



Gradu Amaierako Lana / Trabajo Fin de Grado
Ingeniaritza Elektronikoko Gradua / Grado en Ingeniería Electrónica

Diseño e implementación de un sistema de adquisición de datos basado en el Microcontrolador XMC4500

Egilea/Autor/a:
Patricia García Ferrín
Zuzendaria/Director/a:
Javier Echanobe Arias

Índice

0. Introducción y Objetivos	1
1. Introducción a los Microcontroladores	2
1.1. Introducción	2
1.1.1. Historia de los microcontroladores	2
1.1.2. Microcontroladores vs. Microprocesadores	2
1.2. Arquitectura	3
1.3. Aplicaciones	4
1.4. Principales fabricantes y productos	4
1.5. Familia XMC4000 de Infineon	5
2. Kit de Desarrollo <i>Hexagon Application Kit</i>	6
2.1. <i>Hexagon Application Kit</i>	6
2.2. Microcontrolador XMC4500	6
2.2.1. Subsistema CPU	6
2.2.2. Memorias <i>on-chip</i>	7
2.2.3. Periféricos de comunicaciones	8
2.2.4. Periféricos analógicos <i>frontend</i>	9
2.2.5. Periféricos de control industrial	9
2.2.6. Líneas de Entrada/Salida	10
2.2.7. Soporte para <i>debug on-chip</i>	10
2.3. CPU Board	10
2.3.1. CPU Board XMC4500 General Purpose (CPU_45A-V3)	11
2.4. Placas satélite	13
2.4.1. Automation I/O Card (AUT_ISO-V1)	13
2.4.2. Standard Human Machine Interface Card (HMI_OLED-V1)	14
2.4.3. Ethernet/CAN/RS485 Interface Card (COM_ETH-V1)	15
2.4.4. Placa de Extensión de Pines (UNI_EXT01-V2)	17
2.5. Entorno de desarrollo de software DAVE TM (Versión 4)	17
2.5.1. Composición de las APPs de DAVE TM	18
2.5.2. Programación mediante el entorno gráfico (GUI) de DAVE TM	18
3. Sistema de Adquisición de Parámetros Ambientales	23
3.1. Introducción	23
3.2. Sensores	23
3.2.1. Sensor de Temperatura (MCP9700A)	23
3.2.2. Sensor de Humedad (808H5V5)	24
3.2.3. Sensor de Luminosidad (LDR)	24
3.2.4. Sensor de Presencia (PIR)	25
3.3. Adquisición de datos digitales	28
3.3.1. Adquisición física de los datos: Canal Serie Síncrono	28
3.3.2. Tratamiento de los datos mediante Software	31
3.4. Adquisición de datos analógicos	31
3.4.1. Funcionamiento, configuración y conexionado	31
3.4.2. Tratamiento de los datos mediante Software	35
3.5. Interfaz de Usuario	36
3.5.1. Interfaz Humano-Máquina (HMI): Pulsadores Táctiles (<i>Touch Pads</i>)	36
3.5.2. Método de selección de opciones	36
3.6. Visualización de los datos	37
3.6.1. Navegación por los Menús	37
3.6.2. Display de Opciones para los Sensores de Temperatura y Humedad	40
3.6.3. Display de Opciones para el Sensor de Luminosidad	42
3.6.4. Display de Opciones para el Sensor de Presencia	43
3.7. Esquema general del programa	44
4. Conclusiones	46

0. Introducción y Objetivos

El objetivo de este trabajo es el desarrollo de una aplicación para el Microcontrolador XMC4500 de Infineon Technologies AG instalado en el Kit de Desarrollo *Hexagon Application Kit*, cuya finalidad sea la adquisición, procesamiento y representación de parámetros ambientales.

Se utilizarán los periféricos de comunicaciones, para adquirir los datos tanto digitales como analógicos de los sensores de Presencia, Temperatura, Humedad y Luminosidad de los que disponemos.

Por otra parte, se hará uso de la Interfaz Humano-Máquina de la que dispone el Kit de Desarrollo para representar los datos adquiridos y permitir la interacción con el usuario dando la posibilidad de cambiar la medida que se muestra por pantalla mediante los botones táctiles que incorpora la placa.

En el primer capítulo se hará una introducción a los microcontroladores. Partiendo desde la historia de estos dispositivos electrónicos, se hablará de las características más generales en cuanto a arquitectura interna, de sus múltiples aplicaciones, y se compararán características de diferentes modelos.

En el segundo capítulo se entrará a conocer en detalle las características del Microcontrolador XMC4500 y del kit de desarrollo *Hexagon Application Kit* con el cual se ha realizado este trabajo. Por otro lado, se dará a conocer el programa DAVE creado por la empresa Infineon para el desarrollo de aplicaciones para los microcontroladores de la familia XMC.

En el tercer capítulo se describirá el funcionamiento de las funcionalidades del microcontrolador necesarias para la adquisición, procesamiento y representación de los valores medidos. Además, se mostrarán las características de los sensores utilizados y el montaje del sistema.

En el cuarto y último capítulo, se aportarán unas conclusiones sobre el trabajo realizado.

1. Introducción a los Microcontroladores

1.1. Introducción

Un microcontrolador (MCU) es un circuito integrado programable que en su interior contiene, como mínimo, una unidad central de procesamiento (CPU), unidades de memoria (RAM y ROM), periféricos programables de entrada/salida. Estas partes están interconectadas dentro del microcontrolador, y en conjunto forman lo que se le conoce como microcomputadora. Por lo que en el caso de un microcontrolador se trata de una microcomputadora encapsulada en un circuito integrado [4] [20].

1.1.1. Historia de los microcontroladores

En el año 1969, un equipo de ingenieros japoneses de la compañía BUSICOM llegó a Estados Unidos buscando una idea para fabricar sus calculadoras con el menor número posible de circuitos integrados. Este proyecto se le propuso a Intel, donde dirigido por Marcian Hoff y más tarde apoyado por Federico Faggin desarrollarían el primer microprocesador (μ P) de la historia, el Intel 4004 en 1971. A este primer microprocesador que trabajaba con datos de 4 bits le seguirían el 8008 que ya trabajaba con 8 bits, y más adelante, el famoso 8080 y otros de diferentes compañías como el Motorola 6800 o el Z80 de Zilog [4].

Sin embargo, estos dispositivos requerían chips externos para implementar un sistema funcional, aumentando el precio del sistema completo. En 1974 se crearía el primer microcontrolador, llamado TMS 1000 y fabricado por Texas Instruments; el cual incluía además del microprocesador, memoria ROM, memoria RAM y un reloj integrado [20]. En 1977 se empezó a comercializar el Intel 8084 que llevaba integrados los elementos fundamentales para que pudiera ser llamado microcontrolador: memoria ROM, memoria RAM (en este caso de 64 bytes a 256 bytes) y una interfaz de entrada/salida [4].

En la década de los 80, el desarrollo de microprocesadores y microcontroladores tomó caminos diferentes. Mientras que los μ P evolucionan buscando solución al manejo de grandes volúmenes de información, los MCU buscan incorporar unidades funcionales con capacidades superiores de interacción con el medio físico en tiempo real y un mejor desempeño y robustez en aplicaciones industriales [4].

De ahí en adelante, la incorporación de memorias EEPROM que permitían el borrado eléctrico rápido de la memoria supuso un gran salto, sobre todo en reducción de costes. En 1993 la empresa Microchip lanzó al mercado el PIC16x84, un microcontrolador de 8 bits que ya incluía este tipo de memorias. Ese mismo año, Atmel da un paso más allá lanzando el primer microcontrolador con memoria *flash* [20].

Poco a poco se ha ido aumentando el tamaño de estas memorias y el número de periféricos integrados en los microcontroladores, y reduciendo cada vez más el consumo de los mismos.

1.1.2. Microcontroladores vs. Microprocesadores

La principal diferencia entre un microcontrolador y un microprocesador, es la cantidad de elementos integrados que posee un microcontrolador en comparación con un microprocesador.

El μ P es un circuito integrado que contiene la Unidad Central de Proceso o CPU, que el sistema encargado de interpretar las instrucciones de un programa informático mediante la realización de operaciones básicas aritmético-lógicas y de entrada/salida. A partir de ahí, el μ P es un sistema abierto, al que externamente se le pueden conectar los periféricos requeridos a través de los buses de direcciones, datos y control que saca hacia el exterior.

El MCU por el contrario se trata de un sistema cerrado que contiene un computador completo encapsulado, con sus periféricos de memoria y los módulos de entrada/salida integrados [14].

En la Tabla 1 se hace una comparativa de aspectos más específicos.

Microcontroladores	Microprocesadores
Los dispositivos genéricos son de 8bits, actualmente hay desarrollo de 16,32 y 64 bits.	Tienen mayores longitudes de palabra (16, 32, 64bits).
Incorpora en una misma pastilla capacidad de memoria y manejo de puertos.	Necesitan chips externos para apoyar las funciones de transferencia y almacenamiento.
La memoria es limitada aunque algunos permiten incorporar KB externos.	Manejo de mayor capacidad de memoria.
Tarjetas simples alojadas en espacios pequeños.	Gran volumen (físico) en su adecuación y funcionamiento.
Consumo de milivatios o unos pocos vatios. Es posible alimentarlo con baterías.	Consumo de potencia en cientos de vatios.
Dispositivos complejos de comunicación que permiten control en tiempo real.	Manejo de dispositivos periféricos informáticos.
Robustos e inmunes a ruidos industriales.	Susceptibles a ruido industrial.
Sistemas mínimos.	Sistemas informáticos.

Tabla 1: Principales diferencias entre los Microcontroladores y los Microprocesadores [4].

1.2. Arquitectura

Las arquitecturas que pueden encontrarse en la CPU de un microcontrolador son las siguientes:

- **Arquitectura de Von Neumann** [4]: En la arquitectura Von Neumann la CPU se conecta a una memoria principal única, generalmente del tipo RAM, donde se almacenan los datos y el programa. A estos se accede a través de un sistema de buses único, como son el bus de dirección, control y datos. El ancho del bus que comunica la memoria con la CPU determina el tamaño de la unidad de datos o instrucciones, por lo que para acceder a datos de mayor tamaño que este bus hará falta acceder a memoria en más de un ciclo de reloj.
- **Arquitectura Harvard** [4]: La arquitectura Harvard se caracteriza por tener separados los bloques de memoria de datos e instrucciones y acceder a ellos por buses independientes de dirección, datos y control. La independencia de buses permite tener accesos simultáneos e independientes a la memoria de datos e instrucciones. El contenido y longitud de las direcciones de memoria pueden ser distintos para los datos e instrucciones permitiendo una optimización en el uso de la memoria, ya que existe la posibilidad de superponer los tiempos de acceso al acceder a la memoria de programa y de datos en el mismo ciclo de reloj.
- **Arquitectura Reduced Instruction Set Computer (RISC)** [3]: Esta arquitectura tiene dos características principales:
 - Las instrucciones tienen un tamaño fijo y están presentadas en un reducido número de formatos.
 - Sólo las instrucciones de carga y almacenamiento acceden a la memoria de datos.

Esta estrategia de diseño de la CPU está basada en la idea de que un conjunto simple de instrucciones en combinación con una arquitectura capaz de ejecutar esas instrucciones utilizando menos ciclos de procesador, puede aportar un rendimiento mucho más alto. Otra característica de este tipo de arquitectura es que solo se accede a la memoria a través de determinadas instrucciones, no es accesible dentro de otras instrucciones o como parte de ellas.

Hoy en día esta es una de las arquitecturas más utilizadas para todos los microcontroladores del mercado, ya que la mayoría de ellos integran un microprocesador con especificaciones funcionales de tipo ARM[®] (*Advanced RISC Machine*).

- **Arquitectura Complex Instruction Set Computer (CISC)** [4]: Esta arquitectura se basa en el diseño de instrucciones que sean capaces de ejecutar varias operaciones de bajo nivel (como una lectura de la memoria, una operación aritmética y una escritura en memoria) o que sean capaces de ejecutar instrucciones más complejas en un sólo ciclo de CPU. Al contrario que la arquitectura RISC tienen un conjunto de instrucciones que se caracteriza por ser muy amplio y permitir operaciones

complejas entre operandos situados en la memoria o en los registros internos.

A día de hoy y con el objetivo de conseguir paralelismo entre instrucciones, los sistemas con CISC de alto rendimiento implementan sistemas que convierten instrucciones complejas en simples del tipo RISC, denominadas microinstrucciones.

1.3. Aplicaciones

Cada vez existen más productos que incorporan un microcontrolador con el fin de aumentar sustancialmente sus prestaciones, reducir su tamaño y coste, mejorar su fiabilidad y disminuir el consumo.

Sus aplicaciones son muy diversas, y varían desde las aplicaciones de uso industrial hasta otras de uso doméstico y cotidiano como pueden ser las lavadoras o los televisores, y sistemas más grandes como los de vigilancia y alarma, calefacción y aire acondicionado entre otros. Las comunicaciones y los productos de consumo general absorben más de la mitad de la producción de microcontroladores; el resto se distribuye entre el sector de automoción, la informática y los procesos industriales.

En cuanto al uso de estos dispositivos en la industria, se utilizan sobre todo para realizar funciones de adquisición de datos y actuación gracias a los conversores A/D y D/A que incorporan la mayoría de los dispositivos actuales. Se usan también para realizar el control secuencial de procesos automáticos debido al gran número de entradas y salidas inmunes al ruido que poseen. Y por último, son muy utilizados en el ámbito de la robótica, por ejemplo, para el control de motores y captura de señales de diferentes sensores necesarios para el funcionamiento autónomo del sistema.

La industria informática acapara gran parte de la producción de microcontroladores, ya que casi todos los periféricos del computador, desde el ratón o el teclado hasta la impresora, son regulados por un microcontrolador. Las comunicaciones y sus sistemas de transferencia de información utilizan microcontroladores incorporándolos en los grandes automatismos y en determinadas partes de los teléfonos actuales.

Otra de las grandes industrias que hacen uso de estos dispositivos es la industria de la automoción, donde se utilizan para aplicaciones esenciales como puede ser el control del motor o el sistema ABS, y en otros más actuales como los sistemas de conducción autónoma y asistencia a la conducción.

Por último, uno de los campos en auge para este sector es el de las aplicaciones médicas, donde la necesidad de instrumentación electrónica para el control de los procesos y el procesado de imágenes obtenidas está impulsando cada vez más su utilización [2].

