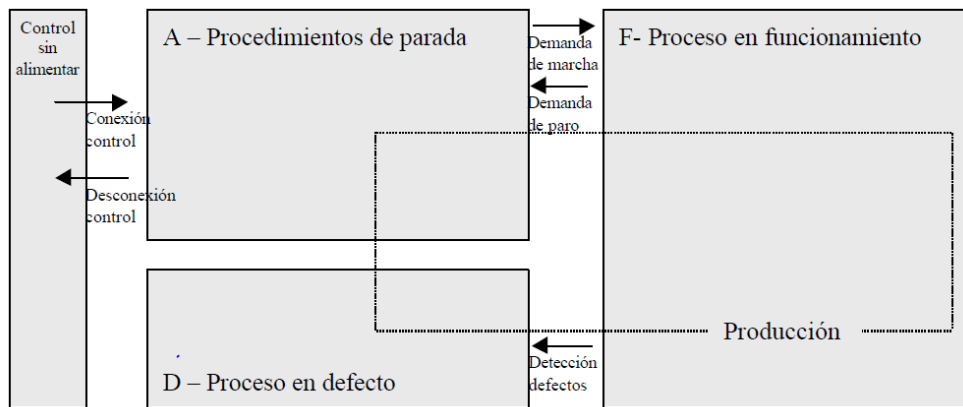


Figura 1.7-1: Modos de funcionamiento de la Guía GEMMA

Cada una de los modos de funcionamiento se puede subdividir en varios estados, de forma que al final hay 17 estados de funcionamiento posibles, que estudiaremos a continuación, como muestra la figura 1.7-2. No todos los procesos necesitarán usar todos los estados, pero sí podremos relacionar los estados necesarios en cada proceso con parte de los que propone la Guía GEMMA.

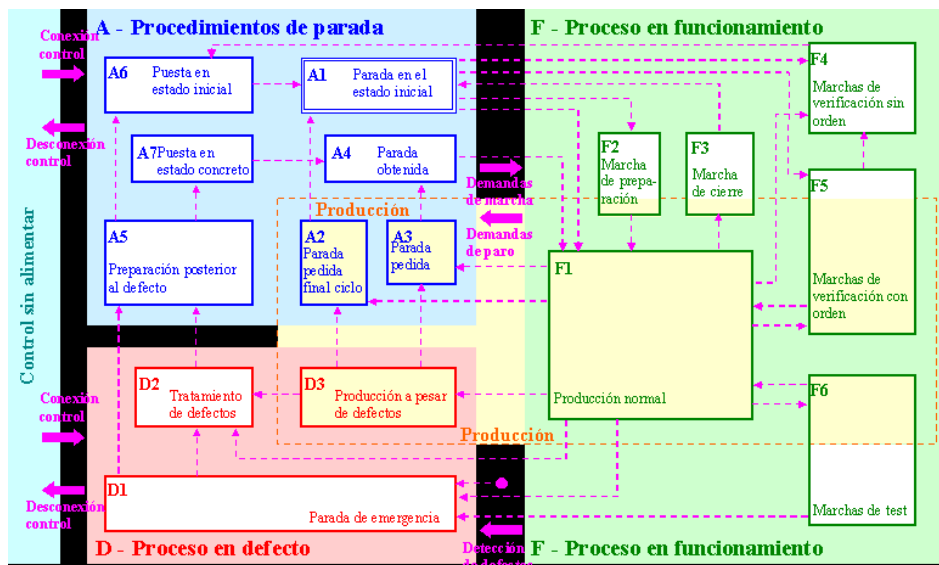


Figura 1.7-2: Estados de la Guía GEMMA

1.7.2.1. Grupo F: Procedimientos de funcionamiento

Este grupo contiene todos los modos de funcionamiento necesarios para la obtención de la producción, es decir, los de funcionamiento normal (F1 a F3) y los de prueba y verificación (F4 a F6).

F1: Producción normal

Es el estado más importante, ya que es en el que la máquina produce normalmente, es decir, el estado en el que se realizan las tareas para la que ha sido diseñada la máquina. Al funcionamiento dentro de este estado se le puede asociar un GRAFCET que llamaremos GRAFCET de base.

F2: Marcha de preparación

Corresponde a las acciones necesarias que se han de realizar antes de que la máquina comience a producir (precalentamiento, preparación de componentes, calibración, etc.)

F3: Marcha de cierre

Son las acciones que se deben realizar antes de parar la máquina o del cambio de algunas de las características del producto, corresponde a la fase de vaciado y/o limpieza.

F4: Marchas de verificación sin orden

En este caso la máquina, normalmente por orden del operario, puede realizar cualquier movimiento o unos determinados movimientos preestablecidos. Es el denominado control manual y se utiliza para funciones de mantenimiento y verificación.

F5: Marchas de verificación con orden

En este caso la máquina realiza el ciclo completo de funcionamiento en orden, pero al ritmo fijado por el operador. Al igual que el estado F4, se usa para tareas de mantenimiento y verificación. En este estado existe la posibilidad de que la máquina produzca. En general se asocia al control automático.

F6: Marchas de test

Se usa para realizar operaciones de ajuste y mantenimiento preventivo, por ejemplo, comprobar si la activación de los sensores se realiza en un tiempo máximo, curvas de comportamiento de algunos actuadores, etc.

1.7.2.2. Grupo A: Procedimientos de parada y puesta en marcha

Este grupo contiene todos los modos en los que el sistema está parado (A1 y A4), los que llevan a la parada del sistema (A2 y A3) y los que permiten pasar el sistema de un estado de defecto a un estado de parada (A5 a A7). Corresponde a todas las paradas por causas externas al proceso.

A1: Parada en el estado inicial

Este estado es el estado normal de reposo de la máquina y, como puede verse en la figura 1.8-2, se representa con un rectángulo doble. La máquina normalmente se representa en este estado en los planos, esquemas eléctricos, esquemas neumáticos, etc.

A2: Parada solicitada al final del ciclo

Es un estado transitorio en el que la máquina, que hasta aquel momento estaba produciendo normalmente, debe producir sólo hasta acabar el ciclo actual y pasar a estar parada en el estado inicial.

A3: Parada solicitada en un estado determinado

Es un estado en que la máquina, que hasta aquel momento estaba produciendo normalmente, se detiene en un estado determinado que no coincide con el final de ciclo. Es un estado transitorio de evolución hacia A4.

A4: Parada obtenida

Es un estado de reposo de la máquina que no coincide con el estado inicial.

A5: Preparación para la puesta en marcha después de un defecto

Se realizan todas las acciones necesarias (vaciado, limpieza, reposición de un determinado producto, etc.) que son necesarias para poner la máquina nuevamente en funcionamiento después de que se haya producido un defecto.

A6: Puesta del sistema en el estado inicial

El sistema es llevado hasta la situación inicial (normalmente situación de reposo), una vez realizado, la máquina pasa a estar parada en el estado inicial.

A7: Puesta del sistema en un estado determinado

El sistema es llevado hasta una situación concreta diferente de la inicial, una vez terminado, la máquina pasa a estar en reposo.

1.7.2.3. Grupo D: Procedimientos de defecto

Este grupo contiene todos los modos en los que el sistema está en defecto, tanto si está produciendo (D3), está parado (D1) o está en fase de diagnóstico o tratamiento del defecto (D2). Corresponde a todas las paradas por causas internas al proceso.

D1: Parada de emergencia

No es simplemente la parada de emergencia, sino que en este estado también se realizan todas las acciones necesarias para llevar el sistema a una situación de parada segura.

En la figura 1.8-2 podemos observar como hay un camino de entrada a D1 que no viene desde ningún sitio. Este hace referencia a que se puede llegar a este estado desde cualquier otro, y como en la mayoría de los casos suele haber las mismas condiciones, se representa de esta forma para no complicar el diagrama.

D2: Diagnóstico y/o tratamiento de fallos

En este estado la máquina es examinada después de un defecto, y con ayuda o sin del operador, indica los motivos de fallo para su rearme.

D3: Producción a pesar de los defectos

Corresponde a aquellos casos en los que se debe continuar produciendo a pesar de que el sistema no trabaje en condiciones óptimas. Incluye los casos en los que, por ejemplo, se produce para agotar un reactivo no almacenable o aquellos otros en los que no se sigue el ciclo normal porque un operador sustituye a la máquina en una determinada tarea a causa de la avería, etc.

1.7.3. Método de Aplicación

La Guía GEMMA contiene todos los estados posibles en la mayoría de instalaciones automatizadas. El diseñador del automatismo debe estudiar cada uno de estos estados uno por uno y determinará cuáles son necesarios y cuales no para la instalación.

En caso de que un estado no sea posible o sea innecesario, hará una cruz indicando que ese estado ha sido descartado.

Una vez determinados los estados que se tendrán en cuenta, es el momento de analizar entre que estados será posible la evolución. La Guía GEMMA indica los caminos de evolución más comunes entre los estados. Para indicar los caminos que el diseñador ha considerado necesarios, se traza una línea continua que una los diferentes estados; repasando las líneas discontinuas que aparecen en el diagrama o

en el caso que el camino deseado no esté propuesto, trazando un nuevo camino de evolución.

Por último, se dibujan las transiciones igual que se hace en GRAFCET, y se indican las condiciones que se deben cumplir para franquear las transiciones. En algunos casos es posible que un camino no tenga una condición determinada, en esos casos se puede no indicar nada o poner como condición que finalice el estado anterior.

1.8. Unity Pro

En este capítulo se hará una breve descripción con los conceptos más importante que se deben conocer sobre el programa Unity Pro, centrándonos en la programación de un PLC Modicon M340.

1.8.1. Crear un nuevo proyecto

Una vez que se ha abierto el programa Unity Pro, hay dos formas de crear un nuevo proyecto: pulsando el icono que aparece en la esquina superior izquierda, o abriendo el menú **Fichero** y seleccionando la opción **Nuevo**, como se ve en la figura 1.8-1.

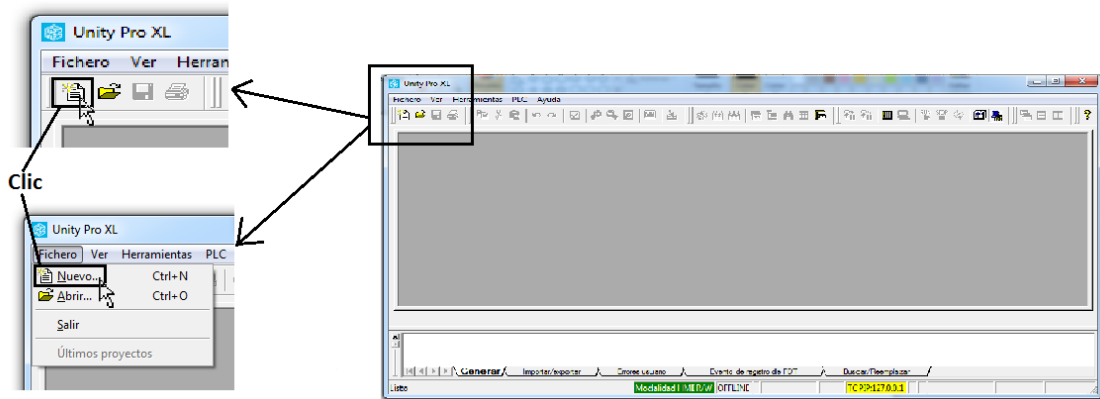


Figura 1.8-1: Crear nuevo proyecto de Unity Pro

Cuando se ha creado un nuevo proyecto, se abre una ventana emergente en la que se debe seleccionar el procesador que se va a usar.

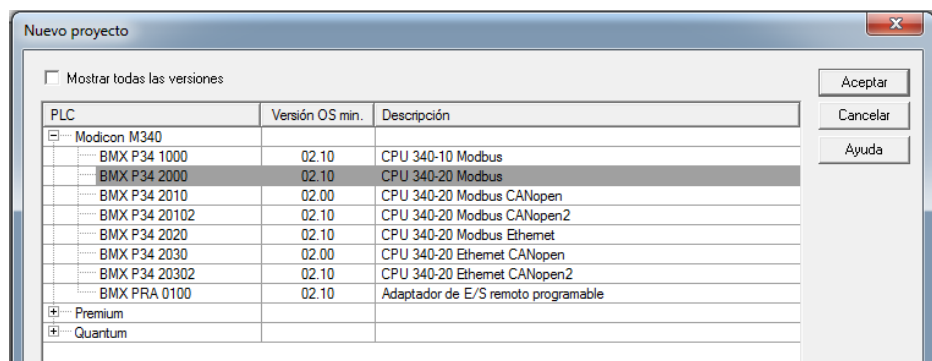


Figura 1.8-2: Seleccionar CPU

Luego se selecciona el bastidor dentro del Explorador de Proyectos, se abrirá la ventana de editor del Bus PLC donde se selecciona los módulos que componen el autómatas haciendo doble clic en la posición correspondiente.

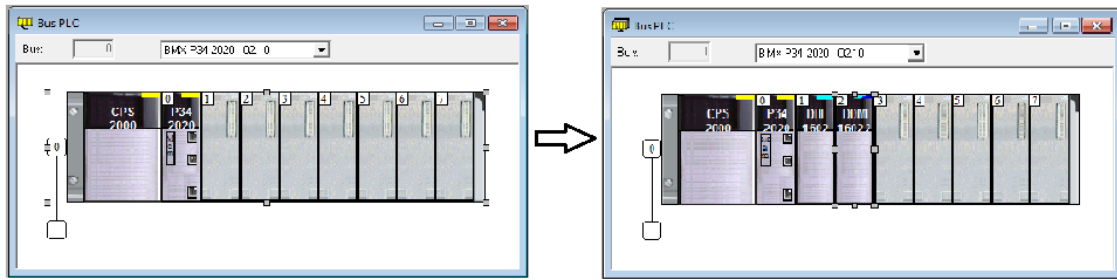


Figura 1.8-3: Configuración del PLC

1.9.2. Interfaz de usuario

La interfaz de usuario se compone de varias ventanas configurables y de barras de herramientas que se muestran en la figura 1.8-4

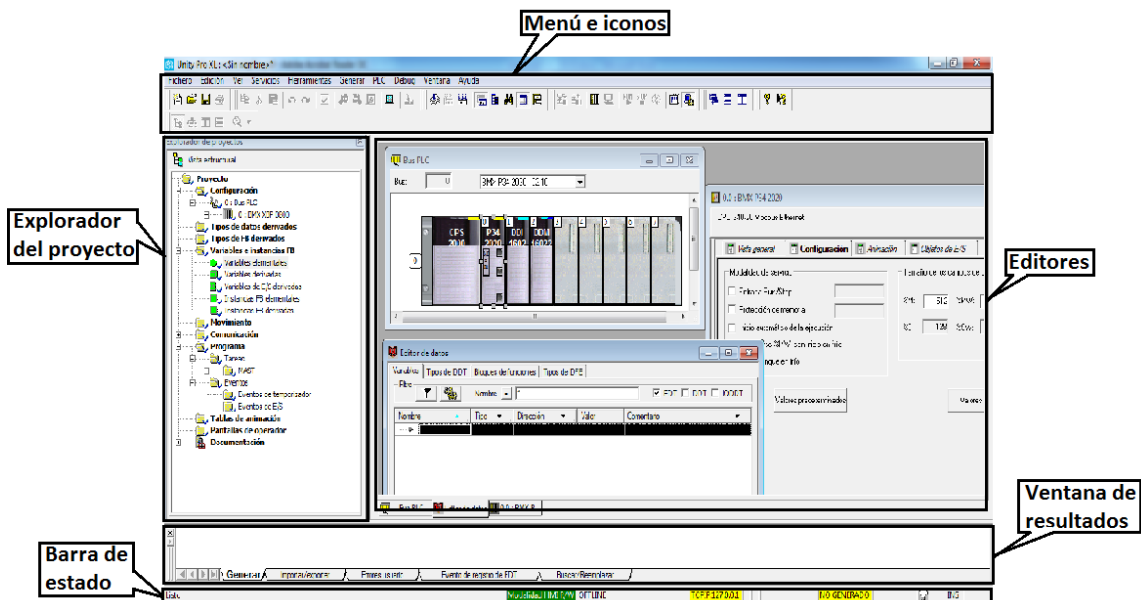


Figura 1.8-4: Interfaz de Usuario de Unity Pro

Menú e iconos

Los menús se usan para acceder a todas las funciones de la herramienta Unity Pro, y los iconos para acceder a las funciones más utilizadas. Se puede personalizar haciendo clic derecho sobre la zona de iconos y seleccionar los que nos interese visualizar.

Explorador de proyecto

Representa el árbol del proyecto y permite acceder a las diferentes partes de un proyecto. Hay dos vistas posibles: estructural o funcional.

Editores

Cuando se hace un doble clic en una parte del explorador de proyectos, se abre la ventana de configuración, edición o programación correspondiente. Por ejemplo, editores de datos, Ethernet, Bus PLC, etc.

Ventana de resultados

Aparece la información relacionada con las operaciones de: importar/exportar, fue exitosa o fallida; analizar/generar un proyecto, aparecen los fallos debidos a errores de programación, tipos de datos incompatible o de comunicación de red; buscar o reemplazar variables, fue exitosa o fallida.

Barra de estado

Aparece la información relacionada con el estado del proyecto en el PC y PLC, y la configuración de la conexión con el PLC.

1.9.3. Variables

Una variable es una entidad de memoria. Cada variable contiene, como mínimo, un nombre y un tipo de datos. Además es posible añadirle una dirección, un valor por defecto y un comentario.

Una variable alocatada es una variable asociada a un módulo de entradas/salidas o a una referencia de memoria. Los tipos de señales principales con los que se puede trabajar a nivel de E/S son: %I (Entrada digital); %Q (Salida digital); %IW (Entrada analógica); %QW (Salida analógica).

Una variable no alocatada no está asociada a ninguna E/S ni a ninguna referencia de memoria.

Las constantes no pueden ser modificadas por el programa mientras se está ejecutando.

En el explorador de proyectos podemos encontrar los diferentes tipos de datos existentes en Unity Pro.

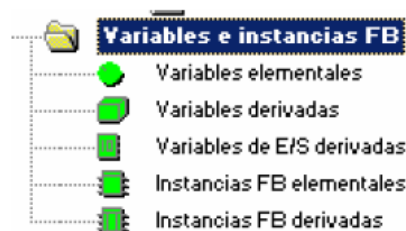


Figura 1.8-5: Familias de tipos de datos de Unity Pro

1.8.3.1. Variables elementales

Las variables elementales nos permiten almacenar un dato o valor cambiante en el programa. Para declarar una variable elemental basta con darle un nombre y seleccionar el tipo de variable en el editor que se abrirá al seleccionar Variables elementales en el Explorador de proyectos. Los tipos más comunes de variables elementales son las de tipo binario (booleano, entero, time).

Tipos Booleanos

- **BOOL** -> almacena un valor lógico, FALSE (=0, valor predeterminado) o TRUE (=1).
- **EBOOL** -> almacena un valor lógico, FALSE (=0, valor predeterminado) o TRUE (=1), pero también incluye información relativa a la gestión de los flancos y el forzado.

Tipos Enteros

Los tipos enteros permiten representar un valor en diferentes bases: base 10 (decimal), forma predeterminada, ejemplo, 154; base 2 (binaria), el valor se escribe con el prefijo 2#, ejemplo, 2#1011; base 8 (octal), el valor se escribe con el prefijo 8#, ejemplo, 8#077; y base 16 (hexadecimal), el valor se escribe con el prefijo 16#, ejemplo, 16#7FFF.

- **INT** (entero) -> tipo con signo y formato de 16 bits.
- **DINT** (entero doble) -> tipo con signo y 32 bits.
- **UINT** (entero sin signo) -> tipo sin signo y formato de 16 bits.
- **UDINT** (entero doble sin signo) -> tipo sin signo y formato de 32 bits.

Tipo Time

El tipo Time se representa mediante un tipo entero doble sin signo con el prefijo T# o TIME#. Las unidades de tiempo permitidas para representar el valor son: milisegundos (MS), segundos (S), minutos (M), horas (H) y días (D). En la tabla 1.8-1 se indican las posibles formas de introducir el valor máximo del tipo Time.

Diagrama	Comentario
T#4294967295MS	Valor en milisegundos
T#4294967S_295MS	Valor en segundos/milisegundos
T#71582M_47S_295MS	Valor en minutos/segundos/milisegundos
T#1193H_2M_47S_295MS	Valor en horas/minutos/segundos/milisegundos
T#49D_17H_2M_47S_295MS	Valor en días/horas/minutos/segundos/milisegundos

Tabla 1.8-1: Ejemplos valor de variable tipo Time

1.8.3.2. Variables derivadas

Las variables derivadas son estructuras de datos que nos permiten agrupar en una misma estructura un conjunto de variables elementales.

Para crear una variable derivada es necesario crear la estructura previamente, para ello seleccionamos Tipos de datos derivados en el Explorador de proyectos. En el editor que se abrirá, le damos un nombre a la estructura y a continuación declaramos las variables (nombre y tipo) por las que estará formada.

Una vez creada nuestra estructura de datos, podemos utilizarla como tipo de datos en cualquier proyecto.

La creación de una variable derivada es igual que la de una variable elemental, pero en el tipo en lugar de elegir uno de los que viene en el programa por defecto, seleccionaremos el nombre de la estructura que hemos creado.

1.8.3.3. Variables de E/S derivadas

Las variables de E/S derivadas (IODDT), son entradas y salidas a modo de canal en una misma estructura de datos. Las IODDT son estructuras ya definidas para los distintos módulos de los autómatas que nos permiten un acceso a las E/S de los módulos mucho más simple.

1.8.3.4. Instancias FB elementales

Las instancias FB elementales son las declaraciones de los distintos bloques de función definidos en las librerías de Unity Pro.

1.8.3.5. Instancias FB derivadas

Las instancias FB derivadas son las declaraciones de los distintos bloques de función diseñados previamente dentro de un bloque DFB. Los bloque DFB, son estructuras de código que nos permite una mayor estructuración de nuestro programa, así como una reutilización máxima de todos los proyectos.

Un bloque DFB se utiliza de la misma manera que un bloque de las librerías de Unity Pro, la diferencia reside en que los bloques SFB los podemos programar para que realicen una operación concreta. Pudiéndose, de esta forma, personalizar las librerías de Unity Pro y reutilizar el código en distintas aplicaciones.

Para crear un bloque DFB seleccionamos en el explorador de proyectos la opción Tipos de FB derivadas e introducimos el nombre del bloque DFB. Luego declaramos las entradas y salidas que tendrá el bloque; las variables que usará, tanto públicas (pueden ser usadas por todo el programa) como privadas (sólo se pueden usar en ese

bloque DFB); y las acciones que realizará el bloque cuando se ejecute, las acciones se indican en la ventana Secciones.

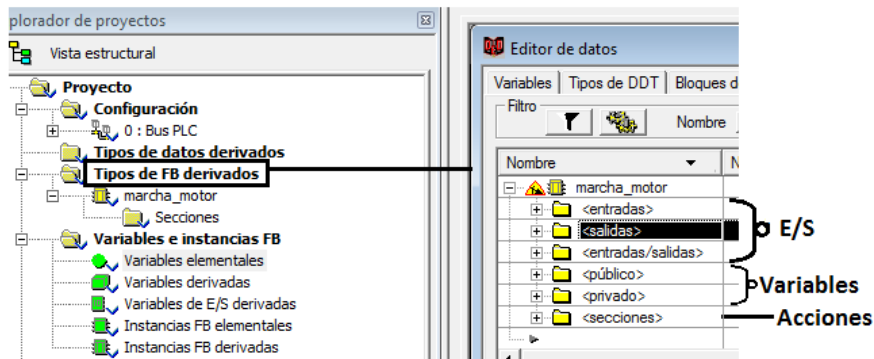


Figura 1.8-6: Crear un bloque DFB

1.8.4. Estructura de la aplicación

La estructura de una aplicación en Unity Pro puede estar compuesta por cuatro tipos de tareas: maestra, rápida, auxiliar y de procesamiento de evento. No todas las tareas tienen la misma prioridad, como puede verse en la figura 1.8-7; la tarea de evento es la de mayor prioridad de todas, la tarea rápida tiene más prioridad que la tarea maestra, y la tarea auxiliar es la que tiene menor prioridad de todas.

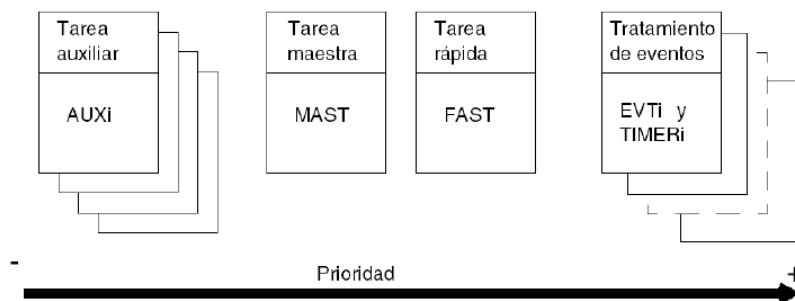


Figura 1.8-7: Nivel de prioridad de las tareas de Unity Pro

Cada tarea se puede controlar mediante programa, bits y palabras del sistema.

1.8.4.1. Tarea maestra

La tarea maestra (MAST) representa la tarea principal del programa y se compone por secciones y subrutinas. Esta tarea es obligatoria y se crea de forma predeterminada.

