

Trabajo de Final de Grado

Ingeniería de Sistemas TIC

HOMOLOGACIÓN DEL SOFTWARE DE UN SISTEMA DE PESAJE

Autor: Nil Antonell

Tutor: Teresa Escobet

Convocatoria: Julio

Escuela Politécnica Superior de
Ingeniería de Manresa



**Enginyeria de
Sistemes TIC**

ÍNDIX

1. RESUMEN DEL PROYECTO	7
2. ALCANCE DEL PROYECTO	8
3. DESCRIPCIÓN DEL EQUIPO	10
3.1. Funcionamiento general	11
3.1.1. Permiso de marcha.....	11
3.1.2. Iniciar y parar.....	11
3.1.3. Modo transporte	11
3.2. Pantalla táctil de control	12
3.2.1. Pantalla principal.....	12
3.2.2. Usuarios y seguridad de las pantallas	13
4. USO DE LA GUÍA WELMEC.....	15
4.1. Estructura general	15
4.2. Selección de los apartados adecuados.....	16
4.2.1. Configuración básica (P o U)	16
4.2.2. Configuraciones TI aplicables	17
4.2.3. Requisitos específicos del instrumento (extensión I).....	17
4.2.4. Clase de riesgo aplicable	17
5. REQUISITOS ESPECÍFICOS PARA EL TIPO P	19
5.1. P1: Documentación	19
5.1.1. Descripción del software legalmente relevante	19
5.1.2. Descripción de la exactitud de los algoritmos de medida	28
5.1.3. Descripción del hardware del sistema	29
5.2. P2: Identificación del software.....	36
a) Omron	36
b) Panasonic	36
5.3. P3: Influencia sobre el software a través de la interfaz de usuario	37
5.4. P4: Influencia sobre el software a través de interfaces de comunicación.....	40
5.5. P5: Protección frente a cambios accidentales o no intencionados	41
a) Omron	41
b) Panasonic	41
5.6. P6: Protección ante las modificaciones intencionadas	41
5.7. P7: Protección de parámetros.....	42

6. EXTENSIÓN T: TRANSMISIÓN DE DATOS DE MEDIDA MEDIANTE REDES DE COMUNICACIÓN	43
6.1. T1: Completitud de los datos transmitidos	43
6.2. T2: Protección frente a los cambios accidentales o no intencionados	43
6.3. T3: Integridad de los datos.....	44
6.4. T4: Autenticidad de los datos transmitidos	44
6.5. T5: Confidencialidad de las claves.....	44
6.6. T6: Tratamiento de datos corruptos	44
a) Omron	45
b) Panasonic	45
6.7. T7: Retraso en la transmisión.....	45
6.8. T8: Disponibilidad de los servicios de transmisión.....	46
7. EXTENSIÓN S: SEPARACIÓN DEL SOFTWARE	47
7.1. S1: Realización de la separación del software	47
7.2. S2: Indicación mixta	47
7.3. S3: Interfaz de software protectora	48
8. EXTENSIÓN D: DESCARGA DE SOFTWARE LEGALMENTE RELEVANTE	49
8.1. D1: Mecanismo de descarga	49
8.2. D2: Autenticación del software descargado	49
8.3. D3: Integridad del software descargado	50
8.4. D4: Trazabilidad de la descarga del software legalmente relevante	50
9. VALIDACIÓN	51
10. CONCLUSIONES	54
11. AGRADECIMIENTOS.....	55
12. REFERENCIAS BIBLIOGRÁFICAS	56
ANEXO A: OIML D31.....	58
A.1 Soporte a la protección de la durabilidad.....	58
ANEXO B: SWIFT DE UTILCELL	59
B.1. Aplicación Check-weigher	59
B.1.1 Funcionamiento por tecla	60
B.1.2 Funcionamiento por entrada externa.....	60
B.1.3 Funcionamiento por comando serie	60
B.1.3 Funcionamiento por nivel de peso.....	61
B.2. Comunicaciones	61



B.2.1 Protocolo RS-232.....	61
B.2.2 Protocolo propio (RS-485).....	63
ANEXO C: SOFTWARE.....	65

ÍNDIX DE FIGURAS

Figura 1. Sistema de pesaje.....	10
Figura 2: Pantalla principal.....	12
Figura 3: Menú de la pantalla principal.....	13
Figura 4. Diagrama de estados del bloque “Sistema_Pesaje”	22
Figura 5. Diagrama de estados del bloque “Configuracion_FB”	26
Figura 6. Función de trigger	29
Figura 7. Diagrama de bloques Omron	30
Figura 8. Diagrama de bloques Panasonic	30
Figura 9: Transductor AED.....	32
Figura 10: Pantalla digital DWS2103	33
Figura 11. Conversor RS-232 TTL.....	34
Figura 12: PLC Omron.....	35
Figura 13: PLC Panasonic.....	35
Figura 14. Swift fabricado por Utilcell.....	59
Figura 15. Proceso de pesaje.....	59

ÍNDIX DE TABLAS

Taula 1. Correspondencia entre el diagrama de estados y el bloque de función “SistemaPesaje”	24
Taula 2. Correspondencia entre el diagrama de estados y el bloque de función “Configuracion_FB”	28

1. RESUMEN DEL PROYECTO

El trabajo de final de grado se realiza en el marco de un convenio de cooperación educativo en la empresa Tesi Industrial Europa SL., empresa especializada en el diseño y construcción de máquinas específicas y estándar para líneas de producción y envasado de productos agroalimentarios. Concretamente el trabajo realizado se ha focalizado en desarrollar un software que permite la comunicación entre el terminal de pesaje y el sistema de control de la línea de producción. Por ello, se definen las especificaciones que debe cumplir un programa dedicado a obtener el correcto funcionamiento de un sistema de pesaje dinámico, y se evalúan las restricciones que deben incorporarse para asegurar que el sistema funcione según lo esperado, con el fin de conseguir el máximo número de pesadas por minuto y con la máxima fiabilidad.

The final degree project is carried out under an educational cooperation agreement in the company Tesi Industrial Europa SL., company specialized in the design and construction of specific machines and standard production lines and packaging of agricultural products. Specifically, the project has focused on developing software that enables communication between the weighing terminal and the control system of the production line. Thus, it defines specifications to be met by a dedicated program to obtain the correct functioning of a dynamic weighing system, and will evaluate the restrictions to be incorporated to ensure that the system works as expected, in order to get the maximum number of weighed per minute and maximum reliability.

2. ALCANCE DEL PROYECTO

Este proyecto realizado consiste en el desarrollo de un bloque de funciones destinado a lograr la integración de un sistema de pesaje con un controlador lógico programable (PLC), concretamente con los PLCs de marca Omron y Panasonic. Un PLC es una computadora utilizada para automatizar procesos electromecánicos. Para realizar el pesaje dinámico, cada controlador se comunicará con un terminal SWIFT fabricado por Utilcell y con un terminal AED fabricado por HBM. Indicar que el terminal SWIFT de Utilcell es un indicador de pesaje y transmisor de alta velocidad utilizado tanto en aplicaciones de pesaje estático como dinámico, mientras que el terminal AED de HBM, es un dispositivo que se encarga de digitalizar las señales analógicas de las células de carga, utilizado también tanto en aplicaciones de pesaje estático como dinámico.

En el trabajo se presenta la documentación necesaria para homologar el software de comunicación desarrollado. En concreto se muestra la documentación presentada para la homologación de software que permite la comunicación del terminal AED con los PLCs Panasonic y Omron. También, se presenta el software desarrollado para comunicar los terminales tipo SWIFT de Utilcell con PLCs Panasonic y Omron, con la respectiva documentación del SWIFT de Utilcell en el ANEXO B: SWIFT DE UTILCELL En este segundo caso no se ha requerido la homologación del software, pero se le ha dado una utilidad adicional, se ha desarrollado un bloque de funciones preparado para comunicar en modo bus un máximo de 16 SWIFT's de Utilcell con el PLC, que pueden pertenecer cada uno a un equipo de pesaje, y retornarían los datos del pesaje al PLC.

Tal y como se ha mencionado, el objetivo del trabajo, a parte de su homologación, es conseguir el correcto funcionamiento de estos 4 tipos de sistema de pesaje. Uno de los requerimientos en el diseño del software ha sido su portabilidad, es decir desarrollar un software que sea fácilmente exportable a otras arquitecturas, es por ello que el bloque de funciones desarrollado se ha realizado utilizando el texto estructurado (ST) como lenguaje de programación. Gracias a este lenguaje, en el software solo se deberán cambiar algunos detalles referentes al tipo de controlador o al tipo de terminal. El software desarrollado se muestra en el documento llamado "Anexo_C".

Para homologar el software desarrollado se han seguido las directrices indicadas en la "Guía del software Welmec 7.2 (edición 5)" [\[1\]](#). WELMEC es el resultado de una cooperación entre

las autoridades de metrología legal de los estados miembros de la Unión Europea y la Asociación Europea de Libre Comercio.

Más concretamente, la guía WELMEC pertenece a la directriz OIML (Organización Internacional de Metrología Legal), una organización intergubernamental cuyo principal objetivo es armonizar las regulaciones y controles metrológicos aplicados por los servicios de metrología nacionales u organizaciones afines, de sus estados miembros. Va dirigida a fabricantes de instrumentos de medida y a organismos notificadores responsables de la evaluación de la conformidad de sus productos. En el ANEXO A: OIML D31 se podrá observar una restricción extraída de la guía OIML D31 [\[4\]](#) que debe ser cumplida y que en la guía WELMEC no está considerada.

Para cumplir con una de las especificaciones detalladas en la guía WELMEC se ha utilizado el Arduino Uno, un dispositivo que permite ser adaptado a varios tipos de proyectos para cumplir con una serie de necesidades, tal y como se ha observado a lo largo de la carrera de Ingeniería de sistemas TIC. Se ha utilizado para poder comprobar que la identificación del software sea la correcta. Se comenta con más detalle en el apartado 5.2. P2: Identificación del software

3. DESCRIPCIÓN DEL EQUIPO

En este apartado se va a hacer una pequeña descripción del sistema de pesaje comandado por un PLC Omron. El otro sistema comandado por un PLC de Panasonic es similar al que se va a comentar, la principal diferencia es que en lugar de rechazar el producto sino se encuentra dentro de los intervalos de peso permitidos, este sistema tiene la función de etiquetar el producto con su peso y una serie de datos pertenecientes al producto.

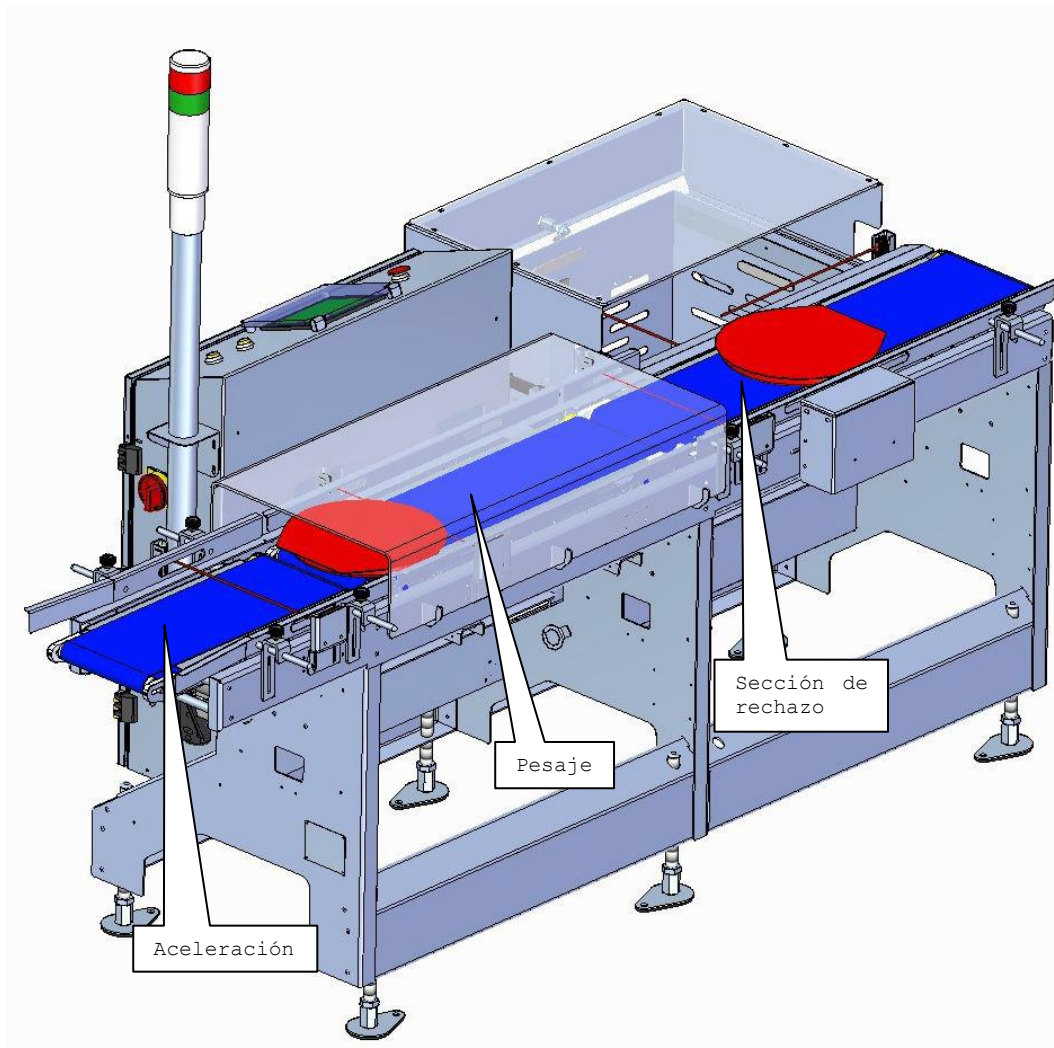


Figura 1. Sistema de pesaje

La Figura 1. Sistema de pesaje, muestra la instalación en el cual se ubicará el equipo de pesaje. Se trata de una controladora de peso con rechazo por pizza formado por tres cintas transportadoras y un expulsor.