1.4. Principales fabricantes y productos

El mercado de los microcontroladores se ha incrementado en los últimos años y han ido apareciendo nuevas empresas con productos cada vez más competitivos. A continuación se muestran algunos de estos fabricantes [1]:

- | | | |
|--------------------------|-------------------------|-------------------------|
| ■ Altera | ■ Holtek | ■ Parallax |
| ■ Analog Devices | ■ Hyperstone | ■ Rabbit Semiconductor |
| ■ Atmel | ■ Infineon | ■ Renesas Electronics |
| ■ Cypress MicroSystems | ■ Intel | ■ Rockwell |
| ■ Maxim Integrated | ■ Lattice Semiconductor | ■ Sillicon Laboratories |
| ■ ELAN Microelectronics | ■ Microchip Technology | ■ Sillicon Motion |
| ■ Energy Micro AS | ■ National Seiconductor | ■ Sony |
| ■ EPSON Semiconductor | ■ NXP Semiconductors | ■ Spansion |
| ■ Freescale Seiconductor | ■ NEC | ■ STMicroelectronics |
| ■ Fujitsu | ■ Panasonic | ■ Texas Instruments |

- Toshiba
- Xemics
- XMOS
- Uvicom
- Xilinx
- ZiLOG

1.5. Familia XMC4000 de Infineon

En este proyecto se trabajará con el Microcontrolador XMC4500 de Infineon, el cual forma parte de la familia de Microcontroladores XMC4000 [8]. Todos los dispositivos esta la familia se benefician de las capacidades para el control en tiempo real y el procesado de señales de la CPU con especificaciones ARM Cortex-M4; combinado con la memoria *Flash* embebida y el conjunto de periféricos específicamente diseñados para el control en tiempo real de los entornos industriales. La integración de temporizadores flexibles y generadores de pulsos en conjunto con conversores A/D y D/A de alta velocidad, permite la creación de esquemas de control sensor-actuador muy precisos. Además, se incluyen una serie de periféricos de comunicaciones actuales como Ethernet y USB; y periféricos para la interacción humano-máquina.

En la Figura 1.1 se muestra una tabla creada por Infineon para la comparación de las características de los diferentes dispositivos de esta familia.

		low-end high-end				
		XMC4100	XMC4200	XMC4400	XMC4500	XMC4700*
System Performance	Core				AMR® Cortex™-M4	
	CPU frequency (at 125 °C)	80 MHz	80 MHz	120 MHz	120 MHz	180 MHz
	Co-proc	Floating Point Unit				
	Flash size	128 kB	256 kB	512 kB	1 MB	2.5 MB
	RAM size	20 kB	40 kB	80 kB	160 kB	512 kB
Timers	Cache	4 kB	4 kB	4 kB	4 kB	6 kB
	POSIF	1x	1x	2x	2x	2x
	CCU4 (4ch)	2x	2x	4x	4x	4x
	CCU8 (4ch)	1x	1x	2x	2x	2x
	High-resolution PWM (150 ps) channels	4x	4x	4x		
Signal-Processing	ADC 12-bit	2x	2x	4x	4x	4x
	Delta-sigma demodulator			4x	4x	4x
	DAC	2x	2x	2x	2x	2x
Communication	IEEE 1588 Ethernet MAC			1x	1x	2x
	USB	FS DEV	FS DEV	FS OTG	FS OTG	HS OTG
	SD/MMC				✓	✓
	Serial channels (URAT, SPI, I ² C, I ² S)	4x	4x	4x	6x	6x
	Ext. memory I/F				✓	✓
	CAN	2x	2x	2x	3x	3x
Touch button	✓	✓	✓	✓	✓	

* under definition

Figura 1.1: Tabla comparativa de los dispositivos de la familia XMC4000 de Infineon [8].

2. Kit de Desarrollo *Hexagon Application Kit*

2.1. *Hexagon Application Kit*

El Kit de Desarrollo *Hexagon Application Kit* con el que se va a trabajar es una plataforma diseñada para mostrar y probar las capacidades del Microcontrolador XMC45000 de Infineon. Consta de una placa central llamada CPU Board y tres tarjetas satélite (*Automation I/O Card*, *Standard Human Machine Interface Card* y *Ethernet/CAN/RS485 Interface Card*) cuyas funciones son probar lo siguiente respectivamente: capacidades del ISOFACE™ de Infineon junto con el XMC4500 [9], capacidades de interacción entre el humano y el XMC4500 [11] y capacidades de comunicación del XMC4500 [10].

A continuación se ofrece una descripción detallada del hardware y las características de las cuatro placas que forman el Kit de Desarrollo y del Microcontrolador XMC4500.

2.2. Microcontrolador XMC4500

La serie XMC4500 [13] pertenece a la familia de microcontroladores industriales XMC4000. Los dispositivos de la familia XMC4000 están optimizados para control eléctrico de motores, conversión de potencia, conectividad industrial y aplicaciones de adquisición de datos de sensores y control.

Los dispositivos de la serie XMC4500 combinan la funcionalidad extendida y rendimiento del núcleo ARM Cortex-M4, con potentes subsistemas de periféricos y unidades de memoria *on-chip*. A continuación se muestran las características más importantes disponibles en los dispositivos de la serie XMC4500, y en la Figura 2.1 se muestra un diagrama de bloques donde se esquematizan todas estas características del Microcontrolador.

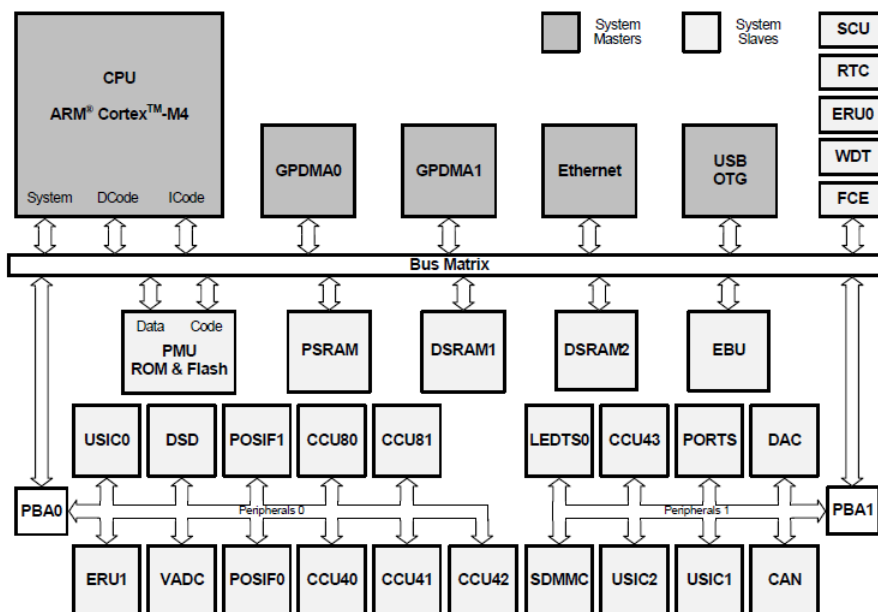


Figura 2.1: Diagrama de bloques del sistema XMC4500 [13].

2.2.1. Subsistema CPU

El núcleo del XMC4500 consiste en la Unidad Central de Proceso o CPU (Central Processing Unit), en la cual se integran la Unidad de Coma Flotante o FPU (Floating Point Unit) y la Unidad de Protección de Memoria o MPU (Memory Protection Unit); y en bloques de interfaz con la memoria para acceder al programa y a los datos almacenados en la misma, incluyendo la Unidad de Memoria de Programa o PMU

(Program Memory Unit).

El núcleo de la **CPU** tiene las siguientes características:

- CPU ARM Cortex-M4 de 32 bits.
- Conjunto de instrucciones Thumb2 de 16 y 32 para aumentar el rendimiento y velocidad del microprocesador.
- Instrucciones DSP tipo MAC, que permite multiplicar, sumar y guardar el resultado en un ciclo de reloj.
- Temporizador del sistema (SysTick) para el soporte de un Sistema Operativo.

La **FPU** realiza operaciones de acuerdo con el estándar para la aritmética en coma flotante IEEE 754 con precisión simple de 32 bits.

La **MPU** mejora la fiabilidad del sistema. Proporciona un control específico de la memoria, permitiendo a las aplicaciones que utilicen múltiples niveles de privilegio, separando y protegiendo código, datos y la pila basándose en una operación tarea a tarea (task-by-task). Es posible utilizar hasta ocho regiones diferentes de memoria además de tener una región de fondo predefinida opcional.

Además, el subsistema de la CPU incorpora las siguientes funciones:

- **Sistema de Interrupciones de Prioridad Múltiple Programable o NVIC:**
El XMC4500 incorpora el NVIC de AMR con 112 nodos de interrupción y 64 niveles de prioridad. La mayor parte de las fuentes de interrupción están conectadas a un nodo de interrupción, este microcontrolador permite además el rutado de la petición de servicio directamente a unidades dedicadas como el de Acceso Directo a Memoria (DMA), el Temporizador o el Conversor Analógico-Digital (ADC).
- **Acceso Directo a Memoria de Propósito General o GPDMA:**
El GPDMA es un controlador configurable del Acceso Directo a Memoria que permite la transferencia de datos a alta velocidad entre periféricos y memorias. Es posible realizar transferencias complejas de datos con una mínima intervención del procesador, reservando los recursos de la CPU para otras operaciones. Proporciona transferencias de tipo *multi block*, *scatter/gather* y *linked list*.
- **Motor Flexible de CRC o FCE:**
El FCE proporciona una implementación de algoritmos de Código de Redundancia Cíclica (CRC), entre los que se encuentran los polinomios del IEE802.3 CRC32, el CCITT CRC16 y el SAE J1850 CRC8. El objetivo del FCE es poder acelerar aplicaciones de software o servicios de Sistema Operativo utilizando firmas CRC.

2.2.2. Memorias *on-chip*

El XMC4500 dispone de los siguientes bloques de memoria:

- **16 KB de Boot ROM o BROM *on-chip*:**
Contiene el código de arranque y la tabla de vectores de excepciones. El *firmware* se ejecuta inmediatamente después de la excepción de Reset.
- **64 KB de Memoria de Programa de alta velocidad *on-chip*:**
El propósito de la memoria RAM de Código (PSRAM) es almacenar código de usuario o del Sistema Operativo. El acceso a esta memoria se hace mediante un bus con organización matricial (*Bus Matrix*) y proporciona a la CPU acceso sin tiempo de espera (*zero-wait-state*) para ejecución de código o acceso a datos.
- **64 KB de Memoria de Datos de alta velocidad *on-chip*:**
La memoria RAM de Sistema (DSRAM1) se utiliza como memoria general para el almacenamiento de datos por parte del usuario. El acceso a los datos almacenados es análogo al utilizado por la PSRAM.

- **32 KB de Comunicación de alta velocidad *on-chip*:**

La memoria RAM de Comunicación (DSRAM2) es la utilizada por las unidades de la interfaz de comunicación como el USB y Ethernet.

- **1024 KB de Memoria *Flash* con 4 KB de Memoria caché *on-chip*:**

La memoria *Flash* se utiliza para código no volátil o almacenamiento de datos constantes. Tiene mecanismos de protección de lectura y escritura y un módulo de hardware de corrección de errores que asegura la consistencia de los datos durante todo el tiempo de vida del dispositivo y bajo condiciones de entorno industrial.

La memoria caché integrada aumenta la velocidad de la ejecución en un factor de 3 en comparación con la ejecución de código fuera de la memoria caché.

Además de estas memorias integradas, con el objetivo de cumplir las necesidades de ciertas operaciones donde se requieren mas periféricos, la Unidad Externa de Bus o EBU (External Bus Unit) proporciona los medios necesarios para añadir de forma opcional una amplia variedad de memorias externas.

2.2.3. Periféricos de comunicaciones

El XMC4500 ofrece un conjunto de periféricos que soportan protocolos avanzados de comunicación. Además de Ethernet, USB, CAN y USIC este microcontrolador proporciona interfaces con varias memorias y una unidad para llevar a cabo una interfaz humano-máquina a través del módulo llamado LED y Touch Sense.

Los principales módulos de comunicaciones son los siguientes:

- **LED y Touch Sense (LEDTS):**

El módulo LEDTS permite medir la capacitancia de hasta 8 pulsadores táctiles utilizando una topología de oscilador a relajación. Por otra parte, permite controlar hasta 64 LEDs en una matriz de LEDs, pudiendo ambas funciones compartir pines para minimizar el número necesario para estas aplicaciones.

- **Interfaz SD/MMC:**

La interfaz *Secure Digital/Multi Media Card* (SDMMC) proporciona una interfaz entre las tarjetas SD/SDIO/MMC y el bus del sistema. Además de estas, soporta tarjetas de tipo SDHC y puede operar hasta a 48MHz. Este módulo puede transmitir a un máximo de 24MB/s para tarjetas SD y a 48MB/s para tarjetas MMC. Algunas de las aplicaciones para las que está diseñado este módulo son la extensión de memoria o la actualización de *firmware*.

- **Unidad Externa de Bus o EBU:**

La EBU proporciona acceso síncrono y asíncrono a las siguientes memorias externas:

- ROMs, EPROMs
- *Flash* tipo NOR y NAND
- RAMs y PSRAMs estáticas
- PC133/100 compatible con SDRAM
- *Burst Flash*

- **Ethernet MAC (ETH):**

Está soportada la Ethernet de 10 y 100Mbit/s. Esta Ethernet puede ser utilizada para implementar aplicaciones conectadas a Internet que utilicen protocolos IPv4 y IPv6. Este microcontrolador incluye soporte para sincronización de tiempos de acuerdo con la norma IEEE1588 para protocolos de Ethernet en Tiempo Real.

- **Bus Universal en Serie o USB:**

El módulo USB soporta las funciones de anfitrión (*device*) y huésped (*host*) donde es configurable el funcionamiento en cualquiera de los dos modos únicamente o como OTG, y cumple las especificaciones del suplemento *On-The-Go* USB 2.0. Este módulo USB 2.0 soporta transferencias a 12Mbit/s.

- **Canal Universal de Interfaz en Serie o USIC (Universal Serial Interface Channel):**

El USIC es un módulo de interfaz flexible que cubre diversos protocolos de comunicación serie como son ASC, LIN, SSC, I2C e I2S.

El XMC4500 tiene 3 módulos USIC con 2 canales independientes en cada uno de ellos. Se implementa en cada uno un buffer FIFO para almacenar los datos de transmisión y hay disponibles varias señales de selección de chip para permitir la comunicación con varios dispositivos utilizando un único canal.

- **Controller Area Network (CAN):**

El módulo MultiCAN contiene tres nodos CAN independientes con total funcionalidad, capaces de intercambiar Tramas de Datos y Tramas Remotas a través de una función puerta de enlace (*gateway*). Cada nodo CAN puede recibir y transmitir tramas estándar con identificadores de 11 bits y tramas extendidas con identificadores de 29 bits. Todas las tramas se tratan de acuerdo con la especificación V2.0 B.

