



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Politécnica Superior de Gandia

Automatización del final de línea de producción en Vicky
Foods

Trabajo Fin de Grado

Grado en Tecnologías Interactivas

AUTOR/A: Garcia Serquera, Natxo

Tutor/a: Lloret Mauri, Jaime

Cotutor/a: Sendra Compte, Sandra

CURSO ACADÉMICO: 2022/2023

Resumen

Este proyecto se va a realizar en la empresa Vicky Foods durante la realización de las prácticas en empresa. Su objetivo es la automatización completa del final de una línea de producción, en la cual se tiene diseñar el algoritmo de funcionamiento, programar el código del autómatas, desarrollar el sistema, crear los planos, crear un programa para PLC y una interfaz gráfica (HMI) donde se pueden controlar algunos parámetros de la línea de producción como las velocidades, aceleraciones. Además, se deberá probar el sistema desarrollado en un entorno real, ya que el montaje de la línea está preparado para realizar pruebas.

Palabras clave

Automatización, producción, HMI, programación

Abstract

This project will be carried out in the company Vicky Foods during the internship in the company. Its objective is the complete automation of the end of a production line, in which you have to design the operating algorithm, program the code of the automaton, develop the system, create the plans, create a program for PLC and a graphic interface (HMI) where you can control some parameters of the production line such as speeds, accelerations. In addition, the developed system must be tested in a real environment, since the assembly of the line is prepared for testing.

Keywords

Automation, production, HMI, program.

Relación de trabajo con los ODS

En el Anexo 1, al final del documento, vemos la relación de este trabajo respecto a los objetivos de desarrollo sostenible pero aquí vemos algo más en detalle por que se ha decidido que estos ODS están relacionados directamente con el trabajo realizado.

Al tratarse de la automatización de una línea de producción podemos ver que está muy relacionado con el ODS 4 Educación de calidad, el ODS 8 trabajo decente y crecimiento económico y por último con el ODS 9 Industria, innovación e infraestructuras, ya que el TFG está realizado dentro de la empresa Vicky Foods.

ÍNDICE

1	Introducción	5
1.1	Historia de Vicky foods	5
1.2	Antecedentes	6
1.3	Motivación	6
1.4	Objetivos	7
1.5	Precedentes del Proyecto.....	8
2	Descripción del controlador	9
2.1	Componentes	9
2.2	Hardware.....	9
2.2.1	Pantalla HMI	9
2.2.2	Autómata programable PLC.....	10
2.2.3	Módulo de entrada.....	11
2.2.4	Módulo de salida	12
2.2.5	Variador.....	13
2.2.6	Fuente de alimentación.....	14
2.3	Software	15
2.3.1	Introducción del hardware dentro del software	16
2.3.2	Red de variadores ethercat	17
3	Descripción de la planta de producción.....	19
4	Diseño del algoritmo	23
4.1	Grafcet.....	23
4.2	Código.....	24
4.2.1	Entradas.....	25
4.2.2	Salidas.....	25
4.2.3	Resets	26
4.2.4	Pilotos y sirenas	28
4.2.5	Alarmas.....	29
4.2.6	Variadores	30
4.2.7	Control variadores	31
4.2.8	General	33
4.3	HMI.....	41
5	Conclusiones.....	45
5.1	Cumplimiento de los objetivos	45
5.2	Conclusiones sobre el proyecto.....	45

5.3	Problemas encontrados y cómo se han solucionado	45
5.4	Aportaciones personales.....	46
5.5	Futuras líneas de trabajo	46
6	Referencias.....	47

Ilustraciones

Figura 1- Interfaz hombre-máquina [7]	9
Figura 2 – Conectores de la interfaz hombre-máquina.....	10
Figura 3 – PLC NJ501-1500 con su fuente de alimentación [8]	11
Figura 4 – Tarjeta de entradas CJ1W-ID211 [9].....	11
Figura 5 – Características de los módulos de entradas.....	12
Figura 6 – Tarjeta de salidas CJ1W-OD212 [10].....	12
Figura 7 – Características de los módulos de salidas	13
Figura 8 – Variador 3G3AX-MX2 [11]	13
Figura 9 – Fuente de alimentación NJ-PD3001 [12]	14
Figura 10– Cuadro eléctrico [13]	15
Figura 11 – Tabla comparativa de las distintas marcas de PLC	16
Figura 12 – Bastidor de expansión.....	17
Figura 13 – Configuración de la red.....	18
Figura 14 – Imagen general del final de la línea de producción.....	19
Figura 15 – Ilustración donde se muestran las distintas zonas de trabajo de la planta	20
Figura 16 –Cinta pulmón	21
Figura 17–Fotocélula pistón	21
Figura 18 –Inductivo de las cintas del pulmón	22
Figura 19 –Vista completa del final de la línea de producción.....	22
Figura 20 –Diseño de GRAFCET	24
Figura 21 –Función de entradas	25
Figura 22 –Función de salidas	26
Figura 23 –Resets de la planta	27
Figura 24 –Reinicio de las alarmas	28
Figura 25 -Código dedicado a los pilotos y las sirenas.....	29
Figura 26 –Alarmas presentes.....	29
Figura 27 –Activación de alarmas	30
Figura 28 –Función del variador 11 para la red ethercat	31
Figura 29 –Control de los variadores.....	32
Figura 30 –Funciones de los variadores	33
Figura 31 –Marcha de la instalación.....	33
Figura 32 –Marcha de la instalación.....	34
Figura 33 –Cambio rápido de señales de intercambio	35
Figura 34 –Sección para saber si cargar las cintas del pulmón de arriba o de abajo	35
Figura 35 – Sección para saber si descargar las cintas del pulmón de arriba o de abajo	36

Figura 36– Sección de carga de la zona inferior	37
Figura 37– Sección de descarga de la zona superior	38
Figura 38– Sección de salidas	39
Figura 39– Sección de salidas	40
Figura 40– Sección de recetas	40
Figura 41– HMI sinóptico	41
Figura 42– HMI menú	42
Figura 43– HMI velocidades	42
Figura 44– HMI habilitación motores	43
Figura 45– HMI recetas	44
Figura 46– HMI alarmas	44

1 Introducción:

La automatización y control de los procesos industriales ha permitido a las empresas realizar importantes avances, permitiéndoles aumentar los procesos productivos y hacerlos más rápidos, seguros y eficientes.

A lo largo del documento, vamos a ver como se ha diseñado e implementado el código de un PLC (autómata programable) [1] y de una interfaz HMI. Estos nos permiten controlar el final de un proceso productivo en la industria alimentaria.

Todo este proyecto ha sido posible gracias a la compañía Vicky foods o antiguamente conocido como Grupo Dulcesol o Dulcesol [2], que me ha permitido elaborar el código y probarlo en un entorno Real.

1.1 Historia de Vicky foods:

Como hemos comentado en el apartado anterior, todo esto ha sido posible gracias a Dulcesol. Una empresa que fue fundada por Antonio, un joven aventurero que empezó en el mundo del emprendimiento desde bien joven. Pasó por distintos negocios de frutas donde fracasaron, llegando hasta tal punto que perdió los créditos del banco y sus amigos.

Después de fracasar en los años 50, Antonio y Victoria Fernández decidieron empezar en la panadería de su padre, la cual era un pequeño negocio familiar que proveía pan a todo el vecindario.

Pasaron los años y llegaron momentos duros para el pequeño negocio, ya que empezaron a crearse numeras empresas rivales. Fue entonces cuando a Victoria se le ocurrió la maravillosa idea de innovar y empezar a producir otros tipos de productos. Fue así como llegaron los primeros bollos, bizcochos... No obstante, a lo largo de los años siguieron produciendo estos productos, pero fue a mediados de los 70 cuando se crearon las famosas "Glorias" unas magdalenas nunca vistas.

A causa de la popularidad de sus productos y de la creciente demanda, fue entonces cuando decidieron abrir la primera fábrica de bollería ubicada en la localidad de Villalonga, en la cual trabajan alrededor de 800 personas. Siguieron pasando los años y la demanda seguía creciendo, sus productos llegaban a todas partes de España, fue entonces cuando decidieron abrir la segunda fábrica ubicada en la localidad de Gandía, la cual es la sede principal de la empresa.

Una vez posicionados como una de las marcas líderes en bollería, decidieron seguir mejorando el producto, por lo que construyeron una mega granja de gallinas la cual consta de alrededor de medio millón de gallinas.

Siguieron evolucionando y decidieron exportar sus productos a otras partes del mundo. En 1993 empezaron las primeras exportaciones a Portugal y en 2014 se creó la primera fábrica del Grupo Vicky Foods fuera de la península, más concretamente en Argelia, gracias a esta gran expansión se ha convertido en la empresa número 1 en el sector de la pastelería. No obstante, no solo produce alimentos de pastelería, sino que también ofrece otros estilos de productos desde cacao, pasando por alimentación infantil y terminando en otros tipos de productos saludables [\[3\]](#).

1.2 Antecedentes:

La tecnología de los autómatas programables ha estado durante hace mucho tiempo presente en las empresas. Se empezaron a implementar en ellas por su gran adaptabilidad, versatilidad y sencillez de implementación y por la cantidad de ventajas que nos ofrecen.

Algunas de las ventajas que nos ofrecen son, la eliminación de tareas repetitivas de los operarios de línea, aumentando de esta forma la eficiencia en la producción, generándole a la empresa ingresos mucho más rápidos.

Por lo que podemos afirmar que los autómatas programables son un elemento imprescindible no solo dentro de las empresas de producción, sino que en muchos otros ámbitos.

1.3 Motivación:

La realización de este proyecto de final de grado o TFG es el resultado de la curiosidad surgida a lo largo de la carrera, visto el gran mercado que ha creado la automatización de procesos. Se me ocurrió investigar un poco más en este ámbito. Una vez dentro de este mundo seguí investigando para saber cómo sería aplicar la automatización en la vida real, por lo que solicite hacer las prácticas en el apartado de automatismos de la empresa Vicky foods. Y de esta forma poder completar aún más el gran abanico de tecnologías, las cual conozco, ya que durante mi instancia en la carrera no se vio una asignatura de automatización como tal, pero sí que tenía las bases de programación, electrónica, robótica y control de procesos.

Siguiendo con esa necesidad de ahondar en la materia, me aceptaron la solicitud de prácticas enviada, por lo que podría lograr uno de mis objetivos.

Aprovechando el periodo de prácticas pregunté por la posibilidad de realizar un programa completo de un automatismo, para de esta forma poder ver los conocimientos que había ido interiorizando a lo largo del periodo de estudio. Por suerte, había la posibilidad de realizar un trabajo de final de grado, ya que disponían de una línea de

producción la cual no estaba en funcionamiento cuando empecé. Con lo que me puse manos a la obra, ya que me llamaba mucho la atención el poder trabajar con maquetas de procesos reales y seguir adquiriendo más conocimientos y sobre todo experiencia dentro de una gran empresa, ya que me brindaron la posibilidad de afrontar un problema real dentro del mundo de los automatismos.

El uso de software de Sysmac Studio [4] es otra de las claras motivaciones por las cuales elegí este proyecto ya que ofrece una gran cantidad de recursos y una amplia libertad de programación. Además de que permite la elaboración de interfaces humano máquinas muy intuitivas y completas para el usuario. No obstante, también tenía curiosidad por ver como conectar el código del PLC [5] con el Hardware utilizado.

1.4 Objetivos:

El objetivo principal del presente proyecto de final de grado (TFG) es desarrollar e implementar un proceso de automatización industrial completo construido por tres etapas principales. Dichas etapas están coordinadas entre sí para el correcto funcionamiento de la planta, además esto permite la manipulación y la actualización del software de una forma fácil e intuitiva. Todo el proceso debe de estar controlado y monitorizado desde una interfaz hombre-máquina para que los trabajadores de la planta puedan controlar el correcto funcionamiento de esta.

Para alcanzar el objetivo principal de este trabajo de final de carrera ha sido necesario la consecución de cada uno de los objetivos descritos a continuación.

- Diseño de un algoritmo capaz de realizar el correcto funcionamiento del final de la línea de producción.

- Aprendizaje del programa, Sysmac studio el cual nos va a permitir transferir el código y la pantalla del programa dentro del autómatas. Se ha elegido este software de programación, ya que el PLC que está en funcionamiento para esta línea de producción es un omron y este ofrece una gran facilidad para conectar con dicho PLC [6].

- Implementar el algoritmo a través del software de programación/simulación Sysmac Studio.

- Diseñar una Interfaz hombre-máquina intuitiva y funcional para que el usuario de planta pueda conseguir el funcionamiento óptimo de la línea.

- Comprobar el correcto funcionamiento del total en su conjunto tanto en simulación como en real.

Durante la realización del objetivo principal, dicho anteriormente, han aparecido otros objetivos, los cuales buscaban solucionar los problemas que iban surgiendo conforme se realizaba el proyecto. Como por ejemplo, la búsqueda de una función para poder controlar los variadores que controlan el motor.

1.5 Precedentes del Proyecto.

En el proyecto titulado “automatización y control de una maqueta de laboratorio mediante terminales táctiles y Sacadas” realizado por Ignacio Chacón Dalmau en 2016.

Consiste en un trabajo similar en cuanto a concepto, pero distinto dentro de las funcionalidades realizadas por cada proyecto. Su TFG simula un proceso de indexado de piezas con dos estaciones de mecanizado, mientras que este proyecto consiste en varios procesos diferenciados para la selección de caminos en una línea de producción.

Podemos ver semejanzas en cuanto a la necesidad de implementar un HMI, y en la parte de control de la línea, ya que ambos necesitamos controlar los motores mediante sensores para tener un control total sobre la línea.

2 Descripción del controlador.

2.1 Componentes:

A lo largo de esta sección vamos a conocer las funcionalidades y características de los siguientes dispositivos y componentes:

Una pantalla HMI omron, un PLC NJ501-1500 de omron, una fuente de alimentación Nj-Px3001, dos módulos de entradas y dos de salidas, 11 variadores en simulación, pero 6 controlados en real, un pistón, fotocélulas y 2 junteon Slave.

2.2 Hardware.

2.2.1 Pantalla HMI.

Las pantallas o también conocidas como interfaces hombre-máquina son las encargadas de visualizar las interfaces creadas dentro del sysmac studio, las cuales van a permitir a los usuarios hacer modificaciones sobre la línea de producción de una forma mucho más intuitiva que en código.

En la figura 1 vemos la interfaz hombre máquina empleada. En ella vamos a modificar las aceleraciones, deceleraciones y visualizar los motores activos, entre otros muchos parámetros.



Figura 1- Interfaz hombre-máquina [7].

Si vemos más en profundidad la figura 2 vemos que es el modelo NA5-9W001S-V1 de omron, la cual tiene unas dimensiones de 9" TFT, de 800 x 480 con 16777216 números de colores, con 2 puertos de Ethernet, 3 puertos USB y un puerto RS-232. La pantalla dispone de tres botones y de una zona táctil. Además, este dispositivo requiere una fuente de alimentación de DC de 24 V.



Figura 2 – Conectores de la interfaz hombre-máquina.