Cada sección de la tarea maestra se puede programar en los lenguajes LD, FBD, IL, ST o SFD, y las subrutinas se programan en LD, FBD, IL o ST y se llaman en las secciones.

El lenguaje SFC sólo se puede utilizar en las secciones de la tarea maestra y hay un número de secciones limitadas que se pueden emplear en este lenguaje.

La ejecución de la tarea maestra se puede elegir cíclica (sección predeterminada) o periódica (de 1 a 255 ms).

1.8.4.2. Tarea rápida

La tarea rápida (FAST) se compone de secciones y subrutinas, y está destinada a los procedimientos de corta duración y periódicos.

Cada sección o subrutina de la tarea rápida se puede programar en los lenguajes LD, FBD, IL o ST. Las subrutinas se requieren en las secciones de la tarea.

La ejecución de la tarea rápida es periódica, con un periodo fijado en la configuración (entre 1 y 255ms), y el programa ejecutado debe ser corto para evitar el rebasamiento de las tareas con menos prioridad.

1.8.4.3. Tarea auxiliar

Las tareas auxiliares (AUX) se componen de secciones y subrutinas. Están pensadas para las tareas cuyo tratamiento es más lento. Como no están disponibles en los autómatas Modicon M340 no se expondrán sus características.

1.8.4.4. Tarea de evento

El proceso de eventos se utiliza para reducir el tiempo de respuesta del programa de aplicación a eventos. Hay dos tipos de eventos según su procedencia: evento de E/S, procedentes de los módulos de entradas/salidas; y eventos de temporizador, procedentes de temporizadores de evento.

Estas tareas de procedimiento se ejecutan con prioridad sobre todas las demás tareas. Por tanto, son idóneas para procesar tareas que requieren un tiempo de respuesta muy corto en relación con el evento.

Una tarea procesadora de eventos se compone de una sola sección y su ejecución es asíncrona, es decir, cuando se produce un evento, el programa de aplicación es redirigido a la tarea asociada al canal de E/S o al temporizador que originó el evento.

A través del bit de sistema %S38, se puede validar o inhibir los distintos tipos de procedimientos de eventos de la aplicación, pero si se produce algún evento mientras están inhibidos se perderá el procesamiento asociado. Además existen dos funciones elementales del lenguaje, MASKEVET() y UNMASKEVET(), que permiten enmascarar o desenmascarar, respectivamente, los tratamientos de sucesos. Si se producen sucesos en el mismo momento en el que se enmascaran, el sistema los almacena y ejecutará los procesos asociados cuando sean desenmascarados.

1.8.4.4.1. Evento de temporizador

En algunas ocasiones, cuando se realiza el automatismo de una instalación, aparecen procesos temporales que requieren la actuación del autómeta cada cierto tiempo. En esos casos, se puede hacer el uso de los eventos de temporizador, pero como el tiempo en el que se repite el proceso puede ser variable, estos eventos ofrecen flexibilidad para actuar conforme a los tiempos.

Para crear un evento de temporizador hay que hacer clic derecho sobre el directorio Eventos del temporizador, situado en el Explorador de proyectos, y seleccionar Nueva sección de evento.

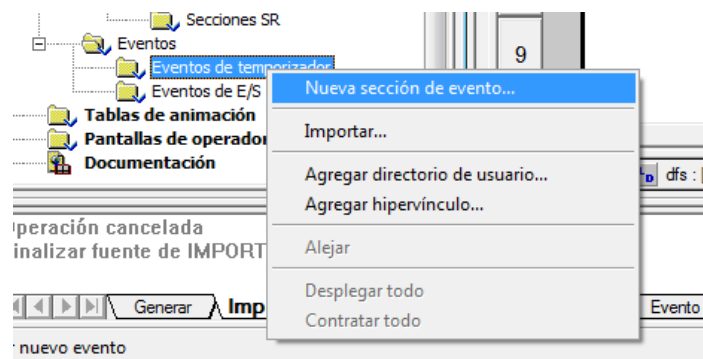


Figura 1.8-8: Crear evento de temporizador

En ese momento se desplegará la ventana que se ve en la figura 1.8-9.

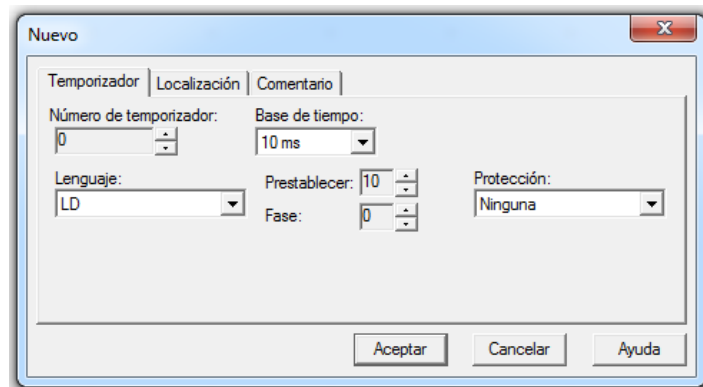


Figura 1.8-9: Configuración del temporizador del evento de tiempo

Dónde:

- Número de temporizador: es un número entero que identifica a cada evento de temporizador.
- Base de tiempo: Base de tiempo del temporizador, indica cada cuanto tiempo se incrementa el contador del timer. Están disponible los valores: 1 ms, 10 ms (valor predeterminado), 100 ms y 1s.
- Preestablecer: valor máximo del contador del timer, admite un rango entre 1 y 1023 y cuando ese valor sea alcanzado se ejecutará el evento y comenzará la cuenta nuevamente. Por tanto, el tiempo que tarda en producirse el evento se obtiene al multiplicar la base de tiempo con el valor de preestablecer.
- Fase: desplazamiento temporal entre el comienzo de la ejecución de la aplicación y el primer evento. Admite un rango de 1 a 1023 y el valor de tiempo se obtiene multiplicando la base de tiempo por el valor de la fase. El valor de la fase tiene que ser menor que el valor de preestablecer.
- Lenguaje: se selecciona el lenguaje de programación en el que se desea crear la sección. Se puede elegir entre los lenguajes: LD, ST, IL y FBD.
- Protección: es la seguridad que le queremos poner a la sección: Ninguna (sin protección), Sólo lectura (protección frente a escritura), Sin lectura ni escritura (protección frente a escritura y lectura).

La función ITCNTRL es un temporizador que activa el proceso de eventos de tipo Timer.

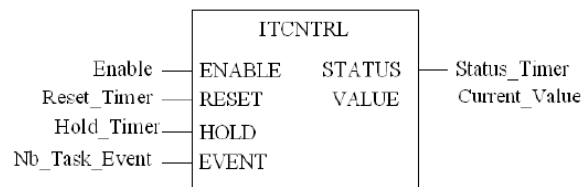


Figura 1.8-10: Representación FBD de la función ITCNTRL

En la tabla 1.8-2 se describen los parámetros de entrada y en la tabla 1.8-3 los de salida de la función ITCNTRL.

Parámetros	Tipo	Observación
Enable	BOOL	Entrada de validación. En estado 1: el procesamiento de eventos se activa una vez transcurrida la temporización. En estado 0: no se ha emitido ningún evento
Reset_Timer	BOOL	En estado 1 reinicializa el temporizador.
Hold_Timer	BOOL	En estado 1 inmoviliza el incremento del temporizador.
Nb_Task_Event	BYTE	Palabra de entrada que determina el número del procesamiento de eventos TIMER que se va a activar.

Tabla 1.8-2: Parámetros de entrada de la función ITCNTRL

El valor Nb_Task_Event debe coincidir con el número de temporizador que debe ser ejecutado cuando se alcanza el valor máximo del contador del temporizador.

Parámetros	Tipo	Observación
Status_Timer	WORD	Palabra de estado: <ul style="list-style-type: none"> • bit 0 = 1; ejecución retardada por un enmascaramiento de la interrupción. • bit 1 = 1; número del procesamiento de eventos no válido • bit 2 = 1; temporizador validado (imagen de la entrada Enable). • bit 3 = 1; temporizador "inmovilizado" (imagen de la entrada Hold_Timer). • bit 4 = 1; cuando ITCNTRL se solicita la primera vez con la entrada Reset_Timer o Hold_Timer a 1 (modo anterior). Se pone a cero en caso de arranque en frío. • bit 5 = 1; pila FIFO de almacenamiento de las interrupciones saturada.
Current_Value	TIME	Valor actual de la temporización. Este valor se incrementa de 0 al valor preestablecido. Una vez alcanzado el valor preestablecido, vuelve a cero. Si se valida el procesamiento de eventos de tipo TIMER asociado, éste se ejecuta.

Tabla 1.8-3: Parámetros de salida de la función ITCNTRL

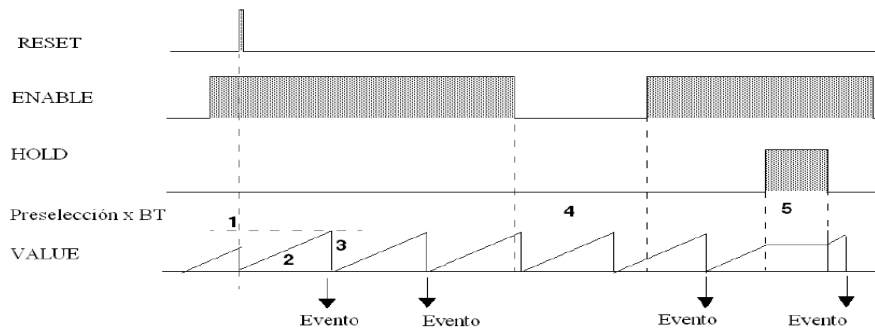


Figura 1.8-11: Tendencias de funcionamiento normal de eventos de temporizador

1.8.4.4.2. Evento de E/S

Cuando se producen cambios tan rápidamente en las entradas o salidas del autómatas, que a este no le da tiempo a procesarlas, se emplean eventos de entrada/salida. Si esto ocurre, se puede usar un equipo de conteo rápido que genera un evento cuando se ha llegado a un valor de cuenta sobre un evento externo que cambia rápidamente. Entonces se interrumpe la ejecución normal del autómatas y se ejecuta el programa del Evento E/S asociado.

Para crear un evento de E/S hay que hacer clic derecho sobre el directorio Eventos de E/S, situado en el Explorador de proyectos, y seleccionar Nueva sección de evento.

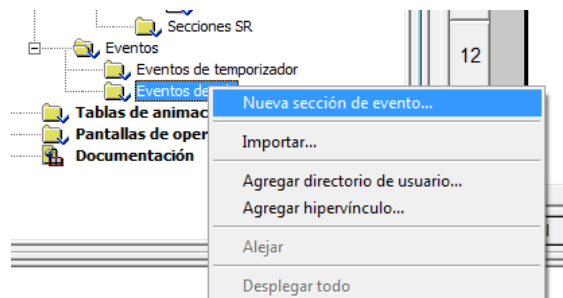


Figura 1.8-12: Crear Evento E/S

En ese momento se desplegará la ventana que se ve en la figura 1.8-13.

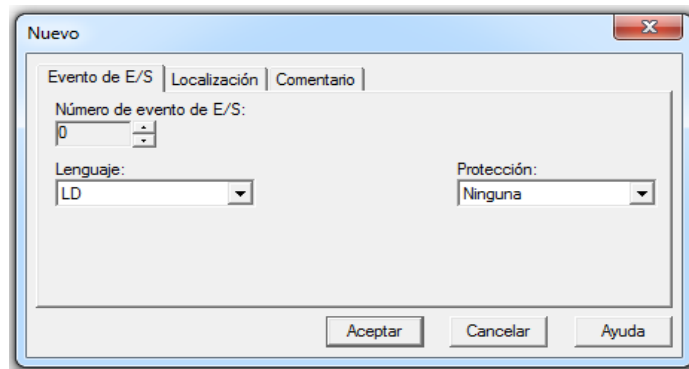


Figura 1.8-13: Configurar Evento de E/S

Dónde:

- Número de evento de E/S: es un número entero que identifica a cada evento.
- Lenguaje: se selecciona el lenguaje de programación en el que se desea crear la sección. Se puede elegir entre los lenguajes: LD, ST, IL y FBD.
- Protección: es la seguridad que le queremos poner a la sección: Ninguna (sin protección), Sólo lectura (protección frente a escritura), Sin lectura ni escritura (protección frente a escritura y lectura).

1.8.5. Lenguajes de programación

En el programa Unity Pro hay 5 lenguajes de programación: SFC, ST, LD, IL y FBD; todos ellos cumplen la norma IEC 61131-3. Una aplicación se desarrolla haciendo uso de secciones. La sección de programación permite escribir el programa que desea ejecutar en cualquiera de los lenguajes posibles. El lenguaje SFC es exclusivo para las secciones de la tarea maestra (MAST), mientras que los otros cuatro se pueden usar en cualquier sección.

Para crear una sección hay que hacer clic derecho sobre el directorio Secciones, situado en el Explorador de proyectos, y seleccionar Nueva sección.

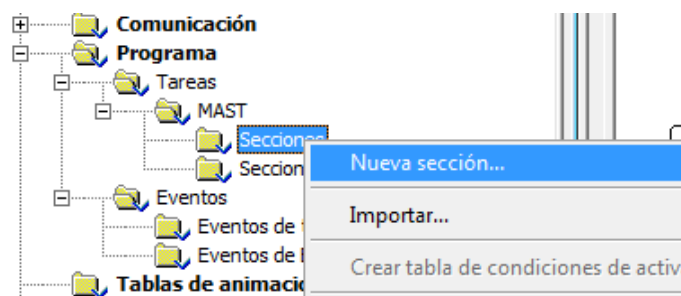


Figura 1.8-14: Crear nueva sección

Aparecerá una ventana, figura 1.8-15, en la que se tendrá que introducir el nombre de la sección (máximo 32 caracteres y sin espacio) y seleccionar un lenguaje de programación entre todos los posibles.

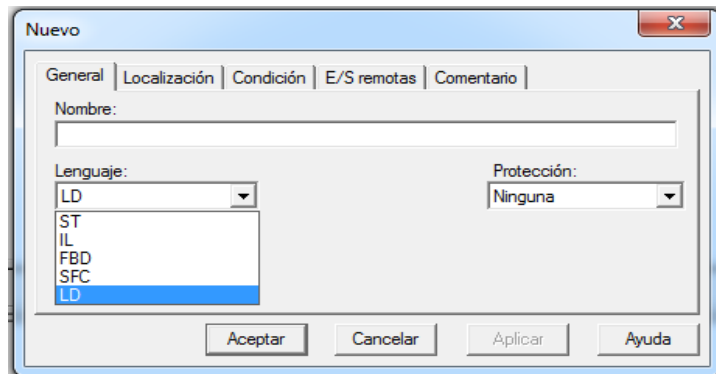


Figura 1.8-15: Configurar nueva sección

Es posible crear varias secciones con lenguajes diferentes. El orden de ejecución de las secciones será definido por el orden de las diferentes secciones creadas. Es posible cambiar este orden arrastrando una sección antes o después de otra.

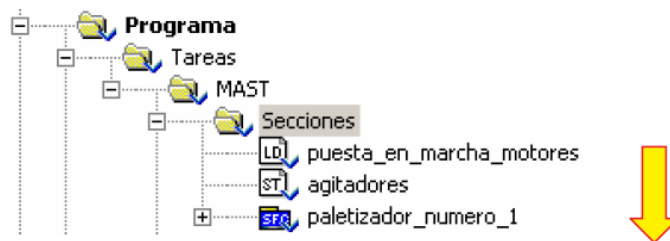


Figura 1.8-16: Orden de ejecución de las secciones

1.8.6. Configuración de una red Ethernet

En general, Ethernet está diseñada para facilitar: z la coordinación entre controladores programables, z la supervisión local o centralizada, z la comunicación con el procesamiento de datos empresariales de producción, z la comunicación con entradas/salidas remotas.

Lo primero que tenemos que hacer es crear la red Ethernet haciendo clic derecho en la carpeta “Redes” del “Explorador de proyectos”, luego hacemos doble clic en la red creada e introducimos la dirección de red y la máscara de señal en la ventana que se abrirá, como se ve en la figura 1.8-17 .

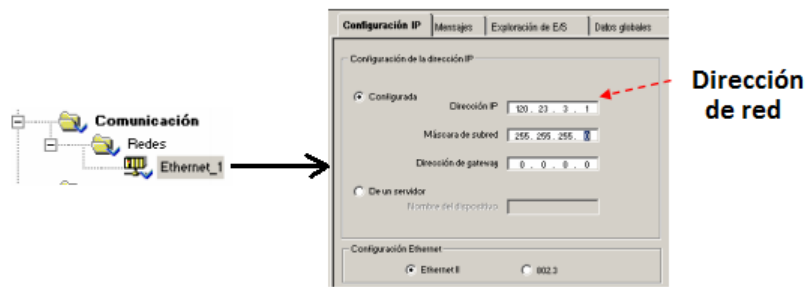


Figura 1.8-17: Crear red Ethernet

Lo siguiente que hay que hacer es vincular la red que acabamos de crear con un módulo del PLC, en nuestro caso el módulo P34 2020. Para esto hacemos doble clic en la conexión de Ethernet del módulo y le asignamos la red que hemos creado en la ventana que se abre, como se ve en la figura 1.8-18.

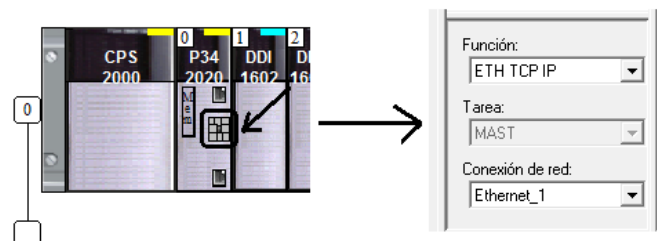


Figura 1.8-18: Vincular red Ethernet al módulo de comunicación

Ya que hemos vinculado la red a nuestro PLC, crearemos una sección para leer las variables. Esta sección se realizará en lenguaje LD, y para ello usaremos los bloques de funciones ADDM y READ_VAR.

La función ADDM se utiliza para convertir una cadena de caracteres en una dirección que las siguientes funciones de comunicación pueden utilizar directamente.

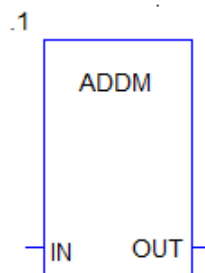


Figura 1.8-19: Bloque de función ADDM

La función READ_VAR se utiliza para leer el valor de uno o más objetos de lenguaje. Los objetos de lectura deben ser siempre consecutivos. Pueden estar ubicados en una CPU remota o en un dispositivo conectado a un canal de

comunicación. En los PLC Modicon M340, la función READ_VAR puede leer hasta 2000 bits consecutivos en un dispositivo remoto.

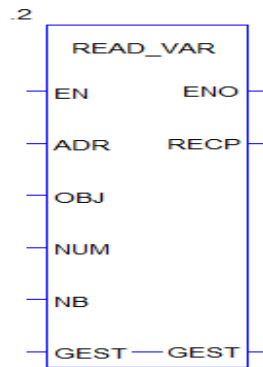


Figura 1.8-20: Bloque de función READ_VAR

1.9. Vijeo Designer

En este capítulo se hará una breve descripción con los conceptos más importante que se deben conocer sobre el programa Vijeo Designer para desarrollar una aplicación para un dispositivo HMI.

1.9.1. Crear un nuevo proyecto

Un proyecto es un archivo creado en el software Vijeo Designer en el que se incluye toda la información (dibujos, alarmas, información del hardware, etc.) para crear un ambiente para la aplicación de supervisión creada por el usuario.

Si se ha abierto el programa Vijeo Designer por primera vez se abre el asistente que nos guía en la creación de un nuevo proyecto. Para crear un nuevo proyecto seleccionamos la opción Crear un nuevo proyecto y hacemos clic en siguiente, como se muestra en la imagen 1.9-1.

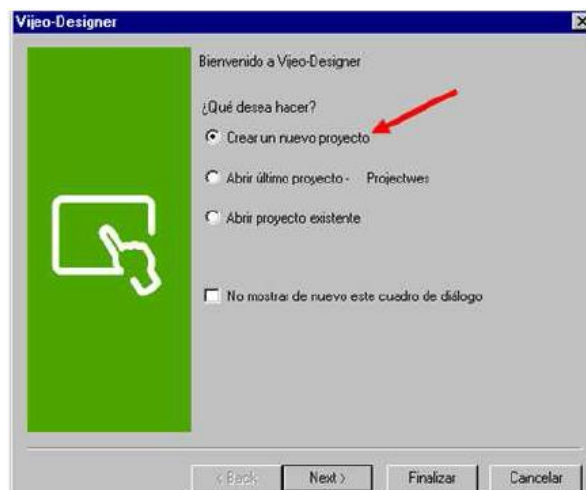


Figura 1.9-1: Crear un nuevo proyecto de Vijeo Designer

En la figura 1.9-2 se ven las ventanas de configuración del proyecto. Primero se abre la ventana A, y cuando se rellenan todos los campos y se pulsa siguiente, se abre la ventana B.

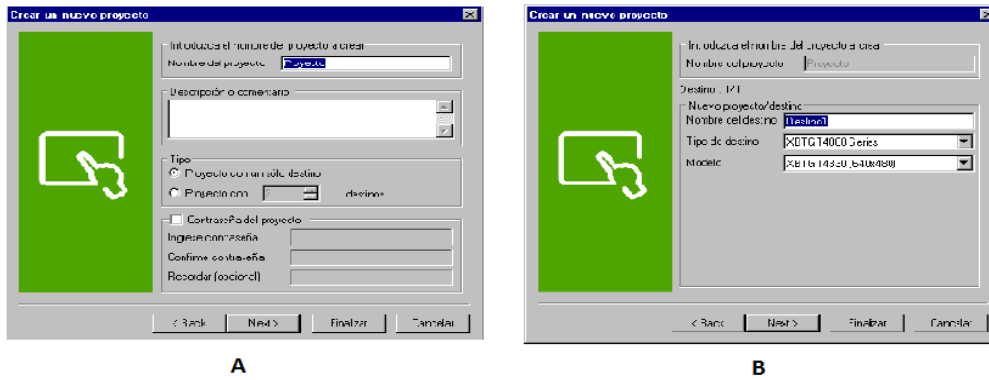


Figura 1.9-2: Configuración del proyecto de Vijeo Designer

En la ventana A de la figura 1.9-2 tenemos los siguientes campos:

- **Nombre del proyecto.**
- **Descripción o comentario.** (opcional)
- **Tipo:** Seleccione si un proyecto tiene un único destino o varios. Si tiene varios destinos, hay que especificar el número.
- **Contraseña del proyecto:** si quiere ponerle contraseña al proyecto debe habilitarlo e ingresar la contraseña y el recordatorio de contraseña

En la ventana B de la figura 1.9-2 tenemos los siguientes campos:

- **Nombre del destino:** Escriba un nombre para su destino.
- **Tipo de destino:** Seleccione que familia de Magelis desea como.
- **Modelo:** de la familia Magelis seleccionadas elija el modelo específico que se desea como destino.

Una vez que se ha terminado de introducir los datos de la ventana B de la figura 1.9-2 y se pulsa siguiente, aparecen las ventanas de configuración del proyecto que se ven en la figura 1.9-3. Primero se abre la ventana A, en la que se introduce la **'dirección IP'** de la máquina de destino (si tiene conexión Ethernet), y cuando se rellenan todos los campos y se pulsa siguiente, se abre la ventana B, en la que se selecciona el equipo con el que se va a comunicar el terminal.

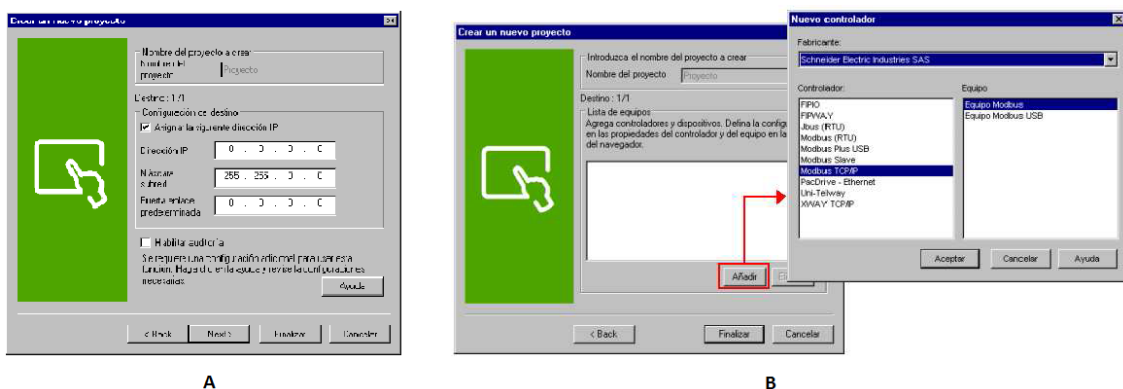


Figura 1.9-3: Configuración del proyecto de Vijeo Designer (continuación)

1.9.2. Interfaz de usuario

La interfaz de usuario se compone de varias ventanas y barras de herramientas que ayudan a una mayor comprensión y orden a la hora de realizar la aplicación.

