



Universidad de Oviedo

Memoria del Trabajo Fin de Máster realizado por

JAVIER ALEJANDRO ALVAREZ CAREAGA

para la obtención del título de

Máster en Ingeniería de Automatización e Informática Industrial

**CARGA AUTOMÁTICA DE LA TOLVA DE UN CAMION
PARA EL TRANSPORTE DE MALTA**

FEBRERO-2019

AGRADECIMIENTOS

Quiero agradecer principalmente a mis padres Grover y Margot , que no solamente fue un sustentó económico, sino también moral durante toda mi estadía en España. También agradezco a mi hermana Micaela que fue un apoyo emocional para mí. A todo el personal de MEYME S.A. que aconsejo de manera técnica en él proyecto. Por último, quiero agradecer a todas las buenas personas que conocí en Gijón, desde mis compañeros de piso hasta el plantel docente de la universidad en especial a mi tutor Felipe Mateos Martín.

RESUMEN

Este Trabajo de Fin de Master “*Carga automática de la tolva de un camión para transporte de malta*” nace de la necesidad de automatizar su sistema de carga de malta a los camiones que transportan este producto desde sus silos de almacenamiento hasta las respectivas factorías. Este trabajo fue desarrollado esencialmente en cuatro fases.

En primer lugar, se tienen en cuenta los antecedentes describiendo a la empresa MEYME S.A. que es la que desarrollará el sistema, así como a la empresa Intermalta – Malteurop que es el cliente de este servicio. También se describen los objetivos del proyecto al mismo tiempo que sus alcances y lo más importante se describe cómo funciona el sistema en la actualidad.

En segundo lugar, se desarrolla la ingeniería del proyecto. Eligiendo los instrumentos para el posicionamiento del camión, medición de la tolva y carga del camión. A continuación, se realiza el modelo en GRAFCET, definiendo sus modos de funcionamiento y planteamiento de la solución. También se realiza la programación con una prueba previa en Visual Studio y programando en Tia Portal de Siemens describiendo el hardware utilizado y se realizan los planos eléctricos. Posteriormente dan detalles de la migración al sistema de control distribuido PCS7 también de Siemens.

En tercer lugar, se elabora en la memoria la planificación y el presupuesto de todo este trabajo realizado, que contempla todos los instrumentos, así como el costo del personal necesario. Por último se sugieren las futuras ampliaciones, así como las conclusiones.

INDICE DE CONTENIDOS

1.	INTRODUCCIÓN	1
1.1.	Datos generales del Trabajo Fin de Máster	1
1.2.	Antecedentes	1
1.2.1.	MEYME S.A.	1
1.2.2.	Malteurop	3
1.2.3.	Instalación previa	3
1.2.4.	Sistema programado en PCS7 actual	8
1.3.	Tipos de tolvas	9
1.4.	Objetivos y tareas a realizar	9
2.	ELECCIÓN DE LA INSTRUMENTACIÓN	11
2.1.	Introducción	11
2.2.	Instrumentación para posicionamiento de camión	12
2.3.	Instrumentación para dimensionamiento de la tolva	13
2.3.1.	Escáner Laser 2D serie R2000	14
2.3.2.	Escáner de LED de haces múltiples serie R2100	14
2.4.	Instrumentación para carga de camión	17
2.5.	Sistema propuesto	18
3.	MODELADO GRAFCET Y MÉTODO DE CÁLCULO	20
3.1.	GRAFCET General	20
3.1.1.	GRAFCET de dimensionamiento de tolva	21
3.2.	GRAFCET de dosificación automática	22
3.3.	Método de obtención de medidas	24
3.3.1.	Requisitos para el uso del escáner de haces múltiples	25
4.	PRUEBAS DE FUNCIONAMIENTO	27
4.1.1.	Pruebas de funcionamiento del escáner R2100	27
5.	ARQUITECTURA HARDWARE DEL SISTEMA	32
5.1.	Topología de red	32
5.1.	Módulo CPU 1511-1 PN	33
5.2.	Remota interface Profinet IM 155-6 PN HF	33
5.3.	Módulo de Comunicación CM PtP	34

5.4.	Módulo de entradas analógicas AI 4xI 2-/4-wire ST	35
6.	PROGRAMACIÓN EN TIA PORTAL	36
6.1.	Entradas y salidas del sistema	36
6.2.	Identificación de ficheros realizados	37
6.3.	Organización de programa de control	38
6.4.	Bloques de Organización implementados	39
6.5.	Explicación de bloques de datos	40
6.6.	Explicación de funciones y bloques funcionales	42
6.6.1.	Comunicación escáner [FC1]	42
6.6.2.	Adaptación sensor [FB1]	44
6.6.3.	Sensor S1 [FC7]	45
6.6.4.	Altura [FB6]	47
6.6.5.	Ancho [FB4]	49
6.6.6.	Longitud [FB3]	50
6.6.7.	Volumen [FC4]	53
6.6.8.	Movimiento de motor [FB8]	57
6.6.9.	Gestión de Volumen [FC11]	58
6.6.10.	Sensor radar [FC10]	59
6.6.11.	Bascula [FC13] [FC14]	60
6.6.12.	Lector de matrícula [FC15] [FC20]	60
6.6.13.	Dosificado Automático [FB5]	60
6.6.14.	Entrada de camión [FB2]	62
6.6.15.	Salida de camión [FB7]	63
6.6.16.	Principal [FC12]	64
6.6.17.	Salidas [FC17]	66
7.	MIGRACIÓN A PCS7	68
6.7.	Generación de fuentes SCL en Tia Portal	68
6.8.	Modificación de código	69
6.9.	Asignación de número de FB y FC	71
6.10.	Programación de FB y FC no migrados	72
6.10.1.	Migración FC Comunicación escáner	73

6.10.2.	Migración FB “ <i>Gestion_Volumen</i> ”	74
6.10.3.	Migración FC12 “ <i>Principal</i> ”	76
7.	ENSAYOS Y RESULTADOS	78
7.1.	Introducción	78
7.2.	Visualización de resultados	79
7.3.	Resultados de las pruebas	83
7.3.1.	Pruebas de volumen	83
7.3.2.	Prueba de medida del ángulo	85
8.	PLANIFICACION	87
9.	PRESUPUESTO	88
10.	CONCLUSIONES	89
11.	FUTURAS AMPLIACIONES	90
12.	BIBLIOGRAFÍA	91
13.	ANEXO: PLANOS ELÉCTRICOS DE LA INSTRUMENTACIÓN	93

INDICE DE FIGURAS

Figura 1: Logo Meyme S.A.	2
Figura 2: Logo Malteurop	3
Figura 3: Esquema de proceso: Dosificación de malta	4
Figura 4: Botonera de para control de movimiento longitudinal	5
Figura 5: Botonera de control de Silos	6
Figura 6: Sistema mecánico de dosificado de tolva de camión	7
Figura 7: Visualización general de la planta de San Adrián	9
Figura 8: Tipos de tolvas	9
Figura 9: Medidas de camión estándar [5]	11
Figura 10: Medidas de camión [6]	12
Figura 11: Sensor fotoeléctrico [7]	13
Figura 12: Escáner Laser 2D serie R2000 [8]	14
Figura 13: Escáner de LED de haces múltiples	15
Figura 14: Encóder con sisga [10]	16
Figura 15: Sistema de medición propuesto	16
Figura 16: Sensor óptico O1D100	17
Figura 17: VEGAPULS 67 [12]	17
Figura 18: Posicionamiento de sensores	18
Figura 19: Representación de la ubicación de la instrumentación	19
Figura 20: Grafcet general	20
Figura 21: Grafcet para el dimensionamiento de tolva	21
Figura 22: Grafcet de dosificado automático	23
Figura 23: Método para obtención de altura y longitud	24

Figura 24: Método para obtención del ancho	25
Figura 25: Configuración de puerto serie	25
Figura 26: Versión de Visual Studio	27
Figura 27: Interfaz de programa prueba	28
Figura 28: Método de evento Click en pulsador Enviar	28
Figura 29: Método de evento “DataRecieved”	29
Figura 30: Concatenado de bytes	30
Figura 31: Ciclo for para concatenado de bytes	30
Figura 32: Interfaz de prueba	31
Figura 33: Configuración hardware equipos de Siemens	32
Figura 34: Configuración de red	33
Figura 35: Módulo CPU 1511-1 PN [14]	33
Figura 36: Modulo de Interface IM 155-6 PN/2 HF [15]	34
Figura 37: Módulo de comunicación CM PtP [16]	34
Figura 38: Módulo de Entradas Analógicas [17]	35
Figura 39: Versión de Tia Portal y WinCC	36
Figura 40: Entradas digitales y analógicas del sistema	36
Figura 41: Salidas del sistema	37
Figura 42: Identificación de ficheros	38
Figura 43: Estructura de llamadas	39
Figura 44: Código del OB Ciclo_100ms	39
Figura 45: Código de OB “Ciclo_200ms”	40
Figura 46: Código de OB “Reset_Interrupt”	40
Figura 47: DB5 “WinCC”	41

Figura 48:DB4 Req_Comunicación	42
Figura 49: Llamada al bloque funcional “Port_Config”	42
Figura 50: Bloque Send_P2P	43
Figura 51: Bloque Recieve_P2P	43
Figura 52: Entradas y salidas de Adaptación_sensor	44
Figura 53: Código para obtención de distancias reales	44
Figura 54: Código para la obtención de coordenadas cartesianas.	45
Figura 55: Código para llenado de historial	45
Figura 56: Código de función Sensor_S1	46
Figura 57: Representación de la recta del sensor óptico	46
Figura 58: Entradas y salidas “Altura”	47
Figura 59: Código para obtención de limites	48
Figura 60: Código para la obtención de señal fuera de rango	48
Figura 61: Código para la obtención de altura	48
Figura 62: Entradas y salidas “Ancho”	49
Figura 63: Código para detección de esquina izquierda	49
Figura 64: Código para llenado de buffer de anchuras	49
Figura 65: Código para obtención de altura total	50
Figura 66: Entradas y salidas de “Longitud”	50
Figura 67: Código para obtención de longitud inicial	51
Figura 68: Código de cálculo de pendiente	52
Figura 69: Código de ángulo y longitud obtenida	53
Figura 70: Entradas y Salidas “Volumen”	53
Figura 71: Código de cálculo de volumen de un ortoedro	54

Figura 72: Código de cálculo de volumen con un chaflán	55
Figura 73: Modelo para cálculo de volumen para tolva con un chaflán	55
Figura 74: Código de cálculo de volumen con un chaflán	56
Figura 75: Modelo para cálculo de volumen para tolva con dos chaflanes	56
Figura 76: Entradas y salidas “Movimiento_de_motor”	57
Figura 77: Código para movimiento de motor	58
Figura 78: Código Gestión Volumen	59
Figura 79: Entradas y salidas “Dosificado automático”	60
Figura 80: Código de dosificado automático	61
Figura 81: Entradas y salidas “Entrada_de_camion”	62
Figura 82: Código de entrada de camión	62
Figura 83: Entradas y salidas “Salida_de_camion”	63
Figura 84: Código de salida de camión	63
Figura 85: Primera parte de la función Principal	64
Figura 86: Segunda parte de función principal	65
Figura 87: Código de salidas	66
Figura 88: Ejemplo de generación de fuente	68
Figura 89: Cambios en programación de SCL	70
Figura 90: Programación FUP de Comunicación_Escaner	73
Figura 91: Programación en CFC para Gestión Volumen	75
Figura 92: CFC de programa principal	77
Figura 93: Organización del entorno de pruebas	78
Figura 94: Pruebas en ambiente controlado.	79
Figura 95: Pantalla de proceso principal	80

Figura 96: Pantalla de escáner de haces múltiples	81
Figura 97. Pantalla de perfil de tolva	82
Figura 98: Pantalla de resultados	82
Figura 99: Aclaración de entorno de pruebas	83
Figura 100: Tiempo por tareas	87

INDICE DE TABLAS

Tabla 1: Datos generales del Trabajo Fin de Máster	1
Tabla 2: Trama para requisito de distancias y ecos	26
Tabla 3: Trama de respuesta de distancias y ecos	26
Tabla 4: Entradas por comunicación del sistema	37
Tabla 5: Generación de fuentes	69
Tabla 6: Tabla de número de función o bloque de función	72
Tabla 7: Medidas de entorno controlado	84
Tabla 8: Resultados de pruebas de repetibilidad	85
Tabla 9: Tabla Resumen	85
Tabla 10. Tabla de pruebas de Ángulos	86
Tabla 11: Presupuesto estimado	88

1. INTRODUCCIÓN

1.1. Datos generales del Trabajo Fin de Máster

Datos Generales del Trabajo Fin de Máster	
<i>Título en español</i>	Carga automática de la tolva de un camión para transporte de malta
<i>Título (en inglés)</i>	Automatic loading of the hopper of a truck for transport of malta
<i>Autor del proyecto</i>	Javier Alejandro Alvarez Careaga
<i>Tutor del proyecto</i>	Felipe Mateos Martín
<i>Titulación</i>	Máster de Ingeniería de Automatización e Informática Industrial
<i>Centro</i>	Escuela Politécnica de Ingeniería de Gijón
<i>Fecha de presentación</i>	Febrero-2019

Tabla 1: Datos generales del Trabajo Fin de Máster

1.2. Antecedentes

El presente proyecto nace de la necesidad de la empresa Intermalta, de automatizar su sistema de carga de malta en las tolvas de camión, que transportan este producto, desde sus silos de almacenamiento hasta las respectivas factorías, de una manera rápida y bien cuantificada.

La empresa asturiana de Montajes y Mantenimientos Eléctricos S.A. (MEYME S.A.) es la encargada de diseñar el sistema automatizado para la empresa Intermalta que es parte del grupo financiero Malteurop.

1.2.1. MEYME S.A.

La empresa MEYME S.A. es una empresa de ingeniería dedicada al diseño y producción de sistemas de control eléctrico e instrumentación para procesos industriales de baja, media y alta tensión y al mantenimiento eléctrico de instalaciones industriales. [1]



Figura 1: Logo Meyme S.A.

En su actividad en el ámbito de la ingeniería eléctrica se incluyen proyectos “llave en mano”, diseño y cálculo de procesos integrales, automatización y regulación de procesos en instalaciones industriales, diseño de sistemas de mando y control, instrumentación, fabricación de armarios eléctricos, programación de autómatas, software industrial, y sistemas de control distribuido.

Los sectores en los que MEYME S.A posee clientes son:

- Industria alimentaria
- Industria química
- Industria energética

Hay que hacer notar que MEYME S.A. tiene una gran trayectoria en la industria de la maltería, siendo de las pocas empresas que ha llevado a cabo la automatización integral de una maltería de tipo Laussman (fabricante alemán) [2]. Debido a esta gran trayectoria en la industria de la malta, se han realizado múltiples proyectos en la industria, entre las que figuran empresas como:

- Intermalta San Adrián (Navarra)
- Maltiberica (Portugal)
- Intermalta (Sevilla)
- Intermalta (Albacete)
- The Swaen Kloosterzande (Países Bajos)
- Malteurop Langerrigen (Alemania)

Consiguientemente el proyecto a realizar ha sido sugerido por Intermalta de San Adrián, el cual será explicado durante el desarrollo de la presente memoria.

1.2.2. Malteurop

Malteurop es un grupo empresarial que proyecta convertirse en el integrador de la cadena de producción desde la cebada a la cerveza. Malteurop es el primer maltero mundial, el grupo Malteurop posee actualmente una capacidad de producción mayor a 2.200.000 toneladas de malta. Está presente en 14 países en Europa, Norteamérica, Oceanía y Asia, con veintisiete parques industriales y 1.183 colaboradores. [3]



Figura 2: Logo Malteurop

Malteurop se convierte en el accionista mayoritario de referencia de Intermalta en 1991. Es una etapa importante en el desarrollo de Malteurop la adquisición de esta participación en la maltería de San Adrián (Navarra).

La maltería de San Adrián es de tipo Lausmann, que tiene el modo de traslado de malta por tren, camión y contenedor. Siendo el objetivo de nuestro proyecto el traslado por camión.

1.2.3. Instalación previa

Antes de adentrarnos en el proyecto que se realizó, se hará una breve descripción del sistema actual, que se encuentra instalado en Intermalta San Adrián (Navarra).

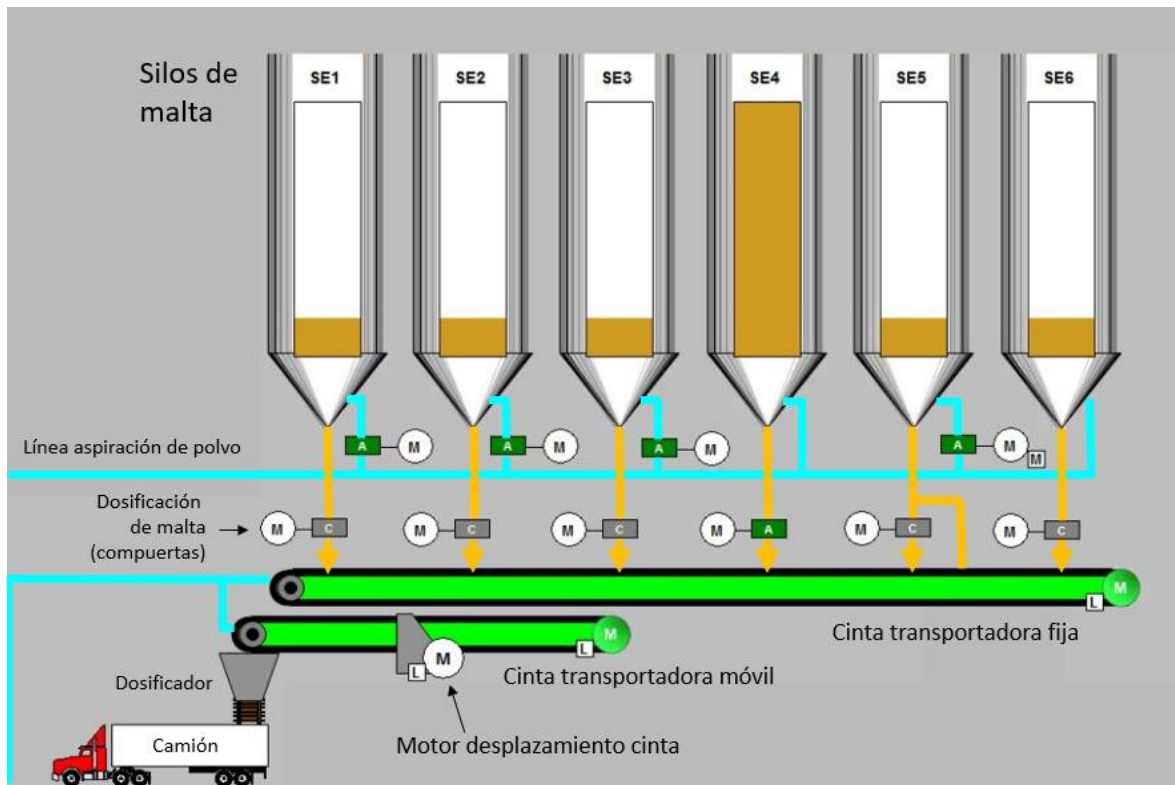


Figura 3: Esquema de proceso: Dosificación de malta

Como se puede ver en la figura 3, el sistema actual consta de cuatro partes principales, los silos de malta donde se almacena la malta lista para ser transportada; hay que indicar que existen seis tipos de malta, donde a través de un selector se escogerá que tipo de producto se desea, posteriormente se dispone de varios motores de aspiración que se encargan de aspirar el polvo que genera la malta al caer. Asimismo, pueden apreciarse los motores para abrir y cerrar las compuertas de los silos y que puedan dosificar material sobre la cinta transportadora fija, y consecutivamente la cinta transportadora móvil que vierte la malta en el camión.

Es importante considerar que el sistema actual se compone de una parte mecánica y otra de control electrónico. La parte mecánica consiste en las cintas transportadoras, la apertura y cierre de silo y el mecanismo de movimiento longitudinal. Este mecanismo es de piñón cremallera, accionado por un motor anclado en la mitad de la cinta transportadora. También se puede indicar que existe un final de carrera de avance de la cinta móvil para evitar que el mecanismo de movimiento longitudinal alcance las posiciones límite en funcionamiento, lo cual podría deteriorar la máquina.

El sistema de control electrónico se compone de un controlador programable del cual se carece de información pero que simplemente gestiona los accionamientos en base a la actuación en modo manual de dos sistemas de botoneras. Una botonera de cinta se utiliza

para maniobrar el movimiento longitudinal de la cinta transportadora y la otra botonera de silos para controlar:

- la apertura y cierre de los silos,
- el sistema de aspiración que se utiliza para aspirar el polvo del ambiente ya que este llega a ser explosivo y
- la cinta transportadora fija.

Además de estos sistemas se tiene una balanza para evitar que los camiones se pasen del peso establecido por ley, para el transporte de vías públicas.

Como se puede ver en la figura 4, la primera botonera de cinta dispone de tres pulsadores, (dos de color verde y uno de color rojo), un selector (de color negro) y una seta de emergencia. De estos objetos solamente se utilizan el selector y la seta de emergencia. El selector se utiliza para el movimiento longitudinal de la cinta transportadora, (así cuando el operario mueva el selector a la izquierda la cinta transportadora se moverá atrás y si el mismo mueve el selector a la derecha la cinta se moverá adelante), y la seta de emergencia que se utiliza en casos de presentarse problemas en el sistema. La cinta detiene el movimiento al llegar al final de carrera correspondiente.



Figura 4: Botonera de para control de movimiento longitudinal

Como se puede ver en la figura 5, la botonera de silos consiste en cuatro selectores (ubicados en la parte superior y de color negro) y ocho pulsadores, dos en la parte superior y otros seis en la inferior. Los selectores consisten en:

- Un selector de automático y manual, que se utiliza para que la maquina funcione de modo manual (con los pulsadores) o automático (con un programa cargado en el PLC),
- Un selector de 6 posiciones para escoger el tipo de malta que se quiere dosificar,

- Un selector de enclavamiento y libre, en el cual enclava ambas cintas transportadoras, para que una cinta no funcione sin que la otra esté funcionando; por otro lado, en libre se permite que ambas cintas transportadoras se activen independientemente del estado de la otra cinta.
- El último selector de arriba-centro que no posee un propósito en específico.

Los pulsadores Abrir y Cerrar de la parte superior se utilizan para activar estas operaciones en modo manual.



Figura 5: Botonera de control de Silos

En la parte inferior se tiene los pulsadores para activar las cintas transportadoras y para activar el sistema de aspiración de polvo.

En la figura 6 se observan, la cinta transportadora encargada de transportar la malta y verterla en el camión a través del dosificador; también se observan los raíles, sobre los cuales la cinta transportadora se mueva longitudinalmente con el objetivo de dosificar material de forma balanceada. Este movimiento se gobierna manualmente a través del selector mostrado en la figura 4, y también se dispone del sistema de aspiración para recoger el polvo que desprende la malta.



Figura 6: Sistema mecánico de dosificado de tolva de camión

Actualmente cuando un camión llega para recoger malta, este se pone por debajo del dosificador y el propio conductor va hacia el tablero de control del silo, selecciona el tipo de malta que se requiere, luego abre el silo y por último activa las cintas transportadoras.

Esto conlleva dos problemas:

- El primero que se tiende a desbordar malta de la tolva, ya que la botonera se encuentra a la altura del piso, por lo tanto, el conductor no tiene una vista de la boca de la tolva y este desconoce cuándo la tolva se encuentra llena.
- El segundo problema es que al no haber un operario controlando el proceso, el conductor tiende a llenar al máximo la tolva, ya que este tiene más beneficio al transportar más malta, y como resultado no se fija por el peso permitido por ley.

Dentro de la empresa, el operario solamente es contactado cuando existe algún problema, por lo tanto, el operario no tiene un contacto directo con el proceso de dosificado que realiza el conductor. Consecuentemente se ve necesario un sistema automático que controle todo el proceso.

1.2.4. Sistema programado en PCS7 actual

Actualmente la maltería de San Adrián tiene programado en el sistema de control distribuido PCS7 de Siemens, casi todo el sistema productivo de la malta.

El sistema productivo en la maltería de San Adrián, en su mayoría está programado en el sistema de control distribuido PCS7 de Siemens como se explica a continuación:

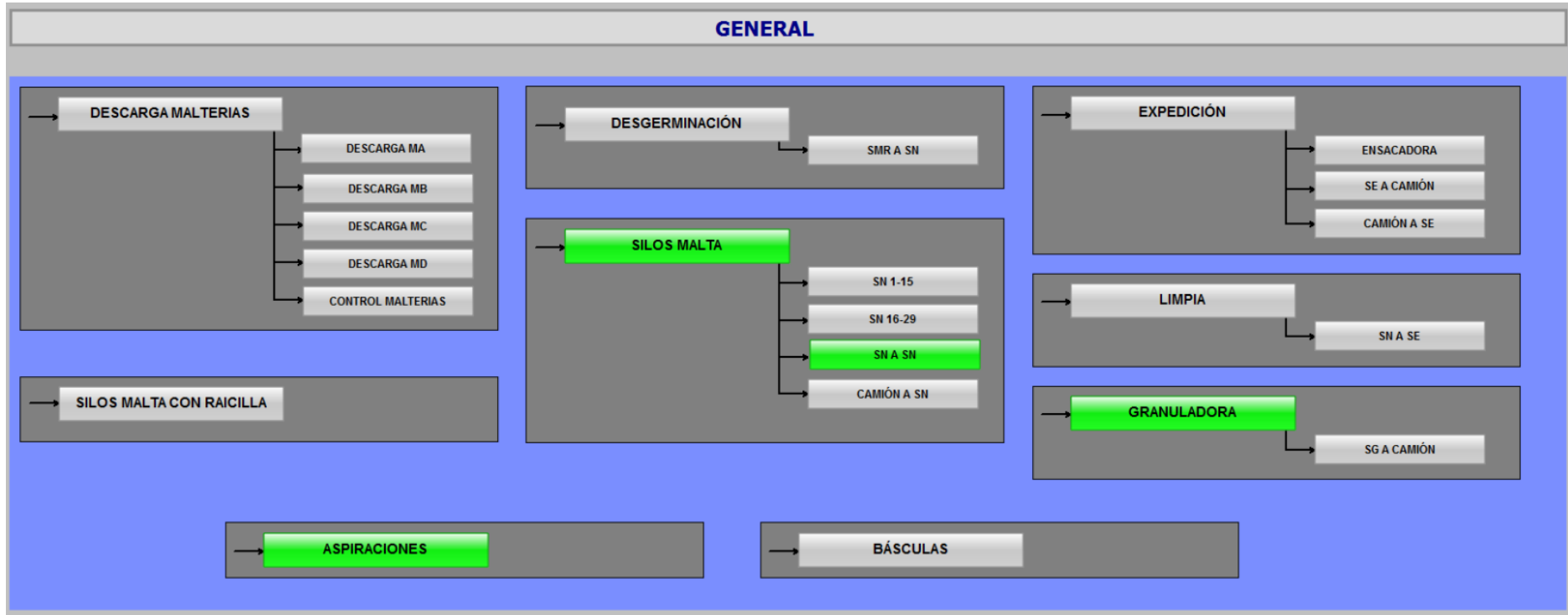


Figura 7: Visualización general de la planta de San Adrián

Como se puede ver en la figura 7, se presenta un vistazo general a través de esta pantalla del sistema PCS7 implantado, los pulsadores para acceso a todas las partes del sistema de producción de malta. En verde aparecen aquellas zonas que en cada momento se encuentran activas.

