



Universidad de Oviedo

Memoria del Trabajo Fin de Máster realizado por

MARCOS RODRÍGUEZ CASTAÑO

para la obtención del título de

Máster en Ingeniería de Automatización e Informática Industrial

Mejora y optimización de la pasarela DALI GW611100 de BES  
KNX (Ingenium)

JULIO 2020

# Resumen

---

En los años 90, se desarrolló un protocolo de comunicaciones orientado a la domótica e inmótica denominado KNX. En paralelo, se creó otro destinado al control digital de la iluminación, el protocolo DALI. En una instalación en la que se desee controlar balastos DALI con un dispositivo conectado a un bus KNX, sería necesario el uso de una pasarela capaz de traducir de un protocolo a otro. La empresa asturiana Ingenium se ha querido diferenciar, incluyendo en su pasarela una pequeña pantalla, con la que un instalador podrá configurar una vivienda sin necesidad de equipos adicionales. En este trabajo fin de máster, se describirá la actualización de este dispositivo, concretamente del algoritmo de programación instalado en la pantalla bajo el protocolo DALI. Además, se ha creado una herramienta de configuración de dispositivos KNX con ETS 5, sustituyendo a un programa propietario incluyéndose en el catálogo de productos de Ingenium. La ventaja de esta actualización radica en la propia filosofía del protocolo, cualquier cliente podría configurar el dispositivo utilizando la herramienta que ofrece la asociación KNX.

# Definiciones

---

**Inmótica:** disciplina que trata la automatización inteligente de edificios no destinados a vivienda.

**Bus (de datos):** medio físico que permite la comunicación entre los dispositivos que estén conectados.

**Telegrama:** mensaje que se envía a través del bus de datos. Está formado por un conjunto de bytes con una estructura definida, en el que cada byte tiene significado por sí mismo.

**Microprocesador:** circuito integrado programable que consta de unidad lógica aritmética, unidad de control y matriz de registro.

**Microcontrolador:** circuito integrado programable en el que se incluye un microprocesador con memoria ROM, RAM y pines de entrada y salida de propósito general (GPIO).

**Plugin:** programa que añade funcionalidades adicionales a otro, complementándolo.

**Escena:** zona de memoria de un balastro DALI, que almacena un byte con un valor de iluminación. Cada balastro DALI posee 16 de éstas.

**Parámetro:** elemento de configuración que definirá cómo funcionará un dispositivo concreto.

**Objeto de comunicación:** elemento que se usa para relacionar entre sí funciones entre múltiples dispositivos. Se traducen como acciones directas en la instalación, es decir, qué pueden hacer otros dispositivos sobre el que se está configurando y qué puede hacer éste sobre otros.

**Flag:** bit utilizado para representar atributos dentro de una estructura de datos.

**EEPROM:** memoria no volátil reprogramable eléctricamente.

# Índice

---

<b>RESUMEN .....</b>	<b>I</b>
<b>DEFINICIONES.....</b>	<b>I</b>
<b>1 INTRODUCCIÓN.....</b>	<b>1</b>
1.1 ANTECEDENTES.....	1
1.1.1 <i>Estado del arte</i> .....	1
1.1.2 <i>Ingenium</i> .....	5
1.2 OBJETIVOS.....	5
1.3 ALCANCE.....	5
1.4 DESCRIPCIÓN DEL BANCO DE PRUEBAS.....	6
<b>2 SOFTWARE DE PROGRAMACIÓN ETS 5.....</b>	<b>10</b>
2.1 PROGRAMACIÓN EN LÍNEA.....	14
2.2 PROGRAMACIÓN FUERA DE LÍNEA.....	15
<b>3 MANUFACTURER TOOL .....</b>	<b>16</b>
<b>4 PROGRAMACIÓN DE LA PANTALLA TFT .....</b>	<b>22</b>
4.1 ALGORITMO.....	22
4.2 VARIABLES Y FUNCIONES UTILIZADAS.....	24
4.2.1 <i>Variables globales</i> .....	24
4.2.2 <i>Funciones</i> .....	28
4.2.3 <i>Funciones de tratamiento de interrupción</i> .....	40
4.2.4 <i>Programa principal</i> .....	43
<b>5 CONCLUSIONES .....</b>	<b>45</b>
5.1 DESARROLLO DE LA PANTALLA.....	45
5.2 DESARROLLO DEL PRODUCTO DEL CATÁLOGO DE ETS 5.....	46
5.3 SUGERENCIAS DE MEJORA.....	47
<b>6 REFERENCIAS.....</b>	<b>48</b>

## Índice de figuras

---

Figura 1. Topologías del protocolo DALI.....	4
Figura 2. Catálogo de fabricantes y dispositivos del ETS 5.....	10
Figura 3. Vista de topología del proyecto de ETS 5.....	11
Figura 4. Despliegue de vista de edificio del proyecto en ETS 5. ....	11
Figura 5. Lista de los parámetros del dispositivo.....	12
Figura 6. Vista de parámetros de un dispositivo de configuración mediante plugin.....	12
Figura 7. Lista de objetos de comunicación de un actuador. ....	13
Figura 8. Aspecto del plugin de parametrización de la pasarela.....	14
Figura 9. Detalle del plugin donde se crean nodos y grupos de forma manual. ....	15
Figura 10. Despliegue del árbol de parámetros estáticos de Manufacturer Tools. ....	16
Figura 11. Detalle de operación para añadir tipos de parámetros nuevos. ....	16
Figura 12. Lista de tipos de parámetros añadidos. ....	17
Figura 13. Lista de valores que puede adquirir “Número de grupos” (Tipo restricción).....	17
Figura 14. Arriba, vista del parámetro de memoria “Power on value B0”. Abajo, vista del parámetro virtual “Nodos”. .....	18
Figura 15. Vista de dos parámetros de memoria en el apartado de “ParameterRefs”. ....	18
Figura 16. Detalle creación y configuración de objetos de comunicación en el apartado “ComObjects”.....	18
Figura 17. Vista de tres objetos de comunicación en el apartado de “ComObject”.....	19
Figura 18. Vista de tres objetos de comunicación en el apartado “ComObjectRef”.....	19
Figura 19. Árbol de objetos dinámicos.....	19
Figura 20. Izquierda, detalle de objetos del nodo 0. Derecha, detalle de objetos del grupo 0.....	20
Figura 21. Aspecto de los parámetros generales del grupo de nodos de la pasarela en ETS 5. ....	20
Figura 22. Izquierda, detalle de los parámetros del nodo 0. Derecha, lista de grupos a los que se puede asignar el nodo 0.....	21
Figura 23. Vista de las escenas del nodo 0. ....	21
Figura 24. Lista de objetos de comunicación activos.....	21

## Índice de tablas

---

Tabla 1. Características de los medios físicos de comunicación del protocolo KNX. ....	2
Tabla 2. Instalación de cable rígido - Clase 1. ....	4
Tabla 3. Instalación de cable trenzado - Clase 2. ....	4
Tabla 4. Descripción de las tramas programadas – Parte I. ....	7
Tabla 5. Descripción de las tramas programadas – Parte II. ....	8
Tabla 6. Lista de variables globales – Parte I. ....	24
Tabla 7. Lista de variables globales – Parte II. ....	25
Tabla 8. Lista de variables globales – Parte III. ....	26
Tabla 9. Lista de variables globales – Parte IV. ....	27
Tabla 10. Lista de variables globales – Parte V. ....	28
Tabla 11. Variables de entrada y salida de la función “configuracion_pic”. ....	28
Tabla 12. Variables de entrada y salida de la función “LCD_command”. ....	29
Tabla 13. Variables de entrada y salida de la función “LCD_Data”. ....	29
Tabla 14. Variables de entrada y salida de la función “LCD_SetPos”. ....	29
Tabla 15. Variables de entrada y salida de la función “LCD_PutChar”. ....	30
Tabla 16. Variables locales de la función “LCD_PutChar”. ....	30
Tabla 17. Variables de entrada y salida de la función “LCD_PutSim”. ....	31
Tabla 18. Variables locales de la función “LCD_PutSim”. ....	32
Tabla 19. Variables de entrada y salida de la función “LCD_PutChar56x80”. ....	32
Tabla 20. Variables locales de la función “LCD_PutChar56x80”. ....	33
Tabla 21. Variables de entrada y salida de la función “Clear_LCD”. ....	33
Tabla 22. Variables locales de la función “Clear_LCD”. ....	33
Tabla 23. Variables de entrada y salida de la función “Init_LCD”. ....	34
Tabla 24. Variables de entrada y salida de la función “pintaMenu”. ....	34
Tabla 25. Variables de entrada y salida de la función “pintaRotuloNodos”. ....	35
Tabla 26. Variables de entrada y salida de la función “inicbuff”. ....	35
Tabla 27. Variables de entrada y salida de la función “pintaNodosActivos”. ....	35
Tabla 28. Variables de entrada y salida de la función “pintaEspera”. ....	36
Tabla 29. Variables de entrada y salida de la función “pintaGrupos”. ....	36
Tabla 30. Variables de entrada y salida de la función “pintaEscenas”. ....	36
Tabla 31. Variables de entrada y salida de la función “pintaRetorno”. ....	37
Tabla 32. Variables de entrada y salida de la función “pintaNivel”. ....	37
Tabla 33. Variables de entrada y salida de la función “pintaNumeros”. ....	37
Tabla 34. Variables locales de la función “pintaNumeros”. ....	38
Tabla 35. Variables de entrada y salida de la función “StringToChar”. ....	38
Tabla 36. Variables locales de la función “StringToChar”. ....	38
Tabla 37. Variables de entrada y salida de la función “Linea_LCD”. ....	39
Tabla 38. Variables locales de la función “Linea_LCD”. ....	39
Tabla 39. Variables de entrada y salida de la función “cadenas”. ....	39
Tabla 40. Variables locales de la función “cadenas”. ....	39
Tabla 41. Variables de entrada y salida del tratamiento de interrupción del timer 0. ....	40
Tabla 42. Variables de entrada y salida del tratamiento de interrupción por cambio externo 0. ....	40
Tabla 43. Variables locales del tratamiento de interrupción por cambio externo 0. ....	40
Tabla 44. Variables de entrada y salida del tratamiento de interrupción por cambio externo 1. ....	41
Tabla 45. Variables local del tratamiento de interrupción por cambio externo 1. ....	41
Tabla 46. Variables de entrada y salida del tratamiento de interrupción por cambio externo 2. ....	41

<b>Tabla 47. Variables de entrada y salida del tratamiento de interrupción por cambio en el puerto B. ....</b>	<b>41</b>
<b>Tabla 48. Variables locales del tratamiento de interrupción por cambio en el puerto B. ....</b>	<b>42</b>
<b>Tabla 49. Variables de entrada y salida del tratamiento de interrupción por dato en el puerto de comunicaciones....</b>	<b>43</b>
<b>Tabla 50. Variables locales del tratamiento de interrupción por dato en el puerto de comunicaciones.....</b>	<b>43</b>
<b>Tabla 51. Variables de entrada y salida del programa principal. ....</b>	<b>43</b>
<b>Tabla 52. Variables locales del programa principal. ....</b>	<b>43</b>

# 1 Introducción

---

## 1.1 Antecedentes

---

La domótica se define como *“los sistemas capaces de automatizar una vivienda o edificación de cualquier tipo, aportando servicios de gestión energética, seguridad, bienestar y comunicación. Pueden estar integrados por medio de redes interiores y exteriores de comunicación, cableadas o inalámbricas, y cuyo control goza de cierta ubicuidad, desde dentro y fuera del hogar”*.

### 1.1.1 Estado del arte

---

El inicio de la domótica se encuentra a finales de los años 80 con el protocolo X10 creado por Pico Electronics, basado en la transmisión de datos por corrientes portadoras, es decir, a través de la línea de potencia del edificio. Este sistema soporta hasta 256 dispositivos en un área de hasta 250 m<sup>2</sup>, la sincronización de datos se realiza en los pasos por cero de la red, por lo que la velocidad de transmisión depende de la frecuencia de red (50 Hz o 60 Hz según el país). El principal inconveniente es que es muy sensible a las perturbaciones de la línea.

Posteriormente, empresas del sector eléctrico europeo se sumaron a la domótica adaptando sus productos destinados a la industria hacia la vivienda, se crearon los protocolos EIB, conocido como Instabus, desarrollado por empresas alemanas, BatiBUS por la empresa francesa Merlin y EHS desarrollado por la asociación europea de sistemas de hogar (EHSA por sus siglas en inglés), con el tiempo se formó la asociación KNX, pasando a extenderse por múltiples países de América, Europa y Asia. Se lanzó al mercado una herramienta de programación, independiente de los fabricantes, de instalaciones domésticas, *ETS (Engineering Tool Software)* en 1993, actualmente se encuentra en la versión *ETS 5*. Se estableció como el estándar ISO/IEC 14543-3.

Todos los dispositivos KNX usan el mismo medio de comunicación e intercambian información a través del bus común. Su topología es descentralizada, no requiere de ninguna unidad central ya que cada dispositivo dispone de su propio microprocesador. Su principal ventaja es que, si un dispositivo falla, el resto de la instalación sigue funcionando. Además de los dispositivos de sistema como pueden ser la fuente de alimentación, la interfaz de programación, los acopladores de línea se distinguen en KNX dos tipos de dispositivos: sensores y actuadores.

Los sensores son elementos que detectan acciones sobre la instalación, ya sea pulsación de una tecla, movimiento, cambios de temperatura, etc., y las convierten en telegramas que se envían al bus. Aquellos elementos que reciben los telegramas y convierten las órdenes ahí contenidas en acciones se denominan actuadores. Los sensores representan los emisores de órdenes, mientras que los actuadores son los receptores y ejecutores de dichas órdenes.

Este protocolo admite hasta 255 dispositivos por línea, pero con el uso de acopladores, potencialmente tiene una capacidad de más de 50000 dispositivos, puesto que de cada acoplador se pueden colgar 255 dispositivos con su respectiva fuente de alimentación, respetando una topología determinada por el procedimiento de comunicación utilizado.

El sistema KNX, es capaz de utilizar múltiples medios y procedimientos de comunicación para el intercambio de tramas entre los dispositivos del bus, lo que define sus parámetros de transmisión de datos:

- Transmisión a través de un par de hilos trenzados (bus): KNX Twisted Pair (KNX TP).
- Transmisión a través de la línea de fuerza 230 V existente: KNX Powerline (KNX PL).
- Transmisión inalámbrica: KNX Radio Frequency (KNX RF).
- Transmisión mediante mensajes IP: (KNX IP).

	Velocidad de transmisión	Procedimiento de acceso al bus	Tensión nominal del bus	Estructura del telegrama
KNX TP	9600 baudios	CSMA/CA	24 V CC	Prioridad: 8 bits Dirección: 5 bytes Datos: 1 - 16 bytes Paridad: 8 bits
KNX PL	1200 baudios	Bytes de preámbulo	Red: 230 V CA	Sincronización: 4 bits Preámbulo: 2 bytes Datos: 9 bits – 32 bytes ID: 8 bits
KNX RF	8192 – 16348 kbps	Comprobación de ACK	No hay medio físico	Sincronización Bloque de datos I: 10 bytes Checksum: 2 bytes Bloque de datos II: 16 bytes Checksum: 2 bytes Sincronización
KNX IP	20 ms por telegrama	TCP/IP UDP/IP	IEEE 802.3	Longitud cabecera Versión del protocolo ID del tipo de servicio Longitud total Cuerpo KNXnet/IP

*Tabla 1. Características de los medios físicos de comunicación del protocolo KNX.*

En 1999 se desarrolló otro protocolo, LonTalk, creado por la empresa Echelon Corporation en Estados Unidos, sus campos de aplicación son la automatización de edificios, alumbrado público, transporte, industria y gasolineras, inicialmente bajo el estándar ANSI/CEA-709.1-B, actualmente se encuentra recogido en la norma ISO/IEC-14908.

En esta época aparecieron a su vez múltiples protocolos propietarios, Modbus, Bus-CAN, BUSing, etc., dedicados a la domótica y sus marcas asociadas Ingenium, Teletask, Deltadore, etc., y otras basadas en IPv4 como IPdomo.

En 2006 se empezaría a utilizar la tecnología inalámbrica para el control de viviendas con protocolos como Zigbee y Zwave.

En la actualidad, KNX y LON dominan el mercado en Europa y Norte América respectivamente.

DALI es un estándar internacional (IEC 62386) en los sistemas de control de iluminación que proporciona un único interfaz de control electrónico para fuentes de luz y dispositivos como controladores de iluminación. Este estándar permite conectar una amplia variedad de dispositivos como: balastos regulables,

transformadores, módulos de relés, accesorios de emergencia y controladores de diferentes fabricantes, combinados en un solo sistema de control. Se consigue a través de un bus formado por dos conductores sin polaridad al que se conectan todos los dispositivos. Está diseñado para el almacenamiento y establecimiento de niveles de iluminación (escenas), agrupación de luminarias en una dirección de grupo, controlar la atenuación, encendido y apagado en los balastos. Otros fabricantes han tenido que crear pasarelas para comunicar su protocolo con DALI, ya que, este protocolo, debido a su especificidad, es preferente su uso frente a otros más genéricos.

La capa física del protocolo consta de dos cables sin polaridad, al que se conectan todos los dispositivos de la instalación. La comunicación se realiza mediante el envío de tramas mediante una codificación Manchester, en la que el estado del bit se define por la transición en el centro de la ventana del bit, es decir, si se produce un flanco ascendente o descendente, el protocolo de acceso al medio es mediante el establecimiento de retardos a la escritura según prioridades, a una velocidad de 1200 baudios, basada en una relación maestro-esclavo entre la unidad de control y los balastos. Para ello, las unidades de control se valen del envío de tramas de 19 bits para mensajes de comunicación, o 27 bits para mensajes de comprobación de estados, y la recepción de tramas de 11 bits.

Las tramas están constituidas por los siguientes datos.

- Trama de comunicación:
  - a. 1 bit de inicio.
  - b. 1 byte de dirección: 1 bit de dirección individual o grupal, 6 bit de dirección, 1 bit de selección.
  - c. 1 byte de datos.
  - d. 2 bits de stop.
- Trama de comprobación de estados:
  - a. 1 bit de inicio.
  - b. 1 byte de dirección: 1 bit de dirección individual o grupal, 6 bit de dirección, 1 bit de selección.
  - c. 2 bytes de datos.
  - d. 2 bits de stop.
- Trama de recepción:
  - a. 1 bit de inicio.
  - b. 1 byte de datos.
  - c. 2 bit de stop.

Adicionalmente, se requiere una fuente de alimentación para cada ramal de DALI para generar la tensión de control. Los niveles de tensión a los que trabaja son más altos que los usados en la lógica TTL, siendo el nivel bajo de tensión de  $-6.5\text{ V}$  a  $6.5\text{ V}$  y el nivel alto de  $9.5\text{ V}$  a  $22.5\text{ V}$ .