El producto circula por la llamada sección de alimentación y separación (denominada aceleración), el cual consta de una cinta transportadora ajustable a distintas formas de

objetos. De la sección de alimentación se pasa a la sección de pesaje, en esta sección se dispone de una cinta transportadora en donde se detecta el peso y la forma del producto. Para terminar, el producto llega a la sección de rechazo, en donde el producto continúa por la cinta si su peso se encuentra dentro de los intervalos permitidos, o de lo contrario, se expulsa el producto de la cinta.

En el panel de operación de la maquina consta de distintos botones inicio/parada y de una pantalla táctil. A continuación se describe la funcionalidad de cada uno de ellos.

3.1. Funcionamiento general

3.1.1. Permiso de marcha

Hay una señal de entrada al equipo que es el permiso de marcha (si no está activa no se pondrán en marcha las cintas transportadoras). La otra señal también es de entrada y permite poder reactivar la señal de permiso de marcha en caso que se haya desactivado. No se puede observar muy bien en la Figura 1, pero estos dos pulsadores se encuentran al lado de la pantalla táctil.

La señal de permiso de marcha se desactiva si sucede uno de los siguientes casos:

- La máquina está parada.
- Alarma a los variadores de las cintas.
- Se transfiere el software al controlador programable.

3.1.2. Iniciar y parar

Para poner en marcha y detener el equipo hay un único pulsador. Con una pulsación se pone en marcha y con otra se detiene. El pulsador tiene un piloto luminoso de color naranja. Cuando el piloto hace intermitencia indica que el equipo está parado. Cuando el piloto está encendido indica que está en funcionamiento.

3.1.3. Modo transporte

El modo transporte hace que el equipo se comporte como una cinta transportadora. No pesa ni expulsa el producto. Para activarlo y desactivarlo se dispone de un pulsador en la pantalla

táctil denominado “Modo transporte”. El color negro indica que está desactivado y el color verde que está activado. Al activarlo se ponen en marcha las cintas independientemente de la señal de permiso de salida.

3.2. Pantalla táctil de control

La elaboración del diseño de la pantalla táctil no forma parte de este proyecto, no obstante describimos brevemente los menús disponibles debido a que la información extraída del programa desarrollado se mostrará en esta pantalla. No se ha mostrado explícitamente que hay dentro de cada menú.

3.2.1. Pantalla principal

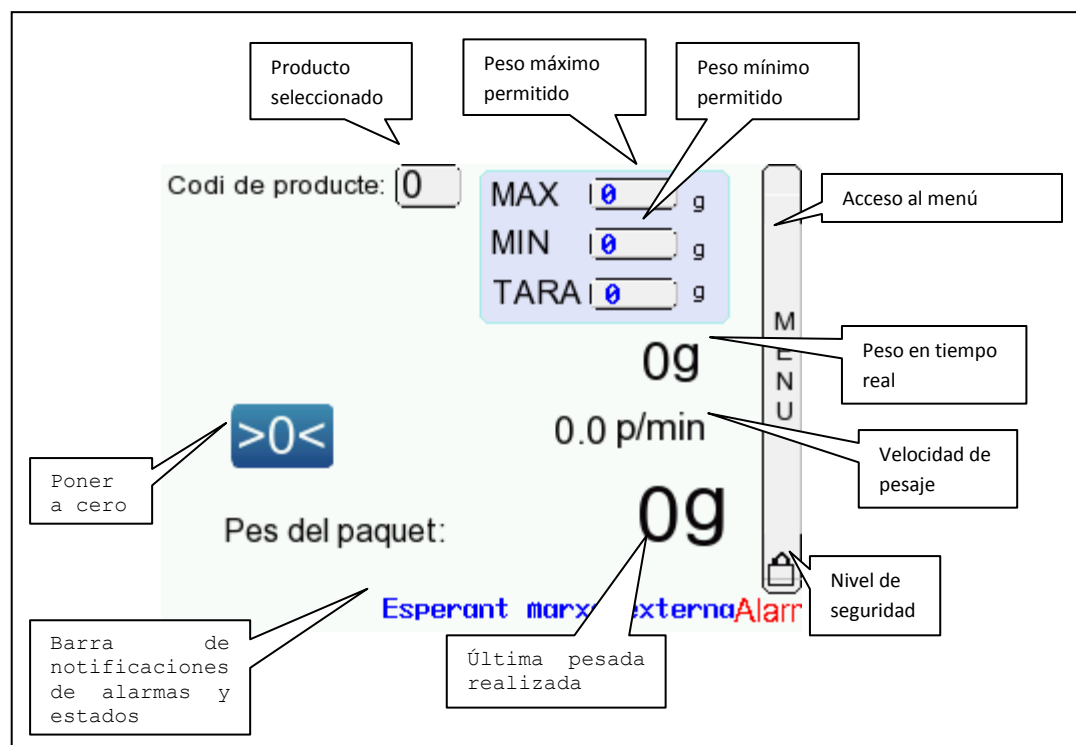



Figura 2: Pantalla principal

- Con la barra de menú se accede a la configuración.
- Para seleccionar un programa de trabajo diferente se debe introducir el código de producto al campo correspondiente situado a la zona superior de la pantalla principal.
- En la parte inferior de la pantalla aparece información sobre las alarmas en el caso de haber alguna de activada.

-  Señala el nivel de acceso actual. Cuando el candado está cerrado indica que no se ha iniciado sesión con ningún usuario. Un candado abierto con un número 1 indica que hay la sesión abierta con nivel 1 de seguridad que permite acceder al menú de control manual y en la configuración de producto. Un candado abierto con un número 2 indica que hay la sesión abierta con el nivel 2 de seguridad; esto significa que tiene acceso a todos los menús.
- A la parte inferior de la pantalla se muestra la última pesada obtenida. Un poco más arriba se encuentra el peso en tiempo real.
- Pulsador de poner a cero (>0<): El botón de cero se queda verde si el visualizador está a 0. En azul si no está a 0. Pulsando en el botón se pondrá a cero la báscula.
- Es posible modificar el peso máximo y mínimo del producto seleccionado desde esta pantalla, pero hay que iniciar sesión de nivel 1.

3.2.2. Usuarios y seguridad de las pantallas

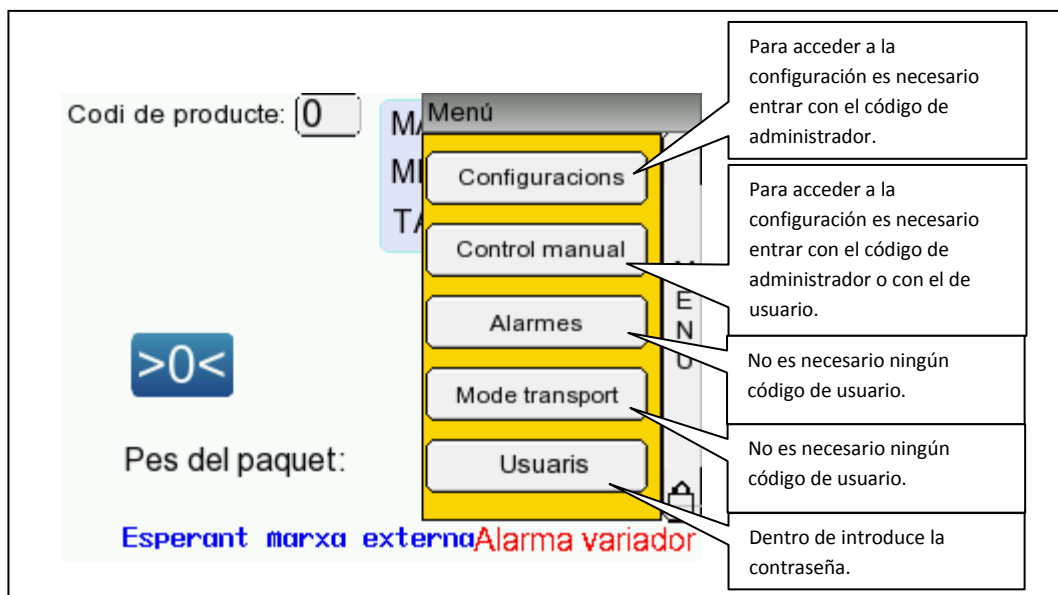


Figura 3: Menú de la pantalla principal

- El acceso a los menús está restringido. Hay 2 niveles de acceso disponibles con contraseña.
 - Usuario.
 - Administrador.



- Todas las sesiones abiertas tienen una duración de 10 min, después de este tiempo se cerrará automáticamente la sesión.

4. USO DE LA GUÍA WELMEC

Como se ha comentado en la introducción, para la homologación del software desarrollado se seguirán las directrices señaladas en la guía WELMEC. En este capítulo se describe como está organizada dicha guía y los requisitos más importantes que deberemos tener en cuenta en el desarrollo del programa.

4.1. Estructura general

La guía se organiza como una serie estructurada de bloques de requisitos:

1. Requisitos de configuraciones básicas de los instrumentos de medida.
2. Requisitos de configuraciones de las Tecnologías de la Información (TI).
3. Requisitos específicos del instrumento.

El primer tipo de requisitos se aplica a todos los tipos de instrumentos, relacionados con ellos se distingue entre instrumentos de medida desarrollados específicamente (P) e instrumentos de medida que usan un ordenador universal (U).

El segundo tipo de requisitos corresponde a las siguientes funciones TI:

- Almacenamiento a largo plazo de los datos de medida (L)
- Transmisión de los datos de medida (T)
- Descarga de software (D)
- Separación de software (S).

Estos requisitos solo se aplican si existe la función correspondiente en el instrumento utilizado.

El último tipo de requisitos define una serie de requisitos específicos del instrumento.

Además, los requisitos de esta guía se diferencian según las clases de riesgo. Se presentan 6 clases de riesgo enumeradas de la A a la F en orden creciente de nivel riesgo. La clase de menor riesgo (A) y la clase de mayor riesgo (F) actualmente no se utilizan, se reservan para el caso en que sean necesarias en un futuro. Las clases de riesgo de la B a la E abarcan todas las clases de instrumentos regulados por la MID. Cada instrumento de medida debe asignarse a una clase de riesgo debido a que los requisitos particulares del software están determinados por la clase de riesgo a la que pertenece el instrumento en cuestión.

4.2. Selección de los apartados adecuados

La guía WELMEC se puede utilizar en una gran variedad de instrumentos, por lo tanto, la primera tarea a realizar será escoger los requisitos adecuados para el sistema de pesaje.

4.2.1. Configuración básica (P o U)

Es necesario decidir si la configuración básica del instrumento se ajusta a un instrumento desarrollado específicamente con software integrado (tipo P) o se ajusta a un instrumento que utilice un ordenador universal (tipo U).

Un instrumento de tipo P tiene un sistema TI integrado con las siguientes características:

- Toda la aplicación se desarrolla para la medición.
- La interfaz de usuario es específica para la medición, es decir, está normalmente en un modo operativo sometido a control legal.
- Si existe un sistema operativo, no hay un intérprete de comandos accesible al usuario, para modificar el programa, enviar comandos al sistema operativo, etc.

En cambio, un sistema de tipo U tiene las siguientes características:

- Está basado en un ordenador de propósito general.
- La interfaz de usuario puede cambiarse de un modo operativo que no esté sometido a control legal a uno que lo esté y viceversa.
- El almacenamiento puede ser fijo (p.ej. disco duro) o extraíble (p.ej. CD_RW).
- Además de la aplicación del instrumento de medida pueden encontrarse a la vez otras aplicaciones de software en el sistema.
- El sistema operativo y los drivers de bajo nivel (p.ej. los drivers de video, del disco, etc.) son legalmente no relevantes a no ser que estén programados para una tarea de medida específica.

Al ver las características que tienen los dos tipos de sistemas se puede concluir que el sistema de pesaje es de tipo P debido a que es un instrumento de medida diseñado y construido específicamente para una tarea concreta. El instrumento de medida no utiliza un ordenador de propósito general para realizar las funciones legalmente relevantes.