2.2.4. Periféricos analógicos *frontend*

Las interfaces de conexión que posee el XMC4500 con el mundo analógico, se exponen a continuación:

- **Conversor Analógico-Digital:**

El módulo del Conversor Versátil Analógico-Digital (VADC) consiste en cuatro núcleos independientes de 8 canales de entrada cada uno que operan de acuerdo con el principio de aproximaciones sucesivas. La resolución es programable desde 8 hasta 12 bits con un tiempo total de conversión inferior a 500ns para la resolución de 12 bits.

Cada núcleo proporciona una máquina de estados versátil que permite medidas complejas. Los núcleos pueden sincronizarse y las conversiones se pueden realizar en segundo plano. Pueden priorizarse múltiples eventos de *trigger* y permitir una medida exacta de señales temporalmente críticas. El almacenamiento y manejo de los resultados evita la pérdida de datos y asegura la consistencia.

La estructura básica de este VADC permite que las tareas solo lean resultados válidos y no sea necesario preocuparse por el inicio de las conversiones. Por otra parte, un conjunto de comparadores *on-chip* para valores fuera de rango se encargan de monitorizar el posible sobrevoltaje en las entradas del VADC.

- **Demodulador Delta-Sigma o DSD:**

Los cuatro canales del DSD convierten los flujos de bits de entrada a valores discretos. Cada canal del demodulador tiene dos filtros digitales programables de tipo SINC o COMB. El filtro rápido puede usarse para comprobar los límites de la señal, mientras que el lento se utiliza para la medida de la señal. Una etapa integradora se encarga de la cancelación de la frecuencia de la portadora y otro mecanismo compensa la posible diferencia de fase entre dos canales. El generador de vectores integrado genera un flujo de bits sinusoidal que puede utilizado en múltiples aplicaciones.

- **Conversor Digital-Analógico o DAC:**

Este módulo consiste en 2 Conversores Digital-Analógico de 12 bits, convirtiendo dos señales de entrada digitales a dos señales de voltaje analógico a la salida a una frecuencia máxima de 5MHz.

El modo integrado de generación de ondas permite la generación de formas de onda entre varias a elegir. Los valores introducidos en el DAC pueden llegar desde la CPU o el DMA directamente a uno o ambos canales. Además es posible añadir un *offset*, escalar la amplitud y utilizar varias fuentes diferentes como *trigger* del sistema.

2.2.5. Periféricos de control industrial

A continuación se muestran los núcleos necesarios para el control de movimiento, control de motor, conversión de potencia y otras aplicaciones basadas en el tiempo:

- **Unidad Captura/Compara 4 o CCU4 (Capture/Compare Unit 4):**

Este módulo está preparado para implementar aplicaciones de control de sistemas de electrónica de

potencia, en las que se necesitan temporizadores de propósito general para monitorizar y acondicionar la señal y generación de señal por Modulación por Ancho de Pulsos o PWM (Pulse Width Modulation).

- **Unidad Captura/Compara 8 o CCU8 (Capture/Compare Unit 8):**

Este módulo se utilizará para aplicaciones que necesiten señales PWM más complejas que las anteriores.

- **Unidad de Interfaz de Posición o POSIF:**

La unidad POSIF es un componente flexible y potente para sistemas de control de motor que utilicen Encoders Rotatorios o Sensores Hall como lazo de realimentación. Esto permite la construcción de lazos de control tanto simples como complejos para aplicaciones con motores industriales y en automóviles, buscando el objetivo de conseguir movimiento y monitorización de posición de altas prestaciones.

Además, se incorporan las siguientes características:

- *Window Watchdog Timer* (WDT) para aplicaciones que requieren mayor seguridad.
- Sensor para la medición de temperatura interna crítica.
- Módulo de *Clock* en Tiempo Real con soporte para alarma.
- Unidad de Control del Sistema o SCU para configuración y control del sistema.

2.2.6. Líneas de Entrada/Salida

A continuación se enumeran las características de las líneas de entrada y salida:

- Módulo Programable de Control de *Drivers* de Puerto (PORTS).
- Direccionable bit a bit (*Individually bit addressable*).
- Modo de entrada tri-estado.
- Modos de salida *push/pull* o drenador abierto.
- Test de escaneo de límites (*Boundary scan test*) mediante la interfaz JTAG.

2.2.7. Soporte para *debug on-chip*

El XMC4500 aporta soporte para las siguientes características de *debug*:

- 8 *breakpoints*
- Arquitectura ARM *CoreSight*
- Traza

Por otra parte, los interfaces que soporta este *debugger* son: ARM-JTAG, ARM-SWD (Serial Wire Debugger) y *single wire trace*.

2.3. CPU Board

Para la placa central que contiene el Microcontrolador XMC4500 Infineon aporta dos opciones. La primera de ellas es la CPU_45A-V3, de propósito general y la segunda, la CPU_45B-V3, enfocada a probar la unidad de bus externa (EBU) del XMC4500. Para este trabajo se utilizará la placa de propósito general, cuyas características se describen a continuación.

2.3.1. CPU Board XMC4500 General Purpose (CPU_45A-V3)

La descripción general de recursos es la siguiente [12]:

- Microcontrolador XMC4500 (basado en ARM[®] Cortex[™] M-4), CPU a 120MHz, 1MB de memoria Flash on-chip, 160KB de memoria RAM, empaquetado LQFP-144.
- Tres conectores (ACT,HMI y COM) para conectar las tarjetas satélite.
- Conector micro USB que soporta USB OTG *Host/Device*.
- Opciones de depuración o debug:
 - *On-board Debugger* mediante USB.
 - *Cortex Debug Connector* (10 pines)
 - *Cortex Debug+ETM Connector* (20 pines)
- Reset mediante botón pulsador.
- 32 MBit de memoria *Flash* quad SPI.
- Conector PowerScale específico para medir la potencia consumida por el MCU.
- 5 LEDs:
 - 3 LEDs de encendido.
 - 1 LED de Usuario.
 - 1 LED de Reset.
 - 1 LED de *Debug*.
- Botón de Usuario.
- Potenciómetro.
- Alimentación:
 - Mediante el conector Micro-USB.
 - Mediante las tarjetas satélite.
 - Mediante el conector *Debug* USB.
 - Con una batería RTC de apoyo.

A continuación se da una descripción detallada del hardware y de su modo de funcionamiento, y una imagen de dicha placa en la Figura 2.2:

Alimentación: La alimentación de esta placa se puede realizar a través del conector USB (5V), sin embargo hay un límite de corriente que puede tomarse desde el PC a través del USB. Si la placa CPU_45A-V3 se utiliza para controlar otras placas y la corriente total necesitada excede los 500mA, entonces es necesario alimentar la placa de la CPU con una fuente externa conectada al USB o a través de una placa satélite que soporte fuentes de alimentación externas.

Reset: El pin de reset es bidireccional. Una resistencia interna *pull-up* mantendrá el pin en alto mientras el funcionamiento sea normal, un nivel bajo de la señal en este pin forzará un reset del hardware. Para el caso de un reset interno, el pin producirá una señal en bajo. En la placa CPU_45A-V3 es posible encontrar un botón físico de reset que permite un reseteado de hardware durante la operación de la CPU.

La señal de reset está rutada a todos los conectores satélite que se resetearán a la vez que esta. El estado del reset se indica en la placa a través del LED rojo (V407), el cual estará encendido durante el proceso de reseteado y apagado en condiciones normales de operación.

Clock: Un cristal externo de 12MHz proporciona la señal de *clock* al Microcontrolador XMC4500. Para el Reloj de Tiempo Real (RTC) se utiliza otro cristal externo e independiente de 32.768kHz.

Opción de arranque (*Boot Option*): Tiene cuatro modos de arranque o *boot* diferentes: El modo normal que arranca desde la memoria *Flash*, otro que hace uso también del Canal Serie Asíncrono y otros dos que utilizan en bus CAN, siendo uno de estos modos configurable en una sección de la memoria.

Interfaz de *Debug*: La placa CPU_45A-V3 soporta el *debug* JTAG mediante tres tipos de conectores diferentes:

- *On-board Debugger*: Este soporta *Serial Wire Debug*, *Serial Wire Viewer* y comunicación Full Duplex UART (*Universal Asynchronous Receiver Transmitter*) mediante un Virtual COM por USB, los interruptores U301 y U303 se utilizan para configurar este tipo de comunicación.
- *Cortex Debug Connector* (10 pines): La placa CPU_45A-V3 soporta la operación *Serial Wire Debug* y *Serial Wire Viewer* a través del conector de 10 pines.
- *Cortex Debug+ETM Connector* (20 pines): A través de este conector de 20 pines es posible utilizar los dos modos de operación mencionados para el de 10 pines, además de la operación en Traza de Instrucciones.

Si se utiliza cualquier *debugger* externo conectado a los conectores de 10/20 pines, el *On-board debugger* se desconecta automáticamente.

Memoria *Serial Flash*: La placa CPU_45A-V3 proporciona una memoria *Flash* de 32Mbit utilizando como interfaz con el XMC4500 la SPI (Serial Peripheral Interface), la cual puede ser configurada como single, dual o quad.

USB: Es posible configurar esta interfaz como anfitrión, huésped o con función dual (OTG). Si está configurado en modo huésped/OTG la placa es capaz de alimentar el dispositivo conectado siempre y cuando no se requieran más de 500mA.

Reloj en Tiempo Real o RTC: La CPU del XMC4500 tiene dos dominios con necesidad de alimentación, el dominio del núcleo y el de hibernación. El RTC se encuentra en el dominio de hibernación, el cual puede operar si se encuentra alimentado a través del pin VBAT donde se puede conectar una batería de 3V o bien si se le proporcionan 3.3V desde el dominio de núcleo y es capaz de "despertar" a todo el sistema del modo hibernación. Sin ningún tipo de alimentación el dominio de hibernación podrá funcionar durante 10 segundos aproximadamente.

LEDs de Usuario y Botones de Usuario: Existe un botón conectado directamente al pin 2.15 del microprocesador y un LED de usuario en el pin 3.9.

Potenciómetro: El potenciómetro presente en la placa tiene el propósito de probar el ADC integrado en el XMC4500, este está conectado al Canal 0 del Grupo 0 de los 4 disponibles en el VADC. La salida del potenciómetro varía desde 0V hasta 3.3V.

Conectores satélite: La placa CPU_45A-V3 tiene 3 conectores satélite para conectar las placas de extensión AUT_ISO-V1, HMI_OLED-V1 y COM_ETH-V1, de 80 pines cada uno.

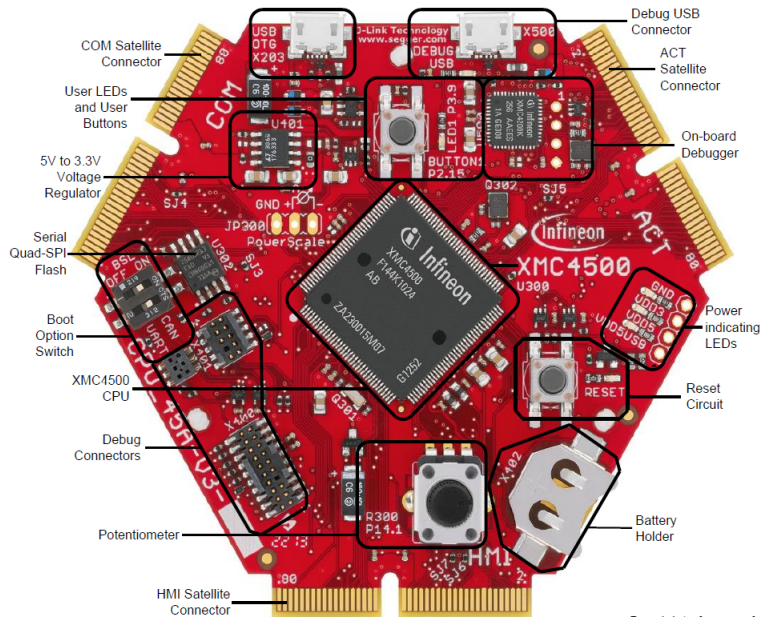


Figura 2.2: Imagen de la placa satélite CPU_45A-V3 con la señalización de sus elementos más importantes [12]

2.4. Placas satélite

2.4.1. Automation I/O Card (AUT_ISO-V1)

Esta tarjeta satélite [9] incorpora dos dispositivos con la tecnología ISOFACE de Infineon, uno para los canales de entrada ISOFACE IN (ISO1H811T) y otro para los de salida ISOFACE OUT (ISO1H812G). El propósito de esta placa es mostrar el funcionamiento de estos dispositivos en conjunto con el Microcontrolador XMC4500.

La descripción general de recursos es la siguiente:

- Conexión a la placa CPU mediante el conector ACT.
- ISOFACE OUT de hasta 8 canales.
- ISOFACE IN de hasta 8 canales.
- Expansor de hasta 8 canales basado en I2C.
- 2 tipos de alimentación:
 - Jack para alimentación externa de 24V.
 - A través del conector ACT.

A continuación se da una descripción detallada del hardware y de su modo de funcionamiento, y una imagen de dicha placa en la Figura 2.3:

ISOFACE OUT e ISOFACE IN: Ambos dispositivos están preparados como salida y entrada de señales digitales respectivamente, alimentados con 3.3V por la parte de la CPU y 24V por la parte de las salidas y entradas respectivamente.

Expansor IO: La placa AUT_ISO-V1 soporta una expansión de Entradas/Salidas de Propósito General o GPIO a través del Expansor de entradas IO-Expander. Este utiliza el bus de datos serie I2C. La placa satélite admite hasta 8 GPIOs.

Alimentación: La alimentación de esta placa proviene de los 5V aportados a través del conector satélite, o bien de una fuente externa de 24V/1A conectada a un *power jack* que posee la placa el cual solo

será necesario en caso que el voltaje proveniente de la placa no sea suficiente.

Esta placa también tiene un convertor DC-DC que reduce los 24V a 5V. El voltaje de entrada puede variar desde 12V a 24V. A su vez, en la placa se está generando un voltaje de 3.3V a partir de los 5V.

Conector satélite: Esta placa tiene un único conector satélite de 80 pines que sólo puede utilizarse para su conexión con la placa de la CPU.