El control de acceso al medio se realiza mediante el establecimiento de cinco niveles de prioridad, donde la prioridad 0 es la más alta y 4 la más baja, que en la práctica se corresponde con el tiempo de espera mínimo para el envío de una trama tras la finalización de la anterior, siendo éstas:

- Prioridad 0 ( $12.00\text{ ms} \pm 480\text{ ms}$ ): para mensajes que han de ser repetidos.
- Prioridad 1 ( $13.00\text{ ms} \pm 480\text{ ms}$ ): comandos de control del usuario que afectan a los ajustes de la iluminación.
- Prioridad 2 ( $14.00\text{ ms} \pm 480\text{ ms}$ ): configuraciones generales.

- Prioridad 3 (15.00 ms ± 480 ms): comandos automáticos de los ajustes de iluminación.
- Prioridad 4 (16.00 ms ± 480 ms): consultas.

Cada maestro es capaz de direccionar hasta 64 dispositivos, si se utiliza el protocolo extendido se puede llegar hasta 128 dispositivos. Admite múltiples topologías: en estrella, árbol o mixto como se muestra en la Figura 1, las direcciones son proporcionadas secuencialmente, siendo independiente de su disposición física. Este protocolo permite la agrupación de dispositivos hasta en 16 diferentes grupos, a los que el maestro se dirigirá con una dirección específica como se explicará más adelante. Por otro lado, los balastos son capaces de almacenar múltiples parámetros como 16 valores de intensidad de iluminación, su dirección propia, las direcciones de los grupos a los que pertenece, el valor de iluminación máximo y mínimo, el valor de iluminación al encendido y el establecimiento de un tiempo para un apagado o encendido progresivo.

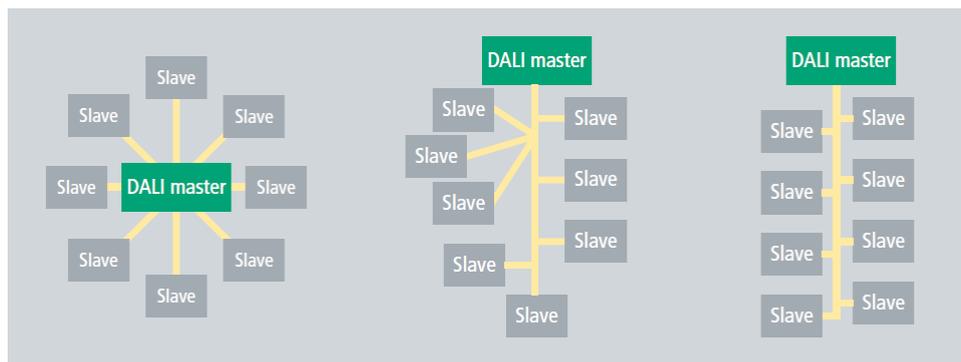


Figura 1. Topologías del protocolo DALI.

La caída de tensión entre fuente y dispositivo no debería exceder los 2.0 V. Se recomienda una instalación de Clase 1, pero con las medidas adecuadas se puede cablear como una instalación de Clase 2. Las longitudes máximas del bus recomendada por la norma, dependiendo del tipo de instalación, del tipo de cable utilizado y de su sección son las descritas en la Tabla 2 y Tabla 3.

Corriente suministrada	Número máximo de dispositivos	Longitud máxima según la sección				Longitud máxima con conexión de cables en paralelo
		1 mm <sup>2</sup>	1.3 mm <sup>2</sup>	1.6 mm <sup>2</sup>	2 mm <sup>2</sup>	
250 mA	128	175 m	275 m	435 m	700 m	900 m

Tabla 2. Instalación de cable rígido - Clase 1.

Corriente suministrada	Número máximo de dispositivos	Longitud máxima según la sección		Longitud máxima con conexión de cables en paralelo	
		CAT 3	CAT 5	CAT 3	CAT 5
250 mA	128	20m	40 m	900 m	900 m

Tabla 3. Instalación de cable trenzado - Clase 2

Ingenium fue fundada en el año 1998 en Oviedo. Es una empresa cuyo objetivo es fabricar equipos de domótica a un precio asequible. Inicialmente se planteó seguir uno de los dos protocolos existentes con más peso en el mercado, KNX® o LonWorks®, pero cualquiera de estas opciones haría muy costosos los productos, por lo que se creó un protocolo propietario, BUSing®, de código abierto.

### 1.1.2 Ingenium

---

Desde su creación, Ingenium ha desarrollado un extenso catálogo de unos 100 dispositivos BUSing®. También dispone de un catálogo de productos de KNX, lo que le permite ser más competitivo en el mercado de la domótica.

El protocolo BUSing® fue concebido desde el principio como un sistema distribuido para la domótica y la inmótica. Por tanto, tiene cabida tanto en viviendas como en oficinas, hospitales o cualquier habitáculo al que se le deseen dar prestaciones de automatización.

Sus aplicaciones abarcan el control de la temperatura de un recinto, control de la iluminación, alarmas, cámaras IP, persianas y sensorización con dispositivos conectados a un bus, o que se comuniquen de forma inalámbrica entre sí. Esto es posible debido a que cada dispositivo posee su propio microprocesador o microcontrolador, lo que les otorga de autonomía, pero también de la capacidad de comunicarse con toda la instalación, dando al conjunto el sentido de sistema de automatización de vivienda.

## 1.2 Objetivos

---

La pasarela creada con la empresa Ingenium bajo la marca “Bes”, es un autómatas capaz de establecer una comunicación entre dispositivos KNX a otros dispositivos DALI. Partiendo de un diseño existente, y basándose en la funcionalidad de un programa desarrollado para este dispositivo, se va a añadir una pantalla TFT de 2.2” dotada de un menú de configuración, con el que se podrán definir múltiples parámetros de los balastos DALI de una instalación o actuar directamente sobre ellos. La comunicación entre la pantalla y la pasarela debe de ser rápida y fiable, puesto que ésta debe atender a las tramas que le lleguen por los buses KNX y DALI. En la pantalla se mostrará la información necesaria para que, de forma inequívoca, el usuario conozca el estado en el que se encuentra tanto la pasarela como la instalación, así como las órdenes que se estén enviando. Desde la pantalla se podrá encender, apagar y regular el nivel de intensidad de los balastos, asignarles direcciones de grupo para un manejo en conjunto de los mismos, direccionar de forma automática los nodos del bus DALI y cambiar el idioma de la interfaz.

Se creará un dispositivo para el catálogo de Ingenium en la herramienta EST5. Este dispositivo es un objeto virtual que contiene los parámetros y objetos de comunicación de la pasarela física. Para la parametrización se creará interfaz visual que dividirá en secciones los distintos nodos y grupos que se van a configurar. Una vez seleccionados éstos y relacionados los objetos de comunicación con los de otros dispositivos de la instalación conectados al bus KNX, como puede ser un regulador de luz, se podrá proceder a cargar el programa en la pasarela de la forma habitual con el EST5.

## 1.3 Alcance

---

Mediante el entorno de programación MPLAB 8, utilizando el lenguaje C, se programará un microcontrolador PIC que hará las veces de controlador de la pantalla. Para ello se deberá tener en cuenta el diseño de la placa sobre la que irá montado. Tanto el puerto de comunicación de este microcontrolador con la pantalla como con la pasarela son serie, sin embargo, para representar los datos en pantalla hacen falta tres elementos más, una señal de modulación de ancho de pulso para el brillo, y dos pines que, dependiendo

de su estado, la pantalla interpretará que la trama que reciba es de datos o de comandos, entre estos dos elementos se establece una relación maestro-esclavo.

Se programarán dos máquinas de estado, una destinada al envío de datos por el puerto, que se ejecutará en el bucle principal, y la otra se encargará de la visualización en pantalla y la creación de la trama que se desee transmitir, que sólo se ejecutará cuando se realice la llamada a la función que la contiene, de este modo, la información en aparecerá estática en caso de que no haya ningún cambio en el estado del menú, de otro modo ésta parpadearía. Se ha dotado de tres pulsadores al microcontrolador que ejecutarán órdenes sólo en caso de ser accionados. Dos de estos pulsadores se utilizarán en la mayoría de los casos para seleccionar una opción del menú y el tercero, generalmente, de confirmación de esta selección. Cualquiera de los tres provocará cambios en variables internas como en la visualización de los datos en pantalla. Se dotará a la pantalla de un estado de espera, durante el cual se transmitirán las tramas, retornando al estado en el que se encontrase anteriormente cuando se haya recibido de la instalación un acuse de recibo (ACK).

Las tramas comunicadas entre el microcontrolador que maneja la pantalla y el que maneja la pasarela estarán formadas por bytes definidos según el protocolo DALI, pero la trama no tendrá el formato de este protocolo por simplicidad. Se ha acordado que los bytes sean: comando, dirección del remitente y dos bytes de datos, a una velocidad de 9600 baudios, transmitiendo la trama de 4 bytes, excepto en el caso de espera de órdenes por parte de la pasarela, cuando se enviarán periódicamente ceros, preguntando al microcontrolador de la pantalla si hay alguna orden preparada, en caso de no haberla, se contestará con otro cero.

Se utilizará la herramienta *Manufacturer Tools* para crear el dispositivo en catálogo de Ingenium para ETS 5. El objeto estará formado por dos elementos, los parámetros y los objetos de comunicación. Los parámetros creados aparecerán como una interfaz que estará dividida en múltiples secciones, mientras que los objetos de comunicación sólo serán activados según se requiera su uso, esto estará definido por los parámetros o *scripts*. En caso de tener todos los objetos visibles en todo momento, sólo la pasarela mostraría más de 500 elementos, que, a pesar de tener un nombre identificativo propio, el instalador tardaría más de lo necesario en encontrar sólo los que él necesita. Ambos elementos se deben relacionar con determinadas posiciones de la memoria de la pasarela donde se encuentran sus correspondientes parámetros u órdenes, según corresponda, con una tabla de correspondencias. Debido a las limitaciones de la herramienta de creación, esto último no ha podido ser realizado, por lo que se deja planteado como una futura ampliación.

## 1.4 Descripción del banco de pruebas

---

En la Figura 2 se representa la topología del banco de pruebas sobre el que se realizó el trabajo descrito en el alcance. En el ordenador se encuentra el programa ETS 5 que se conecta mediante USB a un programador de KNX, éste a su vez se conecta con el mismo bus a una fuente de alimentación y a la pasarela. Por otro lado, se encuentran los balastos DALI, todos ellos alimentados a tensión de red, y conectados por bus DALI a la pasarela. A ésta, además, se conectará una pantalla TFT mediante el protocolo RS232, con la que se comunicará mediante órdenes descritas en el protocolo DALI, pero no con la trama definida por éste por simplicidad.

En el caso de haber una trama preparada, el primer byte se corresponde con un comando, los utilizados son "NORMAL", "REPETIDO", y "EXTENDIDO\_1", cuyos valores decimales son respectivamente 170, 171 y 172. Depende de si es una trama que requiere un dato de la memoria del balastro, se quiere actuar sobre él o

modificar algún parámetro se utilizará uno u otro. El segundo byte es un valor fijo, 254, que se corresponde con la dirección de la pasarela, de este modo, se podrá identificar el origen de la orden enviada. El tercer y cuarto byte se corresponden con datos, su rango irá desde el valor decimal 0 al 255, con ellos se puede transmitir la dirección de destino, un valor de intensidad de iluminación, la orden de guardar un valor en un registro auxiliar, denominado DTR, para que posteriormente sea copiado a un parámetro de configuración (concretamente para esto se requiere el comando “REPETIDO”).

La contestación de la pasarela tendrá la siguiente forma, el primer byte será un acuse de recibo (ACK), tomando los valores 0 o 1. El segundo byte será siempre 0. Y el tercer y cuarto byte contendrá la respuesta del balastro, pudiendo ser éstas variadas. Se muestra más detalladamente en la Tabla 4 y en la Tabla 5.

Nombre de la trama	Valor de la trama	Descripción	Respuesta	
<b>Nueva instalación</b>	EXTENDIDO_1	Borra las direcciones, grupos y escenas actuales de los balastros y se les proporciona una dirección nueva.	0x01	
	0xFE		0x00	
	0X12		0x34	
	0X34		0x12	
<b>Extender instalación</b>	REPETIDO	Prepara la instalación para un nuevo equipo, le da una nueva dirección en caso de que no tenga	0x01	
	0XFE		0x00	
	0XA5		0x00	
	0X00		0x00	
<b>Detección</b>	NORMAL 0xFE 0x00 – 0x7F 0x99	Pregunta desde la dirección 0x00 a la 0x3F, la existencia de un balastro en la instalación en dicha dirección y de qué tipo de balastro se trata.	Inexistencia	
			Presencia	
			0x01	0x01
			0x00	0x00
			0x00	0xFF
			Emergencia: 0x01 LED: 0x06	
<b>Obtener grupos del 0 al 7</b>	NORMAL 0xFE 0x00 – 0x7F 0xC0	Se establece una dirección de nodo al que se le cuestiona a que grupos del 0 al 7 pertenece, contestará con un valor que en binario se interpreta como un vector, en la que los 1 es pertenencia al grupo correspondiente al índice en el que se encuentra.	0x01	
			0x00	
			0xFF	
			0x00 - 0xFF	
<b>Obtener grupos del 8 al 15</b>	NORMAL 0xFE 0x00 – 0x7F 0xC1	Es igual que el anterior, pero el grupo se corresponde a un valor 8 unidades superior al índice en el que se encuentre el valor 1.	0x01	
			0x00	
			0xFF	
			0x00 - 0xFF	

Tabla 4. Descripción de las tramas programadas – Parte I.

Nombre de la trama	Valor de la trama	Descripción	Respuesta
<b>Asignar un nodo a un grupo</b>	REPETIDO	Trama de asignación de un nodo del 0x00 al 0x7F en valor y decimal, al grupo de valor 0 al 15.	0x01
	0xFE		0x00
	0x00 – 0x7F		0x00
	0x60 + (0x00 – 0x0F)		0x00
<b>Desasignar un nodo de un grupo</b>	REPETIDO	Trama de desasignación de un nodo del 0x00 al 0x7F en valor y decimal, al grupo de valor 0 al 15.	0x01
	0xFE		0x00
	0x00 – 0x7F		0x00
	0x70 + (0x00 – 0x0F)		0x00
<b>Establecer el valor de iluminación de una escena</b>	NORMAL	Trama de establecimiento de una escena de valor de 0 al 15, en un nodo del 0x00 al 0x7F en valor decima.	0x01
	0xFE		0x00
	0x00 – 0x7F		0x00
	0x10 + (0x00 – 0x0F)		0x00
<b>Establecer valor de iluminación máximo</b>	NORMAL	Siendo la dirección del nodo de 0x00 a 0x7F, o de 0x80 a 0x9F en caso de ser un grupo, esta orden hace que los balastos de la dirección elegida se pongan en su valor de iluminación máximo.	0x01
	0xFE		0x00
	0x00 – 0x9F		0x00
	0x05		0x00
<b>Establecer valor de iluminación mínimo</b>	NORMAL	Siendo la dirección del nodo de 0x00 a 0x7F, o de 0x80 a 0x9F en caso de ser un grupo, esta orden hace que los balastos de la dirección elegida se apaguen.	0x01
	0xFE		0x00
	0x00 – 0x9F		0x00
	0x00		0x00
<b>Guardar valor en DTR</b>	NORMAL	Guarda un valor del 0x00 al 0xFF en el registro auxiliar.	0x01
	0xFE		0x00
	0xA3		0x00
	0x00 – 0xFF		0x00
<b>Grabar el valor máximo</b>	REPETIDO	Coge el valor del DTR y lo graba en el parámetro de valor máximo del nodo o grupo escogido.	0x01
	0xFE		0x00
	0x00 - 0x9F		0x00
	0x2A		0x00
<b>Grabar el valor mínimo</b>	REPETIDO	Coge el valor del DTR y lo graba en el parámetro de valor mínimo del nodo o grupo escogido.	0x01
	0xFE		0x00
	0x00 – 0x9F		0x00
	0x2B		0x00
<b>Grabar valor de escena</b>	REPETIDO	Coge el valor del DTR y lo graba en la escena indicada del 0 a 15, del nodo escogido.	0x01
	0xFE		0x00
	0x00 – 0x7F		0x00
	0x40 + (0 – 15)		0x00

Tabla 5. Descripción de las tramas programadas – Parte II.

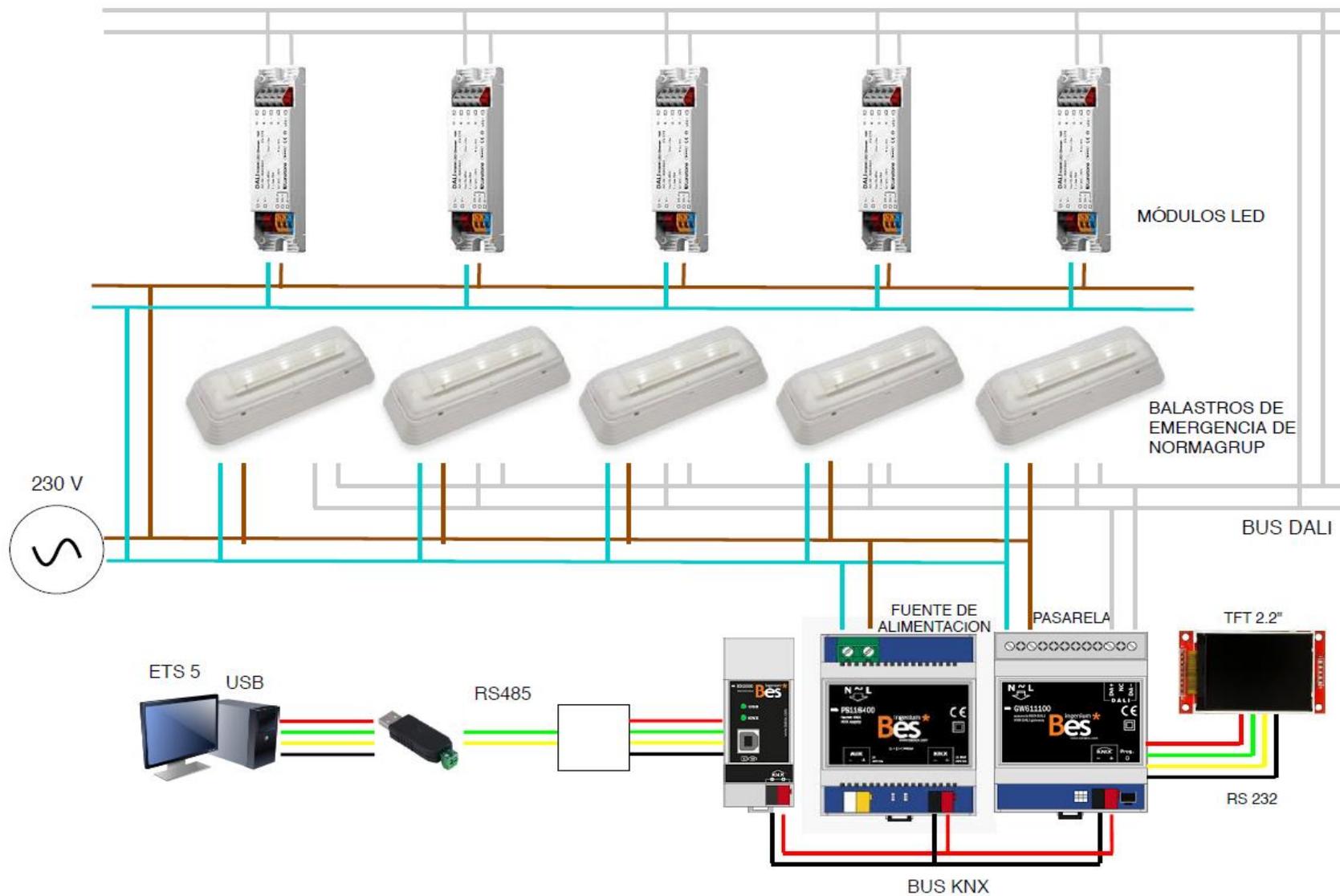


Figura 2. Topología del banco de pruebas.