4.2.2. Configuraciones TI aplicables

Como se ha comentado anteriormente, las configuraciones TI comprenden cuatro extensiones. La serie de requisitos de la extensión L no se utilizarán debido a que en este proyecto no es necesario el almacenamiento a largo plazo de los datos legalmente relevantes. En cuanto al resto de configuraciones TI sí que serán necesarias.

4.2.3. Requisitos específicos del instrumento (extensión I)

La guía también incluye un conjunto de requisitos específicos para cada instrumento de medida. Estos requisitos se identifican como extensión I. Se trata de extensiones que complementan los requisitos generales del software y que no se pueden considerar de forma independiente de las partes P o U ni de otras extensiones.

El instrumento de pesaje utilizado se incluye en la categoría de instrumento de pesaje de funcionamiento automático (IPFA). En esta categoría se distinguen cinco subcategorías, concretamente el sistema de pesaje pertenece a la categoría de seleccionadoras ponderales automáticas.

De las distintas categorías de los instrumentos de pesaje de funcionamiento automático, sólo los totalizadores discontinuos y los totalizadores continuos se han identificado como susceptibles de necesitar requisitos de software específicos de los instrumentos. Esto se debe a que la medición es acumulativa durante un periodo de tiempo relativamente largo y no se puede repetir si aparece un fallo significativo. Al no ser el caso del proyecto que se está realizando, los requisitos de este apartado no se tendrán en cuenta.

4.2.4. Clase de riesgo aplicable

Los requisitos de la guía se asocian a clases de riesgo. Los riesgos hacen referencia exclusivamente al software del instrumento de medida. Una clase de riesgo se define como la combinación de los niveles requeridos de protección, examen y conformidad del software. Cada categoría presenta tres niveles de riesgo: bajo, medio y alto.

En el caso del proyecto del sistema de pesaje la clase de riesgo es de tipo B, por lo tanto:

- **Nivel de protección del software**, será de nivel medio. El software deberá estar protegido frente a los cambios intencionados realizados utilizando herramientas de software simples, comunes y fácilmente disponibles (p.ej. editores de texto).
- **Nivel de examen del software**, será también de nivel medio. Para ello, se realizan comprobaciones funcionales estándar de examen de modelo del instrumento. Además, el software se debe examinar según su documentación. Para verificar la fiabilidad de la documentación y la eficacia de las medidas de protección, pueden realizarse comprobaciones prácticas aleatorias de las funciones compatibles con el software.
- **Nivel de conformidad del software**, será de nivel bajo. La funcionalidad del software implementado en cada uno de los instrumentos individuales debe de ser conforme con la documentación aprobada.

5. REQUISITOS ESPECÍFICOS PARA EL TIPO P

Como se ha comentado anteriormente, el sistema de pesaje es un instrumento de medida desarrollado específicamente (tipo P). A continuación se indican los requisitos que debe cumplir el sistema de pesaje y, en base a ello, se presenta el programa desarrollado.

La serie de requisitos que se citan a continuación son válidos tanto para los instrumentos como para los subconjuntos de los instrumentos desarrollados específicamente.

5.1. P1: Documentación

A continuación se especifica la documentación requerida para cada uno de los requisitos.

5.1.1. Descripción del software legalmente relevante

El software legalmente relevante son los bloques de función llamados “SistemaPesaje” y “Configuracion_FB”¹. Para poder utilizar estos bloques con PLCs Panasonic y Omron, sin necesidad de volver a hacer el programa de nuevo, se ha escogido el texto estructurado (ST) como lenguaje de programación, un lenguaje de alto nivel designado para controladores lógicos programables (PLCs). Tal y como el nombre indica, está concebido para la programación estructurada, es decir, que para determinadas construcciones de uso frecuente, tales como bucles secuenciales, el lenguaje ST ofrece estructuras establecidas para la programación. Esto permite reducir los errores y dar mayor claridad al programa.

A continuación se va a describir la funcionalidad de los dos bloques de función. Como variables de entrada, de salida y de entrada salida del bloque se explican las de los bloques de función hechos para el PLC de Panasonic. En los bloques hechos para el PLC de Omron hay la diferencia de que algunas variables se han definido como entrada salida, debido a que las variables de tipo “Array” i las de tipo “String” se deben definir de esta manera en el programa CX-Programmer.

¹ FB: (Function Block). Bloque de funciones.

a) SistemaPesaje

Entradas del bloque:

- **EN:** Enable. Permite activar el bloque de función. Pertenece al botón que activa la cinta transportadora.
- **bActivacionPesada:** Cuando está activada se va interrogando el peso continuamente según el tiempo de muestreo especificado. Pertenece al botón que activa la cinta transportadora.
- **bPeticiónPesada:** Petición de pesar producto. Se activa cuando la célula de carga detecta un nuevo producto.
- **iTiempoMuestreo:** Tiempo de muestreo. Es el intervalo de tiempo entre dos pesadas sucesivas (se debe introducir un número entero el cual corresponde al tiempo multiplicado por 10ms).
- **iTimeout:** Especifica el tiempo máximo de espera de respuesta (se debe introducir un número entero el cual corresponde al tiempo multiplicado por 10ms).
- **iDuracionPesada:** Duración del pesaje. Se espera este tiempo para hacer la petición de lectura de peso.
- **bModoBus:** Se activará en caso de tener más de una transductor AED conectado a partir de un bus.
- **iNumeroAEDs:** Número de transductores al bus. Puede haber un máximo de 16 AED. Esta variable solo se utilizará en caso de estar en modo bus.
- **iPuerto:** Indica el puerto de comunicación del PLC por donde se enviarán y se recibirán datos.
- **uiSumaComprobacionParametros:** Suma de comprobación de los parámetros legalmente relevantes del bloque de funciones.

Salidas del bloque:

- **ENO:** Enable negado. Indica la ejecución exitosa del bloque de función.
- **awBufferTransmision:** Contiene un "Array of WORD" enviado desde el PLC.
- **awBufferRecepción.** Contiene un "Array of WORD" recibido en el PLC.
- **bErrorRespuesta:** Bit que indica si la respuesta del transductor digital AED ha sido '?\r\n', que significa que el comando enviado no ha sido valido.
- **diPesoTReal:** Indica el valor del peso en tiempo real.

- **diPesoProducto**: Indica el valor de la última pesada.
- **bPesoTReal**: Bit que indica que se ha recibido una pesada nueva en tiempo real.
- **bPesoProducto**: Esta variable booleana indica que se ha recibido una nueva pesada.
- **bErrorComunicacion**: Bit de alarma que indica que no hay comunicación con el AED.
- **bCalibrar**: Bit que indica que hay que calibrar, debido a que ha sucedido un año y medio desde la última calibración.
- **bParametrosModificados**: Este bit indica que los parámetros del bloque de funciones considerados como realmente relevantes se han modificado.

Entradas Salidas:

- **bPeticiónCero**: Bit que permite enviar el comando de poner a cero el peso mostrado en el visualizador. Una vez comprobado que se tiene como peso 0, se desactiva este bit.
- **iDirecciónBus**. Se utiliza cuando se trabaja en modo bus. Es una entrada salida debido a que en modo bus hay que enviar una consulta de peso a cada AED. Por lo tanto, necesitamos modificar el valor de la variable.
- **bCalibraciónRealizada**: Cuando se pulse el botón de calibración en la pantalla táctil se activa este bit. Es una entrada salida debido a que una vez hecho lo adecuado hay que desactivar la variable.

La Figura 4 muestra el diagrama de estados.

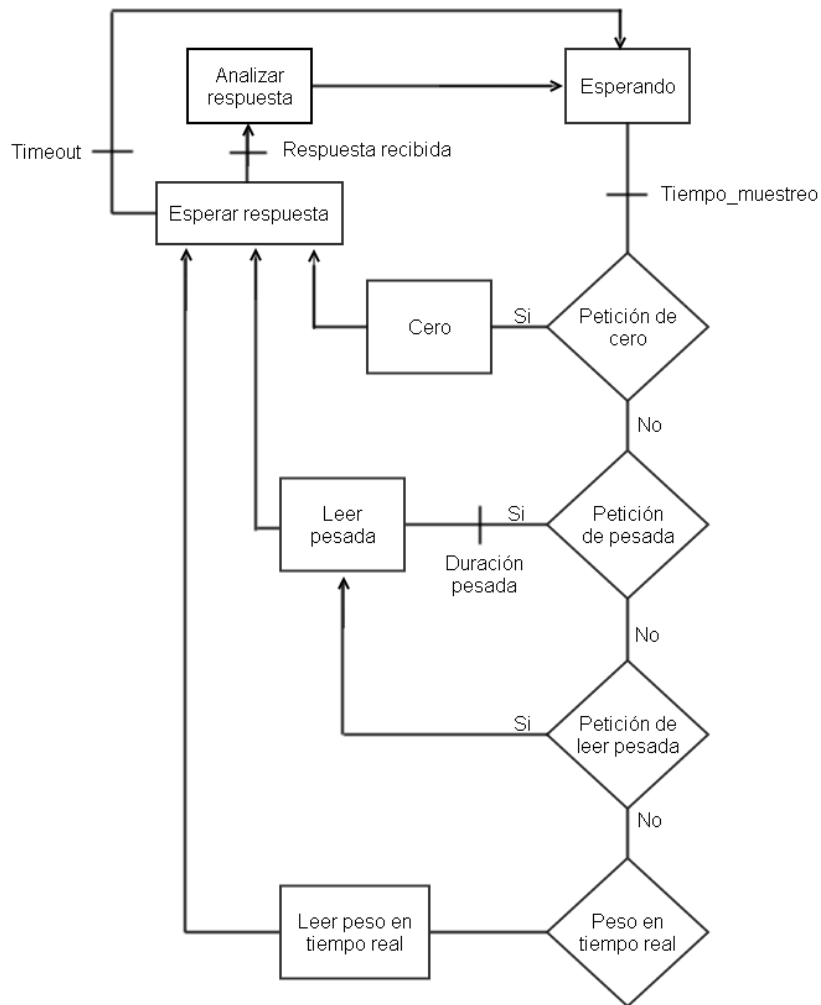


Figura 4. Diagrama de estados del bloque "Sistema_Pesaje"

El bloque de función dispone de una sentencia "CASE" que permitirá indicar en que el estado se encuentra el programa. Al empezar, se encuentra en el estado "Esperando". Pasado el tiempo de muestreo especificado se comprueba si se ha hecho una "petición de cero", una "petición de pesada" o una "petición de leer pesada". En el primer caso se entra en el estado "Cero", donde se envía el comando para poner el peso mostrado en el visualizador a 0. En el segundo caso, se activa un temporizador para que pasado el tiempo definido en la variable "iDuraciónPesada", se entre en el estado "Leer pesada" para enviar el comando de leer el peso. El tercer caso es el mismo que el segundo sin activar el temporizador. En caso de no hacer ninguna de las tres peticiones, se entra en el estado "Leer peso en tiempo real" para enviar el comando que devuelve el peso en tiempo real.

En cualquiera de los tres estados comentados se comprueba si hay modo bus. Esta opción se ha dejado como tarea a ampliar más adelante y no se ha comprobado su correcto funcionamiento. Por lo tanto ahora hay que centrarse en el caso que la variable "bModoBus"=0, donde en estos tres estados, solo se envía por el puerto correspondiente el comando pertinente. Una vez transmitido, el programa se encuentra en el estado "Esperar respuesta".

En el estado "Esperar respuesta" puede suceder que se reciba respuesta o no. Si pasado el tiempo que indica la variable "iTimeout" no se recibe ninguna respuesta, se vuelve al estado "Esperando", donde se volverá a iniciar el ciclo. Si al finalizar el nuevo ciclo no se recibe ninguna respuesta, se activará la entrada salida "bErrorComunicacion", para indicar que no hay una correcta comunicación entre el PLC y el transductor digital AED.

Si se recibe respuesta, se activa un temporizador para luego vaciar el contenido del buffer de recepción. Además, se comprueba si la respuesta del AED ha sido correcta y si es así, se entra en el estado "Analizar respuesta", que pertenece a diferentes estados dentro del bloque de funciones, dependiendo del estado anterior, es decir, dependiendo del estado precedente.

Si se encuentra en el estado "Analizar respuesta 1" es porque se había enviado una "petición de cero". En este caso la respuesta que esperamos es un '0', hecho que indica que el comando ha sido válido. Si no es así, se vuelve al estado "Esperando" para repetir el ciclo y enviar nuevamente la petición de cero. En cambio, si se ha recibido el '0' también se vuelve al estado "Esperando", pero para que se haga una "Petición de leer pesada" con el fin de enviar el comando para leer el peso, para poder comprobar que efectivamente el valor del peso es 0.

Si se encuentra en el estado "Analizar respuesta 2" es porque se había enviado una "petición de pesada" o una "petición de leer pesada". En el primer caso se almacena el valor del peso en una variable de salida del bloque de funciones y se activa otra variable de salida indicando que hay un nuevo peso. En cambio, en el segundo caso hace lo comentado y además se comprueba si el peso es 0. En caso que lo sea, la petición de cero se ha hecho correctamente, y en caso contrario, se volverá a enviar el comando de petición de cero.