2.2.2 Autómata programable PLC.

Es un controlador lógico que se puede programar para que se encargue de automatizar los procesos industriales, electrohidráulicos, electromagnéticos y electroneumáticos. Igualmente, te permite automatizar desde un proceso en su totalidad hasta funciones específicas de cada máquina a la que se asocie. Funciona gracias a dos factores. El primero se trata de la instalación de diversos sensores y dispositivos de entrada a los que se conecta. El segundo, es que recibe la información de estos, procesa datos y los convierte en órdenes dependiendo de los parámetros que se hayan programado.

Además, cabe la posibilidad de conocer el tiempo de ejecución de cada trabajo, la productividad de cada máquina e incluso su temperatura. Es posible detener un proceso, acelerarlo, determinar la puesta en marcha de una alarma y detectar cualquier alteración en la producción. Esta solución se adapta siempre a tus necesidades y es totalmente personalizable.

El plc utilizado para este proyecto es el ya nombrado anteriormente NJ501-1500 de la marca omron observado en la figura 3. El cual tiene una memoria de programa de 20MB, una memoria de variables de 6 MB, distintos puertos de comunicación (ethercat máster, ethercatnet/IP, USB). Además, tiene un tiempo de tareas de ciclo principal o también conocido como ciclo SCAN de 0.5 ms que es el tiempo que tarda en ejecutar todo el código. También, podemos añadirle expansiones al autómata hasta un máximo de 40 módulos.



Figura 3 – PLC NJ501-1500 con su fuente de alimentación [8].

2.2.3 Módulo de entrada.

El módulo de entradas o también conocido como CJ1W-ID/IA. Es el encargado de convertir las señales de los sensores y entradas para que el PLC pueda realizar las operaciones con los datos obtenidos. En nuestro proyecto hemos utilizado 2 módulos CJ1W-ID211 con 16 entradas digitales, con tipos de entrada PNP/NPN. Con un retraso de OFF/ON de 0-32ms, este modelo lo podemos observar en la figura 4.



Figura 4 – Tarjeta de entradas CJ1W-ID211 [9].

En la figura 5 observamos las características de algunos de los modelos:


Product name	Specifications					Current consumption (A)		Model
	I/O points	Input voltage and current	Commons	External connection	No. of words allocated	5 V	24 V	
DC Input Units 	8 inputs	12 to 24 VDC, 10 mA	Independent contacts	Removable terminal block	1 word	0.09	–	CJ1W-ID201
	16 inputs	24 VDC, 7 mA	16 points, 1 common	Removable terminal block	1 word	0.08	–	CJ1W-ID211
	16 inputs (High speed)	24 VDC, 7 mA	16 points, 1 common	Removable terminal block	1 word	0.13	–	CJ1W-ID212
	32 inputs	24 VDC, 4.1 mA	16 points, 1 common	Fujitsu / OTAX connector	2 words	0.09	–	CJ1W-ID231

Figura 5 – Características de los módulos de entradas.

2.2.4 Módulo de salida.

Una salida de módulo es la encargada de suministrar un valor a otras variables para que los actuadores hagan sus funciones. En el proyecto se ha utilizado el módulo CJ1W_OD212 (ver figura 6) con 16 salidas digitales PNP y con una corriente de salida de 0.5 A.



Figura 6 – Tarjeta de salidas CJ1W-OD212 [\[10\]](#).

Podemos ver algunas características de las tarjetas de salida en la figura 7:

Specifications					No. of words allocated	Current consumption (A)		Model
Output type	I/O points	Maximum switching capacity	Commons	External connection		5 V	24 V	
Sourcing	8 outputs	24 VDC, 0.5 A Short-circuit protection	8 points, 1 common	Removable terminal block	1 words	0.10	–	CJ1W-OD204
Sourcing	16 outputs	24 VDC, 0.5 A Short-circuit protection	16 points, 1 common	Removable terminal block	1 words	0.10	–	CJ1W-OD212
Sourcing	32 outputs	24 VDC, 0.5 A Short-circuit protection	16 points, 1 common	MIL connector	2 words	0.15	–	CJ1W-OD232

Figura 7 – Características de los módulos de salidas.

2.2.5 Variador.

El variador se emplea para controlar la velocidad, aceleración y deceleración de un motor eléctrico a fin de mejorar el control de procesos. En la línea de producción que controlamos, encontramos 11 motores habilitados y 6 de motores los cuales se encargan de mover las cintas de toda la línea.

En la figura 8 podemos ver el modelo empleado 3G3AX-MX2.



Figura 8 – Variador 3G3AX-MX2 [\[11\]](#).

2.2.6 Fuente de alimentación.

La fuente de alimentación es la encargada de darle voltaje al autómata programable, así como a los módulos de entradas y salidas y a la memoria del controlador, el modelo usado NJ-PD3001 (Figura 9) tiene una tensión de 24V en continua (DC) con un consumo máximo de 60 W.

Item	Specifications	
Model	NJ-PA3001	NJ-PD3001
Supply voltage	100 to 240 VAC (wide-range), 50/60 Hz	24 VDC
Operating voltage and frequency ranges	85 to 264 VAC, 47 to 63 Hz	19.2 to 28.8 VDC
Power consumption	120 VA max.	60 W max.
Inrush current *1	At 100 VAC: 20 A/8 ms max. for cold start at room temperature At 200 VAC: 40 A/8 ms max. for cold start at room temperature	At 24 VDC: 30 A/20 ms max. for cold start

Figura 9 – Fuente de alimentación NJ-PD3001 [\[12\]](#).

A continuación, en la ilustración 10 podemos observar los anteriores componentes una vez ya instalados en la línea de producción y preparados para iniciar las pruebas en real. También, se pueden observar algunos elementos más, los cuales no son nombrados, pero básicamente son centralitas de seguridad para cumplir con los protocolos estándares de seguridad en las empresas y contactores por si se quiere quitar la luz solo a ciertos elementos del cuadro eléctrico.



Figura 10– Cuadro eléctrico [13].

2.3 Software.

El software elegido como ya nombrado, con anterioridad, ha sido sysmac studio el cual tiene un sistema propio de control de versiones dentro del propio programa. Este programa nos proporciona un entorno de programación de textos estructurados, funciones, diagrama de relés o contactos [14]. Gracias a todo esto, sysmac es un programa muy fácil de usar y muy intuitivo.

También proporciona simulaciones visuales de control de movimiento, como la visualización de resultados 2D y 3D y un gran sistema de compilación de errores.

Por último, se realizó un estudio para ver qué tipo de autómatas fueron los más utilizados (Figura11) y se pudo observar que Siemens controlaba el mercado junto con Rockwell durante el 2017, [15] pero en la empresa donde se realizó este TFG, la mayoría de los autómatas programables son de la marca omron y por tanto utilizan el software sysmac studio o CX-programer.

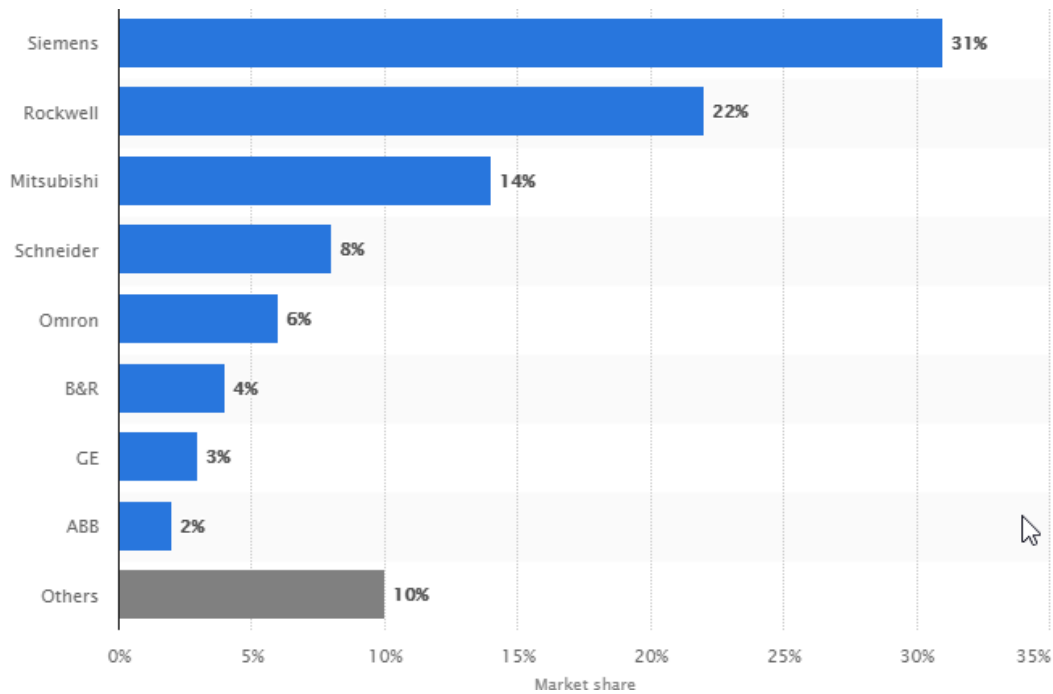


Figura 11 – Tabla comparativa de las distintas marcas de PLC.

2.3.1 Introducción del hardware dentro del software.

Dentro del software de sysmac studio hay un apartado específico para integrar todo el hardware. Se utiliza para poder asignar las variables a los distintos dispositivos. También se utiliza para personalizar el tipo de parámetros que quieres habilitar, dependiendo de sus necesidades.

Los parámetros que se quieren habilitar se modifican seleccionando dentro de cada unidad del módulo y habilitando el número de parámetro correspondiente, cada uno de ellos tiene una breve descripción donde indican para qué se emplean.

En la siguiente imagen (figura 12) podemos ver los elementos de hardware una vez integrados dentro del programa, también conocido como bastidor de expansión.

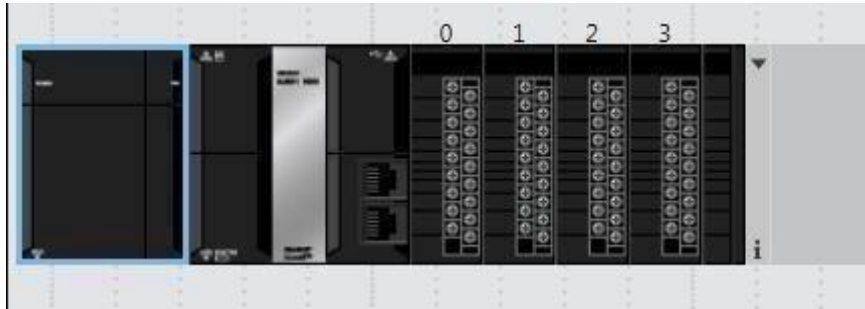


Figura 12 – Bastidor de expansión.

2.3.2 Red de variadores ethercat.

La red de variadores está conectada mediante ethercat que emplea una topología en forma de bus [\[16\]](#). Esta topología es muy empleada y eficaz para conectar y controlar cintas transportadoras.

Si analizamos la figura 13 podemos ver que coincide con la imagen 10, esto es debido a que la estructura de la red dentro y fuera del autómatas tienen que coincidir a la perfección ya que en caso contrario la red entraría en error y daría problemas y no dejaría conectar con los variadores. No obstante, este tipo de topología tiene algunos problemas, por ejemplo, si se rompiera un variador o diese algún error todos los variadores en serie con él caerían, mientras que, si fuese un error de consumo de aceleración o de temperatura, los siguientes motores estarían intactos.

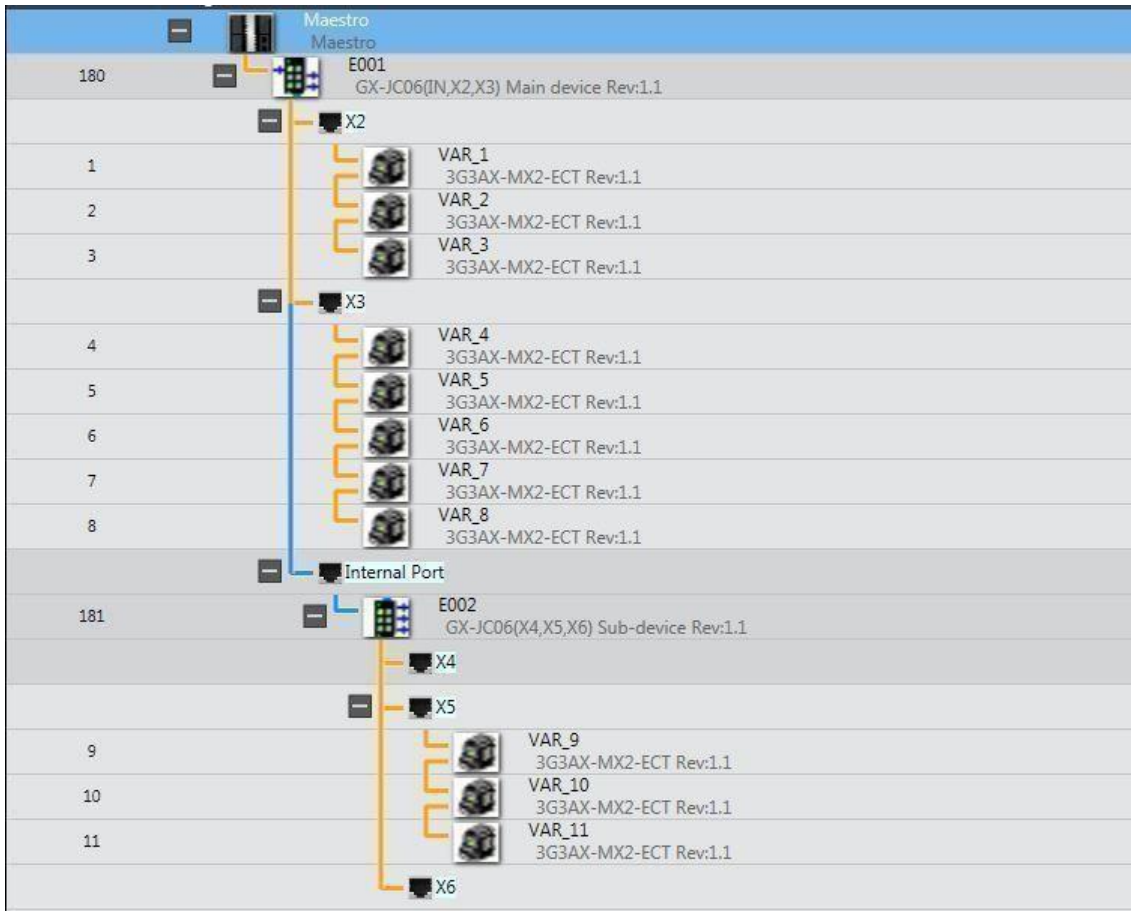


Figura 13 – Configuración de la red.

3 Descripción de la planta de producción.

En esta sección vamos a encontrar una descripción detallada de los sensores y los actuadores que se han implementado en el final del proceso productivo.

Esta línea cuenta con los siguientes dispositivos: 1 fotocélula [17], 1 pistón [18], 11 motores, 1 cuadro eléctrico, 6 cintas y 2 inductivos [19].