## 2 Software de programación ETS 5

El ETS 5 es un software proporcionado para la asociación KNX para la configuración de instalaciones domóticas bajo su protocolo. Siendo independiente de cualquier fabricante, permite la configuración de cualquier instalación domótica basada en KNX. Por lo tanto, cada fabricante puede crear su propio catálogo e importar los parámetros de configuración y objetos de comunicación, haciendo más sencilla la configuración de la instalación, pudiendo llegar a tener un catálogo como el que se muestra a continuación.

En la Figura 3 se muestra el aspecto que tiene el catálogo de la herramienta. En la columna izquierda, aparecen todos los fabricantes de productos de KNX que han decidido incluir sus dispositivos en ETS 5. En el centro se sitúan los dispositivos del fabricante seleccionado con su nombre, medio de comunicación y versión del dispositivo del catálogo. El proyecto de la instalación se realiza seleccionando aquellos necesarios sin importar el fabricante, configurando sus parámetros y relacionando los objetos de comunicación de los dispositivos entre sí.

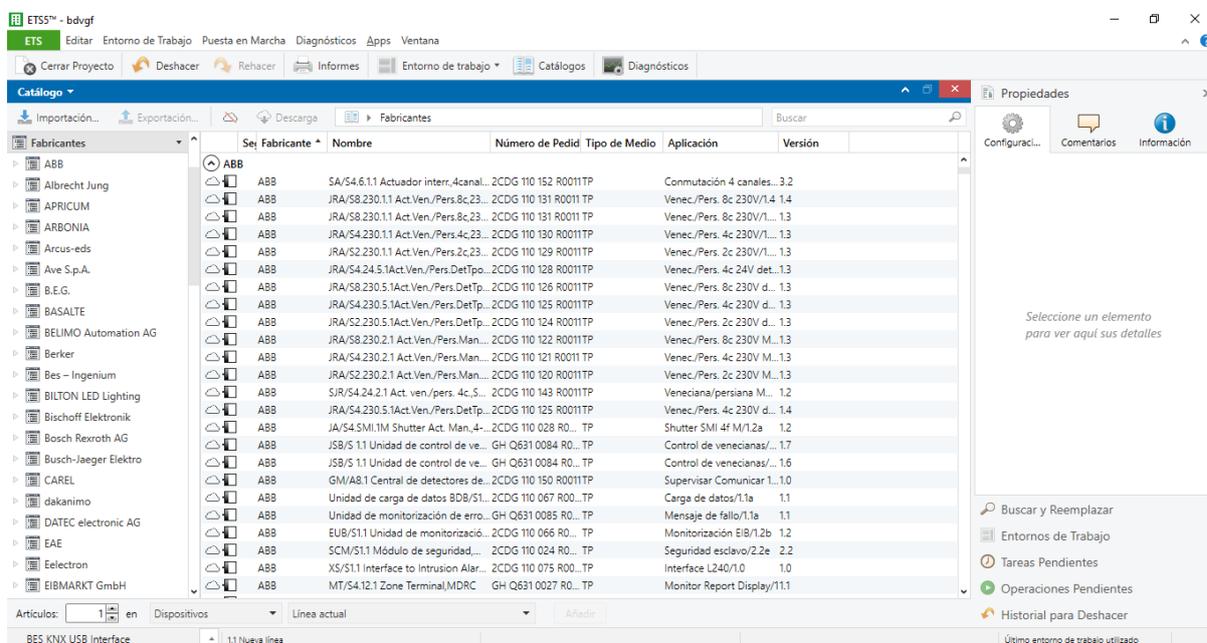


Figura 3. Catálogo de fabricantes y dispositivos del ETS 5.

El ETS5 permite organizar el proyecto de múltiples formas: según la topología, la distribución en el edificio o según los dispositivos instalados.

En el primer caso, como se muestra en Figura 4, la estructura de datos realiza su primera división en áreas que se dividen en líneas a las que se introducen los dispositivos, a las que se pueden añadir finalmente los dispositivos que contienen los parámetros y los objetos de comunicación que se podrán unir entre sí. Como se puede observar en la figura siguiente, los dispositivos adquieren una numeración correspondiente con la estructura de datos. Esta estructura tiene una correspondencia directa con la topología de la instalación.

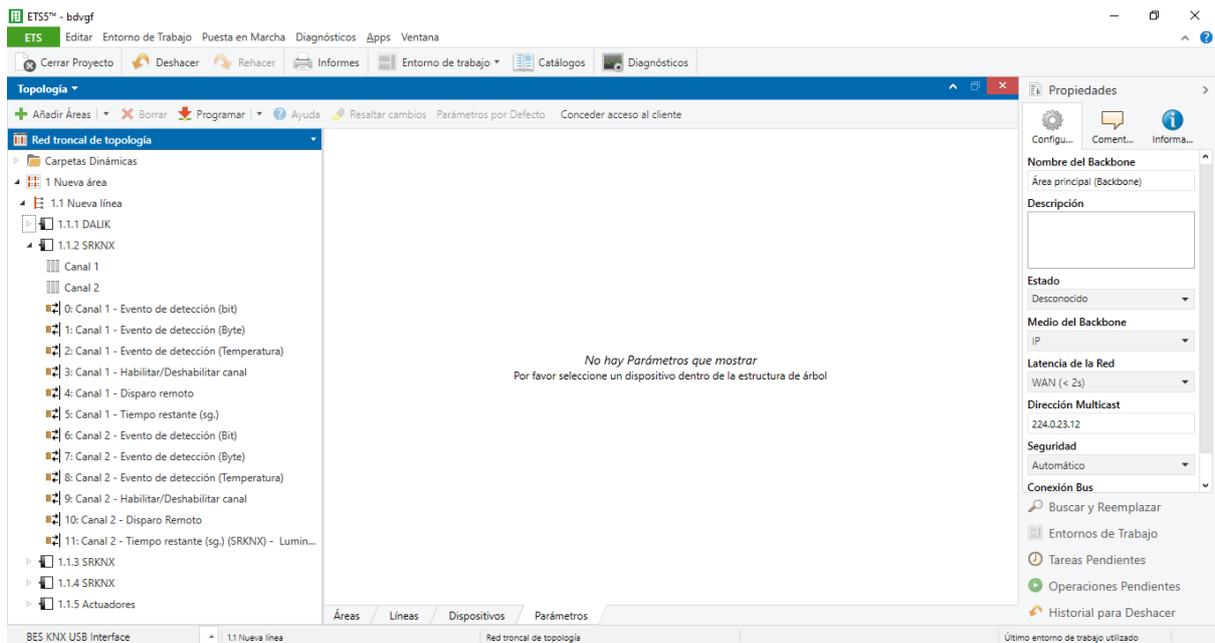


Figura 4. Vista de topología del proyecto de ETS 5.

En la Figura 5 se muestra el aspecto que tiene la estructura de datos en el caso de organizarse en edificios. Se dividen en partes de edificio, que, a su vez se dividen en plantas, éstas en estancias, tal como lo haría un bloque de pisos, a las que se pueden añadir finalmente los dispositivos, éstos mantienen la numeración adquirida en la estructura de topología.

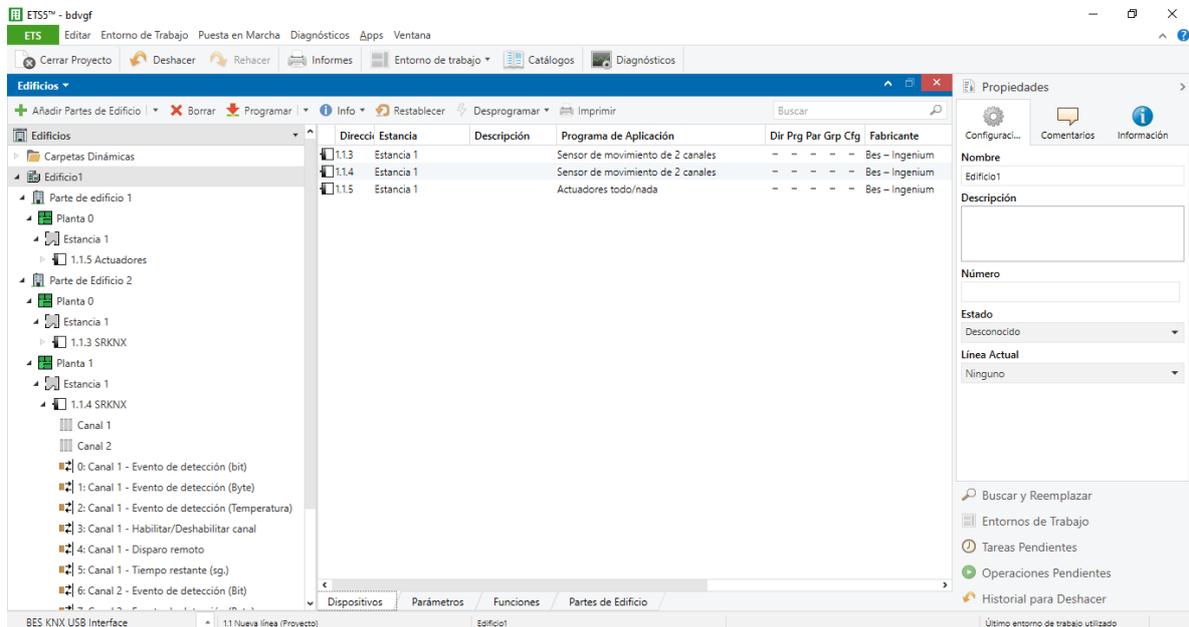


Figura 5. Despliegue de vista de edificio del proyecto en ETS 5.

La lista de parámetros no posee una estructura de datos, sino que se muestra una lista de todos los dispositivos de la instalación, nuevamente, éstos mantienen la numeración adquirida en la topología. En cualquier caso, si se selecciona un dispositivo, se pueden establecer sus parámetros de funcionamiento. En la Figura 6 se muestra la selección del sensor SRKNX de Bes – Ingenium, el menú dependerá del dispositivo seleccionado.

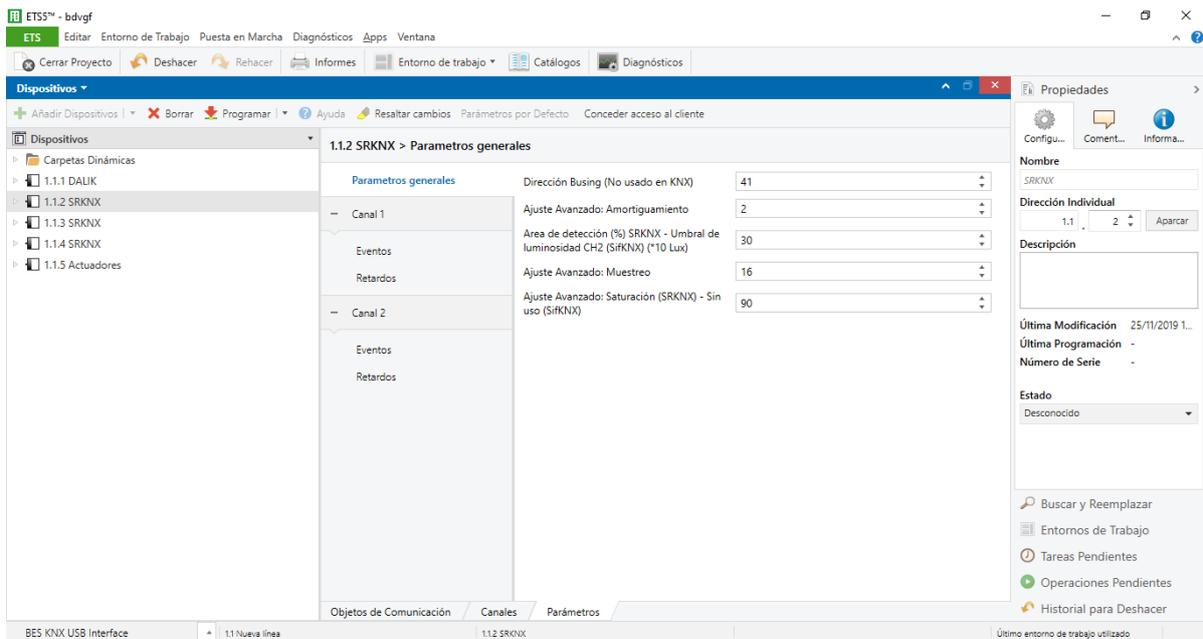


Figura 6. Lista de los parámetros del dispositivo.

La interfaz de parámetros y la creación de los objetos de comunicación se realizan mediante otra herramienta de la asociación KNX denominada *Manufacturer Tool*. Como se desarrollará en un capítulo posterior, ésta es la herramienta que permite a los fabricantes desarrollar los catálogos para KNX.

En el caso de estudio, la pasarela DALIK, se carecía de este menú de parámetros, por lo que para su configuración se debía hacer uso de un plugin externo lanzándolo desde la pestaña de parámetros del ETS5, seleccionando la opción que aparece en pantalla con la etiqueta “*Diálogo de parámetros específicos del producto Abrir*”.

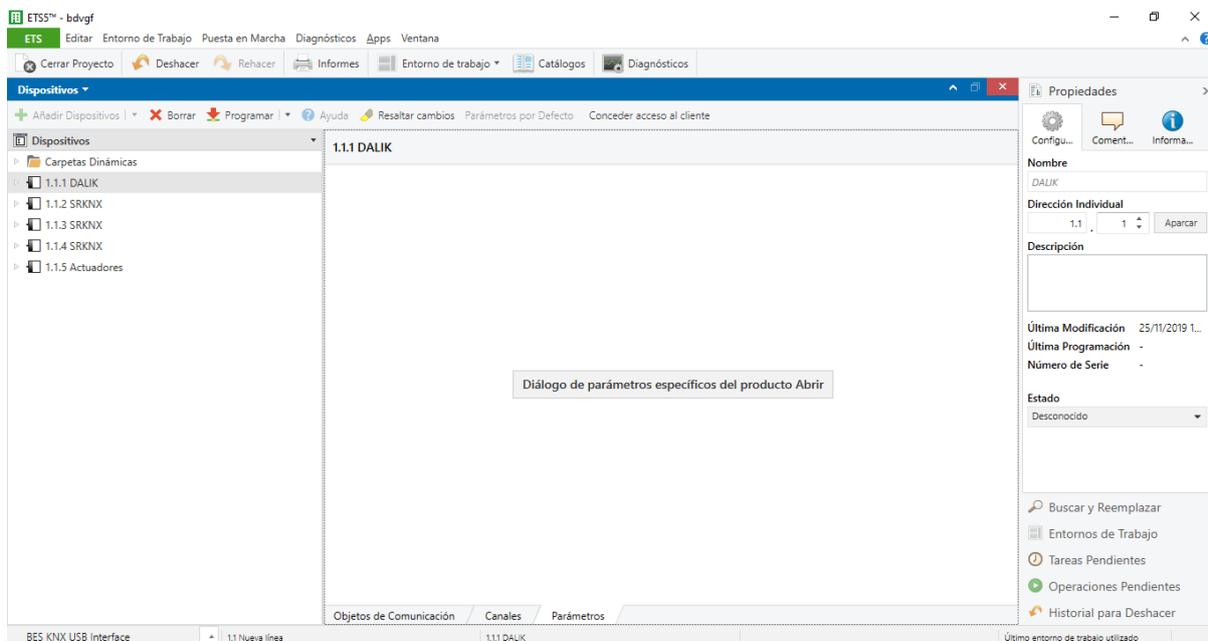


Figura 7. Vista de parámetros de un dispositivo de configuración mediante plugin.

Como se ha explicado, los dispositivos tienen objetos de comunicación que se utilizan para actuar sobre otros o ser modificados por otros que compartan dirección. La dirección tiene un formato del tipo X/X/X, donde X es un valor entero entre 0 y 255, se define en la quinta columna: dirección de grupo.

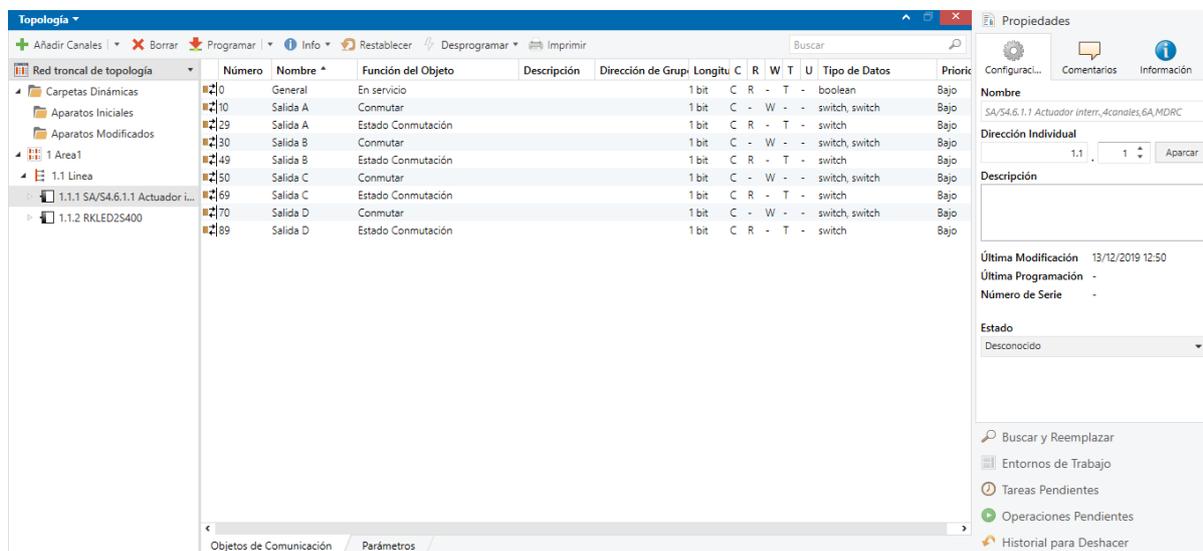


Figura 8. Lista de objetos de comunicación de un actuador.