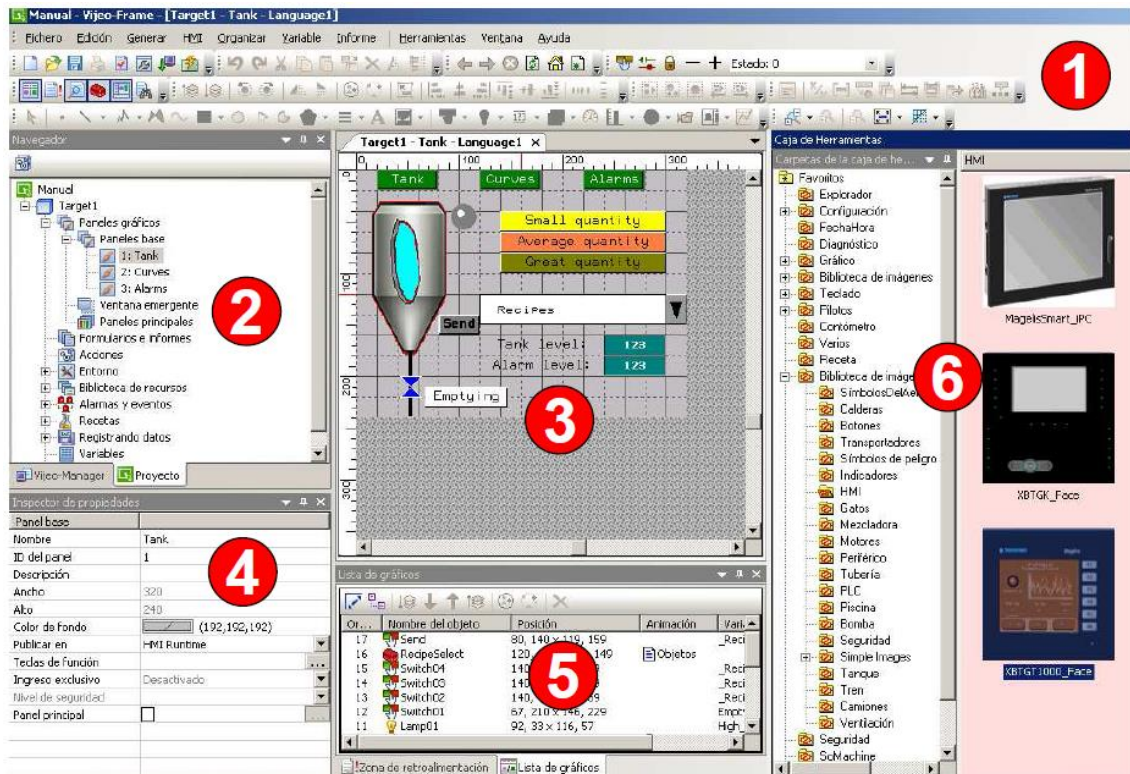


Figura 1.9-4: Interfaz de usuario de Vijeo Designer

1 -> Barras de herramientas: En esta área se muestra el menú contextual y las barras de iconos de las diferentes acciones que se pueden realizar en el proyecto.

2 -> Navegador: Esta área muestra el árbol de las diferentes partes que puede tener la aplicación.

3 -> Área de Trabajo: Es la zona donde el usuario tendrá que trabajar para crear la aplicación.

4 -> Inspector de Propiedades: En esta área se muestra las propiedades que se pueden parametrizar de los objetos seleccionados.

5 -> Zona de retroalimentación/Lista de gráficos: Según la pestaña seleccionada: en la zona de retroalimentación se verá los mensajes de compilación (errores de compilación y advertencias), en cambio en la pestaña de listado de gráficos muestra todos los objetos que hay en ese momento en el panel que hay en el área de trabajo.

6 -> **Caja de herramientas:** Podemos ver los diferentes objetos e imágenes que ya hay creados y guardados en la librería y que podemos incluir en nuestra aplicación y que nos ayudarán a realizarla.

1.9.3. Comunicaciones

Puede comunicarse con autómatas, inversores y otros equipos mediante el puerto serie de la máquina de destino (RS-232C / RS-422), el puerto Ethernet o el módulo/tarjeta de comunicación. Vijeo Designer emplea controladores para permitir la comunicación con equipos, sin que sea necesario escribir programas de comunicación.

Para añadir un controlador de dispositivo hay que hacer clic derecho en el nodo Administrador de E/S y selecciona Nuevo controlador, como se ve en la figura 1.9-5. Una vez hecho esto se sigue un procedimiento parecido al expuesto en el apartado 1.9.1.

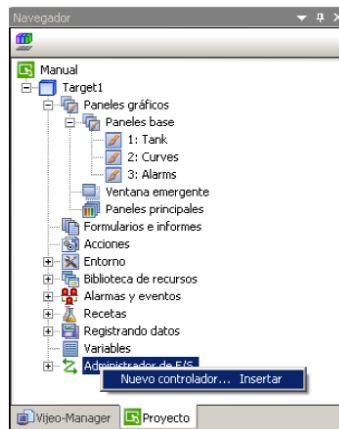


Figura 1.9-5: Añadir un nuevo controlador

1.9.4. Paneles

Un panel es una ventana en la que se pueden colocar objetos (interruptores, lámparas, dibujar elementos). El panel creado se convierte en la pantalla que aparece en las máquinas de destino. El nodo de paneles gráficos proporciona los siguientes tipos de paneles:

- **Paneles principales:** En un panel principal, puede dibujar objetos y poner objetos de la caja de herramientas para crear un panel de objetos comunes que puedan usarse para todos los paneles base en un proyecto.
- **Paneles de ventana emergente:** En el panel de una ventana emergente, puede dibujar objetos y poner y ubicar objetos de la caja de herramientas. Puede abrir el panel de la ventana emergente sobre el panel actual durante Runtime usando un interruptor, animación de contacto o un script.
- **Paneles base:** Un panel es el lienzo en el cual dibuja objetos y pone objetos en la Caja de herramientas.

Para crear un nuevo panel hay que hacer clic derecho en Paneles base en el nodo Paneles gráficos y se selecciona la opción Nuevo panel, como muestra la imagen 1.9-6.



Figura 1.9-6: Crear nuevo panel en Vijeo Designer

Los paneles tienen una serie de variables que se pueden modificar en la ventana Inspector de propiedades.

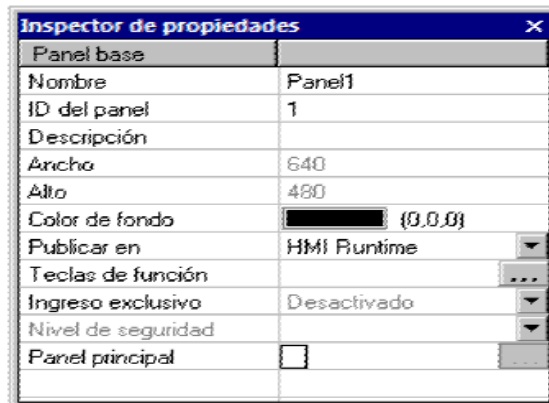


Figura 1.9-7: Ventana Inspector de propiedades

Estas propiedades son:

- **Nombre:** nombre del panel.
- **ID del panel:** es un número que identifica al panel. Ese número está entre el 1 y el 8999 para un panel almacenado en el nodo Paneles gráficos o entre 10001 y 18999 para un panel almacenado en el nodo Ventana emergente.
- **Color de fondo.**
- **Publicar en:** selecciona si el panel se publicará en la máquina de destino (HMI Runtime), Web Gate o en ambos.
- **Nivel de seguridad:** define el nivel de seguridad del panel.

1.9.5. Variables

Las variables pueden ser asociadas a interruptores, lámparas, visualizaciones de datos y otros objetos en el panel.

Para comunicarse con autómatas y otros equipos conectados a la máquina de destino, hay que crear una variable y asignar una dirección de dispositivo. Esas variables se actualizan cada vez que cambian los datos del equipo.

Por tanto, hay dos tipos de variables: externas e internas. Las variables internas sólo se usan para operaciones internas de Vijeo Designer, ya que no tienen ninguna dirección de equipo a la que está asociada; mientras que las variables externas están asociadas a una dirección. En un mismo panel se pueden usar variables externas e internas.

Hay 6 tipos de variables elementales (discreto, entero, flotante, cadena, bloque entero y bloque flotante y también el tipo estructura (carpeta que contiene múltiples variables agrupadas)).






 Discreto	Almacena un valor de 0 o 1.
 Entero	Las variables enteras internas pueden almacenar valores de -2147483648 a +2147483647.
 Flotante	Almacena un valor numérico, de -3.4028e38 a +3.4028e38, o de -1.1754e-38 a +1.1754e-38.
 Cadena	Almacena de 2 a 100 caracteres.
 Estructura	Las estructuras, al igual que las matrices, actúan como carpetas que almacenan múltiples variables. Optimice la administración de las variables mediante la organización de éstas en estructuras. No puede crear estructuras agrupadas (estructuras dentro de otra estructura).

Tabla 1.9-1: Variables elementales de Vijeo Designer

Es posible importar variables creadas en otros programas al programa Vijeo Designer. La posibilidad de vincular los nombres y las direcciones, símbolos y variables configuradas en un software de configuración de un PLC a las variables de Vijeo Designer.

Para poder importar variables creadas en Unity Pro es necesario exportarlas en un formato compatible con Vijeo Designer, en este caso son los archivos con extensión .xvm o .stu.

1.9.6. Animación de gráficos

La animación de objetos se utilizan para que el interface entre el proceso y el operario se más claro y sencillo. Con este fin se puede modificar el tamaño y la posición de los objetos de un panel y generar movimientos ópticos en pantalla con objetos simples.

Hay diferentes tipos de animación: de color, de relleno, de tamaño, de posición, de giro, de pulsación, de valor, de visibilidad y de contacto; que se pueden aplicar a los objetos. A continuación veremos algunos de estos casos

Animación de color

Puede modificarse el color de un objeto en función de los cambios de los valores de variable.

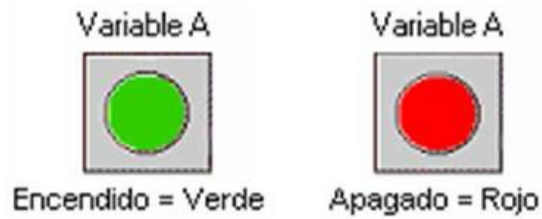


Figura 1.9-8: Animación de color

Animación de relleno

Se puede llenar gradualmente un objeto con color, tanto vertical como horizontalmente, basado en los cambios de valor de la variable.



Figura 1.9-9: Animación de relleno

Animación de valor

Puede visualizar un valor de variable en el panel o permitir la introducción de datos en la variable mediante el teclado. También puede añadir animación de valor a las cadenas de texto para una introducción mediante la pantalla o el teclado. Si utiliza una variable discreta, puede visualizar diferentes mensajes en función del estado de las variables discretas

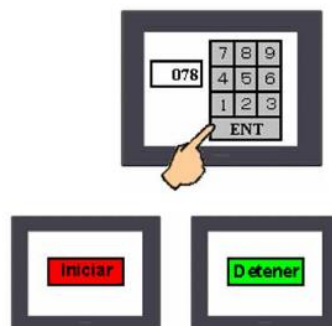


Figura 1.9.10: Animación de valor

1.9.7. Visualización e introducción de datos

En Vijeo-Designer, se puede visualizar e introducir varios tipos de datos en paneles gráficos: datos numéricos, fecha y hora, cadenas y archivos de texto (.txt).

Los campos de entrada de datos visualizados en runtime usan teclados numéricos para la entrada de datos.



Figura 1.9.11: Visualización de datos en Runtime

1.9.8. Ventanas emergentes

Si emplea una ventana emergente para la introducción de datos mediante el teclado, puede abrirla cuando sea necesario y cerrarla cuando haya finalizado la introducción.

Se pueden ver hasta tres ventanas emergentes al mismo tiempo, si pertenece cada una a un grupo de ventanas emergentes distinto. No pueden verse al mismo tiempo múltiples ventanas emergentes de un mismo grupo de ventanas emergentes.

Las ventanas emergentes constan de tres componentes: Carpeta, grupos y paneles.



Figura 1.9.12: Componentes de las ventanas emergentes

Carpeta de ventana emergente

Se genera automáticamente cuando se crea un proyecto. Puede crear grupos de ventanas emergentes y paneles de ventanas emergentes. No se puede copiar, eliminar o cambiar de nombre.

Grupo de ventanas emergentes

Puede agregar múltiples paneles de ventanas emergentes a un grupo. Cuando abre la ventana emergente, puede modificar el panel, siempre que éste pertenezca al mismo grupo.

Panel de ventana emergente

Crear uno o más paneles de ventanas emergentes dentro de un grupo. Puede definir la visualización de determinados paneles de ventanas emergentes dentro de dicho grupo.

1.9.9. Alarmas

Se puede indicar al usuario las alarmas en varios formatos de visualización.

Visualización de Resumen de alarmas

Resumen de alarmas en un panel. Donde se presenta su estado: activo, reconocido, no reconocido y retorno a normal.



Figura 1.9.13: Visualización de Resumen de alarmas

Aviso de alarma

Mensajes de alarmas activos y no reconocidos. Si múltiples alarmas se activan al mismo tiempo, los mensajes de aviso de alarmas se visualizan en el orden de activación.

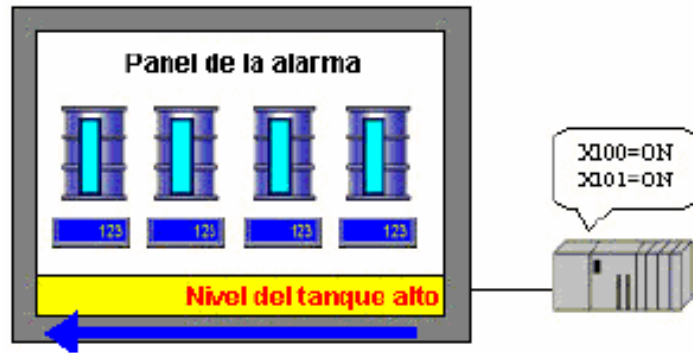


Figura 1.9.14: Aviso de alarma

Sonido

Alertar a los usuarios de la activación de una alarma.

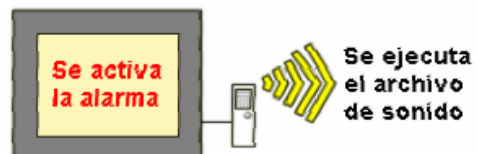


Figura 1.10.16: Alarma de sonido

1.10. Bibliografía

Unity Pro. Lenguajes y estructura del programa. Manual de referencia

Modicon M340 con Unity Pro. Procesadores, bastidores y módulos de fuente de alimentación. Manual de configuración.

Modicon M340 con Unity Pro. Módulos de entradas/salidas binarias. Manual de usuario.

Unity Pro. Modalidades de funcionamiento.

Unity Pro. Estándar. Librería de bloques.

Tutorial Práctico Unity Pro 3.0 – Modicon M340

ISA-UMH Grafcet Estructurado

Modicon M340 para Ethernet. Procesadores y módulos de comunicaciones. Manual de usuario.

Tema 4: GRAFCET. Diseño e implantación de autómatas. Departamento de Ingeniería de Sistemas y Automática. Universidad de Málaga

Introducción al modelado GRAFCET. Universidad Politécnica de Madrid
http://www.elai.upm.es/moodle/pluginfile.php/1171/mod_resource/content/0/GrafcetAmpliacion.pdf

Unity Pro, Sistema, Librería de bloques

Grafcet Diagrama para el diseño de automatismos. Profesor: Pedro Valverde Montero C.G.S. Desarrollo de productos electrónicos.

Funciones fundamentales para controlar los SFC. Universidad de Oviedo

Automatización de procesos mediante la guía GEMMA. Pere Ponsa Asensio, Ramón Vilanova Arbós.

Vijeo Designer. Manual de formación.

Redes de Petri: Modelado e implementación de algoritmos para autómatas programables. Autor. Luis Diego Murillo.

Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías
Industriales

ANEXOS

Autor: Antonio Luis Velasco Martín

Tutor: Luis Fernando Castaño Castaño

Dep. Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla



Sevilla, 2015



Índice de los Anexos

2.1. Guía GEMMA: Automatización del almacenador de bandejas	pág. 71
2.1.1. Grupo f: Procedimiento de funcionamiento	pág. 71
2.1.2. Grupo A: Procedimiento de paradas y puesta en marcha	pág. 72
2.1.3. Grupo D: Procedimiento de defecto.....	pág. 73
2.2. Programar PLC mediante el programa Unity Pro	pág. 74
2.2.1. Crear el proyecto.	pág. 74
2.2.2. Configurar el hardware.	pág. 74
2.2.3. Configurar la comunicación Ethernet	pág. 76
2.2.4. Lista de señales internas y externas	pág. 76
2.2.5. PE	pág. 79
2.2.6. GEMMA.....	pág. 82
2.2.7. MA.....	pág. 84
2.2.8. MP.....	pág. 91
2.2.9. PEI	pág. 95
2.2.10. PB	pág. 96
2.2.11. NBANDJ	pág. 99
2.2.12. FALLO	pág. 100
2.2.13. FCFRP Y FCFRE	pág. 101
2.2.14. AERP Y AERE	pág. 103
2.2.15. FCFCE Y FCFCG.....	pág. 104
2.2.16. AECE	pág. 106
2.2.17. AECG.....	pág. 107
2.2.18. ITP	pág. 109
2.2.19. COM_ETH	pág. 110
2.3. Programar Pantalla Táctil mediante el programa Vijeo Designer	pág. 112
2.3.1. Crear el proyecto	pág. 112
2.3.2. Comunicación con un PLC a través de TCP/IP	pág. 112
2.3.3. Lista de variables	pág. 113
2.3.4. Pantalla Principal	pág. 114
2.3.5. Pantalla Guía GEMMA	pág. 115
2.3.6. Pantalla de información del almacén	pág. 116
2.3.7. Pantalla panel de control	pág. 117
2.3.8. Pantalla de alarmas.....	pág. 118

2.1. Guía GEMMA: Automatización del alimentador de bandejas

Una vez analizados todos los estados y el modo de empleo de la Guía GEMMA, podemos aplicarla al caso que nos ocupa, la gestión automática del almacenador de bandejas.

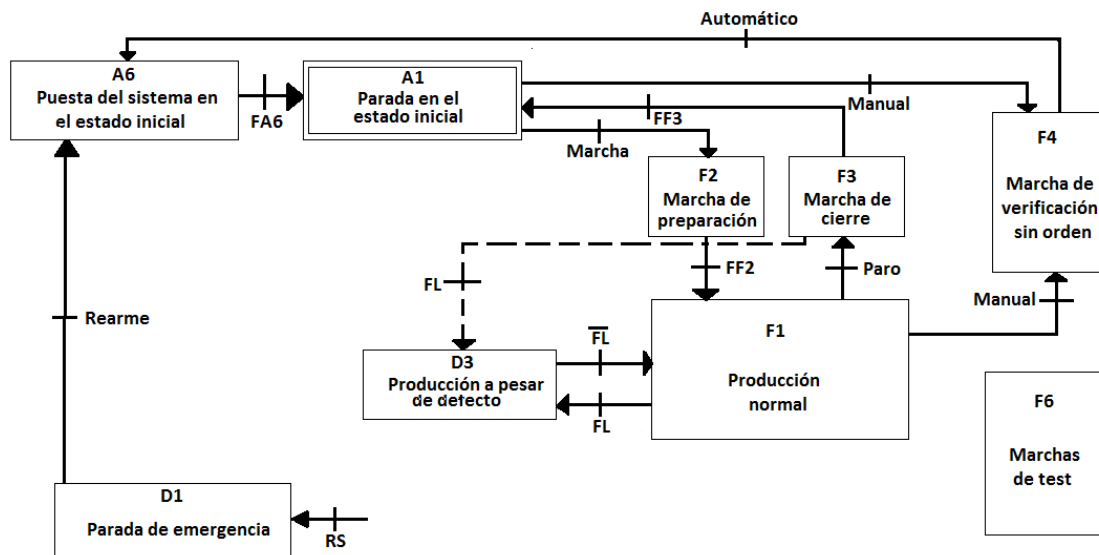


Figura 2.1-1: Guía GEMMA del sistema

A continuación se procede a la explicación de las tres situaciones que controla la Guía GEMMA que hemos diseñado.

2.1.1. Grupo F: Procedimiento de funcionamiento

F1 – Producción normal

Es el modo de funcionamiento en el que el sistema se ejecuta de forma totalmente automática. Puede funcionar en dos modos: local y remoto.

En el caso de funcionamiento local, las peticiones de servir y almacenar una bandeja se realizará usando los pulsadores del panel de control. Estas peticiones se podrán anular, siempre que no haya comenzado a llevarse a cabo, usando los dos botones que hay en la pantalla de explotación.

En el funcionamiento automático en modo remoto, las peticiones de servir o almacenar una bandeja, así como la anulación de estas peticiones, las realizará el programa del autómatas principal a través de una comunicación Ethernet que se explicará en el apartado 2.2.19.

F2 – Marcha de preparación

En este estado se realiza la comprobación y asignación del tiempo que tarda en producirse cada movimiento de los actuadores, e iniciar el contador que indica el número de bandejas que hay almacenada.

F3 – Marcha de cierre

Cuando el sistema recibe la instrucción de parada pasa a este estado, en el que permanecerá hasta que se haya terminado todas las acciones pendientes y se hayan recogidos todas las bandejas. Pero no se aceptará ninguna petición más de servir o almacenar bandejas.

F4 – Marcha de verificación sin orden

En este modo se puede realizar el movimiento de todos los actuadores del sistema de manera independiente.

F6 - Marchas de test

Este estado estará funcionando paralelamente a los modos de funcionamiento de producción normal (F1) y de marcha de cierre (F3), y será el encargado de ejecutar las instrucciones que detecten si hay algún problema en la instalación.

En caso de que se detecte el fallo de un actuador, realizará una comprobación periódica para comprobar si se ha resuelto la incidencia.

En los demás estados de la Guía GEMMA se supone que un operario está junto a la máquina, y en caso de fallo de funcionamiento de alguna de las partes, este lo detectará sin necesidad de apoyo de la aplicación.

2.1.2. Grupo A: Procedimiento de parada y puesta en marcha

A1 – Parada en el estado inicial

Es el estado normal de reposo de la instalación. Coincide con el cilindro elevador en la posición baja y los dos cilindros retenedores en la posición baja.

A6 – Sistema en estado de reposo

En este caso se pretende que el sistema se ponga automáticamente en el punto considerado como inicial. Éste corresponde con los dos cilindros retenedores y el cilindro elevador en la posición baja. Esta posición coincide con la que tendrá el sistema cuando, en modo automático, no esté realizando ningún ciclo.

2.1.3. Grupo D: Procedimientos de defecto.

Cabe destacar que se ha descartado el estado D2 porque estas acciones se llevarán a cabo a través de mensajes de alarma en el dispositivo HMI que se encuentra en el panel de control.

D1 – Parada de emergencia

Es el estado de mayor prioridad del sistema. Cuando este estado se activa se debe detener el proceso que se esté realizando y el sistema debe evolucionar hasta la posición de seguridad. Ésta posición tendrá los cilindros retenedores en la posición baja y el cilindro elevador en la baja, además, si el cilindro de giro está en movimiento, éste se detendrá inmediatamente.

D3 - Producción a pesar de los defectos

Este es el estado al que evoluciona el sistema si se detecta algún fallo grave.

Se considera fallo leve a aquel que por sus características permita al sistema seguir operando, total o parcialmente al sistema, aunque en condiciones menos óptimas usando tiempos de ejecución más largos, como pueden ser:

- Fallo en los sensores de posición del cilindro elevador o de giro: en este caso se operará por tiempos.
- Fallo en el retenedor previo: no se puede servir una bandeja, ya que si llega otra cuando el cilindro elevador esté arriba, al bajar la bandeja estas quedarán solapadas. Pero el sistema podría seguir operando con las bandejas que se encuentren en la cinta, aunque se necesiten mayores tiempos de espera para cada pallet y también se podrían almacenar para detener la máquina.
- Fallo en el retenedor de elevación: en este caso no se puede almacenar una bandeja, ya que no se podría detener en la posición correcta para que el cilindro elevador la eleve. Pero si se podría servir una bandeja.

Se considera fallo grave a aquel que por sus características impida la continuación de la ejecución del sistema, como pueden ser:

- Cintas transportadoras detenidas: en este caso no se podrá realizar ninguna acción ya que no llegarán bandejas al almacenador para ser recogidas ni se despejará este para servir otra.
- Presión del sistema neumático insuficiente: el sistema no podría mover los actuadores.
- Fallo en el cilindro de giro: si no se puede girar el almacén, no se liberarán ni bloquearán las bandejas, y por tanto, no se podrá almacenar ni servir ninguna.
- Fallo en el cilindro elevador: si no se puede mover el cilindro elevador no se podrá subir ni bajar ninguna bandeja.

2.2. Programar PLC mediante el programa Unity Pro

2.2.1. Crear el proyecto

Cuando abramos el programa Unity Pro, lo primero que tenemos que hacer es pulsar el icono “Nuevo proyecto” y se nos abrirá una ventana para seleccionar el PLC que vamos a utilizar. Una vez que lo hemos seleccionado entre todos los que aparecen, en nuestro caso el modelo Modicon M340 BMX P34 2020, pulsamos “Aceptar” y el nuevo proyecto se habrá creado. En la figura 2.2-1 podemos ver dichos pasos.