A continuación, vemos una imagen general (figura 14) del final de la planta de producción con la que ya podemos ver algunos de los actuadores y sensores los cuales hemos nombrado.



Figura 14 – Imagen general del final de la línea de producción.

También se pueden observar las máquinas Lawrence que es la encargada de distribuir el producto a esta zona de producción y la Ilpack una empaquetadora la cual, crea paquetes de productos de una forma muy precisa y a gran velocidad.

De estas máquinas solo tenemos señales de intercambio que nos indican si están encendidas o no, por lo que no se puede controlar los movimientos internos de las propias.

La zona de trabajo se puede dividir en dos zonas, las cuales cabe destacar que funcionan de forma dependiente, es decir, dependen de la señal de intercambio de la Lawrence y la Ilpack como observamos en la figura 15.

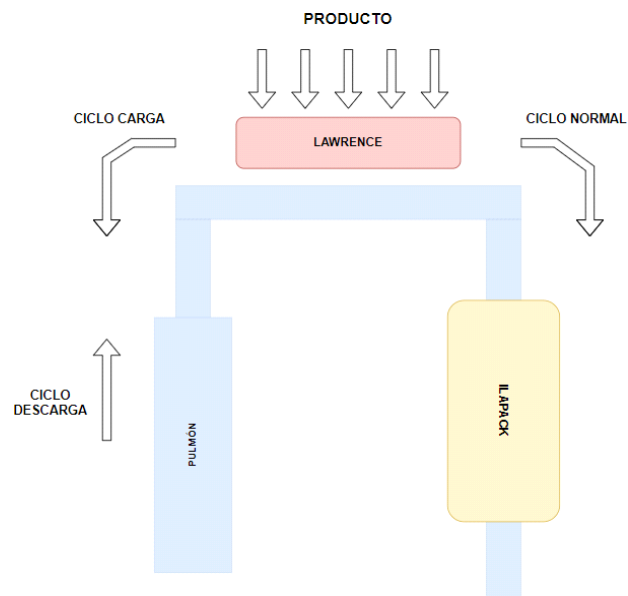


Figura 15 – Ilustración donde se muestran las distintas zonas de trabajo de la planta.

Partes de la línea:

-Ciclo de carga: Es la zona donde se va a ir acumulando el producto que procede de la Lawrence, esta etapa está condicionada con dos factores que venga producto de la distribuidora y que la empaquetadora este en error o apagada, de esta forma el producto que llegue no se desperdiciará y se irá acumulando en las cintas pulmón.

-Ciclo de descarga: Es la encargada de descargar el producto que se va acumulando durante la etapa de carga. El producto tendrá el siguiente orden de descarga: primeramente, se descargará la cinta superior, en caso de que esté vacía, empezará la descarga de la cinta inferior la cual terminara cuando se vacíe todo el producto de la propia cinta, durante el transcurso de esta etapa si se encendiese la Lawrence o si detectase producto saldría automáticamente de la etapa e iniciaría la correspondiente etapa.

-Cintas pulmón: es el lugar donde se acumula el producto en la etapa de carga, cabe destacar que son dos cintas una encima de otra como se muestra en la figura 16. Este proceso se inicia cuando una fotocélula que hay en el pistón detecta una unidad de producto. En ese preciso instante se empieza a cargar la cinta de abajo hasta llegar a la cantidad máxima de la cinta la cual tiene un número variable de producto máximo (manejable desde el código). Esto nos permite acumular productos de distintos tamaños. Una vez llena la cinta de abajo hay una pequeña transición para que el pistón suba y de esta forma avance la cinta superior, esta cinta sí que no tiene límite de producto ya que hay que pensar que el producto sigue saliendo y no se puede acumular en ninguna otra parte, ya que detrás está el horno y el producto no puede parar en él.



Figura 16 –Cinta pulmón.

-Fotocélula pistón: es una fotocélula que se encuentra al final del pistón situado en la parte derecha de la línea del producto, tal y como figura en la ilustración 17. Esta fotocélula es la encargada de contabilizar el número de productos que se carga o se descarga en las cintas del pulmón, es imprescindible su correcto funcionamiento.



Figura 17–Fotocélula pistón.

-Pistón: es el encargado de distribuir el producto a la parte inferior o a la parte superior dependiendo de lo llenas que estén las cintas del pulmón. Si está en la etapa de carga el pistón estará en 0 es decir abajo, en posición de espera una vez se llene esta cinta subirá y empezará la acumulación de la cinta superior hasta que llegue el momento de iniciar la descarga o la etapa de ciclo normal. El pistón se encuentra junto con una cinta que puede ir a forward o a reverse, dependiendo de si está en carga o descarga.

-Inductivos (figura 18): las cintas del pulmón van controladas por impulsos, es decir, por pulsos de 6, lo cual nos quiere decir que cuando el inductivo detecte 6 veces la rueda dentada habrá hecho un ciclo de avance completo y se reiniciará.

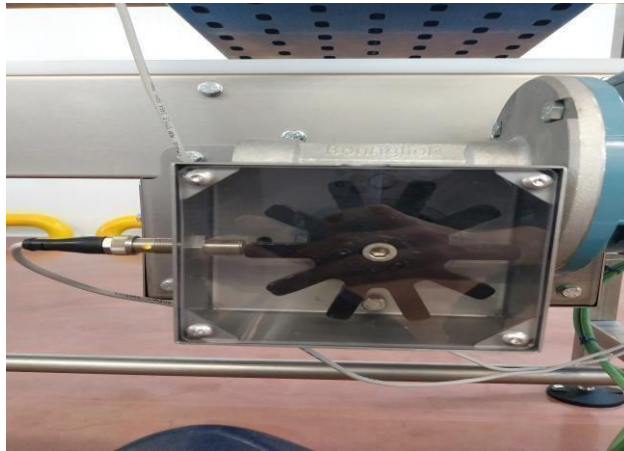


Figura 18 –Inductivo de lascintas del pulmón.

Las cintas se pueden dividir en dos tipos, las reversibles o las de un único sentido. Si vemos la figura 19 del circuito, las cintas de la zona de ciclo normal son de un único sentido, mientras que todas las otras cintas pueden ir a forward o a reverse.

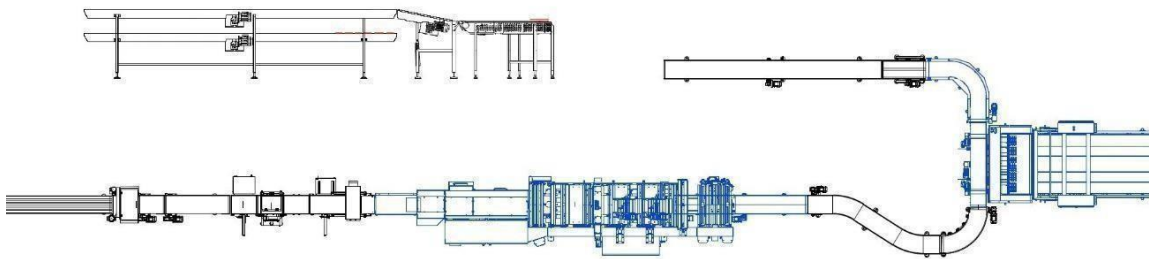


Figura 19 –Vista completa del final de la línea de producción.

4 Diseño del algoritmo

4.1 Grafcet

El código realizado se ha basado en metodología grafcet es decir, sigue un procedimiento de etapa/transición [\[20\]](#), lo cual nos permite tener controlado en todo momento en que etapa nos encontramos y cuáles son las condiciones que se están cumpliendo y las que no, para tener un control total sobre la producción.

Como podemos ver la ilustración 20 iniciará cuando pulsemos marcha y las condiciones estén para iniciar la siguiente etapa que es la de auto, una vez en auto casi inmediatamente se iniciará la etapa de carga-descarga-normal, que es como la etapa principal en la cual nos indica si el producto va a ir directamente a la empaquetadora o si el producto irá a la etapa de carga o de descarga. Podemos ver que las condiciones que se necesitan para el cambio de etapas una vez estamos en carga-descarga-normal es un cambio en la llapack o la Lawrence. Si estamos en el ciclo normal, el producto continuará hasta el final del proceso. Por el contrario, en la etapa de carga o descarga vemos que hay carga y descarga arriba y carga y descarga abajo, lo cual depende de la cantidad de producto que hay en la zona de acumulación. Pero que ocurriría si estamos en una etapa y de repente cambia una de las condiciones importantes es decir que la Lawrence o la llapack se encendieran, este proceso no se podía mostraren la imagen ya que el cambio se podría producir en cualquier momento o tal vez este cambio no se produciría si todo fuera correctamente. Por lo que no está indicado en la imagen, no obstante, en la parte de código veremos que el cambio será inmediato independientemente del momento en el que se encuentre el proceso.

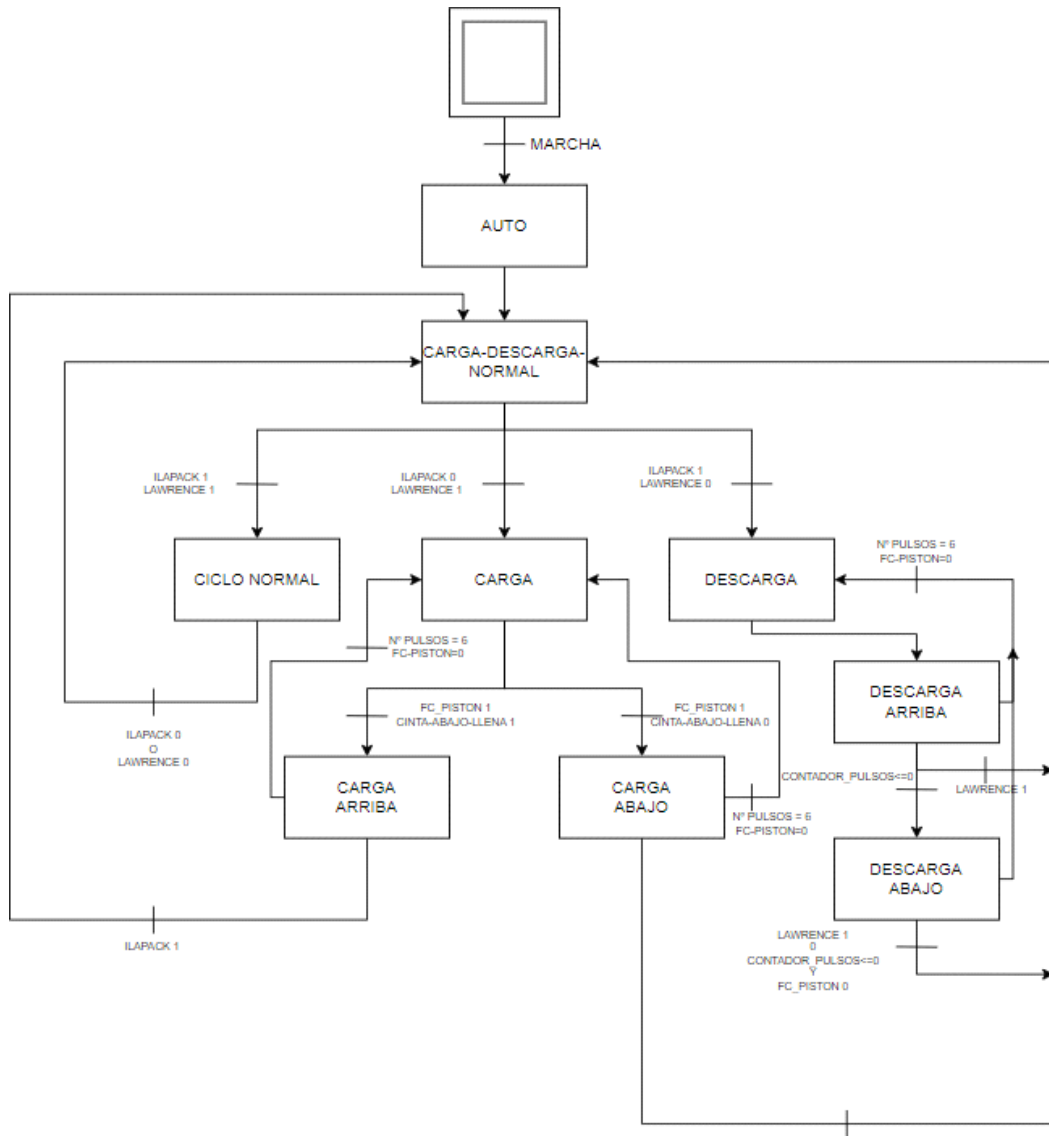


Figura 20 –Diseño de GRAFCET.

4.2 Código

A lo largo del siguiente apartado vamos a ver y analizar las funciones y la parte de código más relevantes que se han implementado para crear todo el proceso productivo desde la parte de la interfaz hombre máquina hasta el punto de los variadores para controlar los motores.

4.2.1 Entradas

En primer lugar, me gustaría empezar por la parte de entradas/salidas que ya se han comentado algunas de ellas en secciones anteriores, pero en esta parte vamos a ver que se ha creado una función específica para este estilo de tarjeta, ya que incluye un número de 16 entradas y otra tarjeta que incluye 16 salidas, no obstante, esta función es específica solo para este estilo de tarjetas. Si tuviéramos una tarjeta de 8 entradas o salidas esta función no se podría emplear porque produciría un error al tener más entradas/salidas de las disponibles.

En la figura 21 podemos ver una función con entradas numeradas de E000 hasta E115 lo cual son las entradas físicas que van conectadas directamente al automático. A la otra parte de la función podemos ver las variables que vamos a usar dentro del código. También, podemos observar que hay muchas entradas las cuales no tienen variables, estas se consideran que están libres para futuras expansiones.

Tarjeta3			FUN_ENTRADAS		
	ENABLE	ENO			
E100	Input_00	Q_Input_00	—	Introducir Variable	
TET	Input_01	Q_Input_01	—	Introducir Variable	
E102	Input_02	Q_Input_02	—	FC_PISTON	
E103	Input_03	Q_Input_03	—	FOTOCELULA...	
E104	Input_04	Q_Input_04	—	IND_M14	
E105	Input_05	Q_Input_05	—	INDUCTIVO D...	
E106	Input_06	Q_Input_06	—	IND_M15	
E109	Input_09	Q_Input_09	—	Introducir Variable	
E110	Input_10	Q_Input_10	—	Introducir Variable	
E111	Input_11	Q_Input_11	—	Introducir Variable	
E112	Input_12	Q_Input_12	—	Introducir Variable	
E113	Input_13	Q_Input_13	—	Introducir Variable	
E114	Input_14	Q_Input_14	—	Introducir Variable	
E115	Input_15	Q_Input_15	—	Introducir Variable	

FUN_ENTRADAS		
ENABLE	ENO	
E000	Input_00	Q_Input_00 — Introducir Variable
E001	Input_01	Q_Input_01 — CENTRALITA_OK
E002	Input_02	Q_Input_02 — Introducir Variable
TEST_FC_LAW	Input_03	Q_Input_03 — FC_LAWRENCE
E004	Input_04	Q_Input_04 — Introducir Variable
E005	Input_05	Q_Input_05 — Introducir Variable
E006	Input_06	Q_Input_06 — Introducir Variable
E007	Input_07	Q_Input_07 — Introducir Variable
E008	Input_08	Q_Input_08 — Introducir Variable
E009	Input_09	Q_Input_09 — Introducir Variable
E010	Input_10	Q_Input_10 — PARO
E011	Input_11	Q_Input_11 — MARCHA pulsador...
E012	Input_12	Q_Input_12 — PULS_RESET
E013	Input_13	Q_Input_13 — Introducir Variable
E014	Input_14	Q_Input_14 — Introducir Variable
E015	Input_15	Q_Input_15 — Introducir Variable

Figura 21 –Función de entradas.

