

Jokab Safety

Adecuación del sistema PLUTO como PLC de seguridad a los requisitos de la norma ISO13849-1 (Parte de un sistema de mando relativa a la seguridad)

División: **LP – Low Voltage Products**

Estado: **Final**

Departamento: **Soporte Técnico y Soluciones**

Revisión: **v1**

Propósito general de la norma ISO13849-1:

Esta norma de tipo B1 está prevista para guiar a aquellos que estén implicados en el diseño y evaluación de los sistemas de mando, así como a los Comités Técnicos que elaboran las normas de tipo B2 o de tipo C.

Debe proporcionar una base clara que permita evaluar el diseño y las prestaciones de cualquier aplicación SRP/CS (sistema de mando relativo a la seguridad) y de la máquina, por una tercera parte o internamente o por un laboratorio de ensayo independiente, por ejemplo.

Proporcionar los requisitos de seguridad y orientaciones sobre los principios para el diseño e integración de las partes de los sistemas de mando relativas a la seguridad (SRP/CS), incluyendo el diseño del soporte lógico (software).

La norma ISO13849-1 en relación a los sistemas de seguridad programables (por ejemplo, PLCs de seguridad):

La norma **ISO13849-1** denomina “**Soporte lógico de aplicación relativo a la seguridad**” a los sistemas (hardware y software) de control programables relacionados con la seguridad de máquinas (ISO13849-1 / Apartado 4.6 / Anexo J) y diferencia entre los siguientes conceptos:

- **Soporte lógico empotrado relativo a la seguridad (SRESW):** sistema suministrado por el fabricante que no es accesible por el usuario de la máquina (ISO13849-1 / Apartado 3.1.37 / Apartado 4.6.2). Es decir, el software inalterable (firmware) de los sistemas, que es desarrollado por técnicos que diseñan componentes de seguridad (por ejemplo, el software de una barrera inmaterial o el sistema operativo de un PLC de seguridad).
- **Soporte lógico de aplicación relativo a la seguridad (SRASW):** soporte lógico específico para una aplicación, implementado por el fabricante de la máquina y que generalmente contiene secuencias lógicas, límites y expresiones que gobiernan las salidas, las entradas, cálculos apropiados y las decisiones necesarias para cumplir los requisitos de las SRP/CS (ISO13849-1 / Apartado 3.1.36 / Apartado 4.6.3). El ejemplo típico de estos sistemas es la programación que se implementa en un PLC de seguridad.

En la misma norma, el **Anexo J** describe un ejemplo en el que se detallan los “pasos típicos” a seguir a la hora de implementar la aplicación de un “soporte lógico” garantizando un nivel de seguridad PLr=D. Estos pasos incluyen desde el diseño de la estructura lógica, pasando por los aspectos a tener en cuenta relativos al ciclo de vida y su verificación, y finalizando con un ejemplo de normas y recomendaciones a seguir durante la programación.

Con respecto a la programación de un “soporte lógico de aplicación relativo a la seguridad”, precisamente por el hecho de que es un dispositivo destinado a asegurar un funcionamiento óptimo de una máquina manteniendo un nivel de seguridad PLr determinado, se ha de tener muy en cuenta que se han de seguir unas pautas y recomendaciones específicas (diferentes a las reglas que se suelen seguir para la programación de otros sistemas de control “generalistas”). Lo que la norma demanda como requisitos generales es que la aplicación de seguridad sea **legible, comprensible, verificable y mantenible** (ver en éste mismo documento el apartado **Reglas y recomendaciones de programación de PLCs de seguridad Pluto según la norma ISO13849-1**). Por lo tanto, las directrices a seguir son:

- **Seguridad.**
- **Simplicidad**
- **Claridad**
- **Accesibilidad**
- **Documentación**

Otra terminología incluida en la norma ISO13849-1:

Parte de un sistema de mando relativa a la seguridad (SRP/CS): Parte de un sistema de mando que responde a señales de entrada y genera señales de salida relativas a la seguridad. Son las partes de los sistemas de mando que tienen asignado desempeñar funciones de seguridad y pueden estar constituidas de soporte material (hardware) y de soporte lógico (software) y pueden estar separadas del sistema de mando de la máquina o ser una parte integral del mismo.

Sistema electrónico programable (PES): Sistema para mando, protección o control cuyo funcionamiento depende de uno o más dispositivos electrónicos programables, incluyendo todos los elementos del sistema tales como las fuentes de alimentación, los sensores y otros dispositivos de entrada, contactores y otros dispositivos de salida.

Sistema de mando de la máquina: Sistema que responde a señales de entrada de partes de las máquinas, de los operadores, de los órganos de mando externos o de cualquier combinación de estos y que genera señales de salida que dan lugar a que la máquina se comporte de una manera prevista.

Lenguaje de variabilidad limitada (LVL): Tipo de lenguaje que proporciona la posibilidad de combinar funciones de bibliotecas, predefinidas, específicas para una aplicación, con el fin de implementar las especificaciones de los requisitos relativos a la seguridad.