Si se encuentra en el estado "Analizar respuesta 3" es porque se había ejecutado el caso "Leer peso en tiempo real". Lo único que hay que hacer es almacenar el valor del peso en tiempo real en una variable de salida del bloque de funciones y activar otra variable de salida indicando que hay un nuevo peso en tiempo real.

Al finalizar el estado *"Analizar respuesta"* se finaliza la sentencia *"CASE"* y el programa no se encuentra en ningún estado, se ejecutan la serie de funcionalidades detalladas a continuación:

- Los temporizadores, que se ejecutan dependiendo del estado en que se encuentre el programa y dependiendo del valor de algunas variables.
- El siguiente trozo de código calcula una suma de comprobación de los parámetros considerados como legalmente relevantes. Si la suma es diferente a la suma de comprobación adecuada para estos parámetros, se activa un bit de alarma indicando que estos parámetros han sido modificados.
- A continuación, se encuentra el software para el aviso de calibración del instrumento de medida. Solo si el bit *"bCalibracionRealizada"* está activado, es decir, cuando se pulse el botón de calibración en la pantalla táctil, se almacena la fecha actual y se calcula la fecha en la cual se deberá volver a calibrar, después de un año y medio. Seguidamente, independientemente de si el bit anterior está activado o no, se calcula la fecha actual para saber si hay que volver a calibrar. Si hay que hacerlo, se activa el bit de alarma *"bCalibrar"*.

Al finalizar el software detallado, se vuelve al estado *"Esperando"* para volver a hacer el ciclo.

En la Taula 1. Correspondencia entre el diagrama de estados y el bloque de función *"SistemaPesaje"* se muestra la correspondencia entre los estados del diagrama de estados de la Figura 1 y los estados del bloque de funciones.

Taula 1. Correspondencia entre el diagrama de estados y el bloque de función *"SistemaPesaje"*

Estado del diagrama de estados	Estado del bloque de funciones
<i>"Esperando"</i>	<i>"Estado"</i> =0
<i>"Cero"</i>	<i>"Estado"</i> =1
<i>"Leer pesada"</i>	<i>"Estado"</i> =2
<i>"Leer peso en tiempo real"</i>	<i>"Estado"</i> =3
<i>"Esperar respuesta"</i>	<i>"Estado"</i> =4
<i>"Analizar respuesta 1"</i>	<i>"Estado"</i> =5
<i>"Analizar respuesta 2"</i>	<i>"Estado"</i> =6
<i>"Analizar respuesta 3"</i>	<i>"Estado"</i> =7

b) Configuracion_FB

Entradas del bloque:

- **EN:** Enable. Permite activar el bloque de función.
- **bLeer:** Cuando esta variable está activada permite hacer una lectura del comando introducido.
- **diP1:** Primer parámetro del comando.
- **diP2:** Segundo parámetro del comando.
- **diP3:** Tercer parámetro del comando.
- **diP4:** Cuarto parámetro del comando.
- **diP5:** Quinto parámetro del comando.
- **iTimeout:** Especifica el tiempo máximo de espera de respuesta (se debe introducir un número entero el cual corresponde al tiempo multiplicado por 10ms).
- **iPuerto:** Indica el puerto de comunicación del PLC por donde se enviarán y se recibirán datos.
- **iPassword:** Si se introduce la contraseña correcta, se permite ejecutar el bloque de función.
- **sDPW:** Permite definir una contraseña, necesaria para la ejecución de algunos comandos.
- **sSPW:** Permite introducir la contraseña necesaria para la ejecución de algunos comandos.

Salidas del bloque:

- **ENO:** Enable negado. Indica la ejecución exitosa del bloque de función.
- **bErrorRespuesta:** Bit que indica si la respuesta del transductor digital AED ha dado error o que el comando que se quería enviar no era válido.
- **bRespuestaOk:** Bit que indica la correcta respuesta del AED.
- **bErrorComunicacion:** Bit de alarma que indica que no hay comunicación con el AED.
- **diRespuestaP1:** Si el comando enviado es de lectura, indica el primer parámetro del comando. En caso de ser de escritura, indica la respuesta.
- **diRespuestaP2:** Indica el segundo parámetro del comando.
- **diRespuestaP3:** Indica el tercer parámetro del comando.

- **diRespuestaP4:** Indica el cuarto parámetro del comando.
- **diRespuestaP5:** Indica el quinto parámetro del comando.
- **awBufferTransmision:** Contiene un “Array of WORD” enviado desde el PLC.
- **awBufferRecepción.** Contiene un “Array of WORD” recibido en el PLC.

Si el comando enviado es de escritura, las 4 últimas respuestas (diRespuestaP2,... P5) son 0, ya que no se deben de considerar.

Entradas Salidas:

- **iNumeroInstruccion:** Si el valor que contiene no es 0 se ejecutará este número de instrucción, siempre que haya la contraseña correcta. Al ser 0 no se ejecutará nada.

La Figura 5 muestra el diagrama de estados, el cual nos permite explicar el programa realizado.

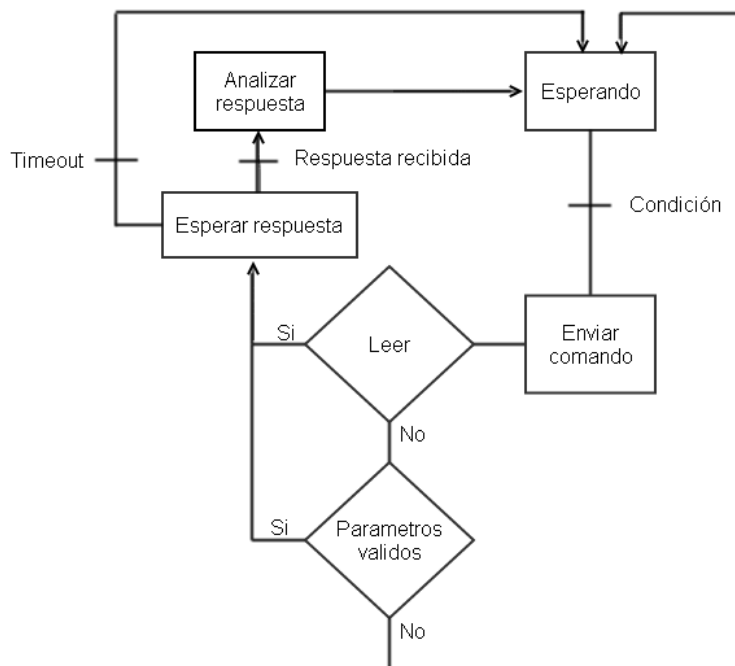


Figura 5. Diagrama de estados del bloque "Configuracion_FB"

Hay que tener en cuenta que para que se ejecute el bloque de función la contraseña introducida debe ser la correcta y hay que activar la variable "EN". Una vez cumplido esto, el bloque de función dispone de una sentencia "CASE" que permitirá indicar en que el estado se encuentra el programa. Al empezar, se encuentra en el estado "Esperando". La condición para

salir de este estado es que el valor de la variable “iNumeroInstruccion” sea diferente de 0. Cuando esto se cumple, se ejecuta el estado “Enviar comando”, que pertenece a diferentes estados dentro del bloque de funciones dependiendo del número de instrucción introducido. Dentro de este estado, si el comando permite una lectura, se comprueba si hay la variable “bLeer” activada. Si es así, se envía el comando indicado de lectura. En caso contrario, si el comando permite una escritura, se comprueba si los parámetros introducidos para que envíe el comando (variables “diP1” a “diP5”) son los permitidos. Si es así, se envía el comando de escritura pertinente. En caso contrario se vuelve al estado “Esperando” y se activa la variable de salida del bloque “bErrorRespuesta”, indicando que el comando no ha sido válido.

Después de enviar el comando por el puerto correspondiente, el programa se encuentra en el estado “Esperar respuesta”.

En este caso, se puede recibir respuesta o no. Si pasado el tiempo que indica la variable “iTimeout” no se recibe ninguna respuesta, se vuelve al estado “Esperando”, donde se volverá a iniciar el ciclo. Si al finalizar el nuevo ciclo no se recibe ninguna respuesta, se activará la entrada salida “bErrorComunicacion”, para indicar que la comunicación no es correcta entre el PLC y el transductor digital AED.

Si se recibe respuesta, se activa un temporizador para luego vaciar el contenido del buffer. Además, se comprueba si la respuesta del AED ha sido correcta y si es así, se entra en el estado “Analizar respuesta”, que depende del estado anterior, es decir, dependiendo de qué comando se había enviado antes.

Dentro de cada estado, si el comando enviado ha sido de lectura se asume que la respuesta es correcta (se activa la variable “bRespuestaOk”) debido a que antes se ha comprobado si la respuesta era válida, y se pone la variable “iNumeroInstruccion” a 0 para indicar que la instrucción se ha hecho con éxito. Además, se almacena la respuesta o respuestas en las variables de salida “diRespuestaP1” a “diRespuestaP5”, dependiendo del número de parámetros que devuelva el comando.

En cambio, si el comando enviado ha sido de escritura se comprueba que la respuesta del AED haya sido ‘0’, hecho que indica respuesta correcta. Si es así, se repite lo mismo, “bRespuestaOk” activada y “iNumeroInstruccion” a 0.

Al finalizar el estado “Analizar respuesta” se finaliza la sentencia “CASE” y el programa no se encuentra en ningún estado, se ejecutan la serie de funcionalidades detalladas a continuación:

- Los temporizadores, que se ejecutan dependiendo del estado en que se encuentre el programa y dependiendo del valor de algunas variables.
- Dependiendo del puerto introducido como variable de entrada, el bit que indica que se han recibido datos por un puerto (“bReceptionDone”) se asigna al bit correspondiente a este puerto.

Al finalizar el software detallado, se vuelve al estado “Esperando” para volver a hacer el ciclo.

En la Taula 2. Correspondencia entre el diagrama de estados y el bloque de función “Configuracion_FB” se muestra la correspondencia entre los estados del diagrama de estados de la Figura 2 y los estados del bloque de funciones.

Taula 2. Correspondencia entre el diagrama de estados y el bloque de función “Configuracion_FB”

Estado del diagrama de estados	Estado del bloque de funciones
“Esperando”	“Estado”=0
“Enviar comando”	“Estado”=1 --> “Estado”=18 ¹
“Esperar respuesta”	“Estado”=40
“Analizar respuesta”	“Estado”=25 --> “Estado”=39, “Estado”=45, “Estado”=46 ²

¹ El estado “Enviar comando” pertenece al “Estado”=1 si el número de instrucción es 1, pertenece al “Estado”=2 si el número de instrucción es 2, y así sucesivamente.

² El estado “Analizar respuesta” pertenece al “Estado”=25 si el estado anterior es 2, pertenece al “Estado”=26 si el estado anterior es 3, y así sucesivamente. Si el estado anterior es 17 o 18, pertenece al “Estado”=45 o el “Estado”=46 respectivamente.

5.1.2. Descripción de la exactitud de los algoritmos de medida

El objetivo del pesaje dinámico es obtener un alto número de pesadas por minuto sin que el producto se detenga. Para conseguirlo se usa la función de trigger, que tiene como objetivo devolver un solo valor de peso al controlador como resultado de la medición dinámica. Esto hace que ya no sea necesario transferir todos los valores medidos (más de 600 por segundo) a través de la interfaz serie durante el pesaje.

La función trigger asociada al transductor digital AED incluye cuatro modos de operación, dos de ellos se activan por nivel y los otros dos por entrada externa. Se ha escogido el modo “External pre-triggering” accesible vía la entrada digital IN1, para detectar que un nuevo

producto entra en la plataforma utilizando un sensor externo localizado al principio de la plataforma (justificándose la denominación “pre-triggering”).

Es decir, la función de trigger se activará cuando la entada digital se active como resultado de detectarse un producto. El funcionamiento de esta función se visualiza en la Figura 6. Al principio hay un tiempo de estabilización (settling time), que es el tiempo que se espera para que el producto a pesar este ubicado completamente sobre la plataforma de pesaje y los filtros digitales respondan. Los valores medidos durante el tiempo de estabilización no se incluirán el resultado de la medición. Al finalizar el tiempo de estabilización, empieza el tiempo de medición (measurement time), donde se calcula la media entre todos los valores medidos y se almacena el valor en la memoria de salida para ser leída con el comando MAV?.