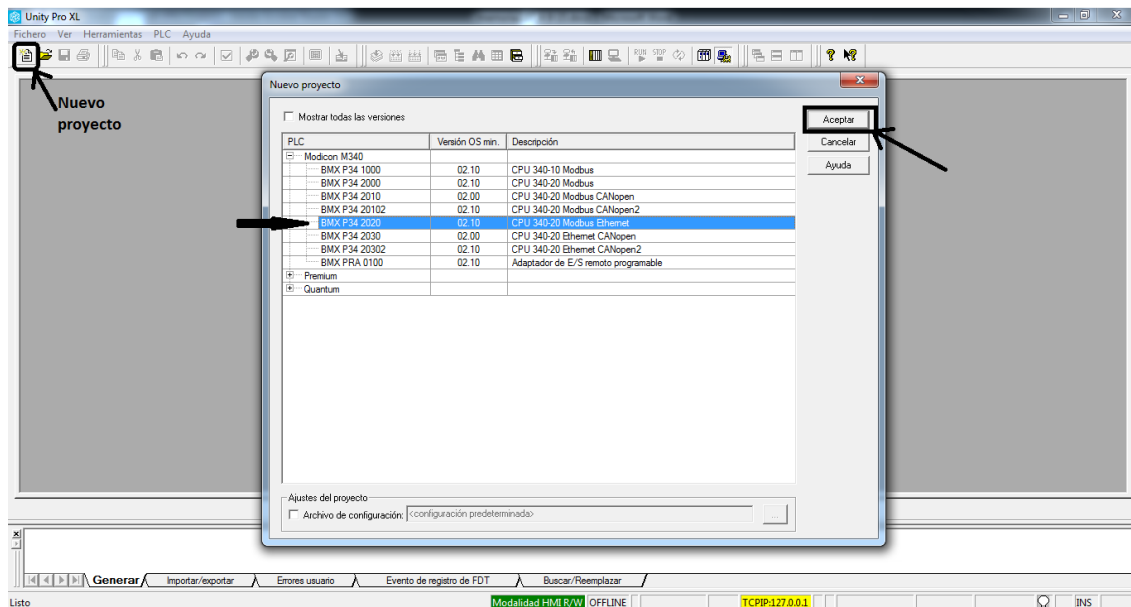


Figura 2.2-1: Creación del proyecto

2.2.2. Configurar el hardware

Lo siguiente que hay que hacer es configurar todos los módulos con los que está compuesto nuestro PLC. Para ello, como se muestra en la figura 2.2-2, hay que hacer doble clic sobre el bastidor del PLC dentro del “Explorador de proyectos” y se abrirá la ventana de edición de módulos. En estos momentos sólo tendremos la fuente de alimentación y el procesador que seleccionamos al crear el proyecto.

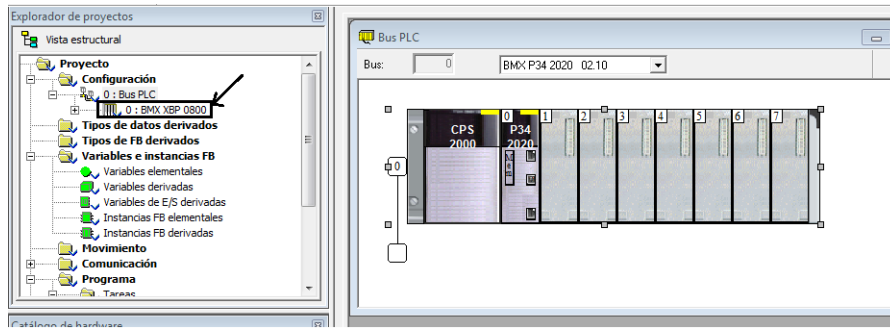


Figura 2.2-2: Ventana de configuración del PLC

Para añadir nuevos módulos basta con hacer doble clic sobre una ranura vacía y se desplegará la ventana “Nuevo dispositivo”, figura 2.2-3, donde seleccionaremos el dispositivo a añadir. Es importante para el correcto funcionamiento del programa que se añadan todos los dispositivos en la misma ubicación que están en el aparato real.

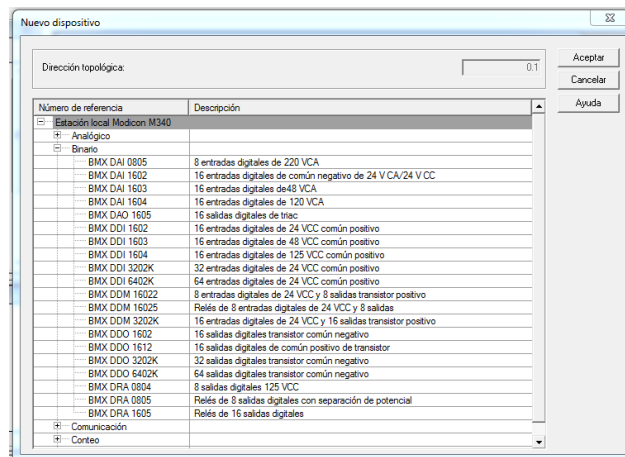


Figura 2.2-3: Ventana Nuevo dispositivo

En nuestro caso debemos añadir el módulo DDI 1602 en la ranura 1 y el módulo DDM 16022 en la ranura 2, quedando finalmente el PLC como se representa en la figura 2.2-4.

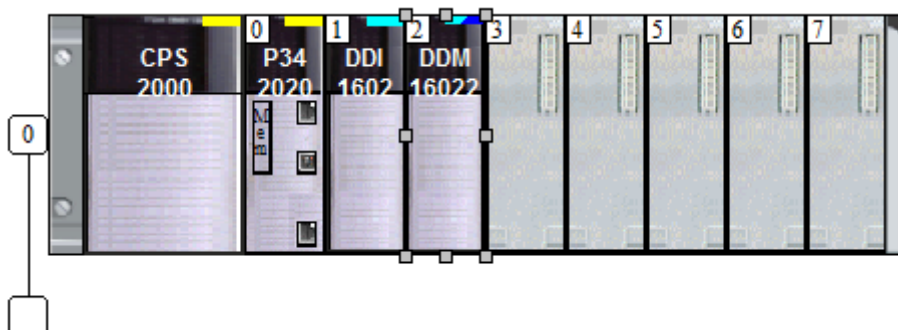


Figura 2.2-4: Composición del PLC

2.2.3. Configurar la comunicación Ethernet

Para configurar la comunicación Ethernet en este proyecto hemos seguido los pasos que se exponen en el apartado 1.9.6. La sección para leer variables tiene los siguientes elementos que se observan en la figura 2.2-5. En nuestro caso la dirección IP es 192.168.0.15

2.2.4. Lista de señales internas y externas

En el PLC contamos con un total de 24 entradas digitales y 8 salidas digitales entre los módulos DDI 1602 y DDM 16022. A continuación podemos encontrar unas tablas en la que se indican las variables que hay conectadas a cada entrada o salida, incluyendo la dirección, y una pequeña descripción sobre su función.

Además de estas señales externas, el programa cuenta con un gran número de señales internas, por tanto, podremos encontrar otra tabla para estas señales en la que se incluirá su nombre, el tipo de variable que es y una breve descripción.

Señales del módulo DDI 1602

PLC	DDI 1602	ENTRADAS DIGITALES				
Descripción	Grupo	Nombre	Tipo	Tarjeta	Canal	Señal
SENSOR DE BANDEJA PREVIA	Entrada digital	S1	ED	Local 1	0	0-24 V
SENSOR DE BANDEJA DE ELEVACIÓN	Entrada digital	S2	ED	Local 1	1	0-24 V
SENSOR ALTO RETENEDOR PREVIO	Entrada digital	S3	ED	Local 1	2	0-24 V
SENSOR BAJO RETENEDOR PREVIO	Entrada digital	S4	ED	Local 1	3	0-24 V
SENSOR ALTO RETENEDOR ELEVACIÓN	Entrada digital	S5	ED	Local 1	4	0-24 V
SENSOR BAJO RETENEDOR ELEVACIÓN	Entrada digital	S6	ED	Local 1	5	0-24 V
SENSOR SUPERIOR CILINDRO ELEVADOR	Entrada digital	S7	ED	Local 1	6	0-24 V
SENSOR INFERIOR CILINDRO ELEVACIÓN	Entrada digital	S8	ED	Local 1	7	0-24 V
SENSOR IZQUIERDO CILINDRO GIRO	Entrada digital	S9	ED	Local 1	8	0-24 V
SENSOR DERECHO CILINDRO GIRO	Entrada digital	S10	ED	Local 1	9	0-24 V
SENSOR DE BARRERA	Entrada digital	S11	ED	Local 1	10	0-24 V
SELECTOR AUTOMÁTICO/MANUAL	Entrada digital	P1	ED	Local 1	11	0-24 V
SELECTOR LOCAL/REMOTO	Entrada digital	P2	ED	Local 1	12	0-24 V
SETA DE EMERGENCIA	Entrada digital	P3	ED	Local 1	13	0-24 V
SEMIAUTOMÁTICO ALMACENAR BANDEJA	Entrada digital	P4	ED	Local 1	14	0-24 V
SEMIAUTOMÁTICO SERVIR BANDEJA	Entrada digital	P5	ED	Local 1	15	0-24 V

Tabla 2.2-1: Señales del módulo DDI 1602

Señales del módulo DDM 16022

PLC	DDM 16022		ENTRADAS/SALIDAS DIGITALES			
Descripción	Grupo	Nombre	Tipo	Tarjeta	Canal	Señal
ERROR DE SEGURIDAD	Entrada digital	FSEG	ED	Local 2	0	0-24 V
APRISIONAMIENTO DE MANO	Entrada digital	APRM	ED	Local 2	1	0-24 V
EXCESO DE BANDEJAS EN ALMACÉN	Entrada digital	EBA	ED	Local 2	2	0-24 V
MAMPARA DE ALMACÉN ABIERTA	Entrada digital	MAA	ED	Local 2	3	0-24 V
SETA DE EMERGENCIA MÁQUINA	Entrada digital	SEMERGM	ED	Local 2	4	0-24 V
PRESIÓN DE AIRE BAJA	Entrada digital	PABAJO	ED	Local 2	5	0-24 V
MARCHA CINTAS TRANSPORTADORAS	Entrada digital	MCT	ED	Local 2	6	0-24 V
LIBRE	Entrada digital		ED	Local 2	7	0-24 V
BAJAR CILINDRO ELEVADOR	Salida digital	V1	SD	Local 2	16	0-24 V
SUBIR CILINDRO ELEVADOR	Salida digital	V2	SD	Local 2	17	0-24 V
ROTACIÓN SENTIDO HORARIO CILINDRO GIRO	Salida digital	V3	SD	Local 2	18	0-24 V
ROTACIÓN SENTIDO ANTIHORARIO CILINDRO GIRO	Salida digital	V4	SD	Local 2	19	0-24 V
RETENEDOR BANDEJA PREVIA	Salida digital	V5	SD	Local 2	20	0-24 V
RETENEDOR BANDEJA ELEVACIÓN	Salida digital	V6	SD	Local 2	21	0-24 V
REARME DESDE EL AUTÓMATA	Salida digital	REARME	SD	Local 2	22	0-24 V
LIBRE	Salida digital		SD	Local 2	23	0-24 V

Tabla 2.2-2: Señales del módulo DDM 16022

Señales internas

Nombre	Tipo	Descripción	Dirección
A1	BOOL	INDICA EL ESTADO A1 DE LA GUÍA GEMMA	
A6	BOOL	INDICA EL ESTADO A6 DE LA GUÍA GEMMA	
AAB	BOOL	ANULAR PETICIÓN ALMACENARA BANDEJA	
AB	INT	INDICAR EL NÚMERO DE BANDEJAS INICIALES	
ASB	BOOL	ANULAR PETICIÓN DE SERVIR BANDEJA	
BI	BOOL	BANDEJA INTRODUCIDA	
BR	BOOL	BANDEJA RETIRADA	
CCE	BOOL	COMPROBAR CILINDRO ELEVADOR	
CCG	BOOL	COMPROBACIÓN DEL CILINDRO DE GIRO	
CFCE	BOOL	COMPROBAR FUNCIONAMIENTO DEL CILINDRO ELEVADOR	
CFCG	BOOL	COMPROBAR FUNCIONAMIENTO DEL CILINDRO DE GIRO	
CFRE	BOOL	COMPROBAR FUNCIONAMIENTO RETENEDOR ELEVACIÓN	
CFRP	BOOL	COMPROBAR FUNCIONAMIENTO DEL RETENEDOR PREVIO	
CICLO	INT	INDICA SI SE ESTÁ REALIZANDO UN CICLO (=1) O NO(=0)	
CONTBA	INT	CONTADOR DE BANDEJAS ALMACENADAS	
CONTBPA	INT	INDICA EL NÚMERO DE BANDEJAS PENDIENTES DE ALMACENAR	
CONTBPS	INT	INDICA EL NÚMERO DE BANDEJAS PENDIENTES DE SERVIR	
CRE	BOOL	COMPROBAR RETENEDOR ELEVACIÓN	
CRP	BOOL	COMPROBAR RETENEDOR PREVIO	
CSFC	BOOL	DESACTIVAR SECCIÓN SFC	
D1	BOOL	INDICA EL ESTADO D1 DE LA GUÍA GEMMA	
D3	BOOL	INDICA EL ESTADO D3 DE LA GUÍA GEMMA	
DCNBS	BOOL	DECREMENTA CONTADOR NÚMERO DE BANDEJAS DEL SISTEMA	
DCONTBA	BOOL	DECREMENTO DEL CONTADOR DE BANDEJAS ALMACENADAS	
DCONTBPS	BOOL	DECREMENTO DEL CONTADOR DE BANDEJAS PENDIENTES DE SERVIR	
F	BOOL	MENSAJE INTRODUZCA EL NÚMERO DE BANDEJAS	
F1	BOOL	INDICA EL ESTADO F1 DE LA GUÍA GEMMA	
F2	BOOL	INDICA EL ESTADO F2 DE LA GUÍA GEMMA	
F3	BOOL	INDICA EL ESTADO F3 DE LA GUÍA GEMMA	
F4	BOOL	INDICA EL ESTADO F4 DE LA GUÍA GEMMA	
F6	BOOL	INDICA EL ESTADO F6 DE LA GUÍA GEMMA	
FA6	BOOL	FIN DE LA PUESTA EN EL ESTADO INICIAL	
FCE	BOOL	FALLO EN EL CILINDRO ELEVADOR	

Tabla 2.2-3: Señales internas del programa Unity Pro

Nombre	Tipo	Descripción	Dirección
FCG	BOOL	FALLO EN EL CILINDRO DE GIRO	
FF2	BOOL	FIN DE LA MARCHA DE PREPARACIÓN	
FFCE	BOOL	FIN FALLO CILINDRO ELEVADOR	
FFCG	BOOL	FIN FALLO CILINDRO DE GIRO	
FFRE	BOOL	FIN FALLO RETENEDOR ELEVADOR	
FFRP	BOOL	FIN FALLO RETENEDOR PREVIO	
FRE	BOOL	FALLO EN EL RETENEDOR DE ELEVACIÓN	
FRP	BOOL	FALLO EN EL RETENEDOR PREVIO	
FSACE	BOOL	FALLO EN EL SENSOR ALTO DEL CILINDRO ELEVADOR	
FSBCE	BOOL	FALLO EN EL SENSOR BAJO DEL CILINDRO ELEVADOR	
FSDCG	BOOL	FALLO EN EL SENSOR DERECHO DEL CILINDRO DE GIRO	
FSICG	BOOL	FALLO EN EL SENSOR IZQUIERDO DEL CILINDRO DE GIRO	
FT	BOOL	FIN TEMPORIZADOR	
FTMCE	BOOL	FIN TEMPORIZADOR	
FTMCG	BOOL	FIN DEL TEMPORIZADOR DE MOVIMIENTO DEL CILINDRO DE GIRO	
FTRE	BOOL	FIN TEMPORIZADOR DEL RETENEDOR DE ELEVACIÓN	
FTRP	BOOL	FIN TEMPORIZADOR DEL RETENEDOR PREVIO	
FTSCE	BOOL	FIN TEMPORIZADOR SENSOR CILINDRO ELEVADOR	
FTSCG	BOOL	FIN TEMPORIZADOR SENSOR CILINDRO DE GIRO	
ICNBS	BOOL	INCREMENTA CONTADOR NÚMERO DE BANDEJAS DEL SISTEMA	
ICONTBA	BOOL	INCREMENTO DEL CONTADOR DE BANDEJAS ALMACENADAS	
ICONTBPA	BOOL	INCREMENTO DE CONTADOR DE BANDEJAS PENDIENTES DE ALMACENAR	
ICONTBPS	BOOL	INCREMENTO DE CONTADOR DE BANDEJAS PENDIENTES DE SERVIR	
IT	BOOL	INICIAR TEMPORIZADOR	
ITMCE	BOOL	INICIA TEMPORIZADOR DE MOVIMIENTO DEL CILINDRO ELEVADOR	
ITMCG	BOOL	INICIAR TEMPORIZADOR DE MOVIMIENTO DEL CILINDRO DE GIRO	
ITRE	BOOL	INICIAR TEMPORIZADOR DEL RETENEDOR DE ELEVACIÓN	
ITRP	BOOL	INICIAR TEMPORIZADOR DEL RETENEDOR PREVIO	
IRSB	BOOL	INICIAR TEMPORIZADOR DE SERVIR BANDEJA	
ITSCE	BOOL	INICIAR TEMPORIZADOR SENSOR CILINDRO ELEVADOR	
ITSCG	BOOL	INICIAR TEMPORIZADOR SENSOR DEL CILINDRO DE GIRO	
LDC	BOOL	CARGAR EL VALOR DEL CONTADOR BANDEJAS ALMACENADAS	
LDNBS	BOOL	CARGAR VALOR CONTADOR BANDEJAS SISTEMA	
MARCHA	BOOL	MARCHA / PARO	
MCE	BOOL	MOVIMIENTO DEL CILINDRO ELEVADOR	
MCG	BOOL	MOVIMIENTO CILINDRO DE GIRO	
MRE	BOOL	MOVIMIENTO DEL RETENEDOR DE ELEVACIÓN	
MRP	BOOL	MOVIMIENTO DEL RETENEDOR PREVIO	
NBS	INT	NÚMERO DE BANDEJAS EN EL SISTEMA	
RCONTBPA	BOOL	RESET DE CONTADOR DE BANDEJAS PENDIENTES DE ALMACENAR	
RCONTBPS	BOOL	RESET DE CONTADOR DE BANDEJAS PENDIENTES DE SERVIR	
SG	INT	SENTIDO DE GIRO	
SACE	INT	SENTIDO DE AVANCE DEL CILINDRO ELEVADOR	
TMMCE	TIME	TIEMPO MÁXIMO DE MOVIMIENTO DEL CILINDRO ELEVADOR	
TMMCG	TIME	TIEMPO MÁXIMO DE MOVIMIENTO DEL CILINDRO DE GIRO	
TMRE	TIME	TIEMPO MÁXIMO DEL RETENEDOR DE ELEVACIÓN	
TMRP	TIME	TIEMPO MÁXIMO DEL RETENEDOR PREVIO	
TMSCE	TIME	TIEMPO MÁXIMO SENSOR DEL CILINDRO ELEVADOR	
TMSCG	TIME	TIEMPO MÁXIMO SENSOR DEL CILINDRO DE GIRO	
TAB_PET_REM	ARRAY[1..4] OF INT	TABLA DE PETICIONES REMOTAS	%MW10
TAB_PET_REM[1]	INT	PETICIÓN SERVIR BANDEJA	%MW10
TAB_PET_REM[2]	INT	PETICIÓN ALMACENAR BANDEJA	%MW11
TAB_PET_REM[3]	INT	ANULAR PETICIÓN DE SERVIR BANDEJA	%MW12
TAB_PET_REM[4]	INT	ANULAR PETICIÓN DE ALMACENAR BANDEJA	%MW13
TAB_GEST_LECT	ARRAY[1..4] OF INT	TABLA DE GESTIÓN DE LECTURA	%MW40
TAB_ESC	ARRAY[1..8] OF INT	MENSAJES PARA EL AUTÓMATA PRINCIPAL	%MW200
TAB_ESC[1]	INT	SIRVIENDO BANDEJA	%MW200
TAB_ESC[2]	INT	ALMACENANDO BANDEJA	%MW201
TAB_ESC[3]	INT	NO SE PUEDE SERVIR BANDEJA	%MW202
TAB_ESC[4]	INT	NO SE PUEDE ALMACENAR BANDEJA	%MW203
TAB_ESC[5]	INT	PETICIÓN DE SERVIR BANDEJA RECIBIDA	%MW204
TAB_ESC[6]	INT	PETICIÓN DE ALMACENAR BANDEJA RECIBIDA	%MW205
TAB_ESC[7]	INT	ANULADA PETICIÓN DE SERVIR BANDEJA	%MW206
TAB_ESC[8]	INT	ANULADA PETICIÓN DE ALMACENAR BANDEJA	%MW207

Tabla 2.2-4: Señales internas del programa Unity Pro (continuación)

Nombre	Tipo	Descripción	Dirección
TAB_TMMCG	ARRAY[1..2] OF TIME	TABLA TIEMPO MÁXIMO DE MOVIMIENTO DEL CILINDRO DE GIRO	
TAB_TMMCG[1]	TIME	DESDE EL SENSOR S9	
TAB_TMMCG[2]	TIME	DESDE EL SENSOR S10	
TAB_TMSCE	ARRAY[1..11] OF TIME	TABLA TIEMPOS MÁXIMOS DE SENSORES DEL CILINDRO ELEVADOR	
TAB_TMSCE[1]	TIME	0 BANDEJAS	
TAB_TMSCE[2]	TIME	1 BANDEJA	
TAB_TMSCE[3]	TIME	2 BANDEJAS	
TAB_TMSCE[4]	TIME	3 BANDEJAS	
TAB_TMSCE[5]	TIME	4 BANDEJAS	
TAB_TMSCE[6]	TIME	5 BANDEJAS	
TAB_TMSCE[7]	TIME	6 BANDEJAS	
TAB_TMSCE[8]	TIME	7 BANDEJAS	
TAB_TMSCE[9]	TIME	8 BANDEJAS	
TAB_TMSCE[10]	TIME	9 BANDEJAS	
TAB_TMSCE[11]	TIME	10 BANDEJAS	

Tabla 2.2-5: Señales internas del programa Unity Pro (continuación 2)

2.2.5. PE

La sección PE (Parada de emergencia) se asocia al estado D1. Es el de mayor prioridad de todos. Cuando este estado se activa se debe detener el proceso que se esté realizando y el sistema debe evolucionar hasta la posición de seguridad. Ésta posición, que se alcanza sin necesidad de dar ninguna instrucción, ya que está cableada, tendrá los cilindros retenedores en la posición baja y el cilindro elevador en la baja, además, si el cilindro de giro está en movimiento, éste se detendrá inmediatamente. Si se pulsa una seta de emergencia o se activa cualquier elemento de seguridad que tiene la máquina se realizará una parada de emergencia para evitar dañar la instalación o a un operario.

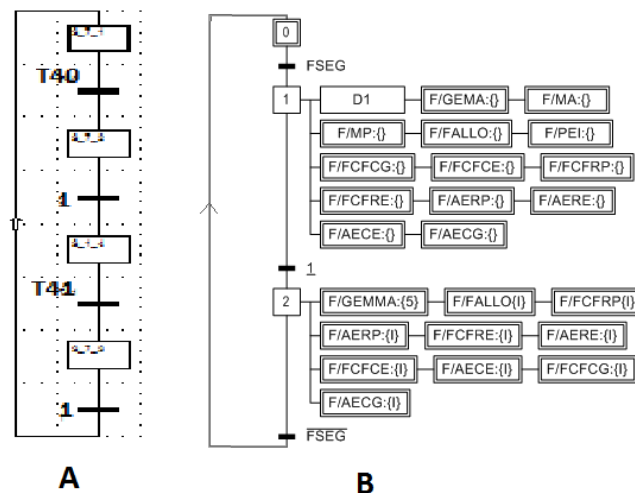


Figura 2.2-5: SECCIÓN SFC (A) Y GRAFCET PE (B)

Cuando se inicia la ejecución del programa, se activa el estado inicial de este GRAFCET y permanece en esta etapa hasta que se produce una incidencia de seguridad. Esta incidencia se detecta a través de un contactor normalmente cerrado de un relé, si alguno de los elementos de seguridad se activa, se abre el contactor del relé

y se detecta el fallo de seguridad cuando la entrada correspondiente del PLC está activa, en ese momento se franquea la transición T40.

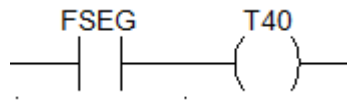


Figura 2.2-6: Transición T40 de la sección PE

Luego se actualizan las variables que indican en qué estado de la Guía GEMMA está la máquina, para que en la pantalla se vea que pasa al estado D1, se desactivan todos los GRAFCETs que pudieran dar una instrucción de movimiento de algún actuador y se pasa a la posición de seguridad.

La desactivación de los GRAFCETS se hace a través del bloque de función CLEARCHART, hay que usar una para cada uno, haciendo un total de 13 en esta etapa. Estos GRAFCETS son: GEMMA, MA, MP, PEI, FALLO, FCFCG, FCFCE, FCFRP, FCFRE, AERP, AERE, AECE y AECG, que se explicarán en apartados posteriores.