El presente proyecto se realizará en la zona de Silos de malta.

1.3. Tipos de tolvas

Hay que destacar que, para un cálculo correcto del volumen de la tolva de camión, se tiene que conocer a priori los tipos de tolva que llegan a la maltería de San Adrián. Como se muestra en la figura 8 se trabaja con tres tipos de tolvas, que son las únicas presentes en la factoría para la carga de malta.



Figura 8: Tipos de tolvas

La tolva en forma de ortoedro, que para calcular su volumen se utiliza la fórmula de volumen de un ortoedro [4]. La tolva ortoedro con un chaflán se descompondrá en un ortoedro y un prisma triangular. Finalmente, para calcular el volumen de una tolva con dos chaflanes se usa la fórmula del ortoedro más los dos prismas triangulares.

1.4. Objetivos y tareas a realizar

Con las premisas anteriormente expuestas, se pretende automatizar este sistema de llenado de la tolva de modo que se requiera una intervención mínima por parte del operario, siendo también necesaria su integración en el sistema de control distribuido PCS7 de Siemens que gobierna el conjunto del proceso productivo de la malta, su almacenamiento y dosificación.

Para llevar a cabo estos objetivos generales se requiere realizar un conjunto de tareas que identifican el alcance de este trabajo y entre las que se encuentran las siguientes:

- Estudio detallado de la instalación de partida, tanto del sistema de cinta transportadora como del sistema de control distribuido.

- Especificación de requisitos de la instalación desde el punto de vista del sistema de carga, el usuario y la integración en el control general de la planta.
- Selección de la instrumentación adecuada (sensores y actuadores), para la determinación de la forma y volumen de la tolva, el posicionamiento de la cinta, la altura del producto durante la carga, el peso, etc...
- Diseño de la integración hardware y software del sistema de carga en el control distribuido PCS7 de Siemens.
- Desarrollo de esquemas eléctricos de mando y de fuerza para conexión de señales al controlador y los accionamientos.
- Realización de programas de control y pantallas de explotación y supervisión en el control distribuido PCS7 de Siemens para integración de esta nueva instalación. Inicialmente se podrá desarrollar en Tia Portal para posteriormente trasladar al control distribuido.

2. ELECCIÓN DE LA INSTRUMENTACIÓN

2.1. Introducción

En esta etapa de selección de la instrumentación se pudo hacer un análisis de todos los sensores que cumplieran con el cometido buscado. Los condicionantes se corresponden con las medidas características de las tolvas de camión y de las condiciones del ambiente en el que se desarrolla el proceso.

Con relación las medidas de la tolva, se debe tener en cuenta que los sensores medirán la longitud, el ancho y la altura de la tolva del camión, para calcular su volumen. Es así que estos sensores deben tener la capacidad de medida de esas distancias, con la máxima precisión disponible en el mercado.

Los condicionantes debidos al entorno de trabajo, pueden llegar a afectar al funcionamiento correcto de los sensores, por lo tanto, es importante en la elección de la instrumentación tener en cuenta el ambiente en el cual se utiliza, al poder existir polvo o humedad, que pueden afectar el funcionamiento del sensor.

La instrumentación inicial deseada debe permitir que el camión se encuentre en posición para su pesaje, bastarán dos detectores de presencia a distancia equidistantes del centro de la báscula y a una altura tal, que cubra el abanico de tolvas de los camiones que realizan las labores de carga en la factoría.

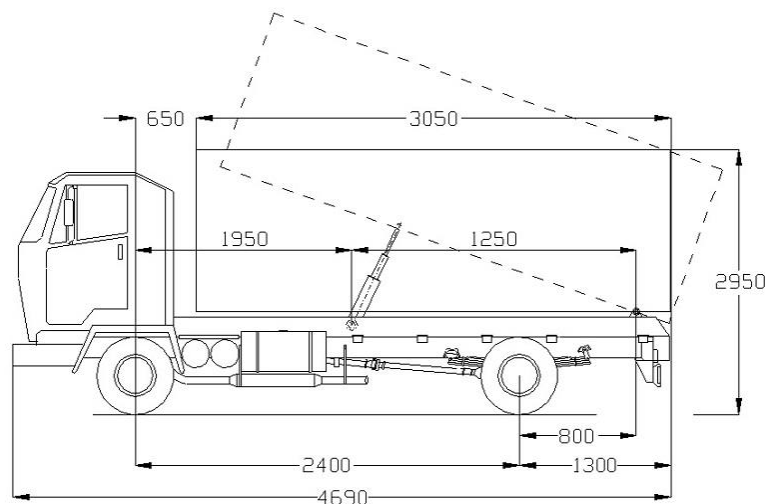


Figura 9: Medidas de camión estándar [5]

Como se puede ver en la figura 9, las medidas aproximadas de una tolva de camión son de 3,05 metros, con una altura aproximada de 2 metros.

Como una primera aproximación para la medida de dimensiones de la tolva que permita calcular su volumen, se decidió un mecanismo que entre en la tolva con la instrumentación adecuada y que se explicará en este capítulo.

Además, se necesita de un instrumento resistente al polvo que permita saber cuánto material se cargó dentro de la tolva, y si esta zona se encuentra llena de malta, activar el desplazamiento de la cinta transportadora para una mejor distribución de la carga.

Teniendo en cuenta estos requerimientos se hizo la elección, de cada uno de los instrumentos. Durante la elección de la instrumentación, la empresa aconsejó las siguientes marcas:

- Pepperl Fuchs
- Siemens
- VEGA
- IFM Electronic
- Schneider Electric

2.2. Instrumentación para posicionamiento de camión

Para el posicionamiento del camión se necesitaba de dos sensores de presencia, con un alcance de aproximado del ancho del camión. Se investigó sobre los diferentes tipos de camión, en la figura 10 se puede ver un tipo de camión característico.

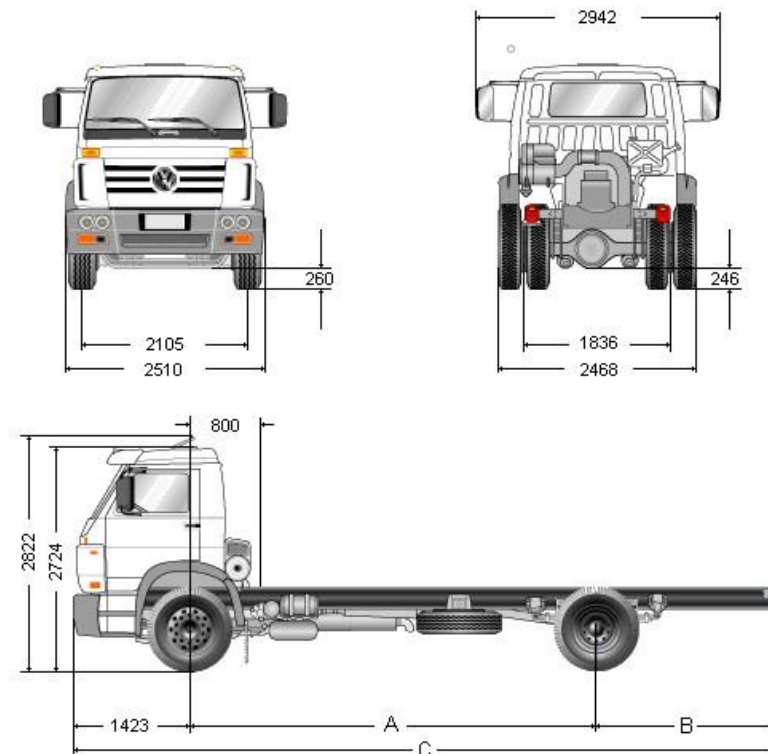


Figura 10: Medidas de camión [6]

En la Figura 10, se puede observar las dimensiones del ancho del camión son de 2,5 metros de ancho, por lo tanto, la capacidad de medida del sensor tiene que ser mayor al ancho anteriormente mencionado. Consultado con la empresa se escogió un sensor, acorde a esta necesidad, que es el que se muestra en la Figura 11. Es un sensor de la marca Schneider Electric de tipo fotoeléctrico, con salida relé con un contacto conmutado, y un alcance nominal de 5 metros, que es más que suficiente para la aplicación que se necesita.

Hoja de características del producto
Características

XUK9ARCNL2
sensor fotoeléctrico - XUK - polarizado - Sn 5 m -
24..240 VCA/CC - cabo de 2 m



Figura 11: Sensor fotoeléctrico [7]

Este detector es capaz de funcionar en todos los sistemas estándar de detección fotoeléctrica:

- Sin accesorio: proximidad, y proximidad con borrado de plano posterior.
- Con accesorio (reflector o emisor XUZC50): Réflex polarizado y barrera.

Realizada una alineación adecuada puede ejecutarse el procedimiento de *Aprendizaje Automático del Entorno*, eliminando primeramente el objeto a detectar del plano y tras el ajuste proceder a colocar el objeto a detectar y verificar su correcto funcionamiento.

2.3. Instrumentación para dimensionamiento de la tolva

Una de las decisiones más complejas fue decidir qué instrumento se utilizaría para el dimensionamiento de la tolva, pues no se tenía conocimiento de qué sensor se debía utilizar, además de cuantos se necesitarían.

Para obtener el volumen de la tolva el sensor o sensores deben medir el ancho, alto y la longitud, y además tener en cuenta que la longitud puede variar si se tiene uno o dos chaflanes. Asimismo, los sensores deben tener un alcance de hasta 10 metros para el largo, 2,5 metros para el ancho y 2 metros para el alto según las dimensiones máximas de las tolvas de camión.

Teniendo en cuenta estos requerimientos, se hizo una primera aproximación con sensores que nos permitan leer distancias en varias direcciones.

2.3.1. Escáner Laser 2D serie R2000

En la primera aproximación se decidió el Escáner laser 2D serie R2000, que venía siendo la mejor elección, según apreciación propia.

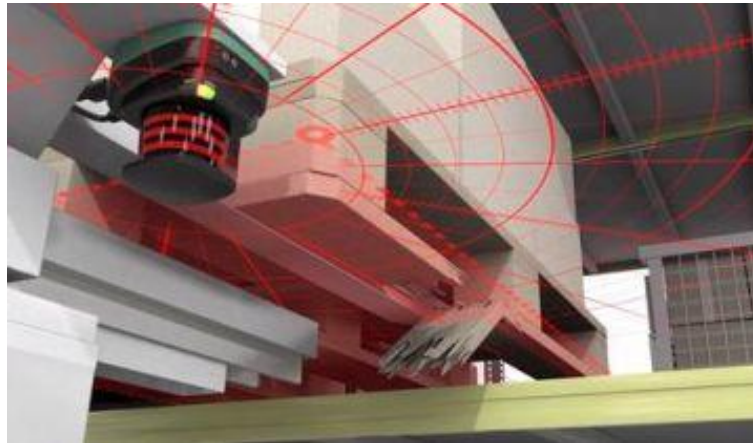


Figura 12: Escáner Laser 2D serie R2000 [8]

Como se puede ver en la figura 12, el sensor permite una medición en 360 grados y con alta precisión, por lo tanto, si se quisiera medir el volumen de la tolva, se podrían tomar las medidas de longitud y ancho con el presente sensor. Para medir la altura se utilizaría otro sensor a láser.

Entonces el sistema para la medición de volumen de la tolva, consistiría en un sistema móvil, que a través de un mecanismo bajaría los sensores dentro de la tolva, luego obtendría las medidas de longitud y ancho, además de medir la altura mediante el sensor láser para su posterior cálculo de volumen.

Sin embargo, tras consulta con el proveedor Pepperl Fuchs, recomendó que se tome en cuenta la comunicación del sensor con el PLC. Como se sabía con anterioridad, el sensor poseía una conexión Ethernet, no obstante, el comercial nos explicó que el sensor tenía una tasa alta de transmisión de datos, 54000 mediciones por segundo y este se transmitía por Fast Ethernet.

Se consultó por algún modulo que soporte Fast Ethernet, para el PLC Siemens, indicando su disponibilidad, pero a un precio muy elevado para que sea accesible. Como consecuencia se consultó al proveedor Pepperl Fuchs por otra solución, quien recomendó el sensor de la serie R2100.

2.3.2. Escáner de LED de haces múltiples serie R2100

Como indica su nombre el escáner LED de haces múltiples; es un sensor que genera once haces de luz en forma de abanico con una separación de 8 grados cada uno dando un barrido total de 88 grados.

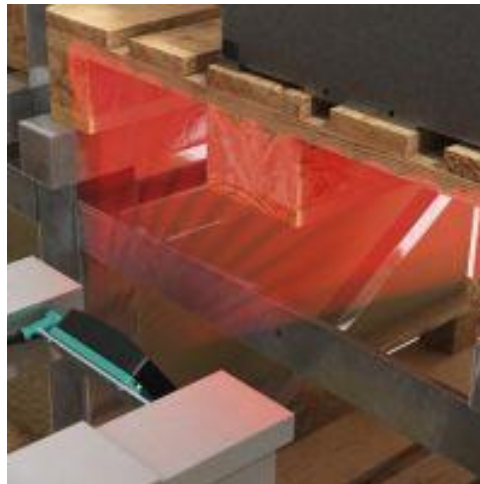


Figura 13: Escáner de LED de haces múltiples

A diferencia del anterior sensor de serie R2000, la interface de comunicación para este sensor R2100 [9] está basada en la interface RS-232, por lo tanto, lo hace propicio para una conexión con un PLC. También hay que tener en cuenta que la comunicación por RS-232 no soporta grandes distancias así que será adecuado poner una pasarela de conversión de señal de RS-232 a RS-485.

La colocación de este sensor R2100 elegido ha de ser tal que el conjunto de haces se encuentre en un plano perpendicular al largo de la tolva lo que permite obtener medidas del alto y el ancho. Es por tanto necesario otro instrumento adicional, para obtener la longitud de la tolva.

2.3.2.1. Encóder con sisga ECA10TL-PROFINET

Como el sistema actual de carga contempla una cinta transportadora, que se mueve de forma longitudinal a la tolva, se propuso que en esta se montará un encóder lineal para obtener la distancia de recorrido. De esta forma se haría un mapeo de toda la tolva del camión.

Para la medición de la longitud de la tolva primero se escogió el encóder con cable de arrastre (sisga).

Como se puede ver en la figura 14, el encóder con cable de arrastre (sisga) es de la marca Pepperl Fuchs, con una longitud de medición de hasta 10 metros, y una resolución de 25 bits y en comunicación Profinet para transmitir datos al PLC. Es un instrumento suficientemente preciso para la aplicación planteada.



Figura 14: Encóder con sisga [10]

Como se muestra en la figura 15, el sistema consistiría en el encóder fijo y la cinta transportadora que se mueve longitudinalmente a la tolva. Consecuentemente el escáner mediría el ancho y alto de la tolva, mientras que para medir la longitud, el escáner detectaría la entrada de la tolva, dando la señal al encóder que empiece a medir. Cuando el escáner detecte el final de la tolva, el encóder recibiría la señal de salida de la tolva, terminando la medición de la longitud de la tolva

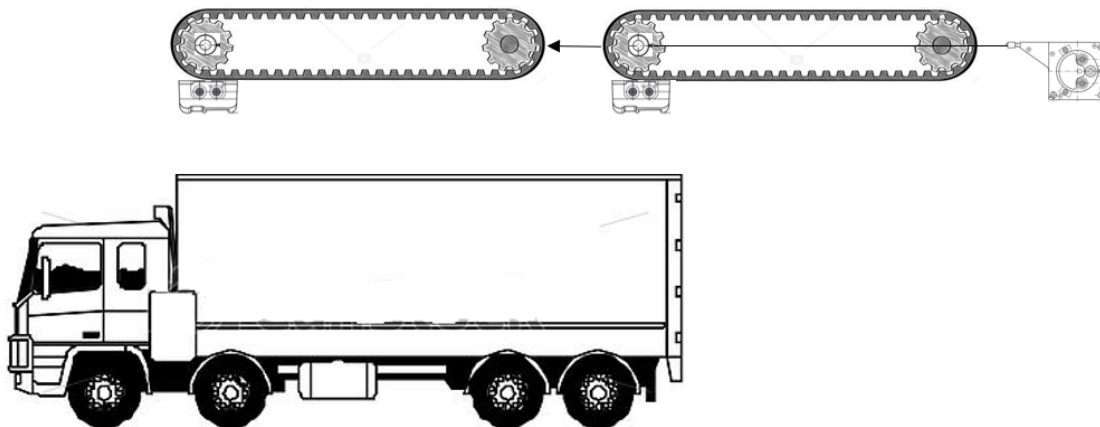


Figura 15: Sistema de medición propuesto

La aplicación ya estaba definida de esta forma, no obstante al hacer la consulta del precio del encóder era demasiado caro, en torno a 3.000€, así que se optó por otro instrumento.

2.3.2.2. Sensor de distancia óptico O1D100

A continuación, se optó con un sistema híbrido en el cual se usará, el escáner de múltiples haces R2100, para medir el ancho y alto de la tolva y se utilizarán dos sensores ópticos para medir la longitud de la tolva. Para este objetivo se escogió el sensor óptico O1D100 de IFM Electronic.



Figura 16: Sensor óptico O1D100

El presente sensor tiene un alcance de hasta 10 metros con salida analógica de 4 a 20 mA, que es más que suficiente para medir el largo de cualquier tolva, con una repetibilidad de ± 14 mm, esta repetibilidad se debe a la gran distancia alcanzada por el sensor. [11]

Por tanto con un sensor de haces múltiples R2100 de Pepperl Fuchs y dos unidades del sensor O1D100 de IFM Electronic es posible obtener las dimensiones suficientemente precisas de la tolva del camión.

2.4. Instrumentación para carga de camión

Para la instrumentación que aporte una medida de la carga de malta que se está realizando en la tolva del camión por medio de la cinta, era necesario un dispositivo que fuese resistente al polvo y que ofrezca una medida del llenado y por tanto de la distancia del sensor a la cinta en cada momento. Si la altura alcanzada por el material en la tolva supera cierto valor, la cinta se moverá delante quedando el producto más homogéneamente distribuido y evitando desbordamientos.

Teniendo en cuenta estos requerimientos, se optó por un sensor tipo radar al que no le afectan perturbaciones por polvo; el sensor escogido es el modelo VEGAPULS 67 de VEGA que se muestra en la figura 17.



Figura 17: VEGAPULS 67 [12]

El presente sensor es una solución de bajo coste que permite la medición del nivel de sólido a granel con un rango de medición de 15 metros, con una salida analógica de 4...20 mA, precisión de 2mm y adecuado en cuanto al tipo de protección.

2.5. Sistema propuesto

Una vez decididos los instrumentos a utilizar, se hizo una propuesta de configuración del mecanismo que entraría en la tolva conteniendo los sensores, y que se diseñó de la siguiente manera:

El posicionamiento de los sensores, se realizó de forma que se puedan determinar diferentes medidas con el escáner y con los sensores de distancia fotoeléctricos, Utilizando el escáner se calcula el ancho y la altura, mientras que con los sensores de distancia se calcula la longitud y el ángulo del chaflán de la tolva.

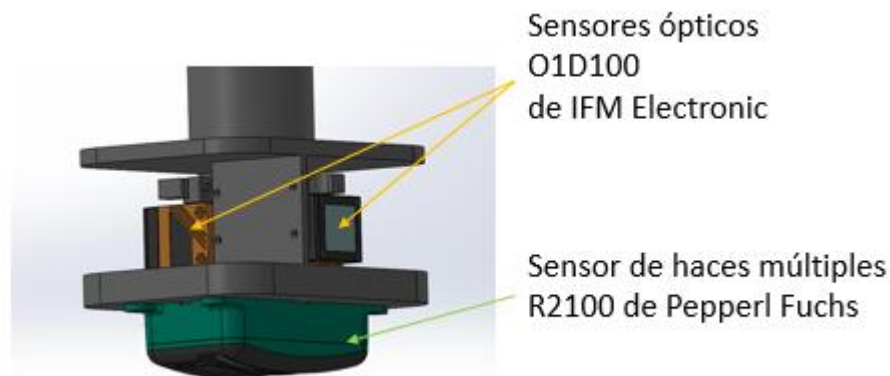


Figura 18: Posicionamiento de sensores

Como se puede ver en la figura 18, el sistema se monta en un cabezal acoplado a un brazo que entra verticalmente dentro de la tolva, calcula las medidas de longitud, ancho y altura, se retrae quedando las medidas disponibles para el cálculo del volumen. Si alguna medida fuera errónea, volvería a entrar en la tolva del camión repitiendo el proceso.

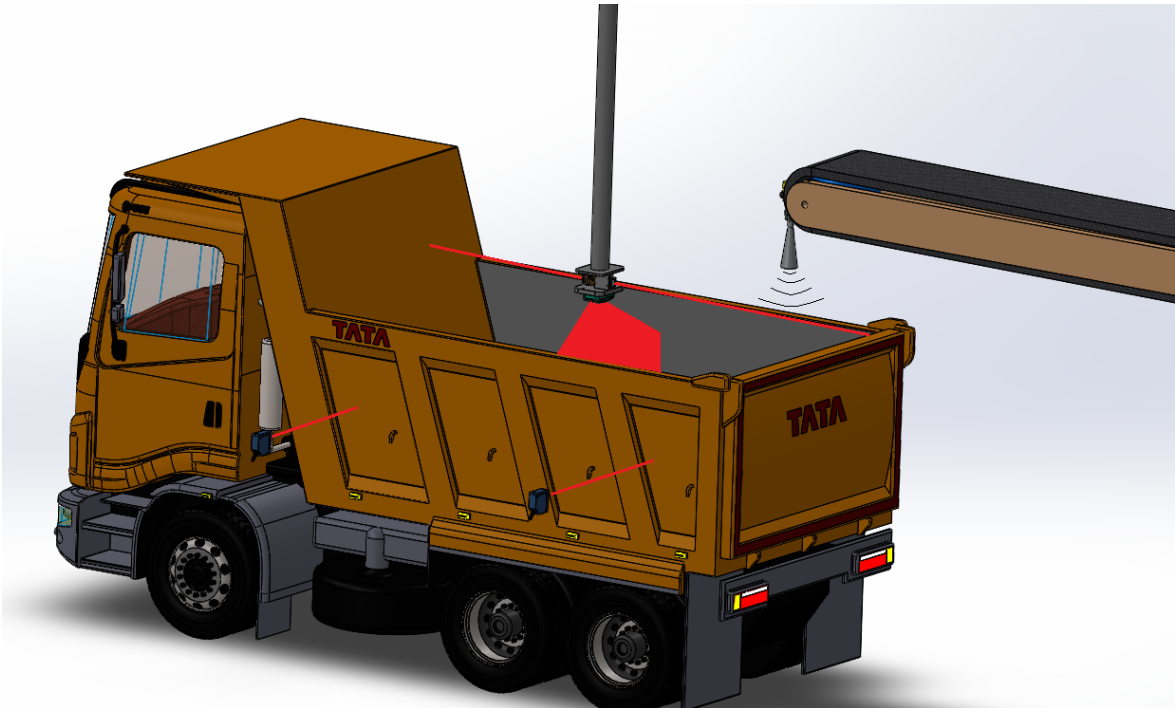


Figura 19: Representación de la ubicación de la instrumentación

3. MODELADO GRAFCET Y MÉTODO DE CÁLCULO

Para la definición del funcionamiento de la instalación se decidió realizar un modelado del sistema a partir de la metodología GRAFCET (del francés “*Grphe Fonctionnel de Commande Etape Transition*”) [13], que significa diagrama de control con etapas y transiciones.

Se emplea una definición de grafkets de nivel 1, es decir que donde solo se incluye una descripción funcional de los componentes sin tener en cuenta los aspectos tecnológicos de la instalación.

Durante el proceso de desarrollo del proyecto, se apreciaron tres procesos adecuados donde aplicar dicha metodología dando lugar a sendos grafkets que se corresponden con: el sistema general, el sistema de dosificación automático y el sistema de dimensionamiento de la tolva.

3.1. GRAFCET General

A continuación, se describe el grafket general de todo el proceso que se identifica con una estructura lineal. Es conveniente subrayar que los estados 1, 2 y 3 son macroetapas que posteriormente serán descritas con mayor detalle.

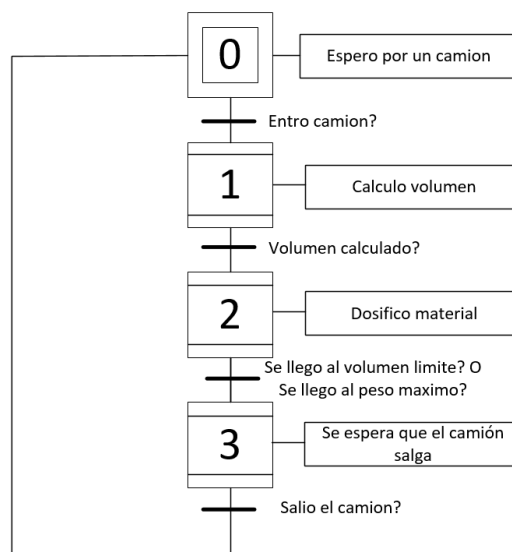


Figura 20: Grafket general

Como se puede ver en la figura 20 en la etapa inicial se hace una espera por el camión; si un camión entra en la báscula se hace el cálculo del volumen de la tolva, esta vendría a ser una macroetapa con la activación del grafket posteriormente explicado.

Después si el volumen ya se ha obtenido, se activa la macroetapa para dosificar material, el cual se encargará que llegue al volumen límite o de peso máximo permitido por ley para el camión.

Posteriormente se espera que el camión salga de la báscula; a continuación se esperará por otro camión repitiendo el proceso.

Vale la pena indicar que estos modelados serán utilizados como una guía para la programación del sistema aunque no se implemente en lenguaje SFC (del inglés, “*Sequential Function Chart*”), que sería una traducción casi idéntica de los graficets a la programación del PLC basado en el estándar IEC 61131-3.

Se ha optado por el lenguaje de programación en Texto Estructurado (ST) dada la versatilidad que este representa y la posibilidad de una migración más directa del código desarrollado con la herramienta Tia Portal para PLCs a la plataforma PCS7 donde finalmente estas aplicaciones deben ser integradas.

3.1.1. GRAFCET de dimensionamiento de tolva

Considerando el mecanismo propuesto, se realizó un graficet como el de la figura siguiente para obtener el volumen de la tolva.