4.2.2 Salidas

En la figura 22, por el contrario, podemos ver la función de salidas. Esta función es muy parecida a la de entradas, pero su función es muy diferente, ya que podemos ver que a la izquierda se encuentran las variables que se usan dentro del código y a la derecha vemos las salidas del automático, las cuales van a ir cableadas directamente a las salidas como el pistón, las sirenas...

salidas			
FUN_SALIDAS			
	ENABLE	Q	ENO
SIRENA	Output_00	Q_Output_00	S000
PILOTO_ROJO	Output_01	Q_Output_01	S001
PILOTO_VERDE	Output_02	Q_Output_02	S002
Introducir Variable	Output_03	Q_Output_03	S003
REARME_KR1	Output_04	Q_Output_04	S004
Introducir Variable	Output_05	Q_Output_05	S005
SUBIR_PISTON	Output_06	Q_Output_06	S006
K11_LAWRENCE	Output_07	Q_Output_07	S007
K12_LAWRENCE Camino haC...	Output_08	Q_Output_08	S008
Introducir Variable	Output_09	Q_Output_09	S009
Introducir Variable	Output_10	Q_Output_10	S010
Introducir Variable	Output_11	Q_Output_11	S011
Introducir Variable	Output_12	Q_Output_12	S012
Introducir Variable	Output_13	Q_Output_13	S013
Introducir Variable	Output_14	Q_Output_14	S014
Introducir Variable	Output_15	Q_Output_15	S015

Figura 22 –Función de salidas.

4.2.3 Resets

Por otro lado, vamos a ver una de las partes importantes de código, la cual es la encargada de los resets de la planta.

En la figura 23 empezamos a ver código en diagrama de contactos o de relés. En este tipo de lenguaje existen distintos tipos de contactos: los contactos abiertos y los cerrados.

-Contacto abierto: se activan cuando la señal de entrada está a 1.

-Contacto cerrado: estos están activos cuando su valor es 0. [\[21\]](#)

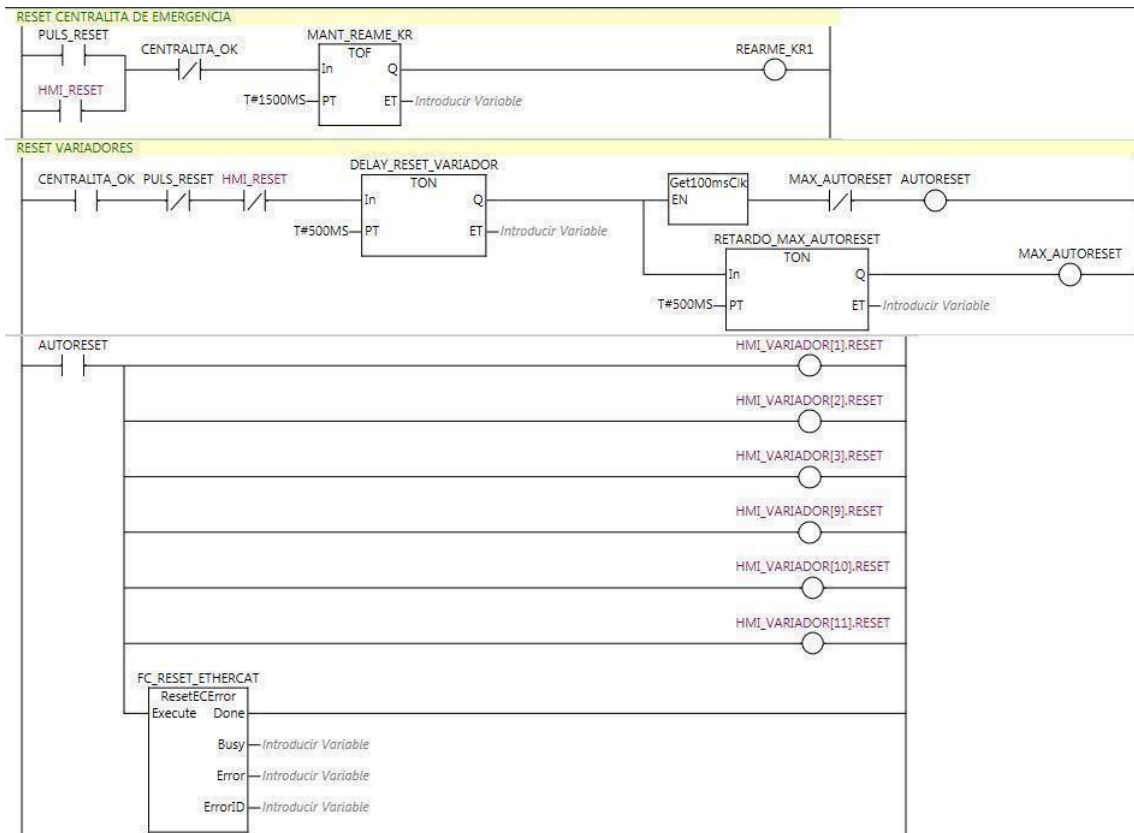


Figura 23 –Resets de la planta.

En la imagen anterior vemos que si pulsamos el pulsador de reset físico o el botón de pantalla y la CENTRALITA_OK no esté rearmada, se utilizará para rearmarla el contacto REARME_KR1 es importante, ya que es la centralita de seguridad que se encuentra en las instalaciones de autómatas reales.

Luego podemos ver el apartado del resets de los variadores, esta sección ya tiene algunas funciones especiales.

-TON: es un retardo a la conexión, lo que hace es esperar a que se cumpla la condición durante un periodo de tiempo que se establece. Una vez se cumple es cuando nos proporciona la salida.

-Get100msCLK: esta función genera pulsos cada 100 ms cuando tiene la señal de entrada.

Si analizamos la sección vemos que, si la centralita está a 1 y no está pulsado ni el pulsador de pantalla de reset ni el botón físico, vemos que generará flancos ascendentes cada 100 ms y activaran la marca de autoreset, los cuales se van a utilizar en la siguiente sección para reiniciar la red ethercat que es la que une los variadores y también lo utilizamos para reiniciar si algún variador está en error.

Para finalizar con el apartado de resets es importante reiniciar las alarmas para que la producción pueda continuar, para ello se ha creado este apartado que su función principal es cuando se pulse el botón de reset, poner todo el array de ALARMA_ACTIVADA a FALSE (Figura 24).

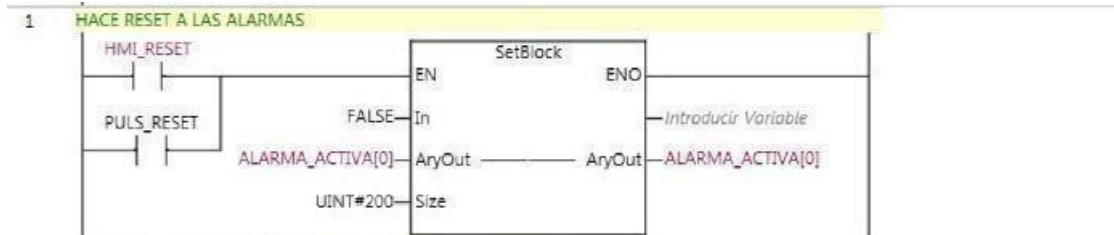


Figura 24 –Reinicio de las alarmas.

4.2.4 Pilotos y sirenas

Ahora vamos a analizar el apartado de los pilotos y sirenas, estos elementos se utilizan como indicadores de que la planta de producción está en marcha o si está en error. Si analizamos el código de la figura 25 vemos que el PILOTO_VERDE se activa si está en MARCHA y en auto o si está en auto, ya que está en paralelo. También activaremos una sirena los 5 primeros segundos desde que se enciende el PILOTO_VERDE.

Si vemos cuando se activa el PILOTO_ROJO vemos que se activa si está en paro ya que es un contacto normalmente cerrado o si esta pulsada la seta de emergencia o si hay alguna alarma activa, también activamos una sirena, pero esta vez la activaremos durante un periodo más largo de tiempo alrededor de unos 10 segundos, los cuales hacen que los operarios de la línea se den cuenta de que existen errores.

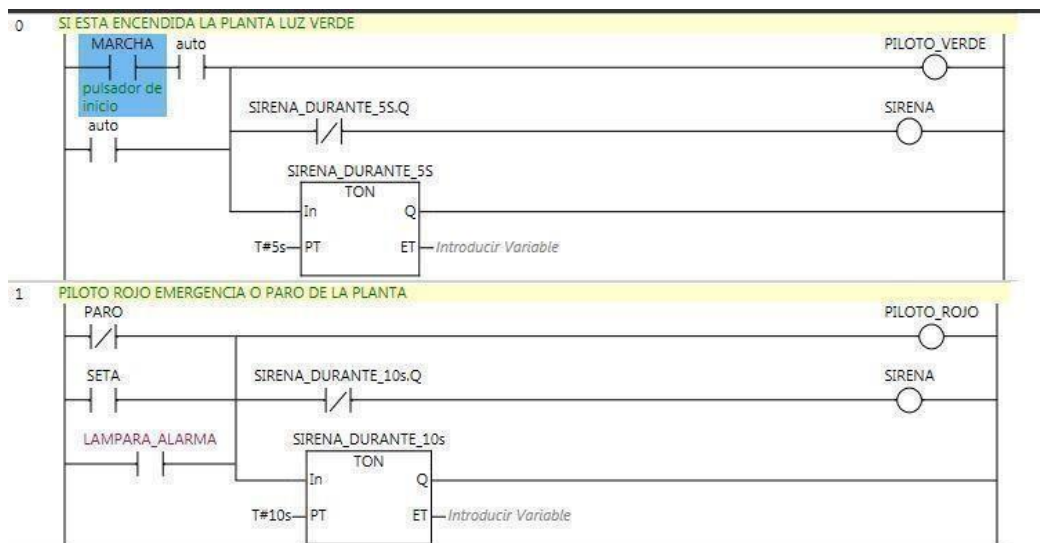


Figura 25 -Código dedicado a los pilotos y las sirenas.

4.2.5 Alarmas

Continuando con el tema de las sirenas vamos a ver como se crearon las alarmas que hacen saltar estas sirenas.

Las alarmas tienen una función “arySearch” que es una función la cual recorre un bucle y compara cada índice con la KEY. En nuestro caso vemos que va a recorrer un array llamado ALARMA_ACTIVADA el cual va a empezar en el índice 0 hasta el 199 ya que su tamaño es de 200, en el caso de que uno de los elementos es true va a activar la salida HMI_ALARMA_PRESENTE el cual generará pulsos cada 1 segundo habilitando la LAMPARA_ALARMA tal y como vemos en la imagen 26.

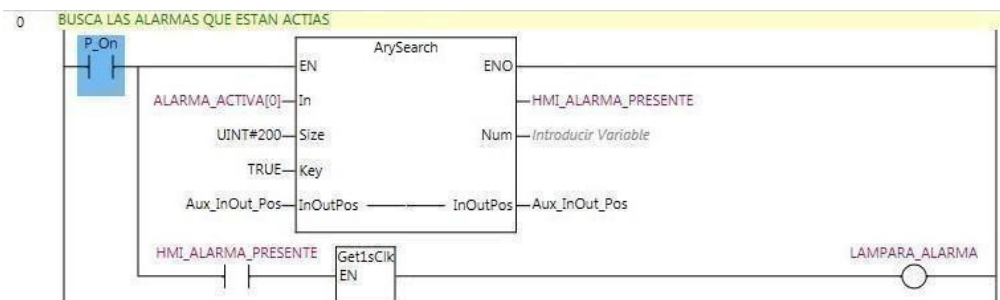


Figura 26 –Alarmas presentes.

Si continuamos con el tema de las alarmas, vamos a ver cuáles son las alarmas que se han creado para proporcionar seguridad a los usuarios, además de saber si ha ocurrido algún tipo de error a lo largo de la producción, para ello nos vamos a guiar analizando la figura 27.

En esta ilustración lo primero que nos encontramos es que necesitamos que la planta este en marcha, por eso las condiciones de MARCHA, CENTRALITA_OK y MAX_AUTORESET. A continuación, vemos que las condiciones clave para que salte la alarma de los variadores es que estén habilitados y si el nodo del variador no está OK es cuando se produciría la alarma. Pero no solo tenemos alarmas de variadores, también si la centralita de emergencias no está rearmada. Además, si está la planta en marcha y una fotocélula está obstruida por 60 s lo cual significa que delante del sensor se ha quedado un trozo de producto enganchado, lo que está provocando que la planta funcione incorrectamente. Otra alarma, es la de sí la Ilapack y la Lawrence están apagadas, esto nos haría imposible que funcionase la planta, ya que no tendría sentido encender la línea de producción.

Por último, tenemos la alarma de la seta de emergencias que se utiliza si ha ocurrido cualquier situación que pone en riesgo a un trabajador por atrapamiento de extremidades o por cualquier situación de peligro.