El bloque de función CLEARCHART tiene dos entradas y una salida. En la primera entrada se pone el nombre del GRAFCET del que se quiere desactivar todas las etapas, en la segunda entrada necesita una señal de tipo BOOL que es la que activa el bloque cuando esta entrada está activa, en nuestro caso se ha conectado directamente porque cuando se activa esta acción nos interesa que ocurra, por lo tanto no es necesario ninguna señal que controle la ejecución de estos bloques. La salida es otro booleano que se pone a uno cuando el GRAFCET indicado no tiene ninguna etapa activa, esta salida no es obligatoria que tenga conexión. En la figura 2.2-7 se puede ver un ejemplo del uso de esta función en lenguaje LD.

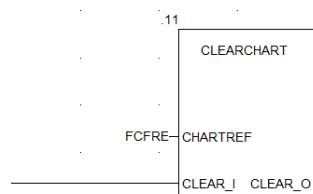


Figura 2.2-7: Bloque de función CLEARCHART

Cuando finaliza la instancia de seguridad y se rearma el relé, a través del pulsador rearme (P6) o del botón con el mismo nombre incluido en la pantalla de explotación, se franquea la transición T41 y se pasa a la siguiente etapa.

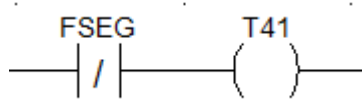


Figura 2.2-8: Transición T41 de la sección PE

En esta nueva etapa se vuelven a activar los GRAFCETs que se desactivaron en la etapa anterior. Para esto se usan dos bloques de funciones INITCHART y SETSTEP. Son necesarias un total de 9 bloques INITCHART, para las secciones FALLO, FCFCG, FCFCE, FCFRE, FCFRP, AERP, AERE, AECE y AECG, y un bloque SETSTEP para la sección GEMMA.

El bloque de función INITCHART activa la etapa inicial del GRAFCET indicado. Las entradas y salidas de este bloque de función son equivalentes al del bloque de función CLEARCHART. En la primera entrada se indica el nombre del GRAFCET al que se va a aplicar la función, la segunda es un booleano que ejecuta la función cuando está activo y la salida es otro booleano que se activa cuando la función ha terminado de ejecutarse y ha activado las etapas iniciales del GRAFCET. En la figura 2.2-9 se muestra una representación en lenguaje LD de este bloque de función.

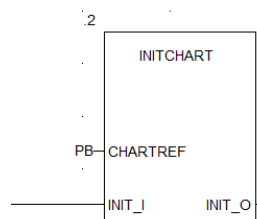


Figura 2.2-9: Bloque de función INITCHART

El bloque de función SETSTEP, se usa para activar la etapa de la sección GEMMA correspondiente al inicio del procedimiento del estado A6 (procedimiento de puesta en el estado inicial). Esta función activa la etapa indicada, pero en las demás etapas que estén activas no tiene ningún efecto, por esa razón, para usar este bloque de función es necesario que esté activa la opción multi-token.

La función SETSTEP tiene una entrada, pero es necesario que la alimentación esté conectada. Cuando la alimentación esté activa, la sección que tiene el mismo nombre que está puesto en la segunda entrada se activa. En la figura 2.2-10 se ve un ejemplo de esta función en lenguaje LD:

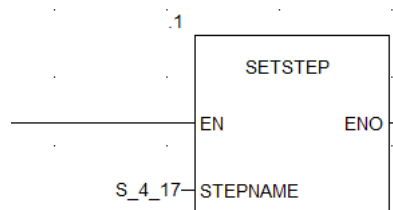


Figura 2.2-10: Bloque de función SETSTEP

Cuando se realizan las acciones de la última etapa de este GRAFCET, se pasa inmediatamente al estado inicial donde se mantiene a la espera para actuar si se produce una nueva incidencia de seguridad.

2.2.6. GEMMA

La sección GEMMA, como su nombre indica, es la encargada de controlar la evolución de los estados de la guía GEMMA. Por tanto, es la segunda con mayor jerarquía, sólo por detrás de la sección que se encarga de las emergencias.

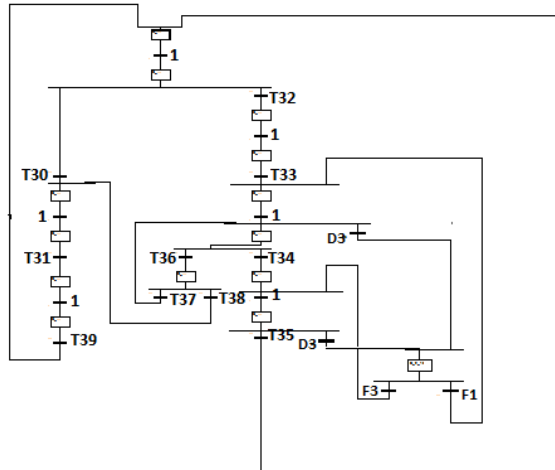


Figura 2.2-11: SECCIÓN GEMMA

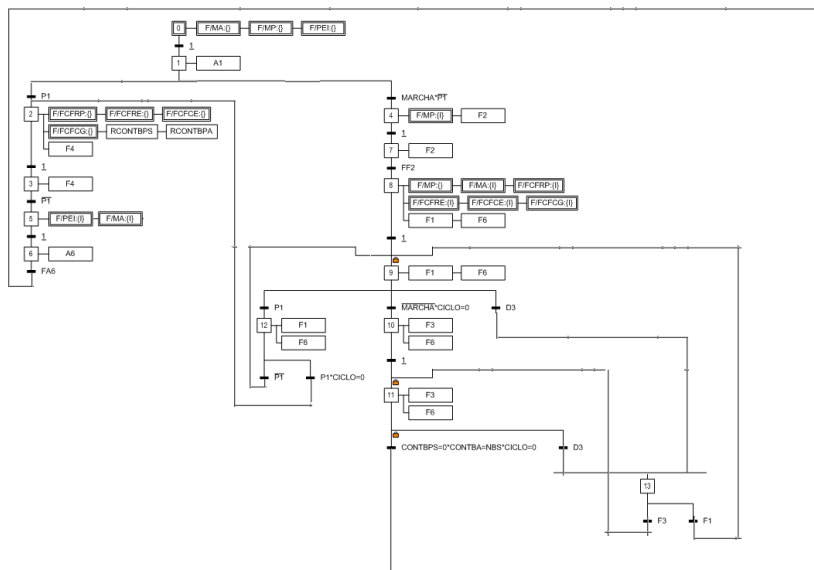


Figura 2.2-12: GRAFCET GEMMA

Cuando se inicia esta sección, en su etapa inicial desactiva las secciones MA, MP y PEI con la función CLEARCHART, estas secciones son las encargadas del funcionamiento de la máquina en los distintos estados de la Guía GEMMA y serán activadas y desactivadas en función de las etapas de este GRAFCET.

Luego nos encontramos con dos secuencias posibles. La de la izquierda que comienza con la transición T30 se franquea si el selector P1 está en modo manual.

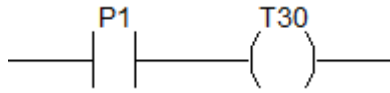


Figura 2.2-13: Transición T30 de la sección GEMMA

Como veremos posteriormente, a la siguiente etapa se puede acceder desde el modo de funcionamiento normal (F1). Se activa la variable del estado F4, se inician los contadores de bandejas pendientes de servir y de almacenar a 0.

Una vez terminado las acciones de la etapa que se expone en el párrafo anterior, se pasa a la espera de que el selector P1 se ponga en la posición automático para que finalice el estado F4. Entonces, se activa la señal del estado A6 y se desactiva la de F4, también se desactiva la sección MA con la función CLEARCHART y se activa la etapa inicial de la sección PEI con la función INICHART.

Luego, cuando la máquina ha llegado a la posición de reposo inicial se finaliza este estado y se vuelve al estado inicial del GRAFCET.

La secuencia de la derecha es la del funcionamiento automático. Esta secuencia se comienza cuando el selector P1 está en posición automática y se ha pulsado el botón de marcha que se encuentra en uno de los paneles de la pantalla de explotación.

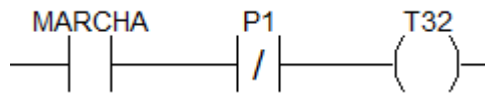


Figura 2.2-14: Transición T32 de la sección GEMMA

Antes de comenzar con el funcionamiento normal (F1) se realiza una marcha de preparación, por tanto, se activa la variable del estado F2 y se desactiva la de A1 y mediante la función INITCHART se activa la sección MP, y se pasa a la espera de que se finalice el proceso de preparación de marcha para pasar al estado de funcionamiento normal (F1).

Una vez que esto ocurre, se activan las variables de los estados F1 y F6, y se desactiva la de F2, se desactiva la sección MP mediante la función CLEARCHART y se activan las secciones MA, FCFRE, FCFRP, FCFCE, FCFCG con la función INICHART. Y se pasa a la espera de que finalice el modo de funcionamiento normal, ya sea porque el selector P1 está en posición manual o porque se ha pulsado el botón de parada de la pantalla de explotación.

Si se ha puesto el selector P1 en la posición manual se espera a que termine el ciclo que se esté realizando antes de pasar al funcionamiento manual, pero si antes de

que finalice el ciclo se vuelve a poner el selector en posición automática, la máquina permanece en el modo de funcionamiento normal (F1).

Si por el contrario se ha pulsado el botón de parada, se pasa al modo de marcha de cierre (F3), en la que se activa la variable que indica que se está en el estado F3 y se desactiva la variable que indica el estado F1. Luego se mantiene a la espera de que finalice la marcha de cierre, esto ocurre cuando todas las bandejas están almacenadas y se ha terminado el ciclo que se estaba ejecutando, esta transición se representa en la figura 2.2-14.

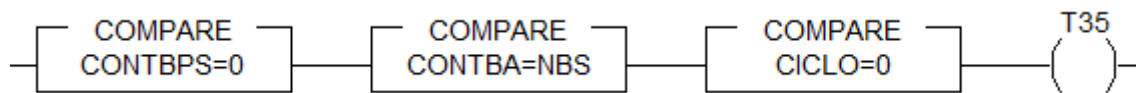


Figura 2.2-15: Transición T35 de la sección GEMMA

Cuando finaliza la marcha de cierre, se vuelve al estado inicial del GRAFCET que se corresponde al estado A1.

Si cuando se está en el estado F1 (Producción normal) o en el estado F3 (Marcha de cierre) se detecta algún fallo de un sensor o actuador, estará activa la variable D3 y se pasará a un estado auxiliar para que se desactiven la variable F1 o F3, dependiendo del estado en el que se encontrara el sistema en el momento que se detecta el fallo.

Cuando este fallo se resuelve, se regresa al estado de funcionamiento normal si la variable marcha sigue activa para que se active la variable F1. Pero si cuando se elimina el fallo la variable marcha está desactiva se regresa a la etapa del estado de marcha de cierre para activar la variable F3.

2.2.7. MA

La sección MA (Modo Automático) es la asociada al estado de funcionamiento automático, si no hay defecto, o al estado de producción a pesar de defectos, cuando se trata de un defecto de los que consideramos leve. Esto se ha hecho de esta manera porque la única diferencia en el control de ambos estados se encuentra en la receptividad de las transiciones a flanquear, por tanto, como se detallará a continuación habrá dos formas de franquear las transiciones.

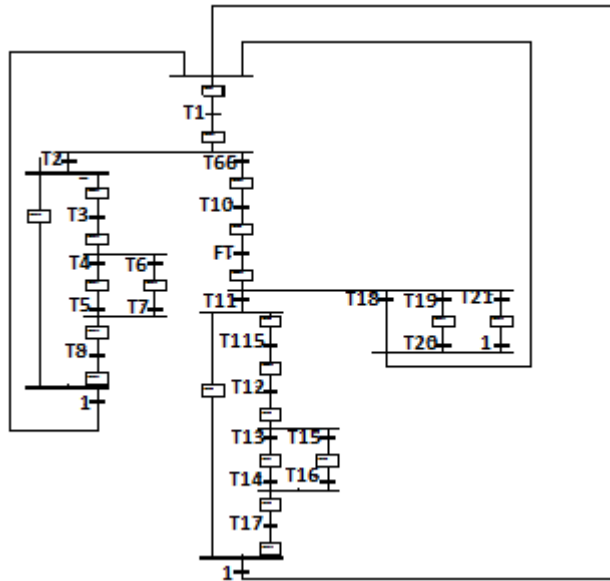


Figura 2.2-16: GRAFCET MA

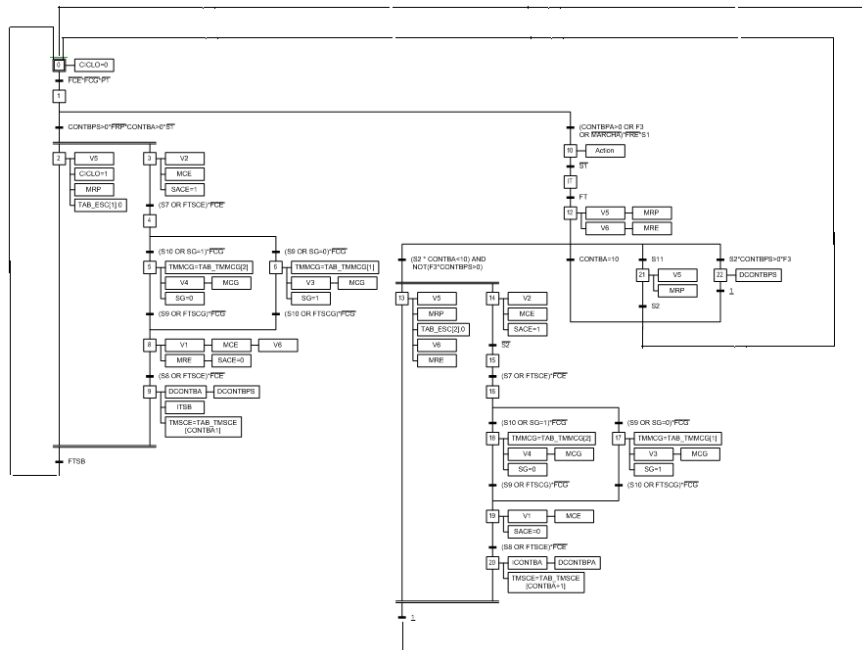


Figura 2.2-17: GRAFCET MA

Antes de comenzar a explicar cada uno de estas posibles secuencias, voy a exponer cada transición, ya que las receptividades de algunas de ellas coinciden.

La transición T1 tiene como condición que el selector P1 está en posición automática y que no haya fallo ni en el cilindro elevador ni en el de giro. Esta transición tiene el siguiente esquema LD.

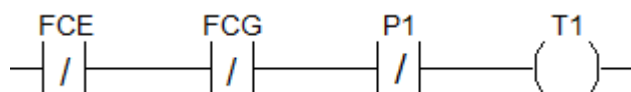


Figura 2.2-18: Transición T1 de la sección MA

La transición T2 tiene como condición que haya bandejas pendientes de servir y no haya ninguna bandeja en la posición del retenedor previo (S1), además no puede haber fallo en el retenedor previo y debe haber alguna bandeja almacenada. Esta transición tiene el siguiente esquema LD.

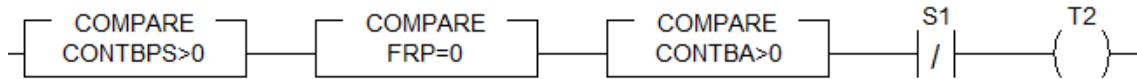


Figura 2.2-19: Transición T2 de la sección MA

Las transiciones T3 y T12 tienen como condición que el pistón elevador llegue a la posición alta. Esto ocurre cuando se activa el sensor superior del cilindro elevador (S7) o, si este sensor está defectuoso, cuando finalice el temporizador que dura el tiempo máximo que tarda el cilindro en subir (FTMCE). Estas transiciones no se pueden franquear si hay un error en el cilindro elevador y tienen el siguiente esquema LD.

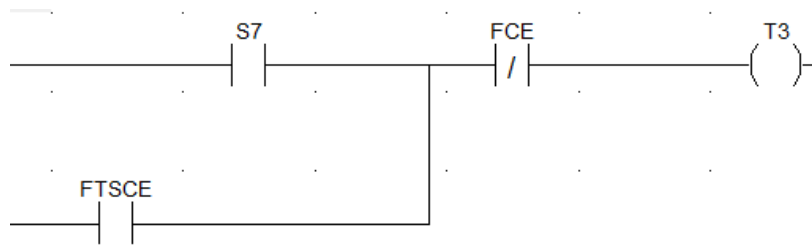


Figura 2.2-20: Transiciones T3 y T12 de la sección

Las transiciones T4, y T13 tienen como condición que el cilindro del almacén esté en la posición derecha. Esto ocurre cuando se activa el sensor derecho del cilindro de giro (S10) o, si este sensor está defectuoso, que la variable del sentido de giro sea 1. Estas transiciones no se pueden franquear si hay un error en el cilindro de giro y tienen el siguiente esquema LD.

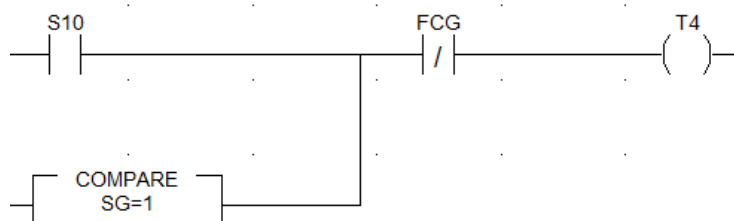


Figura 2.2-21: Transiciones T4 y T13 de la sección MA

Las transiciones T7, y T16 tienen como condición que el cilindro del almacén esté en la posición derecha. Esto ocurre cuando se activa el sensor derecho del cilindro de giro (S10) o, si este sensor está defectuoso, que finalice el contador del tiempo máximo que tarda en activarse el sensor opuesto del cilindro de giro. Estas transiciones no se pueden franquear si hay un error en el cilindro de giro y tienen el siguiente esquema LD.

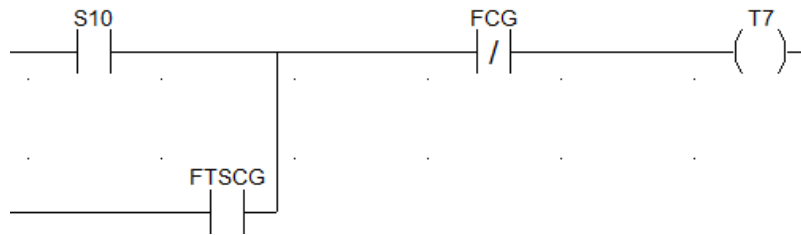


Figura 2.2-20: Transiciones T7 y T16 de la sección MA

Las transiciones T6 y T15 tienen como condición que el cilindro del almacén esté en la posición izquierda. Esto ocurre cuando se activa el sensor izquierdo del cilindro de giro (S9) o, si este sensor está defectuoso, que la variable del sentido de giro sea 0. Estas transiciones no se pueden franquear si hay un error en el cilindro de giro y tienen el siguiente esquema LD.

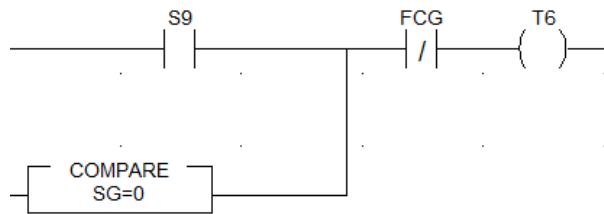


Figura 2.2-21: Transiciones T6 y T15 de la sección MA

Las transiciones T5 y T14 tienen como condición que el cilindro del almacén esté en la posición izquierda. Esto ocurre cuando se activa el sensor izquierdo del cilindro de giro (S9) o, si este sensor está defectuoso, cuando finalice el temporizador que dura el tiempo máximo que tarda el cilindro en girar (FTGCA). Estas transiciones no se pueden franquear si hay un error en el cilindro de giro y tienen el siguiente esquema LD.

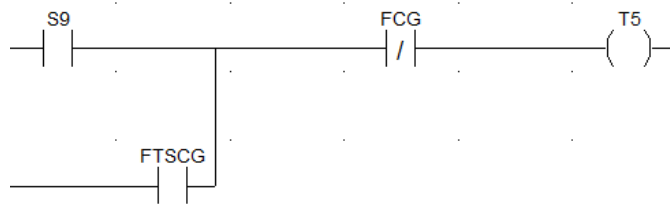


Figura 2.2-22: Transiciones T5 y T14 de la sección MA

Las transiciones T8 y T17 tienen como condición que el pistón elevador llegue a la posición baja. Esto ocurre cuando se activa el sensor inferior del cilindro elevador (S8) o, si este sensor está defectuoso, cuando finalice el temporizador que dura el tiempo máximo que tarda el cilindro en bajar (FTMCE). Estas transiciones no se pueden franquear si hay un error en el cilindro elevador y tienen el siguiente esquema LD.

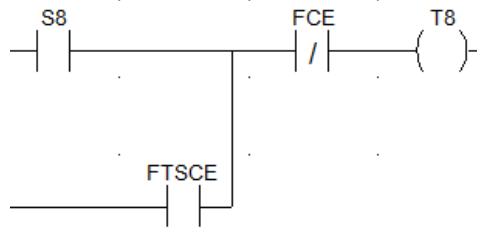


Figura 2.2-23: Transiciones T8 y T17 de la sección MA

La transición T66 tiene como condición que haya alguna petición pendiente de almacenar una bandeja o que este en el estado de marcha de cierre (F3=1) o la variable MARCHA esté desactiva. Además tiene que haber una bandeja en la posición del retenedor previo (S1) y que no haya fallo en el retenedor de elevación (FRE=0).

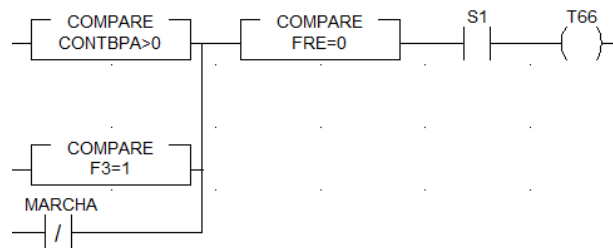


Figura 22-24: Transición T66 de la sección MA

La transición T11 tiene como condición que la bandeja llegue a la posición donde está el almacén (S2) y que haya menos de 10 bandejas almacenadas. Además no puede haber bandejas pendientes de servir y el sistema no puede estar en el estado de marcha de cierre. Esta transición tiene el siguiente esquema LD.

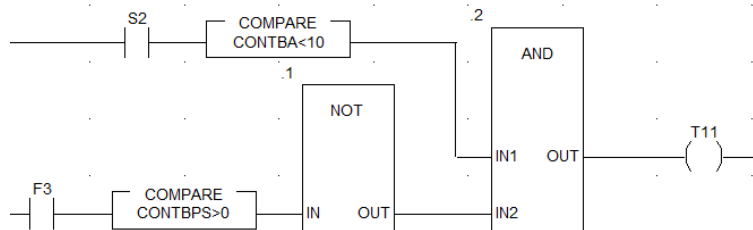


Figura 2.2-25: Transición T11 de la sección MA

La transición T18 tiene como condición que haya 10 bandejas almacenadas, es decir, que el almacén esté completo. Esta transición tiene el siguiente esquema LD.

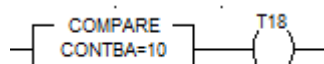


Figura 2.2-26: Transición T18 de la sección MA

La transición T19 tiene como condición que se active el sensor de barrera (S11), es decir, que la bandeja porte un pallet. Esta transición tiene el siguiente esquema LD.



Figura 2.2-27: Transición T19 de la sección MA

La transición T20 tiene como condición que se active el sensor (S2), posición en la que está el almacén. Esta transición tiene el siguiente esquema LD.

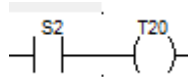


Figura 2.2-28: Transición T20 de la sección MA

La transición T21 tiene como condición que la bandeja llegue a la posición donde está el almacén (S2), que haya alguna bandeja pendiente de servir y que esté en procedimiento de parada. Esta transición tiene el siguiente esquema LD.

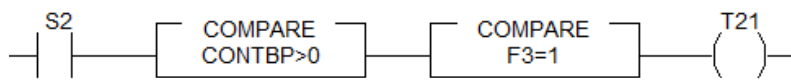


Figura 2.2-29: Transición T21 de la sección MA

La transición T115 tiene como condición que no haya bandejas en la posición del elevador. Esta transición tiene el siguiente esquema LD.



Figura 2.2-30: Transición T115 de la sección MA

Este GRAFCET comienza inicializando la variable CICLO a 0 y si el selector P1 está en posición automática y no hay fallos de los cilindros, se franquea la transición T1. A continuación llegamos a una transición, el camino de la derecha es la secuencia de acciones que se deben realizar para servir una bandeja, mientras que el de la izquierda es el suceso de acciones para almacenarla.

Comenzaremos por la secuencia de servir una bandeja. Si hay alguna bandeja pendiente de servir y no hay bandejas en el retenedor previo ni fallo del retenedor previo, se mantiene activado el retenedor previo y se actualiza la variable CICLO a 1. Paralelamente, se sube el cilindro elevador hasta la posición alta y a continuación se gira el almacén a la posición opuesta a la que esté en ese momento.

Una vez que se termina de girar el almacén, se baja el cilindro elevador y se activa el retenedor elevador, esta situación se mantiene hasta que el pistón elevador llega a la posición baja.

Entonces se desactiva el retenedor de elevación y la bandeja comienza el recorrido. También se reduce uno en el contador de bandejas pendientes de servir y el de bandejas almacenadas, y se actualiza el tiempo máximo de activación de los sensores del cilindro elevador en función del número de bandejas desactiva el retenedor previo e inicia un temporizador de 3 segundos para que le dé tiempo a la bandeja de salir de la zona del almacén, luego vuelve a la etapa inicial donde se pone la variable ciclo a 0 y está listo para realizar una nueva operación.