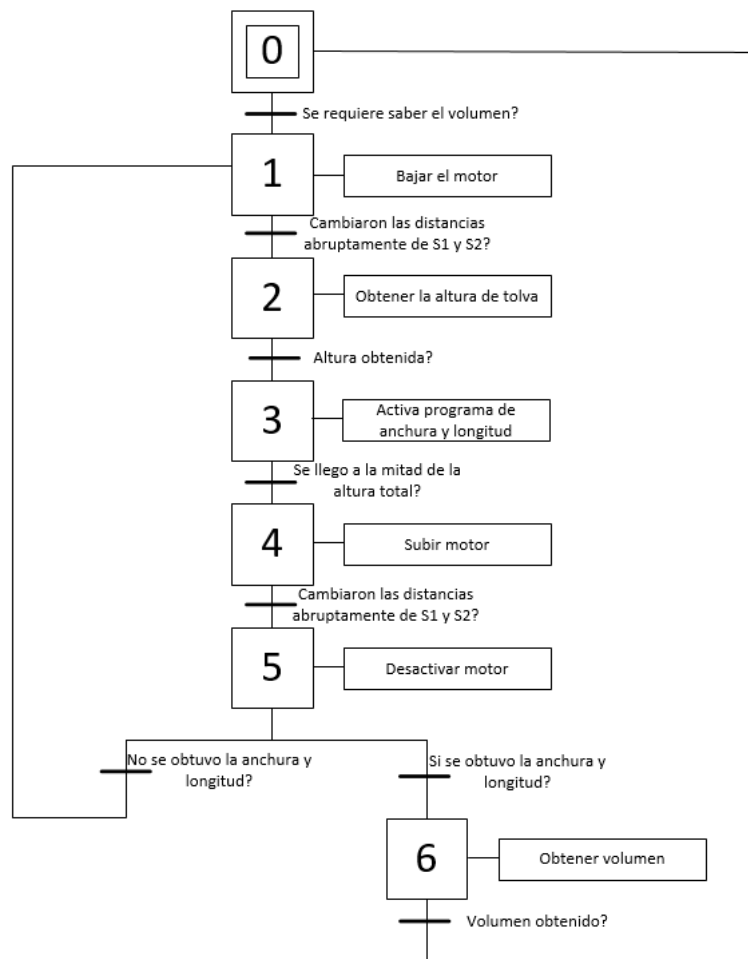


Figura 21: Graficet para el dimensionamiento de tolva

Se procederá a explicar el graficet de la figura 21:

- Se empezará interrogando si se tiene un requerimiento para determinar el volumen de una tolva. Si la respuesta es SÍ se activará el motor para bajada del mecanismo; posteriormente se preguntará si existe un cambio brusco de distancias en los sensores ópticos S1 y S2, lo que significa que se entró dentro de la tolva.
- Si existe tal cambio, se obtiene la altura de la tolva; una vez obtenida la altura se ejecutarán los programas para obtener la longitud y el ancho de la tolva. Estos calcularán mientras el motor baja el mecanismo hasta que este se encuentre a la mitad de la altura de la tolva del camión, luego se subirá.
- Como se hizo con anterioridad, se esperará que exista un cambio abrupto en los sensores ópticos S1 y S2, lo que significa que ya está en una posición más elevada que la tolva del camión.
- Para concluir con la explicación del graficet, se preguntará si se obtuvo correctamente el ancho y la longitud; si la respuesta es SÍ se proseguirá hasta calcular el volumen de la tolva, si la respuesta fuera negativa se repetiría el proceso.

3.2. GRAFCET de dosificación automática

Acerca del dosificado automático de malta se desarrolló un graficet (figura 22), con la peculiaridad de que este tiene una condición de finalización del sistema, que puede darse en cualquier momento. y que se comenta a continuación.

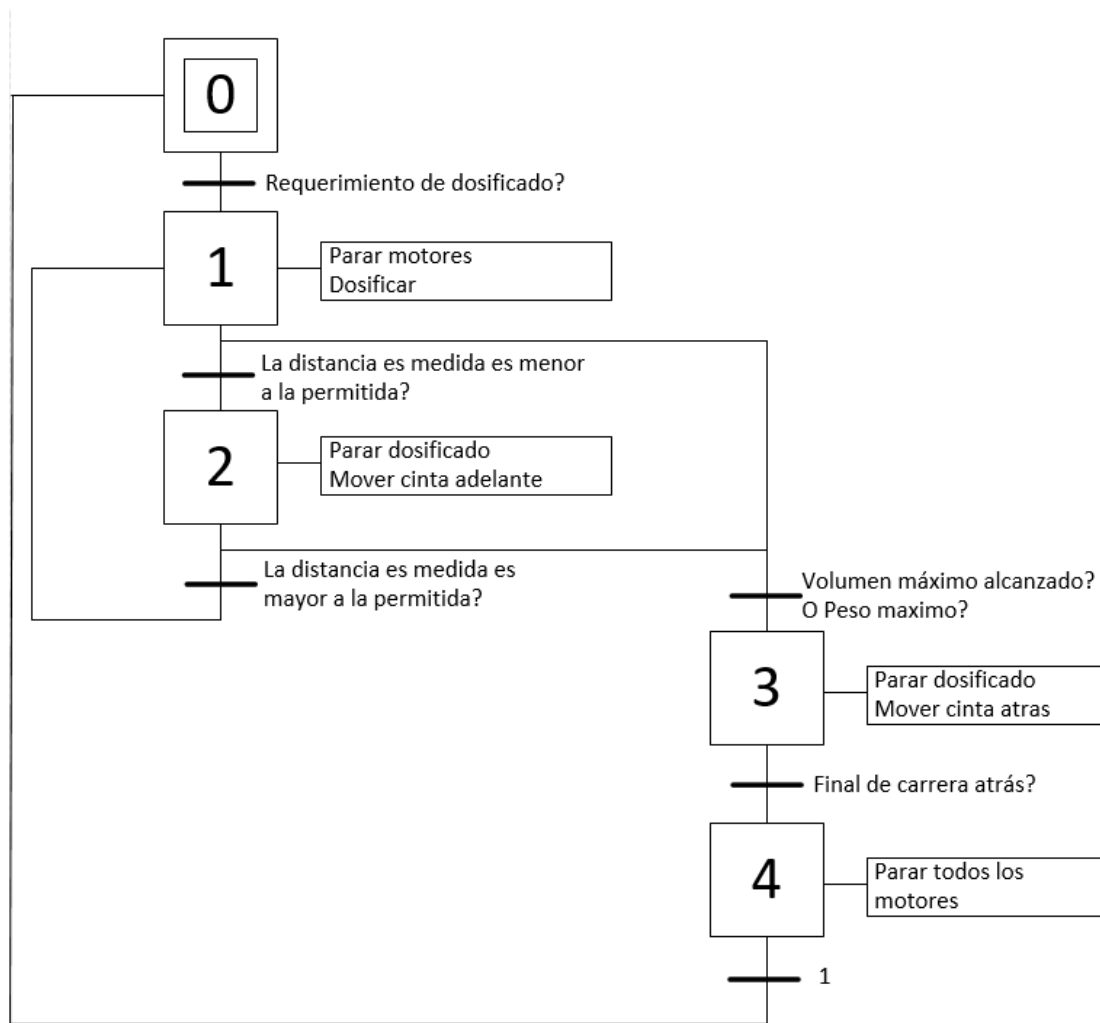


Figura 22: Grafcet de dosificado automático

Este grafcet se explica en dos partes: La primera se realiza la dosificación de la malta que consiste inicialmente en el desarrollo de este proceso en modo automático; si existe tal requerimiento se paran los motores de movimiento longitudinal de la cinta transportadora y se empieza a dosificar, si el sensor de radar detecta una distancia menor a la permitida para que no se desborde la tolva, se para el dosificado y se mueve la cinta transportadora hacia adelante. Si se detecta una distancia mayor a la permitida, se detendrá los motores de movimiento longitudinal y se volverá a dosificar material.

Este procedimiento se repetirá hasta, que el volumen máximo sea alcanzado o el peso máximo permitido para un determinado tipo de camión sea alcanzado; si ocurre alguno de estos sucesos se parará la dosificación y se retraerá la cinta transportadora. Si se llega al final de carrera de la cinta transportadora se pararán todos los motores y se repetirá el grafcet completo.

3.3. Método de obtención de medidas

Para la obtención de cada una de las medidas de la tolva, se siguió un método que se describe con ayuda de un par de imágenes.

Para el cálculo de la altura, puesto que inicialmente los sensores ópticos apuntan fuera de la tolva del camión, al ir bajando el mecanismo estos sensores cambian su lectura bruscamente, lo que significa que entró en la tolva y con la medida del haz vertical del escáner se podrá determinar la altura.

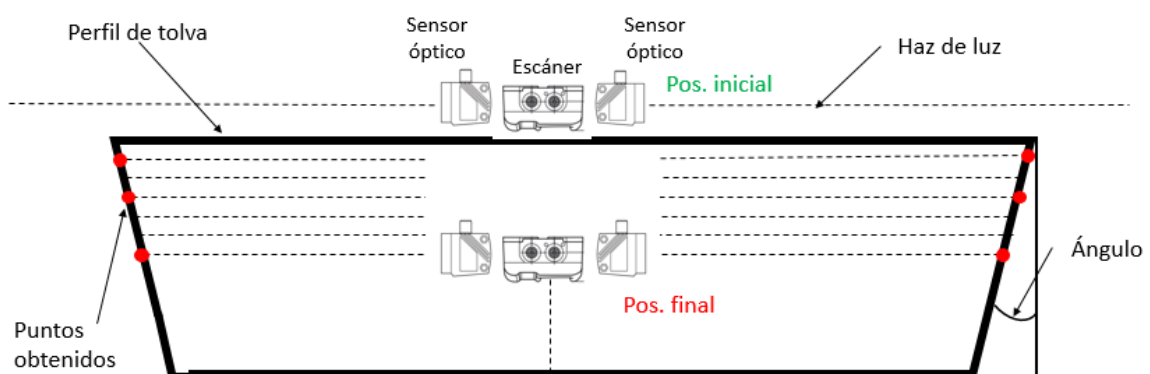


Figura 23: Método para obtención de altura y longitud

Entonces, para obtener la longitud de la tolva, primeramente se divide la altura ya conocida en 20 partes, de modo que a medida que baja el mecanismo se podrá hacer una lectura de los dos sensores ópticos, en cada una de las veinteavas partes de la altura, obteniendo con ello 10 medidas si se lleva solo hasta la mitad de la altura de la tolva.

La combinación de cada punto de altura del escáner con las distancias obtenidas por los sensores ópticos permite obtener por trigonometría el perfil de la tolva y determinar los ángulos del chaflán correspondiente en cada uno de los laterales. Calculando la media del ángulo, si ésta es próxima a cero significa que la tolva carece de chaflanes y es de tipo ortoedro.

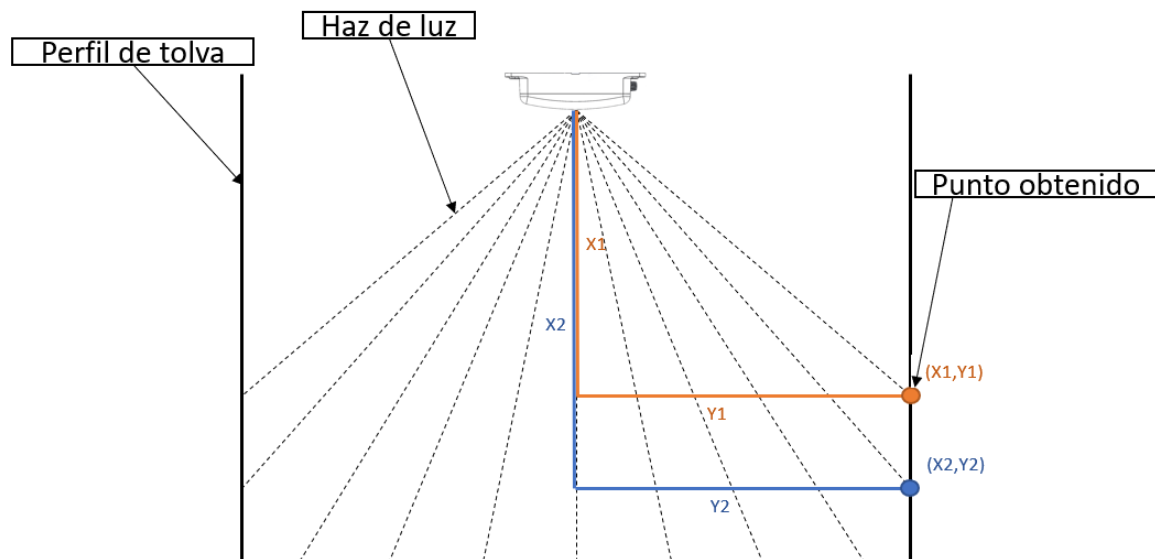


Figura 24: Método para obtención del ancho

Para obtener el ancho de la tolva, primero se determinan las distancias de cada uno de los haces; sabiendo que el ángulo entre cada uno de los haces es de 8 grados, se convirtieron las coordenadas polares en escalares, obteniendo coordenadas en ejes X e Y. Con las coordenadas, se calcularon las diferencias entre las coordenadas Y, si estas diferencias son menores a 7 mm significa que se ha encontrado una pared.

3.3.1. Requisitos para el uso del escáner de haces múltiples

Para el correcto uso del escáner de múltiples haces, se tiene que tener en cuenta varios aspectos previos como sería la configuración del puerto serie, la orden de requerimiento de distancias y por último el interpretar los bytes de respuesta.

Primero, se menciona la configuración del puerto serie:

- Type: RS-232
- Baud Rate: 115200 kBd
- Bits: 8
- Parity: None
- Stop Bits: 1

Figura 25: Configuración de puerto serie

A continuación, se tiene que tener en cuenta, el requisito de distancias y ecos. Esto se logra a través de una trama específica que se muestra en la tabla 2.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
0xde	0x01	0x05	0x59	0x83

Tabla 2: Trama para requisito de distancias y ecos

Como se puede ver en la tabla 2, para cumplir el requerimiento de comunicación se tiene que mandar cinco bytes con diferentes valores. Algo que se tiene que tener en cuenta es que según las instrucciones del protocolo, el requisito puede que ser llamado como máximo cada 20 ms.

Por último, la respuesta del escáner es una trama de 50 bytes formato en hexadecimal.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
0x01	0xde	0x32	0x11	distance 0 (LSB)	distance 0 (MSB)

Byte 6	Byte 7	Byte 8	Byte 9	Byte 10	Byte 11
echo 0 (LSB)	echo 0 (MSB)	distance 1 (LSB)	distance 1 (MSB)	echo 1 (LSB)	echo 1 (MSB)

Tabla 3: Trama de respuesta de distancias y ecos

Como se puede ver en la tabla 3 la trama de respuesta, tiene campos específicos en los cuales se devuelven las distancias y los ecos . Por ejemplo, el byte 4 y 5 tienen la distancia del haz 0. También se puede ver que ambos bytes tienen configuración de byte más significativo y byte menos significativo, por lo tanto, a la hora de programar hay que ordenarlos de forma que se obtenga la distancia deseada.

4. PRUEBAS DE FUNCIONAMIENTO

La programación realizada para el presente proyecto fue elaborada en su mayoría en SCL (del inglés, “*Structured Control Language*”) de Siemens.

No obstante, se hizo un énfasis en el escáner de haces de luz, ya que no existía registro por parte de Pepper Fuchs, de conexión del escáner con algún PLC. Por lo tanto, era propicio que se hiciera alguna prueba para tener la seguridad que el sensor funciona correctamente. Una vez acabadas las pruebas se hizo la conexión a un módulo de comunicaciones CM PtP para conexiones serie RS-232 y RS-485 del PLC de Siemens.

Una vez probadas y tratadas las señales del escáner, se realizaron las conexiones de los sensores ópticos O1D100 de IFM Electronic, a un módulo de entradas analógicas AI 4xI 2-4-wire ST del PLC.

Posteriormente con el escáner y los dos sensores de distancia funcionando, se hicieron las pruebas para el cálculo del volumen.

Por último, se desarrolló la programación tanto para el radar que mide la altura de malta dosificado en la tolva, así como el movimiento de la cinta transportadora y el posicionamiento del camión.

4.1.1. Pruebas de funcionamiento del escáner R2100

Para las pruebas de funcionamiento del escáner R2100 de Pepperl Fuchs se empleó *Visual Studio Community 2017* con la versión mostrada en la figura a continuación. El programa fue realizado en C; con un poco de práctica se pudo entender y modificar el programa para lograr nuestro objetivo. El archivo se encuentra en un zip adjunto a la documentación, con el nombre: “SerialPort.zip”.

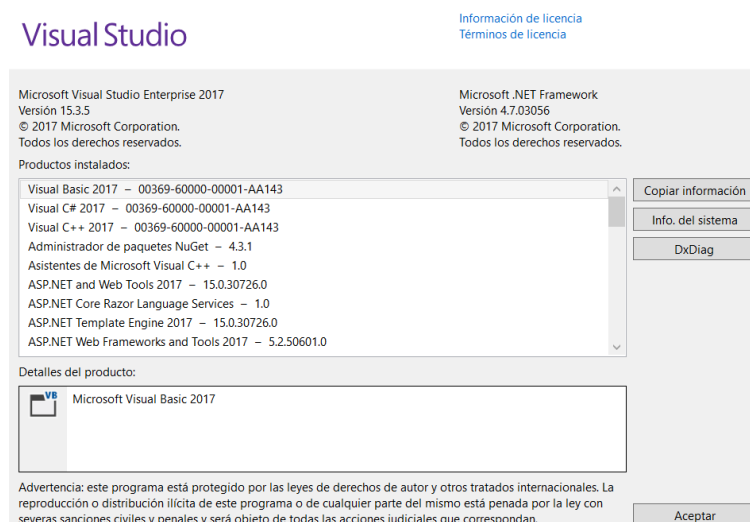


Figura 26: Versión de Visual Studio

Primero es preciso configurar el puerto serie como se muestra en la figura 27, esto se realiza a través de la interfaz, ya desarrollada.

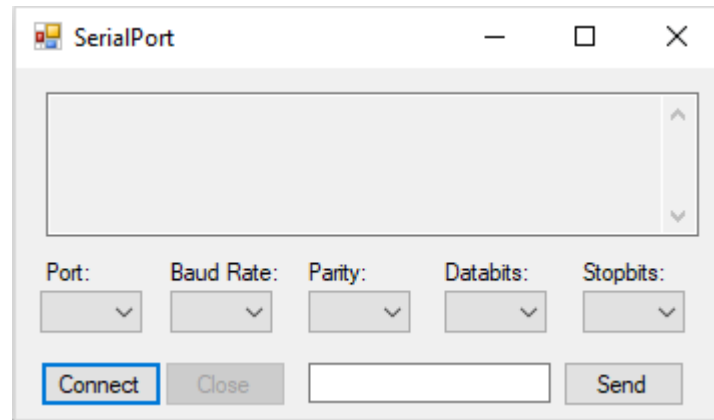


Figura 27: Interfaz de programa prueba

A continuación, hubo que hacer cambios en el código para obtener los resultados deseados. El primer cambio consistía en cambiar el evento a enviar, para que transmita la trama de la tabla 2.

```
private void button1_Click(object sender, EventArgs e)
{
    DateTime dt = DateTime.Now;
    String dtn = dt.ToShortTimeString();
    String data = txtDatatoSend.Text;

    //sport.Write(data);
    sport.Write(new byte[] { 0xDE, 0x01, 0x05, 0x59, 0x83 }, 0, 5);
    txtReceive.AppendText("[ " + dtn + " ] " + "Sent: " + data + "\n");
}
```

Figura 28: Método de evento Click en pulsador Enviar

Como se puede ver en la figura 28, anteriormente el programa obtenía un texto de la interfaz y lo mandaba a través del puerto, por otro lado, para el objetivo nuestro, se hizo un *array* de bytes con la trama de requisitos de la tabla 2. Con esta trama se hace el pedido de las distancias y los ecos.

```
private void sport_DataReceived(object sender, SerialDataReceivedEventArgs e)
{
    CheckForIllegalCrossThreadCalls = false;
    int[] distancias = new int[11];
    int[] posiciones = new int[22] { 4, 5, 8, 9, 12, 13, 16, 17, 20, 21, 24, 25, 28, 29, 32, 33, 36, 37, 40, 41, 44, 45 }; //Posiciones de bytes
    int conteo = 0;
    byte[] buffer = new byte[50]; //Buffer en el que se obtiene las distancias
    sport.Read(buffer, 0, 50);
    for (int i = 0; i <= 21; i++)
    {
        distancias[conteo] = buffer[posiciones[i + 1]] << 8 | buffer[posiciones[i]];
        conteo++;
        i++;
    }

    for (int i = 0; i < 11; i++)
    {
        txtReceive.AppendText("[ " + dtn + " ] " + "Distancia"+ i.ToString()+ ":"+ distancias[i].ToString()+"\n");
    }
}
```

Figura 29: Método de evento “DataRecieved”

Una vez enviada la trama de requisito, el evento “*DataRecieved*” se activa si existe nueva información de entrada. Anteriormente el programa, solamente leía lo que enviaba y se mostraba por pantalla el mismo texto enviado. Por lo tanto, para esta versión, se tuvo que hacer una total reestructuración del método.

Primero, se declaran dos *arrays* de tipo INT; en el primer array se tiene el de distancias, en el cual se almacenará las once distancias obtenidas por el escáner. Segundo se tienen las posiciones de bytes, en las que se encuentran las distancias obtenidas por el escáner. Son veintidós ya que por cada distancia se tiene un byte más significativo y un byte menos significativo, estas posiciones se obtuvieron de las instrucciones de operación dada por el fabricante. Para terminar con la declaración de variables, se tiene el buffer de los cincuenta bytes que se obtendrá del escáner y un contador que empezará en cero, que posteriormente se utilizará.

Como se ve en la figura se tiene la llamada a un método “*sport.Read*” siendo sport el puerto declarado. Con el presente método se obtiene la respuesta del puerto y se almacenan en la variable buffer. Posteriormente se hace el análisis de los datos obtenidos en el buffer que se explicara más detalladamente a continuación:

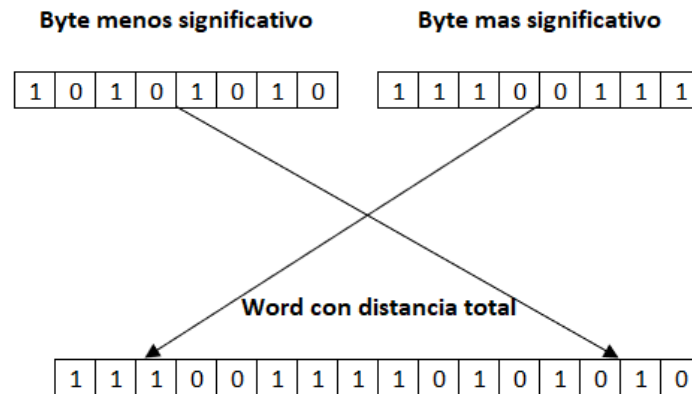


Figura 30: Concatenado de bytes

Como se puede ver en la figura 30, se tienen dos bytes el más significativo y el menos significativo por cada distancia obtenida por el escáner, los cuales hay que concatenar y crear un word con la distancia total. Esto debe ser aplicado a cada distancia, por lo tanto, fue aplicado once veces. Como resultado se programó un bucle “for” para realizar la presente tarea.

```

for (int i = 0; i <= 21; i++)
{
    distancias[conteo] = buffer[posiciones[i + 1]] << 8 | buffer[posiciones[i]];
    conteo++;
    i++;
}

for (int i = 0; i < 11; i++)
{
    txtReceive.AppendText("[ " + dtn + " ] " + "Distancia"+ i.ToString()+ ":"+ distancias[i].ToString()+"\n");
}

```

Figura 31: Ciclo for para concatenado de bytes

Como se puede ver en la primera instrucción “for” se llena el array distancias, para tal objetivo, se llaman los bytes de las distancias, se hace una instrucción OR entre el byte más significativo y el byte menos significativo, con la diferencia de que el byte más significativo se movió 8 bits a la izquierda, así de esa forma se obtiene la distancia requerida.

Después se utiliza la variable “conteo” y se aumenta en una unidad, esto se realiza para que el array “distancias”, se mueva en una posición y así almacenar la nueva distancia. Por otro lado, también se aumenta en una unidad la variable “i” ya que en el array “posiciones”, las posiciones se encuentran consecutivamente y al utilizar la variable “i” para llamar a cada posición se solaparían los bytes de una y otra distancia.

La instrucción “for” consiguiente simplemente, muestra las once distancias obtenidas por el escáner. Una vez acabada la prueba se obtuvieron las distancias satisfactoriamente como se muestra en la siguiente figura.

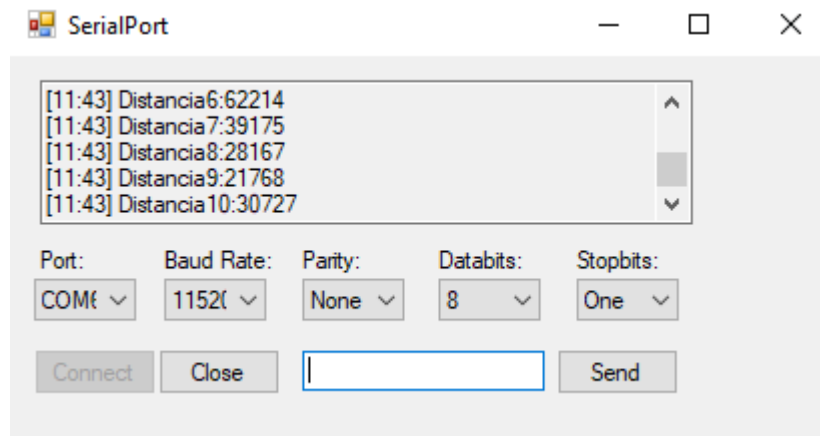


Figura 32: Interfaz de prueba

Como se aprecia, la respuesta del escáner es buena, sin embargo, existen algunas ocasiones donde no recibe respuesta, o solamente se recibe respuesta de algunos haces de luz. Aunque los resultados no fueron los deseados se decidió migrar el proyecto a una interfaz PLC para verificar si el desempeño era el mismo o este mejoraría.

5. ARQUITECTURA HARDWARE DEL SISTEMA

Cuando se verificó que el escáner se podía comunicar con el ordenador, se decidió migrar todo el proyecto a una plataforma de programación en PLC. Previamente a dar este paso se hizo la elección de hardware observando los requerimientos de la instrumentación seleccionada.

5.1. Topología de red

El sistema estará basado en una CPU S7 410-5H correspondiente a la serie S7-400 de Siemens.

Sin embargo para las pruebas se utilizó una CPU 1511-1 PN de la serie S7-1500 de Siemens que conecta vía Profinet con un módulo de periferia descentralizada IM 155-6 PN HF, al que se le añade un módulo de entradas analógicas AI 4xI 2-/4-wire ST y un módulo de comunicación CM_PtP.