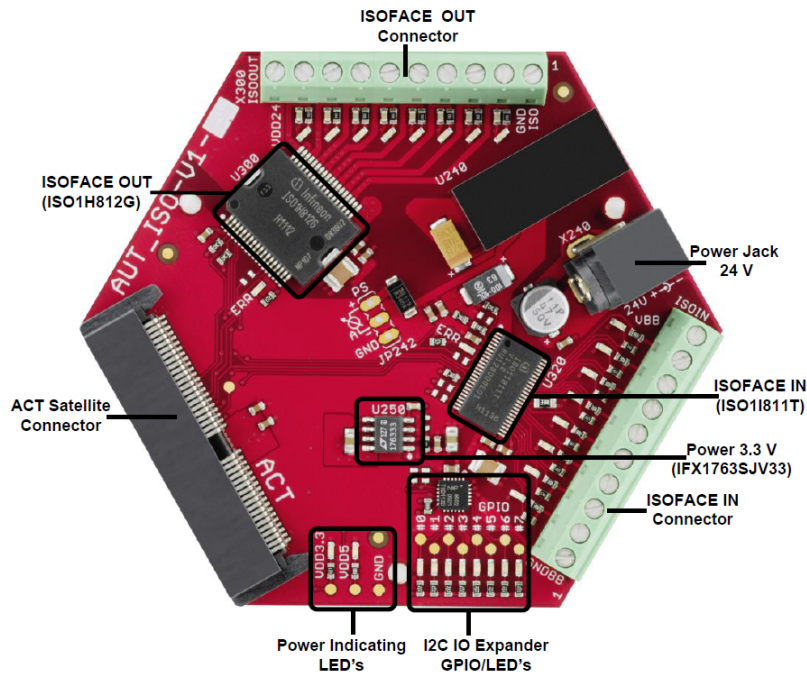


Figura 2.3: Imagen de la placa satélite AUT_ISO-V1 con la señalización de sus elementos más importantes [9]

2.4.2. Standard Human Machine Interface Card (HMI_OLED-V1)

La descripción general de recursos es la siguiente [11]:

- Conexión a la placa CPU mediante el conector HMI.
- Conector para tarjeta MicroSD (modo 4bit)
- Display PMOLED de 1.54", 160x128 píxels.
- Códec de audio en estéreo TLV320AIC3204.
- Receptáculo estéreo de 2.5mm para auriculares con micrófono.
- 2 botones táctiles con LEDs.
- Expansor de hasta 8 canales basado en I2C.
- Alimentación mediante conector satélite.

A continuación se da una descripción detallada del hardware y de su modo de funcionamiento, y una imagen de dicha placa en la Figura 2.4:

Ranura para memoria MicroSD: La placa satélite HMI_OLED-V1 tiene un conector para MicroSD. La interfaz SD/MMC del XMC4500 puede ser usada de 4 modos diferentes (modo SPI, SD de 1 bit, SD de 4 bits, MMC de 8 bits). Esta placa permite la implementación del modo de 4 bits para la interfaz de la SD.

Display OLED: La placa HMI_OLED-V1 tiene un display OLED de matriz pasiva (PMOLED) de Den-sitron. Este display tiene una resolución de 160x128 píxeles en una pantalla de 1.54".

La interfaz utilizada para conectar el display con el XMC4500 es la SPI. El Driver y Controlador del Display PMOLED utilizado por este módulo es el Syncoam SEPS525F de 160 RVA x 128 Puntos, 262K colores. La propia tarjeta implementa la alimentación (13V) para la retroiluminación del display, que se indica con el LED VDD13.

Audio: La placa satélite HMI_OLED-V1 implementa una interfaz de audio con el códec de audio estéreo TLV320AIC3204. El control del códec de audio se realiza mediante el bus I2C, mientras que los datos de audio utilizan el bus I2S como interfaz con el XMC4500.

El pin GPIO P2.10 se utiliza para resetear el códec de audio, el cual tiene integrado un driver para auriculares así como una entrada para micrófono analógico o digital. El receptáculo para los auriculares integrado en la placa es estéreo de 4 pines y 2.5mm.

Expansor IO: Al igual que en la placa AUT_IS0-V1 existe un expansor de hasta 8 pines GPIO con interfaz I2C.

Botones táctiles: La placa satélite HMI_OLED-V1 tiene dos botones táctiles (TOUCH250 y TOUCH251) que se conectan con la CPU a través del conector satélite. Estas señales están a su vez conectadas con los LEDs V250 y V251 que pueden ser configurados.

Alimentación: Esta placa no tiene alimentación externa, se alimenta con 5V provenientes de la placa de la CPU. Se generan 3.3V con el regulador integrado en la placa y los 13V mencionados anteriormente para la retroiluminación del display.

Conector satélite: Esta placa sólo tiene un conector satélite que ha de ser conectado siempre con la placa de la CPU.

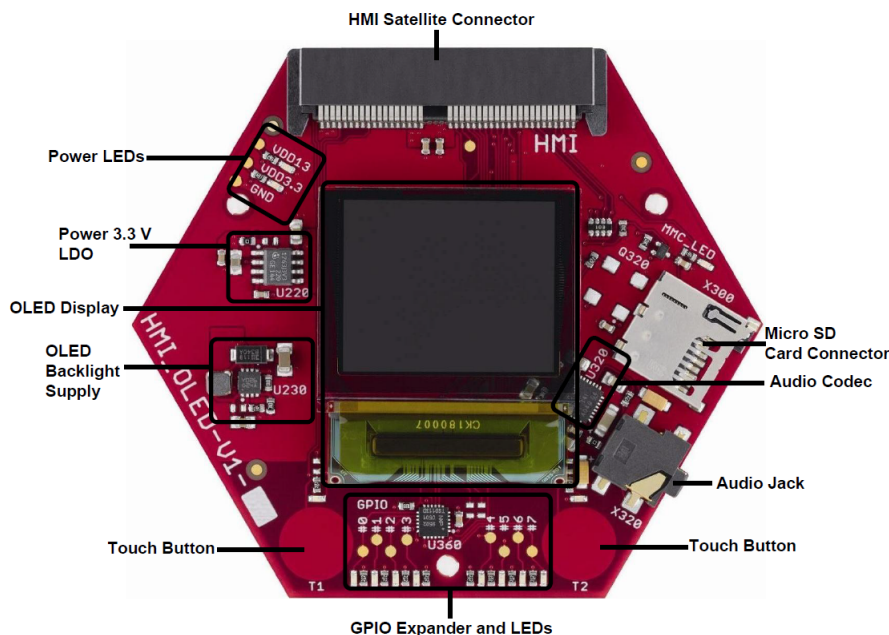


Figura 2.4: Imagen de la placa satélite HMI_OLED-V1 con la señalización de sus elementos más importantes [11]

2.4.3. Ethernet/CAN/RS485 Interface Card (COM_ETH-V1)

La descripción general de recursos es la siguiente [10]:

- Conexión a la placa CPU mediante el conector COM.
- Ethernet (RJ45) 10/100Mbps.
- Transceptor CAN con conector CAN (DE-9 macho).
- Transceptor RS-485, full-duplex, *bootable* (DE-9 hembra).
- Transceptor CAN, *bootable* (cabezal de 9 pines).
- Expansor de hasta 8 canales basado en I2C.
- 2 tipos de alimentación:
 - *Jack* para alimentación externa de 24V.
 - A través del conector COM.

A continuación se da una descripción detallada del hardware y de su modo de funcionamiento, y una imagen de dicha placa en la Figura 2.5:

Ethernet: La placa satélite COM_ETH-V1 incluye un jack RJ45 magnético para la comunicación con Redes de Área Local. En la placa, el dispositivo de la capa física (PHY) DP83848CVV se conecta a la CPU mediante la interfaz RMI (Reduced Media-Independent Interface). Esta interfaz requiere un *clock* que se genera en la placa mediante un oscilador a cristal de 50MHz integrado.

CAN: La placa satélite COM_ETH-V1 soporta una interfaz de tipo CAN mediante un conector DE-9. Un transceptor CAN se conecta a la placa a través del conector satélite de 80 pines.

RS485/CAN Boot Loader: La placa COM_ETH-V1 soporta la interfaz RS485 a través del conector DE-9. De nuevo, se utiliza un transceptor RS485 conectado a la CPU mediante el conector satélite. Es posible elegir si se quiere la opción de arranque CAN o RS485 mediante los conectores X330 y X331 respectivamente.

Expansor IO: Al igual que en las placas AUT_ISO-V1 y HMI_OLED-V1 se pueden añadir hasta 8 entradas o salidas de propósito general.

Alimentación: Esta placa satélite, al igual que la AUT_ISO-V1 cuenta con la posibilidad de alimentación externa de 24V/1A y de 5V a través del conector de la placa satélite.

Conector satélite: De nuevo esta placa presenta un único conector que solamente puede ser utilizado para conectarse con la placa central de la CPU.

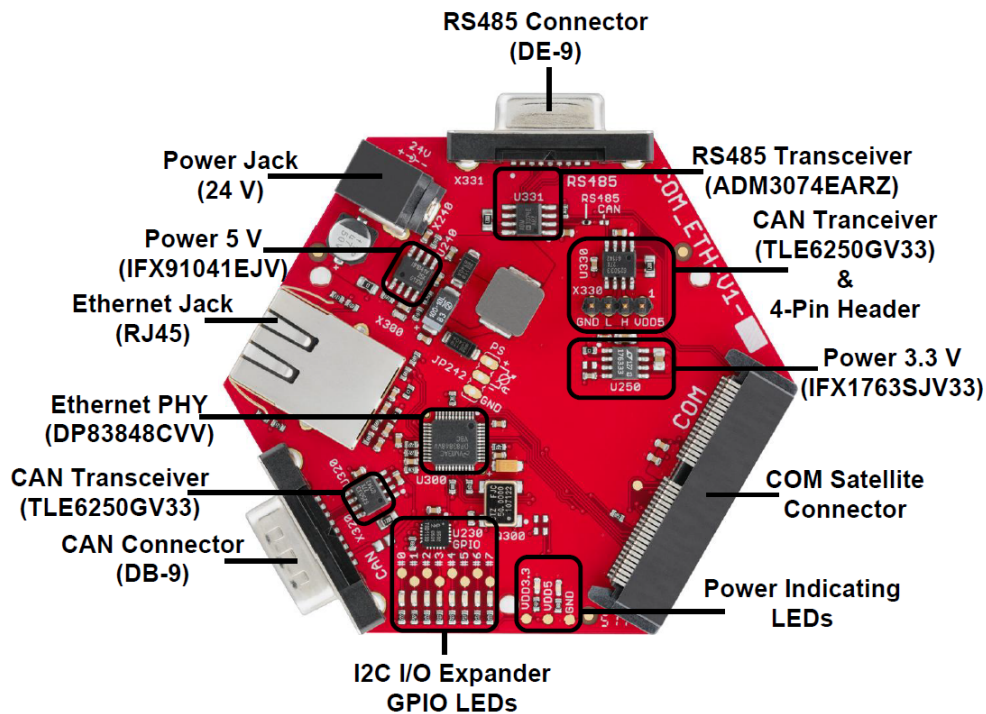


Figura 2.5: Imagen de la placa satélite COM_ETH-V1 con la señalización de sus elementos más importantes [10]

2.4.4. Placa de Extensión de Pines (UNI_EXT01-V2)

Esta placa (Figura 2.6) sirve como adaptador de acceso a los 80 pines de un conector satélite para la ampliación o extensión de medidas y aplicaciones. Encaja con los tres conectores satélite que tiene la placa central de la CPU y se puede conectar entre cualquiera de las tres placas satélite [7].

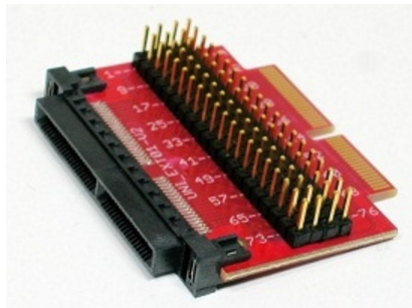


Figura 2.6: Imagen de la placa satélite UNI_EXT01-V2 [7].

2.5. Entorno de desarrollo de software DAVE™ (Versión 4)

DAVE™ (Digital Application Virtual Engineer), una herramienta gratuita de desarrollo de software en lenguaje C/C++ y de generación de código para aplicaciones con microcontroladores de la marca Infineon Technologies AG, es el programa utilizado para programar el Microcontrolador XMC4500. DAVE es un sistema autónomo con módulos independientes de generación automática de código y es adecuado para desarrollar drivers para estos controladores y ayudar al desarrollador con plantillas de código C. En estas plantillas que se generan automáticamente se incluyen las funcionalidades deseadas por el usuario [5]. Las últimas versiones de DAVE incluyen lo necesario para desarrollar, compilar y depurar sobre el objetivo (*debug on target*), todo ello basado en el ARM GCC Tool suite para construir, el GBD ARM para depurar y con Eclipse como Ambiente de Desarrollo Integrado (*Integrated Development Environment* o IDE) [6].

Es posible programar en este entorno de desarrollo directamente escribiendo código en lenguaje C/C++ o utilizando la programación basada en su Interfaz Gráfica de Usuario. Esta programación utilizando la GUI de DAVE consiste en interconectar «APPS»; librerías de código fuente configurable y de generación automática que controlan diferentes funciones del microcontrolador. Este kit de desarrollo de software (SDK) permite configurar de forma gráfica ciertas opciones de las Apps [6].

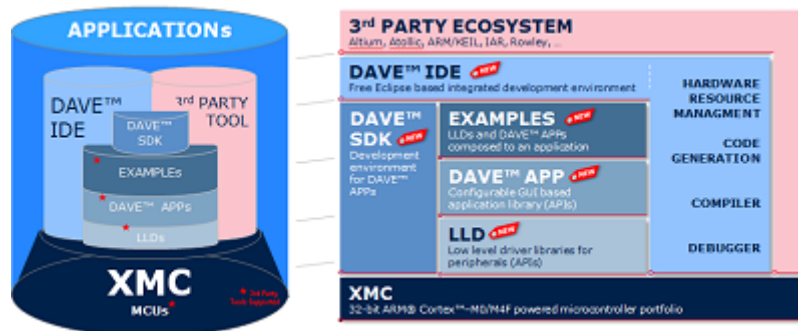


Figura 2.7: Perspectiva de la composición del entorno de desarrollo DAVE [6].

2.5.1. Composición de las APPs de DAVE™

Una APP del entorno DAVE proporciona APIs al usuario; las cuales están centradas en los recursos de hardware posiblemente interconectados, y puede ser utilizada bien independientemente, o bien, dentro de una jerarquía de APPs.

Hay que tener en cuenta que al añadir una APP al proyecto las APIs no están disponibles todavía. Es necesario configurar la APP por medio del entorno gráfico y llevar a cabo una generación de código. Es en ese momento cuando se generan los archivos de código fuente y es en esos archivos donde se encuentra las APIs de las APPs.

Cuando se añade una APP a un proyecto, DAVE crea un «contenedor» de datos para guardar todos los datos pertenecientes a esta APP, por lo que tener múltiples APPs en un proyecto implica tener múltiples contenedores de datos.

La **generación de código** es esencialmente el proceso de generar la librería de código fuente de una APP a partir de los datos configurados acerca de la misma en la interfaz gráfica. Cuando se configura una APP utilizando su GUI, se están configurando los recursos que consumirá esa APP. Cuando se pulsa el botón de generación de código (📄), el motor de código de DAVE recoge el contenedor de datos y crea una estructura de datos que contiene a su vez toda la información de la configuración en lenguaje C. Esta estructura se ha preparado para que el código de la aplicación se la pase a la API de inicialización proporcionada por la APP. La función de la API de DAVE `DaveInit()` se utiliza al principio de cada proyecto para inicializar todas estas estructuras de datos [6].