Triggering (TRC command: parameter 1 = 1)

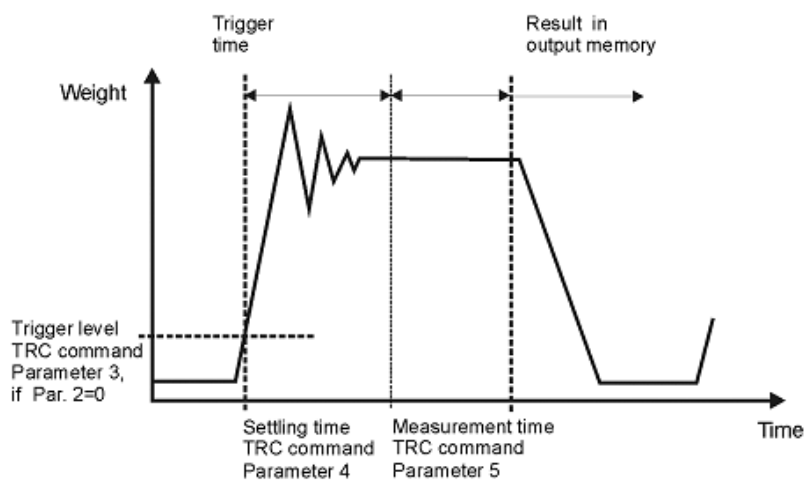


Figura 6. Función de trigger

5.1.3. Descripción del hardware del sistema

a) Omron

En la Figura 7. Diagrama de bloques se puede observar el diagrama de bloques del hardware del sistema de pesaje comandado por el PLC de Omron

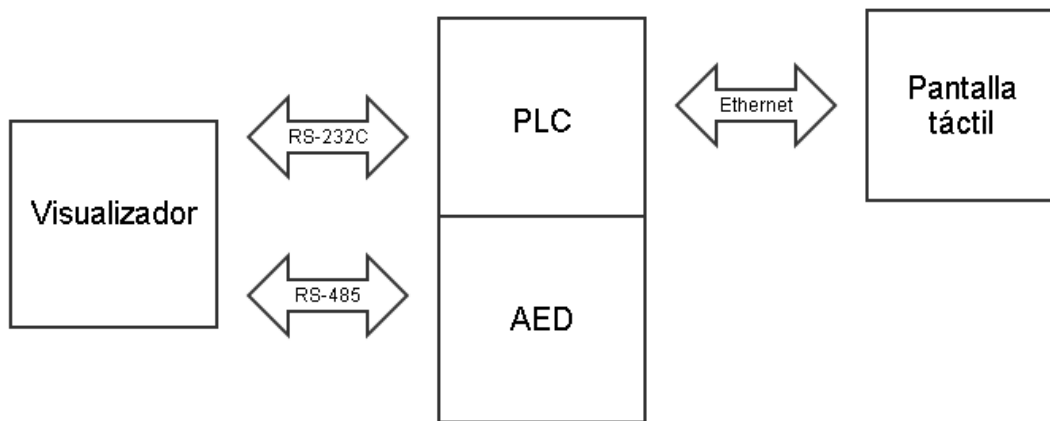


Figura 7. Diagrama de bloques Omron

b) Panasonic

En la Figura 8. Diagrama de bloques Panasonic se puede observar el diagrama de bloques del hardware del sistema de pesaje comandado por el PLC de Panasonic.

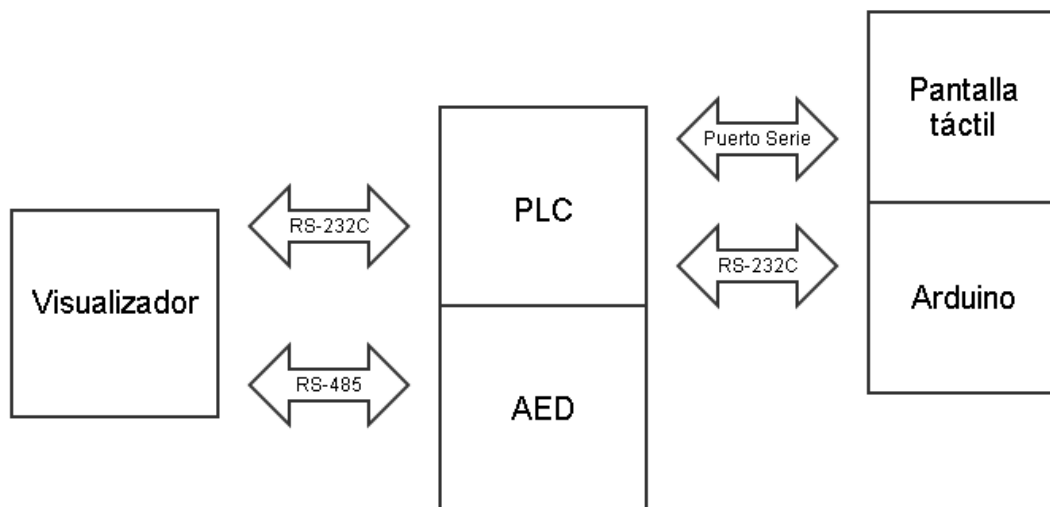


Figura 8. Diagrama de bloques Panasonic

Los diagramas de bloques muestran la configuración genérica del hardware a utilizar. En ellos se pueden diferenciar dos partes, una común y otra diferenciada en función del tipo de PLC a utilizar. De entrada, se han previsto dos configuraciones distintas, pero podría ampliarse a otras configuraciones.

La parte común consta del transductor digital y el visualizador. Con el sistema de pesaje comandado por el PLC de Panasonic también se encuentra en la parte común un Arduino Uno. La parte diferenciada incluye el PLC.

Primeramente, se deben comentar las dos tipos de interfícies utilizadas. La interfície RS-232 se utiliza para el intercambio de datos binarios entre equipos que se encuentran a distancias cercanas (de hasta unos 15 metros) y con velocidades de transmisión bajas. La interfície RS-485 es un estándar de comunicaciones en bus ideal para transmitir a altas velocidades sobre largas distancias (35Mbit/s hasta 10 metros y 100 kbit/s en 1200 metros) y a través de canales ruidosos, ya que reduce los ruidos que aparecen en los voltajes producidos en la línea de transmisión.

A continuación se describe cada uno de los bloques mostrados en la Figura 7. Diagrama de bloques

5.1.3.1 AED, visualizador y Arduino Uno

Para las tareas de medición se utilizará el transductor digital AED9101D. Este transductor incluye una placa de un amplificador digital AD103C, dispositivos base con una interfaz serie y sensores inteligentes con procesamiento de señal integrado. El propósito de estos componentes es convertir las señales analógicas suministradas por las células de carga en datos digitales que se guardan en la memoria del transductor AED. El sistema de pesaje dispone de dos células de carga en la sección de pesaje ilustrada en la Figura 1, una célula de carga justo cuando el producto entra en la sección y la otra cuando sale.

El componente principal del AED escogido es el amplificador digital AD103C, que hace su control de procesos y tareas de monitoreo fácil y conveniente. El AED103C se encarga de digitalizar las señales analógicas de las células de carga.



Figura 9: Transductor AED

Por lo tanto, como se ha comentado, los valores medidos se almacenan en la memoria del AED, el cual está conectado con el visualizador por un puerto RS-485, para que pueda verse el peso en tiempo real por pantalla. De esta manera, el controlador se conectará con el visualizador por un puerto RS-232C, para obtener los datos del pesaje.

El visualizador DWS2103 permite la conexión de diversos transductores digitales por el puerto RS485. Solo un dispositivo puede estar activo a la vez. El peso en tiempo real que devuelva el transductor activo se podrá visualizar por pantalla y este podrá ser parametrizado.

En el caso de este proyecto se usará solo un transductor AED. El visualizador soporta todas las funciones implementadas en el AED y actuará de pantalla principal del transductor digital.

En la Figura 10 se puede observar la pantalla digital DWS2103.



Figura 10: Pantalla digital DWS2103

Para tareas de control se utilizará el Arduino Uno. En el apartado 5.2. P2: Identificación del software se detallara cual es el papel de este dispositivo en el proyecto.

Arduino es una plataforma de desarrollo de computación física de código abierto, basada en una placa que contiene un microcontrolador y un entorno de desarrollo para crear software para la placa.

Se puede usar Arduino para crear objetos interactivos leyendo datos de una gran variedad de interruptores, sensores, controlar todo tipo de luces, motores y otros actuadores físicos. El proyecto de Arduino puede ser autónomo o comunicarse con otros programas. La placa se puede montar al gusto del consumidor o también se puede comprar ya hecha para usar.

En este caso se ha utilizado Arduino Uno, uno de los más sencillos, pero con sus especificaciones sería suficiente.

Cuenta con las siguientes especificaciones, entre otras:

- a) 14 pines entrada / salida (6 pueden ser PWM)
- b) 6 entradas analógicas
- c) Un oscilador de 16MHz
- d) Conexión USB
- e) Conversor ADC
- f) Microcontrolador basado en ATMEGA328

- g) 32Kb de memoria de programa, 2Kb de SRAM y 1Kb de EEPROM
- h) Puerto serie
- i) Timers

Para comunicar el Arduino con el PLC se ha utilizado un conversor RS-232 TTL, un dispositivo muy usado para modificar las memorias flash de smartphones, GPS, DVD y discos duros. En la Figura 11. Conversor RS-232 TTL se puede observar este componente.



Figura 11. Conversor RS-232 TTL

5.1.3.2. PLC

El PLC se encargará de leer las pesadas de los productos o el peso en tiempo real y enviar los valores a la pantalla táctil, donde se podrán visualizar. Al mismo tiempo, el peso en tiempo real será visible en el visualizador.

Referente a la transmisión y recepción de datos entre el PLC y el AED, se realiza con el modo “No-protocol”. Se ha escogido este modo porque es simple, los datos pueden ser enviados y recibidos a través del puerto de comunicaciones de E/S con facilidad, sin conversión o protocolo. El modo “No-protocol” no dispone de procedimientos de comunicaciones tales como el procesamiento de reintento y la conversión del formato de datos.

a) Omron

Para el PLC de Omron se ha escogido el modelo CJ2M, ideal para necesidades de empaquetado y automatización general de máquinas. Para alimentar el controlador se le ha agregado el modulo PA202 y para poder comunicarse con el visualizador se le ha agregado el modulo SCU41-V1, que incluye un puerto RS-232 y un puerto RS-485.

En la Figura 12 se puede observar el PLC con los dos módulos integrados.



Figura 12: PLC Omron

a) Panasonic

Para el PLC de Panasonic se ha escogido el modelo FP-Sigma, un modelo de controlador lógico programable compatible con todos los controladores de Panasonic. Cuenta con un diseño estructural compacto y es capaz de comunicarse con todos los principales productos de la automatización. En la Figura 13: PLC Panasonic se puede observar el controlador comentado.



Figura 13: PLC Panasonic

5.2. P2: Identificación del software

El software legalmente relevante debe estar claramente identificado. La identificación del software estará inequívocamente ligada al mismo. Deberá presentarse mediante un comando o durante el funcionamiento (Apartado 4.2 [\[1\]](#)).

Solamente el software legalmente relevante dispone de la identificación del software. Se está hablando de los bloques de función llamados “SistemaPesaje” y “Configuracion_FB”, que solo pueden ser modificados por el reparador autorizado que disponga de la contraseña para poder editar este software.

a) Omron

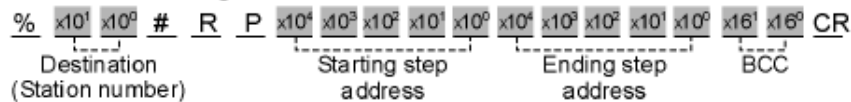
Dentro de los dos bloques se ha añadido una variable de entrada salida que se actualiza automáticamente con la última data que se ha modificado el software. La identificación del software del bloque más importante, el bloque del pesaje, se podrá visualizar en la pantalla principal de la pantalla táctil.

b) Panasonic

Para el sistema de pesaje comandado por un PLC de Panasonic se ha escogido la opción de utilizar un sistema lógico complementario encargado de identificar el software. Un factor importante a determinar en la elección de tal sistema ha sido que utilice un interface de comunicaciones compatible con el PLC. Se ha escogido el Arduino Uno debido a su bajo precio y sus buenas prestaciones, un dispositivo que se ajusta perfectamente en el sistema.

Los PLC's de la serie FP de Panasonic disponen de un protocolo de comunicaciones llamado Mewtocol-Com que dispone de una serie de instrucciones que dan una funcionalidad extra al PLC. Este protocolo es utilizado para la comunicación entre el tipo de PLC's de la serie FP y un ordenador, en nuestro caso, se ha utilizado el Arduino. Ha sido necesario configurar un puerto del PLC con este protocolo y conectar los dos componentes a partir del conversor RS-232 TTL.

El protocolo comentado dispone de la instrucción RP, que permite leer el programa guardado en el PLC. A continuación se puede observar el formato de las tramas de transmisión y de recepción de la comanda detallada.

Command message**Response message**

El programa situado entre las direcciones especificadas es retornado por el controlador programable.

Por lo tanto, el software cargado en el Arduino se encarga de leer cada un tiempo establecido el software guardado en el PLC y calcular una suma de comprobación, que se comprobará con la suma de comprobación adecuada. Si coinciden, se permitirá que el sistema de pesaje actúe con normalidad debido a que el programa no se habrá modificado. En caso contrario, se activará una alarma en el sistema de pesaje y se desactivará su funcionamiento.

Se quería conseguir la identificación del software de los bloques de función legalmente relevantes pero no es posible con este protocolo, debido a que al modificar el software legalmente no relevante la suma de comprobación del bloque legalmente relevante cambia de valor. Por lo tanto, se ha optado por proteger todo el programa.

5.3. P3: Influencia sobre el software a través de la interfaz de usuario

Los comandos introducidos a través de la interfaz de usuario no influirán en el software legalmente relevante ni en los datos de medida de forma inadmisibile (Apartado 4.2 [\[1\]](#)).