El plugin permite la programación de dispositivos tanto si están conectados como desconectados de una instalación. Al añadir nodos o grupos, en el ETS 5 aparecerán sus correspondientes objetos de comunicación, y desde el plugin se podrán parametrizar las opciones que se detallan en la mitad derecha de la Figura 9.

La ventaja de realizar la tarea de configuración de una instalación mediante el catálogo respecto al plugin reside en que su interfaz resulta más intuitiva para un instalador. El protocolo KNX permite la agrupación de dispositivos según el direccionamiento configurado, siendo éste independiente de los grupos que se hayan creado con DALI. Por el contrario, con *Manufacturer Tools* tan sólo se pueden crear hasta 255 objetos de comunicación, lo que puede llegar a limitar en gran medida la funcionalidad del dispositivo. A esto se añade la ventaja del plugin, que es capaz de establecer una comunicación con la instalación, es decir, puede enviar órdenes y también recibir tramas, lo que le permite realizar la función de escaneo de la instalación, de esta forma se pueden conocer la dirección de los nodos instalados en caso de que se trabaje en línea. Permite la función de trabajo fuera de línea, utilizada tan sólo para configurar la pasarela. Para ello es necesario especificar el número de nodos, grupos y en que direcciones se encuentran según el protocolo DALI. La Figura 9 muestra el aspecto del *plugin* al que se le ha añadido un balastro en la dirección 0.

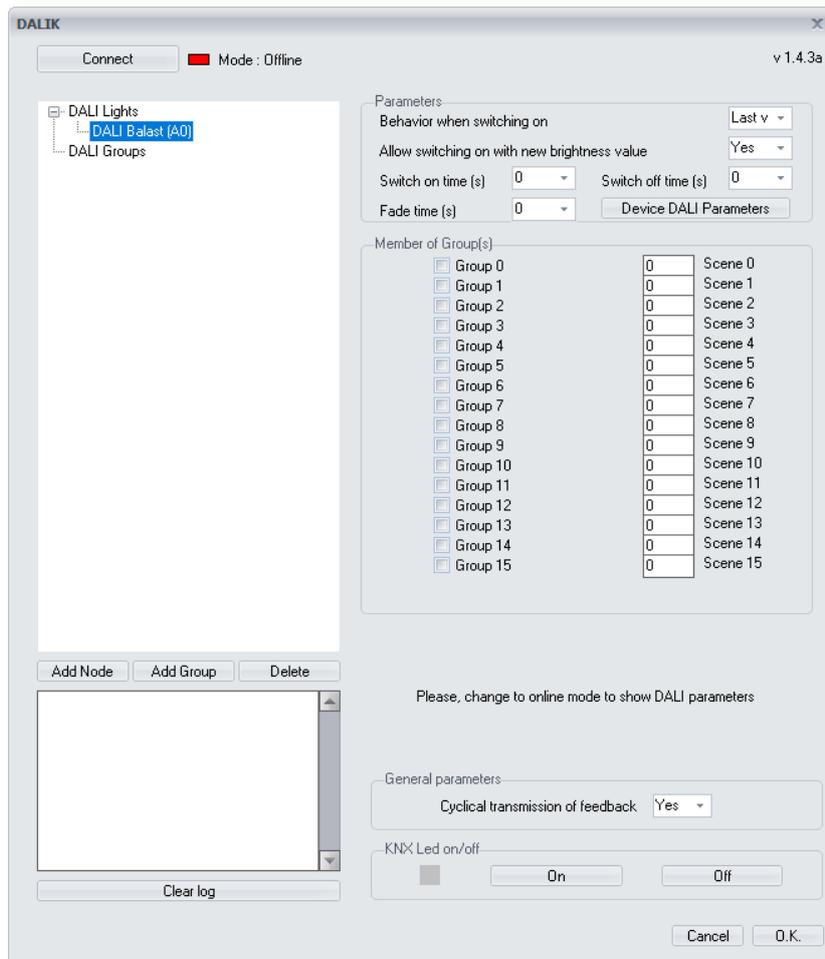


Figura 9. Aspecto del plugin de parametrización de la pasarela.

## 2.1 Programación en línea

Este modo de programación está pensado para el diseño y configuración de proyectos cuando la pasarela se conecta tanto a una instalación KNX como a una instalación DALI. Previamente, se requiere la parametrización de los balastos DALI, debe ser llevada a cabo por el programador desde el ETS 5.

Durante la programación en línea, la pasarela debe estar conectada a la instalación KNX y a la fuente de alimentación. Así mismo, cada dispositivo DALI debe ser conectado a la línea DALI y a la fuente de alimentación, en caso contrario no serán detectados correctamente durante el proceso de búsqueda.

En la puesta en marcha en línea de la instalación DALI hay tres funciones principales disponibles. Estas funciones están centradas en añadir todos los balastos o grupos DALI presentes en la instalación al proyecto ETS.

- Nueva instalación: la pasarela localizará cada balastro DALI llevando a cabo un escaneo progresivo de la línea DALI, identificando todos los dispositivos y asignando una dirección DALI a cada uno. El tiempo invertido para la búsqueda completa puede variar dependiendo del número de dispositivos DALI conectados.
- Detección: la pasarela localizará cada balastro DALI llevando a cabo un escaneo progresivo de la línea DALI, identificando todos los dispositivos y añadiéndolos a su memoria. En este caso las direcciones

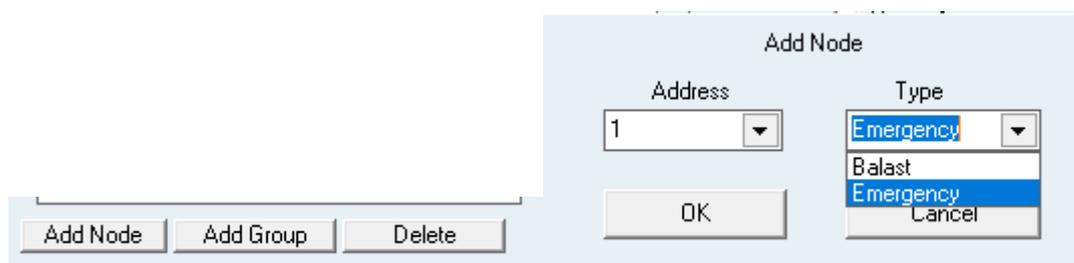
DALI de cada nodo no cambiarán. Debe utilizarse esta función cuando la línea DALI ya ha sido direccionada.

- Extensión de la instalación: Esta función está pensada para su uso en instalaciones DALI ya direccionadas, con el objetivo de incluir un nuevo balasto sin direccionar. La pasarela buscará el balasto DALI que no tiene una dirección en el árbol del proyecto y le asignará una libre.

## 2.2 Programación fuera de línea

---

Este modo permite añadir al árbol de nodos cada balastro individualmente indicando su dirección y tipo (normal o emergencia), así como los grupos que se van a utilizar y asignar nodos a éstos.



*Figura 10. Detalle del plugin donde se crean nodos y grupos de forma manual.*

Se puede comprobar la correspondencia con la instalación real cambiando a modo en línea y utilizando la función de detección. Una vez el proceso de detección haya terminado, el plugin comprobará las diferencias entre la instalación detectada y la programada, dando la opción si se descartarán los nodos no existentes en la instalación y añadiendo los que no estén especificados en el plugin.

### 3 Manufacturer Tool

*Manufacturer Tool* es la herramienta para la creación de entradas en las bases de datos de productos conformes con KNX. Cada base de datos contiene parámetros y objetos de comunicación. El proyecto creado en esta herramienta se divide en dos grandes secciones, la parte estática (*Static*), que contiene la definición de los parámetros y objetos de comunicación con la posición de memoria que ocupan, y la parte dinámica (*Dynamic*), que se utiliza para la representación en pantalla de los parámetros y definir qué objetos de comunicación estarán activos para su uso. Se estructuran en un árbol como el que se muestra en la Figura 11.

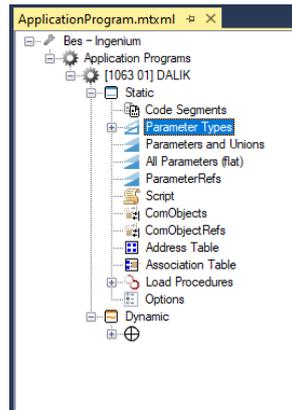


Figura 11. Despliegue del árbol de parámetros estáticos de *Manufacturer Tools*.

Los parámetros son elementos que definirán cómo funcionará el dispositivo. Para crear un parámetro, lo primero que se debe hacer es definir los tipos de parámetros, es decir, si se va a tratar de un número porcentual, si es un rango de valores, de cuántos bits es, si es entero, flotante, un parámetro de tiempo, ect. Los tipos se crean en el apartado "*Parameter Types*" dentro de *Static* abriendo los cuadros de diálogo mostrados a continuación.

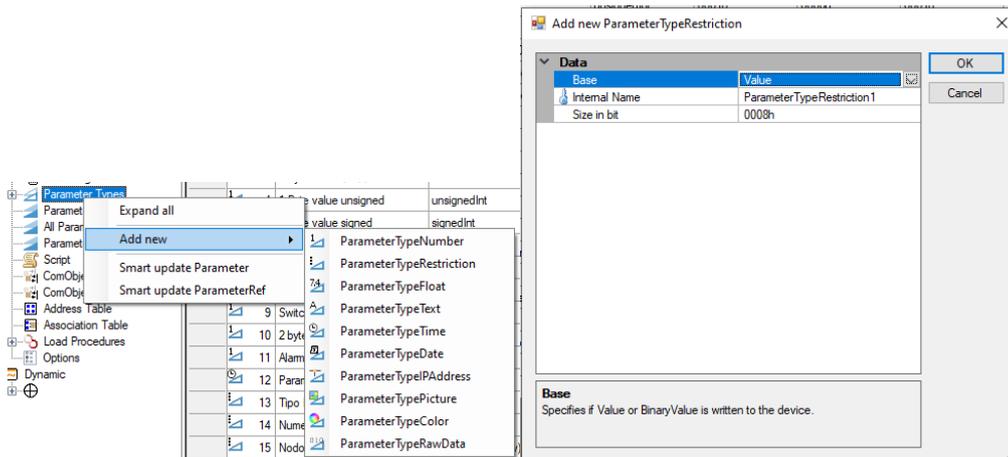


Figura 12. Detalle de operación para añadir tipos de parámetros nuevos.

Ha sido necesaria la creación de 12 tipos diferentes de parámetros, siendo utilizados los tipos *número*, *restricción* y *tiempo* para su definición. Se les ha dotado de un nombre que define su propia utilidad, lo que facilita su asignación posterior a los parámetros. Todos ellos tienen acotado el rango de valores que podrán

adquirir, en la Figura 13 se muestra el rango de tres parámetros en las columnas “MinInclusive” y “MaxInclusive”, no ha sido necesario que se defina en las demás, debido a que una variable tipo *entero* o *tiempo* va a adquirir todos los valores dentro del rango, mientras que a un tipo *restricción* se le puede definir qué valores concretos va a poder tomar sin necesidad de que sean consecutivos, o se muestra en la Figura 14.

	Index	Internal Name	Type	Size in bit	MinInclusive	MaxInclusive	Internal Descript...
▶	1	Enable/disable	Restriction	0008h			
	2	1 Byte value 0-100 %	Restriction	0008h			
	3	0-255	unsignedInt	0008h	0000h	00FFh	
	4	Parametro temporal	Time	0008h	0000h	00FFh	
	5	Tipo Balastro	Restriction	0008h			
	6	Numero de nodos	Restriction	0008h			
	7	NodoAGrupo	Restriction (Binary)	0008h			
	8	Numero de grupos	Restriction	0008h			
	9	Power ON	Restriction	0008h			
	10	GrupoANodo	Restriction	0001h			
	11	0/1	unsignedInt	0001h	0000h	0001h	
	12	Direccion de nodo	Restriction	0008h			

Figura 13. Lista de tipos de parámetros añadidos.

En el caso particular del tipo restricción, además, se debe crear una tabla con el rango de valores que va a poder adquirir, consta de un índice, de un valor en hexadecimal y de una etiqueta con la que se representará el parámetro, la figura que se muestra a continuación es el caso concreto de “Número de grupos”.

	Index	Value	Text
▶	1	0000h	0
	2	0001h	1
	3	0002h	2
	4	0003h	3
	5	0004h	4
	6	0005h	5
	7	0006h	6
	8	0007h	7
	9	0008h	8
	10	0009h	9
	11	000Ah	10
	12	000Bh	11
	13	000Ch	12
	14	000Dh	13
	15	000Eh	14
	16	000Fh	15
	17	0010h	16

Figura 14. Lista de valores que puede adquirir “Número de grupos” (Tipo restricción).

Una vez concretados los tipos de parámetros, en el apartado de “Parameters and unions” se podrán crear los parámetros atendiendo al uso que se le va a dar. Si va a ser utilizado para cálculos internos o comprobación de condiciones será un *parámetro virtual* que no será descargado en la memoria del dispositivo, si va a definir cómo funcionará el dispositivo se trata de un *parámetro de memoria*, en caso de que el parámetro sea una propiedad de un objeto de la interfaz será un *parámetro de propiedad*. Estos dos últimos son descargados en la memoria del dispositivo.

Index	Internal Name	Unique Number	Location	Parameter Type	Default Value	Initial Value	Size in bit	Text
1	Power on value B0	0001h	[4800h] Parameters+0000h	Power ON	00Ffh=Last Value		0008h	Power on value
13	Nodos	0007h	-	Número de nodos	0000h=0		0008h	Number of nodes installed

Figura 15. Arriba, vista del parámetro de memoria “Power on value B0”. Abajo, vista del parámetro virtual “Nodos”.

Con la creación de los parámetros va asociada la creación de sus referencias, mostradas en el apartado “ParameterRefs”. Son utilizadas para su uso en la estructuración de la interfaz visualización en el apartado “Dynamic”. Se tratan de punteros a los parámetros que cada referencia tiene asociada. Se identifican con un número único que apuntan al número único del parámetro. En *Manufacturer Tool* tiene un aspecto como el que se muestra en la Figura 16.

Index	Parameter	Internal Name	Unique Number	Location	Parameter Type	Default Value	Initial Value	Size in bit	Text
1	[0001h] Power on value ...	Power on value B0	0001h	[4800h] Paramet...	Power ON	00Ffh=Last Value		0008h	Power on value
2	[0002h] Switch with new...	Switch with new value...	0002h	[4800h] Paramet...	Enable/disable	0001h=Enable		0008h	Switch with new ...

Figura 16. Vista de dos parámetros de memoria en el apartado de “ParameterRefs”.

La creación de los objetos de comunicación se realiza escogiendo la opción “Add new ComObject” en el apartado “ComObjects”, visible en las opciones mostradas en la Figura 11, con lo que aparecerá una ventana de diálogo en la que se redactará el nombre con el que se va a identificar y su tamaño. Posteriormente, se le debe definir en la fila “DataPoint Type” (DPT), mostrada a la derecha en la Figura 17, el tipo de objeto que va a ser. Estos tipos vienen provistos por la propia herramienta de KNX al crear el proyecto. Existen hasta 51 DPT con sus correspondientes subdivisiones, el formato de la nomenclatura es del tipo X.YYY, donde X es el tipo de DPT y YYY la subdivisión que define su utilidad. Cada objeto contiene unas banderas que se han de habilitar o deshabilitar según el uso que el bus de datos le vaya a dar, estos son: comunicación (C), lectura (R), escritura (W), transmisión (T) y actualización (U). Al igual que ocurre con los parámetros, al crear un objeto también lo hace su referencia, que, a diferencia de las referencias de parámetros, no se utilizan para estructurar la interfaz, si no para permitir el uso de los objetos a los que apuntan.

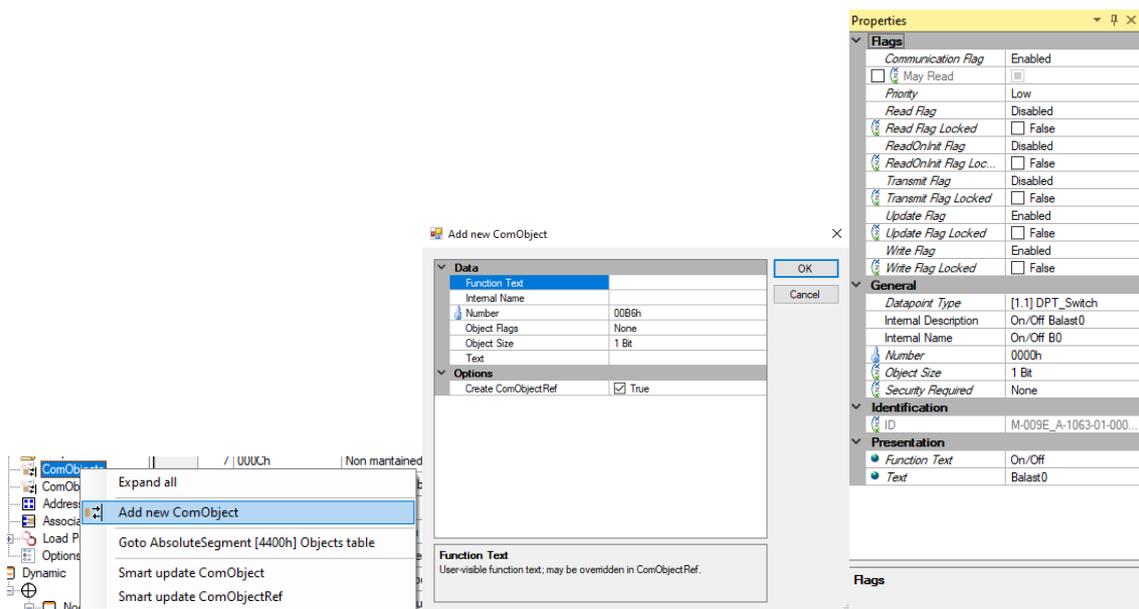


Figura 17. Detalle creación y configuración de objetos de comunicación en el apartado “ComObjects”.

Index	Number	Internal Name	Text	Function Text	Object Size	Priority	Flags	Datapoint Type
1	0000h	On/Off B0	Balast0	On/Off	1 Bit	Low	-WC-U-	[1.1] DPT_Switch
2	0001h	On/Off status B0	Balast0	On/Off status	1 Bit	Low	R-CT-	[1.2] DPT_Bool
3	0002h	Value B0	Balast0	Value	1 Byte	Low	-WC-U-	[5.1] DPT_Scaling

Figura 18. Vista de tres objetos de comunicación en el apartado de "ComObjects".

Index	ComObject	ComObject Num...	Unique Number	Internal Name	Text	Function Text	Object Size	Priority	Flags	Datapoint Type
1	[0000h] On/Off B0-On/Off	0000h	0001h		Balast0	On/Off	1 Bit	Low	-WC-U-	[1.1] DPT_Switch
2	[0001h] On/Off status B0-On/Off sta...	0001h	0002h		Balast0	On/Off status	1 Bit	Low	R-CT-	[1.2] DPT_Bool
3	[0002h] Value B0-Value	0002h	0003h		Balast0	Value	1 Byte	Low	-WC-U-	[5.1] DPT_Scaling

Figura 19. Vista de tres objetos de comunicación en el apartado "ComObjectRef".