Soporte lógico de aplicación: Soporte lógico específico para una aplicación, implementado por el fabricante de la máquina y que generalmente contiene secuencias lógicas, límites y expresiones que gobiernan las entradas, las salidas, cálculos apropiados y las decisiones necesarias para cumplir los requisitos de las SRP/CS.

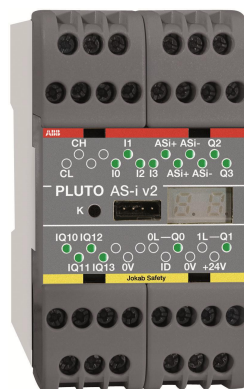
Soporte lógico de aplicación relativo a la seguridad (SRASW): soporte lógico específico para una aplicación, implementado por el fabricante de la máquina y que generalmente contiene secuencias lógicas, límites y expresiones que gobiernan las salidas, las entradas, cálculos apropiados y las decisiones necesarias para cumplir los requisitos de las SRP/CS (ISO13849-1 / Apartado 3.1.36 / Apartado 4.6.3). Escrito en LVL y que cumpla los siguientes requisitos, puede alcanzar un nivel de prestaciones de seguridad PL desde PLa hasta PL_e. El ejemplo típico de estos sistemas es el PLC de seguridad.

Adecuación del PLC de seguridad PLUTO y la herramienta de software PLUTO MANAGER a la normativa de seguridad de máquinas:

Los PLCs de seguridad de la gama **PLUTO** están diseñados para cumplir con las exigencias de la Directiva de Máquinas de la UE (2006 / 42 / CE) en relación con la seguridad de los sistemas de control, independientemente de si el sistema se puede utilizar en otras aplicaciones como la industria de transformación, los hornos, los ferrocarriles, etc., que tienen similares requisitos.

Los PLCs de seguridad **PLUTO** están diseñado de acuerdo a las siguientes normas de seguridad funcional para sistemas de control:

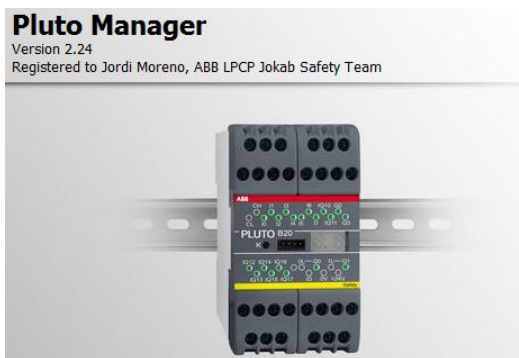
- EN 954-1, Categoría 4
- EN 62061, SIL3
- EN 13849-1, Categoría 4 y nivel de seguridad Performance Level e (PL_e)
- IEC 61508-, SIL 3
- IEC-EN 61511-, SIL 3
- EN 50156-1



**PL_e, Categoría 4
SIL 3**

Para que una aplicación de seguridad cumpla los estándares anteriormente indicados, es necesario que el diseño y la instalación de todo el sistema de control de la seguridad relacionado (incluyendo sensores y actuadores, no sólo lo relativo al PLC **PLUTO**) cumpla dichos requisitos.

PLUTO MANAGER es la herramienta de software que permite configurar y programar los PLCs de seguridad de la gama **PLUTO** según los requerimientos indicados en la normativa de seguridad de máquinas utilizando lenguajes gráficos especificados en la normativa estándar IEC 61131-3.



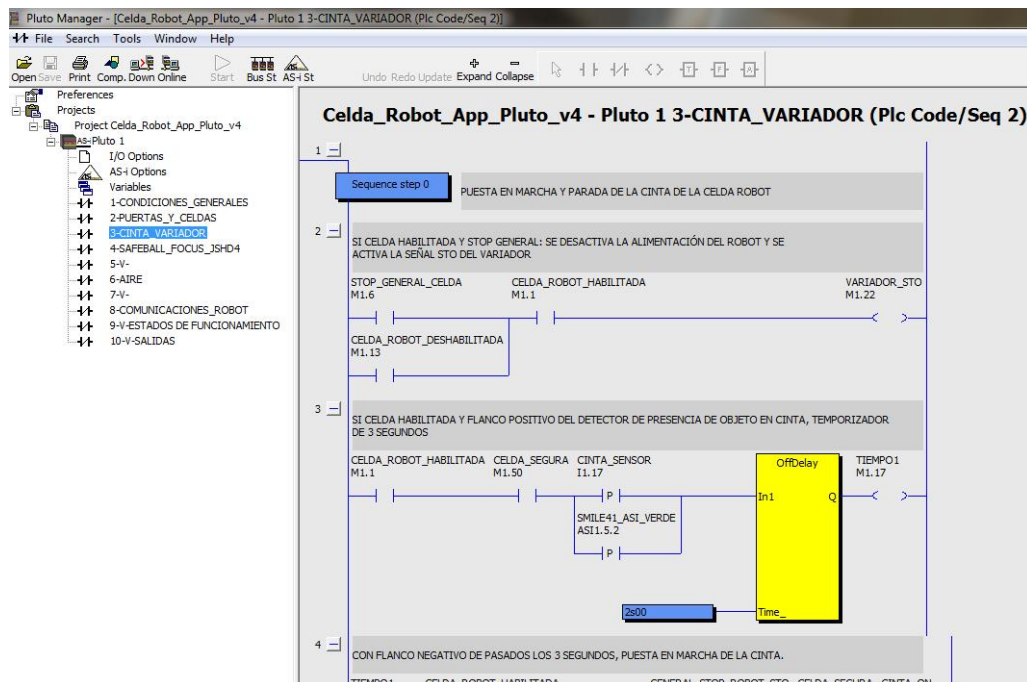
Reglas y recomendaciones de programación de PLCs de seguridad Pluto según la norma ISO13849-1:

La programación de un PLC de seguridad debe asegurar que la aplicación a controlar mantiene el nivel de seguridad PL_r adecuado en cada momento del su ciclo de vida, por lo que se debe segmentar de manera que muestre una estructura general coherente y comprensible, que permita localizar fácilmente los diferentes procesos de tratamiento. Esto implica seguir las siguientes reglas y recomendaciones:

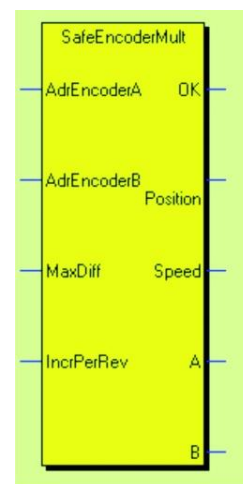
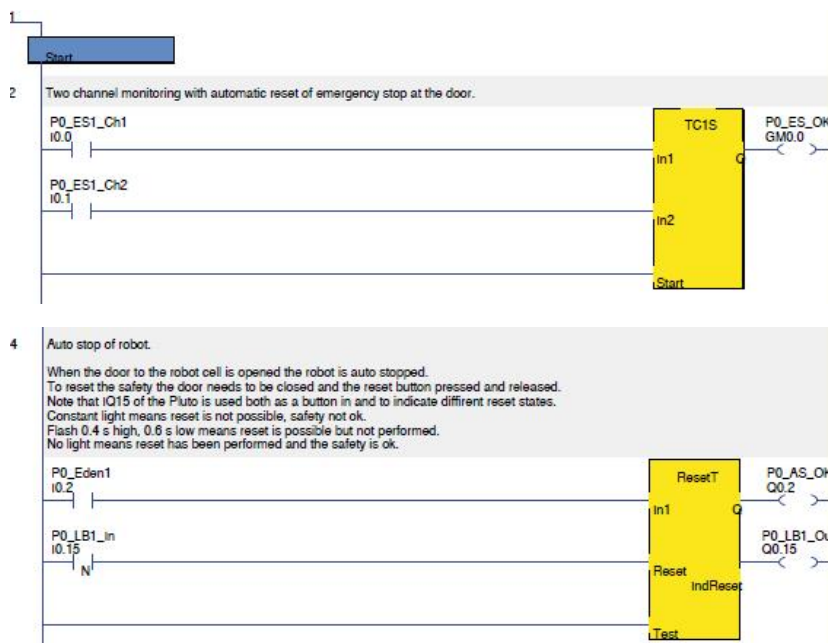
- 1- La utilización programas tipo (plantillas) o para bloques funcionales (ISO13849-1 / Apartado 4.6.3 – B y C).
 - a. Siempre que sea posible se deben utilizar bibliotecas de bloques funcionales validados, ya sean procedentes de bibliotecas relativos a la seguridad o bibliotecas validadas específicamente para una aplicación concreta.

- b. Están sumamente recomendados los lenguajes gráficos (por ejemplo, diagrama bloque de una función, diagrama jerarquizado).

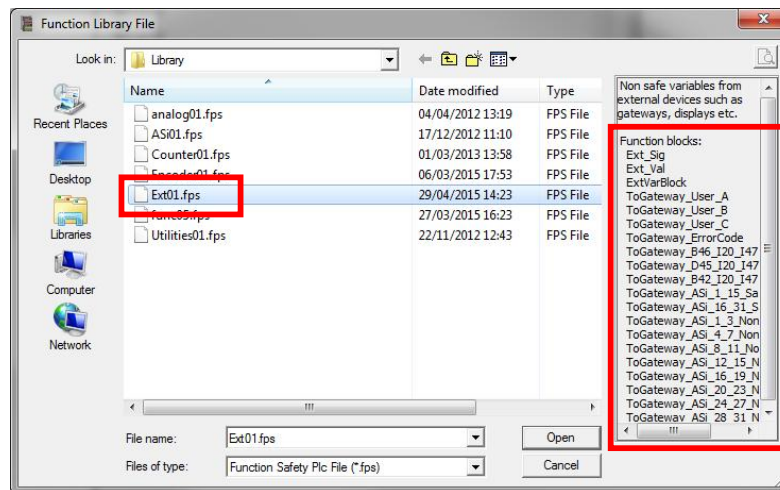
La herramienta **PLUTO MANAGER** de configuración y programación de PLCs de seguridad **PLUTO**, permite la programación de aplicaciones en lenguaje LADDER o LD (lenguaje de contactos y bloques de función) totalmente gráfico. La implementación resulta sencilla de programar, de modificar y de mantener.