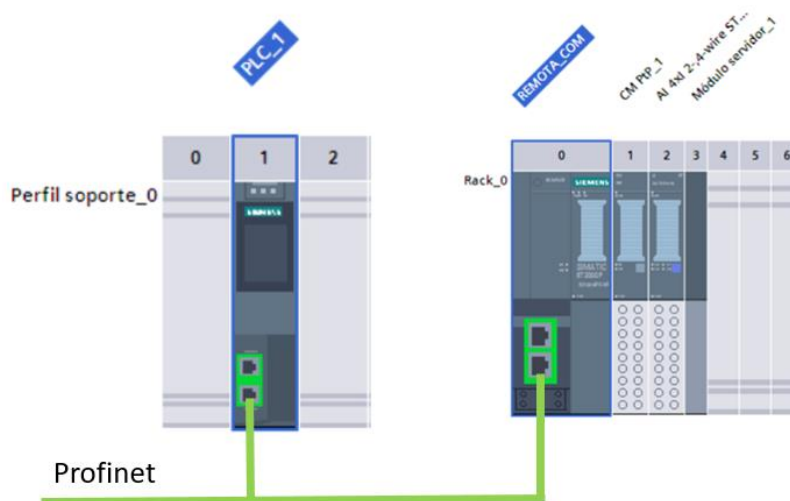


Figura 33: Configuración hardware equipos de Siemens

El autómatas programable se puede conectar con el ordenador mediante la interfaz Profinet, y a través de la misma se conecta, a la remota y a un sistema PC, que se utiliza para hacer visualizaciones en el software SCADA WinCC de Siemens.

El autómatas se configura con la dirección IP 192.138.0.1, la remota en la 192.138.0.151 y el sistema PC en la dirección 192.138.0.152.

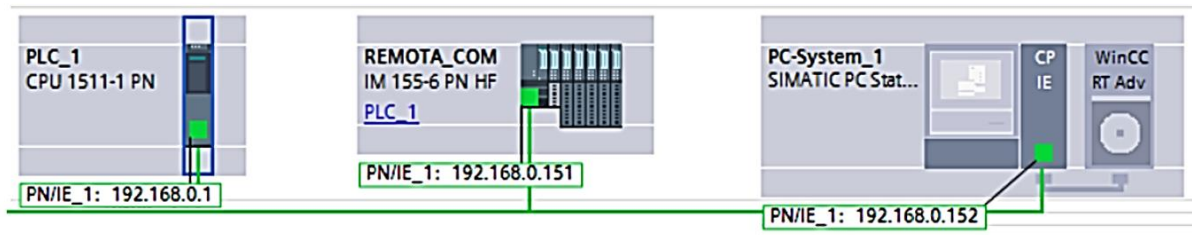


Figura 34: Configuración de red

5.1. Módulo CPU 1511-1 PN

El módulo CPU 1511-1 es un sistema de control modular para una serie de aplicaciones de automatización, la CPU es estándar en el segmento de potencia que posee Siemens, puede implementar aplicaciones PROFINET IO, puede realizar aplicaciones pequeñas y medianas con una memoria de trabajo de 1,23 M Bytes y un tiempo de ejecución de operaciones con bits de 60 ns.



Figura 35: Módulo CPU 1511-1 PN [14]

5.2. Remota interface Profinet IM 155-6 PN HF

El módulo de interface IM 155-6 PN HF posee una conexión a PROFINET IO, con una configuración máxima de 64 módulos de periferia ET 200SP + 16 módulos ET 200AL con un bus de fondo de 1 m.



Abbildung ähnlich

Figura 36: Módulo de Interface IM 155-6 PN/2 HF [15]

5.3. Módulo de Comunicación CM PtP

Por un lado, se tiene el escáner que tiene una comunicación a través de RS-232, sin embargo, se desea que la comunicación se haga a través de RS-485 para alcanzar mayores distancias entre el PLC y el escáner. Por lo tanto, se necesita de un módulo que permita esta comunicación, como ya se mencionó con anterioridad para el presente proyecto se instalara una remota ET-200SP, pero se necesitaba de un módulo que sea compatible con esta remota.



Figura 37: Módulo de comunicación CM PtP [16]

Se escogió el módulo de comunicación CM PtP que permite comunicación por RS-232, RS-485 y RS-422. Lo que lo hace propicio para nuestro objetivo ya que en una primera instancia se comunicó a través de RS-232 y posteriormente al integrar la pasarela de RS-232 a RS-485 se podría comunicar ambos sin problemas.

5.4. Módulo de entradas analógicas AI 4xI 2-/4-wire ST

A continuación, se necesitaba leer lecturas analógicas de 4 a 20 mA, por lo que se necesitaba de un módulo de entradas analógicas.



Figura 38: Módulo de Entradas Analógicas [17]

Como se puede ver en la figura 38, se tiene el módulo de entradas analógicas para dos y cuatro hilos, que lo hace propicio para la conexión de nuestros dos sensores de distancia y nuestro sensor radar, que son conectados a dos hilos.

6. PROGRAMACIÓN EN TIA PORTAL

Como se explicó con anterioridad, se realizó una programación en el programa *Tia Portal v14*. Además se utilizó “*WinCC Basic*” para realizar unas pantallas de explotación .Las versiones de ambos programas se muestran en la siguiente figura.

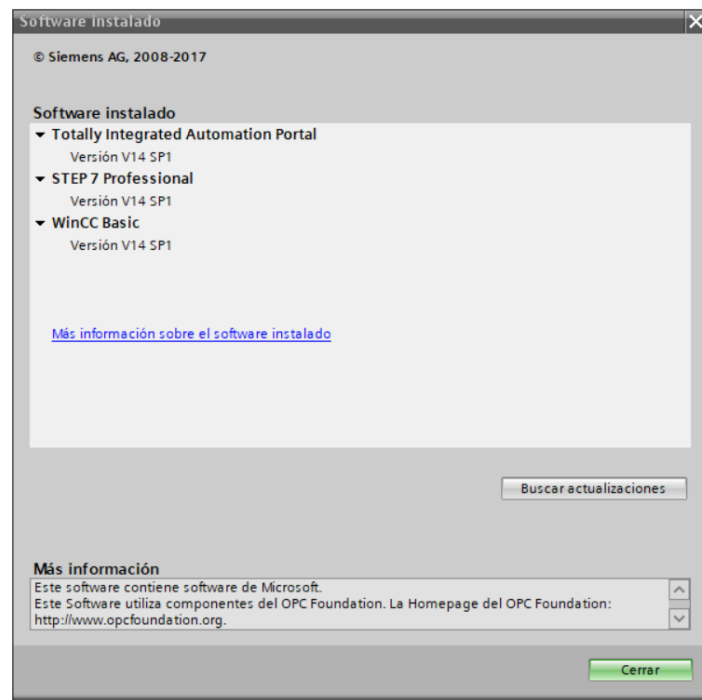


Figura 39: Versión de Tia Portal y WinCC

Para esta parte de la documentación, primero se hará una identificación de las entradas y salidas del sistema, segundo se identificarán los ficheros, tercero se presenta un árbol de llamadas de todos los programas, y finalmente se explicará cada uno de los bloques de organización, los bloque de datos globales, funciones y bloques funcionales.

6.1. Entradas y salidas del sistema

En el presente apartado se mostrarán las entradas y salidas globales del sistema.

	Nombre	Tipo de datos	Dirección	Rema...	Acces...	Escrib...	Visibl...	Supervisión	Comentario
1	i_xSF1	Bool	%M0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Sensor fotoelectronico de presencia de cami.
2	i_xSF2	Bool	%M0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Sensor fotoelectronico de presencia de cami.
3	i_wSO1	Word	%IW10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Sensor optico para medicion de chaffan 1
4	i_wSO2	Word	%IW8	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Sensor optico para medicion de chaffan 2
5	i_wSR	Word	%IW12	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Sensor de radar para dosificado automatic
6	i_xEmpezar	Bool	%M1.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Boton de Inicio de sistema
7	Final_Carrera	Bool	%M1.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Final de Carrera de cinta transportadora

Figura 40: Entradas digitales y analógicas del sistema

Como se puede ver en la figura 40 se tienen los diferentes tipos de entradas utilizadas, el nombre en el sistema, tipo de variable, la posición de memoria de entradas y un pequeño comentario de cada una de las entradas. Hay que destacar que las posiciones de memoria

para los sensores fotoeléctricos, el botón de empezar y el final de carrera, se han direccionado como marcas ya que no se disponía físicamente de dichos elementos y de esta forma es posible el forzado de valores durante la depuración en las pruebas.

ENTRADAS POR COMUNICACIÓN DEL SISTEMA				
Ordinal	Nombre tecnico	Interfaz	Protocolo	Tipo de variable
1	Escaner de haces multiples R2100	RS-232	Libre	Bytes
2	Bascula	Desconocido	Desconocido	Real
3	Lector de matriculas	Desconocido	Desconocido	Real

Tabla 4: Entradas por comunicación del sistema

Por otro lado, se tienen las entradas al sistema que se generan a través de comunicación, como se muestra en la tabla 4. Se tienen tres dispositivos que se comunican a través de un tipo de comunicación, de los cuales solamente se conoce la interfaz y el protocolo del escáner láser, desconociendo la comunicación de la báscula y el lector de matrículas que no forman parte de este trabajo.

A continuación se muestran las salidas del sistema.

	Nombre	Tipo de datos	Dirección	Rema...	Acces...	Escrib...	Visibl...	Supervisión	Comentario
1	q_xKM_Cinta	Bool	%M2.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Contactador para activar cinta de dosificado
2	q_xKM_Mov_Long_Derecha	Bool	%M2.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Contactador para que cinta completa se mu...
3	q_xKM_Mov_Long_Izquierda	Bool	%M2.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Contactador para que cinta completa se mu...
4	q_xKM_Mec_Abajo	Bool	%M2.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Contactador para bajada de mecanismo
5	q_xKM_Mec_Arriba	Bool	%M2.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Contactador para subida de mecanismo

Figura 41: Salidas del sistema

Como se puede ver en la figura 41 aparecen en la tabla el nombre en el sistema, el tipo de variable, la posición en la memoria y un pequeño comentario de cada una de las salidas.

Se trata de tres motores, dos con cambio de sentido de giro que se corresponden: primero al mecanismo de subida y bajada del sistema de dimensionamiento, segundo para el movimiento longitudinal de la cinta, de derecha e izquierda. Por último, se tiene el motor de la cinta transportadora activando la dosificación del material.

Es importante indicar en la posición de la memoria es que estas son marcas, ya que no se tiene bien definido la salida de cada motor no pudiendo hacer pruebas con motores reales.

6.2. Identificación de ficheros realizados

Para la realización del programa se han realizado los siguientes módulos de programación que aparecen en la imagen siguiente y que corresponden con el fichero Tia Portal identificado con el nombre: “Automatizado_de_dosificado_de_Malta.zapl4”.

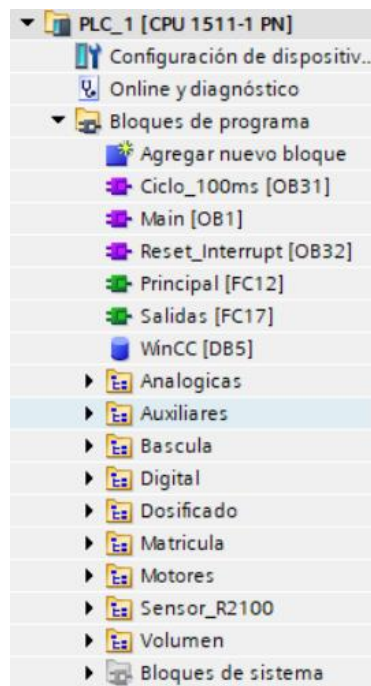


Figura 42: Identificación de ficheros

Como se puede ver en la figura 42, se aprecian de forma explícita en el árbol del proyecto tres módulos de organización (OB1, OB31 y OB32), dos funciones (FC12 y FC17) y el bloque de datos global DB5 denominado “WinCC”. Por otro lado se tienen diferentes carpetas donde se encuentran organizados todos los FB y FC necesarios que componen la aplicación.

6.3. Organización de programa de control

Como ya se explicó, en el apartado de modelado se organizan llamadas a los diferentes bloques de forma que la programación resulte lo más parecido a los graficets diseñados.

1	Main	OB1		0	0
2	Ciclo_200ms	OB30		8	8
3	Adaptacion_sensor, Adaptacion_...	FB1, DB7	@Ciclo_200ms ▶ Seg1	16	8
4	WinCC	DB5	@Ciclo_200ms ▶ Seg1	8	0
5	Ciclo_100ms	OB31		0	0
6	Principal	FC12	@Ciclo_100ms ▶ Código del programa	0	0
7	Dosificado_Automatico, Dosi...	FB5, DB17	@Principal ▶ Seg3	24	24
8	Dosificado_Automatico_DB	DB17	@Principal ▶ Seg3	0	0
9	Dosificado_Automatico_DB	DB17	@Principal ▶ Seg4	0	0
10	Dosificado_Automatico_DB	DB17	@Principal ▶ Seg2	0	0
11	Entrada_Camion, Entrada_Ca...	FB2, DB13	@Principal ▶ Seg1	0	0
12	Entrada_Camion_DB	DB13	@Principal ▶ Seg1	0	0
13	Entrada_Camion_DB	DB13	@Principal ▶ Seg2	0	0
14	Gestion_Volumen	FC11	@Principal ▶ Seg2	8	8
15	Salida_Camion, Salida_Camio...	FB7, DB18	@Principal ▶ Seg4	0	0
16	Salida_Camion_DB	DB18	@Principal ▶ Seg4	0	0
17	Salida_Camion_DB	DB18	@Principal ▶ Seg1	0	0
18	WinCC	DB5	@Principal ▶ Seg2	0	0
19	WinCC	DB5	@Principal ▶ Seg3	0	0
20	WinCC	DB5	@Principal ▶ Seg3	0	0
21	Salidas	FC17	@Ciclo_100ms ▶ Código del programa	0	0
22	Sensor_S1	FC7	@Ciclo_100ms ▶ Código del programa	8	8
23	Sensor_S2	FC9	@Ciclo_100ms ▶ Código del programa	8	8
24	Reset_Interrupt	OB32		0	0
25	RESET_DB	FC6	@Reset_Interrupt ▶ Seg1	0	0

Figura 43: Estructura de llamadas

Como se puede ver en la figura 43, se tiene tres OBs (*Bloques de Organización*) que llaman a todas las funciones y bloques funcionales del programa:

- El OB1 “Ciclo_200ms”, que llama cada 200ms al bloque funcional FB1 “Adaptacion_sensor” sensor para obtener el valor las lecturas del escáner.
- El OB30 “Ciclo_100ms” que llama cada 100ms a las funciones FC12 “Principal”, FC17 “Salidas”, FC7 “Sensor_S1” y FC9 “Sensor_S2”. Por otra parte, la función FC12 “Principal” llama a los bloques FB5 “Dosificado_Automatico”, FB2 “Entrada_Camion”, FC11 “Gestion_Volumen” y FB7 “Salida_Camion”, y de igual forma que el graficet propuesto en la figura 20.
- Por último se programa el OB31 “Reset_Interrupt” que llama a la función FC6 “Reset_DB”, que en ocasiones se utilizó para resetear los valores de los DBs de instancias de “Altura”, “Ancho” y “Longitud”.

6.4. Bloques de Organización implementados

Como ya se mencionó se han implementado 3 OBs (Bloques de Organización), que gestionan todo el programa, los cuales todos serán explicados a fondo en este apartado.

El OB30 “Ciclo_100ms” es el OB principal del programa que se encarga de llamar a cuatro funciones “Principal”, que activa el programa principal, “Sensor_S1” que llama a la lectura del sensor óptico, “Sensor_S2” que llama a la lectura del otro sensor óptico, “Salidas” que activa las salidas si estas tuvieran alguna señal de actuación.

```

1 "Principal"();
2 "Sensor_S1"();
3 "Sensor_S2"();
4 "Salidas"();

```

Figura 44: Código del OB Ciclo_100ms

El OB30 “Ciclo_200ms” es el OB que se encarga de gestionar la comunicación con el escáner láser, a través de la llamada al bloque funcional “Adaptacion_sensor”. Como se puede ver en la figura 45 se observa una llamada a dicho FB, que solo posee una entrada llamada REQ que significa *Requisito*; esta se activa a través de una variable de un DB global, si esta variable es puesta a TRUE entonces “Adaptacion_sensor” se ejecutaría cada 200 ms.

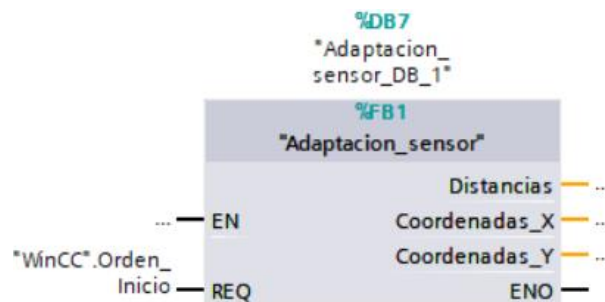


Figura 45: Código de OB “Ciclo_200ms”

El OB31 “Reset_Interrupt” se utilizó con fines prácticos, para resetear valores de los DB de instancia de “Altura”, “Ancho” y “Longitud”, además de algunos valores que afecten el cálculo del volumen de la tolva. Como se puede ver en la figura 46 se hace una llamada al FC6 “Reset_DB”, que posee una entrada para activar su función de *Reset*.

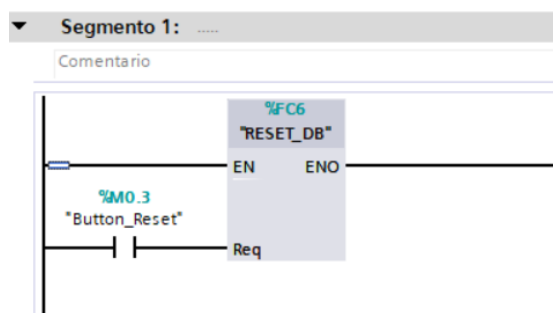


Figura 46: Código de OB “Reset_Interrupt”

6.5. Explicación de bloques de datos

Los DBs (Bloques de Datos) contienen datos de la memoria del PLC organizados como se desee en base a la declaración de variables y tipos asociados a dichos DBs. Se consideran estas variables como globales pudiendo acceder a ella con el nombre del DB, un punto y el nombre de la variable.

Sin embargo también se crean DBs de instancia cuando se llama a un FB (Bloque funcional), y en ellos guardan los valores de entradas/salidas y las variables internas de la llamada a dicho bloque funcional.

En este apartado se hará mención solo a los primeros pues los segundos (DBs de instancia) su estructura está implícita en las variables definidas para el bloque funcional correspondiente.

El DB global más utilizado es el DB5 "WinCC", cuyo nombre es porque está relacionado con la aplicación de visualización. El presente DB posee 7 arrays de once posiciones, en cada uno de ellos se almacena un historial con los últimos 11 datos obtenidos del sensor escáner de haces múltiples y los sensores ópticos.

Entre los arrays, se tiene:

- "Coordenadas", estos valores son valores cartesianos en eje X e Y, para saber en qué posición el haz de luz del escáner rebotó sobre un objeto.
- "Historial_X" (5, 6 y 4), son valores de distancia para el número de haz, mostrado en su respectivo ordinal, por ejemplo Historial_X_5, se corresponde con el historial del 5º haz.
- "Historial_Dist" (S1 y S2), son datos de distancia obtenidos de los dos sensores ópticos de distancia.
- Posteriormente se tienen valores de tipo LREAL, que son variables de resultados obtenidos, en el cálculo del "Ancho", "Largo", "Longitud", "Volumen", "Ángulo_1" y "Ángulo_2".
- Abajo se definen tres variables booleanas, que son utilizadas para saber si el volumen ha sido calculado, si un carro (camión) se encuentra en posición y para dar la orden de inicio al escáner de haces múltiples.

	Nombre	Tipo de datos	Valor de arranq...
	Static		
	Coordenadas_X	Array[0..10] of LReal	
	Coordenadas_Y	Array[0..10] of LReal	
	Historial_X_5	Array[0..10] of LReal	
	Historial_X_6	Array[0..10] of LReal	
	Historial_X_4	Array[0..10] of LReal	
	Historial_Dist_S1	Array[0..10] of LReal	
	Historial_Dist_S2	Array[0..10] of LReal	
	Ancho	LReal	0.0
	Altura	LReal	0.0
	Longitud	LReal	0.0
	Volumen	LReal	0.0
	Angulo_1	LReal	0.0
	Angulo_2	LReal	0.0
	OK_Volumen	Bool	false
	Carro	Bool	false
	Orden_Inicio	Bool	false

Figura 47: DB5 "WinCC"

El DB4 "Req_Comunicacion" se creó con la finalidad de tener un espacio de memoria, en el que se tengan:

- Los requerimientos de comunicación mostrados en la tabla 2, siendo un array de 5 bytes.

- Las posiciones de los bytes de distancia.
- Los ángulos de cada uno de los haces en radianes.

Además de un espacio de memoria para depositar los 50 bytes de respuesta, que el escáner laser responde, donde se encuentran los valores de distancia y eco de cada uno de los haces de luz.

Req_Comunicacion			
	Nombre	Tipo de datos	Valor de arranq...
1	Static		
2	Pedido	Array[0..4] o...	
3	Recibidos	Array[0..49] of Byte	
4	Posiciones	Array[0..21] of Int	
5	Angulos	Array[0..10] of LReal	

Figura 48:DB4 Req_Comunicación

6.6. Explicación de funciones y bloques funcionales

6.6.1. Comunicación escáner [FC1])

Para esta función solamente se tiene una entrada llamada “REQ”, que toma el valor TRUE, para que la función se ejecute.

Para la comunicación con el escáner, se utilizaron los bloques funcionales de la librería de Siemens “Port_Config”, “Send_P2P” y “Recieve_P2P”.

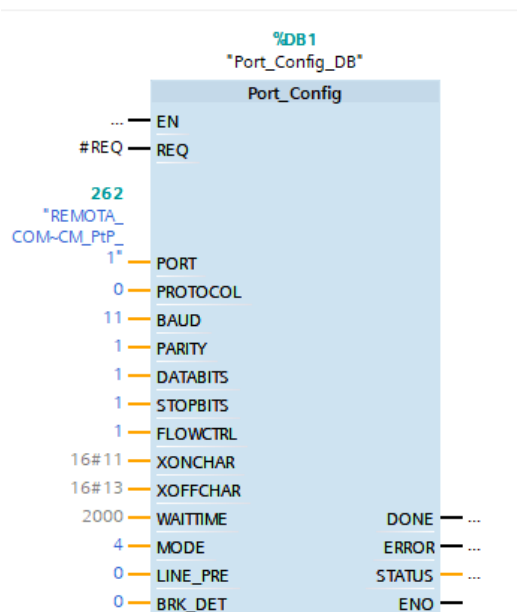


Figura 49: Llamada al bloque funcional “Port_Config”

Como se muestra en la figura 49, se tiene algunos parámetros que configurar, que se muestran en la figura 25, donde se ve la configuración del puerto, por lo tanto, con estas especificaciones y con el manual de uso de Port_Config se utilizó el bloque.

En el parámetro PORT se indicó el puerto del módulo de comunicación, en el parámetro PROTOCOL se puso cero para la configuración en Freeport, el parámetro BAUD 11 para la velocidad a 115200 Baudios, PARITY 1 para que no tenga bit de paridad, DATABITS 1 para 8 bits de datos, STOPBITS 1 para indicar un bit de parada. Posteriormente los parámetros se dejaron como se configuraron por defecto, excepto por el parámetro MODE el cual se especificó 4 para RS-485 half dúplex, siendo la configuración para la pasarela de RS-232 a RS-485. [18]

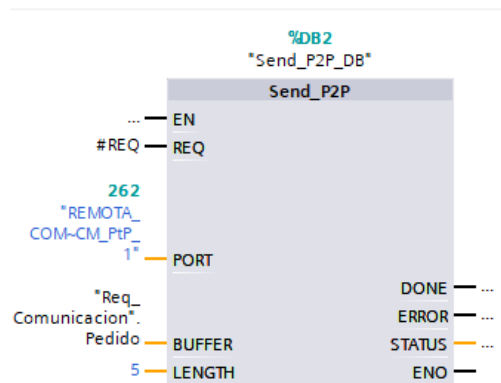


Figura 50: Bloque Send_P2P

Como se puede ver en la figura 50 al igual que el anterior bloque, este también posee algunos parámetros de entrada, entre ellos uno es de requerimiento que será activado cuando se tenga una solicitud de enviar alguna trama, y se configura el puerto al igual que el anterior bloque.

Luego se tiene la entrada BUFFER, a la cual hay que mandar una trama de cinco bytes mostrada en la tabla 2. Esta trama fue cargada en el DB "Req.Comunicacion", en forma de un array llamado "Pedido", por último, se determinó el tamaño del array que en nuestro caso es 5.

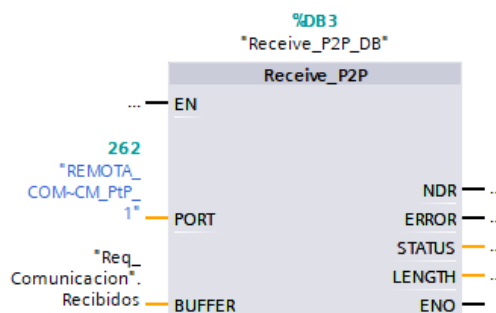


Figura 51: Bloque Recieve_P2P

Como se muestra en la figura 51 para recibir la respuesta del escáner, se utilizó el FB “*Recieve_P2P*”. En este bloque se ponen los parámetros de entrada que son el PORT, que es el puerto de la tarjeta de comunicación y el BUFFER donde se almacenará la respuesta del puerto, en este caso la respuesta será almacenada en el DB “*Req_Comunicacion*” en la variable “*Recibidos*”.

6.6.2. Adaptación sensor [FB1]

Para el presente FB se tiene las siguientes entradas y salidas.

Nombre	Valor predet.	Remanencia	Accesible d...	Escrib...	Visible en ...	Valor de a..	Supervisión	Comentario
Input			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
REQ	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Requerimiento de la entrada
Output			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Distancias	Int	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Distancias obtenidas
Coordenadas_X	LReal	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Coordenada en eje X
Coordenadas_Y	LReal	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Coordenada en eje Y

Figura 52: Entradas y salidas de *Adaptación_sensor*

Por lo que se refiere a la programación, se dividió el programa en tres partes: la obtención de distancias reales, la conversión de coordenadas polares a escalares y el llenado de un histórico de las distancias, que será utilizado con posterioridad.

```
#Contador := 0;

REGION Distancias_Reales
  FOR #I := 0 TO 21 BY 2 DO
    #Aux := "Req_Comunicacion".Recibidos["Req_Comunicacion".Posiciones[#I + 1]] * 256;
    #Distancias[#Contador] := #Aux + "Req_Comunicacion".Recibidos["Req_Comunicacion".Posiciones[#I]];
    #Contador := #Contador + 1;
  END_FOR;
END_REGION
```

Figura 53: Código para obtención de distancias reales

Primero, hay que explicar que antes de esta parte de código, se hizo una llamada a la función “*Comunicación_Escaner*”, para así obtener los cincuenta bytes de respuesta que se necesitan. Después se ejecuta la sección del código de la figura 53, en la cual se tiene un contador iniciado en 0, seguido por un ciclo “*for*” que va de 0 a 21 que avanza de dos en dos, consiguientemente se tiene un auxiliar que alojará el *byte* más significativo; el objetivo de este auxiliar es realizar el concatenado de ambos *bytes* del más y menos significativo.