En el caso en el que este marcado la etapa inicial y se haya solicitado que se recoja la próxima bandeja, cuando llegue una bandeja al retenedor previo comienza el proceso de recogida de la bandeja. Lo primero que se hace es poner la variable ciclo a 1 y se espera que la bandeja pase el retenedor previo para activarlo, para esto es necesario incluir un pequeño retardo para que no se golpee la bandeja que pasa con el retenedor, y detener cualquier posible bandeja que llegara durante el tiempo que se almacena esta. Entonces se comprueba si se puede recoger la bandeja o no.

Si el contador de bandejas almacenadas tiene un valor de 10, esto significa que está completo y por tanto no se puede almacenar. En este caso se deja pasar la bandeja. Luego vuelve a la etapa inicial donde se actualiza la variable ciclo a 0 para comenzar un nuevo proceso.

Si el sensor de barrera se activa significa que la bandeja porta un pallet y por tanto no se puede recoger y se espera que la bandeja llegue a la posición del retenedor elevador para bajar el retenedor previo y permitir que las demás bandejas sigan avanzando. Luego vuelve a la etapa inicial donde se actualiza la variable ciclo a 0 para comenzar un nuevo proceso.

Si la bandeja llega a la posición del retenedor de elevación sin que se active el sensor de barrera significa que la bandeja está vacía. Entonces tenemos dos opciones, recogerla o dejarla pasar. Si hay alguna bandeja pendiente de servir y la máquina está en el estado de marcha de cierre, entonces se deja pasar la bandeja y se reduce uno en el contador correspondiente. Luego vuelve a la etapa inicial donde se actualiza la variable ciclo a 0 para comenzar un nuevo proceso.

En el caso de que cuando llegue la bandeja a la altura del retenedor de elevación, el almacén no esté completo y no haya peticiones pendientes, la bandeja es recogida. En este caso se eleva el retenedor previo para que ninguna bandeja que pueda llegar interfiera en el proceso. Paralelamente, se sube el retenedor de elevación para detener la bandeja en la posición adecuada y se comienza a subir el cilindro elevador hasta la posición alta y a continuación se gira el almacén a la posición opuesta a la que esté en ese momento.

Una vez que se termina de girar el almacén, se baja el cilindro elevador hasta que llega a la posición baja.

También se incrementa uno en el contador de bandejas almacenadas y se descuenta uno del contador de bandejas pendientes de almacenar, y se desactiva el retenedor previo, luego vuelve a la etapa inicial donde se pone la variable ciclo a 0 y está listo para realizar una nueva operación.

2.2.8. MP

La sección MP (Marcha de Preparación) está asociada al estado Marcha de preparación (F2). Esta sección sólo se usa para indicarle al sistema el número de bandejas que hay al comenzar la producción automática, ya que se carece de otra forma de conseguir esa información, y comprobar el tiempo que tarda en cada etapa, ya que las condiciones de la instalación pueden variar si se modifica la presión del aire, para inicializar las variables de tiempo con las que se comprobará el funcionamiento de los elementos.

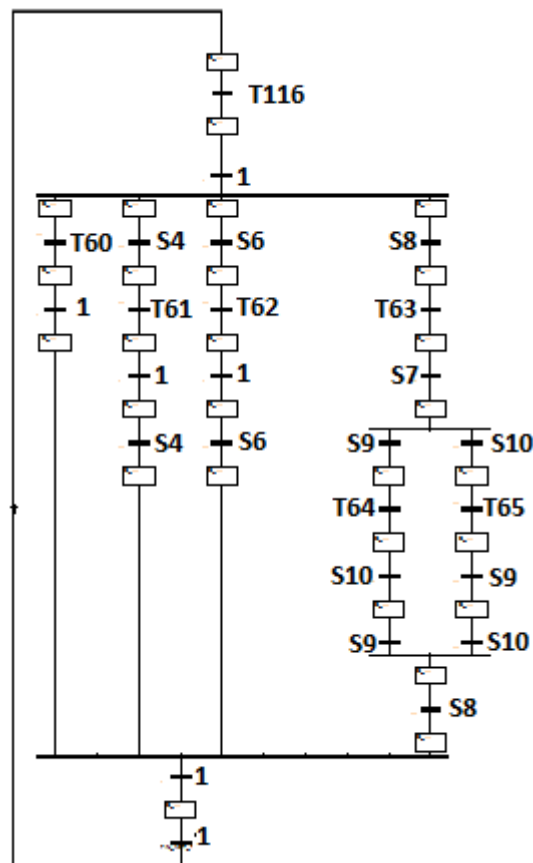


Figura 2.2-31: SECCIÓN MP



Figura 2.2-32: GRAFCET MP

Para conocer si el tiempo que tarda en producirse cada movimiento es constante se realizaron una serie de medidas en un ciclo de servir bandeja, modificando el número de bandejas que había en el almacén, y se obtuvieron los resultados que se muestran en la tabla 2.2-6.

Nº BAND.	TMMCE	TMMCG	TMSCE	TMSCG	TMRP	TMRE
1	350MS	860MS	1S_500MS	500MS	80MS	80MS
1	350MS	880MS	1S_510MS	480MS	80MS	70MS
1	350MS	870MS	1S_500MS	490MS	70MS	70MS
1	340MS	870MS	1S_510MS	490MS	80MS	70MS
2	300MS	1S_220MS	1S_510MS	500MS	80MS	70MS
2	350MS	1S_240MS	1S_550MS	510MS	80MS	80MS
2	350MS	1S_250MS	1S_550MS	510MS	80MS	80MS
2	350MS	1S_250MS	1S_550MS	510MS	80MS	80MS
3	330MS	850MS	1S_570MS	590MS	80MS	70MS
3	350MS	880MS	1S_590MS	590MS	80MS	80MS
3	350MS	870MS	1S_590MS	590MS	80MS	80MS
3	350MS	890MS	1S_590MS	580MS	80MS	70MS
4	360MS	1S_320MS	1S_600MS	550MS	80MS	80MS
4	350MS	1S_320MS	1S_640MS	540MS	80MS	80MS
4	350MS	1S_310MS	1S_640MS	550MS	80MS	70MS
4	350MS	1S_320MS	1S_640MS	540MS	80MS	80MS
5	320MS	860MS	1S_670MS	630MS	80MS	80MS
5	350MS	870MS	1S_700MS	640MS	80MS	80MS
5	360MS	890MS	1S_700MS	620MS	90MS	80MS
5	350MS	910MS	1S_710MS	600MS	80MS	80MS
6	360MS	1S_380MS	1S_800MS	550MS	90MS	90MS
6	350MS	1S_380MS	1S_780MS	560MS	80MS	80MS
6	340MS	1S_370MS	1S_780MS	550MS	80MS	70MS
6	350MS	1S_380MS	1S_780MS	540MS	80MS	80MS
7	340MS	840MS	1S_850MS	680MS	80MS	80MS
7	350MS	840MS	1S_870MS	690MS	80MS	80MS
7	360MS	840MS	1S_870MS	680MS	80MS	100MS

7	350MS	840MS	1S_870MS	680MS	80MS	70MS
8	360MS	1S_310MS	1S_960MS	570MS	100MS	70MS
8	340MS	1S_330MS	1S_990MS	570MS	70MS	80MS
8	350MS	1S_340MS	1S_990MS	570MS	80MS	80MS
8	350MS	1S_340MS	1S_980MS	570MS	90MS	90MS
9	320MS	880MS	2S_90MS	710MS	80MS	80MS
9	360MS	870MS	2S_90MS	710MS	90MS	80MS
9	350MS	850MS	2S_90MS	690MS	80MS	90MS
9	350MS	920MS	2S_100MS	680MS	90MS	80MS

Tabla 2.2-6: Tiempos de los actuadores

De la tabla anterior se puede ver que el tiempo de movimiento del cilindro elevador (TMMCE), el tiempo de los sensores del cilindro de giro (TMSCG), y los tiempos de movimiento de los retenedores previo y de elevación (TMRP Y TMRE, respectivamente) son aproximadamente constantes para cualquier número de bandejas.

También se observa que independientemente del número de bandejas, el tiempo que tarda en comenzar a rotar el almacén (TMMCE), depende del sentido del giro. Por tanto se ha creado una tabla en la que se guarda el tiempo de giro de cada sentido, y durante la ejecución del programa se modifica el valor del contador leyendo el valor de esta tabla.

Además el tiempo que tarda en terminar de subir el cilindro elevador (TMSCE) va aumentando mientras más bandejas almacenadas haya. Por esta razón, se definió una tabla en la que se guarda el valor de tiempo en el que debe terminar de subir el cilindro en función del número de bandejas almacenadas.

Para realizar la puesta en marcha de la máquina, se inicializa a -1 las variables del número de bandejas del sistema y los contadores de bandejas pendientes de servir y de almacenar se inicializan a cero. Luego se ejecutan cuatro secuencias en paralelo.

En la primera por la izquierda, se activa un mensaje en la pantalla táctil para que el operario introduzca un número entre 0 y 10, ese será el número de bandejas del sistema, y se espera hasta que se dicho número cumpla esta condición. Luego se carga ese valor en el contador de bandejas almacenadas y en el contador de bandejas totales del sistema, y se espera a que terminen las demás secuencias.

La segunda y tercera secuencias son análogas, una para el retenedor previo y otra para el retenedor de elevación. Se comienza subiendo el retenedor, y cuando se desactiva el sensor bajo del retenedor se franquea la transición. En la siguiente etapa se guarda el tiempo que ha estado activa la etapa en la que se subía el retenedor, hay una variable de tiempo para cada una.

Para obtener dicho valor de tiempo se recurre a la estructura SFCSTEP_STATE que contiene todos los datos relativos al estado de la etapa o de la macroetapa. En concreto la variable de esta estructura que guarda el tiempo que lleva la etapa activa, o en caso de estar desactiva, el tiempo que ha estado activa. Para acceder a este valor

es necesario escribir el nombre de la etapa que se desea conocer seguido de .t, en la figura 2.2-34 se ve un ejemplo en el que se guarda el valor en una variable TIME usando el lenguaje ST.

```
TMRP:=S_5_18.t;
```

Figura 2.2-33: Obtención de duración de activación de una etapa

Luego se espera a que se vuelva a activar el sensor bajo del retenedor y se franquea la última transición donde se mantiene hasta que finalicen todas las demás secuencias que se están ejecutando.

En la última secuencia se comprueba el que tarda en realizarse las operaciones del cilindro elevador y del cilindro de giro.

Se comienza subiendo el cilindro elevador hasta que se desactiva el sensor bajo, cuando esto ocurre se franquea la transición y se guarda el tiempo que ha tardado, que posteriormente se usará para saber si el cilindro elevador funciona correctamente, y se sigue subiendo el cilindro hasta que se activa el sensor alto, momento en el cual se pasa la transición y se guarda ese tiempo en otra variable con la que se comprobará si se ha producido algún fallo en los sensores del cilindro elevador.

Luego se realiza un procedimiento equivalente para el cilindro de giro, en el que se girará hacia el lado en el que no está el cilindro, y se obtendrá tanto el tiempo que tarda en detectarse que se ha movido el cilindro como el tiempo que tarda en activarse el sensor opuesto. Finalmente se vuelve a girar el cilindro hasta la posición inicial para evitar liberar una bandeja. Por último se baja el cilindro elevador hasta que se activa el sensor bajo.

Una vez que han finalizado las cuatro secuencias se añade un margen de tiempo a los valores que se han obtenido, ya que si no se hace esto se detectan falsos errores, y de esta forma se corrige ese problema. Esto se hace mediante una acción en lenguaje ST, como se ve en la figura 2.2-40.

```
TMMCE:=TMMCE+t#300MS;  
TMRP:=TMRP+t#30MS;  
TMRE:=TMRE+t#30MS;  
TMSCG:=TMSCG+t#400MS;  
TMSCE:=TAB_TMSCE[CONTBA+1];
```

Figura 2.2-34: Incrementar variables TIME

Finalmente, se indica que se ha finalizado de ejecutar este estado, para que se pase al siguiente estado de la Guía GEMMA.

Los valores de las variables TMMCE, TMRE, TMRP Y TMSCG se obtienen en cada inicio de la ejecución, por lo que si cambian las condiciones de presión de la instalación, estos valores cambiarán.

Sin embargo, los valores que hay almacenados en las tablas para las variables TMSCE Y TMMCG, al igual que los mostrados en la tabla 2.2-6, están obtenidos y ajustados para una presión de 4 bares, y si la presión del sistema es diferente no se garantiza el correcto funcionamiento de los módulos de detección y comprobación de fallos.

2.2.9. PEI

La sección PEI (Puesta en el Estado inicial) se asocia al estado A6 de la Guía GEMMA. Con estas acciones se lleva la máquina hasta la posición de parada inicial. Siguiendo las mismas estructuras que en las secciones expuestas en apartados anteriores, esta sección funcionará tanto en situaciones normales como en situaciones de fallo leve.

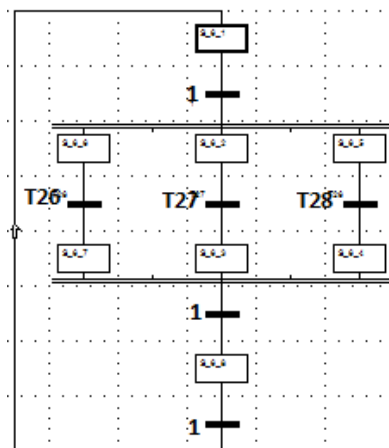


Figura 2.2-35: SECCIÓN PEI

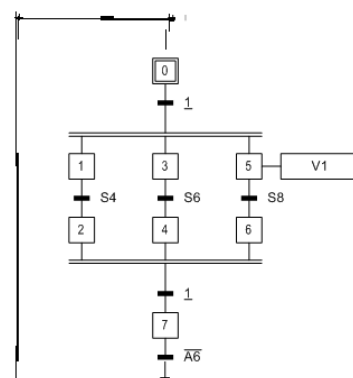


Figura 2.2-36: GRAFCET PEI

Se realizan tres secuencias simultáneamente. Dos iguales para cada retenedor, se desactiva los retenedores y se espera que lleguen a la posición baja, luego se espera a que terminen los demás procesos, y una tercera secuencia en la que se baja el cilindro elevador y se espera a que llegue hasta la posición baja. Una vez que terminan las tres secuencias, se indica que el sistema está en la posición inicial.

2.2.10. PB

La sección PB se encarga de gestionar las peticiones que se realizan en el modo de funcionamiento automático, tanto en modo local como remoto. En el caso de que el selector P2 esté en posición local, las peticiones tanto de servir como de almacenar bandeja se realizarán a través de los pulsadores P4 y P5 y la anulación de estas peticiones se realizarán usando la pantalla de explotación. Mientras que en el caso en el que el selector P2 esté en posición remota, estas peticiones se leerán a través de la comunicación de Ethernet que se ha creado.

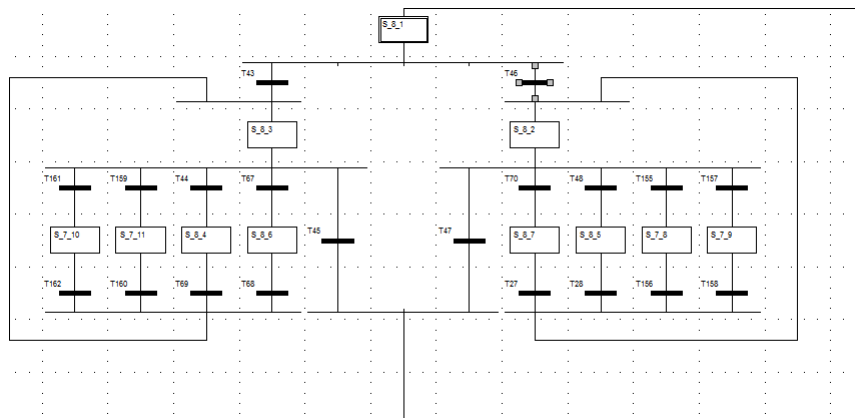


Figura 2.2-37: SECCIÓN PB

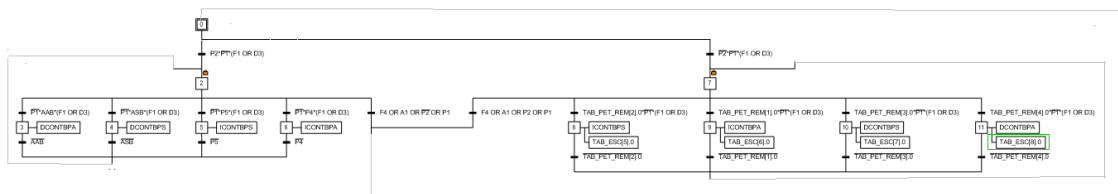


Figura 2.2-38: GRAFCET PB

Ambas secuencias del GRAFCET son equivalentes, una para el funcionamiento en modo local y otra para el funcionamiento en modo remoto.

Si se ha activado el interruptor de MARCHA de la pantalla de explotación y el selector P1 está en modo automático, estando la máquina en modo de funcionamiento

automático (F1) o en modo de funcionamiento a pesar de defectos (D3), se franquea una de las dos primeras transiciones, la que se franquee dependerá de la posición del selector P2.

Si el selector P2 está en posición local se franqueará la transición T43. Pero si el selector está en posición remota se franqueará la transición T46. Ambas transiciones son iguales, a excepción de la variable P2, en el caso de la transición T43 esta variable será un contacto normalmente abierto, mientras que en la transición T46 será un contacto normalmente cerrado. En la figura 2.2-43 se puede ver el diagrama LD de la transición T43.

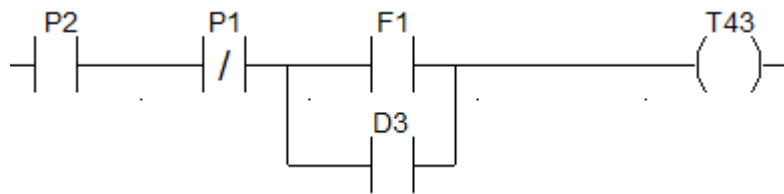


Figura 2.2-39: Transición T43 de la sección PB

En el modo de control local se usan los pulsadores del panel de control para para realizar las peticiones de alto nivel. Se usará el pulsador P5 para servir una bandeja y el pulsador P4 para almacenar la próxima bandeja que llegue al almacén. Y para anular las peticiones se usan los botones de la pantalla de explotación “Anular petición de servir bandeja” y “Anular petición de almacenar bandeja”, asignados a las variables del autómeta ASB y AAB, respectivamente.

Al igual que ocurre en las transiciones T43 y T46, las transiciones T44, T67, T159 y T161 son equivalentes, sustituyéndose en el caso de la transición T44 la variable P4 por la P5, en el caso de la transición T159 se sustituye P4 por ASB y en el caso de la transición T161 se sustituye por AAB. En la figura 2.2-44 se representa el diagrama LD de la transición T67.

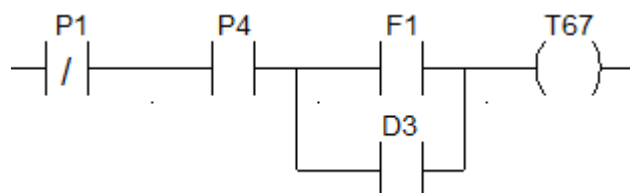


Figura 2.2-40: Transición T67 de la sección PB

Mientras que en caso de que el selector P2 esté en posición remota, las peticiones serán leídas del autómeta principal a través de una red Ethernet.

Para hacer la petición de almacenar una bandeja se usa el primer bit de la variable TAB_PET_REM[1] y para la petición de servir una bandeja se usa el primer bit de la variable TAB_PET_REM[2].

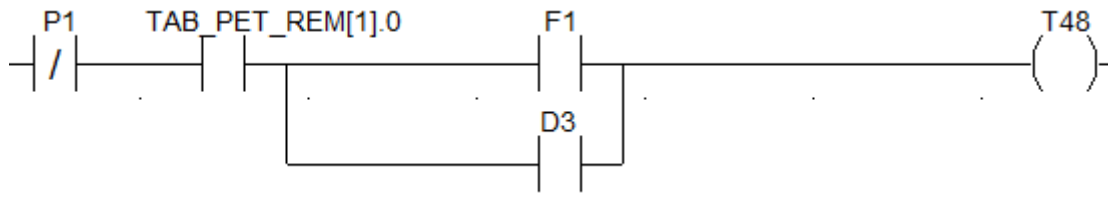


Figura 2.2-41: Transición T48 de la sección PB

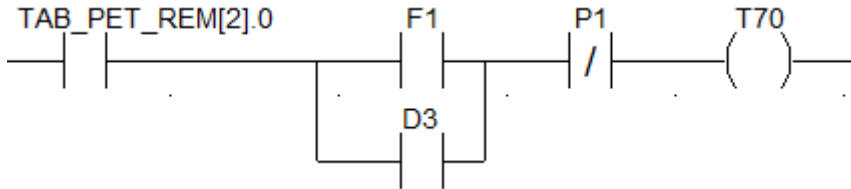


Figura 2.2-42: Transición T70 de la sección PB

Mientras que para anular la petición de servir una bandeja se usa el primer bit de la variable TAB_PET_REM[4] y para anular la petición de almacenar una bandeja se usa el primer bit de la variable TAB_PET_REM[3].

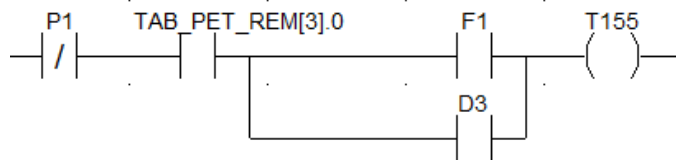


Figura 2.2-43: Transición T155 de la sección PB

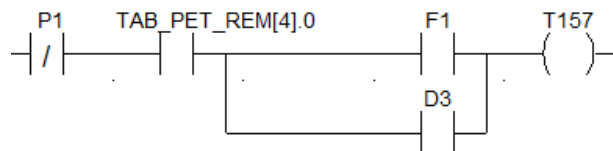


Figura 2.2-44: Transición T157 de la sección PB

Cuando se recibe una petición de servir bandeja se incrementa el contador de bandejas pendientes de servir, mientras que si se recibe la señal de almacenar la próxima bandeja, se incrementa el contador de bandejas pendientes de almacenar. Si por el contrario se recibe una petición de anular petición de servir una bandeja se descuenta uno del contador de bandejas pendientes de servir, mientras que si se recibe la petición de anular petición de almacenar la próxima bandeja se descuenta uno del contador de bandejas pendientes de almacenar. En todos los casos se espera a que se finalice la señal de petición para esperar a recibir una nueva petición.

En el caso de la comunicación remota, se le indica al autómata principal que se ha recibido la petición para qué deje de indicar la acción correspondiente, en el apartado 2.2.19 se puede ver como se emplea la función READ_VAR para realizar estas operaciones.

Si se cambia la posición del selector P2 o se solicita la parada de la máquina se pasa a la etapa inicial del GRAFCET.

2.2.11. NBANDJ

Esta sección lleva la cuenta del número de bandejas total del sistema. Si se presiona el botón de BANDEJA RETIRADA de la pantalla de explotación, se descuenta uno al contador, mientras que si se presiona el botón BANDEJA INTRODUCIDA se añadirá uno al contador.

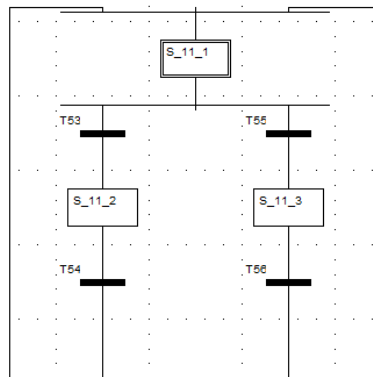


Figura 2.2-45: SECCIÓN NBANDEJ

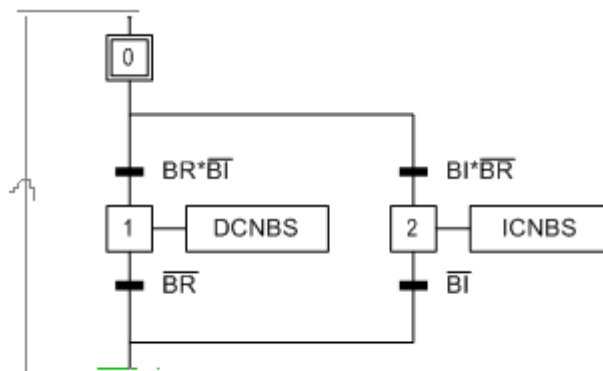


Figura 2.2-46: GRAFCET NBANDEJ

2.2.12. FALLO

Esta sección se encarga de que en la pantalla de explotación se vea la Guía GEMMA el estado D3 si se produce algún fallo.