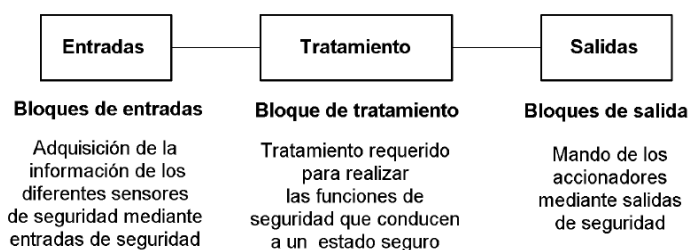
PLUTO MANAGER dispone de una librería de bloques de función específicos y pre-programados para componentes de seguridad aprobados por la entidad certificadora TÜV Rheinland. Dichos bloques están destinados al control específico tanto de dispositivos como de funciones de seguridad determinadas y aseguran el correcto tratamiento de las mismas en función de los requisitos de la normativa de seguridad de máquinas.



PLUTO MANAGER incorpora librerías de bloques de función específicos para el tratamiento de señales analógicas, tratamiento de señales procedentes de *encoders* de seguridad Jokab, tratamiento de funciones de seguridad específicas (por ejemplo, funciones de inhibición o *muting*, gestión de mandos bimanuales Jokab, gestión de barreras de seguridad, etc.), implementación de comunicaciones con dispositivos externos y con diversos protocolos (mediante pasarelas Jokab Gateway), y funciones varias (contadores, temporizadores, básculas, etc.).



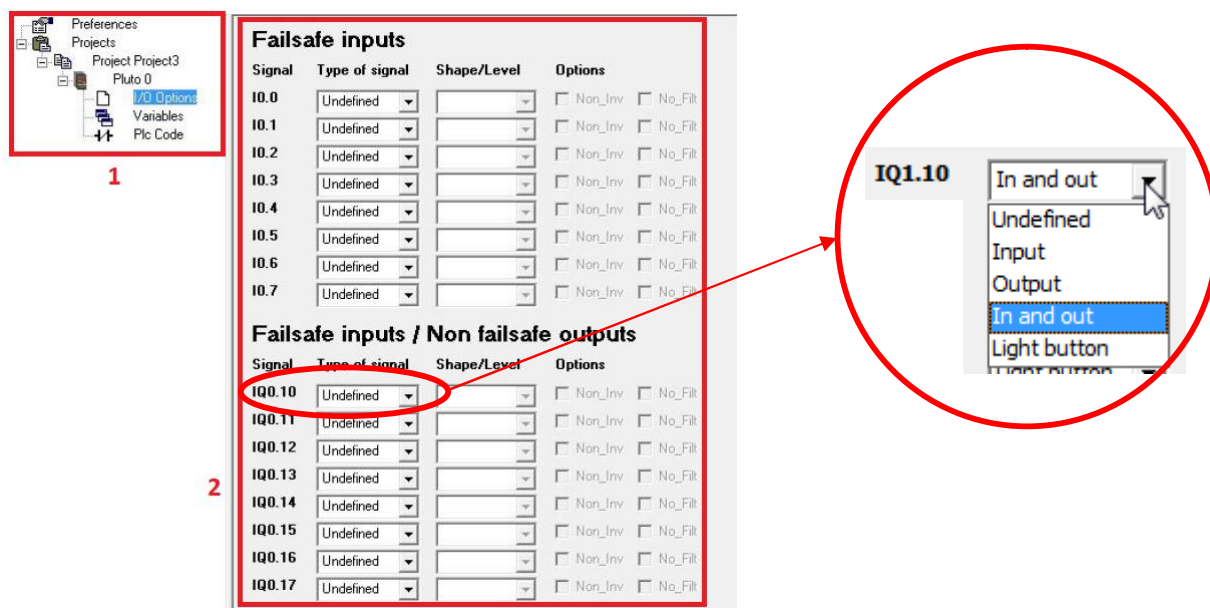
- 2- El diseño de la aplicación lógica ha de contemplar la partición del programa en módulos o segmentos con el fin de identificar las partes principales correspondientes a las “entradas”, “procesos de tratamiento” y “salidas” (ISO13849-1 / Apartado 4.6.3 – C / Anexo J).