Segundo, se obtiene el *byte* más significativo de una distancia, del array de 50 bytes obtenidos de la función “*Comunicación_Escaner*”. De las posiciones en la que se encuentra los *bytes* de distancias, estas posiciones se guardaron en el DB de “*Req_Comunicacion*” en el array “*Posiciones*”, como la posición del *byte* más significativo está al lado del *byte* menos significativo, se indica que del array “*Posiciones*” se posicione una unidad a la derecha ([i+1]).

Como se puede ver en la misma línea, el *byte* más significativo es multiplicado por 256, para que este *byte* se mueva en 8 *bits* a la izquierda y así poder sumarlo al *byte* menos significativo. En la siguiente línea de código obtenemos la distancia sumando ambos *bytes*,

el *byte* más significativo y el *byte* menos significativo. Posteriormente se suma el contador en una unidad, para así llenar de forma correcta el *array* de “Distancias”.

```

REGION Coordenadas_X_e_Y
  FOR #J := 0 TO 10 DO
    #Coordenadas_X[#J] := #Distancias[#J] * COS("Req_Comunicacion".Angulos[#J]);
    #Coordenadas_Y[#J] := #Distancias[#J] * SIN("Req_Comunicacion".Angulos[#J]);
  END_FOR;
END_REGION

```

Figura 54: Código para la obtención de coordenadas cartesianas.

Obtenidas las distancias se conseguirán las coordenadas en el eje X e Y. Para adquirir este resultado simplemente se multiplicó la distancia por el seno y coseno del ángulo de cada haz de luz. Ya que cada haz de luz tiene un ángulo de 8 grados entre cada haz y haz, simplemente se multiplicó el array “Ángulos” del DB “Req_Comunicacion” con los respectivos ángulos en radianes; esto se hizo ya que las funciones de seno y coseno del Tia Portal funciona en radianes, como resultado se obtuvo las coordenadas en X e Y de cada haz de luz.

```

REGION WinCC
  "WinCC".Coordenadas_X := #Coordenadas_X;
  "WinCC".Coordenadas_Y := #Coordenadas_Y;
  "Buffer_Llenado"(Buffer_llenado => "WinCC".Historial_X_4,|
    Dato := #Coordenadas_X[0],
    Size := 10);

  "Buffer_Llenado"(Buffer_llenado => "WinCC".Historial_X_5,
    Dato := #Coordenadas_X[5],
    Size := 10);

  "Buffer_Llenado"(Buffer_llenado => "WinCC".Historial_X_6,
    Dato := #Coordenadas_X[1],
    Size := 10);
END_REGION

```

Figura 55: Código para llenado de historial

Con respecto a la asignación de valores de historial, se llenó el DB llamado “WinCC”. Por lo tanto como se puede ver en la figura 55 se llena el *array* del DB “WinCC” con las coordenadas obtenidas, después se tiene la llamada a la función llamada “Buffer_Llenado” el cual consiste en llenar y obtener un historial de los últimos 11 datos recopilados por el sensor.

Por último, se hace la llamada a la función “Comunicacion_escaner”, pero con el requisito a FALSE, lo que significa que no se requerirán nuevos datos del escaner.

6.6.3. Sensor S1 [FC7]

Ahora bien, antes de explicar, sobre cómo se obtuvo el volumen de la tolva, se explicara cómo se gestionaron los sensores ópticos de distancia O1D100 de IFM Electronic.

```

REGION Escalado
    #Resultado := 0.36168981 * "i_wS01";
END_REGION
REGION Historico
    //Llenamos un historico del sensor de distancia l
    "Buffer_Llenado"(Dato := #Resultado,
                    Buffer_llenado => "WinCC".Historial_Dist_S1,
                    Size := 10);
END_REGION

```

Figura 56: Código de función Sensor_S1

Como se puede ver en la figura 56 se tiene dos regiones de código, en la primera región se tiene el escalado, posteriormente se tiene la región del histórico. En la región de escalado se realiza una multiplicación que es la pendiente de la recta que será explicado a continuación.

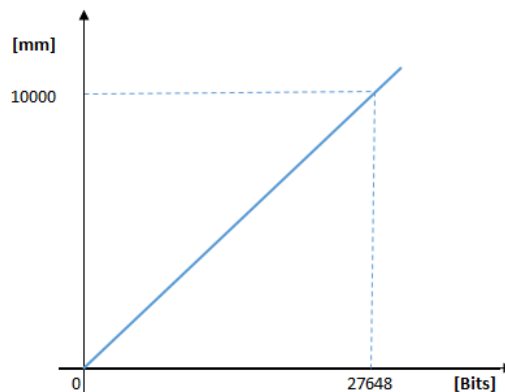


Figura 57: Representación de la recta del sensor óptico

En la figura 57 se muestra la representación de la recta, donde en abcisas se tiene los Bits y en ordenadas la distancia en mm, es una relación de la entrada analógica y el valor real de distancia que le corresponde. Como se puede suponer la entrada analógica responde a la resolución de bits, sin embargo aunque la resolución es de 15 bits el rango nominal no los utiliza todos, por lo tanto leyendo la representación de rangos de entrada, de la hoja de características del módulo de entradas analógicas, se obtiene que el valor decimal medido al 100% es de 27648. Haciendo la analogía con nuestro sensor que mide hasta diez metros obtuvimos la pendiente de la recta representado en la figura 57.

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

$$m = \frac{10000 - 0}{27648 - 0}$$

$$m = 0.36168981$$

Sabiendo la ecuación de la recta:

$$y = mx + b$$

dónde “*m*” es la pendiente y “*b*” la ordenada en el origen, y teniendo en cuenta que la recta pasa por el origen se obtiene una multiplicación entre la pendiente y el valor de entrada, como se muestra en el código de programa de la figura 56.

Posteriormente se hizo el llenado del historial en el *array* “Historial_Dist_S1” en el DB “WinCC”, el uso del presente historial se explicará posteriormente. Por otro lado, como se tienen dos sensores de este tipo, por lo tanto se tiene otra función con la única diferencia de que este llena otro historial y está direccionado para su correspondiente entrada analógica.

6.6.4. Altura [FB6]

El presente FB tiene las siguientes entradas, salidas y variables estáticas.

	Nombre	po de datos	Valor predet.	Remanencia	Accesible d...	Escrib...	Visible en ..	Valor de a..
1	Input				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	<Agregar>				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	Output				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	Altura	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5	Altura_Ok	ool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6	InOut				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7	<Agregar>				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8	Static				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9	Limite_Sup_1	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
10	Limite_Inf_1	Real	0.0	No remane...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
11	Limite_Sup_2	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
12	Limite_Inf_2	Real	0.0	No remane...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
13	Sig_S1	ool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
14	Sig_S2	ool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Figura 58: Entradas y salidas “Altura”

Con respecto a la obtención de altura de la tolva, se programó un FB que obedece el siguiente orden:

- Primero para acceder al FB, el primer valor de los *arrays* “Historial_S1” y “Historial_S2” tiene que ser diferente de cero, de esta forma nos aseguramos que el historial del sensor S1 y S2 no se encuentra vacío, además de esta condición se tiene otra variable llamada “Altura_Ok” que se utiliza para saber cuándo la función ha acabado, en una primera instancia tiene que estar a FALSE.

```

IF (ABS("WinCC".Historial_Dist_Sl[5] - "WinCC".Historial_Dist_Sl[4]) <= 15) AND
  (ABS("WinCC".Historial_Dist_Sl[4] - "WinCC".Historial_Dist_Sl[3]) <= 15) AND
  (ABS("WinCC".Historial_Dist_Sl[3] - "WinCC".Historial_Dist_Sl[2]) <= 15) AND
  (ABS("WinCC".Historial_Dist_Sl[2] - "WinCC".Historial_Dist_Sl[1]) <= 15) AND
  (ABS("WinCC".Historial_Dist_Sl[1] - "WinCC".Historial_Dist_Sl[0]) <= 15) THEN

  #Limite_Sup_1 := "WinCC".Historial_Dist_Sl[2] + 1500;
  #Limite_Inf_1 := "WinCC".Historial_Dist_Sl[2] - 1500;

END_IF;

```

Figura 59: Código para obtención de límites

- Segundo, se obtuvo unos límites tanto superiores como inferiores. Como se puede ver en la figura 59, se calculó la diferencia de los últimos 5 datos del historial de los sensores, si estos tienen una diferencia menor de 15 mm significa que el sensor está midiendo la distancia a un objeto estático, por lo tanto, se toma referencia de ese objeto, con un límite superior de un metro y medio sobre nuestra referencia, también se obtiene un límite inferior que está a metro y medio por debajo de la referencia. Por lo tanto, obtenemos unos límites superiores e inferiores que cambiarán constantemente a la lectura de un objeto estático.

```

IF (#Limite_Inf_1 > "WinCC".Historial_Dist_Sl[0] OR #Limite_Sup_1 < "WinCC".Historial_Dist_Sl[0]) THEN
  #Sig_Sl := TRUE;
END_IF;

```

Figura 60: Código para la obtención de señal fuera de rango

- Tercero al entrar un nuevo dato dentro del buffer que no se encuentra dentro de este rango, significa que un objeto apareció frente a nuestro sensor, por lo tanto se dará la señal que tal sensor encontró un objeto fuera de rango, el presente código se puede ver en la figura 60, esto se aplicaría para ambos sensores.

```

IF (#Sig_S1=TRUE) AND (#Sig_S2=TRUE)

  #Altura := "WinCC".Historial_X_5[0];
  "WinCC".Altura := #Altura;
  #Altura_Ok := TRUE;

END_IF;

```

Figura 61: Código para la obtención de altura

Como se puede ver en la figura 61 se tiene que, si se tiene señal de fuera de rango en ambos sensores, significa que se entró dentro de la tolva, entonces se obtendrá la lectura del escáner, para ser más preciso del haz central del mismo, obteniendo la altura de la tolva. Esta altura se guarda dentro del DB "WinCC" y se da la señal que se obtuvo la altura, poniendo la variable "Altura_Ok" a TRUE.

6.6.5. Ancho [FB4]

El presente FB posee los siguientes entradas, salidas y variables estáticas.

	Nombre	Tipo de datos	Valor predet.	Remanencia
1	Input			
2	Req	Bool	false	No remane...
3	Output			
4	Esquina_Izq	LReal	0.0	No remane...
5	Esquina_Der	LReal	0.0	No remane...
6	Anchura	LReal	0.0	No remane...
7	Anchura_OK	Bool	false	No remane...
8	InOut			
9	<Agregar>			
10	Static			
11	Buffer_Anch	Array[0..10] of LReal		No remane...
12	Aux	LReal	0.0	No remane...
13	Aux_1	LReal	0.0	No remane...

Figura 62: Entradas y salidas “Ancho”

Para la obtención del ancho de la tolva se utilizaron las coordenadas cartesianas, obtenidas en el FB “Adaptacion_ñeñal”. Hay que tener en cuenta que para iniciar el presente FB se tiene que recibir en la entrada “Req” el valor TRUE, además de la presente ñeñal al igual que en “Altura” se tiene una variable para monitorizar cuando la obtención de la anchura haya finalizado, esta variable es llamada “Anchura_OK”.

```
#Aux := ABS("WinCC".Coordenadas_Y[0] - "WinCC".Coordenadas_Y[1]); //Se obtiene la diferencia en distancia
//en el eje Y de los haces de mas de la izquierda.

IF #Aux <= 7 THEN
    #Esquina_Izq := ("WinCC".Coordenadas_Y[0] + "WinCC".Coordenadas_Y[1]) / 2; //Calculo la media entre
//ambas distancias para obtener
//una esquina

END_IF;
```

Figura 63: Código para detección de esquina izquierda

En la figura 63 se observa, en la parte superior, la diferencia entre las coordenadas en Y de dos de los haces más lejanos del haz central, se utilizan estos haces ya que son los más probables que encuentren una pared, una vez realizada la diferencia entre ambos haces se pregunta si esta diferencia es menor a 7 mm, lo cual significaría que chocaron frente a una pared, entonces calculo la distancia a la esquina izquierda. De igual forma que se hizo para la esquina izquierda se hizo para la esquina derecha con la diferencia que se utilizaron los haces del otro extremo del escáner. QUITAR LO CORCHETES DE LAS UNIDADES.

```
IF #Esquina_Izq <> 0 AND #Esquina_Der <> 0 THEN
    "Buffer_Llenado"(Dato := #Esquina_Der - #Esquina_Izq,
    Buffer_llenado => #Buffer_Anch,
    Size := 10);

END_IF;
```

Figura 64: Código para llenado de buffer de anchuras

Una vez obtenidas ambas esquinas, se hace la comprobación de que ambas sean diferentes de 0, y de este modo se sabe que se han encontrado ambas esquinas. Sumando ambas distancias se obtiene el ancho que consiguientemente se llenara dentro del buffer de anchos de tolva obtenidos.

```

IF #Buffer_Anch[10] <> 0 THEN
    "Media"(Angulos := #Buffer_Anch,
            Media => #Anchura);
    "WinCC".Anchura := #Anchura;
    #Anchura_OK := TRUE;
END_IF;

```

Figura 65: Código para obtención de altura total

Por último para obtener el ancho total de la tolva, se espera que la última posición del buffer sea diferente de cero, esto significaría que el buffer se encuentra lleno y que se obtuvieron 11 medidas de ancho de la tolva. Posteriormente a través de una función llamada “Media” se obtiene la media de estos 11 datos y el ancho total, y se guarda el resultado en la variable “Ancho” en el DB “WinCC” y se activa la variable de “Anchura_OK”.

6.6.6. Longitud [FB3]

El presente FB posee los siguientes entradas, salidas y variables estáticas.

	Nombre	Tipo de datos	Valor predet.	Remanencia	Accesible d...	Escrib...	Visible en ..	Valor de a...
1	Input							
2	Req	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	Output							
4	Longitud_Ok	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5	Angulo_1	LReal	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6	Angulo_2	LReal	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
7	InOut							
8	<Agregar>							
9	Static							
10	Longitud_Inicial	LReal	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
11	Limite_Sup_2	LReal	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
12	Limite_Inf_2	LReal	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
13	Limite_Sup_1	LReal	0.0	No remane...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14	Limite_Inf_1	LReal	0.0	No remane...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15	Pendiente_S1_Y_1	LReal	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
16	Pendiente_S1_X_1	LReal	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
17	Pendiente_S1_X_2	LReal	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
18	Pendiente_S1_Y_2	LReal	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
19	Pendiente_S2_Y_1	LReal	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
20	Pendiente_S2_X_1	LReal	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
21	Pendiente_S2_X_2	LReal	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
22	Pendiente_S2_Y_2	LReal	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
23	Hist_Angulo_1	Array[0..10] of LReal		No remane...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
24	Hist_Angulo_2	Array[0..10] of LReal		No remane...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
25	Altura_Parcelada	LReal	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
26	C_1	Int	0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
27	C_2	Int	0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
28	Temp							

Figura 66: Entradas y salidas de “Longitud”

La longitud de la tolva fue el dato más dificultoso de obtener; esta dificultad radica en que no solamente se obtiene una medida, sino que también se determina el ángulo de inclinación

del chaflán de la tolva de camión, siempre y cuando este exista, si no fuera así el ángulo tiene que ser un valor despreciable.

Al igual que las anteriores funciones para el cálculo de medidas, se tiene unas condicionantes para que el bloque funcional empiece, se tiene una variable de entrada “Req”, además de la variable para determinar si ya se obtuvo la longitud de la tolva “Longitud_OK”.

```
IF (#Longitud_Inicial = 0) THEN

    #Longitud_Inicial := "WinCC".Historial_Dist_S1[0] + "WinCC".Historial_Dist_S2[0];
    #Pendiente_S1_Y_2 := "WinCC".Historial_Dist_S1[0];
    #Pendiente_S1_X_2 := "WinCC".Historial_X_5[0];

    #Pendiente_S2_Y_2 := "WinCC".Historial_Dist_S2[0];
    #Pendiente_S2_X_2 := "WinCC".Historial_X_5[0];

    #C_1 := 19;
    #C_2 := 19;
    #Altura_Parcelada := "Altura_DB".Altura / 20;

END_IF;
```

Figura 67: Código para obtención de longitud inicial

Como se puede ver en la figura 67, para obtener la longitud inicial de la tolva, se preguntará si la variable de “Longitud_Inicial” es cero, esta será cero en la primera entrada a la función. Esta fue programada de esta forma que la presente función se active cuando la altura haya sido obtenida, lo que significa que el mecanismo telescópico con los sensores se encuentra dentro de la tolva, por lo tanto la primera longitud obtenida será del borde de la tolva.

Después de obtener la longitud total de la tolva, que es la suma de las distancias obtenidas por sensor S1 y S2, se guardarán cuatro valores para el cálculo posterior de la pendiente, así mismo se declaran dos valores “C_1” y “C_2” a 19 que serán utilizados posteriormente, luego se declara la variable “Altura_parcelada”, que es la altura total de la tolva partida entre 20.


```
IF (#Altura_Parcelada * #C_1) >= "WinCC".Historial_X_5[0] THEN

    #Pendiente_Sl_Y_1 := "WinCC".Historial_Dist_Sl[0];
    #Pendiente_Sl_X_1 := "WinCC".Historial_X_5[0];

    "Pendiente"(X1 := #Pendiente_Sl_X_1,
                X2 := #Pendiente_Sl_X_2,
                Y1 := #Pendiente_Sl_Y_1,
                Y2 := #Pendiente_Sl_Y_2,
                Angulo => #Angulo);

    "Buffer_Llenado"(Dato := #Angulo,
                    Buffer_llenado => #Hist_Angulo_1,
                    Size := 10);

    #C_1 := #C_1 - 1;

END_IF;
```

Figura 68: Código de cálculo de pendiente

En la figura 68 se tiene el código para obtener las diferentes pendientes, primero, se tiene una condicionante, que pregunta si la altura actual calculada es menor a la altura parcelado multiplicada por “C_1” que en primer caso será 19, lo que significa que este “if” se ejecutará cuando el mecanismo se encuentre en la decimonovena parte de la altura total de la tolva.

Consiguientemente se obtendrá otro punto de la pendiente, obteniendo los dos puntos necesarios para calcular una pendiente entre dos puntos y aún más importante el ángulo entre los mismos. A esta función se llama “Pendiente”, que nos devuelve el ángulo entre dos puntos, posteriormente se llena el buffer con el nuevo ángulo obtenido, después se rebaja en una unidad la variable “C_1”, por lo que en una siguiente pasada del código se evaluara el “if” para la decimooctava parte de la altura total de la tolva, y este ciclo se repetirá hasta que se llegue a la mitad de la altura total de la tolva, obteniendo 10 ángulos y subiendo el mecanismo a su posición inicial.

```

IF ("Altura_DB".Altura / 2) >= "WinCC".Historial_X_5[0] THEN

  "Media"(Angulos := #Hist_Angulo_1,
           Media => #Angulo_1);
  "Media"(Angulos := #Hist_Angulo_2,
           Media => #Angulo_2);

  IF #Angulo_1 < 5 THEN
    #Angulo_1 := 0;
  END_IF;
  IF #Angulo_2 < 5 THEN
    #Angulo_2 := 0;
  END_IF;

  "WinCC".Longitud := #Longitud_Inicial;
  "WinCC".Angulo_1 := #Angulo_1;
  "WinCC".Angulo_2 := #Angulo_2;

END_IF;

```

Figura 69: Código de ángulo y longitud obtenida

En la figura 69, se tiene una condicionante, que pregunta si la altura actual es menor o igual a la mitad de la altura total, si este es el caso se obtendrán las medias de ambos ángulos y si estos ángulos son menores de 5 grados, tomaran el valor de cero, esto se realizó a causa de que se calculaban ángulos residuales que afectaban a un cálculo preciso del volumen.

Aunque la longitud inicial y el ángulo sean obtenidos, se tiene que dar la orden para salir de la tolva, con este objetivo se aplicó el mismo método que para entrar en la tolva, se obtuvo unos límites superiores e inferiores, si estos cambian bruscamente significa que el mecanismo de medida se salió de la tolva de camión.

6.6.7. Volumen [FC4]

Para la presente función se tiene las siguientes entradas y salidas.

	Nombre	Tipo de datos	Valor predet.
1	Input		
2	Orden	Bool	
3	Angulo_1	LReal	
4	Angulo_2	LReal	
5	Output		
6	Volumen_Total	LReal	
7	Tolva_Octaedro	Bool	
8	Tolva_1_Chaffan	Bool	
9	Tolva_2_Chaffan	Bool	

Figura 70: Entradas y Salidas "Volumen"

Para la obtención del volumen de la tolva se desarrolló una función en SCL, la cual obtendrá como parámetros las medidas de altura, ancho y longitud. Antes de explicar cómo se obtuvo el volumen, se tiene que indicar que se hizo una normalización de las medidas obtenidas,

esto se debe a que en el cálculo la altura y longitud, se despreció la diferencia de distancias entre el sensor laser y el escáner.

Dicho lo anterior, se hizo el cálculo del volumen de la tolva, el presente cálculo varía si esta tiene un chaflán o dos, por lo tanto son excluyentes los cálculos unos de otros.

```
IF #Orden=TRUE AND #Angulo_1=0 AND #Angulo_2=0 THEN //  
    #Volumen_Total := "Anchura_DB".Anchura * #Real_Altura* #Real_Longitud;  
    #Tolva_Octoedro := TRUE;  
    "WinCC".Volumen := #Volumen_Total;  
    "WinCC".OK_Volumen := TRUE;  
    "Entrada_Camion_DB".Entro_Carro := FALSE;  
    "RESET_DB"(Req := TRUE);  
    "RESET_DB"(Req := FALSE);  
END_IF;
```

Figura 71: Código de cálculo de volumen de un ortoedro

En la parte superior de la figura 71 se tiene una condicionante de orden, que será activada si el cálculo del ancho y la longitud fueron obtenidas, además se preguntará si los ángulos obtenidos son cero, si este es el caso se calcula el volumen como un ortoedro.

El volumen se obtendrá con el producto de la longitud, el ancho y altura, después se manda una señal que la tolva tiene forma de ortoedro y se guarda el resultado del volumen en la variable "Volumen" del DB "WinCC", después se pone a FALSE la variable "Entro_Carro" esto se hace para el programa general que será explicado posteriormente, por último se llama a la función "RESET_DB" que resetea todos los valores de los DB de "Altura", "Ancho" y "Longitud", de esta forma se tienen listos los DB para un nuevo uso.

```

IF #Orden = TRUE AND #Angulo_1 <> 0 AND #Angulo_2 = 0 THEN

  #Angulo_1_rad := #Angulo_1 * (3.1416 / 180);
  #X := #Real_Altura * TAN(#Angulo_1_rad);
  #L_C := #Real_Longitud - #X;
  #Volumen_1 := #L_C * "Anchura_DB".Anchura * #Real_Altura;

  #Volumen_2 := ((#X * #Real_Altura) / 2) * "Anchura_DB".Anchura;

  #Volumen_Total := #Volumen_1 + #Volumen_2;

  #Tolva_1_Chaflan:=TRUE;

  "WinCC".Volumen := #Volumen_Total;

  "WinCC".OK_Volumen := TRUE;
  "Entrada_Camion_DB".Entro_Carro := FALSE;

  "RESET_DB"(Req := TRUE);
  "RESET_DB"(Req := FALSE);

END_IF;
  
```

Figura 72: Código de cálculo de volumen con un chaflán

Si se tiene una tolva con un chaflan, el sistema detectará que se tiene un ángulo, y en consecuencia se activa el código que implementa el código para el cálculo del volumen de la figura 72. Para obtener el volumen de esta forma primeramente se dividió el volumen en dos partes, por un lado un ortoedro (color azul) y por otro lado un prisma triangular (color rosa).

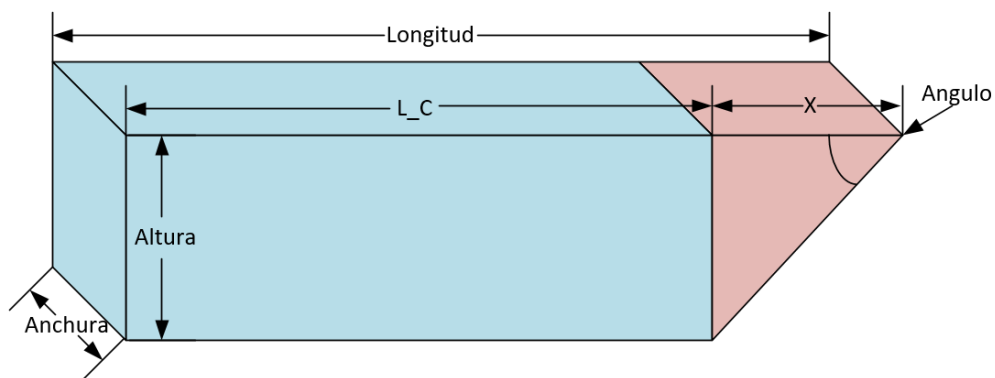


Figura 73: Modelo para cálculo de volumen para tolva con un chaflán

Como se puede ver en la figura 73. Para dividir ambos campos se utilizó el ángulo anteriormente calculado, aplicando trigonometría básica se utilizó la función tangente para calcular "X", que en si es un subconjunto de la longitud calculada, después se obtuvo "L_C" restando de la longitud total "X", como resultado ya se tienen todos los datos para obtener el volumen total.

Para llegar a tal objetivo se dividió el cálculo de volumen en dos partes que consiguientemente se sumarán, por un lado para tener el volumen del ortoedro se multiplicó

la “Altura”, “Ancho” y “L_C”, por otro lado se obtuvo el área del prisma triangular formado por el chaflán, se aplicó la fórmula de área de un triángulo y posteriormente para obtener el volumen se multiplicó por el ancho del camión, obteniendo el volumen restante, después se suman ambos volúmenes y se obtiene el volumen total, se copia el valor del volumen calculado en el DB “WinCC”, se resetea la variable “Entro_Carro” y se hace el *reset* de los DB.

Hay que hacer énfasis que también se aplicó el mismo código, pero para un chaflán que haya sido detectado por el sensor S2, este caso se aplicaría cuando se encuentre chaflán en el otro lado de la tolva

```

IF #Orden = TRUE AND #Angulo_1 <> 0 AND #Angulo_2 = 0 THEN

    #Angulo_1_rad := #Angulo_1 * (3.1416 / 180);
    #X := #Real_Altura * TAN(#Angulo_1_rad);
    #L_C := #Real_Longitud - #X;
    #Volumen_1 := #L_C * "Anchura_DB".Anchura * #Real_Altura;

    #Volumen_2 := ((#X * #Real_Altura) / 2) * "Anchura_DB".Anchura;

    #Volumen_Total := #Volumen_1 + #Volumen_2;

    #Tolva_1_Chaflan:=TRUE;

    "WinCC".Volumen := #Volumen_Total;

    "WinCC".OK_Volumen := TRUE;
    "Entrada_Camion_DB".Entro_Carro := FALSE;

    "RESET_DB"(Req := TRUE);
    "RESET_DB"(Req := FALSE);

END_IF;

```

Figura 74: Código de cálculo de volumen con un chaflán

Al igual que para el cálculo de volumen con un chaflán, para el cálculo de volumen con dos chaflanes se dividió el cálculo de volumen en tres partes.