Hay una serie de comandos que se pueden enviar al AED a partir de la pantalla. Estos comandos se transfieren gracias al bloque de funciones "Configuracion_FB". Dentro de este bloque se ha definido una contraseña para que el reparador autorizado pueda interactuar con el bloque a partir de la pantalla, solo si conoce la contraseña.

A continuación se cuenta la finalidad de todos los comandos que pueden ser enviados al AED mediante el bloque de funciones. El resto de comandos no documentados no se pueden transmitir y no tendrán ningún efecto en los datos de medida legalmente relevantes. Para más

información acerca de los diversos comandos documentados consultar el manual del amplificador AED103C [2].

- INSTRUCCIÓN 1 – Select

Este comando se utiliza cuando hay varios AED conectados entre sí en un bus. Permite seleccionarlos individualmente o conjuntamente. Si se hace de forma individual, los otros nodos del bus no responderán a las peticiones del controlador.

- INSTRUCCIÓN 2 – COF

Se utiliza para configurar el formato de la trama de respuesta de los dos siguientes comandos:

- Leer el peso en tiempo real (MSV?)

- Leer una pesada (MAV?)

- INSTRUCCIÓN 3 – IMD

Sirve para escoger la función de las entradas digitales IN1 i IN2.

- INSTRUCCIÓN 4 – TDD

Este comando sirve para guardar i activar parámetros. El AED dispone de dos memorias EEPROM. Los parámetros personalizados se pueden guardar a la primera EEPROM para protegerlos en caso de un fallo de alimentación. La segunda EEPROM contiene la configuración de fábrica siendo esta no accesible.

- INSTRUCCIÓN 5 – TRC

Se utiliza para configurar la función de trigger.

- INSTRUCCIÓN 6 – TAS

El comando permite configurar la lectura del peso a realizar, peso neto o peso bruto, a partir de los dos comandos para leer peso: “Measured Alternative Data” (MAV?) y “Measured Signal Value” (MSV?).

- INSTRUCCIÓN 7 – PVS

Este comando activa los valores extremos y establece como tiene que ser el valor medido: peso neto, peso bruto o resultado de la función de trigger.

- INSTRUCCIÓN 8 – LFT

El comando permite activar el modo “Legal for trade” (LFT). Cuando esta activado, $LFT > 0$, hay una serie de comandos que solo podrán realizar una lectura: “Cyclic Redundancy Check” (CRC), “Calibration Weight” (CWT), “Load Cell Dead Weight” (LDW), “Load Cell Weight” (LWT). Hay más comandos afectados por el LFT, pero aquí solo aparecen los comandos que se pueden enviar gracias al bloque “Configuracion_FB”.

- INSTRUCCIÓN 9 – DPW

Se usa para definir una contraseña, necesaria para la ejecución de algunos comandos.

- INSTRUCCIÓN 10 – SPW

El comando SPW con la contraseña correcta autoriza la utilización de varios comandos protegidos por contraseña: “Calibration Weight” (CWT), “Load Cell Dead Weight” (LDW), “Load Cell Weight” (LWT), “Legal for trade” (LFT). Solo aparecen los comandos permitidos por el bloque “Configuracion_FB”.

El uso de los comandos protegidos se inhibe después de una caída de tensión.

- INSTRUCCIÓN 11 – FMD

El comando permite seleccionar el tipo de filtros digitales disponibles.

- INSTRUCCIÓN 12 – ASF

El comando permite seleccionar un filtro de paso bajo digital.

- INSTRUCCIÓN 13 – ICR

Define la frecuencia de muestreo de los valores medidos en una consulta de medición (MSV?).

- INSTRUCCIÓN 14 – LDW

Permite ajustar el punto cero de la curva característica de usuario.

- INSTRUCCIÓN 15 – LWT

Este comando determina el valor medido para la gran escala de la curva característica de usuario.

- INSTRUCCIÓN 16 – CWT

Permite la calibración del pesaje para la calibración de carga parcial.

- INSTRUCCIÓN 17 – TCR

Indica el contador del modo “Legal for trade”. Este contador no puede ser restaurado y su función es indicar los cambios de parámetros del comando LFT.

- INSTRUCCIÓN 18 – CRC

Este comando permite calcular externamente una suma de comprobación de todos los parámetros del AED y almacenarlo en el AED. Esto permite que cada intento de manipulación de los parámetros del AED sea detectado.

- INSTRUCCIÓN 19 – CSM

Este comando permite seleccionar la función del estado de la medición. El estado de la medición se usa en los dos siguientes comandos:

- Leer el peso en tiempo real (MSV?)

- Leer una pesada (MAV?)

Si CSM = 1, se calcula el checksum (EXOR) a partir del valor medurado y su valor se muestra en lugar del estado de la medición.

5.4. P4: Influencia sobre el software a través de interfaces de comunicación

Los comandos introducidos a través de interfaces de comunicación del instrumento no influirán en el software legalmente relevante ni en los datos de medida de forma inadmisibles (Apartado 4.2 [1]).

Tal y como se ha comentado en la restricción anterior, solo el reparador autorizado puede enviar una serie de comandos a partir a partir de la pantalla táctil. Esta tarea no puede hacerla cualquier usuario debido a que una configuración errónea de parámetros podría afectar en los datos de medida.

5.5. P5: Protección frente a cambios accidentales o no intencionados

El software legalmente relevante y los datos de medida estarán protegidos frente a modificaciones involuntarias (Apartado 4.2 [\[1\]](#)).

Para proteger el software legalmente relevante, el transductor digital AED dispone de la opción de calcular un checksum a partir del valor del peso. Para activarlo hay que poner el comando CSM=1 i el comando COF=8. Al configurar el AED de esta forma, al leer el valor medido se obtiene en el controlador una trama con el valor medido y el checksum, en formato binario.

a) Omron

Con el PLC de Omron no se ha utilizado el checksum debido a que se ha utilizado un único bloque de función con lenguaje ST para obtener el valor del peso, y la única instrucción disponible para recibir datos provenientes del AED almacena los datos en una variable de tipo "String", tipo no adecuado para tratar con datos binarios.

Debido a lo comentado y que se trabaja con una red cerrada completamente sometida a control legal y con un solo participante, no se ha considerado necesario obtener un campo de checksum para comprobar la integridad de los datos.

b) Panasonic

De todas formas, con el PLC de Panasonic se ha obtenido el valor del checksum, debido a que la instrucción utilizada para recibir datos provenientes del AED almacena los datos en una variable de tipo "Array of Word". Después de varias pruebas se observado un defecto en el instrumento de medida, debido a que al utilizar el checksum, el valor del pesaje no es suficientemente estable. Por este motivo, no se ha utilizado el checksum.

5.6. P6: Protección ante las modificaciones intencionadas.

El software legalmente relevante estará protegido ante modificaciones, cargas o intercambios inadmisibles de la memoria hardware (Apartado 4.2 [\[1\]](#)).

No hay posibilidad de intercambiar la memoria del PLC debido a que tiene memoria interna y no se puede extraer. Por lo tanto, se garantiza que a partir de la memoria que contiene el software legalmente relevante no se pueda manipular el software.

5.7. P7: Protección de parámetros

Los parámetros que fijan las características legalmente relevantes del instrumento de medida estarán protegidos frente a modificaciones no autorizadas (Apartado 4.2 [\[1\]](#)).

Un vez finalizada la configuración de parámetros se ha activado el modo "Legal for trade" (LFT>0) en el AED, para que haya una serie de comandos que sólo permitan hacer una lectura, si se intenta hacer una escritura se obtendrá como respuesta '?crlf'. Por lo tanto, estos parámetros quedarán protegidos y no se podrá modificar su valor. Esta instrucción (LFT) se ha descrito en el apartado 5.3. P3: Influencia sobre el software a través de la interfaz de usuario, donde se han definido una serie de comandos que pueden transmitirse.

Los parámetros que pueden configurarse en el visualizador están documentados en el manual del visualizador DWS2103. Se han considerado como parámetros protegidos, es decir, después de configurar-los adecuadamente a partir del menú de opciones que dispone, la unidad certificada ha protegido el instrumento y ha precintado el interruptor que contiene. A partir del visualizador no será posible modificar los parámetros ni configuración legalmente relevante. Solo se podrá hacer después de pulsar el interruptor, pero para ello hay que quitar la protección, ya que está precintado.

Respecto al bloque de funciones del pesaje ("SistemaPesaje"), se han considerado una serie de parámetros como legalmente relevantes, los que se podrán modificar desde la pantalla táctil solo si dispones de una contraseña, debido que la modificación de sus valores afecta en la medición. Estos parámetros pertenecen a las variables del tiempo de muestreo, el timeout y la duración de la pesada.

Para saber si estos tres parámetros han sido modificados se ha realizado una suma de comprobación, que se compara con la suma de comprobación adecuada para estos parámetros. En caso de que sea diferente, se activa una salida del bloque de funciones llamada "bParametrosModificados", indicando como dice el nombre de la variable que los parámetros legalmente relevantes han sido modificados. Al suceder esto, en la pantalla principal de la pantalla táctil aparecerá un mensaje de alerta.

6. EXTENSIÓN T: TRANSMISIÓN DE DATOS DE MEDIDA MEDIANTE REDES DE COMUNICACIÓN

Esta es una extensión de los requisitos del software de las guías P y U. La extensión en cuestión sólo debe utilizarse si los datos de medida se transmiten a través de redes de comunicación a un dispositivo remoto donde se terminen de procesar y/o se utilicen para fines regulados legalmente.

Por lo tanto, es el caso del sistema de pesaje. La configuración de red que se utilizará es una red cerrada, completamente sometida a control legal. Solo hay un participante conectado, con una identidad, funcionalidad y ubicación claras.

6.1. T1: Completitud de los datos transmitidos

Los datos transmitidos deben contener toda la información relevante necesaria para presentar o procesar posteriormente los resultados de medida en la unidad receptora (Apartado 7.2 [\[1\]](#)).

Se ha considerado suficiente para su procesamiento posterior que el conjunto de datos recibidos al leer una pesada solo contenga un parámetro y el delimitador.

Para escoger el formato de la trama de respuesta se ha utilizado la instrucción COF3, que permite obtener la siguiente trama, con codificación ASCII:

- Valor medido (8 caracteres).
- CRLF: Delimitador. Retorno de carro y salto de línea (2 caracteres).

6.2. T2: Protección frente a los cambios accidentales o no intencionados

Los datos transmitidos estarán protegidos frente a cambios accidentales o no intencionados (Apartado 7.2 [\[1\]](#)).

Tal y como se ha comentado en el apartado 5.5. P5: Protección frente a cambios accidentales o no intencionados, no se ha considerado necesario utilizar una suma de comprobación para comprobar la integridad de los datos.

6.3. T3: Integridad de los datos

Los datos transmitidos legalmente relevantes deben estar protegidos frente a cambios intencionados realizados por herramientas software (Apartado 7.2 [\[1\]](#)).

Este requisito no se aplica a una red cerrada completamente sometida a control legal, como es el caso del sistema de pesaje.

6.4. T4: Autenticidad de los datos transmitidos

El programa que reciba los datos relevantes transmitidos, deberá poder verificar la autenticidad y la asignación de los valores de medida a una medición determinada (Apartado 7.2 [\[1\]](#)).

Nos encontramos delante de una red cerrada, donde solo hay un participante. No se ha considerado necesario aplicar esta restricción debido a que se conoce perfectamente de donde vienen los datos.

6.5. T5: Confidencialidad de les claves

Las claves y los datos que las acompañan deben tratarse como datos legalmente relevantes y mantenerse ocultos y protegidos frente a posibles riesgos originados por herramientas software (Apartado 7.2 [\[1\]](#)).

Este requisito sólo se aplica si hay una clave secreta en el sistema. Al no ser necesario la utilización de una clave secreta, este requisito no se tendrá en cuenta.

6.6. T6: Tratamiento de datos corruptos

Si se detectan datos corruptos, éstos no deben utilizarse (Apartado 7.2 [\[1\]](#)).

Tal y como se ha comentado en el apartado 5.5. P5: Protección frente a cambios accidentales o no intencionados, no se ha considerado necesario utilizar una suma de comprobación para comprobar la integridad de los datos.

En cambio, si el comando enviado al transductor AED no es válido, la respuesta será '?crif'. Al recibir esto se considerará como un fallo de comando transmitido y se activará la variable de salida del bloque de funciones "SistemaPesaje" llamada "bErrorRespuesta".

a) Omron

Al activarse el bit indicado, en el sistema de pesaje comandado por el PLC de Omron, los valores medidos durante este tiempo se descartaran debido a que es un proceso continuo, no se puede parar la máquina y esperar a leer estos valores. Por lo tanto, los productos en que no se haya leído su peso se descartaran, irán a la zona de rechazo, para que se pueda leer su peso más adelante.

b) Panasonic

Al activarse el bit "bErrorRespuesta", en el sistema de pesaje comandado por el PLC de Panasonic, se inhibirá el funcionamiento de la cinta transportadora y no se etiquetará el producto, debido a que no se ha obtenido el peso de forma correcta.