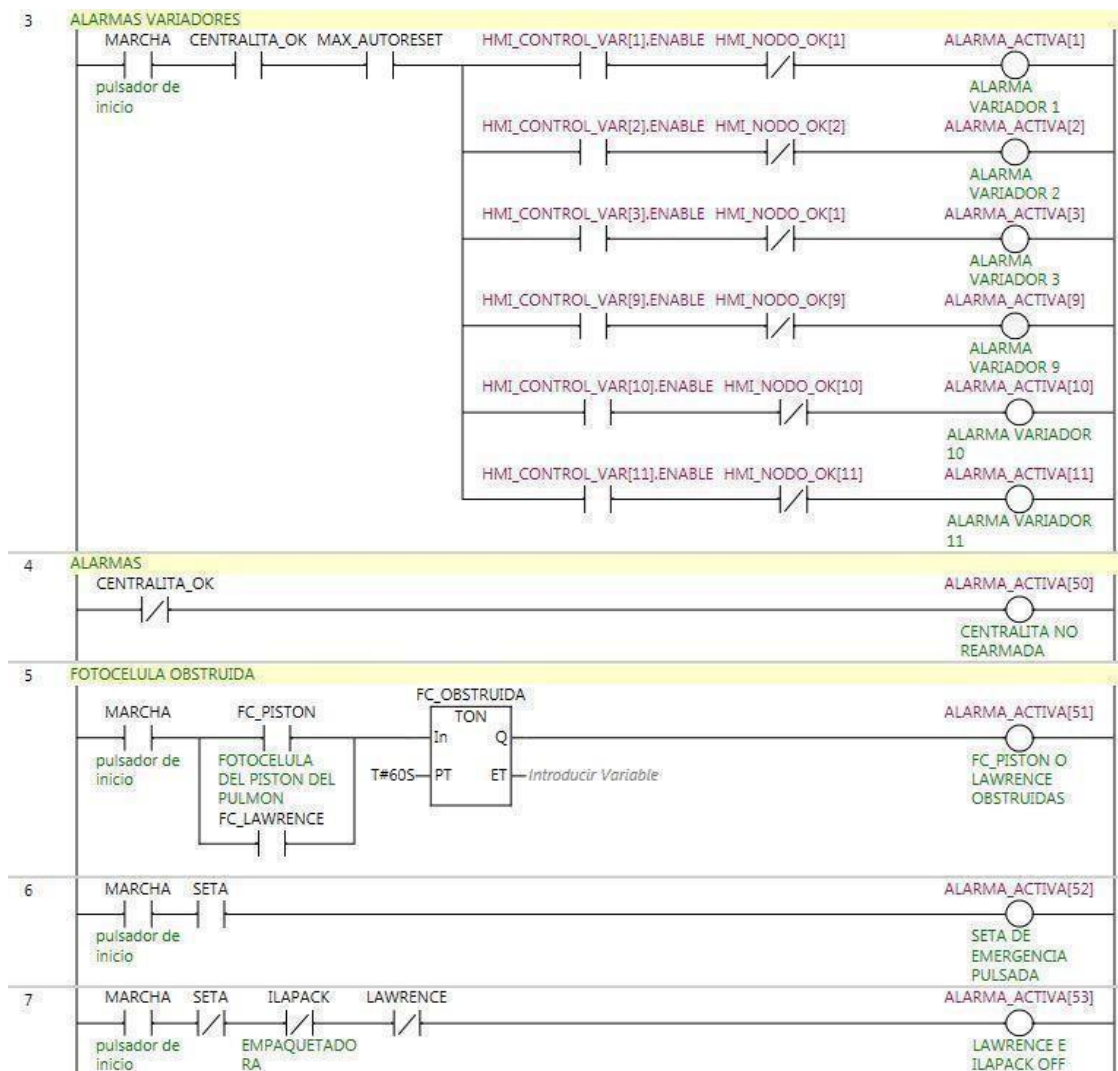


Figura 27 –Activación de alarmas.

4.2.6 Variadores

En la figura 28 vamos a ver una de las partes de código que más dificultad me ha causado que son la parte de controlar los motores a través de variadores, la función que vamos a ver es una función propia de sysmac studio la cual ofrece en su biblioteca, no obstante, el problema principal estaba en ver cómo funcionaba para poder usarla correctamente.

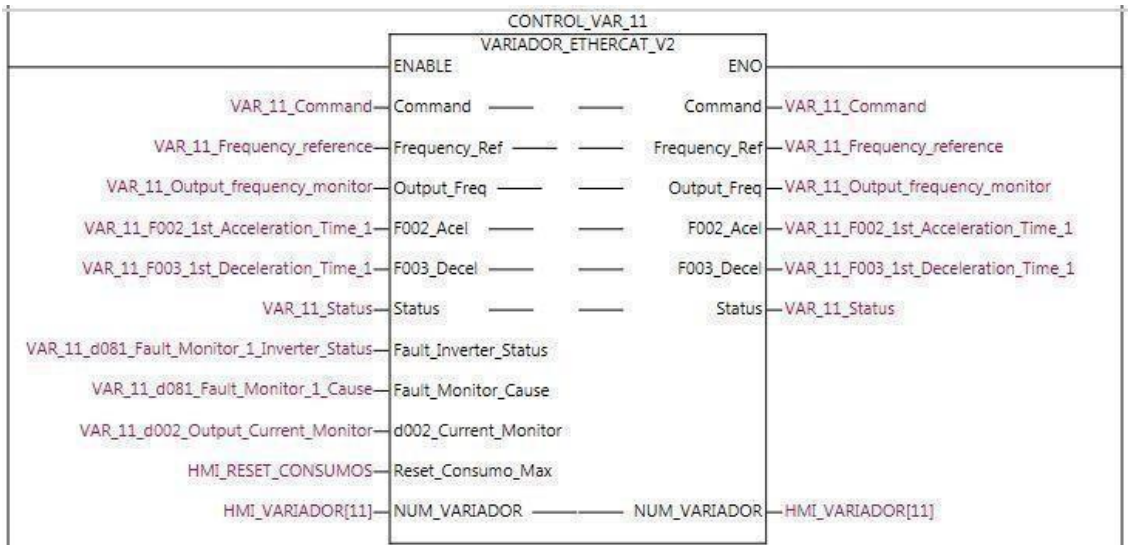


Figura 28 –Función del variador 11 para la red ethercat.

En la parte superior podemos ver la función número 11 del variador, la cual se emplea para controlar los variadores mediante ethercat.

Cabe destacar que se necesitan 11 funciones como estas, 1 por cada motor empleado.

Si analizamos los parámetros el primero VAR_11_command es el variador que quieres controlar, luego vemos el VAR_11_Frecuency_reference es la frecuencia de referencia del variador mientras que Var_11_Output_frecuency_monitor es el valor de frecuencia a la cual irá el motor, luego viene los parámetros de aceleración y deceleración los cuales indican la rampa de aceleración con la que el motor se pondrá en marcha o la rampa de frenado, es decir lo rápido que acelera o desacelera. El siguiente parámetro es ver cómo está el variador para saber si está o no en error. Si vemos la variable Var_11_d081_Fault_monitor_1_Inverter_status nos indica el error exacto que se está produciendo, además nos indica el estado actual es decir si está en run. Luego vemos Var_11_d081_fault_monitor_1_Cause, esta variable va ligada al "status" ya que nos indica el momento exacto cuando ha ocurrido el error. Var_11_d002_Output_Current_monitor es la corriente de salida que nos da el motor. Y por último el parámetro de reinicio de consumo y el número de variador.

4.2.7 Control variadores

Una vez ya asignadas las variables que va a tener el variador vemos que lo que realmente importante es el número del variador ya que contiene una estructura interna la cual nos va a permitir acceder a los parámetros de aceleración, deceleración o ver si está habilitado el motor.

Analizando el código observado en la figura 29 con más profundidad, se observa que el motor se pondrá en forward si está en automático, está habilitado el motor de pantalla, está su nodo OK y si está habilitada una salida M3 la cual se encuentra en la sección GENERAL. Si se cumplen estas condiciones se hace un MOVE el cual pone la velocidad que se ha establecido en pantalla dentro del variador.

También, tenemos un modo manual por si se queda producto atascado o por si se quieren realizar algunas pruebas o incluso para la limpieza de las cintas.

Luego vemos dos instrucciones MOVE las cuales pasan los datos de la variable HMI_CONTROL_VAR[x]. ACCEL a HMI_VARIADOR lo que está ocurriendo es que los parámetros de pantalla de aceleración y deceleración los estamos incluyendo dentro de las variables. En la segunda parte de la imagen se puede observar que las condiciones que hacen que se active el motor son distintas, ya que este motor puede ir tanto a forward como a reverse. Por lo que se tiene que adaptar el código a este. Podemos observar más diferencias como que el siguiente motor tiene dos tipos distintos de velocidad dependiendo del momento en el que se encuentre la planta.

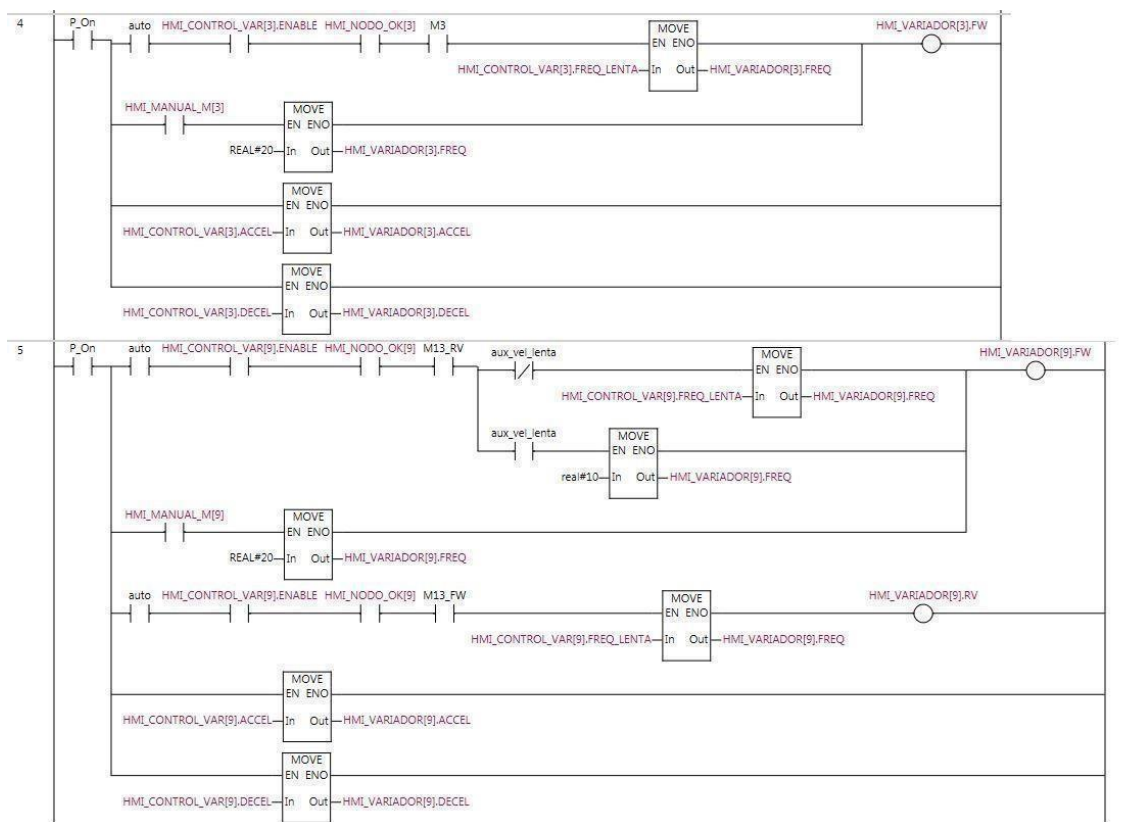


Figura 29 –Control de los variadores.

Para finalizar con la parte de variadores vamos a analizar dos funciones muy simples (figura 30)

pero que son necesarias para el funcionamiento de los variadores. La primera función es la encargada de saber si los nodos están en funcionamiento, es decir, es la encargada de ver que los variadores no se encuentran en error. La segunda función es la responsable de poner todos los parámetros de aceleración y deceleración de todos los variadores a 1 ya que si se encuentra a 0 cuando arranquen los motores pueden entrar en fallo.

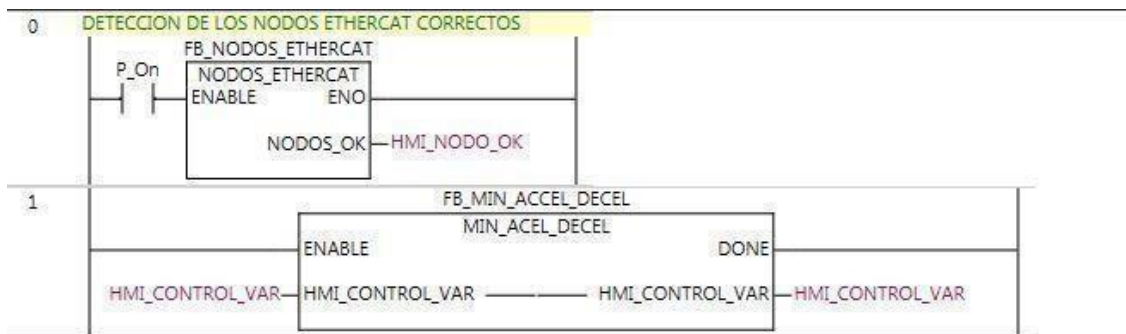


Figura 30 –Funciones de los variadores.

4.2.8 General

Este es el apartado más importante de todos es la parte principal del código donde está escrita toda la logística y todos los caminos posibles. Como hemos dicho en anterioridad se basa en una metodología GRAFCET por lo que la figura 20 nos permite seguir esta sección a la perfección, por lo que podemos ver los distintos caminos con mucha facilidad.

En primer lugar, vamos a analizar la marcha de la instalación (figura 31), la cual se enciende cuando pulsamos el pulsador de marcha y no está pulsado ni el paro ni la seta de emergencias, en este lugar podemos ver una peculiaridad y es que el paro funciona como inverso, cuando no está pulsado está a 1 y cuando lo pulsamos se pone a 0, por lo que si se cumplen estas condiciones activan la bobina auto que por un pulso hace un set a la etapa de CARGA_DESCARGA_NORMAL. Una vez aquí, observamos todas las etapas en una misma fila, esto nos sirve por si hay una avería localizar en error de una forma fácil e intuitiva.

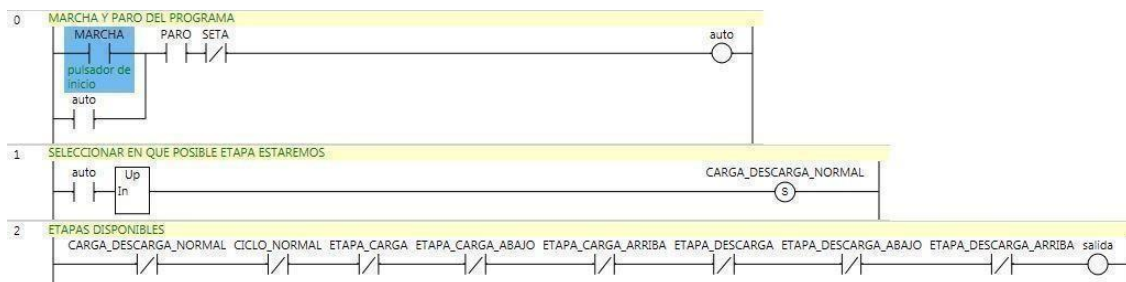


Figura 31 –Marcha de la instalación.

En segundo lugar, vamos a observar el lugar de cambio entre etapas, así como sus condiciones para provocar el cambio (figura 32).

Como vimos en la figura 20 vemos que las condiciones para que se produzcan las etapas coinciden con los contactos, por ejemplo:

Para que se inicie el ciclo normal se tiene que producir que la Ilapack y la Lawrence estén encendidas durante 0.5 segundos (tiempo de transición) de esta forma activará la salida mediante un “set” es decir, la marca se quedará activada hasta que se produzca un “reset”.

Este método de trabajo se utiliza para que únicamente se encuentre en una sola etapa por eso cada vez que se hace un set a una etapa se reinician las otras.

Si analizamos cuando entra la zona de carga vemos que se necesita que la ilapack esté apagada y que la Lawrence esté activa, mientras que para la descarga se necesita que la ilapack este a 1 y la Lawrence a 0.