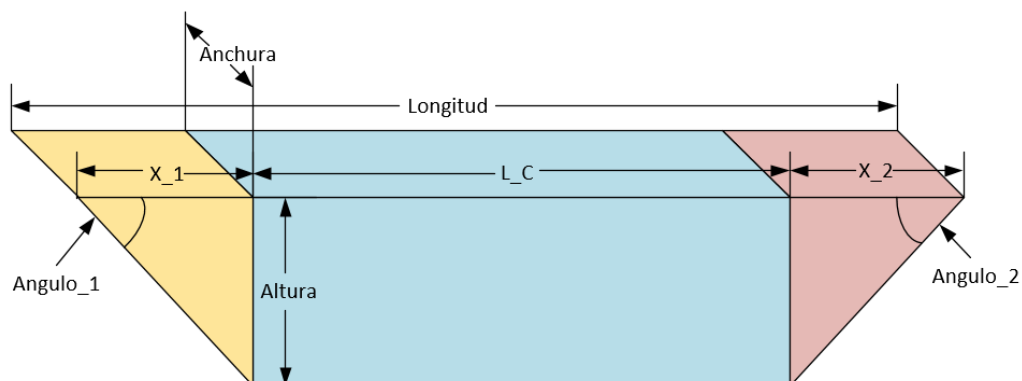


Figura 75: Modelo para cálculo de volumen para tolva con dos chaflanes

Como se puede ver en el modelo, nuestro objeto fue dividido en tres partes. Para su cálculo se utilizó la función tangente para así obtener los valores de “X_1”, “X_2” y “L_C”, por lo tanto, una vez obtenidos estos tres datos, se hizo el cálculo del volumen para cada una de las partes. Para el ortoedro se calculó el volumen multiplicando los vértices calculados, mientras que para la zona amarilla y roja se calculó primero sus respectivas áreas de triángulos y se multiplicaron por el ancho de la tolva. Obtenidos los tres volúmenes se suman y se obtiene el volumen total, como en los anteriores casos se copia el valor del volumen obtenido al DB “WinCC”, se da la señal de un ortoedro con dos chaflanes y se resetean los DB.

6.6.8. Movimiento de motor [FB8]

Para la presente función se tiene las siguientes entradas, salidas y variables estáticas.

	Nombre	Tipo de datos	Valor predet.	Remanencia
1	Input			
2	Req	Bool	false	No remane...
3	Output			
4	Adelante_motor	Bool	false	No remane...
5	Atras_motor	Bool	false	No remane...
6	InOut			
7	<Agregar>			
8	Static			
9	Paso	Int	0	No remane...
10	Fin_motor	Bool	false	No remane...

Figura 76: Entradas y salidas “Movimiento_de_motor”

Para el control de movimiento del motor, se realizó una función en SCL, al realizar un nuevo bloque de función el movimiento es independiente del cálculo de medidas de la tolva.

```

IF (#Req = TRUE AND #Fin_motor = FALSE) THEN
  IF (#Paso = 0) THEN
    #Paso := 1;
  END_IF;
  CASE #Paso OF
    1:
      #Adelante_motor := TRUE;
      #Atras_motor := FALSE;
      IF ("Altura_DB".Altura <> 0) THEN
        IF ("Altura_DB".Altura / 2 >= "WinCC".Historial_X_5[0]) THEN
          #Paso := #Paso + 1;
        END_IF;
      END_IF;

    2:
      #Atras_motor := TRUE;
      #Adelante_motor := FALSE;
      IF ("Longitud_DB".Longitud_Ok = TRUE) THEN
        #Paso:=#Paso+1;
      END_IF;

    3:
      #Atras_motor := FALSE;
      #Adelante_motor := FALSE;

      IF ("Anchura_DB".Anchura_OK = FALSE) THEN
        #Paso := 0;
        "Longitud_DB".Longitud_Ok := FALSE;

      ELSE
        #Fin_motor := TRUE;
      END_IF;

  ELSE
    ;
  END_CASE;
ELSE
  #Atras_motor := FALSE;
  #Adelante_motor := FALSE;
END_IF;

```

Figura 77: Código para movimiento de motor

Para el movimiento de motor se hizo el código de la figura 77, que funciona de forma secuencial. Primero, para que el código se ejecute este tiene que tener una indicación de requisito “Req” para que se muevan los motores y que no haya terminado el funcionamiento de la presente función controlado por la variable “Fin_Motor”. Posteriormente como se muestra en el código se dará el valor “1” a la variable “Paso”, esto se hace de forma que se active la opción “1” de la instrucción “Case”, en el presente paso se da la orden que se active el motor adelante, lo que hace que el mecanismo baje hasta que se encuentre a la mitad de la altura, una vez llegado a la mitad de la altura, se sumará una unidad a la variable “Paso” lo que hará que se active la opción “2”, que activa que el motor vaya hacia atrás, retrayendo el mecanismo. Esta opción seguirá activa hasta que el bloque de función de “Longitud” haya terminado, si ese es el caso entonces se desactivara el motor, posteriormente se tiene una condicionante que pregunta si el bloque de función “Ancho” ha terminado, si ese es el caso la variable “Paso” se pondrá a “0”, si ese no es el caso se vuelve a declarar que “Paso” sea “1”, esto se debe a que si no se encontraría el ancho de la tolva el mecanismo volvería a bajar y subir hasta que se encuentre el ancho.

6.6.9. Gestión de Volumen [FC11]

La presente función solamente tiene una entrada “Req” de requerimiento de la función y una salida “Resultado” donde se almacenará el valor del volumen calculado.

Una vez obtenidas las medidas de la tolva, se necesita de una función que integre todas las funciones ya programadas. Por tanto, se desarrolló esta función programada en lenguaje de contactos, ya que solamente se tiene que gestionar las entradas y salidas de las funciones.

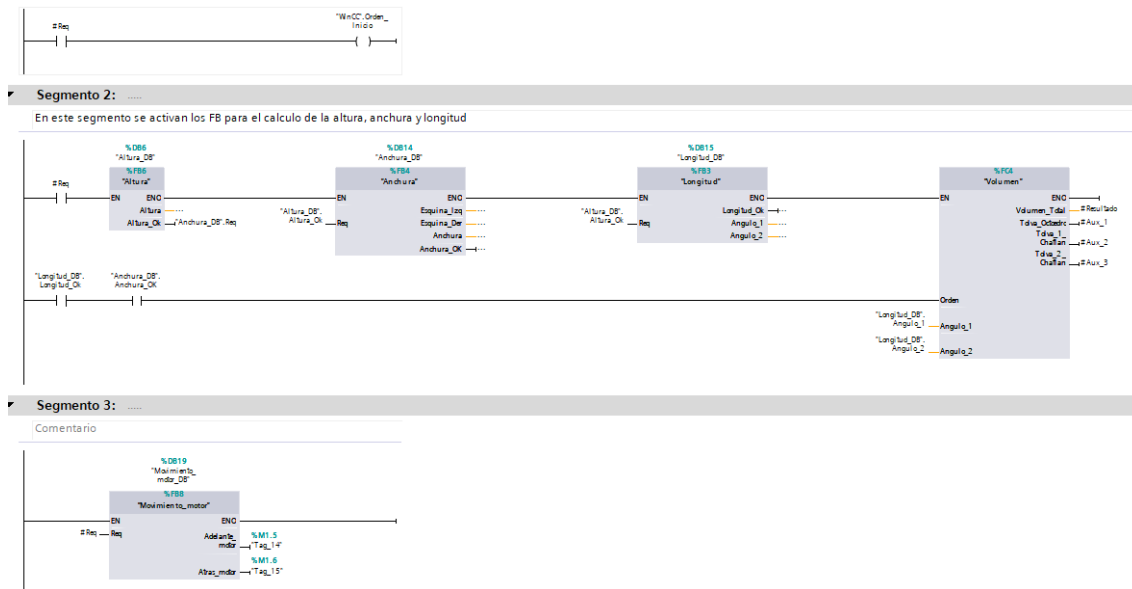


Figura 78: Código Gestión Volumen

Como se puede ver en la figura 78 se tiene la programación en lenguaje escalera, la presente función posee una entrada llamada “Req” que se utiliza para empezar el programa.; en los tres segmentos programados se puede encontrar en esta variable declarada.

En el primer segmento se activa la variable “Req”, para que se ponga a TRUE la variable “Orden_Inicio”. Esta es una variable global en el DB “WinCC”, que activa la comunicación con el escáner LED, posteriormente se utiliza el variable “Req” para activar la función “Altura”, si la función finaliza activará las funciones “Ancho” y “Longitud”. Por debajo para activar la función “Volumen” se necesita que la función “Longitud” y “Ancho” finalicen, además se reciben los ángulos calculados por longitud.

En el último segmento se llama a la función de “Movimiento_motor”, que al igual que anteriores segmentos dispone de la variable “Req”, esto significa que una vez que esté activa la función “Gestion_de_volumen” la función “Movimiento_motor” activará el motor del mecanismo, hasta que finalice su procedimiento.

6.6.10. Sensor radar [FC10]

El sensor radar genera una señal de corriente en el rango 4 a 20 mA, por lo tanto, al igual que los sensores de distancia, se utilizó una entrada analógica de la tarjeta del PLC ya descrita. Sin embargo, a diferencia de los sensores ópticos en este caso no se precisan valores históricos sino solo una lectura cuando se requiera obtener el valor de la altura de la tolva.

6.6.11. Bascula [FC13] [FC14]

Para el uso de la báscula no se tiene información detallada. Sin embargo, se tiene constancia de que se deberá realizar una comunicación entre la báscula y el sistema de control propuesto. Por lo tanto, se prepararon dos FC, uno con los bloques de comunicación “Send_P2P” y “Recieve_P2P” y otra función que utilice el anterior FC y gestione los datos obtenidos. Finalmente se desea que los FC de gestión nos devuelva un REAL, que será utilizado por el FB “Dosificado_automático”.

6.6.12. Lector de matrícula [FC15] [FC20]

Con respecto a la lectura de la matrícula para obtener el peso permitido por camión, no se tiene información a priori totalmente definida, lo que si se entiende es que de la lectura de la matrícula se hará una relación en una base de datos, que tendrá el peso permitido por camión. Para obtener tal información se prevén dos funciones, la primera mandará el dato de matrícula y la segunda recibirá el peso permitido por la tolva en una variable de tipo REAL.

6.6.13. Dosificado Automático [FB5]

Para el presente FB se tienen las siguientes entradas, salidas y variables estáticas.

	Nombre	Tipo de datos	Valor predet.	Remanencia
1	Input			
2	Volumen_Obtenido	Bool	false	No remane...
3	Volumen_tolva	LReal	0.0	No remane...
4	Distancia_maxima	LReal	0.0	No remane...
5	ESPERA_TEMP	Bool	false	No remane...
6	Final_carrera_trasero	Bool	false	No remane...
7	Output			
8	Fin_Dosificado	Bool	false	No remane...
9	Cinta_adelante	Bool	false	No remane...
10	Cinta_atras	Bool	false	No remane...
11	Dosificar	Bool	false	No remane...
12	InOut			
13	<Agregar>			
14	Static			
15	Peso_permitido	String	"	No remane...
16	Peso_per_real	Real	0.0	No remane...
17	Primer_peso	LReal	0.0	No remane...

Figura 79: Entradas y salidas “Dosificado automático”

Para el dosificado automático en la tolva, se realizó un FB en el cual se gestionarán las entradas del radar, la báscula y el lector de matrícula, además de hacer el control de los motores para el movimiento de la cinta transportadora y para el movimiento de traslación de dicha cinta.

```

IF #Volumen_Obtenido = TRUE AND #Fin_Dosificado=FALSE THEN
  "Sensor_Radar" (Resultado => #Sensor_radar);
  "Adaptacion_Bascula" ("Resultado" => #Peso_Actual);
  "Peso_Obtenido" ("Resultado" => #Peso_per_real);

  IF (#Primer_peso = 0) THEN
    #Primer_peso := #Peso_Actual;
  END_IF;

  #Volumen_Actual := (#Peso_per_real - #Primer_peso) * #Peso_Especifico;

  IF #Sensor_radar >= #Distancia_maxima THEN

    #Cinta_adelante := FALSE;
    #Cinta_atras := FALSE;
    #Dosificar := TRUE;

  ELSE
    #Dosificar := FALSE;
    #Cinta_adelante := TRUE;
  END_IF;

  IF #Peso_Actual >= #Peso_per_real OR #Volumen_Actual >= #Volumen_tolva THEN

    #Dosificar := FALSE;
    #Cinta_atras := TRUE;
    IF (#Final_carrera_trasero = TRUE) THEN
      #Cinta_atras := FALSE;
      #Fin_Dosificado := TRUE;
      "WinCC".OK_Volumen := FALSE;
      #Primer_peso := 0;
    END_IF;

  END_IF;

ELSE
  ;
END_IF;

```

Figura 80: Código de dosificado automático

Como se muestra en la figura 80, se tiene el código de dosificado automático, que comenzará si el volumen ya fue obtenido y siempre que no se haya completado la presente función; luego obtenemos la lectura del radar, la lectura de la báscula y el valor del peso permitido.

Entonces se tiene la condicionante de si existe un primer pesaje, se guarda en la variable “*Primer_peso*”, esto se hace de forma que se obtenga el peso del camión sin material. Una vez obtenido el peso del camión, se calculará el peso con material restándole el peso del camión, así se obtendrá el peso del material dosificado. Posteriormente se tiene el cálculo del “*Volumen_Actual*” que se hace a través de un producto entre peso del material y la constante “*Peso_especifico*”.

El presente cálculo de constante de peso específico se hará en pruebas de campo, se concebirá con varias pruebas con un peso controlado, para así obtener una curva de regresión y tener el valor de peso específico.

Posteriormente se tiene la condicionante de lectura de distancia del radar, que consiste en que si la distancia obtenida por el radar es mayor al valor especificado se hace el dosificado del material; de otra forma se para el dosificado y se mueve la cinta adelante, cuando se tenga una distancia mayor a la permitida se parará la cinta y se volverá a dosificar.

Por otro lado, el programa terminará si se cumplieran una de dos condicionantes, una es que el peso actual pase al permitido por la placa del camión, o que el volumen de la tolva se haya alcanzado. Si se cumplieran alguna de estas condiciones se parará el dosificado y se retraerá la cinta; por último si la cinta se retrae y se detecta el final de carrera, se desactivarán todos los motores, se reseteará el valor del peso del camión y se dará la señal de que se finalizó la función.

6.6.14. Entrada de camión [FB2]

Para el presente FB se tienen las siguientes entradas, salidas y variables estáticas.

	Nombre	Tipo de datos	Valor predet.	Remanencia
1	Input			
2	PULSE_IN	Bool	false	No rem...
3	PULSE_OUT	Bool	false	No remane...
4	Output			
5	<Agregar>			
6	InOut			
7	<Agregar>			
8	Static			
9	EN_PULS	Bool	false	No remane...
10	OUT_PULS	Bool	false	No remane...
11	Entro_Carro	Bool	false	No remane...

Figura 81: Entradas y salidas “Entrada_de_camion”

Para la entrada del camión a la báscula, se utilizó un sensor de barrera anteriormente mencionado en el segundo capítulo, razonablemente se tiene un bloque de función asociada a esta entrada, la presente función se explicará a continuación.

```

IF #PULSE_IN = TRUE THEN
    #EN_PULS := TRUE;
END_IF;

IF #PULSE_OUT = TRUE THEN
    #OUT_PULS := TRUE;
END_IF;

IF #EN_PULS = TRUE AND #PULSE_IN = TRUE THEN
    #EN_PULS := FALSE;
    #OUT_PULS := FALSE;
    "WinCC".Carro := TRUE;
    #Entro_Carro := TRUE;
    "Salida_Camion_DB".Salio_Camion := FALSE;
END_IF;

```

Figura 82: Código de entrada de camión

Como se puede ver en la figura 82, el presente código espera por pulsos de los dos sensores de barrera. Si existe un flanco ascendente en ambos sensores de barrera, entonces se indicará que existe un camión en la báscula, al igual que anteriores funciones ponemos la variable en este caso “Camion” en TRUE indicando que existe un camión y una variable de “Entro_Carro” a TRUE para indicar que se ejecute la siguiente acción. Por consiguiente se tiene la variable “Salio_Camion” a FALSE para desactivar la acción anterior.

6.6.15. Salida de camión [FB7]

Para el presente FB se tienen las siguientes entradas, salidas y variables estáticas.

	Nombre	Tipo de datos	Valor predet.	Remanencia
1	Input			
2	PULSE_IN	Bool	false	No rem...
3	PULSE_OUT	Bool	false	No remane...
4	Output			
5	<Agregar>			
6	InOut			
7	<Agregar>			
8	Static			
9	OUT_PULS	Bool	false	No remane...
10	IN_PULS	Bool	false	No remane...
11	Salio_Camion	Bool	false	No remane...

Figura 83: Entradas y salidas “Salida_de_camion”

De la misma forma que en caso de entrada de un camión, se desarrolló un programa para cuando el camión sale de la báscula.

```

IF #PULSE_IN = FALSE THEN
    #IN_PULS := TRUE;
END_IF;
IF #PULSE_OUT = FALSE THEN
    #OUT_PULS := TRUE;
END_IF;

IF #OUT_PULS = TRUE AND #PULSE_OUT = TRUE THEN
    #IN_PULS := FALSE;
    #OUT_PULS := FALSE;
    "WinCC".Carro := FALSE;
    #Salio_Camion := TRUE;
    "Dosificado_Automatico_DB".Fin_Dosificado := FALSE;
END_IF;

```

Figura 84: Código de salida de camión

Como se puede ver en la figura 84 se tiene el código para la salida del camión. El presente código interroga si existe un flanco de bajada en ambos sensores de barrera, si este es el caso entonces el camión se considera que salió de la báscula, y se da la orden que el camión salió

del recinto poniendo “*Salio_Camion*” a TRUE, por otro lado, se pone a FALSE la variable “*Carro*” indicando que no se encuentra el camión y por último se resetea la variable “*Fin_Dosificado*” permitiendo que la acción anterior se desactive.

6.6.16. Principal [FC12]

Para poder integrar todas las funciones anteriormente realizadas, se hizo una función que gestione todo el proceso, de forma que cumpla con todas las exigencias del Grafcet principal.

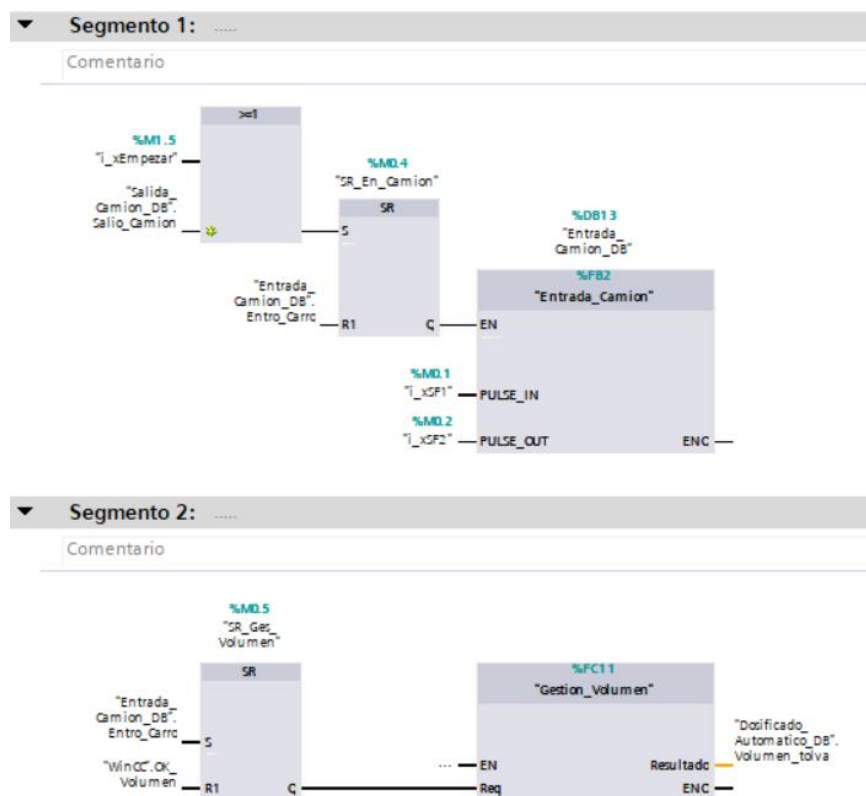


Figura 85: Primera parte de la función Principal

Como se puede ver en la figura 85, esta función Principal está realizada en lenguaje de bloques. Esto se hizo de esta forma para que se puedan manejar mejor las entradas y salidas de cada FB o FC, además de esta forma que se pueda ver en conexión online los resultados de cada bloque.

En el primer segmento se tiene un bloque SR conectado a la entrada “*i_xEmpezar*” del FB “*Entrada Camion*”. El bloque SR que es de preferencia al Reset se utiliza para activar el bloque de función, como se puede ver el bloque SR tiene su entrada al Set con un OR entre “*Salio_Camion*” y el pulsador “*i_xEmpezar*” que es una entrada de nuestra función. Por otro lado la entrada Reset tiene conectada la variable “*Entro_Carro*” del DB de instancia del FB “*Entrada_Camion*”, por lo tanto esto significa que cuando un camión entre en el recinto, este bloque se desactivará.

Posteriormente cuando el FB de “Entrada_Camion” se desactive, se activará el de “Gestión volumen” y al igual que con “Entrada_Camion” cuando “Gestión_Volumen” acabe mandará la señal a la variable “OK_Volumen” desactivando su propio FB y activando la siguiente acción.

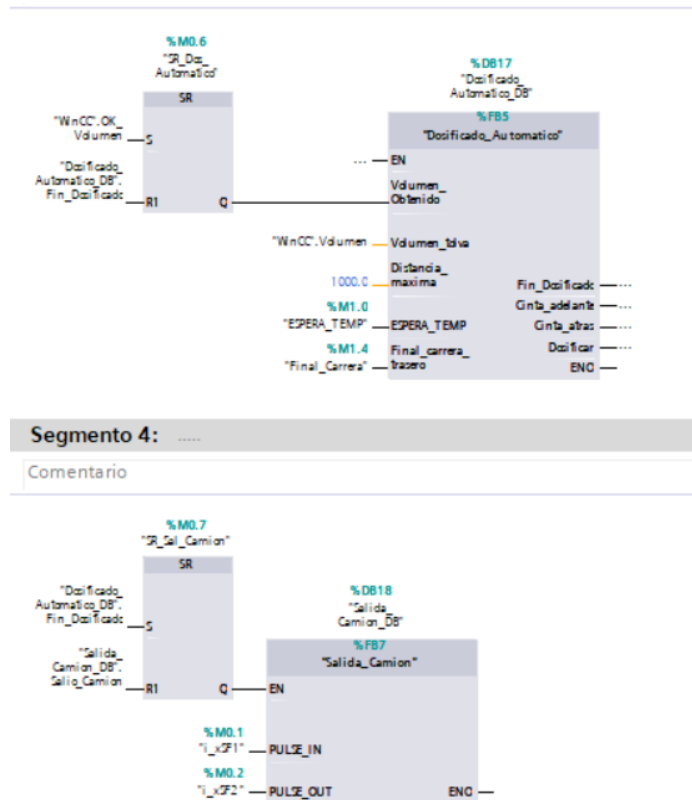


Figura 86: Segunda parte de función principal

Luego al activarse la variable “OK_Volumen”, activamos el FB de “Dosificado_Automatico”, como se puede ver el bloque de dosificado automático tiene como parámetros, el volumen tolva, ESPERA_TEMP que es una marca auxiliar, la distancia máxima que el sensor radar puede medir, antes que se mueva la cinta transportadora, además se tiene la entrada del final de carrera, que dará un valor de alto lógico cuando la cinta transportadora se ha retraído completamente. Si el FB a acabo entonces pondrá a TRUE la variable “Fin_Dosificado”, por lo que desactivará el FB de “Dosificado_Automatico” y activará la acción “Salida_Camion”.

Como se puede ver al igual que en el FB “Entrada_Camion”, en “Salida_Camion” también se tiene como parámetro de entrada las dos entradas digitales de los sensores de barrera. El bloque de función “Salida_Camion”, cuando termine la función pondrá a TRUE la variable “Salio_Camion”, desactivando el bloque de función “Salida_Camion” y reiniciando todo el ciclo.

6.6.17. Salidas [FC17]

Para gestionar mejor las salidas del sistema se desarrolló una función tomando en cuenta los tres motores que se manejan:

- Motor de activación de la cinta transportadora para dosificación.
- Motor para movimiento del mecanismo con los sensores.
- Motor para el desplazamiento derecha/izquierda de la cinta de dosificación.

La empresa MEYME en las que se realizaron las prácticas posee un FB propio para el manejo de motores que contempla los modos de funcionamiento y las posibles alarmas.

Actualmente los motores de la cinta transportadora se manejan en modo manual y local a través de la botonera anteriormente mostrada, pero en el trabajo que se presenta se manejará de forma automática y remota atendiendo a los nuevos requerimientos y con llamadas al FB mencionado.

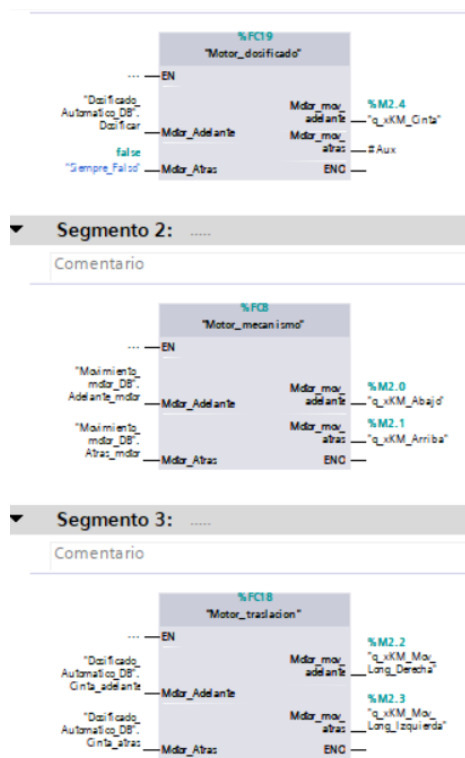


Figura 87: Código de salidas

Como se puede ver en la figura 87, se tiene una función llamada "Motor_dosificado", que es una función que gestiona el FB desarrollado por la empresa MEYME S.A. Por lo que se conectó las salidas de este FB, a la función de "Motor_dosificado", así poder visualizar las salidas del FB y conectarlas a los contactores del motor.

También hay que denotar que en la función FC17 “*Salidas*” se hizo una función por cada motor, así poder gestionar varios FBs de motor MEYME, como sean necesarios. Por recomendación de MEYME S.A. el contenido de los FCs mostrados en la figura 87, son privados ya que muestran información de su FB de motores.

7. MIGRACIÓN A PCS7

Por lo que se refiere a la migración de la aplicación desarrollada para Tia Portal al sistema de control distribuido PCS7 ambos de Siemens, se realizaron los siguientes pasos:

- Se generó el código fuente con estructura FCs y FBs en SCL para Tia Portal.
- Se creó una librería en PCS7 y se importaron los módulos anteriores.
- Se modificó el código PCS7 previamente importado.
- Se asignaron números de FC y FB libres en el PLC.
- Se volvieron a desarrollar algunos FBs en PCS7 que no se pudieron migrar.
- Se integró todo en dos programas nuevos realizado en lenguaje de programación CFC en la aplicación PCS7.