Las aplicaciones creadas con el software de programación **PLUTO MANAGER** requieren un orden y organización jerárquica en forma de árbol de los distintos proyectos y dispositivos a programar, totalmente configurable a nivel de usuario.

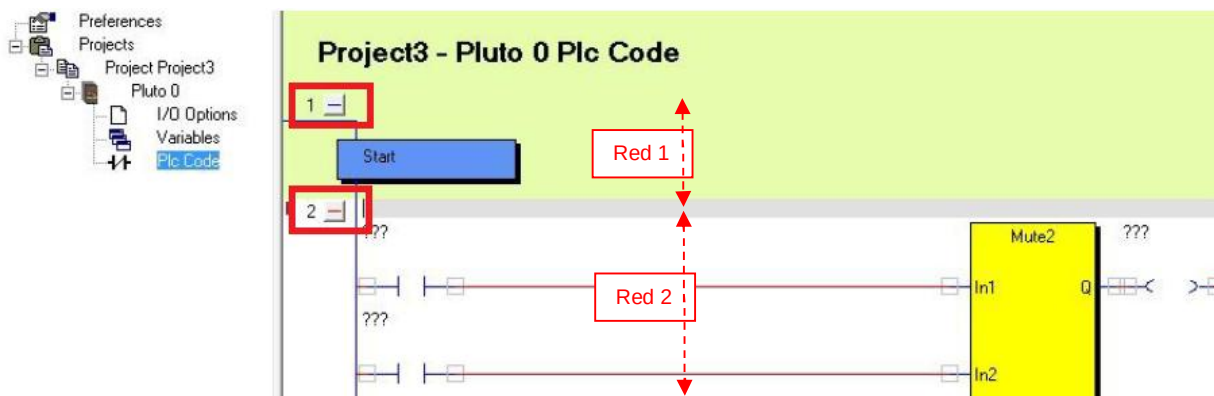
Una vez definido el dispositivo a programar (p.ej. “PLUTO 0”), este clasifica su configuración en tres apartados o módulos fundamentales (ver delimitación “1” de la imagen inferior): **Opciones I/O** (entradas y salidas), **Variables** y **Código PLC**. Esta organización no es modificable.

La parametrización de las entradas y salidas quedarán recogidas en el módulo **I/O Options**, donde podrán ser configuradas (delimitación “2” de la imagen inferior).



El módulo **Variables** recoge la definición de todo el conjunto de variables que se utilizan en la programación del PLC de seguridad (ver punto 4 de éste mismo documento).

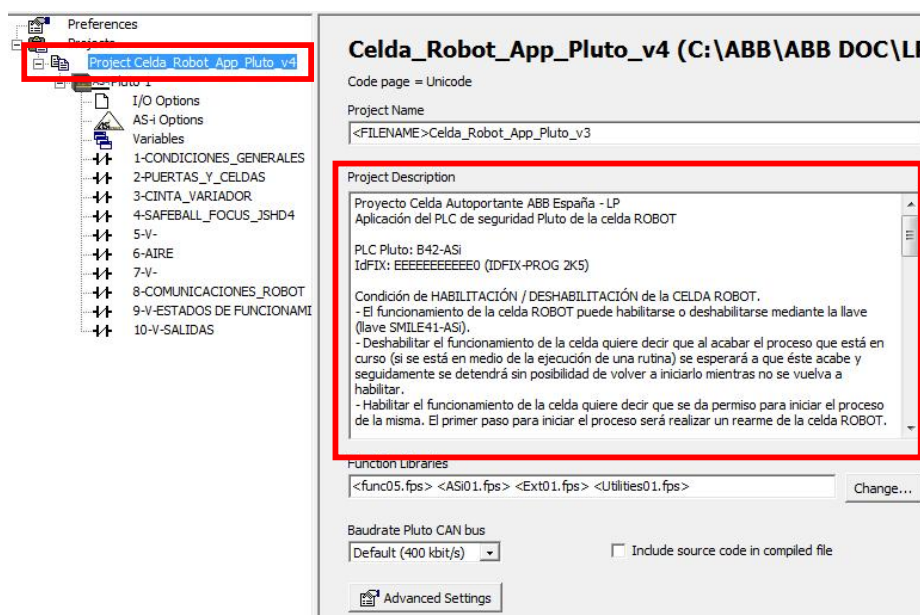
El módulo **PLC Code** contiene el código de la programación del PLC en lenguaje LADDER o LD (lenguaje de contactos y bloques de función). Dicho código se estructura en redes, y se debe crear una red por cada salida que requiera el programa. Esto permite un mayor orden y gestión del "Bloque de Tratamiento" (tratamiento requerido para realizar las funciones de seguridad que conducen a un estado seguro), establecido por la ISO13849-1 / Apartado 4.6.3 – C.