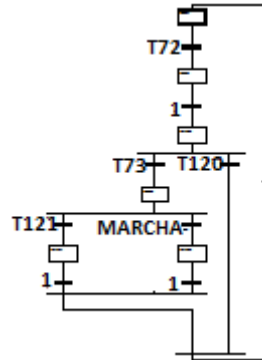


Figura 2.2-47: GRAFCET FALLO

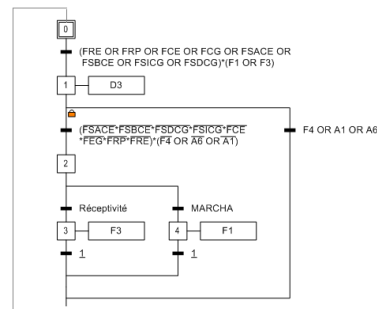


Figura 2.2-48: GRAFCET FALLO

Para ello la primera transición se franquea si se detecta algún fallo y la máquina está en alguno de los estados desde los que se puede pasar a esta situación, esta transición tiene el siguiente esquema LD.

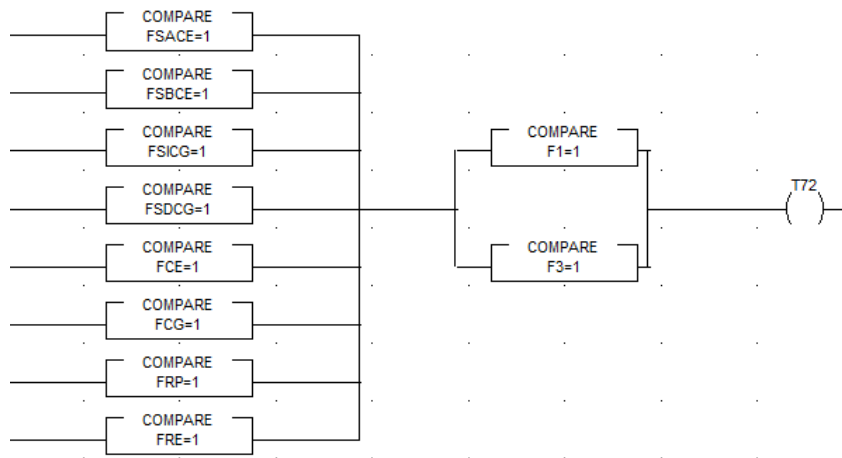
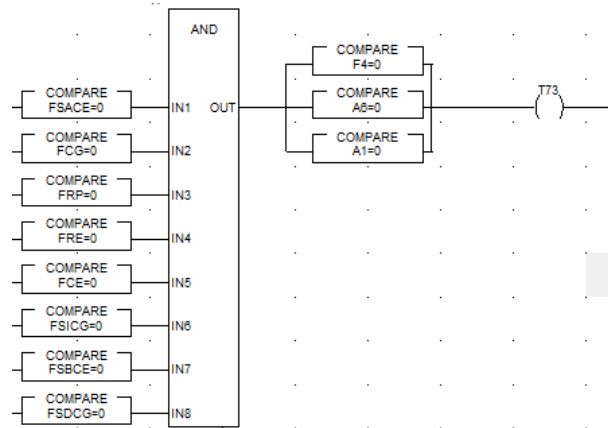


Figura 2.2-49: Transición T72 de la sección FALLO

Si se franquea esta transición se pone la variable del estado D3 a 1 y las variables de los estados F1 y F3 a 0. El GRAFCET se mantendrá en esta etapa hasta que se franquee la siguiente transición. Esto ocurrirá cuando no haya ningún fallo de sensores o actuadores y que no se haya pasado a los estados F4, A6 o A1 y tiene el esquema que se muestra en la figura 2.2-41.



Figuro 2.2-50: Transición T73 de la sección FALLO

Mientras que si se ha pasado a alguno de los estados F4, A6 o A1, como se vio en la sección GEMMA, cuando se va a estos estados se desactiva la variable D3, y por tanto no es necesario hacer nada en esta sección.

Cuando se finaliza el fallo de todos los sensores y actuadores, y se sigue en alguno de los dos estados que pueden funcionar con defecto (F1 y F3). Si la variable MARCHA está activa, se activa la variable del estado F1, y si no lo está se activa la variable del estado F3, en ambos caso se desactiva la variable D3, y se vuelve al inicio para una futura ejecución de la secuencia.

2.2.13. FCFRP Y FCFRE

Las secciones FCFRP y FCFRE se encargan de comprobar el correcto funcionamiento de los retenedores previo y de elevación, respectivamente. Ambas secciones son idénticas, a excepción de las variables que activan o deben activarse para franquear las transiciones.

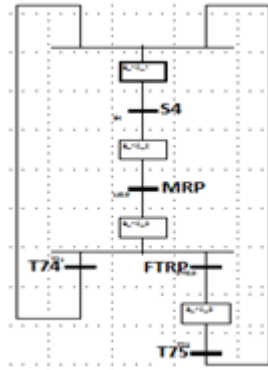


Figura 2.2-51: SECCIONES FCFRP y FCFRE

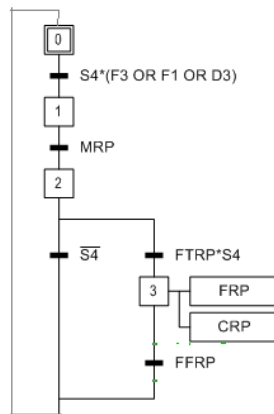


Figura 2.2-52: GRAFCETs FCFRP Y FCFRE

Debido a que estos GRAFCET sólo comprueban el correcto funcionamiento en los estados F1 y F3, cuando se está trabajando en alguno de esos estados y el retenedor está en la posición baja, el sensor se activa y pasa la primera transición.

Luego se espera a que se active la señal que indica que se ha dado la orden al retenedor de que suba. En ese momento se inicia el temporizador del retenedor, cuyo valor máximo se obtuvo en la marcha de preparación (F2). Si se desactiva el sensor antes de que el temporizador finalice, el actuador funciona correctamente. Si por el contrario el temporizador finaliza la cuenta sin que se desactive el sensor, el actuador no funcionará correctamente y se detectará el fallo.

Cuando se detecta el fallo del retenedor, se activa la variable que indica el fallo del retenedor en cuestión. Luego se franquea la transición y se pasa a la siguiente etapa.

En esta última etapa se activa una variable que habilita la función ITCNTRL asociada al evento de temporización que comprueba si se ha reparado el actuador. A partir de este momento cada vez que el temporizador del evento llegue al final de la cuenta se ejecutará dicho evento. Cuando se activa el evento de tiempo se activa una variable con la que se comienza la ejecución de la sección que comprueba si el

retenedor ha vuelto a funcionar. En la figura 2.2-52 podemos ver un ejemplo del uso de la función ITCNTRL.

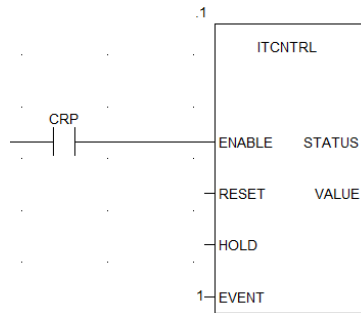


Figura 2.2-53: Ejemplo función ITCNTRL

Si se ha corregido el fallo, la variable que indica el fallo del retenedor estará desactiva y se franqueará la última transición, volviendo a la etapa inicial a la espera de realizar futuras comprobaciones.

2.2.14. AERP Y AERE

Las secciones AERP y AERE se encargan de comprobar que el retenedor ha vuelto a funcionar correctamente después de que se haya detectado un fallo en ese actuador. Ambas secciones son idénticas, una para cada retenedor, a excepción de las variables que activan o deben activarse para franquear las transiciones.

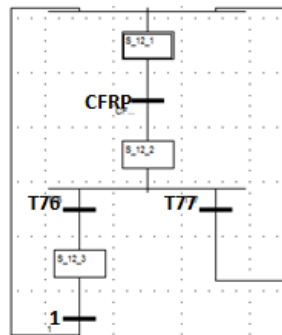


Figura 2.2-53: SECCIONES AERP y AERE

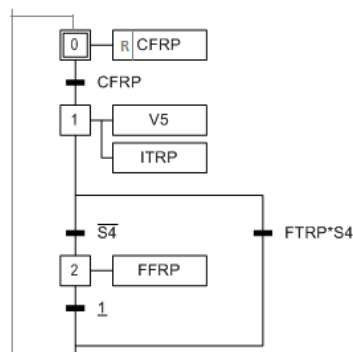


Figura 2.2-54: GRAFCETS AERP y AERE

Cada vez que se produce un evento de tiempo del retenedor se activa la variable que hace que la primera transición sea franqueable. La sección del evento de tiempo debe ejecutarse lo más rápido posible, por tanto, como se ve en la figura 2.2-54, lo único que hace el evento de tiempo es activar una variable y las acciones que se deben llevar a cabo tras el evento lo hace este GRAFCET.

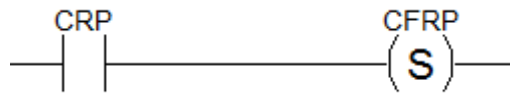


Figura 2.2-55: Sección del evento temporizador

Cuando se activa el evento y se franquea la primera transición se activa la salida del PLC que hace que suba el retenedor y se activa el temporizador de movimiento del retenedor.

Si se recibe la señal de fin de temporizador antes de que se desactive el sensor, significa que no se ha reparado el fallo, y se vuelve a la etapa inicial a la espera de que se produzca un nuevo evento de tiempo.

Si por el contrario, se desactiva el sensor bajo del retenedor antes de que finalice el temporizador, el actuador habrá sido reparado y se desactivará la variable que indica si el funcionamiento del retenedor es correcto.

2.2.15. FCFCE Y FCFCG

Las secciones FCFCE y FCFCG se encargan de comprobar el correcto funcionamiento de los cilindros de elevación y de giro, respectivamente. Ambas secciones son idénticas, a excepción de las variables que activan o deben activarse para franquear las transiciones.

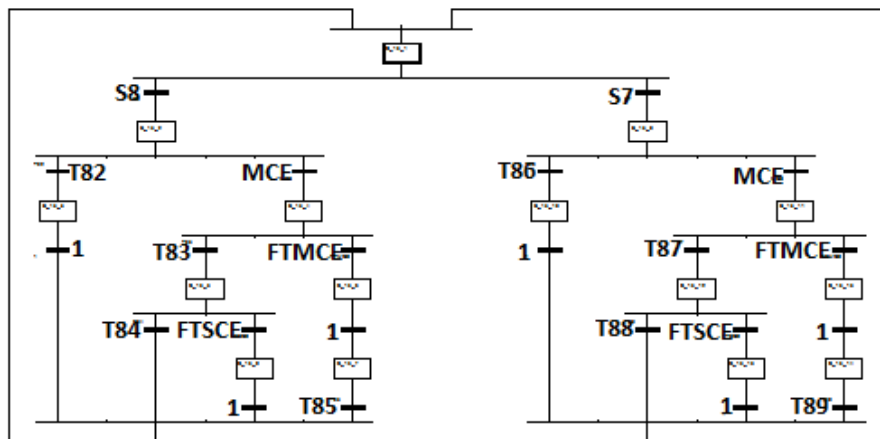


Figura 2.2-56: SECCIONES FCFCE y FCFCG

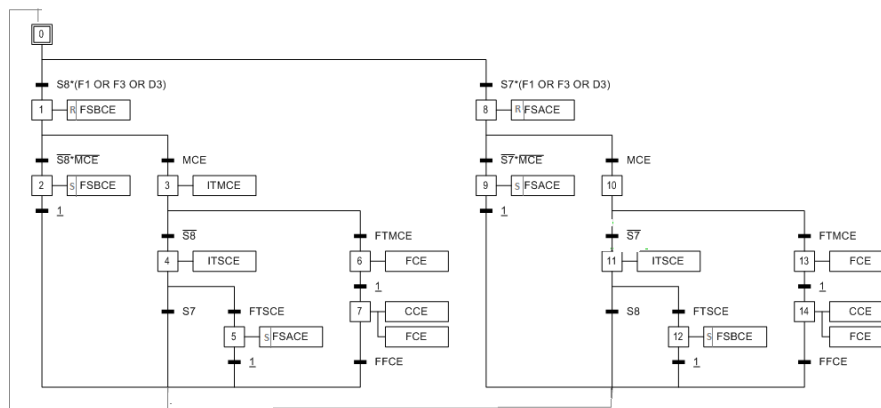


Figura 2.2-57: GRAFCETs FCFCE y FCFCG

En cada una de estas secciones se distinguen dos secuencias, una para cada posición del cilindro. En el caso del cilindro elevador, una se realizará cuando está activo el sensor S7 y otra cuando lo está el sensor S8, y en el caso del cilindro de giro una secuencia se realiza cuando está activo el sensor S9 y otra cuando lo está el sensor S10. Por tanto, explicaremos una de estas secuencias y las otras tres serán análogas.

Cuando se activa un sensor, se pasa la primera transición y se desactiva la variable que indica el fallo de ese sensor, y se mantiene en esa etapa hasta que ocurre una de estas dos cosas

Si se desactiva el sensor y no se ha activado la señal de movimiento del actuador, se detectará como fallo de dicho sensor, y por tanto, se activará la variable que indica el fallo de ese sensor, y luego se vuelve a la etapa inicial.

Lo otro que puede pasar es que se active la señal que indica el movimiento del cilindro. Cuando se activa dicha señal, se inicia el temporizador de movimiento del cilindro, cuyo valor máximo se obtuvo en el estado de marcha de preparación.

Si se desactiva el sensor antes de que finalice este temporizador significa que el actuador funciona correctamente, por lo que se desactivará la variable que indica fallo del cilindro. También activa el sensor que mide el tiempo que debe tardar en activarse el sensor de la otra posición del actuador. Si se activa el sensor antes de que termine el temporizador se vuelve a la etapa inicial. Pero si se no se ha activado el sensor opuesto cuando finalice el temporizador, se activa la variable que indica el fallo en ese sensor, y luego se vuelve a la etapa inicial.

Si cuando se indica el movimiento del cilindro, finaliza el temporizador antes que se desactive la el sensor que estaba activado, se detectará como fallo del actuador.

Cuando se detecta el fallo del actuador, se activa la variable que indica el fallo del cilindro en cuestión. Luego se franquea la transición y se pasa a la siguiente etapa.

En esta última etapa se activa una variable que habilita la función ITCNTRL asociada al evento de temporización que comprueba si se ha reparado el actuador. A partir de este momento cada vez que el temporizador del evento llegue al final de la cuenta se ejecutará dicho evento. Cuando se activa el evento de tiempo se activa una variable con la que se comienza la ejecución de la sección que comprueba si el retenedor ha vuelto a funcionar, al igual que se hace en el caso de las secciones FCFRP y FCFRE.

Si se ha corregido el fallo, la variable que indica el fallo del CILINDRO estará desactiva y se franqueará la última transición, volviendo a la etapa inicial a la espera de realizar futuras comprobaciones.

2.2.16. AECE

La sección AECE se encarga de comprobar que el cilindro elevador ha vuelto a funcionar correctamente después de que se haya detectado un fallo en ese actuador.

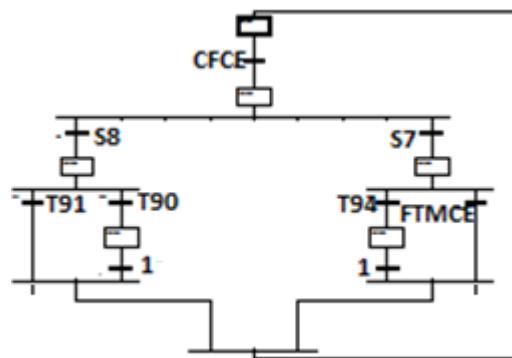


Figura 2.2-58: SECCIÓN AECE

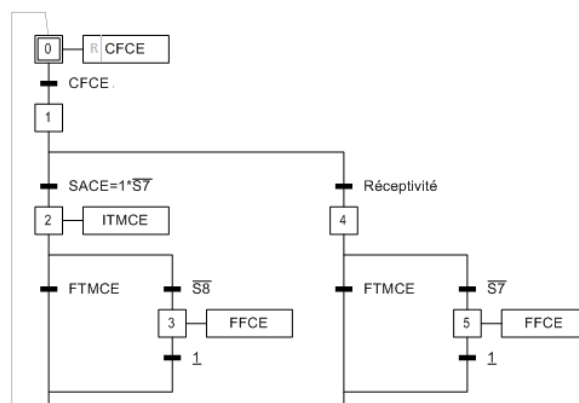


Figura 2.2-59: GRAFCET AECE

Cada vez que se produce un evento de tiempo del cilindro elevador se ejecuta la sección de ese evento. Como se indica en el apartado 2.2.15, la sección del evento de tiempo debe ejecutarse lo más rápido posible, por tanto, lo único que hace el evento de tiempo es activar la variable que permite comenzar las acciones de comprobación

del correcto funcionamiento del cilindro elevador, es decir, hace que la primera transición de este GRAFCET sea validada.

Cuando se comienza la comprobación del funcionamiento del cilindro elevador puede haber DOS situaciones, que el actuador haya dejado de funcionar en la posición baja o en la posición alta, no puede estar entre ambos ya que se trata de un actuador de dos posiciones.

En el caso de que estuviera en la posición baja, estaría el sensor S8 activo. Entonces se intentaría subir el cilindro para saber si ha vuelto a funcionar. Simultáneamente, se inicializa el temporizador del tiempo máximo que tarda en detectarse que el cilindro se ha movido. Si se desactiva el sensor antes de que finalice el temporizador, se desactiva la variable de fallo del cilindro elevador y se baja el cilindro hasta que se vuelve a activar el sensor bajo (S8). Si por el contrario se recibe la señal de fin del temporizador, se asume que el actuador sigue averiado. En ambos casos se vuelve a la etapa inicial a la espera de una nueva comprobación.

En el caso de que estuviera en la posición alta, estaría el sensor S7 activo. Entonces se intentaría bajar el cilindro para saber si ha vuelto a funcionar. Simultáneamente, se inicializa el temporizador del tiempo máximo que tarda en detectarse que el cilindro se ha movido. Si se desactiva el sensor antes de que finalice el temporizador, se desactiva la variable de fallo del cilindro elevador y se continúa bajando el cilindro hasta que se activa el sensor bajo (S8). Si por el contrario se recibe la señal de fin del temporizador, se asume que el actuador sigue averiado. En ambos casos se vuelve a la etapa inicial a la espera de una nueva comprobación.

2.2.17. AECG

La sección AECG se encarga de comprobar que el cilindro de giro del almacén ha vuelto a funcionar correctamente después de que se haya detectado un fallo en ese actuador.

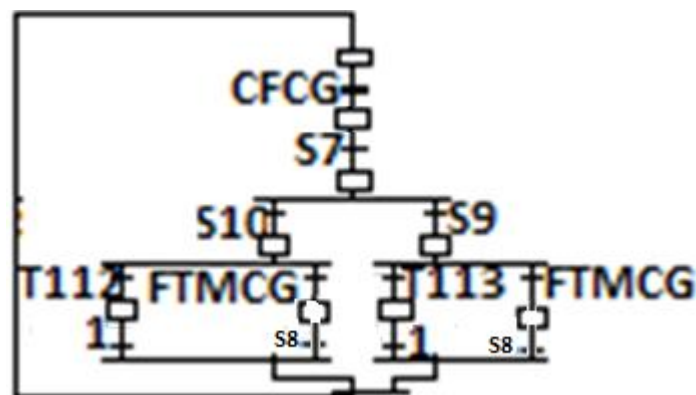


Figura 2.2-60: SECCIÓN AECG

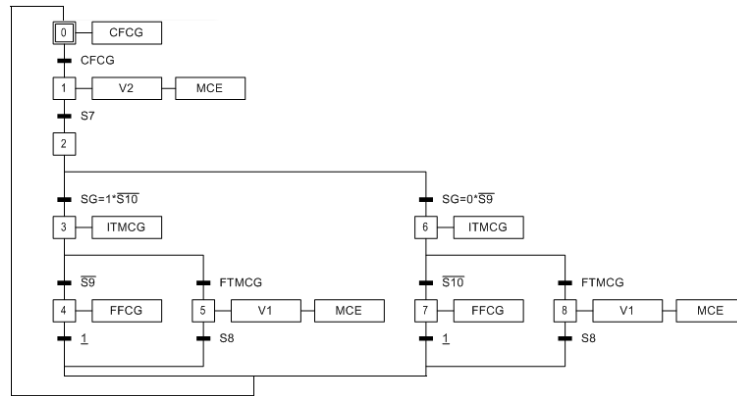


Figura 2.2-61: GRAFCET AECG

Cada vez que se produce un evento de tiempo del cilindro que gira el almacén se ejecuta la sección de ese evento. Como se indica en el apartado 2.2.14, la sección del evento de tiempo debe ejecutarse lo más rápido posible, por tanto, lo único que hace el evento de tiempo es activar la variable que permite comenzar las acciones de comprobación del correcto funcionamiento del cilindro elevador, es decir, hace que la primera transición de este GRAFCET sea validada.

El cilindro de giro está ajustado para que no pueda girar sino están levantadas las bandejas, es decir, para que se produzca el giro del almacén primero hay que subir el cilindro elevador. Por lo tanto subimos el cilindro elevador hasta que se ha activado el sensor alto (S7).

Al igual que pasa en el caso de fallo del cilindro elevador, el cilindro de giro puede fallar en dos posiciones, cuando está en alguna de las dos posiciones extremas .

Cuando el cilindro está en una posición intermedia se franquea la transición que da paso a la secuencia de la izquierda del GRAFCET. En esta secuencia se comprueba a través de una variable, que se modifica en la sección MA, en qué sentido estaba girando el almacén.

Si el fallo del cilindro que gira el almacén se produce en una de las posiciones extremas, estará activo el sensor izquierdo o el sensor derecho. Dependiendo de qué sensor esté activo se activará la señal para que el almacén gire hasta la otra posición y se iniciará el temporizador que indica el tiempo máximo en el que se detecta el movimiento del actuador.

Si finaliza el temporizador sin que se haya desactivado el sensor que esté activo, significa que no se ha corregido el defecto. Y se pasa a la etapa inicial hasta que se produzca un nuevo evento de tiempo para comprobar el funcionamiento de este actuador.

Si por el contrario se desactiva el sensor antes de que finalice el temporizador, se activa la señal para que el almacén gire en sentido contrario y vuelva a la posición en la que se encontraba antes de comenzar la comprobación, además se desactiva la variable que indica el funcionamiento del cilindro de giro. Luego se vuelve a la etapa inicial a la espera de un nuevo evento de tiempo asociado a este actuador.

2.2.18. ITP

Esta sección se usa, en caso de que se hubiera detectado un fallo de alguno de los actuadores de la máquina, para indicarle al autómatas principal en qué condiciones se encuentra el alimentador de bandejas, es decir, si no se puede almacenar, no se puede servir o ninguna de las dos opciones está en ese momento operativo. Con esta información el autómatas principal podrá actuar en consecuencia y adaptar las operaciones que se realizarán a la condición en la que se encuentre la planta.

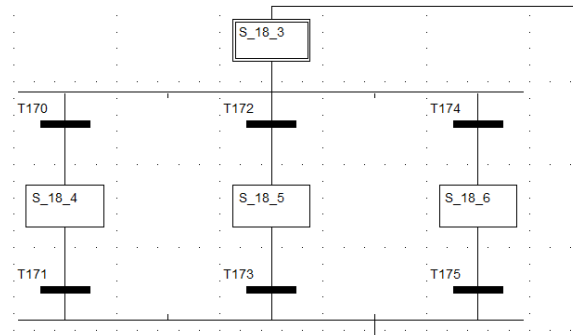


Figura 2.2-62: SECCIÓN ITP

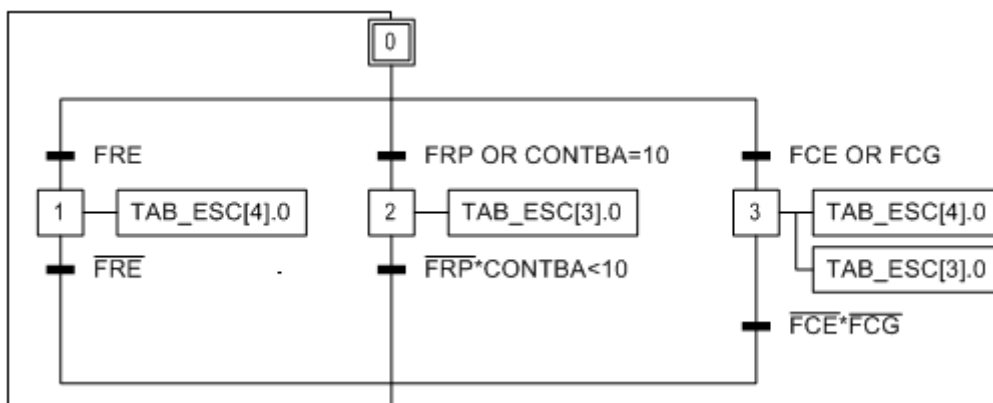


Figura 2.2-63: GRAFCET ITP

Si falla el retenedor de elevación no se puede almacenar ninguna bandeja ya que esta no se podría detener en la ubicación adecuada para ser recogida, por tanto, se activa el bit menos significativo de la variable $TAB_ESC[4]$ que está ubicada en una dirección de memoria que debe leer el autómatas principal, esto se explicará con más detalle en el apartado siguiente.

Si falla el retenedor previo no se puede servir ninguna bandeja ya que esta podría quedar montada sobre otra bandeja al no poder detenerse, por tanto, se activa el bit menos significativo de la variable TAB_ESC[5] que está ubicada en una dirección de memoria que debe leer el autómata principal.

Si el fallo es de alguno de los dos cilindros que mueven el almacén, el de elevación o el de giro, no se puede servir ni almacenar ninguna bandeja, por tanto, se activa el bit menos significativo de la variable TAB_ESC[5] y el bit menos significativo de la variable TAB_ESC[4].