Una vez realizados todos los pasos mencionados, se da por concluida con la migración a PCS7 a falta de la puesta en marcha de la instalación que no se ha podido completar por la planificación del proyecto.

6.7. Generación de fuentes SCL en Tia Portal

Con el objetivo de que la migración de Tia Portal a PCS7 sea más fácil, se generaron fuentes de todos los bloques programados en SCL de la siguiente manera.

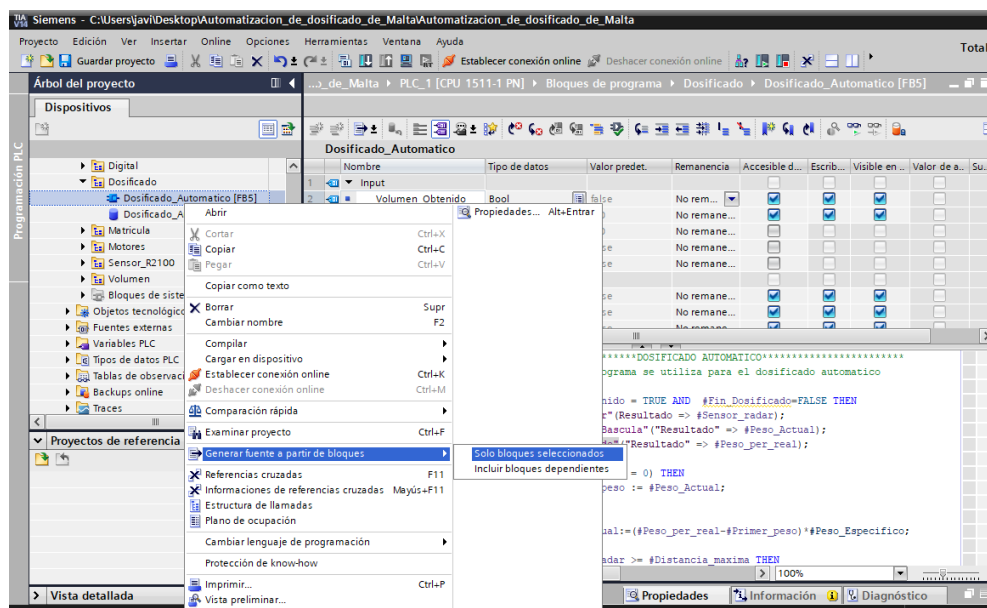


Figura 88: Ejemplo de generación de fuente

Se estableció una carpeta con todas las fuentes generadas, exceptuando algunos FB y FC.

Ordinal	Nombre de programa	Nro de FB o FC	Lenguaje de programación	Migración
1	Principal	FC12	FUP	No
2	Salidas	FB17	FUP	No
3	Sensor_Radar	FC10	SCL	Si
4	Sensor_S1	FC7	SCL	Si
5	Sensor_S2	FC9	SCL	Si
6	Buffer_Llenado	FC2	SCL	Si
7	Media	FC5	SCL	Si
8	Pendiente	FC3	SCL	Si
9	RESET_DB	FC6	SCL	Si
10	Adaptación_Báscula	FC14	SCL	Si
11	Comunicación_Báscula	FC13	FUP	Si
12	Entrada_Camion	FB2	SCL	Si
13	Salida_Camión	FB7	SCL	Si
14	Dosificado_Automático	FB5	SCL	Si
15	Lectura_Matricula	FC15	SCL	Si
16	Peso_Obtenido	FC20	SCL	Si
17	Motor_dosificado	FC19	FUP	No
18	Motor_mecanismo	FC8	FUP	No
19	Motor_traslación	FC18	FUP	No
20	Comunicación_Escáner	FB1	FUP	No
21	Adaptación_Sensor	FB1	SCL	Si
22	Gestion_Volumen	FC11	KOP	No
23	Volumen	FC4	SCL	Si
24	Altura	FB6	SCL	Si
25	Ancho	FB4	SCL	Si
26	Longitud	FB3	SCL	Si
27	Movimiento_motor	FB8	SCL	Si

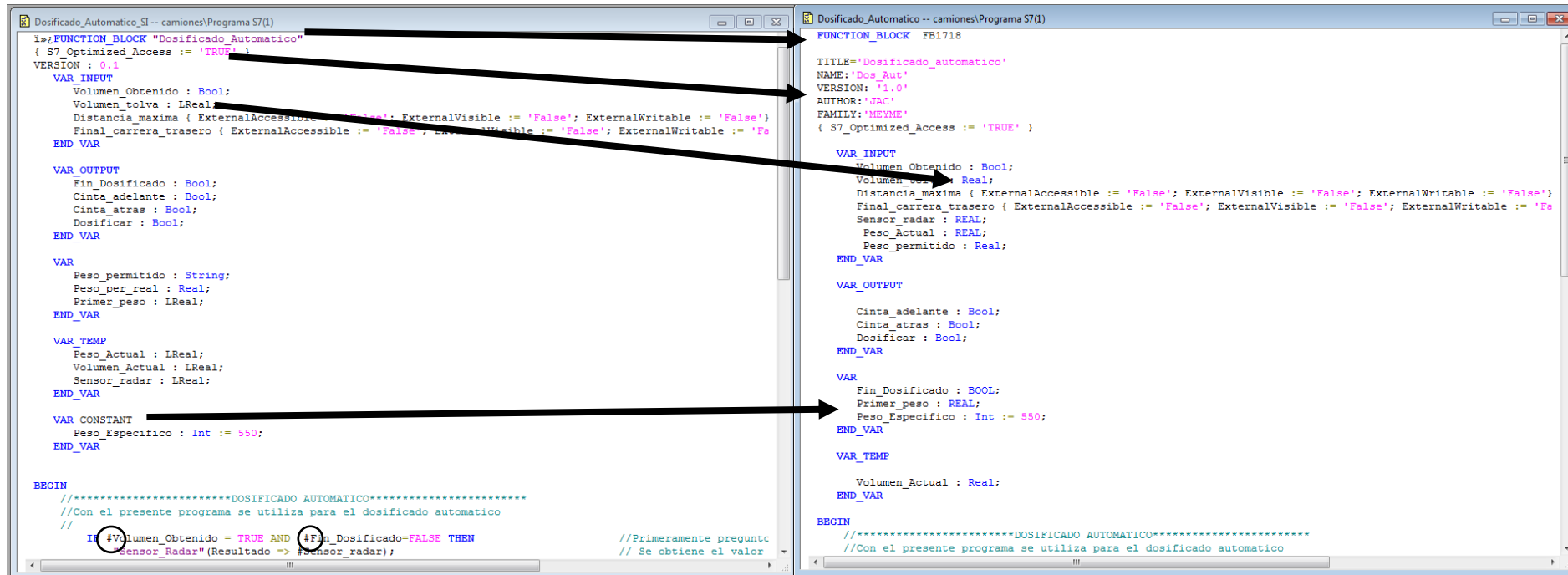
Tabla 5: Generación de fuentes

Como se puede ver en la tabla 5, se pudieron migrar casi todas las funciones y bloques de funciones del proyecto, excepto los que no fueron programados en SCL. Esto conlleva algunas dificultades ya que algunos de nuestros bloques más importantes fueron programados en FUP y KOP, por lo tanto, se tendrán que volver a programar en la herramienta PCS7.

6.8. Modificación de código

Previamente a la modificación de código se abrió una nueva librería en PCS7, dentro del multiproyecto de la planta de San Adrián, posteriormente se importaron todas las fuentes obtenidas de Tia Portal.

Para cada una de las funciones, así como para los bloques funcionales, se tuvieron que hacer algunos cambios en programación, de la programación en Tia Portal y PCS7, los cambios en su gran mayoría fueron genéricos, y se explicará a continuación



```

1>FUNCTION_BLOCK "Dosificado Automatico"
{ S7_Optimized_Access := 'TRUE' }
VERSION : 0.1
VAR_INPUT
  Volumen_Obtenido : Bool;
  Volumen_tolva : LReal;
  Distancia_maxima { ExternalAccessible := 'False'; ExternalVisible := 'False'; ExternalWritable := 'False'; }
  Final_carrera_trasero { ExternalAccessible := 'False'; ExternalVisible := 'False'; ExternalWritable := 'False'; }
END_VAR

VAR_OUTPUT
  Fin_Dosificado : Bool;
  Cinta_adelante : Bool;
  Cinta_atras : Bool;
  Dosificar : Bool;
END_VAR

VAR
  Peso_permitido : String;
  Peso_per_real : Real;
  Primer_peso : LReal;
END_VAR

VAR_TEMP
  Peso_Actual : LReal;
  Volumen_Actual : LReal;
  Sensor_radar : LReal;
END_VAR

VAR CONSTANT
  Peso_Especifico : Int := 550;
END_VAR

BEGIN
  //*****DOSIFICADO AUTOMATICO*****
  //Con el presente programa se utiliza para el dosificado automatico
  //
  IF #Volumen_Obtenido = TRUE AND #Fin_Dosificado=FALSE THEN //Primeramente pregunt
    #Sensor_Radar"(Resultado => #Sensor_radar); // Se obtiene el valor
  END_IF

```

```

FUNCTION_BLOCK FB1718
TITLE='Dosificado automatico'
NAME:'Dos_Aut'
VERSION: '1.0'
AUTHOR:'JAC'
FAMILY:'MEYME'
{ S7_Optimized_Access := 'TRUE' }

VAR_INPUT
  Volumen_Obtenido : Bool;
  Volumen_tolva : Real;
  Distancia_maxima { ExternalAccessible := 'False'; ExternalVisible := 'False'; ExternalWritable := 'False'; }
  Final_carrera_trasero { ExternalAccessible := 'False'; ExternalVisible := 'False'; ExternalWritable := 'False'; }
  Sensor_radar : REAL;
  Peso_Actual : REAL;
  Peso_permitido : Real;
END_VAR

VAR_OUTPUT
  Cinta_adelante : Bool;
  Cinta_atras : Bool;
  Dosificar : Bool;
END_VAR

VAR
  Fin_Dosificado : BOOL;
  Primer_peso : REAL;
  Peso_Especifico : Int := 550;
END_VAR

VAR_TEMP
  Volumen_Actual : Real;
END_VAR

BEGIN
  //*****DOSIFICADO AUTOMATICO*****
  //Con el presente programa se utiliza para el dosificado automatico

```

Figura 89: Cambios en programación de SCL

Como se puede ver en la figura 89, se observa una comparación entre el bloque de función migrado y el bloque de función ya modificado y compilado en PCS7. Se hicieron remarcaciones con líneas para hacer la comparación entre ambos códigos, se explicará desde la parte superior bajando a través de cada uno de los cambios.

El primer cambio mostrado en la parte superior, es el nombre del bloque de función en PCS7, en Tia Portal el número de FB se da automáticamente, el usuario simplemente pone un nombre al FB, por otro lado, en PCS7 se tiene que poner el número de FB explícitamente, como resultado se tendrá que poner un número de FB o FC a cada FB o FC migrado posteriormente se mostrará una tabla con el número de FB y FC migrados.

Además se tiene un cambio más de forma en el cual se pone el nombre, título, versión, autor y familia del programa, esta parte de código fue indicado en cada uno de los FB y FC.

A continuación, en Tia Portal existe la variable *LReal*, mientras que en PCS7 solamente se tiene la variable *Real*.

Además, otra diferencia entre programas es que en PCS7 no dispone de variables constantes por lo que se pasó a variables y se inicializó ahí un valor.

Por último a lo que refiere a la figura se tiene que para llamar a variables locales en Tia Portal se pone el símbolo “#” antes de la variable, mientras que en PCS7 no se pone nada.

Además hay que remarcar que se tiene una diferencia más a lo que refiere al código y esta se refiere a la llamada a función. En Tia Portal la llamada a funciones se hace a través del nombre de la función, sin embargo en PCS7 se tiene que hacer a través del número.

6.9. Asignación de número de FB y FC

Para la asignación de número de FB y FC, para cada uno de las funciones programadas, se tuvo que ver qué espacio de memoria se disponía, esto se encuentra delimitado por el espacio de usuario permitido por PCS7 (que afecta a funciones y DB) y por el espacio de usuario ya utilizado dentro del multiproyecto de San Adrián.

Para las funciones se observó que los espacios de memoria permitido por el PCS7, era de la posición 60 a la posición 700, mientras que en espacio de memoria utilizado en el proyecto se observaron varios FC hasta la posición 499, por lo tanto, todos los FC serán a partir del 500.

A lo que refiere a los bloques de función, solamente se escogió un espacio de memoria no utilizado con anterioridad, viendo entre bloques de funciones ya utilizados se dio cuenta que a partir del FB1700 no existían bloques funcionales.

Ordinal	Nombre	Tipo	Numero de FB o FC
1	Comunicación_Bascula	Función	FC500
2	Funcion_Adaptacion	Función	FC501
3	Sensor_S1	Función	FC502
4	Sensor_S2	Función	FC503
5	Sensor_Radar	Función	FC504
6	Media	Función	FC505
7	Buffer_Llenado	Función	FC506
8	Pendiente	Función	FC507
9	Lectura_Matricula	Función	FC509
10	Peso_Obtenido	Función	FC510
11	Adaptacion_Bascula	Función	FC511
12	Entrada_Camion	Bloque de función	FB1700
13	Salida_Camion	Bloque de función	FB1701
14	Adaptacion_Sensor	Bloque de función	FB1712
15	Altura	Bloque de función	FB1713
16	Ancho	Bloque de función	FB1714
17	Longitud	Bloque de función	FB1715
18	Volumen	Bloque de función	FB1716
19	Movimiento_motor	Bloque de función	FB1717
20	Dosificado_Automatico	Bloque de función	FB1718

Tabla 6: Tabla de número de función o bloque de función

Como se puede ver en la tabla 6, se tienen todos los programas ya migrados a PCS7, a cada uno de ellos se le dio un número de FC o FB respectivo, respetando los espacios de memorias ocupados. Hay que remarcar que la función Volumen en la versión de Tia Portal, era una función, mientras que en PCS7 es un bloque de función.

6.10. Programación de FB y FC no migrados

Entre las funciones que no se pudieron migrar están las siguientes:

- *FC12 “Principal”*
- *FC17 “Salidas”*
- *FC11 “Gestion_Volumen”*
- *FC8 “Motor_mecanismo”*
- *FC18 “Motor dosificado”*

- FC19 “Motor traslación”
- FC1 “Comunicacion_Escaner”
- FC13 “Comunicacion_Bascula”

Entre las funciones anteriormente mencionadas las funciones asociadas a la gestión de motores ya que los bloques de motores ya se encuentran dentro del proyecto. Por otro lado, la función de salidas al ser también dependiente de las funciones de motores tampoco fue migrado.

Como resultado se hará la migración de las siguientes funciones FC12 “Principal”, FC11 “Gestion_Volumen”, FC1 “Comunicacion_Escaner” y FC13 “Comunicacion_Bascula”.

6.10.1. Migración FC Comunicación escáner

Como se puede ver en la figura 90 se tiene la migración directa de “Comunicacion_Escaner”, en lenguaje FUP en Tia Portal a “Comunicacion_Escaner” en PCS7.

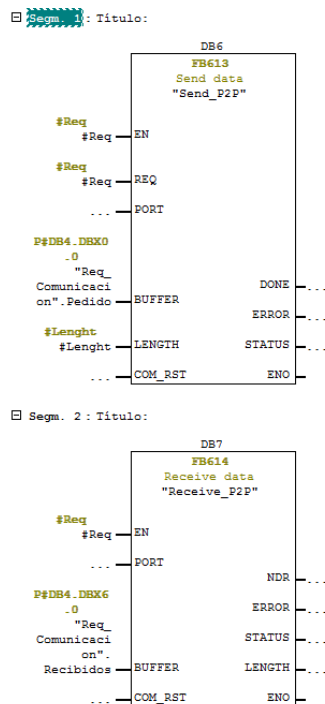


Figura 90: Programación FUP de Comunicación_Escaner

Como se puede ver en la figura 90, la programación en FUP es casi idéntica que en Tia Portal, sin embargo, los presentes FBs de “Send_P2P” y “Recieve_P2P” no tienen los mismos parámetros de entrada que anteriormente se tenía en la versión de Tia Portal, además que a priori no se tiene el puerto, ya que no se tiene la tarjeta de comunicaciones conectada al PLC de San Adrián.

6.10.2. Migración FB “*Gestion_Volumen*”

Como se puede ver en la figura 91, para la programación del FC11 “*Gestion_Volumen*” se decidió que la mejor forma era desarrollarlo en un CFC que integrara todos los bloques utilizados en el código de Tia Portal. Uno de los principales problemas que se tuvo a la hora de programar en PCS7, fue que, en la anterior versión, se utilizaban y se cambiaban variables internas, de DB de cada FB realizado, esto ocurría casi en toda la integración de estas funciones, por lo tanto, gran parte de estas variables o fueron exportadas a un DB global o fueron convertidas en entradas o salidas.

Ahora sabiendo lo anteriormente mencionado, se hicieron los cambios en diferentes FB. Como ejemplo en el primero, en el bloque de Altura, que se le aumento una entrada *Req* para que cuando se ponga la señal a TRUE se activara el mismo; abajo se puede ver una entrada *Reset*. Esta fue implementada en todos los FB de cálculo de alguna medida del camión para resetear todos los valores de cada FB, de modo que pueda poner a cero todos los valores del FB y se pueda hacer una nueva medición de volumen.

Además, se hace notar que existen algunos bloques que no se encontraban ahí en la versión de Tia Portal, entre otros *Sensor_S1*, *Sensor_S2* y *Fun_Adap*. Anteriormente estos bloques estaban dentro del OB que se ejecutaba cíclicamente, sin embargo, para la programación en CFC se le asigna un OB por defecto, el OB que la mayoría de los bloques utiliza es el OB35 que es un OB de activación cíclica de 100 ms. Respecto a la mayoría de los bloques esto no vendría a afectar, sin embargo, el bloque *Fun_Adap* que tiene la comunicación con el escáner, tiene que tener una activación más lenta, por lo tanto, se le asignó el OB34 que se activa cada 200 ms. Consiguientemente se tienen conectados los bloques de *Sensor_S1* y *Sensor_S2* a entradas analógicas físicas.

También se aprecia que existen entradas a bloques que vienen de un programa Principal (parte izquierda), además de que la única salida conocida también se encuentra conectada a Principal (parte derecha).

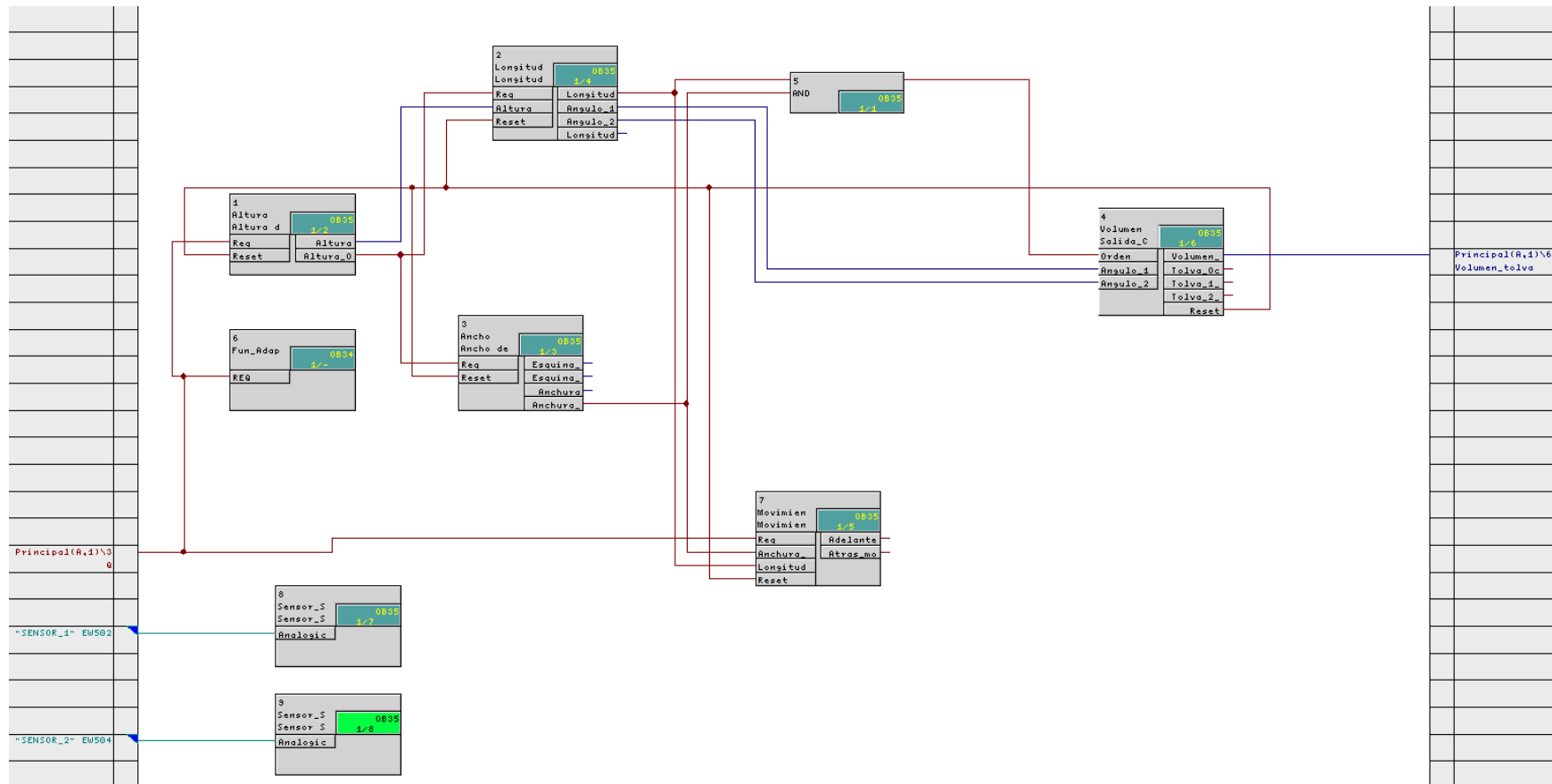


Figura 91: Programación en CFC para Gestión Volumen

6.10.3. Migración FC12 “Principal”

Se trató de replicar la función “Principal” de Tia Portal en PCS7, sin embargo existen algunas diferencias entre la función de PCS7 y Tia Portal entre ellas se pueden mencionar:

- La función en PCS7 es programado en CFC.
- Al no poder cambiar los valores de un DB instanciado por un FB, se decidió utilizar el DB global “WinCC” de forma que cada vez que un FC o FB acabe su funcionamiento ponga una variable a TRUE y a través de los bloque de SR(Set y Reset) activen el siguiente FB o FC, los mismos cuando acaben su acción pondrán su variable a TRUE para activar el siguiente FB o FC y la anterior variable en FALSE para desactivar su bloque SR. De esta forma se hasta terminar el ciclo principal y este vuelva a empezar.
- Las función “Adaptacion_señal” es implementado en el CFC y su salida es una entrada de nuestra función “Dosificado_Automatico”, de igual forma se hizo para la funciones “Sensor_Radar” y “Peso_Obtenido”.

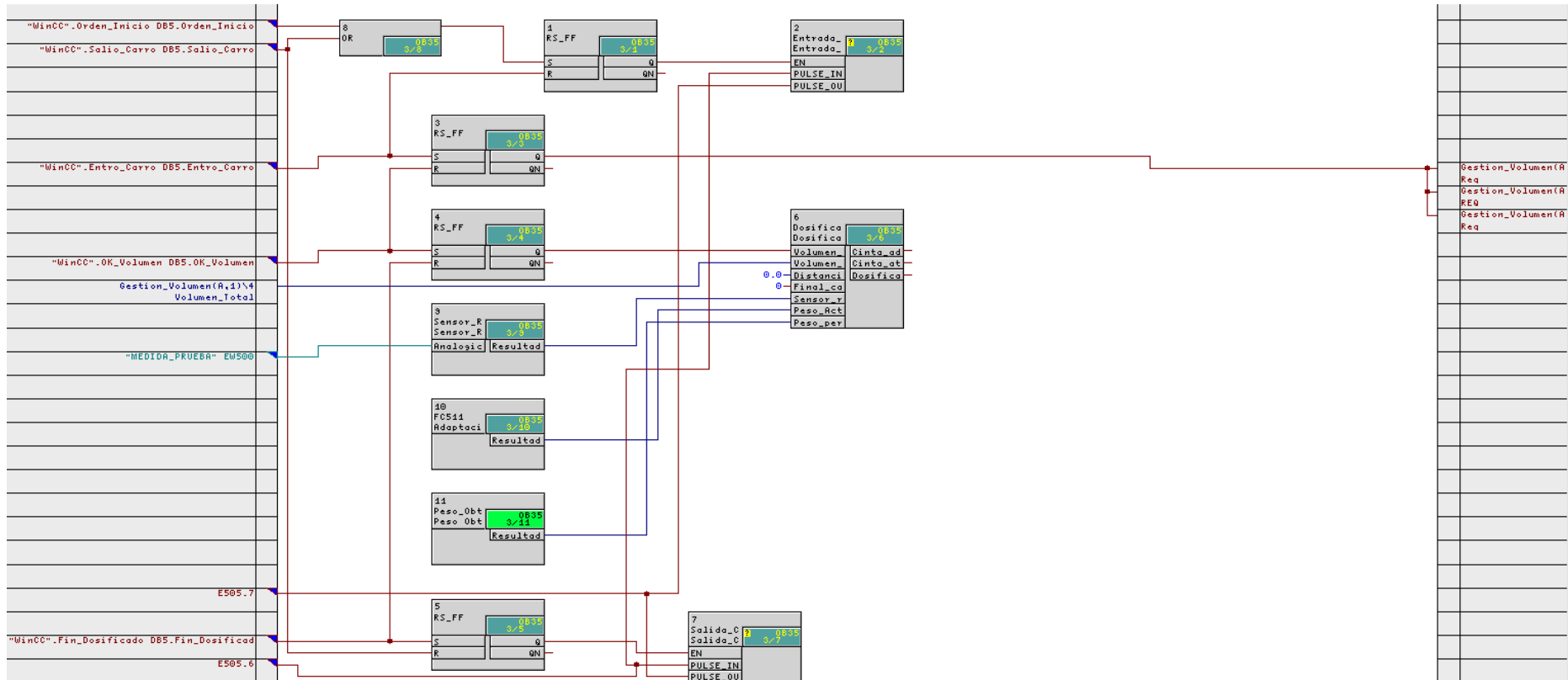


Figura 92: CFC de programa principal

7. ENSAYOS Y RESULTADOS

7.1. Introducción

Respecto a las pruebas sobre el sistema, se realizaron en un entorno controlado, en el cual se tenían las medidas a priori. Se preparó un contorno alrededor de los sensores según la disposición que se presenta en la imagen siguiente donde se aprecian:

- El escáner LED de haces múltiples R2100 de Pepperl Fuchs.
- Los dos sensores ópticos O1D100 de IFM Electronic.