2.5.2. Programación mediante el entorno gráfico (GUI) de DAVE™

El entorno de desarrollo DAVE tiene varias pantallas que permiten configurar diferentes aspectos de las APPs que forman un proyecto. En la Figura 2.8 se muestra un ejemplo típico de la presentación del programa.

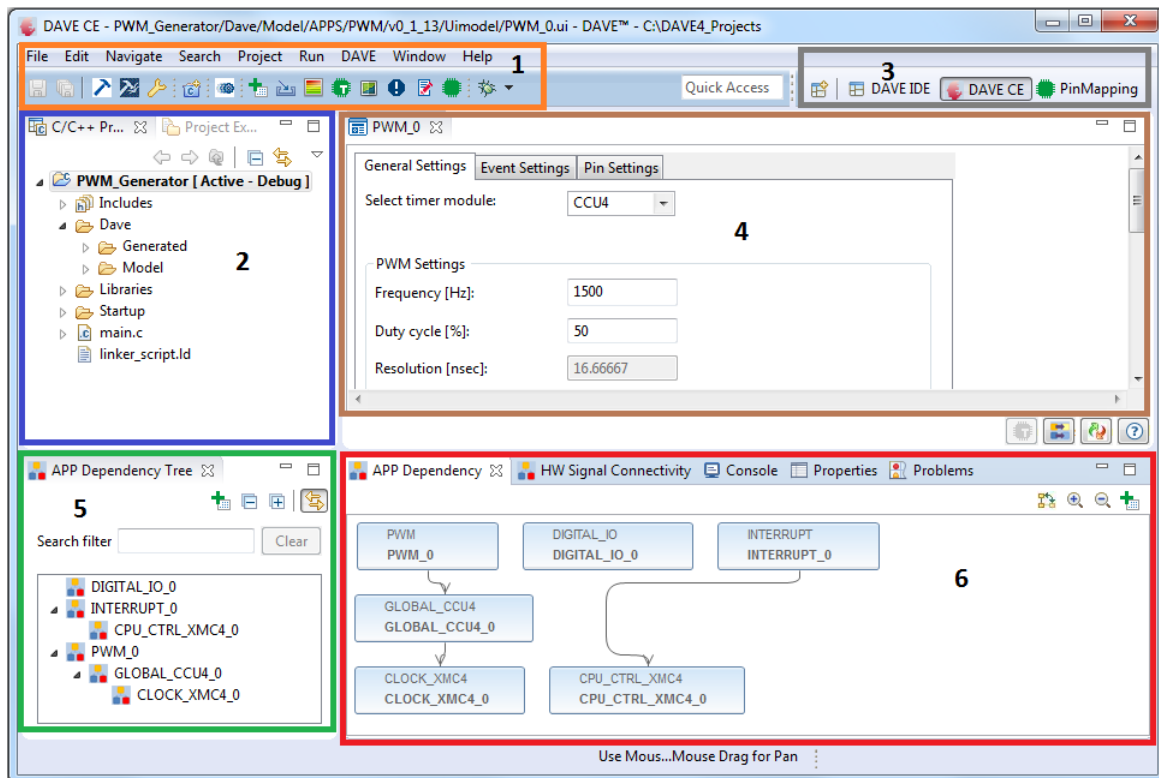




Figura 2.8: *Layout* habitual del entorno de desarrollo de software DAVE [6].

Los diferentes paneles que se aprecian en la Figura 2.8 son los siguientes:

1. Menu de herramientas: Aquí se encuentran iconos como el de generación de código o *debug* (🔧).
2. Explorador del proyectos: En este panel se pueden encontrar todos los proyectos que forman parte del espacio de trabajo (*Workspace*) activo. Desde aquí es posible ver la estructura de los archivos que forman el proyecto activo e incluso añadir/eliminar archivos y carpetas.
3. Perspectivas: DAVE tiene cuatro perspectivas: IDE, CE, *Debugger* y *Pin Mapping*. La perspectiva CE permite ver más cosas relacionadas con las APPs como puede ser la dependencia entre ellas o la vista de conectividad entre recursos de hardware con respecto a la perspectiva IDE. La perspectiva del *debugger* muestra las acciones típicas de este como son la ventana de código de programa, *breakpoints* o la traza de la pila por ejemplo. Por último, la perspectiva *Pin Mapping* (Figura 2.9) muestra la asignación y ocupación de pines del microcontrolador de forma gráfica.
4. Configuración de APPs y editor de código fuente: En este panel se pueden configurar las APPs (con la perspectiva DAVE CE) y ver o editar el código del proyecto (con las perspectivas DAVE IDE y CE).
5. Árbol de Dependencia de APPs: En este panel se puede observar la lista de APPs del proyecto y las APPs que dependen de cada una de ellas en una estructura de tipo árbol. Haciendo doble click en cualquiera de ellas se abre en el panel número 4 el entorno gráfico para su configuración.
6. Dependencia de APPs y Conexionado de las Señales de Hardware: Este panel aparece en la perspectiva DAVE CE. Ofrece dos representaciones importantes, la primera de ellas es la dependencia entre APPs, mientras que la segunda se trata de un esquema de conexionado de señales. Desde este panel se pueden añadir nuevas APPs al proyecto con el botón *Add New APP* (+).


Por otra parte, la vista del conexionado de hardware (*HW Signal Connectivity*) muestra de forma gráfica las conexiones formadas entre los recursos de diferentes APPs. Esta se muestra en la Figura 2.10.

Por último, hay dos opciones más para configuración de APPs mediante entorno gráfico importantes: Distribuidor Manual de Pines (*Manual Pin Allocator* ) y Asignación Manual de Recursos (*Manual Resource Assignment* )

La interfaz gráfica del *Manual Pin Allocator* tiene el aspecto de la Figura 2.11. En ella se pueden configurar manualmente los pines del Microcontrolador que necesitará cada APP dependiendo de su funcionalidad.

En la Figura 2.11 se observan los pines asignados para tres APPs diferentes que son ADC_MEASUREMENT_0, GUI_SEGGERLIBRARY_0 y SPI_MASTER_0, cuyas funciones son controlar el ADC, controlar la representación de datos en el display PMOLED y controlar el bus serie respectivamente. Se han reservado en primer lugar cuatro pines que se corresponden con cuatro canales diferentes del ADC para las medidas analógicas que se quieren hacer, el pin de la librería segger se configura por defecto y en el caso del canal SPI, al ser el encargado tanto de las entradas y salidas digitales como de la comunicación con la pantalla se reservan también los pines encargados del *chip select*.

El subíndice 0 en cada una de las APPs indica la instancia de la misma en el programa. En este caso es 0 y tan solo hay una instancia de cada una, sin embargo DAVE permite realizar más instancias de una misma APP dentro de un proyecto.

En la interfaz gráfica del *Manual Resource Assignment* se pueden configurar manualmente los recursos. El resultado final de esta asignación de recursos puede verse en *Report* () , y un ejemplo de su interfaz gráfica se muestra en la Figura 2.12. Aquí se pueden ver por ejemplo los recursos consumidos por el ADC, como canales utilizados, pines utilizados y eventos de resultado para cada medida.

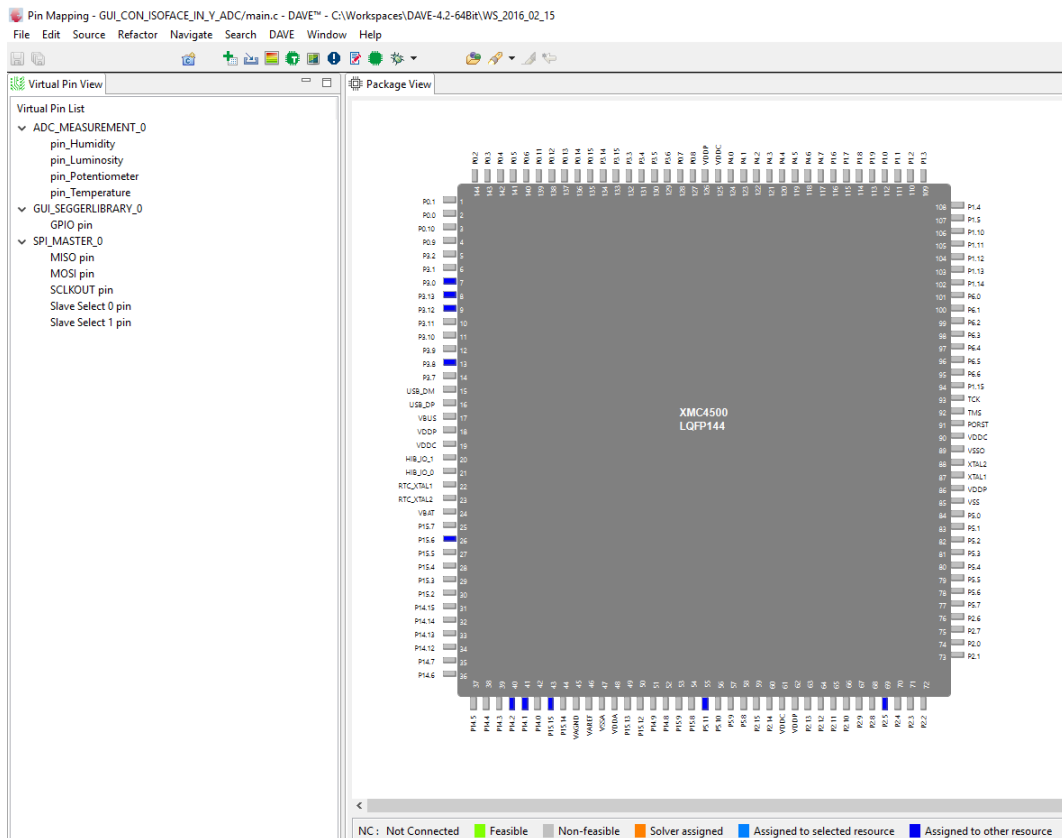


Figura 2.9: Perspectiva *Pin Mapping* (izq.) de este proyecto

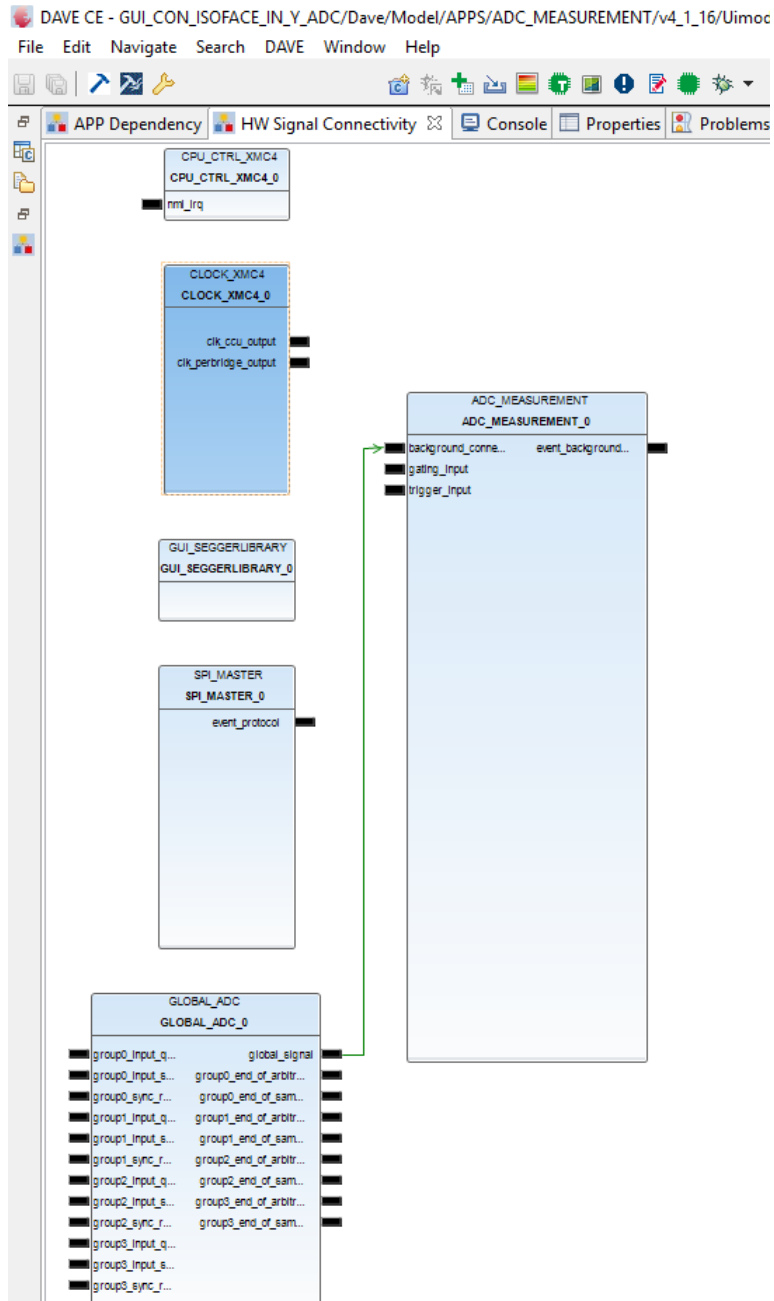


Figura 2.10: Pestaña de *HW Signal Conectivity* este proyecto

Manual Pin Allocator

Filter: ALL

APP Instance Name	APP Pin Name	Pin Number (Port)
▼ ADC_MEASUREMENT_0	pin_Humidity	#43 (P15.15)
	pin_Luminosity	#26 (P15.6)
	pin_Potentiometer	#41 (P14.1)
	pin_Temperature	#40 (P14.2)
▼ GUI_SEGGERLIBRARY_0	GPIO pin	#55 (P5.11)
▼ SPI_MASTER_0	MISO pin	#69 (P2.5)
	MOSI pin	#8 (P3.13)
	SCLKOUT pin	#7 (P3.0)
	Slave Select 0 pin	#9 (P3.12)
	Slave Select 1 pin	#13 (P3.8)

Figura 2.11: Interfaz gráfica del *Manual Pin Allocator* en este proyecto.