3- La documentación debe ser completa, manejable, explícita y comprensible:

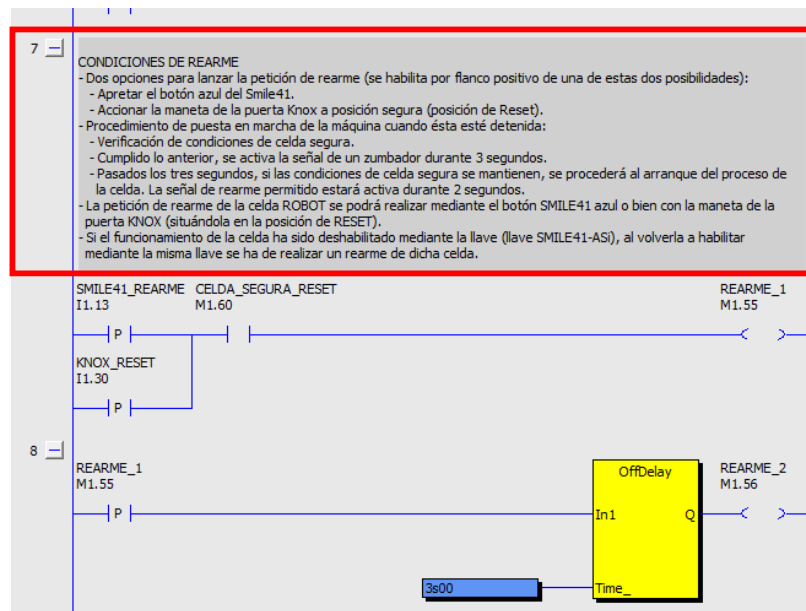
- a. Deben estar documentados todos los apartados, módulos y funciones (código fuente), así como todo el ciclo de vida y las actividades de modificación de la aplicación (ISO13849-1 / Apartado 4.6.3 – A, E y G).

Durante la configuración del proyecto **PLUTO MANAGER** habilita espacios dedicados a la documentación del mismo. En dichos apartados, a rellenar en formato texto, el usuario debe escribir los comentarios, las notas y otros aspectos relevantes relativos a la programación. Este espacio permanece disponible durante todo el ciclo de vida del programa y cada modificación, realizada en el programa, debe verse reflejada a modo de comentario en dichos espacios dedicados. Es decir, es necesario documentar cada actualización del código con datos relevantes, fechas de actualización, modificaciones realizadas, etc.

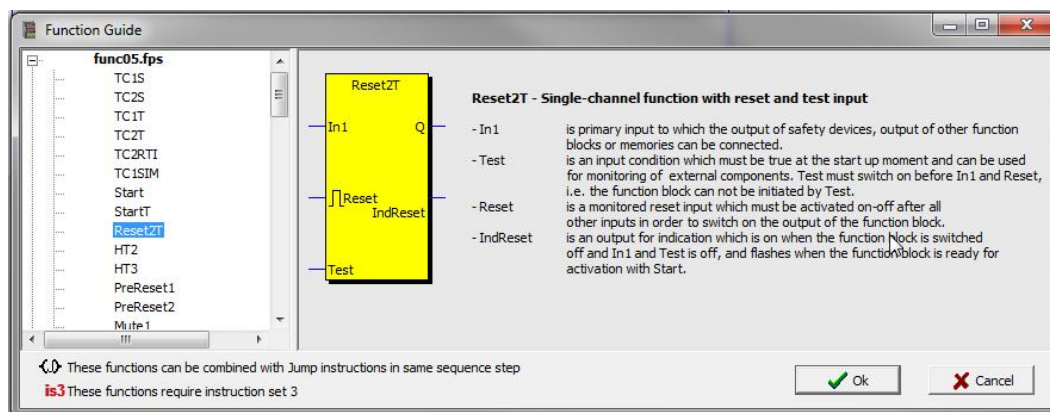


- b. Los comentarios en cada sección del programa en el código fuente para facilitar la puesta al día de los comentarios en caso de modificación en el caso de los bloques funcionales. La descripción del papel que tiene dicho bloque funcional ha de estar accesible cuando se hace una llamada a dicho bloque (ISO13849-1 / Apartado 4.6.3 – A, E y G).

PLUTO MANAGER permite la edición de comentarios en cada red, lo que permite documentar e identificar cada una de las diferentes partes que componen la estructura del código fuente.



Del mismo modo, todos los bloques funcionales integrados en **PLUTO MANAGER** disponen de una breve descripción de su funcionamiento.



- La ubicación de memoria se debería utilizar solamente por un único tipo de datos, identificándolos por una única etiqueta (utilización de variables). El código fuente ha de ser legible, comprensible y verificable por lo que cada variable global, entrada o salida, debería tener un nombre nemotécnico suficientemente explícito y estar descrita mediante un comentario en el programa fuente (ISO13849-1 / Apartado 4.6.3 - e).