6.7. T7: Retraso en la transmisión

Los retrasos en la transmisión no influirán de forma inadmisibile en la medición (Apartado 7.2 [1]).

Cuando la célula de carga detecta un nuevo producto en la cinta se activa un bit en el bloque de función "SistemaPesaje", que automáticamente activa un temporizador para qué pasado el tiempo que dura una pesada se envíe el comando de leer el peso y se guarde el peso. Se ha comprobado que en el peor de los casos, los posibles retardos en la transmisión no afectaran a los datos de medida.

6.8. T8: Disponibilidad de los servicios de transmisión

Si los servicios de red dejan de estar disponibles, no se debe perder ningún dato de medida (Apartado 7.2 [\[1\]](#)).

Si el controlador no recibe respuesta a las peticiones que hace, pide una vez más el valor de la medición. Si al segundo intento tampoco recibe respuesta, se activará una salida del bloque de funciones “SistemaPesaje” llamada “bErrorComunicacion”.

Al activarse este bit, se realizará el mismo procedimiento que el comentado en el apartado 6.6. T6: Tratamiento de datos corruptos.

7. EXTENSIÓN S: SEPARACIÓN DEL SOFTWARE

La separación del software es una metodología de diseño opcional que permite al fabricante modificar fácilmente el software legalmente no relevante.

En general, los instrumentos de medida o sistemas controlados por software disponen de una funcionalidad compleja y contienen módulos legalmente relevantes y módulos que no lo son. Es una ventaja para el fabricante y por el examinador separar estos módulos de software del sistema de medida, aunque no es obligatorio.

7.1. S1: Realización de la separación del software

Habrà una parte del software que contenga todo el software y los parámetros legalmente relevantes que estará claramente separada de las otras partes del software (Apartado 8.2 [\[1\]](#)).

El bloque de función “SistemaPesaje” permite guardar en una variable el valor del peso obtenido a partir de la función de trigger, leer el peso en tiempo real y hacer una petición de cero. El bloque “Configuracion_FB” permite configurar varios parámetros del transductor digital AED, que algunos contribuyen al cálculo del valor de medida. Por lo tanto, esto dos bloques de función forman parte del software legalmente relevante.

Para modificar el software de los bloques de función comentados se necesita disponer de una contraseña. Así, solo el reparador autorizado podrá hacer modificaciones en este software.

El bloque “SistemaPesaje” se ha protegido para escritura y el bloque “Configuracion_FB” para escritura y visualización, debido a que se dispone de una contraseña que permite identificarte para interactuar con el bloque, es decir, para poder enviar comandos al AED.

7.2. S2: Indicación mixta

La información adicional generada por el software que no es legalmente relevante, sólo podrá mostrarse en pantalla o impresa, en caso de que no haya posibilidad de confusión con la información originada en la parte legalmente relevante (Apartado 8.2 [\[1\]](#)).

La información que va a mostrar por pantalla el software legalmente no relevante se transfiere a través de la interfaz protectora al software legalmente relevante. La interfaz protectora es el bloque de función, que dentro tiene definido una serie de entradas, salidas y entradas salidas.

Desde el software legalmente no relevante se introduce el valor de las variables de entrada y algunas variables de entrada salida, y al tener definido el tipo de cada variable no permite que introduzcas un valor de otro tipo. Es decir, se inserta información admisible dentro del software legalmente relevante, que devuelve por pantalla el resultado de las variables de salida dependiendo de lo que se ha introducido.

7.3. S3: Interfaz de software protectora

El intercambio de datos entre el software legalmente relevante y el software que no lo sea se hará mediante una interfaz de software protectora, que incluya las interacciones y el flujo de datos (Apartado 8.2 [\[1\]](#)).

La interfaz de software protectora son los bloques de funciones, como se ha comentado. Desde el software legalmente no relevante se permite actuar sobre el software legalmente relevante modificando el valor de las entradas y entradas salidas del bloque, en lugar de modificar directamente el software.

La descripción completa de los dos bloques de función que forman parte del software legalmente relevante se encuentra en el apartado 4.1. a).

La finalidad de encapsular los datos dentro de los bloques de función permite declarar variables locales en la parte legalmente relevante. La interfaz se implementa como una subrutina que pertenece al software legalmente relevante y es llamada desde el software legalmente no relevante.

8. EXTENSIÓN D: DESCARGA DE SOFTWARE LEGALMENTE RELEVANTE

Esta extensión es de aplicación a la descarga de software legalmente relevante mientras las características metrológicas permanezcan inalteradas y la declaración de la conformidad sea válida. La descarga del software legalmente relevante debe cumplir los requisitos comentados a continuación.

8.1. D1: Mecanismo de descarga

La descarga y la instalación posterior del software serán automáticas y garantizará que al finalizar el proceso el entorno de protección del software se encuentre en el nivel aprobado (Apartado 9.2 [\[1\]](#)).

Para realizar la descarga se debe conectar un ordenador al controlador a partir de un cable USB, o también se puede hacer a través de un cable Ethernet. Antes de transferir el software al PLC hay que introducir una contraseña. Es decir, se comprobará la autenticidad, solo podrá transferir el programa el reparador autorizado. Si se introduce correctamente la contraseña, se instala automáticamente el software en la ubicación correcta dentro del controlador. Cada sistema de control tiene su propio método de descarga.

Si la descarga del software al PLC falla se mostrará una advertencia por la pantalla del ordenador. El estado del controlador puede verse afectado, mostrará un mensaje de error permanente y su funcionamiento se inhibirá hasta que se corrija la causa del error.

Durante la descarga y la instalación posterior del software descargado, se desactiva automáticamente la cinta transportadora y la función de medición del instrumento, ya que no se puede garantizar una medición correcta durante este tiempo. Una vez finalizada la descarga se puede volver a activar la cinta y la medición mediante el pulsador de permiso de marcha.

8.2. D2: Autenticación del software descargado

Se emplearán medios para garantizar que el software descargado es auténtico, y para indicar que ha sido aprobado por un organismo notificado (Apartado 9.2 [\[1\]](#)).

El PLC no comprobará automáticamente si el software descargado es auténtico y si está aprobado para este modelo de instrumento de medida.

Solo el reparador autorizado podrá descargar el software, debido a que se ha configurado el programa utilizado para programar el PLC para que pida una contraseña antes de transferir el software. Si el reparador autorizado tiene intención de actualizar el software legalmente relevante deberá comunicar los cambios al organismo notificado responsable. Este decide si es o no necesario una adicional al certificado de examen de modelo.

En el caso que se descarga el software sin consultar al organismo notificado, la identificación del software asignada de forma inequívoca a la versión aprobada del software no será la misma que la documentada en el apartado 5.2. P2: Identificación del software

8.3. D3: Integridad del software descargado

Se emplearán medios para garantizar que durante la descarga el software descargado no haya sido modificado de forma inadmisibile (Apartado 9.2 [\[1\]](#)).

Este requisito no se aplicará debido a que el método de transferencia del software es considerado fiable. Como se ha dicho anteriormente se descarga el software a partir de un cable USB que conecta el controlador con el ordenador, o también se puede hacer a través de un cable Ethernet.

8.4. D4: Trazabilidad de la descarga del software legalmente relevante

Se garantizará mediante los medios técnicos adecuados que las descargas del software legalmente relevante puedan rastrearse dentro del instrumento para realizar controles posteriores (Apartado 9.2 [\[1\]](#)).

No se aplicará este requisito debido a que tanto el PLC de Omron como el de Panasonic no permiten rastrear las descargas del software legalmente relevante durante un período de tiempo determinado.

9. VALIDACIÓN

Este capítulo describe las pruebas realizadas con el sistema de pesaje hasta conseguir su correcto funcionamiento.

Primeramente, se ha procedido a validar la transmisión de información entre equipos, en concreto se ha procedido a comprobar que se enviaban las tramas adecuadas por el puerto RS232 del PLC. Para ello, ha sido necesario conectar el puerto de programación del PLC con el conector adecuado y este conectarlo con un conversor USB para RS232 para así disponer de un puerto USB que se conecta con el ordenador. Gracias a esta conexión se puede transferir el programa al PLC. Para comprobar las tramas enviadas y recibidas por el puerto del PLC se ha conectado un conversor USB desde el puerto hasta el ordenador. Se ha utilizado el programa Docklight Scripting, correctamente configurado, para poder observar el dialogo. Este programa es una herramienta para protocolos de comunicación serie, que permite monitorizar las comunicaciones entre dos dispositivos serie o testear la comunicación serie de un solo dispositivo.

Después de varias pruebas se ha conseguido enviar los comandos adecuados por el controlador y se ha proseguido a comprobar la comunicación entre el PLC y el visualizador DWS2103.

Para ello, hay que hacer las conexiones adecuadas para el correcto funcionamiento del AED y conectarlo con el visualizador por el puerto RS485. A continuación se debe alimentar el visualizador y comprobar si detecta el AED. Si no lo detecta habrá que revisar las conexiones hasta conseguirlo. Una vez establecida la conexión, ya se puede conectar el visualizador por el puerto RS232 hasta el puerto RS232 del PLC con el conector adecuado. A partir de este puerto del PLC se enviaran comandos al visualizador y se recibirá la respuesta. Se ha observado el dialogo entre los dos dispositivos mediante el programa Docklight Scripting.

Una vez confirmada la correcta comunicación entre el PLC y el visualizador, se prosigue a desarrollar el software del bloque de función del pesaje ("SistemaPesaje") para conseguir capturar las respuestas del visualizador a las peticiones realizadas.

Para validar el correcto funcionamiento del bloque de función del pesaje se ha comprobado que todos los estados de la máquina de estados correspondiente a la Figura 4 funcionaran como es debido.

Seguidamente, hay que conectar el AED con las células de carga y configurar el AED para que calcule correctamente la pesada. Para ello, se ha conectado el AED con un ordenador, a partir del conector adecuado y un conversor USB para poder conectar al puerto USB del ordenador. Se ha configurado el AED a partir del programa AED_Pannel.

Una vez configurado se han hecho pruebas pesando productos para comprobar si se obtenía el peso deseado. Después de varias pruebas cambiando algún parámetro del AED se ha conseguido el funcionamiento deseado del sistema de pesaje.



10. CONCLUSIONES

Este proyecto me ha permitido introducirme al mundo de la automatización, precisamente al mundo del pesaje. En la asignatura de “Sistemes automàtics i robotica” habíamos visto una pequeña introducción a la programación para PLCs. Me pareció interesante y por eso he realizado el trabajo final de grado sobre esta temática. Referente a este proyecto, puedo concluir que juntado los bloques de funciones que he desarrollado con el software proporcionado por la empresa, se ha obtenido un buen funcionamiento del sistema de pesaje. Para el sistema de pesaje comandado por el PLC de Omron se han obtenido unas 120 pesadas por minuto y para el sistema comandado por el PLC de Panasonic se han obtenido unas 40 pesadas por minuto. Este factor depende también de la velocidad de la cinta transportadora, la distancia entre productos, la longitud de los productos, etc. La principal diferencia entre el primer sistema y el otro es que para imprimir la etiqueta al producto se hace en estático y una vez imprimida se vuelve a activar la cinta transportadora. En cambio, en el primer sistema de pesaje se calcula el peso y si es necesario se rechaza, todo en dinámico.

Se debe destacar que el hecho de haber de homologar el software ha representado realizar el programa de una forma diferente al que se había hecho hasta ahora. Para solventar el problema de la identificación del software, en el sistema de pesaje comandado por el PLC de Panasonic, se ha utilizado el Arduino, un dispositivo que permite adaptarse a un gran número de aplicaciones.

11. AGRADECIMIENTOS

Quería agradecer a la Teresa Escobet que es mi tutora del proyecto, por el soporte y la confianza que me ha aportado a lo largo del proyecto.

También al equipo de la empresa Tesi Industrial S.L., que me han ayudado siempre que lo he necesitado, principalmente quiero agradecer su apoyo al Albert, Marc, David, Ivan, Ferran y Jordi.

Agradecer a Tesi Industrial S.L. la confianza que han tenido conmigo para realizar un proyecto de estas características.

12. REFERENCIAS BIBLIOGRÁFICAS

[1] WELMEC 7.2. Guía del software Edición 5 [En línea]. Mayo 2011 [Consulta: 5 febrero 2015].

Disponible en:

http://www.welmec.org/fileadmin/user_files/publications/WELMEC_07.02_issue_5_Spain_translation.pdf

[2] Online help programa AED_Panel32 [En línea]. Versión 3.4 [Consulta: 15 febrero 2015].