Report

Resource Mapping | Pin Allocator | Signal Assignment | APPs

type filter text

APP Instance Name	Resource	Mapped Resource
▼ ADC_MEASUREMENT_0	Background	vadc/0/backgnd
	Background class	vadc/0/class/0
	Humidity	vadc/0/group/3/ch/7
	Humidity Result	vadc/0/group/3/result/4
	Humidity_pin	p/15/pad/15
	Luminosity	vadc/0/group/2/ch/6
	Luminosity Result	vadc/0/group/2/result_a...
	Luminosity_pin	p/15/pad/6
	Potentiometer	vadc/0/group/0/ch/0
	Potentiometer Result	vadc/0/group/0/result_fil...
	Potentiometer_pin	p/14/pad/1
	Source Event Interrupt	cpu/0/nvic/interrupt/16
	Temperature	vadc/0/group/1/ch/2
	Temperature Result	vadc/0/group/1/result_a...
	Temperature_pin	p/14/pad/2
▼ CLOCK_XMC4_0	CCU	scu/0/clkctrl/0/ccu
	CPU	scu/0/clkctrl/0/cpu
	EBU	scu/0/clkctrl/0/ebu
	HIBERNATE	scu/0/pwrctrl/0/hibernate
	PERIBRIDGE	scu/0/clkctrl/0/perbridge
	PLL	scu/0/clkctrl/0/pll
	RTC	scu/0/rtccksel
	SDMMC	scu/0/clkctrl/0/sdmmc

Export...

Close

Figura 2.12: *Report* con todos los recursos utilizados en este proyecto.

3. Sistema de Adquisición de Parámetros Ambientales

3.1. Introducción

En este capítulo se va a explicar el montaje y funcionamiento del sistema de adquisición de datos que se ha desarrollado sobre la plataforma *Hexagon Application Kit* descrita en el capítulo anterior, en concreto se utilizarán las placas CPU_45A-V3, AUT_ISO-V1, HMI_OLED-V1 y UNI_EXT01-V2.

El sistema construido consta de cuatro sensores diferentes: Temperatura, Humedad, Luminosidad y Presencia; de los cuales, los tres primeros aportan una salida de voltaje analógica mientras que el último es un sensor digital. El *Hexagon Application Kit* no proporciona adaptación para ninguno de estos sensores, por lo que el primer paso para la realización del proyecto consiste en adaptar la salida de los mismos a los métodos y características de entrada, tanto del kit de desarrollo, como del microcontrolador. De ahí en adelante, se mostrará el método de captura, acondicionamiento y almacenamiento de los datos adquiridos, el procedimiento de visualización de dichos datos mediante la navegación por diferentes menús y la presentación por pantalla.

3.2. Sensores

El sistema que se montará para adquirir parámetros ambientales consta de cuatro sensores: Temperatura, Humedad, Luminosidad y Presencia. Cada uno tendrá unas características de entrada y salida y necesidades de alimentación diferentes. En este apartado se explican esas características y se muestra la manera buscada para adaptarlas a las características del Microcontrolador XMC4500 y del *Hexagon Application Kit*.

Tanto la alimentación de 5V como la de 24V necesarias se obtendrán de las salidas que proporciona la placa satélite Automation I/O Card tal y como se indicó en el Apartado 2.4.1.

3.2.1. Sensor de Temperatura (MCP9700A)

El sensor de temperatura MCP9700A [15] es un sensor analógico que aporta un valor de voltaje en la salida proporcional a la temperatura del ambiente en el que esté instalado. El rango de salida se encuentra entre 100mV y 1.75V para unas temperaturas de entre -40°C y 125°C respectivamente, donde la salida para 0°C son 500mV. Este sensor tiene una sensibilidad de 10mV/°C y un error de $\pm 2^\circ\text{C}$ para medidas entre 0°C y 70°C y de $\pm 4^\circ\text{C}$ para el rango de medida completo.

3.2.1.1 Adaptación del sensor al sistema

La salida se encuentra dentro del rango aceptado por el ADC, por lo que no es necesario colocar una red de adaptación a la salida. Como contrapartida se obtiene que se perderá una parte importante de la resolución del mismo, ya que las temperaturas a medir se encontrarán en un rango muy pequeño y el valor máximo de salida del sensor no coincide con el valor máximo de la entrada del conversor.

Teniendo en cuenta la resolución del ADC de 12 bits se obtiene la recta de ajuste de la Ecuación 1, necesaria para la representación de la temperatura mediante software.

$$T = \frac{V_{OUT} - offset}{sensibilidad} \quad (1)$$

Donde:

$$offset = \frac{4095}{3300} \cdot 500 \quad (2)$$

$$sensibilidad = \frac{4095}{3300} \cdot 10 \quad (3)$$

El esquema completo se muestra en la Página 27.

3.2.2. Sensor de Humedad (808H5V5)

El sensor de humedad 808H5V5 [18] es un sensor analógico capacitivo que proporciona un valor de voltaje de salida proporcional a la humedad relativa del ambiente (%RH) en la atmósfera.

Este sensor se alimenta con una tensión de 5V y la salida a 25°C se encuentra entre 0.8V y 3.9V para valores de humedad relativa del 0% y 100% respectivamente. Introduce un error inferior a $\pm 6\%$ RH en el rango completo de salida, mientras que este se reduce a menos de $\pm 4\%$ RH entre 30% RH y 80% RH a 25°C.

Al ser la humedad un parámetro dependiente de la temperatura ambiental, la recta de ajuste del voltaje de salida frente a la humedad relativa varía si existe un cambio en la temperatura. Para una medida más exacta, el fabricante aporta la Ecuación 4 para el voltaje de salida con compensación de la temperatura:

$$V_{OUT}(T, \%RH, V_{OUT}) = (5 \cdot 10^{-5} \cdot \%RH - 0.0036)T - 0.0013 \cdot \%RH + 0.0925 + V_{OUT} \quad (4)$$

La cual, teniendo en cuenta que la compensación a 25°C para 50% RH es 0, se puede reducir a la Ecuación 5 donde la tensión de salida solo depende de la humedad relativa y la tensión de alimentación:

$$V_{OUT}(\%RH, V_{IN}) = (0.0157 \cdot V_{IN} - 0.0475) \%RH - 0.6536 \cdot V_{IN} + 4.068 \quad (5)$$

3.2.2.1 Adaptación del sensor al sistema

Dadas las características del ADC del Microcontrolador que admite una entrada de hasta 3.3V y la salida del sensor de humedad, es necesario colocar una red de adaptación a la salida del mismo para no saturar el conversor. Para ello, se coloca un potenciómetro en configuración de divisor de tensión con el objetivo de obtener un valor máximo de tensión de 3.3V. Las medidas experimentales nos muestran unos valores máximos y mínimos de voltaje de 3.22V y 0.66V respectivamente, calculados a partir de los valores reales de las resistencias colocadas.

Esta adaptación supone un cambio en la Ecuación 5, ya que toda la expresión de V_{OUT} queda multiplicada por $\frac{R_2}{R_2+R_1} = \frac{212}{257}$ obtenido previamente de forma experimental. La expresión final a partir de la cual se extraerá el valor de la humedad relativa se muestra en la Ecuación 6.

$$\%RH = \frac{V_{OUT} \cdot \frac{257}{212} + 0.6536 \cdot V_{IN} - 4.068}{0.0157 \cdot V_{IN} - 0.0475} \quad (6)$$

Al igual que para la temperatura es necesario adaptar esta expresión a los 12 bits de resolución del conversor para poder calcular mediante software la humedad relativa, esta adaptación de la Ecuación 6 se muestra en la Ecuación 7:

$$\%RH = \frac{\frac{3300}{4095} \cdot V_{OUT_{digital}} \cdot \frac{257}{212} + 0.6536 \cdot V_{IN} - 4.068}{0.0157 \cdot V_{IN} - 0.0475} \quad (7)$$

Donde V_{IN} debe estar expresado en mV. El esquema de esta adaptación se muestra en la Página 27.

3.2.3. Sensor de Luminosidad (LDR)

El sensor de luminosidad consiste en un fotorresistor o LDR (*Light-Dependent Resistor*) [15], el cual varía su resistencia en función de la luz que incide sobre él. Este sensor presenta una resistencia en oscuridad de 20M Ω y una resistencia de entre 5k Ω y 20k Ω en la luz, cuando inciden sobre él al menos 10lux. El fabricante no recomienda la utilización de este sensor para medidas exactas, ya que indica que la saturación en presencia de luz es muy rápida, por lo que es más adecuado para distinguir entre «muchas luz» y «poca luz».

3.2.3.1 Adaptación del sensor al sistema

El LDR no es un sensor que produzca una salida en voltaje en sí, por lo que es necesario alimentarlo y medir la caída de tensión en algún punto de la red construida. Como se mencionó anteriormente, este sensor no sirve para obtener una medida exacta de la luminosidad del ambiente, por lo que el voltaje de salida medido tiene que ser transformado a un nivel subjetivo de mucha o poca luminosidad. Para que

esta medida sea ajustable, la carga sobre la que se medirá la caída de tensión está constituida por un potenciómetro ($R_n=103.7k\Omega$) tal y como se muestra en la Página 27. Realizando los cálculos del voltaje de salida que se medirá en la carga obtenemos los siguiente:

$$V_{OUT} = V_{IN} \cdot \frac{R_b}{R_b + R_{LDR} + R_a} = V_{IN} \cdot \frac{R_b}{100 + R_{LDR}} \quad (8)$$

Donde R_b es la resistencia de carga que variará según cómo se encuentre el cursor del potenciómetro, R_{LDR} es la resistencia del sensor de luminosidad en un momento determinado y R_a es la resistencia que queda entre el cursor y el terminal opuesto al de R_b .

Se puede observar que la relación entre la salida en voltaje y la resistencia marcada por el potenciómetro es lineal, por ello, ya sea cuando R_{LDR} es máxima ($20M\Omega$) o mínima ($\sim 5k\Omega$) la diferencia entre el voltaje máximo y mínimo (rango de salida) que se puede obtener será mayor cuanto mayor sea R_b . Por esta razón se ha fijado el valor del potenciómetro al máximo que puede ofrecer, su valor nominal $R_n=103.7k\Omega$.

Por otra parte, y de nuevo para no saturar el conversor con más de 3.3V, se reducirá la tensión que entra al LDR. Para aprovechar al máximo la resolución del ADC se ha colocado un potenciómetro a la entrada del LDR (Página 27), y se ha ajustado incidiendo el máximo de luz posible sobre el sensor para que a la entrada del ADC se disponga de 3.3V cuando el sensor está saturado. De esta forma, se ha medido que la tensión con la que se alimenta el LDR es 3.7V.

3.2.4. Sensor de Presencia (PIR)

El sensor de presencia o sensor PIR (*Passive Infra-Red*) [16] es un sensor piroeléctrico consistente en un receptor de infrarrojos y una lente de enfoque. Este sensor aporta una salida digital que establece el nivel de salida en alto (entre 3V y 6V coincidente con la alimentación del sensor) cuando detecta variaciones en la cantidad de radiación infrarroja recibida.

3.2.4.1 Adaptación del sensor al sistema

Dadas las características de las entradas digitales, las cuales solo admiten voltajes superiores a 12V para detectar un 1 lógico, es necesario realizar una red de adaptación a la salida del sensor de presencia, ya que este nos aportará en su pin de salida como máximo el valor de su alimentación, es decir, 5V. Con el objetivo de utilizar las alimentaciones de 5V y 24V que proporciona la placa se ha diseñado un circuito como el de la Página 27 para obtener un voltaje superior a 12V cuando el sensor proporcione la salida alta y menor que 12V cuando el sensor proporcione la salida baja.

El circuito optoacoplador 4N25 [19] de la Página 27 se utiliza para aislar las dos alimentaciones y tierras de mayor y menor voltaje, evitando así cualquier interferencia o lazo de tierra que pudiera introducirse, bien en cualquiera de las placas o bien en el propio microcontrolador pudiendo producir una sobretensión y causando un mal funcionamiento del equipo.

Para poder obtener en la salida el voltaje deseado hay que tener en cuenta la impedancia de entrada que presentan las entradas digitales de la placa, ya que estas se colocarán en serie con la resistencia que de colector del fototransistor en el optoacoplador; el esquemático de estas entradas puede verse en la Figura 3.1. Por esta razón se realizó la medida experimental de la impedancia de entrada dando un resultado de $9.23k\Omega$, por lo que la resistencia de colector ha de ser inferior a $9.23k\Omega$ si se quiere tener un voltaje superior a la mitad de la alimentación de 24V.

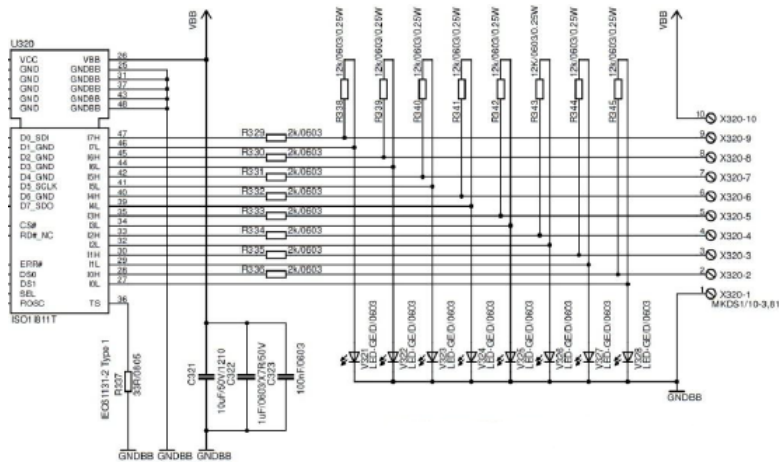


Figura 3.1: Esquemático de las entradas digitales de la placa AUT_ISO-V1 [9].