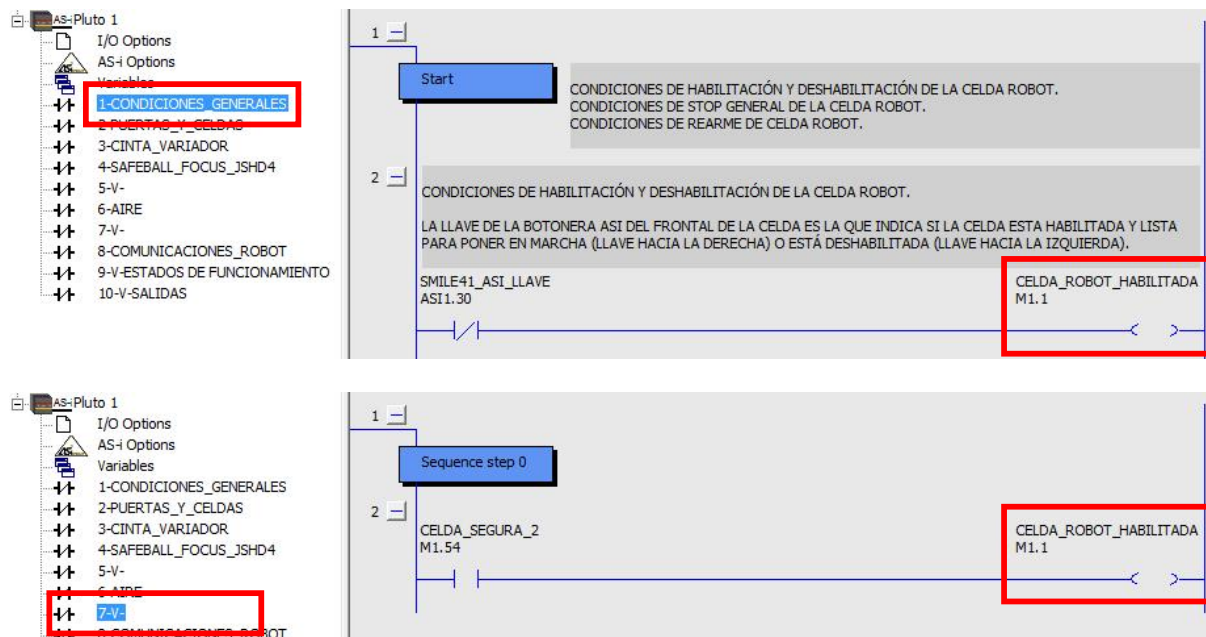
PLUTO MANAGER permite la edición, configuración e implementación de distintos tipos de variables. Cada variable se crea asignando un nombre y un comentario o descripción a una señal física del PLC (entrada o salida) o a un registro interno de memoria. Para facilitar su uso en la aplicación a programar, las variables se clasifican según la señal a la que están asignadas: **Safety inputs** (entradas de seguridad), **Safety outputs** (salidas de seguridad), **Global Memories** (registros digitales internos), etc.

Variable attributes: [G] Global variable. Variable is visible to other Plutos and gateways on the bus. [E] Exported variable. Variable is visible to other Plutos on the bus via extra telegrams.			
[Safety Inputs] Nonsafety Inputs Safety Outputs Nonsafety Outputs Global Memories Memories Registers Double Registers			
Status	Variable	Symbolic Name	Description
	I1.0	[G] SMILE41_INFO	SETA ACTIVA = SEÑAL OFF / SETA INACTIVA = SEÑAL ON
	I1.1	[G] KNOX_BLOQUEADA	ON = KNOX CERRADA EN POSICION SEGURA (BLOQUEADA)
	I1.2	[G] EDEN_ROBOT	ON = EL ROBOT ESTÁ TRABAJANDO EN UNA POSICION SEGURA
	I1.3	[G] EDEN_UNION_CINTAS_1	(EDEN PUERTA CORREDERA) OR (EDEN ENTRE CELDAS)

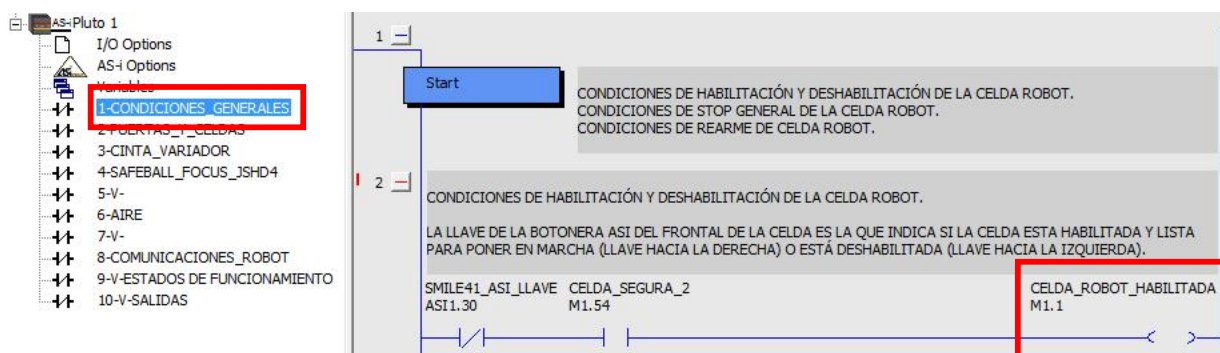
A cada elemento utilizado en la programación de la aplicación se le ha de asignar una variable; la información configurada dicha variable aparecerá al pasar el ratón por encima del elemento al que ha sido asignado.