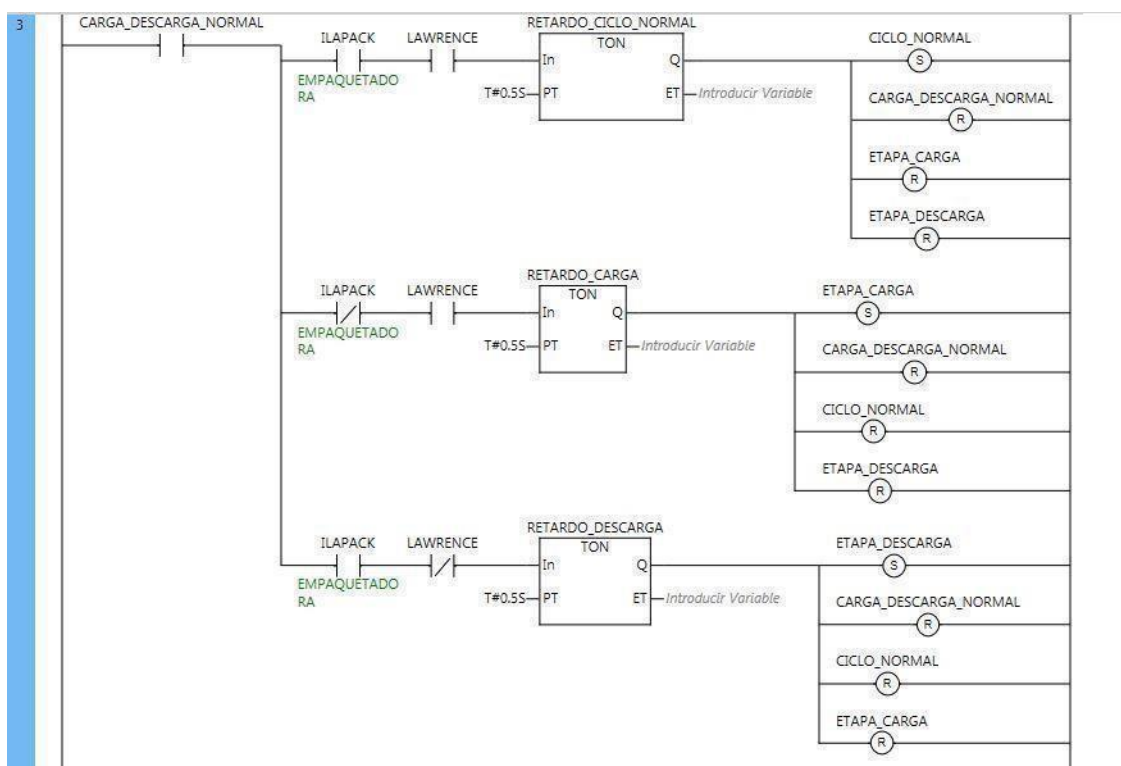


Figura 32 –Marcha de la instalación.

Observando la figura 33 vemos que esta sección se ha creado únicamente para el cambio rápido de etapas, es decir si se produce un cambio en el estado de la ilapack o de la Lawrence estando dentro de una etapa concreta, esta activará la salida de CARGA_DESCARGA_NORMAL la cual nos llevará de nuevo a la imagen 32 volviendo a empezar el proceso.

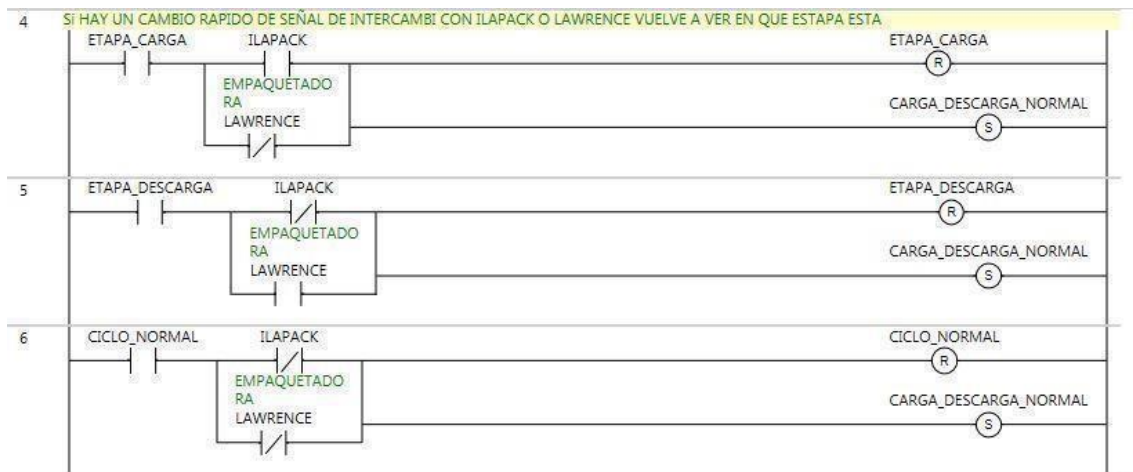


Figura 33 –Cambio rápido de señales de intercambio.

El código de la figura 34 se ha empleado exclusivamente para saber que parte de las cintas del pulmón se deben cargar. Toda esta sección depende de la marca CINTA_ABAJO_LLENA, que es la que determina cuál cinta llenar.

También se observa que si la FC_PISTON y la FC_LAWRENCE no detectan durante 30 segundos, se vuelve a la etapa de CARGA_DESCARGA_NORMAL para verificar otra vez las condiciones.

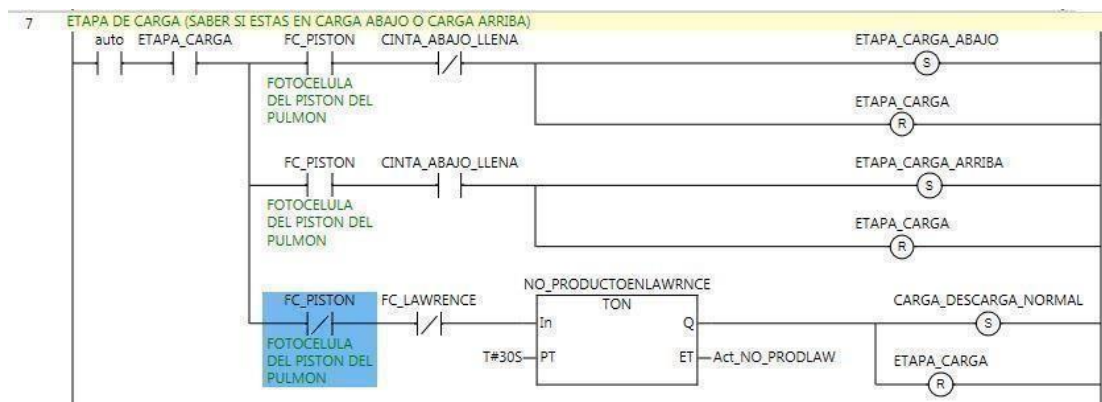


Figura 34 –Sección para saber si cargar las cintas del pulmón de arriba o de abajo.

Tal y como se muestra en la figura 35, podemos ver que esa parte de código va dedicada exclusivamente para saber la cantidad de producto que hay en la cinta superior e inferior para de esta forma mediante la función set activar la etapa de descarga correspondiente.

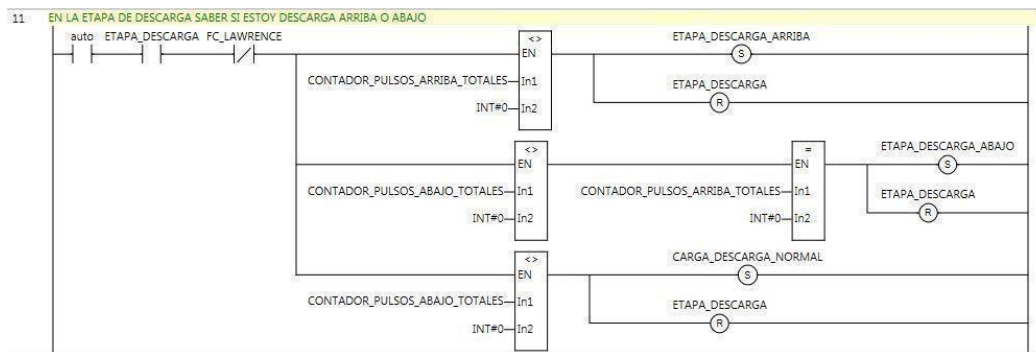


Figura 35 – Sección para saber si descargar las cintas del pulmón de arriba o de abajo.

A continuación, se va a analizar la sección dedicada para la carga de producto, más concretamente la zona de carga inferior (figura 36), no obstante, la zona de carga superior es muy semejante.

Se puede observar que a lo largo de esta etapa se necesita que la planta este en auto es decir encendida y rearmada. Una vez aquí vemos que si estamos en la etapa de carga abajo se va a encender la marca del motor 15 RV. Ahora viene la parte importante, la cual es que cada vez que el inductivo marque mientras el motor este encendido va a sumar 1 al número de pulsos abajo y al número de pulsos totales. Una vez los pulsos sea igual a 6, vamos a hacer un “MOVE” del número de pulsos a 0 para poder reiniciar el ciclo y cuando la fotocélula no detecte el producto vamos a hacer set a CARGA_DESCARGA_NORMAL para que vuelva a detectar las condiciones. Pero, todo este proceso se repetirá hasta que el contador de pulsos totales sea igual a 24, entonces activará la marca de CINTA_ABAJO_LLENA dando paso a la zona de carga superior. Las variables de los pulsos fueron creadas ya que se pidió que la etapa de carga fuera a impulsos.

Además, hay una condición para asegurar el correcto funcionamiento y es que si la llapack se enciende durante más de 2 segundo automáticamente volverá a comprobar las condiciones en la etapa de CARGA_DESCARGA_NORMAL, ya que con total seguridad se pondrá en la etapa de ciclo normal, que es con que debería estar encendida durante todo el proceso.

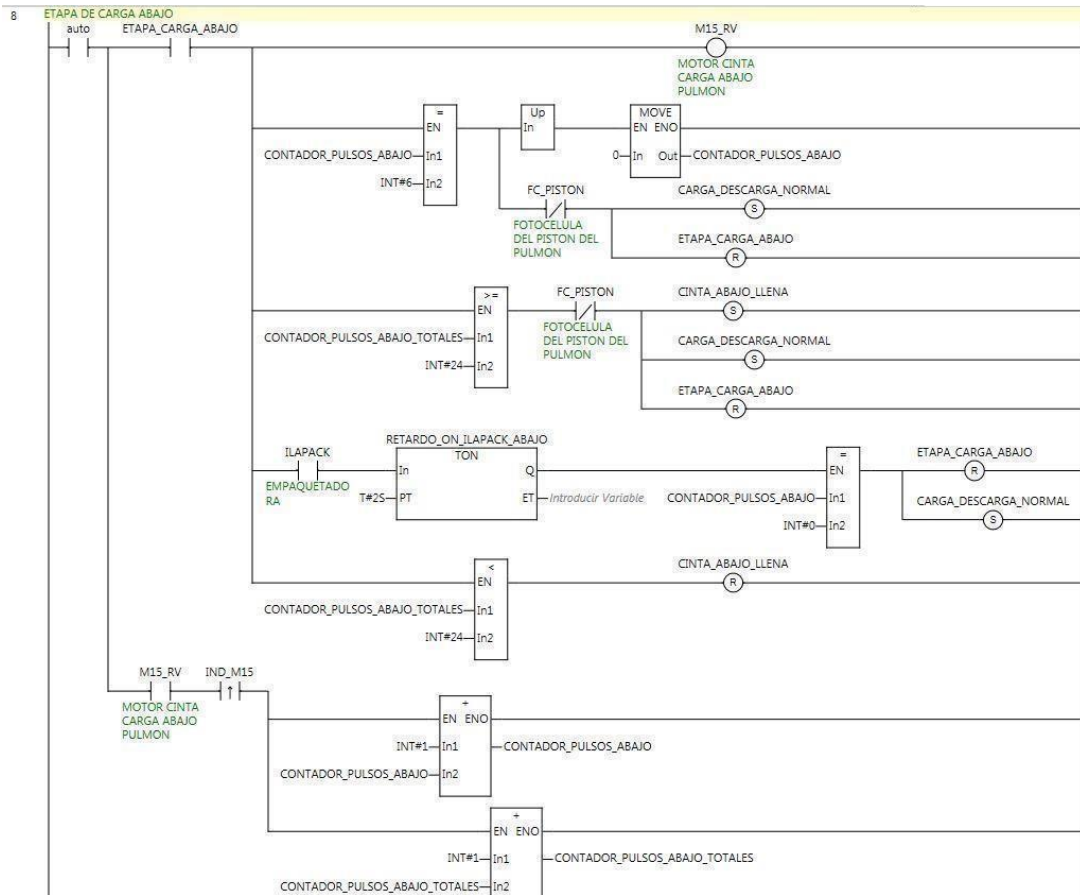


Figura 36– Sección de carga de la zona inferior.

En la siguiente sección vamos a dar paso a la descarga de producto que tiene su diferencia respecto a la parte de carga.

El funcionamiento de la descarga de la cinta superior es bastante similar al proceso de carga, pero su principal diferencia es que no funciona a pulsos ya que se decidió que fuera una descarga continua y sin pausas, por lo que no se tuvo en cuenta los pulsos solo se tuvieron en cuenta para que empezara la zona de descarga de abajo, ya que primeramente se iniciaba la zona de descarga de arriba al contrario que la carga que se empezaba desde la cinta inferior. Tal y como podemos ver en la figura 37

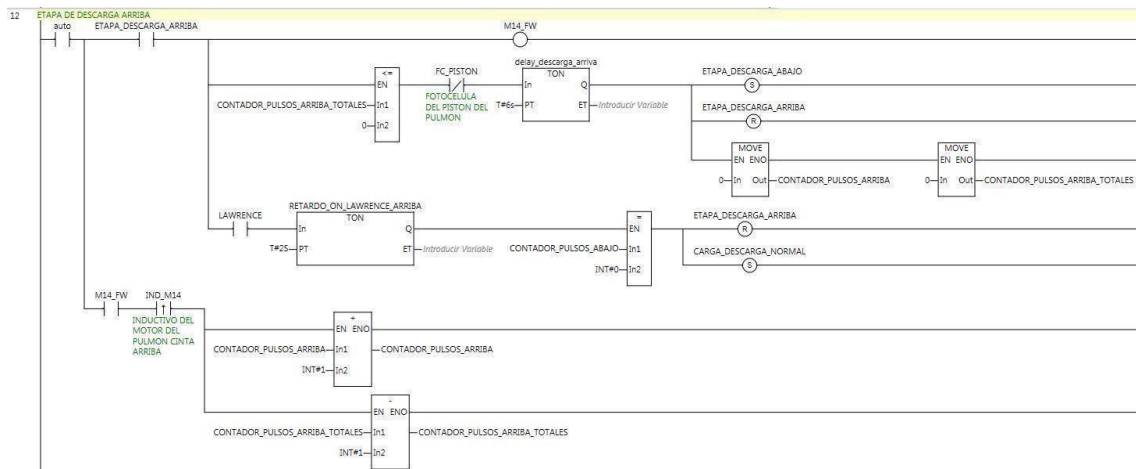


Figura 37– Sección de descarga de la zona superior.