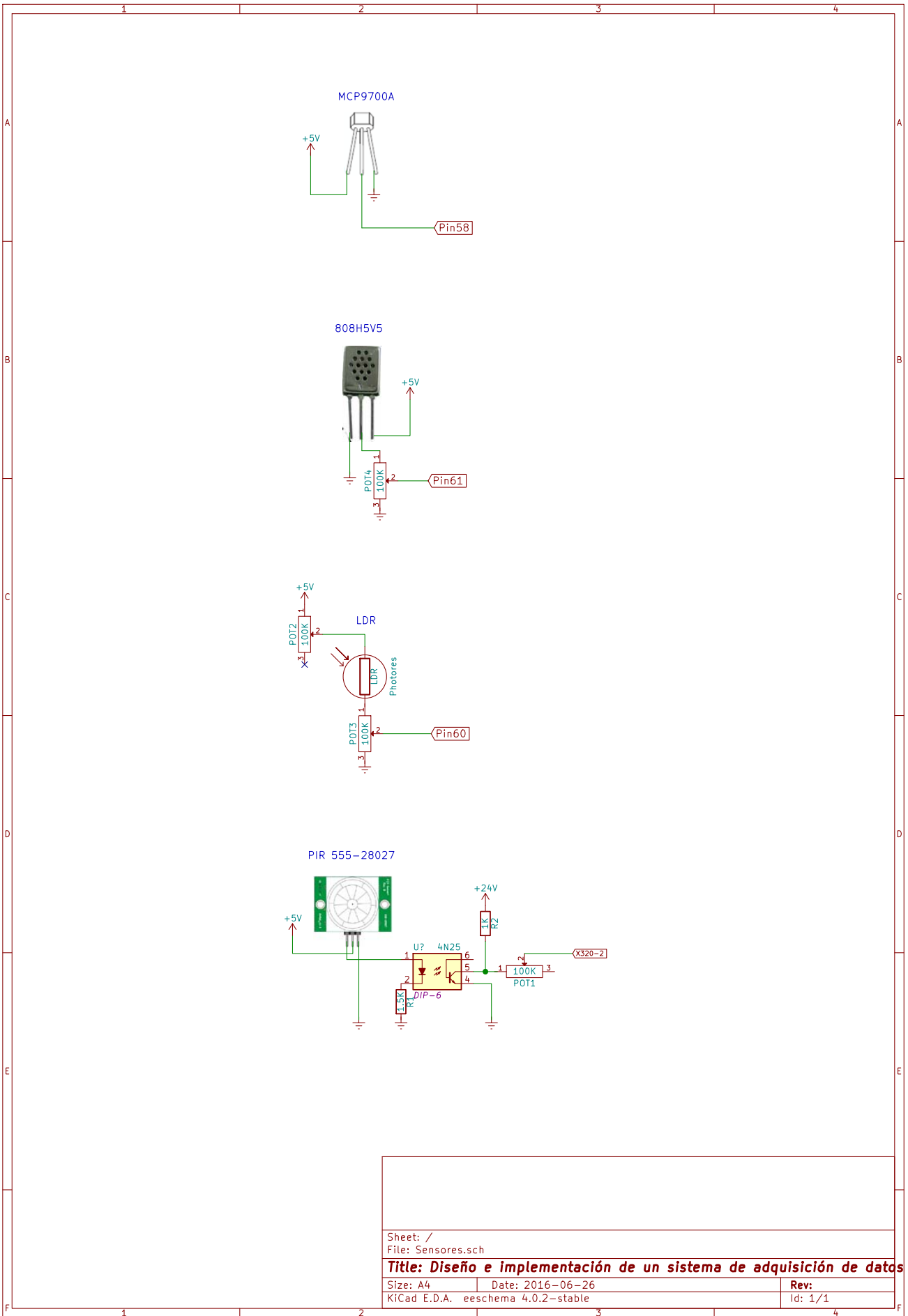
Por otra parte, con el objetivo de no someter al sensor PIR a una corriente excesiva se coloca en el pin 2 del optoacoplador una resistencia de $1.5k\Omega$. Esto supone que el diodo emisor de luz no funcione al 100 % de su capacidad, resultando en que el fototransistor no trabaje en la zona de saturación cuando la salida del sensor está en 5V. Al no trabajar en saturación, es necesario añadir a los cálculos una resistencia en paralelo a la impedancia de entrada calculada anteriormente, que resulta ser lo suficientemente alta como para perturbar la medida y que se obtenga un voltaje mayor que 12V, lo cual para este sistema equivale a un 1 digital en lugar del 0 que se busca.

Para solucionar este problema, se ha colocado un potenciómetro en serie con la resistencia de colector de $1k\Omega$ como se muestra en la Página 27 y se ha ajustado para que la resistencia en serie de ambos sea mayor que el paralelo de las otras dos. La medida experimental de esta resistencia mostró un valor de $3k\Omega$.

En resumen, el valor de la resistencia en serie colocada en el colector debía ser:

$$R_S \in (3, 9.23)k\Omega$$

Ajustado de forma manual, el valor final de la resistencia en serie es de $3.72k\Omega$.



Sheet: /	
File: Sensores.sch	
Title: Diseño e implementación de un sistema de adquisición de datos	
Size: A4	Date: 2016-06-26
Rev:	Id: 1/1
KiCad E.D.A. eeschema 4.0.2-stable	

3.3. Adquisición de datos digitales

Para poder adquirir los datos del sensor digital es necesario conocer las características de las entradas digitales de la placa, que como ya se mencionó en anteriormente (Apartado 2.4.1) permiten un rango de voltaje de entrada de 12V a 24V, considerando cualquier valor inferior a 12V un 0 lógico. Para que el sistema ISOFACE pueda funcionar es necesario alimentar con 24V la última de las entradas digitales que se muestran en la Figura 2.3. Este voltaje se puede obtener de la última salida digital que aporta una salida constante de 24V, siempre y cuando la placa se encuentre alimentada por el *Jack* del que dispone.

Por otra parte, para poder realizar la adquisición y tratamiento de datos a través del Software es necesario conocer el funcionamiento de la interfaz de estas entradas con el XMC4500, es decir, el Canal Serie Síncrono o SSC (*Serial Sincronous Channel*).

3.3.1. Adquisición física de los datos: Canal Serie Síncrono

A continuación se describe el funcionamiento de este canal y sus principales características configurables.

3.3.1.1 Funcionamiento

El SPI (*Serial Peripheral Interface*) es uno de los protocolos configurables para los canales del módulo USIC. La APP de DAVE **APP SPI_MASTER** [6] permite configurar las principales características de los modos de funcionamiento de este canal, que se muestran a continuación:

1. Tasa de baudios para la comunicación: Desde 0.1 hasta 60000kHz es posible seleccionar la velocidad de transmisión de baudios.
2. Modo de transferencia de datos: El modo de operación del maestro SPI es seleccionable entre cuatro posibles (Figura 3.2):
 - Standard Full Duplex: En este modo la APP utiliza dos pines diferentes para la transmisión (MISO) y la recepción (MOSI), permitiendo la comunicación bidireccional paralela.
 - Standard Half Duplex: Únicamente se permite la configuración del pin MISO.
 - Dual SPI: Se utilizan 2 pines para establecer una conexión Half Duplex.
 - Quad SPI: En este caso se utilizan 4 pines para establecer una conexión Half Duplex.

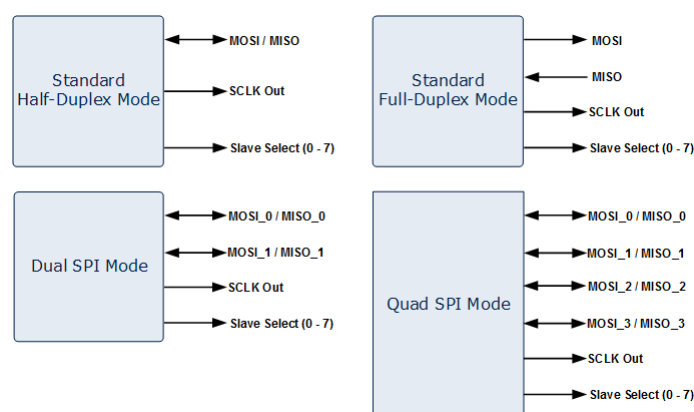


Figura 3.2: Modos de transmisión soportados por el SPI [6].

3. Modo de Transmisión y Recepción: Hay 3 mecanismos de transmisión y recepción de datos disponibles:
 - Interrupción: Se utilizará un nodo dedicado del módulo de interrupciones NVIC para transmitir o recibir datos.
 - Acceso Directo a Memoria: Hay un canal DMA dedicado para la transmisión/recepción de datos.
 - Directo o Polling

4. Soporte para la conexión de diferentes esclavos: Existe la posibilidad de elegir hasta 8 líneas de selección de esclavo.
5. Dirección de la transmisión: Existe la posibilidad de elegir el orden de los bits transmitidos o recibidos entre la transmisión/recepción, primero MSB o primero LSB.
6. Configuración de la longitud de *Frame* y *Word*: La longitud del *Frame* (FLE) es seleccionable desde 1 hasta 64. Si FLE es inferior a 64 se genera la señal de selección de esclavo o *chip select* (CS) automáticamente con el primer bit de la trama y se desactiva con el final del último bit transmitido incluyendo el *delay*. Si FLE es 64, el usuario deberá aportar la información de comienzo y final de una trama. Para este proceso es necesario trabajar a nivel de registro con los bits SOF y EOF. En cuanto a la longitud de *Word*, es posible seleccionar desde 1 a 16 los bits transmitidos o recibidos como un solo *Word*.
7. Ajustes de *delay* e inversión de polaridad del *clock* para diferentes rangos de esclavos: Define los ajustes del nivel pasivo de clock y el flanco para el cual tiene que llevarse a cabo la transmisión de los datos. Hay cuatro combinaciones posibles, dos para el nivel de clock bajo cuando la transmisión está en estado pasivo o inactivo, y dos para cuando ese nivel de clock está en alto, que se muestran en la Figura 3.3. En cuanto al *delay*, es posible configurar dos tipos diferentes de *delay* en μs o en ciclos de reloj.

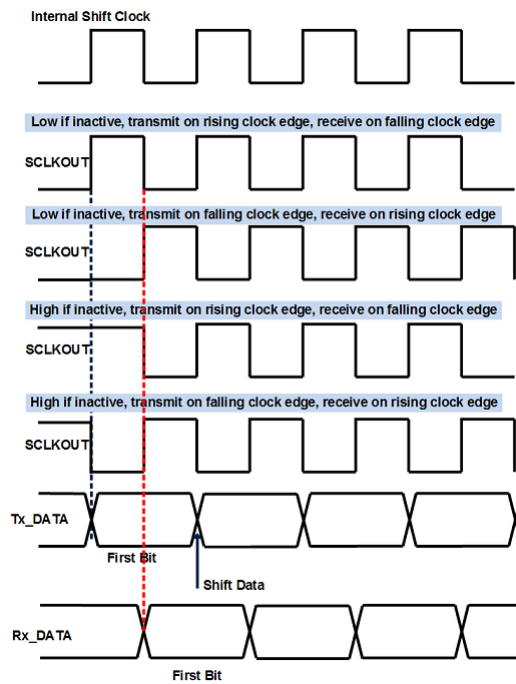


Figura 3.3: Configuraciones disponibles del *clock* para la transferencia de datos [6].

8. Cola FIFO con tamaño variable para transmisión y recepción: El usuario puede activar y desactivar estas colas y determinar su tamaño en número de *Words* que pueden contener, desde un mínimo de 2 hasta un máximo de 32.
9. Selección del modo Parity Mode para la validación de los datos: Consiste en añadir un bit a final de cada *Frame* y puede ser configurado para que marque una trama par o impar.
10. Mecanismo de *callback* para gestionar los errores de Paridad, Transmisión y Recepción: Existe la posibilidad de habilitar/deshabilitar tres funciones *callback* para su ejecución cuando se den interrupciones de tipo: Fin de Transmisión, Fin de Transferencia/Recepción y Error de Paridad. La primera de ellas indica que todos los datos en el *buffer* de usuario han sido colocados en el *buffer* de transmisión, la segunda sí garantiza que los datos han sido recibidos o enviados, y la última se ejecuta cuando la paridad de los datos no coincide con el bit extra de paridad. Es posible seleccionar un nivel de prioridad entre 0 y 63.

3.3.1.2 Configuración del SSC mediante la GUI de DAVE

En primer lugar, hay que configurar la APP que permitirá utilizar el bus SPI para interactuar tanto con las entradas digitales como con la pantalla. Es por esta razón por lo que es imprescindible seleccionar dos líneas de selección de esclavo y asignarles los pines correspondientes del Microcontrolador a cada una de ellas. Todas las pestañas de configuración de los parámetros explicados anteriormente se muestran en la Figura 3.4 y la configuración de los pines para entrada (MOSI) y salida (MISO) de datos, clock (SCLKOUT) y *chip select* (CS) en la Figura 3.5. Cabe recordar que con el fin de poder controlar el mayor número de periféricos posibles a través del bus SPI con la menor asignación de pines del XMC4500 posible, el único pin que es particular de cada periférico es el *chip select*.

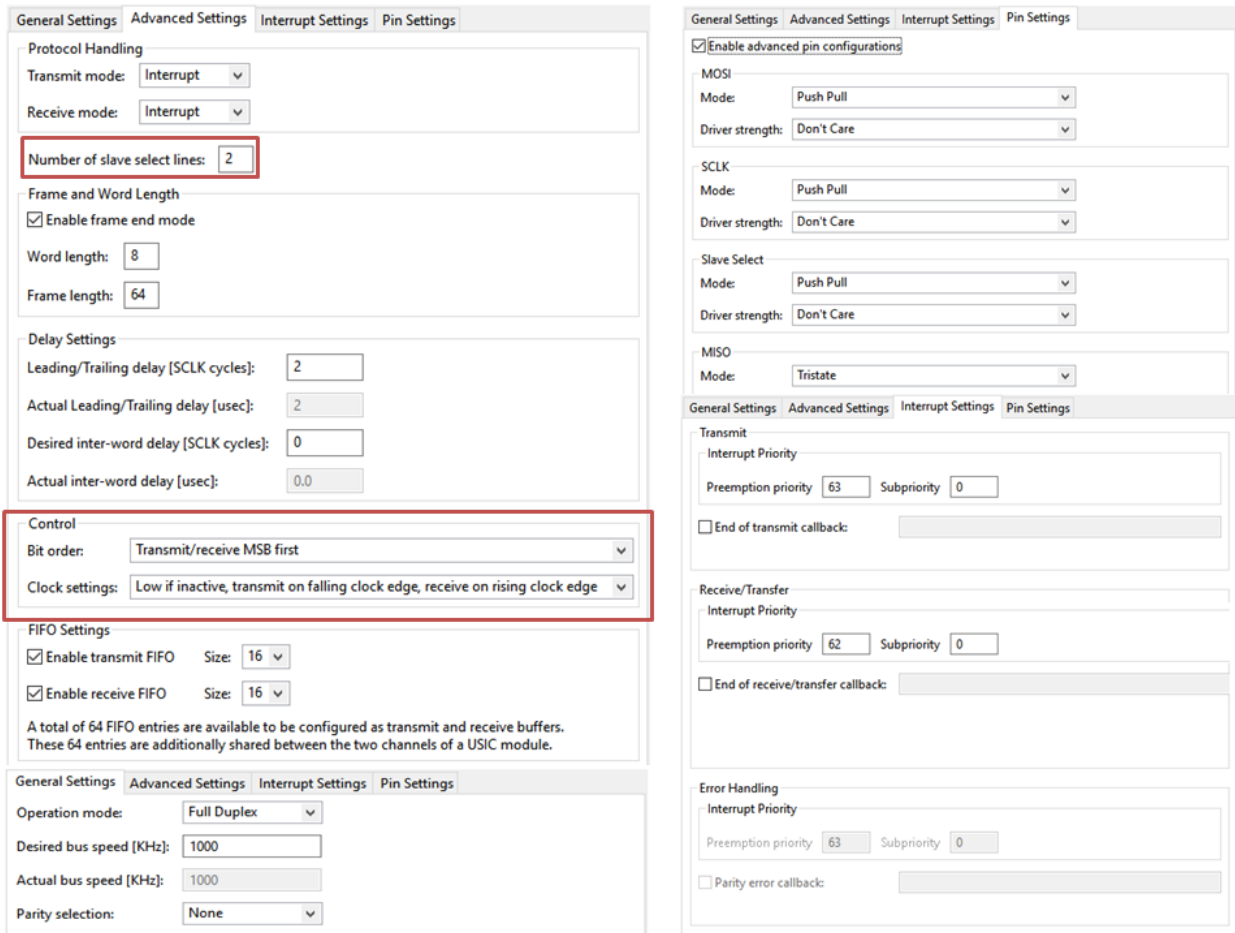


Figura 3.4: Interfaz gráfica para las configuraciones de la interfaz SPI realizada en este proyecto.