- 5- Para lograr la coherencia del valor calculado o forzado de señales, el programa debería estar estructurado de tal manera que las ecuaciones para la puesta al día de una variable estén centralizadas y la activación o desactivación de cualquier salida se debería realizar una sola vez (condiciones centralizadas) (ISO13849-1 / Apartado 4.6.3 – C).

PLUTO MANAGER permite al usuario actualizar variables de salida en varios puntos de la aplicación (siempre que sea en en varias redes, ya que cada red sólo admite configurar una sola variable de salida).



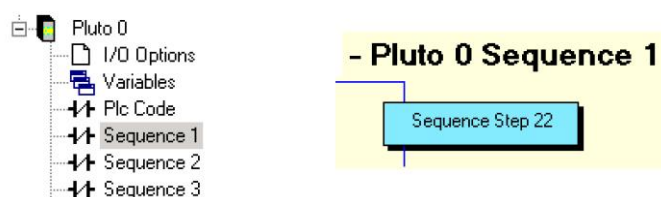
No obstante, y siguiendo la indicación de la norma ISO13849-1 / Apartado 4.6.3 – C como una buena práctica de programación de PLCs **PLUTO**, se recomienda que cada variable de salida necesaria en la aplicación se ejecute únicamente en un solo punto de la programación.



- 6- La secuencia de trabajo no debería depender de opciones tales como direcciones de saltos calculados durante la ejecución del programa. Se autorizan los saltos condicionales (ISO13849-1 / Anexo J).

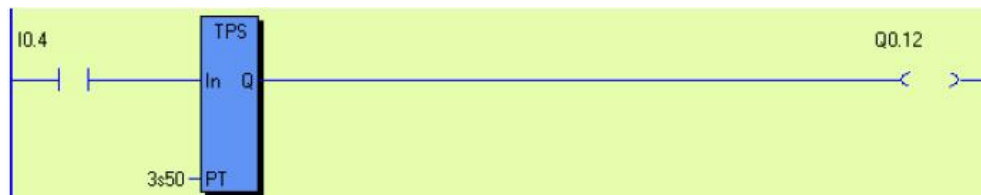
La programación de saltos condicionales está permitida en **PLUTO MANAGER**, pero no se autoriza su uso si puede poner en peligro el **estado seguro** del sistema. Se trata de partes de programa secuenciales, que se realizan paralelamente al programa principal y que deben ser configuradas por el usuario.

Sequence programming in Pluto Manager:



- 7- La herramienta ha de permitir la creación de bloques funcionales de usuario con las siguientes especificaciones (ISO13849-1 / Anexo J):

Fuera de los bloques funcionales predefinidos de los que dispone **PLUTO MANAGER** (explicados en el punto 1 de este mismo documento), existe la posibilidad de crear y aplicar bloques funcionales diseñados por el usuario. Son los llamados **bloques “azules”**.



Dichos bloques “azules” configurados por el usuario, deben cumplir con las especificaciones siguientes, según la normativa comentada anteriormente:

- Parámetros: máximo ocho digitales y dos entradas número entero, una salida;
- Código fuente de la función: máximo diez variables locales, máximo 20 ecuaciones booleanas.
- Los bloques funcionales no deberían modificar las variables globales.
- Se debería controlar un valor digital con respecto a las cotas prefijadas con el fin de garantizar su dominio de validez.
- Un bloque funcional debería tratar de detectar las incoherencias de las variables a tratar.
- El código de defectos de un bloque debería ser accesible para discriminar un defecto de los demás.
- Se deberían describir los códigos de defectos y el estado del bloque después de la detección de un defecto se deberían describir mediante comentarios.
- Se deberían describir el rearme del bloque o el restablecimiento de un estado normal mediante comentarios.

Links:

Link al documento “Seguridad en sistemas de control según la norma EN ISO 13849-1 (ABB Jokab Safety)”:

<https://library.e.abb.com/public/fabc9b502be2b522c125796c003a0f1b/Seguridad%20y%20sistemas%20de%20control%20Jokab.pdf>

Link oficial para la descarga de la norma ISO13849-1:

<http://www.aenor.es/aenor/normas/normas/fichanorma.asp?tipo=N&codigo=N0042221&pdf=#.VjCElCuI5UU>