En la siguiente parte de código vamos a ver cuándo se activan las salidas físicas, estas son las encargadas de encender las marcas que se encuentran en la sección de control variadores de la figura 29.

En la figura 38, podemos ver una salida que indica la activación de una velocidad lenta de la cinta, esta velocidad se aplica a la cinta del pistón y es empleada para que cuando el pistón esté ascendiendo no se desperdicie nada de producto y no se caiga al suelo, una vez el pistón ya esté en posición 2 segundos después la cinta vuelve a entrar a velocidad normal.

En esta misma imagen podemos ver cuando se activan los motores M1, M2, M3, M9, M10_FW, las condiciones de activación son las siguientes, si está en ciclo normal o en etapa de descarga o etapa de descarga arriba o en etapa de descarga abajo. La etapa de descarga es una etapa de transición para saber dónde descargar en este momento también se encienden los motores para que las cintas no se paren y funcionen de forma fluida.

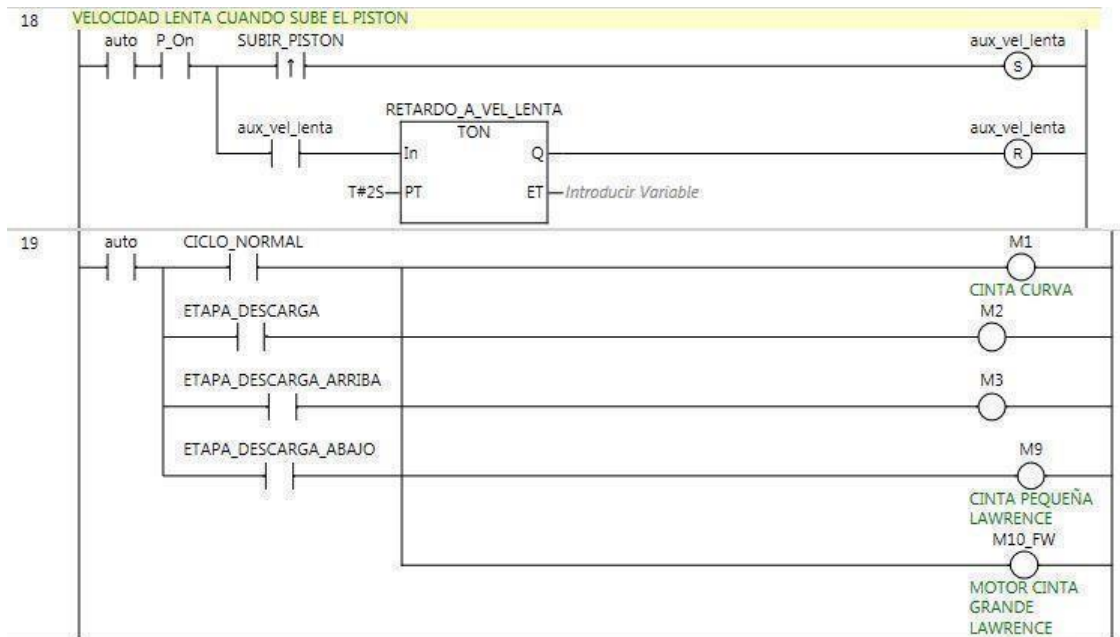


Figura 38– Sección de salidas.

En la siguiente imagen, figura 39, vamos a ver la segunda sección de salidas. A lo largo de esta sección podemos encontrar los motores que pueden ir tanto en forward como en reverse, dependiendo de la etapa en la que se encuentre el proceso. También, podemos ver cuando se activara el pistón, este se activa cuando el número de pulsos es igual o superior a 24, por lo que también nos indica que la cinta inferior está llena. Además, podemos ver cuando se tiene que reiniciar el pistón para iniciar la descarga de la zona inferior.

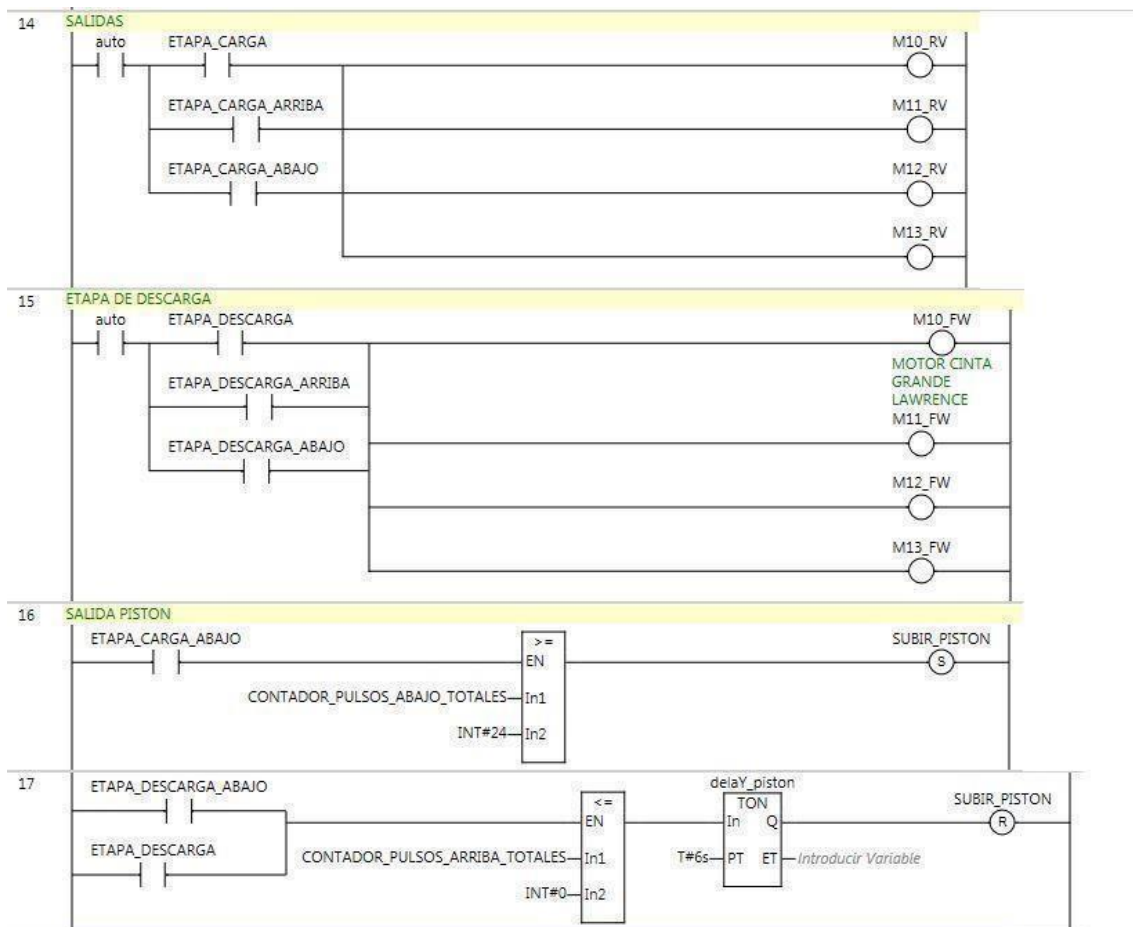


Figura 39– Sección de salidas.

Para finalizar se planteó que todos los parámetros de velocidad, aceleración y deceleración se guardaran en recetas para que de esta formase pudieran realizar distintos productos o distintos formatos de paquetes.

Si lo analizamos la figura 40, vemos que cuando pulsamos guardar_receta esta se pondrá a TRUE se guardarán todos los datos de cada variador en el número de receta seleccionado, de esta forma si se selecciona una nueva receta automáticamente se cargarán los parámetros de esa receta.

```

2
1 IF(guardar_receta=TRUE)THEN
2   FOR initial_value:=INT#1 TO INT#11 BY INT#1 DO
3     datos_receta[Num_receta].vel_lenta[initial_value]:=HMI_CONTROL_VAR[initial_value].FREQ_LENTA;
4     datos_receta[Num_receta].acel[initial_value]:=HMI_CONTROL_VAR[initial_value].ACCEL;
5     datos_receta[Num_receta].decel[initial_value]:=HMI_CONTROL_VAR[initial_value].DECEL;
6   END_FOR;
7 END_IF;
8 IF(Num_receta<>num_receta_anterior)THEN
9   FOR initial_value:=INT#1 TO INT#11 BY INT#1 DO
10    HMI_CONTROL_VAR[initial_value].FREQ_LENTA:=datos_receta[Num_receta].vel_lenta[initial_value];
11    HMI_CONTROL_VAR[initial_value].ACCEL:=datos_receta[Num_receta].acel[initial_value];
12    HMI_CONTROL_VAR[initial_value].DECEL:=datos_receta[Num_receta].decel[initial_value];
13  END_FOR;
14 END_IF;
15

```

Figura 40– Sección de recetas.

4.3 HMI

Una de las funcionalidades que se quería implementar desde el principio era una interfaz hombre-máquina en la cual un operador de línea pudiera modificar los parámetros de la planta para tener un control total en primera línea.

La primera pantalla (figura 41), contiene un diseño de la planta donde se pueden ver los actuadores para poder reaccionar de una forma más sencilla ante una avería. En ella podemos ver un icono de menú, un botón de “login”, además de la cantidad de pulsos de cada cinta.

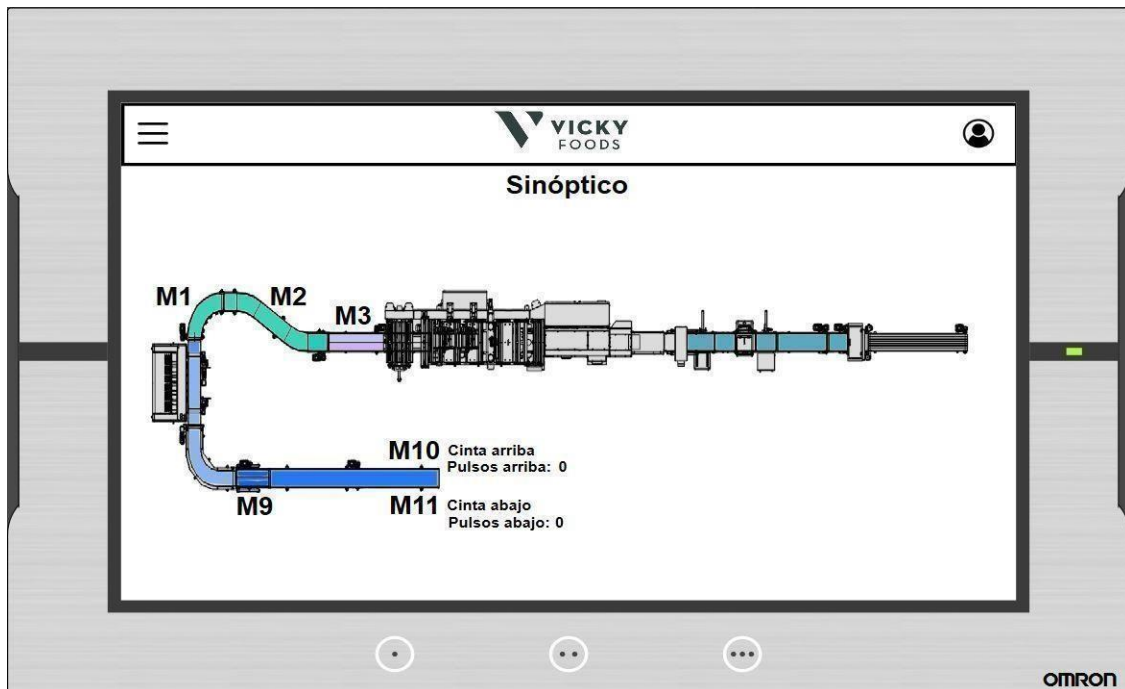


Figura 41– HMI sinóptico.

Como hemos comentado la pantalla contiene un menú para poder ir de una forma intuitiva a todas las pantallas, en la figura 42 podemos ver las distintas secciones dentro del menú. Además, podemos ver que se ha iniciado sesión como administrador. Dónde disponemos de un botón para cerrar sesión.

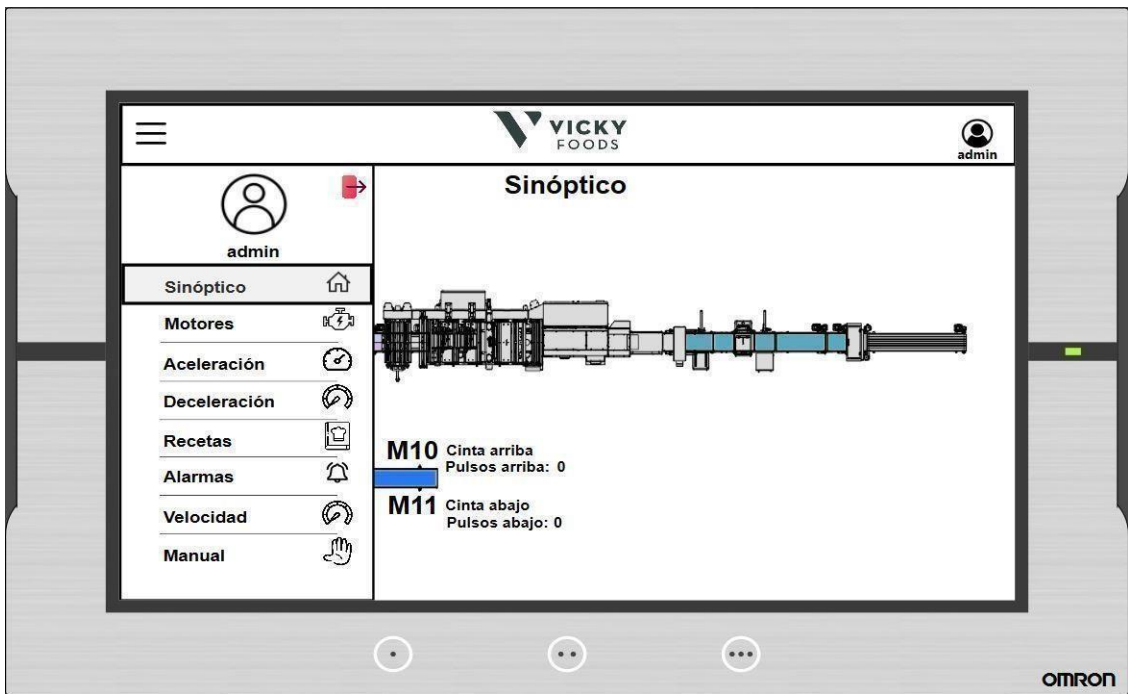


Figura 42– HMI menú.

En la figura 43 podemos encontrar unas casillas las cuales se utilizan para establecer la velocidad a la cual pueden ir las cintas, esta pantalla es muy parecida a la sección de aceleración y deceleración, ya que se planteó que las pantallas tuviesen un similar para que fuese más intuitivo.