El apartado de la interfaz de visualización, en la pestaña "Dynamic", inicialmente se encontrará vacía de contenido. Permite añadir: bloques de parámetros, en los que se podrán introducir referencias a referencias a parámetros, a objetos o una condición, que en función del valor de una variable activará o desactivará secciones de la interfaz. En este proyecto se ha realizado la estructura mostrada en la Figura 20.

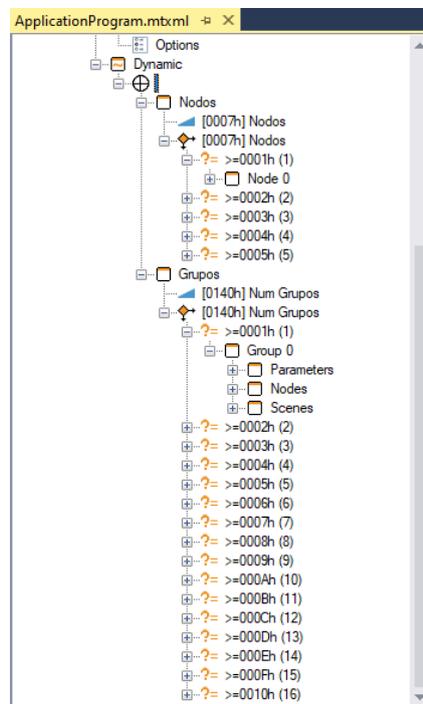


Figura 20. Árbol de objetos dinámicos.

Se han separado por un lado los nodos y por otro los grupos. Al inicio de cada apartado se ha colocado un parámetro virtual con el que se seleccionará el número de nodos y grupos que se configurarán, de este modo, en la interfaz sólo aparecerá información útil. El valor de ese parámetro será comprobado por el condicional que aparece a continuación representado por un rombo, que será el encargado de mostrar u ocultar la información pertinente en la interfaz, concretamente, se mostrarán los parámetros, la posibilidad de asignar nodos a grupos y las escenas del número de nodos y grupos seleccionados.

Para cada nodo la estructura será la siguiente. Se crearán apartados de parámetros, grupos y escenas. En el primero aparecerán los parámetros de modo de funcionamiento de los balastros, que dependerán del tipo de balastro seleccionado, estos pueden ser de iluminación o de emergencia. En el segundo se podrá seleccionar a qué grupos pertenecerá cada nodo. Por último, en escenas, se podrá decidir, en valor porcentual, el nivel de brillo de cada escena para cada balastro. El aspecto final es el mostrado en la imagen de la izquierda de la Figura 21. El apartado de grupos será similar a este, con la diferencia de que carecerá de

los parámetros y objetos de los nodos de emergencia, se detalla en la imagen derecha de la Figura 21. Se podrán asignar nodos a grupos y escenas grupales.

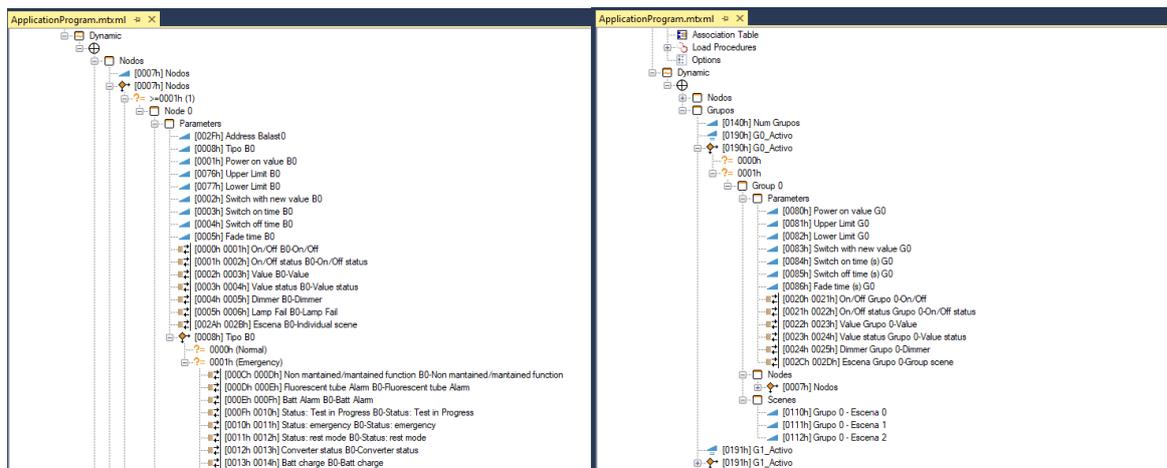


Figura 21. Izquierda, detalle de objetos del nodo 0. Derecha, detalle de objetos del grupo 0.

Finalmente, este proyecto se podrá exportar al ETS 5. Se visualizará como se muestra en la Figura 22. Para el ejemplo se han activado dos nodos y dos grupos. Se puede observar que aparece la misma estructura en la interfaz que la descrita en el árbol de las imágenes superiores.

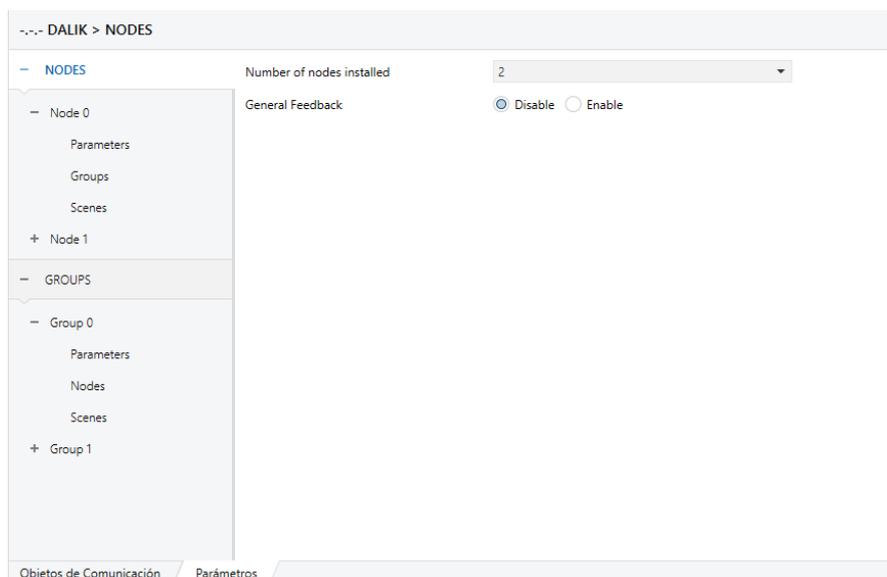


Figura 22. Aspecto de los parámetros generales del grupo de nodos de la pasarela en ETS 5.

Los parámetros, representados con un símbolo azul en la Figura 21, aparecerán en la interfaz en el orden en que hayan sido colocados en el árbol y con una representación correspondiente al tipo que se haya asociado de los creados al principio. Se muestra en las Figura 22 y Figura 24 el ejemplo del nodo 0, que es extrapolable al resto de nodos y grupos. Como se ha descrito anteriormente, en la lista de dispositivos, aparecerán los objetos de comunicación de cada uno según se vaya actuando sobre los parámetros. En la Figura 25 se puede ver un ejemplo de los objetos de comunicación que pueden ser utilizados.

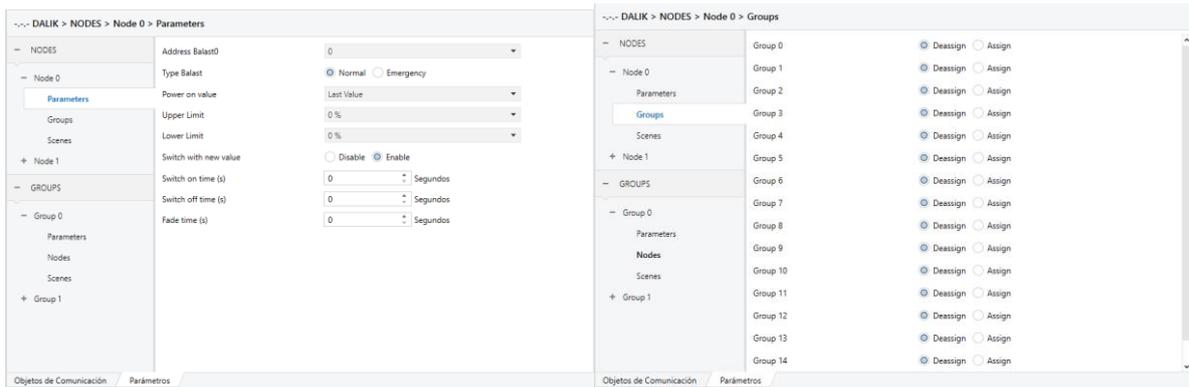


Figura 23. Izquierda, detalle de los parámetros del nodo 0. Derecha, lista de grupos a los que se puede asignar el nodo 0.

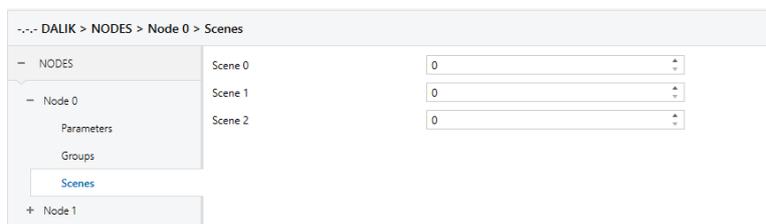


Figura 24. Vista de las escenas del nodo 0.

Número	Nombre	Función del Objeto	Descripción	Dirección de Grupo	Longitu	C	R	W	T	U	Tipo de Datos	Pric
0	Balast0	On/Off			1 bit	C	-	W	-	U	switch	Bajo
1	Balast0	On/Off status			1 bit	C	R	-	-	T	boolean	Bajo
2	Balast0	Value			1 byte	C	-	W	-	U	percentage (0..100%)	Bajo
3	Balast0	Value status			1 byte	C	R	-	-	T	percentage (0..100%)	Bajo
4	Balast0	Dimmer			4 bit	C	-	W	-	U	dimming control	Bajo
5	Balast0	Lamp Fail			1 bit	C	R	-	-	T	switch	Bajo
42	Balast0	Individual scene			1 byte	C	-	W	-	U	scene control	Bajo
11	Balast1	Lamp Fail			1 bit	C	R	-	-	T	boolean	Bajo
43	Balast1	Individual scene			1 byte	C	-	W	-	U	scene control	Bajo
10	Balast1	Dimmer			4 bit	C	-	W	-	U	dimming control	Bajo
9	Balast1	Value status			1 byte	C	R	-	-	T	percentage (0..100%)	Bajo
8	Balast1	Value			1 byte	C	-	W	-	U	percentage (0..100%)	Bajo
7	Balast1	On/Off status			1 bit	C	R	-	-	T	boolean	Bajo
6	Balast1	On/Off			1 bit	C	-	W	-	U	switch	Bajo
32	Grupo 0	On/Off			1 bit	C	-	W	-	U	switch	Bajo
33	Grupo 0	On/Off status			1 bit	C	R	-	-	T	boolean	Bajo
34	Grupo 0	Value			1 byte	C	-	W	-	U	percentage (0..100%)	Bajo
35	Grupo 0	Value status			1 byte	C	R	-	-	T	percentage (0..100%)	Bajo
36	Grupo 0	Dimmer			4 bit	C	-	W	-	U	dimming control	Bajo
44	Grupo 0	Group scene			1 byte	C	-	W	-	U	scene control	Bajo
40	Grupo 1	Value status			1 byte	C	R	-	-	T	percentage (0..100%)	Bajo
39	Grupo 1	Value			1 byte	C	-	W	-	U	percentage (0..100%)	Bajo
45	Grupo 1	Group scene			1 byte	C	-	W	-	U	scene control	Bajo
37	Grupo 1	On/Off			1 bit	C	-	W	-	U	switch	Bajo
41	Grupo 1	Dimmer			4 bit	C	-	W	-	U	dimming control	Bajo

Figura 25. Lista de objetos de comunicación activos.

Se describen con más detalle los parámetros y objetos de comunicación en el manual de programador en el documento denominado como MP\_GW611100\_version\_Academica.pdf.

## 4 Programación de la pantalla TFT

---

### 4.1 Algoritmo

---

El objetivo es controlar una instalación de balastros DALI desde una pantalla TFT que se comunica con una pasarela KNX-DALI.

Se utiliza un PIC 18F25K22 como controlador de la pantalla utilizando una comunicación por el puerto serie para la transmisión de datos a una velocidad de 9600 baudios, a su vez se comunica con otro microcontrolador PIC 18F1420-20SS situado en la pasarela mediante un protocolo RS-232 a 9600 baudios, al que está asociado una interrupción por recepción de dato. Las variables y funciones de las que se hablarán en este apartado se encuentran detalladas en [4.2. Variables y funciones utilizadas](#).

Para navegar por el menú de la pantalla se han incluido tres pulsadores en el PIC 18F25K22 usando su capacidad de lanzar una interrupción por flanco. En la placa se han situado en vertical, con motivo de hacer una interfaz más intuitiva, se ha decidido que el pulsador superior sea para navegar hacia arriba en las opciones del menú, el pulsador inferior para navegar hacia abajo y el central, dependiendo del estado, se utilizará para confirmar la opción seleccionada o para retroceder al menú anterior dependiendo del estado en el que se encuentre. Mediante la comparación del estado anterior y actual de los pulsadores se verifica el tipo de flanco que ha sucedido, de este modo, sólo cuando el pulsador es accionado el programa progresa. El esquema se encuentra en la página 1 del anexo.

La pasarela, periódicamente, envía al microcontrolador de la pantalla el valor cero, con ello se lanza la interrupción por recepción de datos. Durante unos milisegundos, la pasarela esperará una respuesta de éste, pudiendo darse dos situaciones, que haya una trama preparada para enviarse, con lo que se vuelca el contenido del vector *tx* en orden desde el índice 0 al 3 por el puerto de comunicación, o, por el contrario, que no haya ninguna trama que enviar, entonces, se responderá con un cero.

El menú se ha dispuesto como una máquina de estados en la función *pintaMenu*, cuya entrada será un valor numérico correspondiente con cada estado. Se utilizan tres variables para navegar por las opciones. La variable *aux\_menu*, en el que se guarda un número que será utilizado como el identificador del estado del menú, el menú principal es identificado con el número 1; *contador\_menu* que se utiliza como un contador que identifica en qué posición se encuentra el cursor en el menú, se incrementa al accionar el pulsador inferior y se decrementa con el pulsador superior siendo su rango entre los valores 1 y 4 inclusive; la tercera variable es *scroll*, se usa como un contador del número de veces se ha desplazado el menú hacia las opciones que están fuera del menú visible, si se desplaza hacia abajo se incrementa una unidad y al contrario se decrementa en la misma cantidad. Sumando las dos últimas variables descritas se puede conocer a qué opción del menú se está apuntando. Si se llegase a un estado no contemplado por esta máquina de estados, se mostraría una pantalla de error, deteniéndose la transmisión de datos. Para seguir operando, con pulsar el botón central se volvería a la pantalla del menú inicial.

Para evitar un mal funcionamiento del programa se ha limitado el valor de algunas variables: *contador\_menu* se ha limitado a 1 su valor mínimo y a 4 su máximo, porque se muestran hasta cuatro opciones en pantalla; la suma de ésta con *scroll* está limitado como máximo al valor de *opciones\_menu*, que se modifica para cada

estado de *pintaMenu*; tanto *valor\_sup*, como *valor\_inf*, como *valor\_escena* están acotados entre los valores 0 y 100, ya que se trata del valor porcentual de la intensidad de iluminación, sobrepasar este valor daría lugar a escrituras erróneas en los balastros tras convertirlo a una escala entre 0 y 255, tal como exige el protocolo que sea dado el valor de intensidad.

Por defecto, una vez se confirma la selección con el pulsador central, se suman los valores de *contador\_menu*, *scroll* y *opciones\_menu* y se multiplica el resultado por 10 para calcular el siguiente estado. Se ha decidido proceder de esta forma ya que se pueden identificar el valor de los estados como subestados, es decir, el estado 4110 significa que se ha escogido la primera opción del estado 410. Existe un problema de limitación del tipo de variable, que es capaz de alcanzar hasta el valor 65535, si se encadenan varios menús, el valor crece muy rápido pudiendo llegar a desbordar, por lo que en algunos casos se decide incrementar con un factor de 5 en lugar de 10. Existen los casos especiales de los estados 40 y 50, que sin importar cuál sea la selección siempre se va a pasar al mismo estado, al 410 en el primer caso y al 510 en el segundo, pero guardando en una variable cuál ha sido la selección, ya que se trata de la dirección del nodo o el grupo, respectivamente, que se quieren configurar y utilizando esta información para construir la trama que se enviará desde la máquina de estados del programa principal. Otro caso especial, común a la mayoría de los menús, es la última opción del menú, que se ha dispuesto que sea un retorno a la pantalla anterior, esto es posible porque cada estado guarda de dónde vino en *menu\_anterior*.

Las tramas sólo se enviarán cuando en una de las variables, *envio* o *envio\_2* se escriba el valor 1, sólo sucederá cuando se haya definido completamente la trama que se va a enviar en la función *pintaMenu*, que, en cuanto se reciba respuesta de confirmación de recibo de la pasarela, se reiniciarán al valor cero. Esta función establece que la escritura en esas variables sea excluyente, lo que evita fallos en la máquina de estados del bucle principal.

Por otro lado, la visualización de caracteres en pantalla se realiza mediante las funciones *LCD\_PutChar56x80*, *LCD\_PutSim* y *LCD\_PutChar*.

La primera se utiliza para representar el logo de "Bes" cuando se entra en el salvapantallas, debido a que son caracteres con una tipografía específica, de una altura que no es múltiplo a 16 y por tanto recorrer los bytes que conforman cada carácter se debe hacer con unos factores exclusivos para esta fuente. La segunda función se utiliza para representar los símbolos de la flecha de selección de opción que aparece en la derecha del menú y la flecha que indica retorno al menú anterior. Por último, la tercera función, hace uso de tres fuentes de caracteres, de alturas 16, 32 y 48 píxeles, identificadas con los valores 1, 2 y 3 respectivamente en la variable *tam*, en cualquier caso, creadas como filas de bytes, pero en tantas secciones de 8 bits de ancho como se indica en su tamaño y de una altura igual a la del carácter completo. Cada sección se va colocando a la derecha de la anterior completando la representación del carácter.

Se realiza mediante el uso de tres bucles anidados, el más externo se utiliza para recorrer cada sección, los dos internos recorren las filas y columnas de los caracteres de la biblioteca y los representa en pantalla existiendo una relación directa entre bit y píxel, considerando que encontrar un 1 significa que el bit es un píxel del objeto y un 0 que pertenece al fondo. Cada una de estas fuentes está guardada en sendas bibliotecas a las que se realizan llamadas según el tamaño que se quiera representar.