2.2.19. COM_ETH

Con esta sección, que se ha realizado en lenguaje LD, se leen las variables que indican una petición de servir bandeja o una petición de almacenar la próxima bandeja, así como las anulaciones de dichas peticiones.

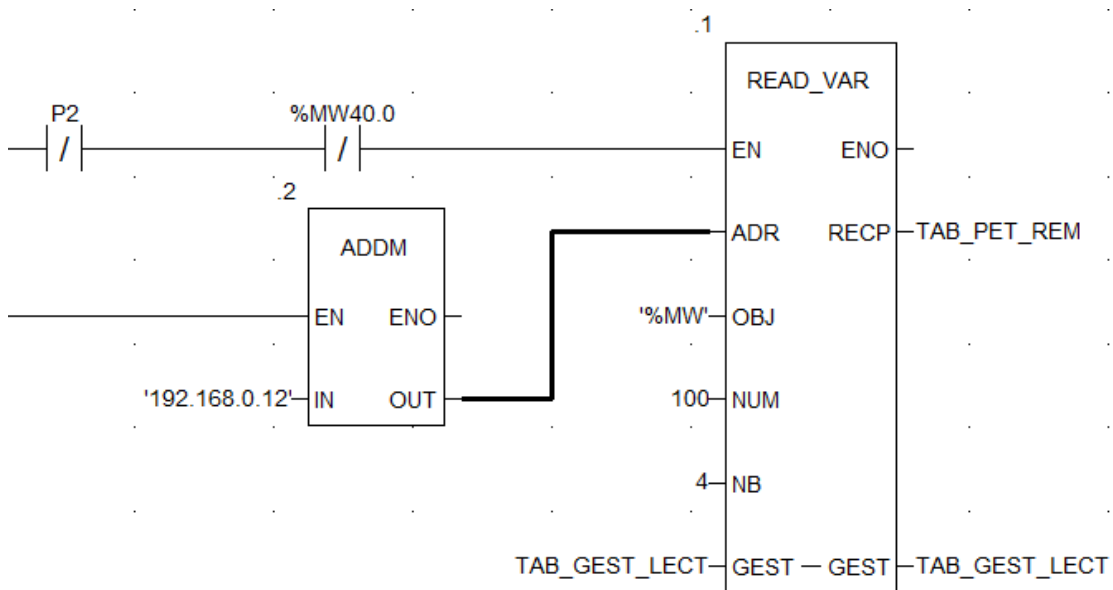


Figura 2.2-58: Diagrama LD de lectura de variables remota

En la entrada IN del bloque de función ADDM se pone la dirección IP del PLC del que se leerá la variable.

El correcto funcionamiento de cualquier instrucción de mensajería ha de ir condicionado a la negación del bit de actividad, esto es el bit cero de la primera palabra de gestión configurada en dicha instrucción, en nuestro caso tiene la siguiente dirección de memoria %MW40.0.

Por esta razón se pone en la entrada del bloque de función READ_VAR el primer elemento del array de la entrada GEST, y para que solo lea cuando está en modo remoto también debe estar el selector P2 en posición de funcionamiento remoto.

En la entrada OBJ se indica el tipo de objeto de lectura, en el caso del Modicon M340 puede ser bits internos (%M) o palabras internas (%MW).

En la entrada NUM se indica el índice del primer objeto a leer y en la entrada NB el número de objetos que se van a leer, en la primera ubicación de memoria se debe indicar las peticiones de servir una bandeja y en la segunda la de almacenarla.

En la salida RECP se contienen los valores que se han leído, en nuestro caso estos valores se guardan en la tabla TAB_PET_REM. El elemento TAB_PET_REM[1] (%MW10) es la petición de almacenar la próxima bandeja, el elemento TAB_PET_REM[2] (%MW11) el de servirla (%MW11), el elemento TAB_PETREM[3] (%MW12) anula una petición de servir una bandeja y, por último, el elemento TAB_PET_REM[4] (%MW13) anula una petición de almacenar la próxima bandeja.

Todas las variables que se usan para la comunicación con el autómata principal se encuentran en la tabla TAB_ESC, cuyo primer elemento está en la posición de memoria %MW200 y contiene un total de 8 elementos. A continuación se detallará que elemento indica cada cosa.

El elemento TAB_ESC[1] (%MW200) indica que se está sirviendo una bandeja, el elemento TAB_ESC[2] (%MW201) indica que se está almacenando una bandeja, el elemento TAB_ESC[3] (%MW202) indica que no se puede servir bandejas, el elemento TAB_ESC[4] (%MW203) indica que no se puede almacenar bandeja, el elemento TAB_ESC[5] (%MW204) indica que se ha recibido la petición de servir bandeja, el elemento TAB_ESC[6] (%MW205) indica que se ha recibido la petición de almacenar bandeja, el elemento TAB_ESC[7] (%MW206) indica que se ha anulado una petición de servir bandeja y el elemento TAB_ESC[8] indica que se ha anulado una petición de almacenar la próxima bandeja.

En el PLC principal debe haber una sección como esta para la lectura de las respuestas del autómata que controla el alimentador de bandejas. Para que la comunicación funcione correctamente es necesario que las variables de peticiones del autómata principal estén ubicadas entre las direcciones de memoria %MW100 y %MW103 y en el orden que se explica para la tabla TAB_PET_REM.

Además el autómata principal deberá tener una sección para la lectura de las respuestas del otro autómata en la que se comience a leer en la dirección %MW200 y se lean ocho ubicaciones de memoria consecutivas. La información obtenida en cada ubicación es la que se indica para la tabla TAB_ESC.

2.3. Programar Pantalla Táctil mediante el programa Vijeo Designer

2.3.1. Crear el proyecto

Cuando abramos el programa Vijeo Designer, lo primero que aparece es el asistente para crear un nuevo proyecto, y seguimos los pasos que se indican en el apartado 1.10.1. Es decir, pulsamos siguiente, en la siguiente ventana le damos un nombre al proyecto y seleccionamos el número de destinos que tendrá, en nuestro caso dejamos el que viene por defecto, ya que sólo tendremos un destino. Una vez hecho esto pulsamos siguiente, y en la nueva ventana seleccionamos el dispositivo para el cuál se realizará esta aplicación.

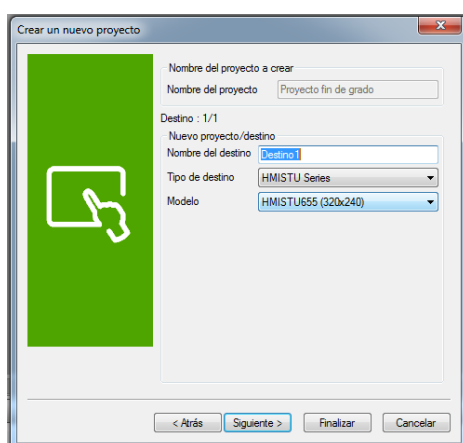


Figura 2.3-1: Selección del dispositivo HMI

En nuestro caso es el modelo 655 de la serie HMISTU, por tanto, primero se selecciona la serie y luego el modelo, como se ve en la figura 2.3-1. El nombre del destino es como identificaremos al dispositivo en el programa para crear los paneles de este, como sólo tenemos uno se ha mantenido el que viene por defecto.

2.3.2. Comunicación con un PLC a través de TCP/IP

Para añadir un controlador de dispositivo hay que hacer clic derecho en el nodo Administrador de E/S y selecciona Nuevo controlador, como se ve en la figura 2.3-2.

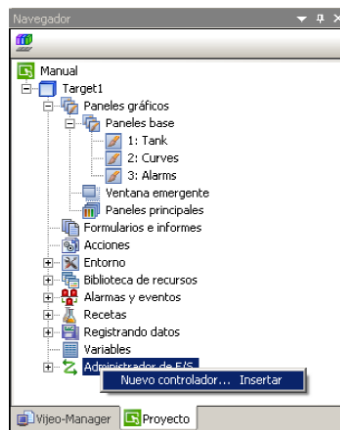


Figura 2.3-2: Añadir un nuevo controlador

Luego seleccionamos el controlador de la comunicación en la ventana que se abrirá. En el caso del módulo P34 2020 es la que se ve en la figura 2.3-3.

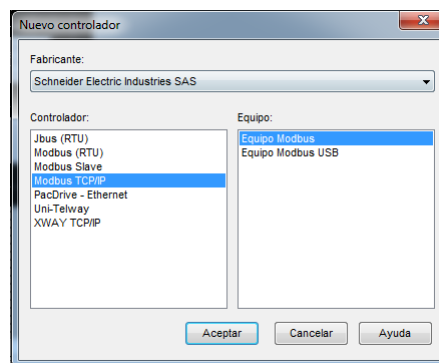


Figura 2.3-3: Selección del controlador de comunicación

Por último se asigna la dirección de red del PLC al que se va a conectar el dispositivo HMI.

2.3.3. Lista de variables

Las variables que se van a usar son las mismas que las que se definieron en el programa Unity Pro, ya que han sido exportadas con extensión .xvm e importadas al proyecto de Vijeo Designer.

Aunque en Vijeo Designer se pueden importar variables tanto con extensión .xvm como .stu, es aconsejable usar la extensión .xvm, ya que, a diferencia de la extensión .stu, no es necesario tener instalado Unity Pro para poder utilizarla.

2.3.4. Pantalla Principal

El panel “Pantalla Principal” es el panel que aparecerá en la pantalla cuando se comience la simulación. Este panel cuenta con una serie de botones que nos permiten desplazarnos entre los diferentes paneles.

La estructura del “Panel Principal” es la que se muestra en la figura 2.3-4.

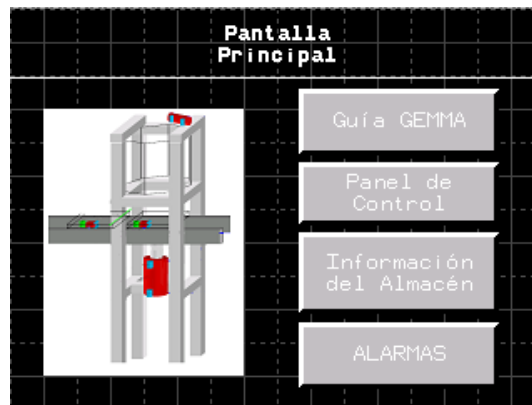


Figura 2.3-4: Pantalla Principal

Ahora veremos como se ha configurado un botón para que esto sea posible, el resto se configura de forma análoga a este.

Primero seleccionamos el icono “Interruptor” de la barra de herramientas como se ve en la figura 2.3-5, y con el ratón dibujamos el tamaño que deseamos.

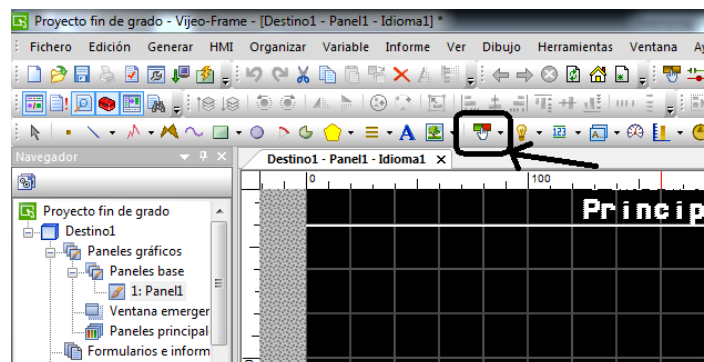


Figura 2.3-5: Crear botón en la pantalla

Una vez que hemos dibujado el interruptor, nos aparecerá la ventana de configuración del interruptor, figura 2.3-6, en la que seleccionaremos la operación “Panel” e introduciremos el identificador del panel al que queremos desplazarnos, es imprescindible pulsar el botón “Agregar” para que la operación quede guardada. También podemos incluir un piloto, modificar el estilo del interruptor, el color, etc.

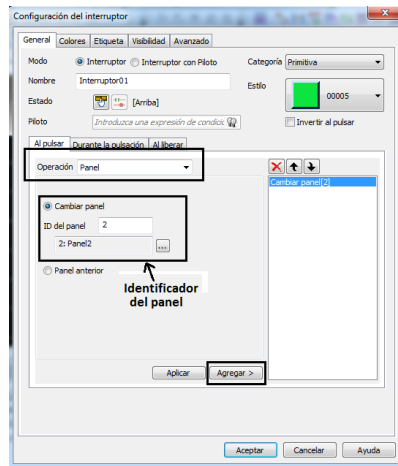


Figura 2.3-6: Configurar botón

2.3.5. Pantalla Guía GEMMA

En esta pantalla se indicará en qué estado de la Guía GEMMA se encuentra la máquina. Y a través del botón “HOME” se vuelve a la “Pantalla Principal”

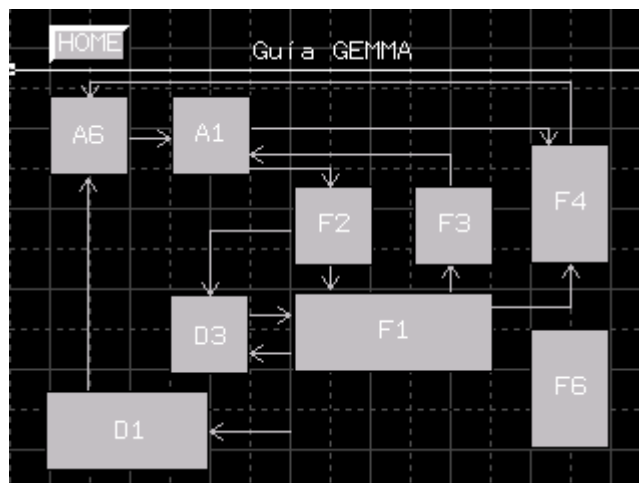


Figura 2.3-7: Pantalla Guía GEMMA

Para conseguir este resultado se ha dibujado la Guía GEMMA diseñada para el alimentador de bandejas con rectángulos, como se ve en la figura 2.3-7, y cada rectángulo tiene asociado una animación de relleno en función del valor de las variables que se expusieron en el Unity Pro y cuyo valor se modifica con el GRAFCET de la sección GEMMA. Esta variable tendrá valor 1 si está activa y 0 si está desactiva. A continuación veremos cómo se han animado el relleno de los rectángulos.

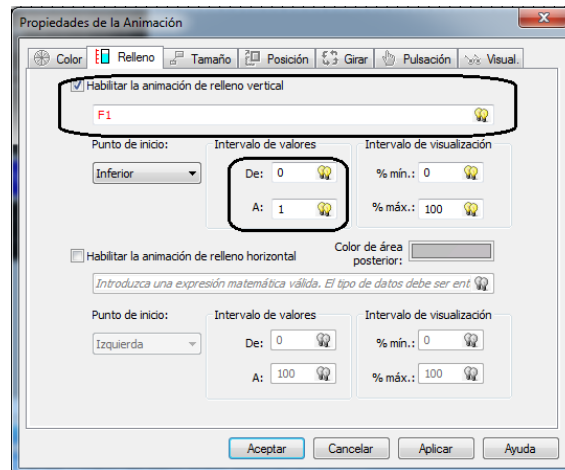


Figura 2.3-8: Animación de relleno

Para crear la animación de relleno hay que hacer doble clic en el objeto y se abrirá la ventana que se ve en la figura 2.3-8. En esta ventana seleccionaremos la pestaña “Relleno”, habilitaremos la animación y le asociaremos la variable en función de la cual se va a animar. Luego introducimos el rango de valores y pulsamos “Aceptar”

2.3.6. Pantalla de información del almacén

En esta pantalla se indicará la situación del almacén. Para esto contamos de un gráfico de barra que se rellena en función del número de bandejas almacenadas, y cuatro visualizadores numéricos que nos mostrarán el número total de bandejas, es decir, cuantas bandejas hay en el sistema, hay un piloto que se enciende si el número de bandejas introducido no está en el rango correcto (0-10) para indicar al operario que debe corregirlo; cuantas hay almacenadas y cuantas están pendientes de servir o almacenar. Además hay un quinto visualizador numérico que usaremos para indicar el número de bandejas que hay al comienzo de la ejecución automática. Y a través del botón “HOME” se vuelve a la “Pantalla Principal”.

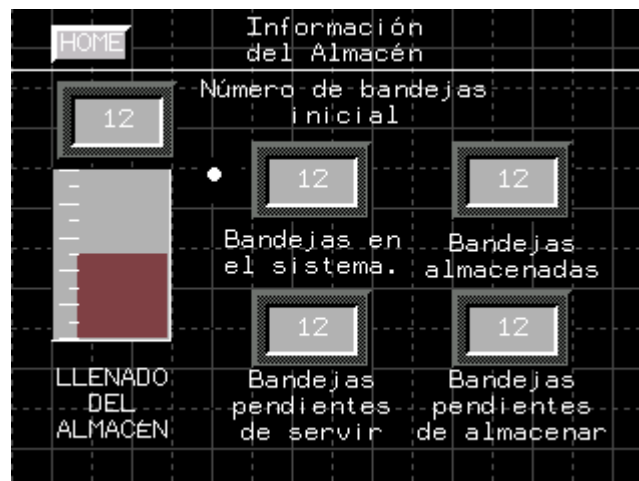


Figura 2.3-9: Pantalla Información del Almacén

Para visualizar los números sólo hay que hacer doble clic en el visualizador numérico, y en la ventana que se abre asignarle la variable y el número de dígitos que se va a mostrar, como se ve en la figura 2.3-10.

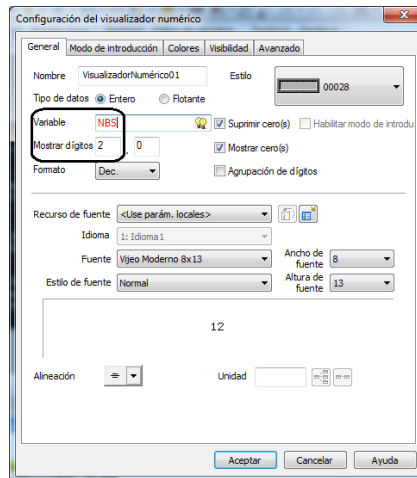


Figura 2.3-10: Configurar visualizador numérico

Y para el llenado del almacén se hace doble clic sobre la gráfica de barras, y en la ventana que se abre, se asigna una variable, y se configura el rango de valores y la escala, como se ve en la figura 2.3-11.

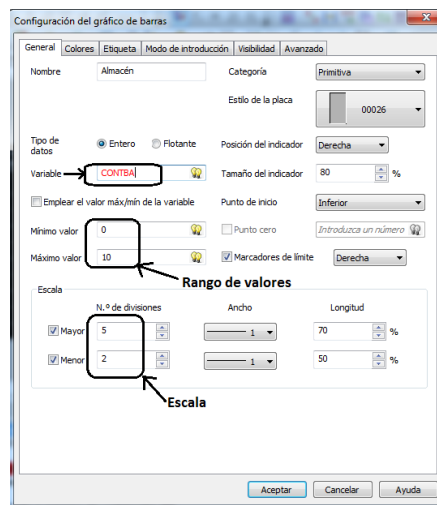


Figura 2.3-11: Configurar gráfica de

2.3.7. Pantalla panel de control

En esta pantalla se añaden opciones del panel de control con las que no se cuenta en el panel físico. Estas opciones son la posibilidad de elegir si se quiere comenzar o no la producción automática (MARCHA/PARO), indicarle al control si se ha retirado o incluido alguna bandeja durante la ejecución, anular una petición de servir o almacenar una bandeja y, aunque sí se cuenta con este botón cableado en el panel de

control, se puede rearmar el relé de seguridad usando una salida del PLC. Y a través del botón "HOME" se vuelve a la "Pantalla Principal".

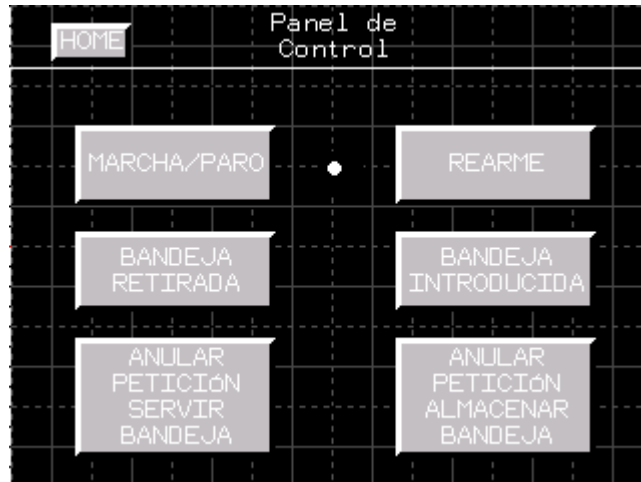


Figura 2.3-12: Pantalla Panel de Control

2.3.8. Pantalla de alarmas

En esta pantalla se mostrarán las alarmas activas, es decir, los fallos de los sensores y actuadores, los errores de seguridad y el mensaje para que se introduzca el número de bandejas del sistema.



Figura 2.3-13: Pantalla de alarmas

Para que se vean los mensajes de alarmas asociados a las variables que las activan, hay que añadir estas variables a un grupo de alarmas.

Hay dos formas de agregar una variable a un grupo de alarma:

1. Desde el editor de variables se selecciona el grupo al que se quiere añadir la variable.

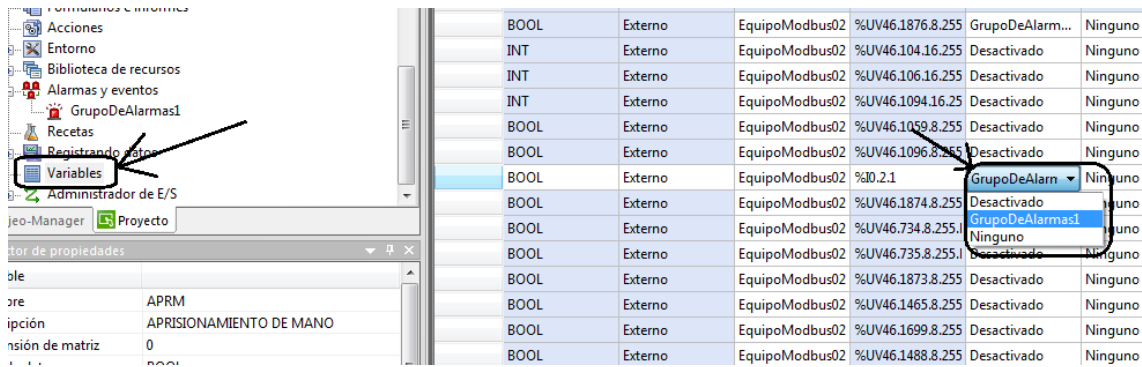


Figura 2.3-14: Añadir variable a grupo de alarma desde el editor de variables

2. Desde la ventana del grupo de alarma se pueden agregar variables, pero también indicar el mensaje que se mostrará si se activa la alarma de la variable asociada.

Para agregar una variable hay que hacer clic en Agregar y seleccionar la variable en la lista de variables que aparecerá en la ventana que se abre al pulsar este botón.

Para agregar un mensaje se hace doble clic en la casilla de mensaje de la variable y se escribe el texto deseado. En la figura 2.3-15 se muestran las alarmas que se mostrarán en la pantalla de explotación cuando surja alguna incidencia para ayudar al operario a identificarlas fácilmente y poder solucionarlas.

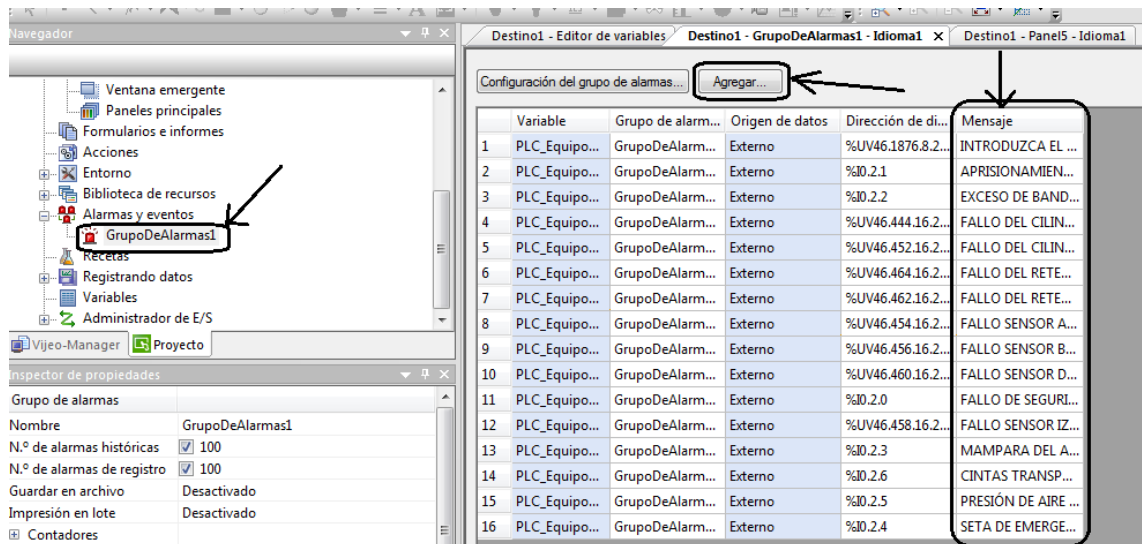


Figura 2.3-15: Agregar variable a grupo de variable desde la ventana de alarmas