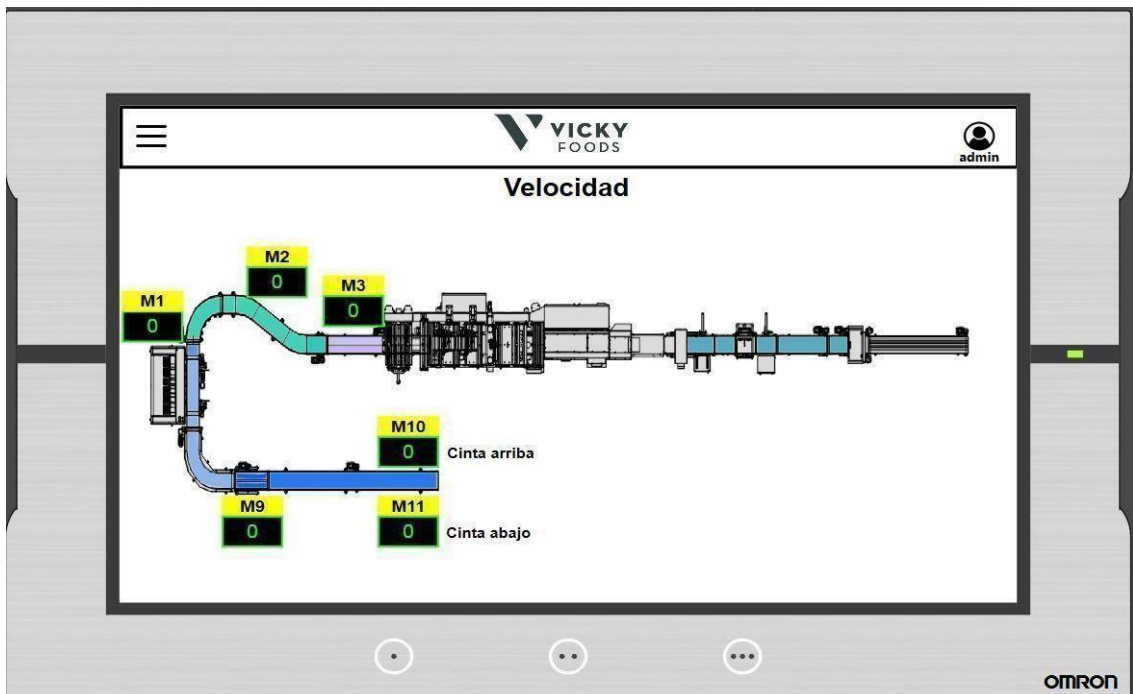


Figura 43– HMI velocidades.

En la figura 44 podemos ver la habilitación de los motores los cuales se encuentran en rojo, una vez la habilitación esté encendida se pondrán en verde. La línea de en medio (amarilla) se pondrá en verde una vez esté en marcha ese motor. También, disponemos de una pantalla para la parte manual, tal y como podemos ver en el menú de la figura 42, esta sección está dedicada para mover los motores pulsando sobre la pantalla directamente.

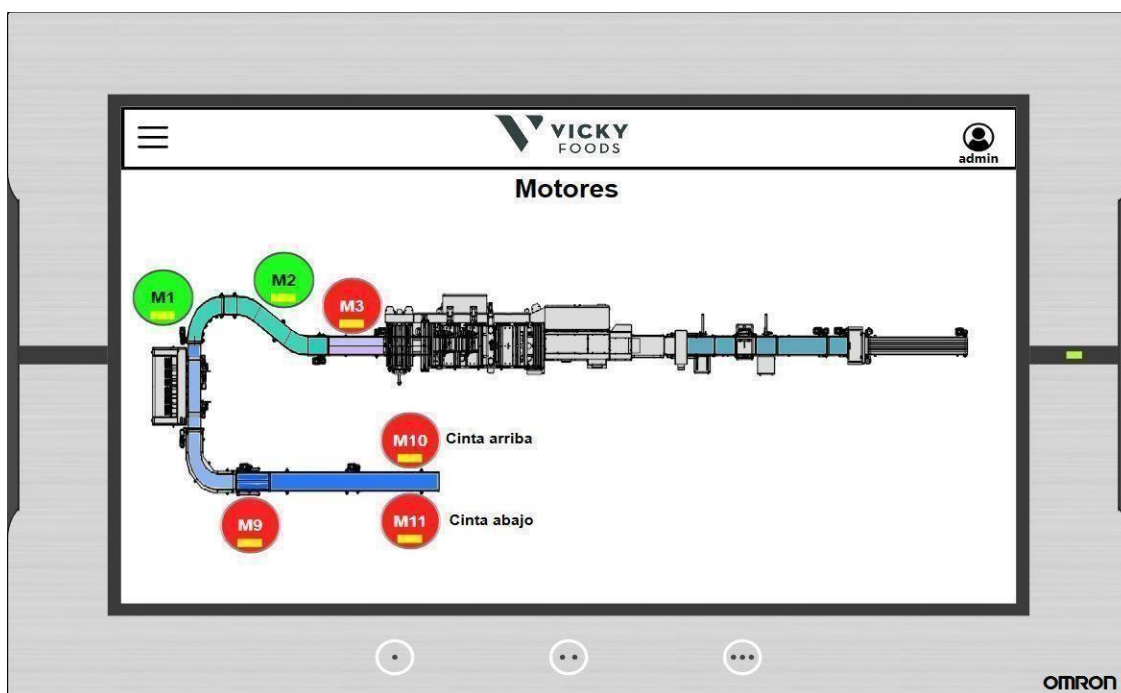


Figura 44– HMI habilitación motores.

En esta sección se observa el apartado de recetas para cambiar de una forma rápida el formato de paquetes o producto de la línea, sección muy útil y utilizada. (figura 45)



Figura 45– HMI recetas.

Para concluir con el tema del HMI, vamos a ver la figura 46, donde podemos ver una sección dedicada única y exclusivamente para ver las alarmas que se crearon en la parte de código de una forma más sencilla para el personal de planta.

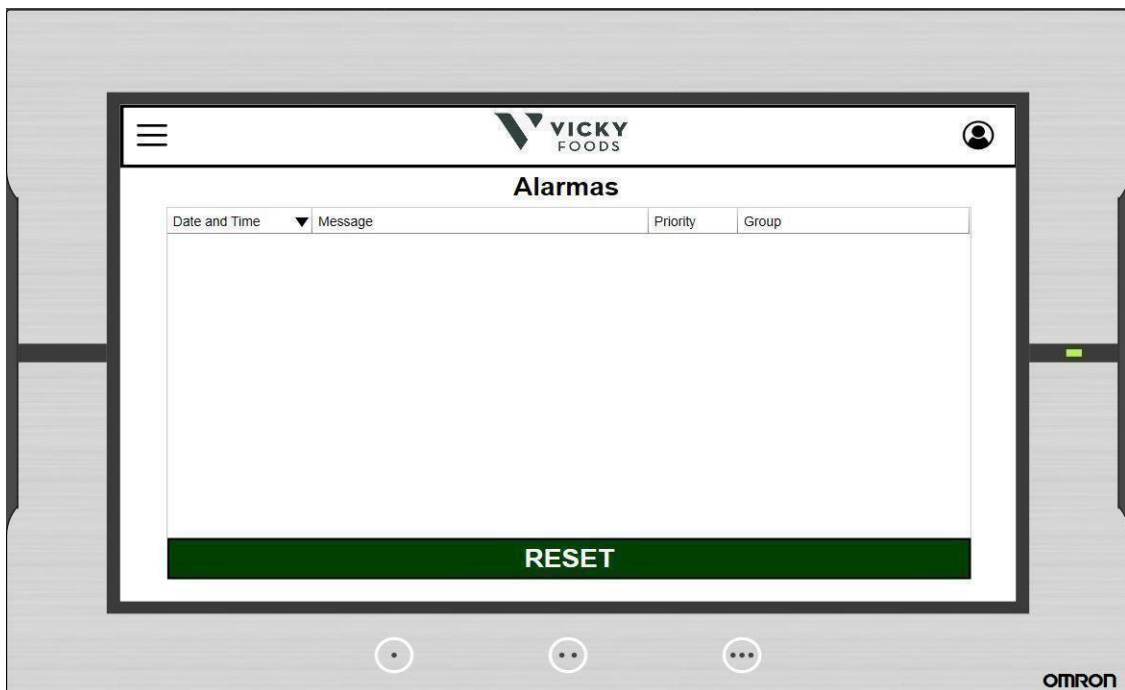


Figura 46– HMI alarmas.

5 Conclusiones

5.1 Cumplimiento de los objetivos

Para concluir podemos decir que hemos cumplido cada uno de los objetivos los cuales nos planteamos al inicio de este desafío. Ya que hemos sido capaces de desarrollar un proceso de automatización industrial completo. Realizando un algoritmo, basado en metodología grafcet con el software de sysmac studio. Una interfaz gráfica intuitiva y funcional. Además, de que ha sido posible probar todo esto en real lo cual nos ha dado la oportunidad de perfeccionar todas esas partes de código las cuales tenían errores.

5.2 Conclusiones sobre el proyecto

Al implementar este proyecto, sabía que me adentraba en un mundo nuevo y complejo, el cual tenía muchas ganas de conocer. Gracias a él me di cuenta de cómo funcionaban de verdad las grandes empresas, las cuales contienen procesos automáticos, también pude perfeccionar mis conocimientos de diseño, implementación, puesta a punto de producto, programación. Y sobre todo conocer cómo realizar un proyecto 100% real.

Además, fui capaz de solventar todos los problemas que me iban surgiendo a lo largo del desarrollo, indagando en la multitud de información que he encontrado y seleccionando lo más relevante.

5.3 Problemas encontrados y cómo se han solucionado

Por otro lado, todo este proceso no ha sido un camino de rosas, han surgido muchos problemas los cuales me ha costado mucho solucionar. Como, por ejemplo, no sabía cómo solucionar un problema en las cintas en la cual en simulación creía que estaba correctamente, pero una vez probado en real las cintas no se movían cuando tocaba, una vez solucionado me surgió otro problema con las cintas ya que después de un tiempo se detenían a priori sin motivo alguno, hasta que me di cuenta de que tenía un fallo grave dentro del código. Este error se debía a que en un momento dado estaba en dos etapas a la vez.

Otros problemas, que han ido surgiendo, es la implementación del HMI desde dentro del software de sysmac studio, ya que esto era completamente nuevo para mí y tuve que ver cómo funcionaba primero para después realizar correctamente la interfaz.

5.4 Aportaciones personales

A lo largo de este proyecto, me he dado cuenta de la importancia que tienen los PLC dentro de las empresas, así como la importancia de saber identificar errores y solucionarlos. Y no solo eso, sino que he ido adquiriendo nuevos conocimientos no adquiridos a lo largo de la carrera.

5.5 Futuras líneas de trabajo

Este proyecto tiene un gran potencial de cara al futuro, ya que la línea de producción siempre se puede mejorar, desde añadir contadores de eficiencia, como ajustando tiempos y velocidades o incluso expandiendo la planta. Por lo que los automatismos tienen el futuro cubierto.

6 Referencias

- [1] Información sobre que es un autómata [Autómata programable - Wikipedia, la enciclopedia libre](#) .(17/07/2023)
- [2] Página web de Vicky Foods [Vicky Foods - grupo multimarca: Dulcesol, BePlus y HnosJuan](#) (17/07/2023)
- [3] Historia de Vicky Foods [Nuestros comienzos ... - Dulcesol](#) (17/07/2023)
- [4] Información sobre el software de Sysmac Studio [Software Sysmac Studio | Omron](#) (18/07/2023)
- [5] Libro sobre PLC, Hardware y lógica cableada de Manuel Alvarez Pulido publicado en 2004 [CONTROLADORES LOGICOS | MANUEL ALVAREZ PULIDO | Casa del Libro](#) (19/07/2023)
- [6] Página web de post grado ingeniería [¿Qué es un autómata programable? | Escuela Postgrado Ingeniería \(postgradoingenieria.com\)](#).(19/07/2023)
- [7] Datasheet del HMI [Datasheet for NA5-9W001S-V1 Omron Optoelectronics | Octopart](#) (19/07/2023)
- [8] Datasheet del PLC [NJ501-1500 | OMRON, Europe](#) (19/07/2023)
- [9] Datasheet sobre la tarjeta de entradas [CJ1W-ID211 | OMRON, Europe](#) (19/07/2023)
- [10] Datasheet sobre la tarjeta de salidas [CJ1W-OD212 | OMRON, Europe](#) (19/07/2023)
- [11] Datasheet sobre el variador MX2 [MX2 | OMRON, España](#) (19/07/2023)
- [12] Datasheet sobre la fuente de alimentación [NJ-PA / PD NJ-Series Power Supply Unit/Specifications | OMRON Industrial Automation](#) (20/07/2023)
- [13] Libro sobre autómatas y cuadros eléctricos publicado en 2006 [AUTOMATISMOS Y CUADROS ELECTRICOS con ISBN 9788496334328 | Casa del Libro](#) (21/07/2023)
- [14] Lenguajes comunes en programación de PLC [Top5 Most Popular Types of PLC Programming Languages \(solisplc.com\)](#) (21/07/2023)
- [15] Visión general de la industria de los PLC [\(95\) An overview of the global PLC industry and its dynamics | by Ravi Kumar. | Products, Platforms, Business & Innovation in Industry 4.0/IIoT | Medium](#) (25/07/2023)
- [16] Información sobre la red ethercat [EtherCAT Technology Group | EtherCAT](#) (25/07/2023)
- [17] Que son y cómo funcionan las fotocélulas [Qué son las fotocélulas y para qué sirven \(safetyglobal.com\)](#) (25/07/2023)

[18] Información sobre los pistones neumáticos [Cilindro neumático - Pistón neumático - Tipos - Catálogo - Nelco Sistemas](#) (25/07/2023)

[19] Información sobre los sensores inductivos [【SENSOR INDUCTIVO】 Funcionamiento y Características 2023 \(sensormania.org\)](#) (26/07/2023)

[20] Metodología grafcet [automati2 \(uniovi.es\)](#) (26/07/2023)

[21] Lenguaje Ladder y cómo funciona [Microsoft Word - Diagrama Escalera.doc \(educacionurbana.com\)](#) (27/07/2023)

Anexo 1 Relación de trabajo con los ODS

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. Fin de la pobreza.				✘
ODS 2. Hambre cero.				✘
ODS 3. Salud y bienestar.				✘
ODS 4. Educación de calidad.	✘			
ODS 5. Igualdad de género.				✘
ODS 6. Agua limpia y saneamiento.				✘
ODS 7. Energía asequible y no contaminante.				✘
ODS 8. Trabajo decente y crecimiento económico.	✘			
ODS 9. Industria, innovación e infraestructuras.	✘			
ODS 10. Reducción de las desigualdades.				✘
ODS 11. Ciudades y comunidades sostenibles.				✘
ODS 12. Producción y consumo responsables.				✘
ODS 13. Acción por el clima.				✘
ODS 14. Vida submarina.				✘
ODS 15. Vida de ecosistemas terrestres.				✘
ODS 16. Paz, justicia e instituciones sólidas.				✘
ODS 17. Alianzas para lograr objetivos.				✘