Con el motivo de una mejora visual, se han querido representar números más grandes en los estados 4155, 4160 y 4170, correspondientes con las elecciones de nivel de iluminación máximo, mínimo y de cada escena. Pero al carecer de memoria suficiente en el microcontrolador para alojar esta fuente, se ha decidido

multiplicar por dos mediante código el tamaño de la fuente de 24 por 48 píxeles, se ha identificado con el valor 7 en la variable *tam*. Para ello se requiere el uso de otros dos bucles anidados a los ya existentes que recorren la pantalla en ancho y alto por cada bit del carácter de la fuente.

Se han creado dos funciones complementarias a *LCD\_PutChar*, que hacen uso de ésta para representar palabras u oraciones si se utiliza la función *StringToChar*, o números en caso de llamar a *pintaNumeros*. Con ello se ha conseguido un ahorro en la memoria ROM del microcontrolador.

La primera hace uso de cadenas de caracteres que se almacenan en la variable *cadena*, la cual puede ser sobrescrita en cada estado de *pintaMenu* o cada vez que se hace una llamada a la función *cadena*. Ésta, es una máquina de estados que contiene las palabras en español e inglés de todas las opciones que se deseen traducir, se trata de dos *switch* anidados, donde el primero es el idioma escogido, codificado como 0 para el español y 1 para inglés; y el segundo será el que sobrescriba en la variable *cadena* la oración pertinente. La principal característica de esta función es que sólo hace falta escoger el idioma, pues cada palabra en español tendrá su correspondencia en inglés bajo el mismo índice. *StringToChar* separará la cadena en caracteres que se pasarán a la función *LCD\_PutChar*, estableciendo el primer carácter de cada palabra una unidad mayor al resto, a excepción de las palabras de cabecera, que tendrán todas el mismo tamaño.

La función *pintaNumeros* dividirá en dígitos los números que se le pasen, cada dígito nuevamente será representado en pantalla con la función *LCD\_PutChar*.

En la página 7 del anexo se encuentra una representación de lo que se muestra en la pantalla de cada menú, en la página 8, las acciones que se realizan en cada estado.

## 4.2 Variables y funciones utilizadas

---

### 4.2.1 Variables globales

---

Nombre	Tipo	Rango	Descripción
<b>nodos[64]</b>	int1	0 - 1	Vector de bits de 64 posiciones. Un 1 en la posición "i" significa que en la instalación existe un nodo con la dirección "i".
<b>v_grupos[16]</b>	int1	0 - 1	Guarda el estado de pertenencia de un nodo a los grupos. Se modifica en el estado 4120.
<b>v_nodos[64]</b>	int1	0 - 1	Guarda el estado de qué nodos pertenecen a un grupo. Se modifica en el estado 5130
<b>scroll</b>	unsigned int16	0 - 65	Contador que indica cuantas veces se ha movido el cursor a opciones que no aparecen en pantalla, aumenta cuando se mueve hacia abajo y disminuye hacia arriba.

Tabla 6. Lista de variables globales – Parte I

Nombre	Tipo	Rango	Descripción
<b>pos_cursor</b>	unsigned int16	0 - 144	Valor de la fila en la que se empieza a representar el cursor
<b>del_cursor</b>	unsigned int16	0 - 144	Valor de la fila anterior en la que se representó el cursor, se utiliza para borrarlo y crear un efecto de movimiento
<b>aux_scroll</b>	unsigend int8	0 - 65	En esta variable se guarda el valor que ha tomado <i>scroll</i> en el estado 40 de la máquina de estados de <i>pintaNodo</i> , se corresponde con la elección del nodo que se desea configurar.
<b>aux_scroll_1</b>	unsigned int8	0 - 17	En esta variable se guarda el valor que ha tomado <i>scroll</i> en el estado 4120 de la máquina de estados de <i>pintaNodo</i> , se corresponde con la elección del grupo al que se desea agregar el nodo escogido.
<b>aux_scroll_2</b>	unsigned int8	0 - 17	En esta variable se guarda el valor que ha tomado <i>scroll</i> en el estado 50 de la máquina de estados de <i>pintaNodo</i> , se corresponde con la elección del grupo que se desea configurar.
<b>aux_scroll_3</b>	unsigned int8	0 - 64	En esta variable se guarda el valor que ha tomado <i>scroll</i> en el estado 5130 de la máquina de estados de <i>pintaNodo</i> , se corresponde con la elección del nodo que se desea agregar al grupo escogido.
<b>aux_scroll_4</b>	unsigned int8	0 - 17	En esta variable se guarda el valor que ha tomado <i>scroll</i> en los estados 4130 y 4165 de la máquina de estados de <i>pintaNodo</i> , se corresponde con la elección de la escena que se desea configurar o establecer.
<b>segunda_parte</b>	int1	0 - 1	Bit que establece si hay que enviar la primera o la segunda trama de un mensaje.
<b>envio</b>	int1	0 - 1	Bit que se pone a 1 si la orden que se desea enviar consta de una trama.
<b>envio_2</b>	int1	0 - 1	Bit que se pone a 1 si la orden que se desea enviar consta de dos tramas.
<b>tx[4]</b>	char	0x00 - 0xFF	Vector en el que se guarda la trama de envío.
<b>tx_2[4]</b>	char	0x00 - 0xFF	Vector en el que se guarda la segunda trama de envío en caso de necesitarse.
<b>rx[4]</b>	char	0x00 - 0xFF	Vector en el que se guarda la trama de recepción
<b>cadena[14]</b>	char	0x1C - 0x5A	Vector que almacena las cadenas de caracteres de las opciones del menú.

Tabla 7. Lista de variables globales – Parte II

Nombre	Tipo	Rango	Descripción
<b>lista_nodos_activos[64]</b>	unsigned int	0 - 63	Vector que guarda de forma ordenada las direcciones de los nodos conectados en la instalación.
<b>i_nodo</b>	unsigned int	0 - 63	Indice que se utiliza para recorrer el vector <i>nodos</i> y navegar por los nodos activos en el estado 40.
<b>i_grupo</b>	unsigned int	0 - 63	Indice que se utiliza para recorrer el vector <i>v_nodos</i> y navegar por los nodos activos en el estado 5130.
<b>contador_menu</b>	unsigned int	1 - 4	Variable que almacena en qué posición del menú visible se encuentra el cursor.
<b>Opciones_menu</b>	Unsigned int	1 - 66	Propiedad de cada menú que indica de cuantas opciones dispone.
<b>nodo</b>	Unsigned int	0 - 127	Dirección del nodo en formato DALI.
<b>Nodo_real</b>	Unsigned int	0 - 63	Dirección del nodo según números cardinales
<b>grupo</b>	Unsigned int	128 - 161	Dirección del grupo en formato DALI.
<b>hit</b>	Unsigned int	0 - 63	Contador de recepciones correctas cuando se escanean nodos en la instalación.
<b>dir</b>	Signed int16	0 - 161	Variable auxiliar que almacena el valor de <i>nodo</i> o <i>grupo</i> según convenga.
<b>fila</b>	Signed int16		Almacena el valor de la fila de cada opción de los menús, que en función de su valor se decide si se representa en pantalla.
<b>valor_sup</b>	Unsigned int	0 - 100	Valor porcentual de la intensidad de iluminación que el balastro guarda en valor superior.
<b>Valor_inf</b>	Unsigned int	0 - 100	Valor porcentual de la intensidad de iluminación que el balastro guarda en valor inferior.
<b>Valor_escena</b>	Unsigend int	0 - 100	Valor porcentual de la intensidad de iluminación que el balastro guarda en la escena que se escoja.
<b>Cbuff[BUF_SIZE]</b>	char	0x00 - 0xFF	Buffer que almacena los datos recibidos por comunicación serie.
<b>Rx_next</b>	Unsigned int	0 - 4	Indice que se utiliza para recorrer el buffer.
<b>Rx_read</b>	Unsignedint	0	Indica que ya no hay más valores que leer del buffer.
<b>salvapantallas</b>	Unsigned int	0 - 60	Variable que se incrementa mediante un temporizador, que, al llegar a su valor máximo, se representa una imagen de Bes Ingenium

Tabla 8. Lista de variables globales – Parte III

Nombre	Tipo	Rango	Descripción
<b>Tipo_dir</b>	int1	0 - 1	Al compartirse algún estado independientemente de si se viene desde el menú de grupos o del menú de nodos, con esta variable permite saber la procedencia.
<b>redireccion</b>	int1	0 - 1	Establece una acción inicial de borrado de pantalla en el estado 40. Se modifica tras haber acabado un escaneo de la instalación.
<b>limite</b>	int1	0 - 1	Flag que bloquea ciertas opciones si se escoge transmitir una trama broadcast, permite modificar el color del menú para simbolizar esto.
<b>Flag_espera</b>	int1	0 - 1	Flag que comunica si el programa se encuentra en un estado de espera.
<b>botonOK</b>	int1	0 - 1	Flag que se pone a 1 al inicio de la interrupción del pulsador central y a 0 al final, se utiliza para comunicar a la función <i>Clear_LCD</i> cuál es el área de borrado.
<b>bloqueo</b>	int1	0 - 1	Flag complementario a limite que identifica cuando se trata de una trama broadcast.
<b>recibido</b>	int1	0 - 1	Flag que se pone a 1 cuando se recibe un ACK en rx[0].
<b>cabecera</b>	int1	0 - 1	Flag que identifica cuando <i>cadena</i> contiene un rótulo o una opción del menú.
<b>Condicion_menu</b>	int1	0 - 1	Propiedad de los menús que identifica si su número de opciones visibles es 1 o 4.
<b>RB5_anterior</b>	int1	0 - 1	Guarda el estado del pulsador situado en el pin 5 del puerto B a la salida de la interrupción del puerto B. Se utiliza como memoria del su estado anterior.
<b>RB6_anterior</b>	int1	0 - 1	Guarda el estado del pulsador situado en el pin 6 del puerto B a la salida de la interrupción del puerto B. Se utiliza como memoria del su estado anterior.
<b>RB7_anterior</b>	int1	0 - 1	Guarda el estado del pulsador situado en el pin 7 del puerto B a la salida de la interrupción del puerto B. Se utiliza como memoria del su estado anterior.
<b>RB5_actual</b>	int1	0 - 1	Guarda el estado del pulsador situado en el pin 5 del puerto B a la entrada de la interrupción del puerto B. Se utiliza como memoria del su estado actual.

Tabla 9. Lista de variables globales – Parte IV

Nombre	Tipo	Rango	Descripción
<b>RB6_actual</b>	int1	0 - 1	Guarda el estado del pulsador situado en el pin 6 del puerto B a la entrada de la interrupción del puerto B. Se utiliza como memoria del su estado actual.
<b>RB7_actual</b>	int1	0 - 1	Guarda el estado del pulsador situado en el pin 7 del puerto B a la entrada de la interrupción del puerto B. Se utiliza como memoria del su estado actual.
<b>Init_cmd[18]</b>	Const unsigned char	0x00 - 0xFF	Vector que almacena los comandos de inicialización de la pantalla.
<b>Init_data[62]</b>	Const unsigned char	0x00 - 0xFF	Vector que almacena los datos de inicialización de la pantalla.
<b>Init_data_count</b>	Const unsigned char	1- 15	Vector que almacena el valor en el que se agrupan los datos de <i>init_data</i> que se tienen que enviar para la inicialización de la pantalla.

Tabla 10. Lista de variables globales – Parte V

#### 4.2.2 Funciones

---

#### CONFIGURACION\_PIC

---

Entradas	Salida
<b>void</b>	<b>void</b>

Tabla 11. Variables de entrada y salida de la función “configuracion\_pic”.

Esta función configura los parámetros de funcionamiento del PIC, siendo estos: la frecuencia de reloj, la salida de una señal de ancho de pulso modulada (PWM) que se controla a través de una temporización con el *TIMER 2*, la comunicación a través de una interfaz serie a un periférico (SPI), que en este caso se trata de una pantalla TFT de 2,2” con una resolución de 240 x 320 píxeles, tres interrupciones por detección de flanco (*INT\_EXT*, *INT\_EXT1*, *INT\_EXT2*) asociadas a sendos pulsadores, una temporización de 2,1 segundos con el *TIMER 0* y por último, una interrupción por recepción de dato (Rx) por el puerto serie (RDA) conectada al pin asociado al envío de datos (Tx) de otro PIC.

Por otro lado, se preparan tres salidas para el envío de datos a la pantalla colocando a nivel alto los pines 4 y 5 del puerto A, y el pin 6 del puerto B.

Sólo se realiza la llamada a esta función una vez al principio durante la ejecución del programa.

---

## LCD\_COMMAND

---

Entrada	Tipo	Descripción	Salida
comando	int8	Valor hexadecimal del comando	void

Tabla 12. Variables de entrada y salida de la función "LCD\_command".

Esa función se utiliza para el envío de comandos de configuración a la pantalla. Estos comandos son datos de 8 bits en formato hexadecimal, con ellos se puede definir la orientación, horizontal o vertical, de la pantalla o el área de representación entre otras cosas. Para ello coloca en primer lugar a nivel bajo el pin 5 del puerto A y luego envía el comando con la función *spi\_write* disponible a través de la librería del microcontrolador.

---

## LCD\_DATA

---

Entrada	Tipo	Descripción	Salida
dato	int8	Valor hexadecimal del dato	void

Tabla 13. Variables de entrada y salida de la función "LCD\_Data".

Esa función se utiliza para el envío de datos a la pantalla. Estos datos son de 8 bits en formato hexadecimal, con ellos se puede definir el color del píxel al que se apunta con la función *LCD\_SetPos*. Para ello coloca en primer lugar a nivel alto el pin 5 del puerto A y luego envía el comando con la función *spi\_write* disponible a través de la librería del microcontrolador.

---

## LCD\_SETPOS

---

Entrada	Tipo	Descripción	Salida
x1	int16	Valor inicio de columna	void
x2	int16	Valor fin de columna	
y1	int16	Valor inicio fila	
y2	int16	Valor fin fila	

Tabla 14. Variables de entrada y salida de la función "LCD\_SetPos".

Mediante llamadas a las funciones *LCD\_Command* y *LCD\_Data*, esta función define el área de representación de los próximos datos que se envíen con la función descrita anteriormente. Las entradas de *LCD\_SetPos* son las coordenadas de los vértices de un polígono rectangular tomando como referencia el píxel situado en la esquina inferior izquierda cuya coordenada es (0, 0).

## LCD\_PUTCHAR

Entrada	Tipo	Descripción	Salida
<b>x</b>	int16	Coordenada en el eje horizontal	<b>void</b>
<b>y</b>	int16	Coordenada en el eje vertical	
<b>c</b>	char	Carácter que se desea representar	
<b>fColor</b>	int16	Color de la fuente	
<b>bColor</b>	int16	Color de fondo	
<b>tam</b>	int8	Factor de escala	
<b>tipo</b>	int1	Área en que se puede representar	

Tabla 15. Variables de entrada y salida de la función "LCD\_PutChar".

## Variables locales

Nombre	Tipo	Rango	Descripción
<b>offset</b>	unsigned int16	0x0000 - 0x03E7	Sitúa un puntero en la tabla de caracteres ASCII apuntando a la primera posición del carácter.
<b>i</b>	int16	0 - 16	Contador que recorre las filas del objeto que se va a representar.
<b>j</b>	int16	0 - 8	Contador que recorre las columnas del objeto que se va a representar.
<b>recC_LCD</b>	int	0 - 8	Contador que recorre las filas del área de la pantalla que se va a representar.
<b>recF_LCD</b>	int	0 - 8	Contador que recorre las columnas del área de la pantalla que se va a representar.
<b>min</b>	int	0 - 8	Valor de límite inferior de la fila en la que se puede representar.
<b>range</b>	int	16 - 48	Número de filas que se van a recorrer de los objetos que se van a representar.
<b>m</b>	unsigned char	0x0000 - 0x03F0	Posición de la tabla de caracteres correspondiente a un píxel concreto.
<b>comparador</b>	unsigned char	0x1C	Punto de partida de la tabla de caracteres.
<b>aux_c</b>	unsigned char	0x1C - 0x5A	Variable que guarda el valor hexadecimal del carácter ASCII.

Tabla 16. Variables locales de la función "LCD\_PutChar".

Se parte de tres tablas de caracteres ASCII de tres tamaños de caracteres diferentes, de 8x 16 píxeles, de 16x32 píxeles y de 24x48 píxeles. Las dos primeras parten desde el carácter en la posición 0x25 en hexadecimal, que se corresponde con el carácter '%', mientras que la tercera, la mayor, parte de la posición 0x2C, es decir, el carácter '+', llegando hasta la posición 0x7A, carácter 'z', en el caso de la tabla de caracteres más pequeños, y en los otros dos hasta la posición 0x5A, correspondiente al carácter 'Z'. La decisión ha sido tomada en base al uso de la memoria del microcontrolador, ya que el resto de los caracteres no han sido incluidos debido a que no se iban a utilizar.

Estas tablas están constituidas como un vector de bytes, en el que cada byte es una fila de píxeles. Cada carácter está definido como tantas filas de bytes como alto se haya definido, y tantos segmentos de bytes

como sean necesarios para representar el ancho del carácter, en el que un 0 se corresponde con el fondo y un 1 con el objeto de que se desea representar, éstas se van recorriendo con la variable *offset*.

La función *LCD\_PutChar* representa en pantalla el carácter ASCII que se le pasa a través de la variable *c*, con el factor de escala escogido en *tam*, en las coordenadas *x*, *y* con del color *fColor* con un fondo de color *bColor*, estos dos últimos parámetros tienen un rango desde el valor 0x0000 al 0xFFFF, correspondiente a un sistema de coordenadas RGB + Luminosidad, en el que cada 4 bits es cada uno de esos canales de color.

Se define *tipo* como un variable de 1 bit que se utiliza para separar en dos regiones la pantalla, de modo que, si su valor es 1, se corresponderá a un carácter del menú por el que se puede navegar, mientras que si es 0 será un carácter correspondiente al rótulo del menú. Con esto se consigue que caracteres del menú no se superpongan a los del rótulo ya que se impide su representación.

Mediante una llamada a la función *LCD\_SetPos* se sitúa el área de representado desde las coordenadas *x*, *y* hasta el número de píxeles con los que representa cada carácter. A continuación, se recorre el carácter ASCII desde la posición 0 a la 8 en columnas y desde la posición 0 hasta *range* en filas, y por cada bit, se recorre el área de representación desde 0 hasta *tam* tanto en filas como columnas escribiendo en la pantalla en color de ese bit, si es fondo u objeto. *Range* cambia su valor en función del tamaño del carácter que se va a representar. Esto se repite para cada segmento del carácter.

Existe un caso especial, se requería introducir una cuarta tabla de caracteres del 0 al 9 de tamaño 48x96, pero se carecía de suficientes recursos de memoria como para albergarla, por lo que se cogió la tabla de los caracteres de mayor tamaño y se duplicaron sus dimensiones por código.