Disponible en: <http://www.hbm.com/en/menu/products/weighing-components/aed-panel32/>

[3] OIML R51-1. Automatic catchweighing instruments [En línea]. Edición 2006 [Consulta: 20

Abril 2015]. Disponible en: https://www.oiml.org/en/files/pdf_r/r051-1-e06.pdf

[4] OIML D31. Automatic catchweighing instruments [En línea]. Edición 2008 [Consulta: 26

Abril 2015]. Disponible en: http://www.fundmetrology.ru/depository/04_IntDoc_all/D031-e08.pdf

[5] HBM. AED Digital Transducer Electronics [En línea]. Easy, Direct and Efficient [Consulta: 12

febrero 2015]. Disponible en: <http://www.hbm.com/en/menu/products/weighing-indicators/aed/>

[6] HBM DWS2103 [En línea]. Digital Scale display, ensures maximum security [Consulta: 12

febrero 2015]. Disponible en: <http://www.hbm.com/en/menu/products/weighing-indicators/dws2103/>

[7] FP-Sigma | Powerful Compact PLC [En línea]. Panasonic Electric Works Europe AG, 2015

[Consulta: 12 mayo 2015]. Disponible en:

<https://www.panasonic-electric-works.com/eu/3222.htm>

[8] PLC | Programmable Logic Controllers [En línea]. Omron Corporation, 2015 [Consulta: 2

marzo 2015]. Disponible en:

http://industrial.omron.eu/en/products/catalogue/automation_systems/programmable_logic_controllers/modular_plc_series/cj2m/default.html

[9] Swift User's Manual [En línea]. Utilcell [Consulta: 20 mayo 2015]. Disponible en:

http://www.utilcell.com/sites/utilcell.com/files/produkty/modelo_swift_manual_en.pdf

[10] Utilcell: fabricación de células de carga [En línea]. Swift Utilcell [Consulta: 20 mayo 2015].

Disponible en:

http://www.utilcell.es/secciones/productos/ficha.aspx?seccio=productos&vista=0&categoria=2&tipus=&model=modelo_swift

ANEXO A: OIML D31

Este documento internacional especifica los requisitos generales aplicables al software con funcionalidad relacionada en los instrumentos de medición y proporciona orientación para la verificación de la conformidad de un instrumento con estos requisitos.

Como se ha comentado al principio del documento, a continuación se detalla una restricción de la guía OIML D31 [\[4\]](#) que debe ser cumplida y en la guía WELMEC no está considerada.

A.1 Soporte a la protección de la durabilidad

Si el software está implicado en la protección de la durabilidad, se requiere una reacción apropiada. La Recomendación OIML pertinente aconseja que se desactive el instrumento de medida o se genere una alarma en caso de que la durabilidad se vea comprometida.

Para garantizar la durabilidad de la medición, después de un intervalo de tiempo prescrito se ha escogido mostrar una alarma en la pantalla principal de la pantalla táctil indicando que se requiere calibrar el instrumento de medida. El software da una advertencia cuando haya transcurrido el intervalo de mantenimiento. Este intervalo será de un año y medio después de la última calibración.

ANEXO B: SWIFT DE UTILCELL

En este apartado se detallan las principales funcionalidades del Swift fabricado por Utilcell. Se trata de indicador de pesaje y transmisor de alta velocidad. La Figura 14. Swift fabricado por Utilcell muestra el equipo comentado.



Figura 14. Swift fabricado por Utilcell

B.1. Aplicación Check-weigher

La aplicación pesa-animales / check-weigher permite realizar el proceso de pesaje en tres etapas:

- Etapa de espera
- Etapa de captura de peso
- Etapa de muestra del resultado



Figura 15. Proceso de pesaje

Al iniciar la pesada se accede a la etapa de espera que se mantiene durante el intervalo “ t_{del} ” configurado en el dispositivo. Este intervalo permite asegurar que la señal de peso es apta para pesar. Una vez finalizada esta etapa se inicia la etapa de captura, que se mantiene durante el tiempo programado “ t_{ACC} ”, durante el cual el equipo captura una serie de lecturas de peso.

Al finalizar esta etapa se realiza una media de los valores capturados que imprime, envía por el puerto serie y/o totaliza durante el periodo de tiempo programado t_{d15} .


Tal y como se ha observado en la Figura 15. Proceso de pesaje hay varios métodos de inicio de una pesada.

1. Por tecla
2. Por entrada externa
3. Por comando serie
4. Por nivel de peso

Una vez finalizada la pesada se pueden hacer las siguientes opciones:

1. Mostrar por pantalla
2. Enviar un tiquet
3. Acumular a un total
4. Enviar por el puerto serie a un PC

B.1.1 Funcionamiento por tecla

Solo hay que configurar el equipo para que funcione por tecla y al pulsar la tecla  se inicia la pesada.

Se ha configurado el dispositivo para que una vez finalizada la pesada se envíe el valor de la pesada por el puerto serie, y utilizado el método por tecla se ha comprobado que la petición de pesada del bloque de funciones realizado con lenguaje ST funcionara correctamente.

B.1.2 Funcionamiento por entrada externa

El funcionamiento por entrada externa será el mismo que en el funcionamiento por teclado pero utilizando la activación de una entrada digital como inicio de la pesada.

Este método será el que se utilizará en un futuro cuando la empresa tenga disponible el sistema de pesaje en sí, ya que en este proyecto se ha comprobado la correcta comunicación entre el Swift de Utilcell y los PLCs de Panasonic y Omron, es decir, que el bloque de funciones realizado funcione correctamente.

B.1.3 Funcionamiento por comando serie

Este modo se puede utilizar siempre que el equipo no esté configurado para el funcionamiento por nivel de peso.

Se ha comprobado el correcto funcionamiento de este método a partir del bloque de funciones realizado. Se ha preparado para que al pulsar un bit, envíe al equipo un comando

para iniciar la pesada y pasado el tiempo de la duración de la pesada envíe al equipo el comando para leer el peso. El peso obtenido es el correcto.

B.1.3 Funcionamiento por nivel de peso

En este modo de trabajo la orden de inicio de pesada se genera cuando el peso neto supera un valor configurado en el equipo. Una vez terminada la pesada, el peso deberá bajar de un valor programado para poder iniciar una nueva pesada.

Este método no se ha utilizado en este proyecto.

B.2. Comunicaciones

El equipo dispone de 2 puertos serie de comunicaciones:

- Puerto serie RS-232
- Puerto serie RS-485

B.2.1 Protocolo RS-232

El protocolo RS-232 se utiliza para comunicar entre dos equipos, punto a punto, con una distancia máxima de 15m. El formato del protocolo se puede ver en el siguiente cuadro:

Comando	CR
---------	----

Este protocolo será el utilizado para comunicar el Swift de Utilcell con los PLCs Panasonic y Omron, sin modo bus, es decir, solo habrá un Swift que devuelva el valor del peso.

En el bloque de funciones desarrollado se han utilizado los siguientes comandos con el formato de trama comentado:

- P Petición de peso en tiempo real según el Formato F1 detallado a continuación.
- Z Petición de cero. Equivalente a la tecla ZERO del dispositivo.
- CWI Arranque ciclo de pesaje.

Trama respuesta:

CWIA<TER>: ACK: Comando aceptado

CWIN<TER>: NAK: Comando no aceptado

CWR Lectura valor ultima pesada.

Trama respuesta: CWR\$vvvvvvv<TER>

S: Estado del peso leído: 0 -> Vacío, 1 -> Nuevo, 2 -> Leído, 3 -> Error

vvvvvvv: Valor del peso. 7 dígitos incluido el punto decimal. Si no hay punto decimal se completa la trama con un '0'

Definición del Formato F1:

Formato F1:

<STX>	POL	pppppppp	U	G/N	S	T
-------	-----	----------	---	-----	---	---

<STX>: Inicio de trama (Start of Text)

POL: Polaridad, 1 carácter: " " : Peso > 0

"-": Peso < 0

pppppppp: Peso, 7 dígitos

U: Unidades, 1 carácter: "K" kg

"T" t

"G" g

"L" lb

" " sin unidad

G/N: Bruto/Neto: G Bruto

N Neto

S: Status: " " Peso válido

M Peso no estable

O Sobrecarga

I Valor peso no válido

T: Terminación: CR
CR + LF
ACK
NAK

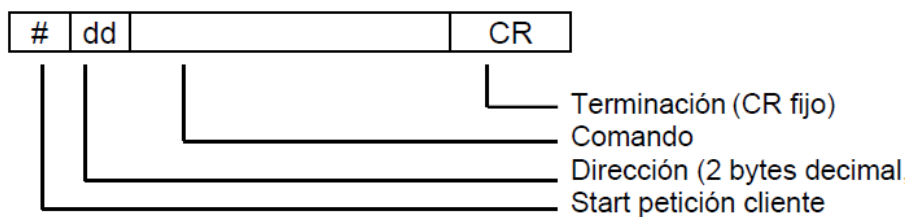
B.2.2 Protocolo propio (RS-485)

Este protocolo permite la comunicación entre varios equipos (máximo 32) en un bus con una distancia máxima de enlace de 1200m.

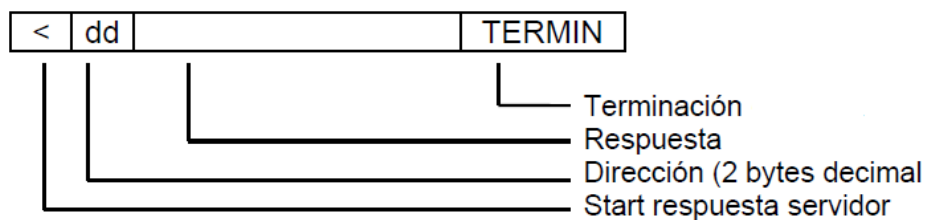
El indicador Swift sólo puede hacer de servidor teniendo como dirección un valor de 1 a 99.

Las tramas de petición del cliente y la respuesta de los servidores tienen los siguientes formatos:

Petición del cliente:



Respuesta del servidor:



La respuesta puede ser de tres tipos:

Datos Se recibió el comando de petición y se responde.
ACK Se recibió el comando y se ha entendido
NAK Se recibió el comando y no se ha entendido



En el bloque de funciones del pesaje también se han utilizado los comandos los comandos detallados en el apartado anterior, con el formato de trama del protocolo propio (RS-485). Es decir, se ha comunicado el Swift en modo bus y sin modo bus.

ANEXO C: SOFTWARE

El software desarrollado a lo largo del proyecto se basa en un terminal AED fabricado por HBM y en un terminal SWIFT fabricado por Utilcell, utilizando PLC's OMRON y Panasonic. La funcionalidad del terminal AED y la descripción del software implementado se han detallado a lo largo de la documentación. En cambio, la funcionalidad del terminal SWIFT se ha detallado en el ANEXO B: SWIFT DE UTILCELL. El software implementado para este Swift tiene la misma estructura que el que se ha desarrollado para el AED, y se le ha agregado la posibilidad de conectar varios SWIFT's de Utilcel con el PLC.

Para desarrollar el software del Arduino se ha programado mediante el uso del lenguaje de programación Arduino, basado en el lenguaje de programación de alto nivel Processing, que es similar a C++. Se ha utilizado el modulo "Timer.cpp" proporcionado en Internet para enviar el comando de leer el software del PLC cada un cierto intervalo de tiempo. Al cargar el programa al Arduino se ejecuta una vez la declaración de variables, los módulos incluidos y el software encontrado dentro de la función setup(). A continuación, se ejecuta repetidamente el código encontrado dentro de la función loop(), donde se comprueba continuamente si se reciben datos por el pin RX del Arduino, que se ha configurado para que sea el 10.

Al recibir respuesta al comando de petición del software del PLC, se hacen una serie de comprobaciones. Primeramente, se busca el inicio de lectura, un número guardado en una variable que se ha introducido al principio del programa a leer. Al encontrarlo, se empieza a calcular la suma de comprobación. Se hace mediante la utilización de las variables "four_characters" y "four_characters2", que como indica el nombre guardan cuatro caracteres recibidos. Un vez llenos los dos buffers, se calcula el valor en decimal del primer buffer que se ha llenado, se suma su valor a la variable "suma" y se vacía este buffer. Este procedimiento se realiza hasta encontrar el final de programa, que será otra variable definida al final del programa a leer. Cuando se encuentra el carácter '&', que indica final de trama de respuesta, hay que borrar los dos últimos caracteres recibidos, que corresponden al checksum de la trama, y ejecutar la función "enviar_siguiete_trama", que envía un comando al PLC para que devuelva la siguiente trama que incluye los valores del programa a leer. Se hace esto debido a que el protocolo realiza una partición de tramas. Al recibir la siguiente trama se busca tres caracteres que identifican la trama y una vez encontrados se continúa con la suma de comprobación. Una vez finalizada la suma de comprobación se debe comparar la variable "suma" con el número adecuado y hacer lo que convenga.

En el documento “Anexo_C” se muestra el software desarrollado para el PLC de Panasonic que comunica con los dos tipos de terminales. Para el PLC de OMRON se ha utilizado el mismo software, adaptándolo a la funcionalidad de este tipo de PLC. Principalmente, ha sido necesaria una modificación de los diversos temporizadores utilizados y una modificación de la forma de comunicación con los dos terminales. Además, se ha agregado el software desarrollado para el Arduino.

Los archivos del software son adjuntados a parte de este documento.

Para abrir los dos archivos desarrollados para el PLC de OMRON hay que tener instalado el software “CX-One V4.30” o una versión más nueva. Una vez instalado hay que abrir el archivo con el programa “CX-Programmer”.

Para abrir el archivo desarrollado para el PLC de Panasonic hay que tener instalado el software “FP WIN Pro 6.0” o una versión más nueva.