Se delimitó el volumen a medir usando la pared y un conjunto de cajas que permitieron preparar las dimensiones apropiadas.

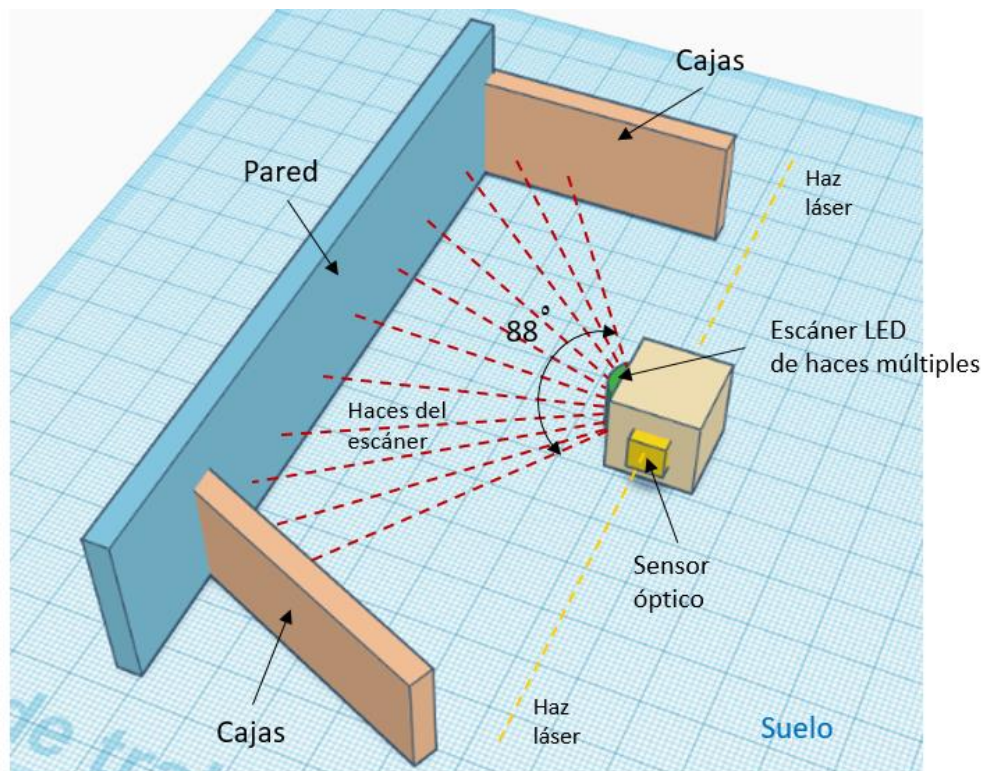


Figura 93: Organización del entorno de pruebas



Figura 94: Pruebas en ambiente controlado.

De esta manera el objetivo era medir el perfil en determinadas posiciones según la posición de los sensores en la caja, a la que realizaba un desplazamiento hacia la pared y perpendicular a la misma tomando las medidas correspondientes.

Teniendo esta instalación de prueba como referencia, se pudo calcular con cierta precisión el volumen de un ortoedro al colocar las cajas que aparecen inclinadas en la imagen a 90° con la pared. Sin embargo, debido al espacio disponible, cuando se colocaba un chaflán, el escáner no encontraba la esquina inferior de modo que se hicieron dos tipos de pruebas que eran igualmente concluyentes: la primera para el cálculo del ortoedro y la segunda simplemente para calcular el ángulo del chaflán obtenido por el sensor.

7.2. Visualización de resultados

Con el objetivo de mostrar la etapa en la que se encuentra el sistema y obtener datos complementarios de los sensores, además de los resultados obtenidos por el dimensionamiento de la tolva, se optó por hacer varias pantallas de visualización mediante una aplicación de WinCC Basic Versión V14 SP1, que se encuentra integrado en el Tia Portal.

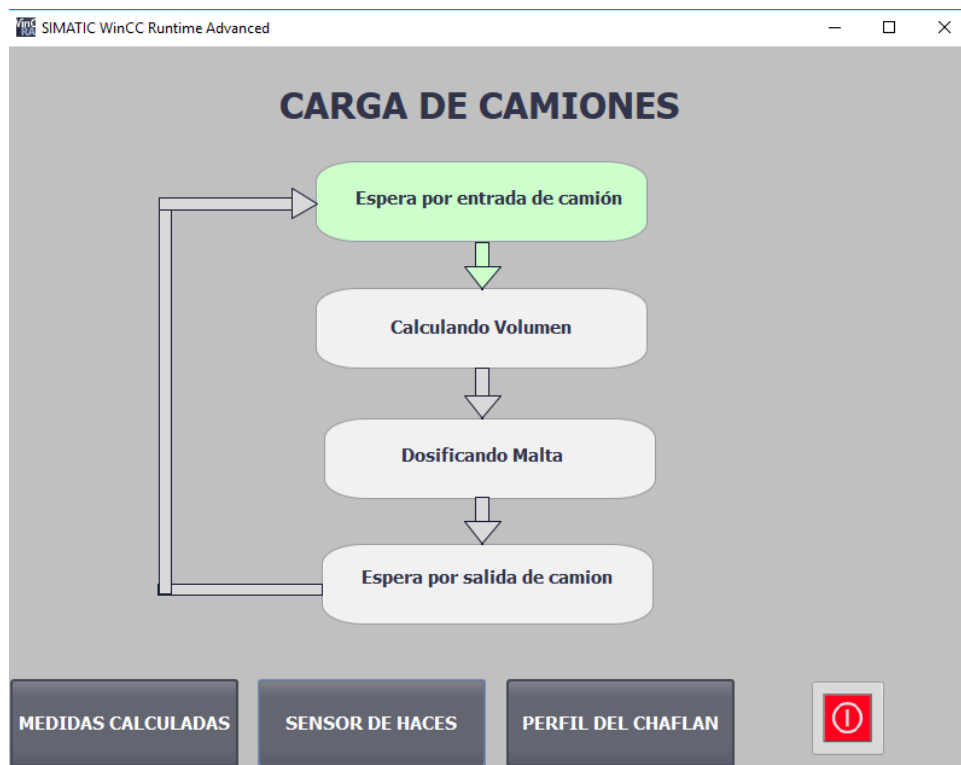


Figura 95: Pantalla de proceso principal

En la pantalla mostrada en la figura 95, se aprecian las etapas del sistema; si el sistema se encuentra en alguna de estas etapas, el recuadro respectivo aparece con fondo verde. De esta forma en la figura 95 se muestra que se encuentra en el estado de espera de camión.

También en cada una de las pantallas se colocaron botones para navegar entre pantallas (abajo) y un pulsador rojo para salir de la visualización.

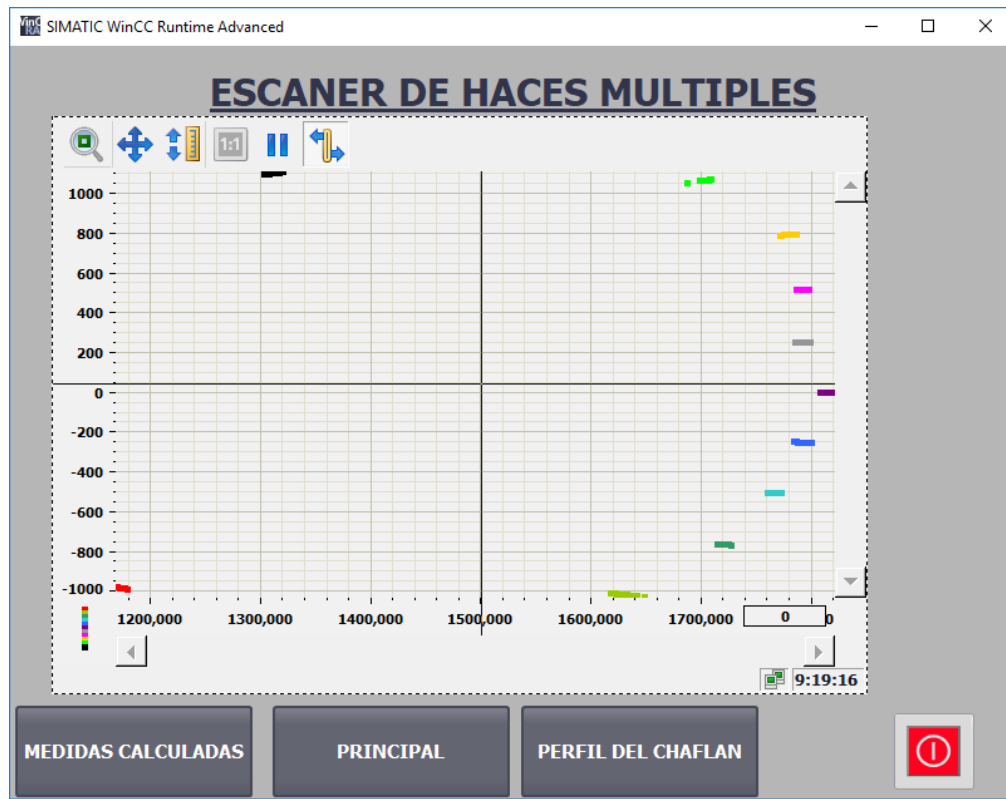


Figura 96: Pantalla de escáner de haces múltiples

En la pantalla mostrada en la figura 96, se tiene una pantalla en la que se muestra los haces del escáner en una representación cartesiana, esto se hace para visualizar donde los haces rebotaron. En la visualización se dibujan puntos de diferentes colores para cada uno de los haces de luz para no confundir un haz de otro.

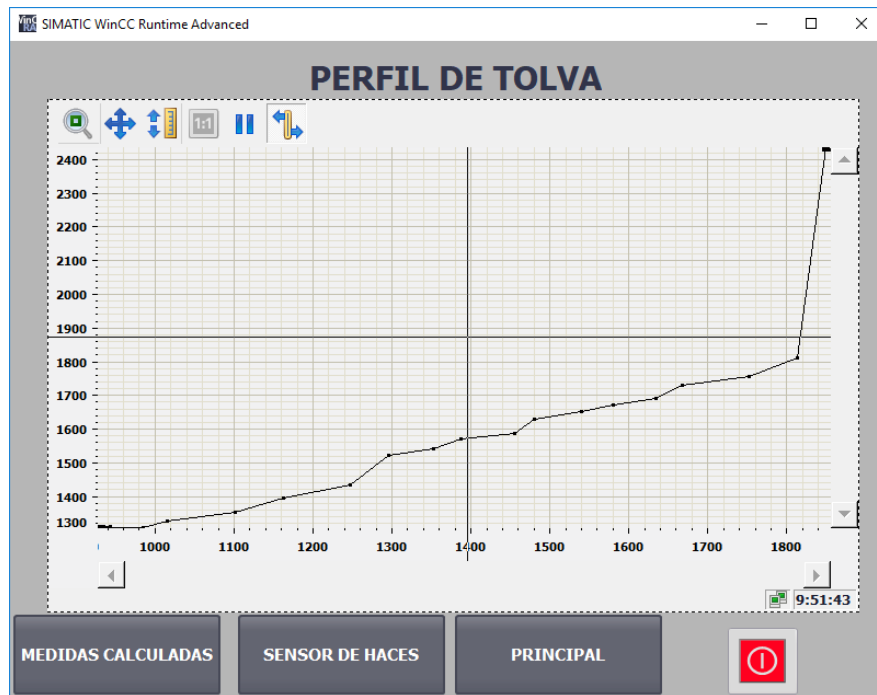


Figura 97. Pantalla de perfil de tolva

Como se puede ver en la figura 97, se observa una visualización del chaflán calculado por los sensores ópticos en forma de gráfico donde se pueden comprobar los resultados obtenidos.



Figura 98: Pantalla de resultados

Como se muestra en la figura 98, se tiene la pantalla con los resultados obtenidos en el dimensionamiento de la tolva, que se actualizarán automáticamente una vez obtenidos los presentes valores.

7.3. Resultados de las pruebas

7.3.1. Pruebas de volumen

Hay que hacer énfasis que las medidas de Ancho y Largo son las mismas, ya que el escáner busca esquinas, mientras que los sensores ópticos de distancia miden de largo que en la prueba realizada viene siendo la misma medida. También se puede ver en la figura 99 que se tienen valores complementarios de para el cálculo de Largo y Altura, los valores A2 y L2 son distancia entre los sensores ópticos (L2) y la distancia entre el escáner y los sensores fotoeléctricos (A2).

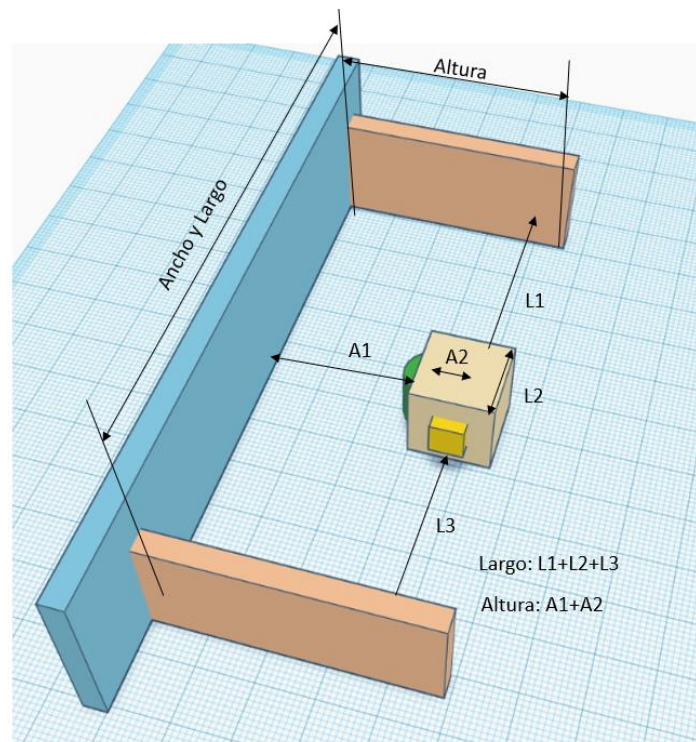


Figura 99: Aclaración de entorno de pruebas

Los datos de partida con las dimensiones reales del contorno establecido tipo ortoedro aparecen en la siguiente tabla, con medidas de longitud en milímetros y volumen en milímetros cúbicos.

MEDIDAS			
Altura [mm]	Ancho [mm]	Largo [mm]	Volumen [mm³]
1970	2100	2100	8687700000

Tabla 7: Medidas de entorno controlado

Una vez definidas las medidas se hicieron diez pruebas de funcionamiento del sistema, buscando la repetitividad y calculando el error en cada una de las medidas y el resultado fue el siguiente.

Nro. REPETICION	CLASIFICACIÓN	MEDIDAS [mm]	ERROR [%]
1	Altura	2115	7,360
	Ancho	2.039	-2,923
	Largo	2.055	-2,165
	Volumen	8.858.502.148	1,966
2	Altura	2039	3,503
	Ancho	2.101	0,028
	Largo	2.064	-1,734
	Volumen	8.838.506.756	1,736
3	Altura	2118	7,513
	Ancho	2.069	-1,487
	Largo	2.065	-1,682
	Volumen	8.968.599.134	3,233
4	Altura	2065	4,822
	Ancho	2.116	0,757
	Longitud	2.063	-1,751
	Volumen	9.014.871.974	3,766
5	Altura	2059	4,518
	Ancho	2.083	-0,798
	Largo	2.051	-2,320
	Volumen	8.798.797.695	1,279
6	Altura	2128	8,020
	Ancho	2.052	-2,291
	Largo	2.053	-2,216
	Volumen	8.966.278.729	3,207

Nro. REPETICION	CLASIFICACIÓN	MEDIDAS [mm]	ERROR [%]
7	Altura	2036	3,350
	Ancho	2.049	-2,428
	Largo	2.065	-1,682
	Volumen	8.613.329.772	-0,856
8	Altura	2042	3,655
	Ancho	2.068	-1,544
	Largo	2.060	-1,924
	Volumen	8.695.642.082	0,091
9	Altura	2104	6,802
	Ancho	2.106	0,287
	Largo	2.057	-2,027
	Volumen	9.143.928.785	5,251
10	Altura	2118	7,513
	Ancho	2.065	-1,666
	Largo	2.059	-1,975
	Volumen	9.003.375.600	3,634

Tabla 8: Resultados de pruebas de repetibilidad

Una vez obtenidos los resultados se preparó una tabla resumen, en la cual se hizo un promedio de todos los resultados obtenidos.

RESULTADOS	
Clasificación	Error [%]
Altura	5,505
Ancho	-1,207
Largo	-1,623
Volumen	2,331

Tabla 9: Tabla Resumen

Como se puede ver en la tabla 9 se tiene una media del error en volumen es del 2,33 %, esto se debe mayormente a que el error de la altura es del 5,706 %.

7.3.2. Prueba de medida del ángulo

Para las pruebas de precisión de la medida del ángulo se configuró una pared con las cajas de 22,5°, obteniéndose los siguientes resultados en 10 ensayos realizados en las mismas condiciones.

Nro Repetición	Angulo obtenido[°]	Error [%]
1	19,51	-13,30
2	26,00	15,56
3	20,70	-8,00
4	21,02	-6,58
5	28,32	25,87
6	20,16	-10,40
7	30,46	35,38
8	23,34	3,74
9	17,17	-23,70
10	26,32	16,97
Media	23,30	16,25

Tabla 10. Tabla de pruebas de Ángulos

Como se puede ver en la tabla 10 el cálculo del ángulo es mucho menos preciso que el del volumen, tiene un error de 16 %.

8. PLANIFICACION

En el siguiente cuadro se presenta el tiempo orientativo a cada tarea realizada.

TAREAS	2018/2019																					
	Septiembre				Octubre				Noviembre				Diciembre				Enero				Febrero	
	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20	S21	S22
INSTALACIÓN PREVIA																						
Estudio de instalación																						
Estudio de planos electricos																						
Estudio de programa PCS7																						
SELECCIÓN DE INSTRUMENTACIÓN																						
Instrumentación para entrada de camion																						
Instrumentación para dosificado automatico																						
Instrumentación para dimensionamiento de tolva																						
DESARROLLO DE INTEGRACIÓN HARDWARE Y SOFTWARE																						
Pruebas de funcionamiento de escaner																						
Programación en Tia Portal																						
Migración en PCS7																						
Planos electricos																						
DOCUMENTACIÓN																						

Figura 100: Tiempo por tareas

9. PRESUPUESTO

El costo del proyecto (ver el tabla 11) alcanza a la suma de 25.154,01 Euros. De este montante, el importe mayor es el Sensor radar VEGAPULS 67 que llega a 1.717,28 Euros, otro componente electrónico importante desde el punto de vista del coste es el Lector de Matrícula, 839,18 Euros. Asimismo, se debe que indicar el coste del ingeniero que realiza el diseño y programación del sistema que alcanza los 16.000,00 euros.

Las herramientas software y equipos de programación se consideran ya disponibles y amortizados no incluyendo, por tanto, estos gastos en el presupuesto.

Equipo	Marca	Referencia	Unidades	Precio unitario	Precio total
Componentes electrónicos					
Escaner LED de haces multiples	Pepperl&Fuchs	OMD8000-R2100-R2-2V15	1	640,38	640,38
Conector hembra de alimentacion para escaner	Pepperl&Fuchs	V1-G-5M-PUR	1	6,30	6,30
Conector hembra de comunicación para escaner	Pepperl&Fuchs	V15-G-5M-PUR-ABG	1	19,96	19,96
Sensor de distancia optico	IFM	O1DLF3KG/IO-LINK	2	216,51	433,02
Cable de conexión de sensor optico	IFM	ADOAH040MSS0002H04	2	9,24	18,48
Sensor radar VEGAPULS 67	VEGAPULS	PS67.RXBYDHAMXX	1	1717,28	1717,28
Lector de matricula			1	839,18	839,18
Sensor fotoelectrico	Schneider Electric	XUK9ARCNL2	2	75,10	150,20
Remota ET 200SP	Siemens	6ES7155-6AU00-0CN0	1	357,00	357,00
Modulo de entradas analogicas	Siemens	6ES7134-6GD01-0BA1	1	214,20	214,20
Modulo de comunicación	Siemens	6ES7137-6AA00-0BA0	2	385,00	770,00
Modulo de salidas digitales	Siemens	6ES7132-6BF00-0CA0	1	59,25	59,25
Modulo entradas digitales	Siemens	6ES7131-6BF00-0CA0	1	71,68	71,68
Base de conexión	Siemens	6ES7193-6BP20-0DA0	4	39,27	157,08
Componentes mecanicos					
Mecanismo			1	3700,00	3700,00
Mano de obra			Horas		
Horas ingeniero			320	50,00	16000,00
				TOTAL	25154,01

Tabla 11: Presupuesto estimado

10. CONCLUSIONES

Durante todo el proceso, del proyecto se pasaron por múltiples pasos.

Primero se hizo la elección de instrumentación, eligiendo los sensores de distancia O1D100, el escáner de LED de haces múltiples serie R2100, el sensor radar VEGAPULS 67 y los sensores fotoeléctricos XUK9ARCNL2, todos escogidos o por elección propia o por recomendación de la empresa, luego se hizo la elección de hardware para cada uno de los presentes componentes.

Segundo se hizo la programación en Tia Portal entre las funciones principales se puede mencionar a todas las funciones para obtener alguna medida de la tolva entre ellos Altura, Ancho y Longitud, dando sus resultados a la función Volumen, obteniendo el volumen real, asimismo se hizo la programación para la entrada y salida de camiones, posteriormente se hizo la programación para el dosificado automático que terminara cuando el volumen se haya alcanzado o se llegue al peso máximo permitido por ley.

Luego se hizo la migración al programa PCS7, que al hacer algunos cambios de sintaxis y cambiando algunos bloques que no eran exportables se hizo la migración completa de TIA PORTAL a PCS7.

Consiguientemente se hizo las pruebas en un ambiente controlado, obteniendo un error en volumen de un ortoedro del 2.11% y en ángulo del 16%. Concluyendo que el error en volumen es permisible, teniendo en cuenta que el error en ángulo se tiene que mejorar. Por último se hizo un presupuesto de todo el proyecto calculando un precio de 25.094 euros netos de todo el proyecto.

11. FUTURAS AMPLIACIONES

Se plantean tres futuras ampliaciones. Primero que se aplicara la guía *GEMMA (Guide d'Etudes des Modes de Marches et d'Arrets)*, para tener más modos de funcionamiento, entre los mismos sería muy conveniente hacer unas marchas de test para poder probar cada uno de las medidas obtenidas de la tolva.

Segundo, un interfaz humano máquina, puede llegar a ser muy conveniente para que se tenga las medidas de longitud, ancho y altura una vez que se haya calculado, además de indicar el estado del dosificado en la tolva.

Tercero se haría un análisis de riesgos, para los conductores de los camiones y se aplicaría medidas de seguridad si fueran necesarias.

12. BIBLIOGRAFÍA

- [1] Meyme S.A. “Página de inicio de la empresa Meyme S. A.”
<https://www.meyme.com/> (Ultimo acceso 01-Febrero-2019)
- [2] Gebr.Laussman “Página de inicio Laussman.org”
<http://www.lausmann.org/EN/index.html> (Ultimo acceso 05-Febrero-2019)
- [3] Malteurop “Malteurop en España”
<https://es.malteurop.com/nuestro-grupo/nuestras-malterias/espana>(Ultimo acceso 01-Febrero-2019)
- [4] Julio Cesar Barredo García “Deducción geométrica de los productos notables en el espacio tridimensional como recurso didáctico en el proceso de enseñanza aprendizaje de la matemática”
Editorial: Iberoamericana
- [5] Ingemecanica “Distribución de Esfuerzos sobre el Bastidor del Vehículo”
<https://ingemecanica.com/tutorialsemanal/tutorialn14.html>) (Ultimo acceso 01-Febrero-2019)
- [6] Viarural “Camiones Volkswagen 17220 M Dimensiones”
<https://www.viarural.com.ar/viarural.com.ar/insumosagropecuarios/ganaderos/camiones/volkswagen/17220m/dimensiones.htm> (Ultimo acceso 01-Febrero-2019)
- [7] Schneider Electric “Hoja de características del producto XUK9ARCNL2”
- [8] Pepperl Fuchs “Escáner Laser serie R2000”
https://www.pepperl-fuchs.com/spain/es/R2000_Detection_laser_scanner.htm (Ultimo acceso 01-Febrero-2019)
- [9] Pepperl Fuchs “Sensor LiDAR 2D”
https://www.pepperl-fuchs.com/global/es/classid_53.htm?view=productdetails&prodid=62235 (Ultimo acceso 01-Febrero-2019)
- [10] Pepperl Fuchs “Encóder con sisga ECA10TL-PROFINET”
https://www.pepperl-fuchs.com/global/es/classid_364.htm?view=productdetails&prodid=72518 (Ultimo acceso 01-Febrero-2019)
- [11] IFM “Sensor de distancia óptico O1D100”
<https://www.ifm.com/es/es/product/O1D100> (Ultimo acceso 01-Febrero-2019)
- [12] VEGA “VEGAPULS 67”

https://www.vega.com/en/home_de/Products/Product-catalog/Level/Radar/VEGAPULS-67 (Ultimo acceso 01-Febrero-2019)

[13] Ramon Piedrafita Moreno “Ingeniería de la Automatización Industrial”, Editorial Ra-Ma, 2004

[14] Indiamart “CPU S7-1500 CPU 1511-1 Pn”
<https://www.indiamart.com/proddetail/simatic-s7-1500-cpu-1511-1-pn-20220044048.html>
(Ultimo acceso 01-Febrero-2019)

[15] TP Automation “Modulo de periferia descentralizada”
<http://www.tpautomation.de/Automation-systems/SIMATIC-ET-200/ET-200SP/Interface-modules/6ES7155-6AA01-0BN0-ET-200SP-PROFINET-Bundle-IM-155-6PN-Standard::32912.html?language=en> (Ultimo acceso 01-Febrero-2019)

[16] DISTRELEC “Modulo de comunicación serial , Siemens”
<https://www.distrelec.de/en/communication-module-serial-siemens-6es7137-6aa00-0ba0/p/11091331> (Ultimo acceso 01-Febrero-2019)

[17] INT TECHNICS “Modulo de entradas analógicas”
<https://plc-trade.com/es/mpn/6es7134-6gb00-0ba1/> (Ultimo acceso 01-Febrero-2019)

[18] Siemens “CM PtP- Configuration for point-to-point connections”, Siemens ,2018

13. ANEXO: PLANOS ELÉCTRICOS DE LA INSTRUMENTACIÓN

Se presentan a continuación los planos, desarrollados en EPLAN por el autor, y correspondientes a la instalación eléctrica de la instrumentación donde interviene la alimentación en continua 24V DC y sus protecciones, así como la conexión de señales (digitales y analógicas), la comunicación de los distintos elementos al PLC y el accionamiento del motor del mecanismo elevador. Y se organizan según:

- Distribución 24V DC y protecciones
- Configuración ET200SP
- Slot 1: 8 Entradas digitales
- Slot 2: 4 Entradas analógicas
- Slot3: 8 Salidas digitales
- Slot 4: Módulo de comunicación
- Accionamiento del mecanismo