---

#### LCD\_PUTSIM

---

Entrada	Tipo	Descripción	Salida
<b>x</b>	int16	Coordenada en el eje horizontal	<b>void</b>
<b>y</b>	int16	Coordenada en el eje vertical	
<b>c</b>	char	Carácter que se desea representar	
<b>fColor</b>	int16	Color de la fuente	
<b>bColor</b>	int16	Color de fondo	
<b>tipo</b>	Int1	Área en que se puede representar	

Tabla 17. Variables de entrada y salida de la función "LCD\_PutSim".

## Variables locales

Nombre	Tipo	Rango	Descripción
<b>aux_x</b>	unsigned int16	0 – 320	Variable que guarda el valor de la coordenada x para operar con él sin alterar el valor de x.
<b>k</b>	Unsigned int16	0 – 7	Contador del segmento del carácter.
<b>m</b>	unsigned char	0x0000 - 0x03F0	Posición de la tabla de caracteres correspondiente a un píxel concreto.
<b>comparador</b>	unsigned char	0x1C	Punto de partida de la tabla de caracteres.
<b>aux_c</b>	unsigned int16	0x1C - 0x5A	Variable que guarda el valor hexadecimal del carácter ASCII.
<b>offset</b>	unsigned int16	0x0000 - 0x03E7	Sitúa un puntero en la tabla de caracteres ASCII apuntando a la primera posición del carácter.
<b>i</b>	unsigned int16	0 – 80	Contador que recorre las filas del objeto que se va a representar.
<b>j</b>	unsigned int16	0 – 8	Contador que recorre las columnas del objeto que se va a representar.

Tabla 18. Variables locales de la función "LCD\_PutSim".

Para este caso, los caracteres son cogidos de una tabla independiente a la ASCII que contienen flechas de diferentes formas. La forma de representación en pantalla es la misma que en la función anterior. Estos caracteres tienen un tamaño de 24x48 píxeles.

## LCD\_PUTCHAR56X80

Entrada	Tipo	Descripción	Salida
<b>x</b>	int16	Coordenada en el eje horizontal	<b>void</b>
<b>y</b>	int16	Coordenada en el eje vertical	
<b>c</b>	char	Carácter que se desea representar	
<b>fColor</b>	int16	Color de la fuente	
<b>bColor</b>	int16	Color de fondo	

Tabla 19. Variables de entrada y salida de la función "LCD\_PutChar56x80".

## Variables locales

Nombre	Tipo	Rango	Descripción
<b>offset</b>	unsigned int16	0x0000 - 0x03E7	Sitúa un puntero en la tabla de caracteres ASCII apuntando a la primera posición del carácter.
<b>i</b>	unsigned int16	0 – 80	Contador que recorre las filas del objeto que se va a representar.
<b>j</b>	unsigned int16	0 – 8	Contador que recorre las columnas del objeto que se va a representar.
<b>aux_x</b>	unsigned int16	0 – 320	Variable que guarda el valor de la coordenada x para operar con él sin alterar el valor de x.
<b>k</b>	Unsigned int16	0 – 7	Contador del segmento del carácter.
<b>m</b>	unsigned char	0x0000 - 0x03F0	Posición de la tabla de caracteres correspondiente a un píxel concreto.
<b>comparador</b>	unsigned char	0x1C	Punto de partida de la tabla de caracteres.
<b>aux_c</b>	unsigned int16	0x1C - 0x5A	Variable que guarda el valor hexadecimal del carácter ASCII.

Tabla 20. Variables locales de la función "LCD\_PutChar56x80".

Su uso es el mismo que el de la función *LCD\_PutChar*, pero en este caso, se toman los caracteres desde otra tabla puesto que sólo se desea a ese tamaño tres caracteres en concreto. Estos caracteres tienen también la característica de estar formados por 7 segmentos de 80 bytes. Para su representación cada segmento se representa a la derecha del anterior formando la figura del carácter, esto se realiza con el contador *k* que apunta al inicio de cada segmento y recorriendo cada uno con los contadores *i*, *j*, y representando los píxeles al igual que en la función anterior

## CLEAR\_LCD

Entrada	Tipo	Descripción	Salida
<b>color</b>	int16	Valor hexadecimal de un color en coordenadas RGB + Luminosidad	<b>void</b>

Tabla 21. Variables de entrada y salida de la función "Clear\_LCD".

## Variables locales

Nombre	Tipo	Rango	Descripción
<b>i</b>	int16	0 - 240	Contador que recorre las filas de la pantalla
<b>j</b>	int16	0 - 320	Contador que recorre las columnas de la pantalla.
<b>d1</b>	int8	0x00 - 0xFF	Primer byte del color.
<b>d2</b>	int8	0x00 - 0xFF	Segundo byte del color.
<b>max</b>	int8	199 - 239	Valor de límite superior de la fila a partir de la cual puede representar.

Tabla 22. Variables locales de la función "Clear\_LCD".

Esta función se utiliza para borrar el contenido de la pantalla escribiendo en cada píxel el mismo color, en este caso se representa el color negro, que se hace coincidir con el valor de fondo de los caracteres para una representación clara de la interfaz.

Con los contadores  $i, j$  se recorren los píxeles de la pantalla coloreándolos con el color formado por los bytes  $d1$  y  $d2$ , que se han de dividir porque la función de envío por el puerto serie solo admite un byte. Se parte desde la fila superior definida por  $max$  hasta la inferior de valor 0. Se utiliza esta variable para decidir si se borra sólo el menú o también el rótulo si se empieza en la parte más alta.

---

#### INIT\_LCD

---

Entradas	Salida
void	void

Tabla 23. Variables de entrada y salida de la función "Init\_LCD".

Con esta función se inicializa la configuración de la pantalla mediante la utilización de las funciones *LCD\_Command* y *LCD\_Data*. Tanto los datos como los comandos son valores fijos almacenados en los vectores *init\_cmd* y *init\_data*, este último tiene los datos agrupados e igualmente se envían por el SPI según se defina el grupo por el valor que se encuentre en *init\_data\_count*.

Sólo se realiza la llamada a esta función una vez al principio durante la ejecución del programa.

---

#### PINTAMENU

---

Entrada	Tipo	Descripción	Salida
escena	unsigned int16	Estado de la máquina de estados que define el menú que se debe representar en la pantalla y qué características tiene dicho estado.	void

Tabla 24. Variables de entrada y salida de la función "pintaMenu".

Con esta función se realiza la representación de los menús de la interfaz, la navegación se realiza mediante la actuación sobre tres pulsadores, dos de selección y uno de confirmación.

La representación de los menús se realiza mediante llamadas a las funciones *cadena* y *StringToChar* según convenga, la primera función se utiliza en caso de tratarse una opción que se desee tener en dos idiomas, mientras que la segunda representa la opción independientemente del idioma.

Cada uno de los veintitrés estados guarda en sendas variable el valor del estado en el que se encuentra (*aux\_menu*), del estado anterior (*menu\_anterior*), de cuantas opciones tiene ese menú (*opciones\_menu*) y cuantas son el número máximo de opciones visibles del menú (*condicion\_menu*), aunque se trata de una variable binaria se asocia en las interrupciones por pulsación a la variable *max\_opciones* que es la que realmente establece el límite de opciones visibles.

En esta función también se escribe en un vector la trama que se desea enviar a la instalación para que realice una acción concreta, y además se puede seleccionar el idioma del menú escribiendo en un registro de la *EEPROM* del microcontrolador el valor 0 para el idioma español o el valor 1 para el inglés. Se ha decido guardarlo en memoria no volátil para evitar que se pierda esta información en caso de falta de alimentación en la pasarela.

---

## PINTAROTULONODOS

---

Entradas	Salida
void	void

*Tabla 25. Variables de entrada y salida de la función "pintaRotuloNodos".*

Por motivos de ahorro de memoria se decidió crear esta función para escribir los rótulos de los distintos menús, ya que interesaba mantener los mismos caracteres el profundizar en la máquina de estados, permitiendo saber en todo momento cual fue la opción escogida para configurar los balastros de la instalación.

---

## INICBUFF

---

Entradas	Salida
void	void

*Tabla 26. Variables de entrada y salida de la función "inicbuff".*

Reinicializa con ceros todas las posiciones del buffer de entrada de datos del microcontrolador, que le llegan desde el PIC situado en la pasarela.

---

## PINTANODOSACTIVOS

---

Entradas	Salida
void	void

*Tabla 27. Variables de entrada y salida de la función "pintaNodosActivos".*

Tras realizar un escaneo a la instalación se conocen las direcciones de los nodos instalados, esta información se guarda en un vector de 64 posiciones que se aprovecha en esta función para descartar las direcciones no existentes y navegar solamente por las que han dado respuesta durante el escaneo.

Esta función es llamada en la máquina de estados cuando ésta entra en los estados parametrizados como 40 y 5130, lo que se representa en estos casos es la cifra correspondiente a la dirección del nodo, mediante una llamada a la función *pintaNumeros* que se explicará más adelante en este apartado, y un rótulo indicando desde dónde viene la máquina de estados.

En el caso del estado 40, se puede escoger una opción más, se trata de enviar las órdenes que se deseen a la instalación completa en lugar en a un nodo o grupo concreto.

---

## PINTAESPERA

---

Entradas	Salida
void	void

Tabla 28. Variables de entrada y salida de la función "pintaEspera".

Por motivos de ahorro de memoria se decidió crear esta función para hacer un borrado de pantalla con la función *Clear\_LCD* y a continuación con *StringToChar* escribir en pantalla "ESPERE" en las coordenadas deseadas, ya que se utiliza en múltiples casos de la máquina de estados.

---

## PINTAGRUPOS

---

Entradas	Salida
void	void

Tabla 29. Variables de entrada y salida de la función "pintaGrupos".

Con un formato similar al de *pintaNodosActivos*, esta función representa en pantalla todos los grupos a los que se puede acceder según el estándar DALI, un total de 16 numerados del 0 al 15, con una frase distintiva en la que se podrá leer "ASIG GRUPO".

Esta función es llamada en la máquina de estados cuando ésta entra en los estados parametrizados como 4120 y 50, representando también en pantalla el rótulo correspondiente a la opción escogida anteriormente, en el caso del estado 4120 aparecerá escrito "NODO" y su dirección, en el estado 50 aparecerá "GRUPOS".

---

## PINTAESCENAS

---

Entradas	Salida
void	void

Tabla 30. Variables de entrada y salida de la función "pintaEscenas".

Comparte el mismo formato similar que *pintaEscenas*, esta función representa en pantalla todas las escenas a los que se puede acceder según el estándar DALI, un total de 16 numerados del 0 al 15, con una frase distintiva en la que se podrá leer "ASIG ESCENA". Las escenas son las posiciones de memoria de un nodo en las que se pueden guardar diferentes niveles de iluminación, su rango de valores va del 0 al 255, que se corresponde con apagado y máxima iluminación respectivamente.

---

## PINTARETORNO

---

Entrada	Tipo	Descripción	Salida
<b>col</b>	int16	Coordenada del eje horizontal en el que se representará el símbolo de volver al menú anterior.	
<b>fila</b>	int	Coordenada del eje vertical en el que se representará el símbolo de volver al menú anterior.	<b>void</b>
<b>tam</b>	int	Factor de escala para el símbolo de retorno.	

Tabla 31. Variables de entrada y salida de la función "pintaRetorno".

Esta función se llama cuando se desea representar un símbolo para representar la opción de retorno al menú anterior. Se le pasan los parámetros *col* y *fila* para las coordenadas donde se desee colocar y *tam* para el factor de escala. Con estos parámetros se llamará a la función *LCD\_PutChar* para que coloque el símbolo.

---

## PINTANIVEL

---

Entrada	Tipo	Descripción	Salida
<b>nivel</b>	int16	Valor decimal del nivel de iluminación.	<b>void</b>

Tabla 32. Variables de entrada y salida de la función "pintaNivel".

Esta función es llamada en los casos 4155, 4160 y 4170 de la máquina de estados de *pintaMenu*, que se corresponden respectivamente a la selección de nivel de iluminación máximo y mínimo de nodos o grupos, y de las escenas de un nodo concreto. Con ella se representa en pantalla un número de tres cifras a la izquierda de la pantalla mediante una llamada a *pintaNumeros*, y los símbolos "+", "-" y el de retorno en columna a la derecha de la pantalla haciéndose coincidir en altura con los respectivos pulsadores que se han de accionar para subir o bajar el nivel o volver al menú anterior.

---

## PINTANUMEROS

---

Entrada	Tipo	Descripción	Salida
<b>x</b>	int16	Coordenada en el eje horizontal.	
<b>y</b>	int16	Coordenada en el eje vertical.	
<b>numero</b>	int16	Carácter que se desea representar.	
<b>fColor</b>	int16	Color de la fuente.	<b>void</b>
<b>bColor</b>	int16	Color de fondo.	
<b>tam</b>	int8	Factor de escala.	
<b>tipo</b>	int1	Área en que se puede representar.	
<b>digitos</b>	int	Cantidad de dígitos que tiene <i>número</i> .	

Tabla 33. Variables de entrada y salida de la función "pintaNumeros".

## Variables locales

Nombre	Tipo	Rango	Descripción
<b>cent</b>	int8	0 - 1	Dígito de las centenas
<b>dec</b>	int8	0 - 9	Dígito de las decenas.
<b>unit</b>	int8	0 - 9	Dígito de las unidades.

Tabla 34. Variables locales de la función "pintaNumeros".

Esta función recibe como entrada un valor entero positivo que lo descompone en dígitos para su representación en pantalla mediante la función *LCD\_PutChar*.

Se utiliza en los estados 40 y 5130 para representar la dirección del nodo que se desea seleccionar, en los estados 50 y 4120 para los grupos y en los estados 4130 y 4165 para las escenas, en todos los casos siendo números de dos cifras. En el caso de los estados 4155, 4160 y 4170 se representa el valor de iluminación con tres dígitos.

## STRINGTOCHAR

Entrada	Tipo	Descripción	Salida
<b>cadena</b>	char*	Cadena de caracteres que se desea representar en pantalla.	
<b>x</b>	int	Coordenada en el eje horizontal.	
<b>y</b>	int	Coordenada en el eje vertical.	<b>void</b>
<b>aux_tam</b>	int8	Factor de escala de las letras pequeñas.	
<b>tipo</b>	int1	Área en que se puede representar.	

Tabla 35. Variables de entrada y salida de la función "StringToChar".

## Variables locales

Nombre	Tipo	Rango	Descripción
<b>tam</b>	int	1 - aux_tam+1	Factor de escala de la letra que se va a representar.
<b>color</b>	int16	0x0000 - 0xFFFF	Color del carácter en coordenadas RGB + Luminosidad.

Tabla 36. Variables locales de la función "StringToChar".

Con esta función se recorre la cadena de caracteres que le entra por *cadena*, y la divide en caracteres individuales.

En condiciones normales, la primera letra de la palabra la pondrá de un tamaño una unidad mayor al que se le pasa por *aux\_tam* y lo guarda en *tam*, posteriormente el valor de esta variable se cambia al de *aux\_tam* para que las siguientes letras sean de un tamaño inferior. Cuenta con detección de espacio en blanco, que, en caso de ocurrir, *tam* volvería a ser una unidad mayor que *aux\_tam* para representar la letra que inicia la palabra más grande.

Esto no ocurre en caso de especificarse que la cadena pertenece a un rótulo si se da la condición de que *cabecera* es verdadera, que todas las letras tendrían el mismo tamaño.

---

## LINEA\_LCD

---

Entrada	Tipo	Descripción	Salida
<b>x</b>	int16	Coordenada en el eje horizontal.	<b>void</b>
<b>y</b>	int	Coordenada en el eje vertical.	
<b>largo</b>	int16	Longitud de la línea en píxeles.	
<b>alto</b>	int1	Ancho de la línea en píxeles	
<b>color</b>	int16	Color de la línea.	

Tabla 37. Variables de entrada y salida de la función "Linea\_LCD".

### Variables locales

---

Nombre	Tipo	Rango	Descripción
<b>i</b>	int16	0 - 320	Contador de píxeles en horizontal que se van a representar.
<b>j</b>	int16	0 - 2	Contador de píxeles en vertical que se van a representar.

Tabla 38. Variables locales de la función "Linea\_LCD".

Función meramente estética, representa una línea en este caso de color naranja (0xFB00) en la parte superior de la pantalla que separa el rótulo del menú navegable.

Con *x*, *y* se selecciona la coordenada de inicio de la línea y con *largo* y *alto* sus dimensiones. Con dos bucles anidados se recorre desde el inicio al límite los píxeles incrementando las variables *i*, *j*.

---

## CADENAS

---

Entrada	Tipo	Descripción	Salida
<b>seleccion</b>	int	Valor con que se entra a una máquina de estados para representar información en pantalla.	<b>void</b>

Tabla 39. Variables de entrada y salida de la función "cadenas".

### Variables locales

---

Nombre	Tipo	Rango	Descripción
<b>idioma</b>	int	0 - 1	Valor que representa el idioma que en el que se desea la interfaz.

Tabla 40. Variables locales de la función "cadenas".

Si se desea que la información del menú varíe en función del idioma escogido (español o inglés), se leerá el valor de un registro de la EEPROM del microcontrolador que servirá como entrada a uno de los estados de esta máquina de estados separados por un *switch*, de valor 0 para el idioma español y 1 para el inglés.

Con *selección* se escoge la palabra que se desea representar independientemente del idioma, es decir si *selección* toma un valor por ejemplo de 10, se escribirá cierta palabra en la interfaz en español o su correspondiente en inglés dependiendo del valor de *idioma*.

### 4.2.3 Funciones de tratamiento de interrupción

---

#### TIMERO\_ISR

---

Entradas	Salida
void	void

Tabla 41. Variables de entrada y salida del tratamiento de interrupción del timer 0.

Es una interrupción temporizada que ocurre cada 2,1 segundos, como se ha programado en la configuración inicial. Incrementa en una unidad la variable *salvapantallas*, excepto si la pantalla se encuentra en el estado 7000, que se corresponde con la acción de representar el logotipo de “Bes-Ingenium”.

---

#### EXT\_ISR

---

Entradas	Salida
void	void

Tabla 42. Variables de entrada y salida del tratamiento de interrupción por cambio externo 0.

#### Variables locales

---

Nombre	Tipo	Rango	Descripción
Max_opciones	int	1 - 4	Define la cantidad opciones visibles en pantalla

Tabla 43. Variables locales del tratamiento de interrupción por cambio externo 0.

Se encarga de tratar la interrupción asociada a un flanco provocado por el pulsador inferior. Está siempre habilitado excepto durante el salvapantallas y accionarlo reinicia la variable que coloca la pantalla en ese estado.

Dependiendo de en qué estado se encuentre la pantalla realizará diferentes acciones. La más utilizada es la de navegación en los menús hacia las opciones situadas en posiciones inferiores, pero en los estados 4155, 4160 y 4170 se utiliza para disminuir de 5 en 5 unidades el valor en porcentaje del valor superior e inferior de iluminación y el de una escena específica respectivamente.

Durante la navegación, se realizará el efecto de movimiento de una flecha situada en la parte derecha, que se limitará hasta la fila de valor positivo más bajo, es decir 0. Este efecto se consigue con aumentando en una unidad la variable *contador\_menu* cada vez que se realiza una pulsación, tomando como valor 1 el de la opción visible superior y 4 como el valor de la opción visible inferior, si se deseara acceder a opciones situadas por debajo pero no representadas se hace uso de *scroll*, esta variable se incrementa si se acciona el pulsador y además el cursor está apuntando ya a la opción inferior, se calcula un nuevo valor de *fila* con lo que desaparece la opción superior, el menú visible se mueve una posición hacia arriba y aparece la opción oculta que estaba justo debajo. En ciertos estados resulta útil guardar el valor de esta variable para saber, por ejemplo, a qué dirección de nodo, grupo o escena se estaba apuntando.

---

EXT1\_ISR

---

Entradas	Salida
void	void

Tabla 44. Variables de entrada y salida del tratamiento de interrupción por cambio externo 1.

Variables locales

---

Nombre	Tipo	Rango	Descripción
pos_menu	int	1 - 65	Posición del cursor en términos absolutos, se calcula como la suma de <i>contador_menu</i> y <i>scroll</i>
actual	int	1 - 4153	Suma de <i>pos_menu</i> y <i>aux_menu</i> . Sirve para saber de forma combinada el estado y la posición absoluta del cursor.

Tabla 45. Variables local del tratamiento de interrupción por cambio externo 1.

Esta función trata la interrupción lanzada por un flanco en el pulsador central. Normalmente se utiliza para confirmar las selecciones, con lo que se progresa en la navegación del menú. La forma habitual de hacerlo es multiplicar por 10 el valor de *pos\_menu* para calcular el nuevo estado, existen ciertos casos en los que esto no se cumple, generalmente para evitar que el número asociado al estado crezca por encima del límite del tipo de variable. Para regresar al menú anterior, se ha de seleccionar dicha opción, identificada cuando *pos\_menu* es igual al número guardado en la variable *opciones\_menu* que contiene la cantidad de opciones que se puede seleccionar en un menú.

---

EXT2\_ISR

---

Entradas	Salida
void	void

Tabla 46. Variables de entrada y salida del tratamiento de interrupción por cambio externo 2.

Se encarga de tratar la interrupción asociada a un flanco provocado por el pulsador superior, por lo que su funcionalidad será similar a la función de tratamiento de interrupción del pulsador inferior, pero en este caso, servirá para navegar hacia las opciones superiores del menú, limitando la posición del cursor a la coordenada 144. Análogamente, decrementará una unidad *contador\_menu* y *scroll* en caso de cumplirse la condición. Igualmente modifica el valor porcentual de *valor\_sup*, *valor\_inf* y *valor\_escena* incrementándolo de 5 en 5 unidades.

---

RB\_ISR

---

Entradas	Salida
void	void

Tabla 47. Variables de entrada y salida del tratamiento de interrupción por cambio en el puerto B.

## Variables locales

Nombre	Tipo	Rango	Descripción
<b>Max_opciones</b>	int	1 - 4	Define la cantidad opciones visibles en pantalla
<b>pos_menu</b>	int	1 - 65	Posición del cursor en términos absolutos, se calcula como la suma de contador_menu y scroll
<b>actual</b>	int	1 - 4153	Suma de pos_menu y aux_menu. Sirve para saber de forma combinada el estado y la posición absoluta del cursor.

Tabla 48. Variables locales del tratamiento de interrupción por cambio en el puerto B.

Se encarga de tratar la interrupción asociada a un cambio en los pines del 4 al 7 del puerto B. Una acción en este puerto evita la entrada en el salvapantallas. Los pulsadores están repartidos de la siguiente forma: pulsador de selección arriba en el pin 5, pulsador de confirmación en el pin 6 y pulsador de selección abajo en el pin 7.

Al inicio de la interrupción lo primero que se comprueba es el estado de los tres pines, y sólo se opera con ellos en caso de que haya ocurrido un flanco de bajada en el pin correspondiente, esto se realiza guardando el estado de cada uno en la entrada anterior de la interrupción y comparándolo con el estado en el momento actual. Esto se realiza con *RBx\_anterior* y *RBx\_actual*, descritas en las variables globales, siendo *x* los pines 5, 6 y 7 según corresponda.

Respecto al pulsador de selección arriba, dependiendo de en qué estado se encuentre la pantalla realizará diferentes acciones. La más utilizada es la de navegación en los menús hacia las opciones situadas en posiciones inferiores, pero en los estados 4155, 4160 y 4170 se utiliza para disminuir de 5 en 5 unidades el valor en porcentaje del valor superior e inferior de iluminación y el de una escena específica respectivamente.

Durante la navegación se realizará el efecto de movimiento de una flecha situada en la parte derecha, que se limitará hasta la fila de valor positivo más bajo, es decir 0. Este efecto se consigue con aumentando en una unidad la variable *contador\_menu* cada vez que se realiza una pulsación, tomando como valor 1 el de la opción visible superior y 4 como el valor de la opción visible inferior, si se deseara acceder a opciones situadas por debajo pero no representadas se hace uso de *scroll*, esta variable se incrementa si se acciona el pulsador y además el cursor está apuntando ya a la opción inferior, se calcula un nuevo valor de *fila* con lo que desaparece la opción superior, el menú visible se mueve una posición hacia arriba y aparece la opción oculta que estaba justo debajo. En ciertos estados resulta útil guardar el valor de esta variable para saber, por ejemplo, a qué dirección de nodo, grupo o escena se estaba apuntando.

Con el pulsador de confirmación de selección se progresa en la navegación del menú. La forma habitual de hacerlo es multiplicar por 10 el valor de *pos\_menu* para calcular el nuevo estado, existen ciertos casos en los que esto no se cumple, generalmente para evitar que el número asociado al estado crezca por encima del límite del tipo de variable. Para regresar al menú anterior, se ha de seleccionar dicha opción, identificada cuando *pos\_menu* es igual al número de opciones que tiene el menú, guardado en la variable *opciones\_menu*.

La funcionalidad del pulsador de selección abajo será similar a la función de tratamiento de interrupción del pulsador arriba, pero en este caso, servirá para navegar hacia las opciones superiores del menú, limitando la posición del cursor a la coordenada 144. Análogamente, decrementará una unidad *contador\_menu* y *scroll* en caso de cumplirse la condición. Igualmente modifica el valor porcentual de *valor\_sup*, *valor\_inf* y *valor\_escena* incrementándolo de 5 en 5 unidades.

---

RDA\_ISR

---

Entradas	Salida
void	void

*Tabla 49. Variables de entrada y salida del tratamiento de interrupción por dato en el puerto de comunicaciones.*

Variables locales

---

Nombre	Tipo	Rango	Descripción
<b>Trama_rx</b>	int	0 - 1	Flag que indica cuándo hay que dejar de ignorar los ceros que entran por la recepción de la comunicación serie.

*Tabla 50. Variables locales del tratamiento de interrupción por dato en el puerto de comunicaciones.*

Esta función de interrupción se ejecuta cuando se recibe un dato por el puerto de comunicación serie del microcontrolador. Por lo general, se reciben ceros, pero en caso de enviar un mensaje por este puerto, se obtendrá un acuse de recibo (ACK) y una trama de mensaje que se almacenará en el vector *cbuff* que, una vez completado, se volcará su contenido en *rx*.

En la página 4 del anexo se detalla el funcionamiento de la rutina de interrupción por cambio en el puerto B del microcontrolador.

#### 4.2.4 Programa principal

---

MAIN

---

Entradas	Salida
void	void

*Tabla 51. Variables de entrada y salida del programa principal.*

Variables locales

---

Nombre	Tipo	Rango	Descripción
<b>g</b>	int	0 - 8	Contador que se utiliza para recorrer el vector <i>v_nodos</i> guardando en él en qué grupos se encuentra el nodo escogido.
<b>parp</b>	int	0 - 100	Se utiliza para hacer el efecto de parpadeo del símbolo asterisco durante la pantalla de espera.

*Tabla 52. Variables locales del programa principal.*

Se está ejecutando continuamente por un bucle infinito. En él se encuentra una máquina de estados usada para transmitir las tramas, y gobernada por las variables *envio*, *envio\_2* y *segunda\_parte*. Su estado inicial es con *envio* y *envio\_2* con valor 0, transimiendo ceros de forma continua por el puerto serie.

Si *envio* adquiere el valor 1, significa que se ha seleccionado una orden que consta de una única trama que se enviará, existe el caso especial del estado 30, que es en el que se escanea la instalación, que se estará enviando la misma trama hasta haber completado las 64 direcciones de nodos posibles existentes en una instalación.

En caso de que *envio\_2* adquiriera el valor 1 significará que la orden consta de dos tramas, la primera es enviará mientras *segunda\_parte* se encuentre a 0. En el momento en el que se reciba un ACK, ésta cambiará al valor 1 procediendo a enviarse la segunda trama.

En la página 2 del anexo se encuentra el diagrama de estados que describe el funcionamiento del programa principal.

## 5 Conclusiones

---

### 5.1 Desarrollo de la pantalla

---

Inicialmente, se desarrolló un producto que utilizaba una pantalla con una comunicación en paralelo con su microcontrolador. Este sistema permite una transmisión de datos de múltiples bytes simultáneamente, se traduce en una actualización de datos en pantalla lo suficientemente rápida como para que el ojo humano sólo note un parpadeo en el momento de la escritura. La programación del controlador de pantalla consiste en la creación de menús enlazados representados como una lista de opciones apiladas en vertical. Su funcionalidad es similar a la del producto final, careciendo del establecimiento de escenas e idiomas. Los pulsadores siempre actúan sobre el sistema de la misma forma, el superior sirve para mover el cursor de selección hacia arriba, el inferior hacia abajo y el central para confirmar la selección. La representación de los caracteres de los menús se realiza cogiendo un mapa de caracteres ASCII cargado en la memoria del microcontrolador de 8 píxeles de ancho y 16 de alto, aumentando por software el tamaño de su representación en pantalla. Este proceso provoca que los caracteres tengan espacios del color del fondo intercalados con los píxeles del objeto.

Para comprobar su correcto funcionamiento, se ha conectado la pasarela al banco de pruebas mediante el bus DALI y se han enviado tramas a través de él. Con un osciloscopio conectado al canal de transmisión de datos del microcontrolador hacia la pasarela se ha realizado una doble verificación, ya que inicialmente el banco de pruebas no respondía según se esperaba, llegando a dos hipótesis, la comunicación no se realizaba de forma correcta o las tramas no eran las correctas. De esta manera, se pueden observar los bits que forman las tramas de órdenes en el osciloscopio a la vez que el comportamiento del banco de pruebas, verificando de esta forma que la comunicación se está realizando y que las tramas enviadas son las adecuadas.

Alegando a un abaratamiento del coste de producción, se modificó el diseño de la pasarela cambiando la pantalla a otra que se comunica mediante dos hilos, es decir, por puerto serie. De este modo los bytes solo pueden ser transmitidos de uno en uno, ralentizando la actualización de pantalla respecto a la descrita anteriormente. El método de verificación de funcionamiento ha seguido siendo el mismo. En el microcontrolador de esta pantalla se ha programado el establecimiento de escenas y el cambio de idioma de los menús. Pero el aspecto es el mismo al del caso anterior, al igual que la funcionalidad de los pulsadores.

Una vez comprobado que todas las órdenes funcionaban correctamente en el banco de pruebas, se procedió a modificar la representación de datos en la pantalla. Creando de esta forma cinco tipos de representación: una lista de opciones, la enumeración de nodos y grupos, la selección de niveles de iluminación, la pantalla de espera y la de localización de nodos y grupos. A los tres primeros se les ha dotado, además, de un rótulo que indica el nodo o grupo que ha sido escogido para ser configurado.

La lista de opciones tiene el aspecto que se ha explicado, una sucesión en vertical de opciones por las que se puede navegar hacia arriba y abajo con los pulsadores superior e inferior respectivamente. En la enumeración, se representa un rótulo en la parte superior y un número de dos dígitos en el centro de la pantalla que se corresponderá con la dirección de un nodo o grupo, este número se podrá modificar

actuando sobre los pulsadores superior e inferior. La representación de la selección de niveles de iluminación es similar a la anterior, a excepción de que el número es de tres dígitos y además se indica que propiedad se está modificando: nivel máximo o mínimo de iluminación o el de una escena. En la pantalla de espera, se muestra en la fila central el rótulo “Espere” en letras mayúsculas junto con un asterisco que parpadea para que el usuario no perciba que la pasarela está operando. La pantalla de localización mostrará “Localizando” en el centro de la pantalla, pero sin animaciones, en este caso será la instalación la que mostrará una respuesta al usuario haciendo que el nodo o el grupo escogido se encienda y se apague repetidamente.

Mantener cargado el mapa de caracteres en la memoria del microcontrolador supone un uso de la ROM muy alto, llegando a ocupar junto con el código fuente hasta el 99% de su capacidad. Los caracteres representados, aunque legibles, exhibían un aspecto que no se consideró adecuado como presentación de un producto final, por lo que se tomó la decisión de escoger un microcontrolador con una memoria superior capaz de albergar más mapas de caracteres de distintos tamaños, el PIC 18F25K22, que posee el doble de memoria que el anterior. Esto propició la creación de un salvapantallas con un aspecto más comercial en el que se muestra el modelo del producto, que tipo de dispositivo es, el logo de la marca “Bes - Ingenium” y la referencia web. Se crearon caracteres específicos para el logo “Bes” con la tipografía propia de la marca. Los demás caracteres fueron creados partiendo de la fuente que se había utilizado, tomando cada carácter como una imagen, mediante funciones de MatLab se realizó una transformación de escala obteniendo así las nuevas fuentes que se cargarían en memoria.

## 5.2 Desarrollo del producto del catálogo de ETS 5

---

La elaboración del producto involucró los conocimientos adquiridos durante la creación de la programación de la pantalla y, requisitos de aspecto y funcionalidad tomados de los instaladores y comerciales.

En una primera etapa, basándose únicamente en criterios funcionales, se realizó una aproximación que permitía la parametrización de los nodos individuales y de grupos, la asignación de nodos a grupos y establecer hasta 3 escenas para un total de 5 nodos y 16 grupos, porque, como se ha descrito, el entorno actual limita a 255 el número de objetos de comunicación que se pueden crear para un dispositivo.

Posteriormente, se presentaron los resultados a aquellas personas que iban a usar la herramienta. A raíz de la consulta, la herramienta adquirió una interfaz que resultaba más intuitiva y manejable, similar a otras conocidas por los instaladores, y nuevas funcionalidades de las que se carecían en un principio. Como consecuencia, el programa adquirió escalabilidad, es decir, se posibilitó añadir tantos nodos como fueran necesarios con la herramienta de creación y presentarlos en la interfaz, ya que todos poseen las mismas propiedades, tal como se describe en el apartado 3.2. Manufacturer Tool.

Esta herramienta no ha podido ser puesta a prueba sobre la instalación, se carecía de la tabla de relaciones entre los parámetros y objetos con las posiciones de memoria de la pasarela, en la que se alojan las tramas de comunicación con los balastros. El método de verificación consistiría en conectar la pasarela al bus KNX y cargarle el programa con la herramienta *ETS 5*, de este modo se configurarían los nodos de la instalación. Finalmente, desde la pantalla, se podría manipular el banco de pruebas para examinar que se comporta como se espera.

### 5.3 Sugerencias de mejora

---

Como mejoras del proyecto se proponen tres cambios, uno referente al código programado en el microcontrolador de la pantalla, otro a la interfaz del ETS 5 y, en tercer lugar, otro referente al diseño del producto.

Actualmente, el envío de tramas se realiza mediante una máquina de estados que se ejecuta cíclicamente dentro del programa principal dentro de un bucle infinito. El microcontrolador PIC posee la característica de enviar datos mediante una interrupción, es decir, es capaz de enviar tramas sólo cuando se le requiera, lo que supone un ahorro energético. El método habitual de utilizar esta interrupción consiste en mantenerla deshabilitada hasta que llegue al microcontrolador un dato, en este momento, se ejecutará la interrupción por recepción de dato en la que se habilitará la de envío. Ésta se ejecuta en el momento en el que *buffer* de salida se encuentra vacío, estado en el que se encuentra desde el principio, entonces, la rutina de interrupción enviará la trama que corresponda y se deshabilitará, evitando que entre nuevamente al haberse vaciado el *buffer*.

Utilizando una versión posterior de *Manufacturer Tool*, la limitación de objetos de comunicación sería menos restrictiva, pudiendo así incluir los nodos restantes al catálogo del ETS 5.

Por último, la mejora que se propone va referida al diseño de la pasarela. Se trata de un producto de domótica capaz de configurar los parámetros de los dispositivos de una instalación que estén conectados al bus DALI y al bus KNX. Por lo que resultaría de gran utilidad que además fuese capaz de utilizar el protocolo TCP/IP para establecer comunicaciones con otros dispositivos, ya sea en la red local o a través de un router NAT. Posibilitaría al instalador usar una plataforma móvil, como puede ser una tablet, para realizar las configuraciones pertinentes desde cualquier lugar del edificio, o incluso realizar labores de telemantenimiento desde una oficina situada en otra localidad, evitando transportes innecesarios de técnicos.

## 6 Referencias

---

Beckhoff. "DALI – digital standard for room-related light management". Notas de aplicación DK9222-0810-0031.

KNX Conocimientos Básicos KNX. <[https://www.knx.org/wAssets/docs/downloads/Marketing/Flyers/KNX-Basics/KNX-Basics\\_es.pdf](https://www.knx.org/wAssets/docs/downloads/Marketing/Flyers/KNX-Basics/KNX-Basics_es.pdf)>

KNX Principios de funcionamiento. KNX. <[https://www.knx.org/wAssets/docs/downloads/Marketing/Flyers/KNX-System-Principles/KNX-System-Principles\\_es.pdf](https://www.knx.org/wAssets/docs/downloads/Marketing/Flyers/KNX-System-Principles/KNX-System-Principles_es.pdf)>

Martínez Pérez, Pascual. "Creación de un Master DALI. Control de una/varias luminarias basado en un microcontrolador Cortex M4". Trabajo Fin de Máster (Máster Universitario Ingeniería de Telecomunicación). Universidad Oberta de Catalunya, 2016.

National Electrical Manufacturers Association (NEMA). "Digital Addressable Lighting Interface (DALI) Control Devices Protocol PART 1". NEMA STANDARD PUBLICATION 243-2004. Borrador version 1.13. Virginia, Octubre 2004.

National Electrical Manufacturers Association (NEMA). "Digital Addressable Lighting Interface (DALI) Control Devices Protocol PART 2". NEMA STANDARD PUBLICATION 243-2004. Borrador version 1.13. Virginia, Octubre 2004.