SPI_MASTER_0		
MISO pin	#69 (P2.5)	▼
MOSI pin	#8 (P3.13)	▼
SCLKOUT pin	#7 (P3.0)	▼
Slave Select 0 pin	#9 (P3.12)	▼
Slave Select 1 pin	#13 (P3.8)	▼

Figura 3.5: Asignación de pines para la interfaz SPI realizada en este proyecto.

La comunicación de ambos dispositivos se realiza en Full Duplex, y como en este caso no se va a trabajar con interrupciones, no hay que inicializar ningún *callback*. La segunda de las opciones resaltada en la Figura 3.4, se corresponde con el orden de los bits, que por defecto es MSB, y con los ajustes de *clock*. El primer ajuste se quedará invariable, sin embargo, el segundo se deberá modificar por software, pues la recepción y transmisión, o dicho de otra forma la lectura y escritura, se hacen con los ajustes de *clock* inversos.

En ambos se mantiene que la transmisión (Tx) se realiza en el flanco descendiente y la recepción (Rx) en el flanco ascendiente, sin embargo, se trabaja con el nivel pasivo en alto cuando se leen los datos digitales en las entradas, y con el nivel pasivo en bajo cuando se escriben datos en la pantalla. Es decir, nada más se establece la conexión, en el caso de las entradas digitales la primera acción será Tx, y en la pantalla por el contrario, ejecutará Rx. (Figura 3.6)

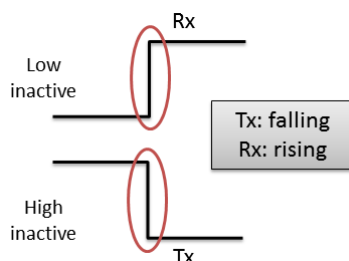


Figura 3.6: Transmisión y recepción de datos para diferentes referencias en el nivel inactivo del *clock*.

3.3.2. Tratamiento de los datos mediante Software

La sección principal de código que se ha de ejecutar en el *main* del programa para la adquisición de datos se muestra en la Figura 3.7. Con el objetivo de obtener el mismo número de medidas analógicas y digitales, no se hará la adquisición digital si antes no ha habido una analógica. Por lo demás, hay que invertir el *clock* dos veces, una para medir y de nuevo para devolverlo a su configuración original.

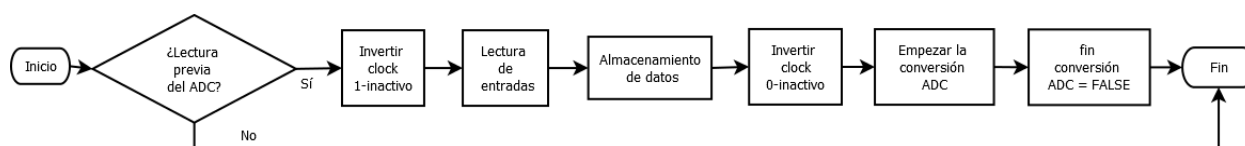


Figura 3.7: Diagrama de flujo de la rutina implementada para la lectura de datos digitales [CapturaDatos()].

Para el almacenamiento de los datos obtenidos se crea una estructura, cuyos elementos principales son el valor digital actual y el conteo total de transiciones de detección a no detección, que para esta configuración de entrada es el paso de un 1 digital a un 0.

El objetivo de guardar estos valores en la estructura es poder mostrar por pantalla el estado actual del sensor y el cambio de un estado a otro además de el número de veces que ha ocurrido este cambio. La utilización de los miembros de la estructura para la representación y el propio software de representación se explicará en la sección de visualización de los datos (Apartado 3.6).

3.4. Adquisición de datos analógicos

El Conversor Analógico-Digital del XMC4500 es el encargado de adquirir los datos analógicos. Para poder controlar y configurar su comportamiento es necesario conocer el funcionamiento de la APP que lo controla y realizar el conexionado correcto de acuerdo con la configuración que se establezca a través de la interfaz gráfica de DAVE.

3.4.1. Funcionamiento, configuración y conexionado

3.4.1.1 Funcionamiento

La APP de DAVE `ADC_MEASUREMENT` [6] permite configurar las principales características del ADC y proporciona las siguientes funcionalidades, que pueden dividirse en tres:

1. Medida de una secuencia lineal de entradas analógicas: Los nombres configurados en la pestaña «Measurement Table» de la GUI de la APP se asignan a canales individuales del periférico VADC bajo la petición de muestreo en segundo plano. Este tipo de muestreo se realiza sin necesidad de utilizar ciclos de la CPU para ello, y es automático para todos los canales. Los canales que se le asignen

a esta petición de muestreo pueden ser de cualquier módulo de los cuatro disponibles, siempre y cuando estén asociados correctamente con el pin correspondiente de entrada al Microcontrolador. Se realizará la conversión en todos los canales asignados siguiendo un orden fijo y lineal, desde el canal más alto (7) del grupo más alto (3) hasta el canal más bajo (0) del grupo más bajo (0). Se puede observar un ejemplo de este orden en la Figura 3.8.

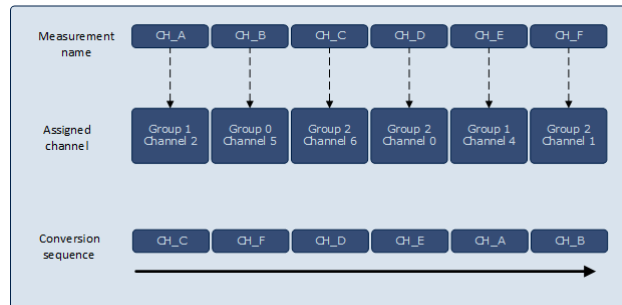


Figura 3.8: Ejemplo del orden muestreo de canales seguido por el ADC [6].

2. Modo de inicio de la medida: Para poder comenzar una conversión la APP proporciona cuatro opciones:

- Modo de Único Disparo por Trigger de Hardware
- Modo Continuo por Trigger de Hardware
- Modo de Único Disparo por Trigger de Software
- Modo Continuo por Trigger de Software

Los modos de comienzo por Software necesitan la API `ADC_MEASUREMENT_StartConversion()` para que empiece la conversión en los canales asignados. También puede ser disparado por Software el comienzo seleccionando la opción «Start conversion after initialization» (Comenzar la conversión tras inicializar) en los ajustes generales de la APP.

Los modos de comienzo por Hardware necesitan una señal de trigger externo para empezar las conversiones en los canales asignados. Este modo puede habilitarse desde los ajustes generales de la APP fijando el campo «Trigger Edge Selection» (Selección de Flanco del Trigger) a una de las opciones Rising/Falling/Both edges (Ascendente/Descendiente/Ambos flancos). Existe la posibilidad de que la señal venga de otras APPs como la de PWM.

Con ambos modos se puede configurar el ADC para que muestree las entradas de forma repetida habilitando la opción «Enable continuous conversion». Si el modo seleccionado es el de Único Disparo, la siguiente conversión (o secuencia de conversiones) deberá ser disparada de nuevo, ya sea mediante Hardware o Software. En la Figura 3.9 se muestran gráficamente los cuatro modos de inicio de conversión.

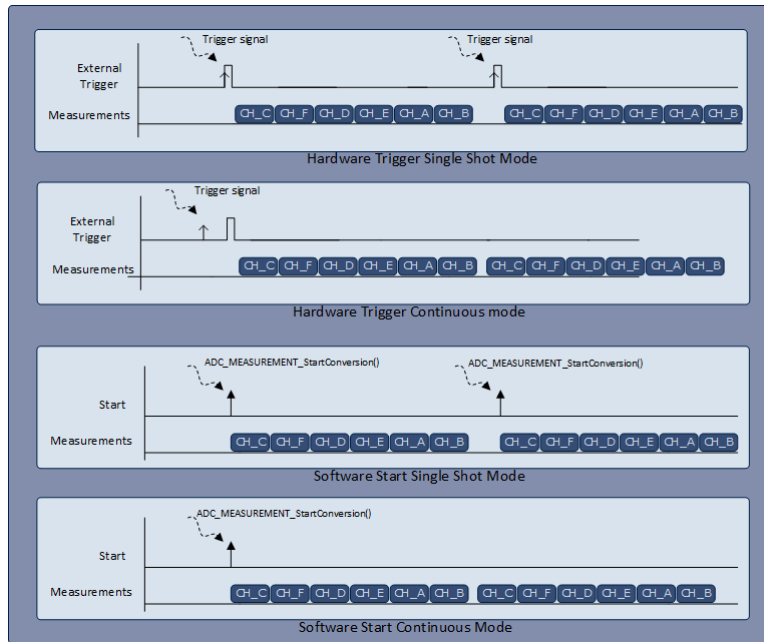


Figura 3.9: Modos de conversión para el ADC disponibles [6].

3. **Notificaciones de Interrupción:** Si están activadas, se genera una interrupción después de que se haya completado la medida de todos los canales una sola vez, es decir, tras una secuencia de conversiones. Las interrupciones se habilitan en la pestaña «Interrupt Settings». Además, de forma adicional es posible activar las interrupciones de resultado que se generan después de una medida en cada canal. Esta opción se activa habilitando la opción «Result event» en los ajustes. El sistema de interrupciones puede observarse de forma gráfica en la Figura 3.10.

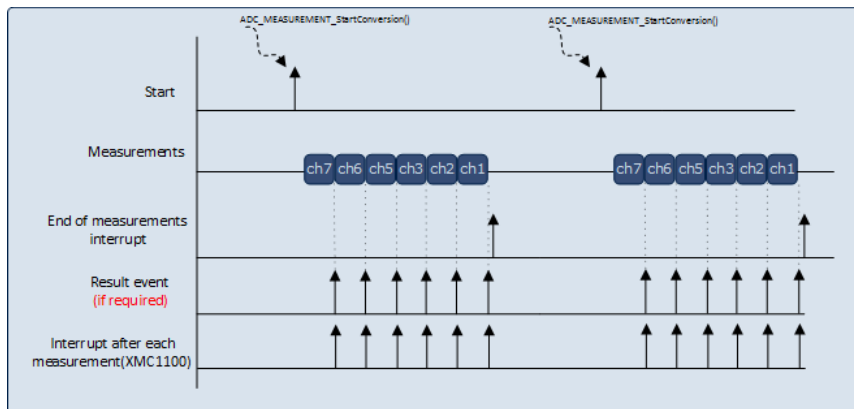


Figura 3.10: Tipos de interrupciones disponibles generadas por el ADC [6].

3.4.1.2 Configuración del ADC mediante la GUI de DAVE

La configuración que se ha fijado para los diferentes aspectos del ADC se muestra en la Figura 3.11. Aquí se puede ver que se ha seleccionado la opción de 4 medidas, por lo que nos permite elegir 4 nombres mediante los cuales se podrá acceder a los resultados almacenados en los *buffers* del canal correspondiente. Se ha fijado la resolución del ADC a 12 bits e inhabilitado la conversión continua y el trigger externo. Se controlará por Software el inicio de cada conversión, empezando la primera nada más inicializarse las APPs. Por último, se ha elegido el nombre de la función de gestión de la interrupción y su prioridad.

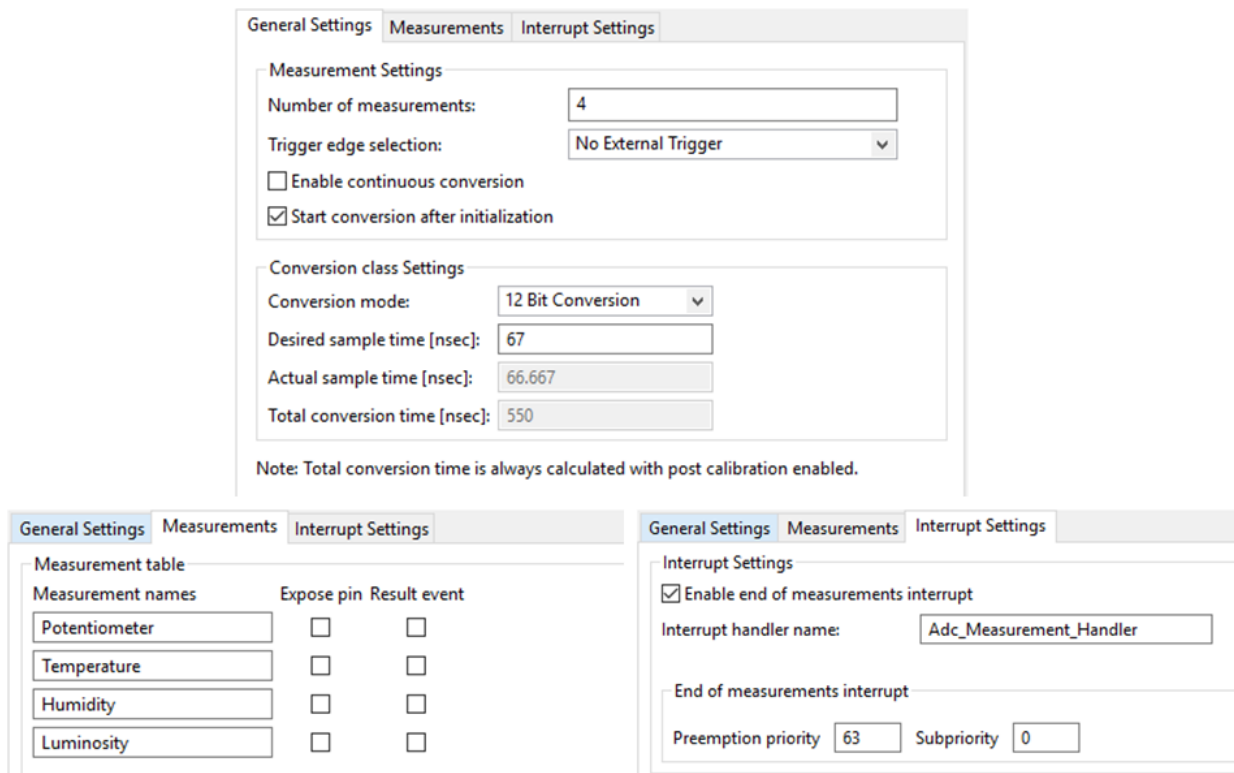


Figura 3.11: Interfaz gráfica para las configuraciones del ADC realizada en este proyecto.

Por otra parte, los pines a los que se han conectado físicamente los sensores hay que configurarlos como pines de entrada (Figura 3.12).

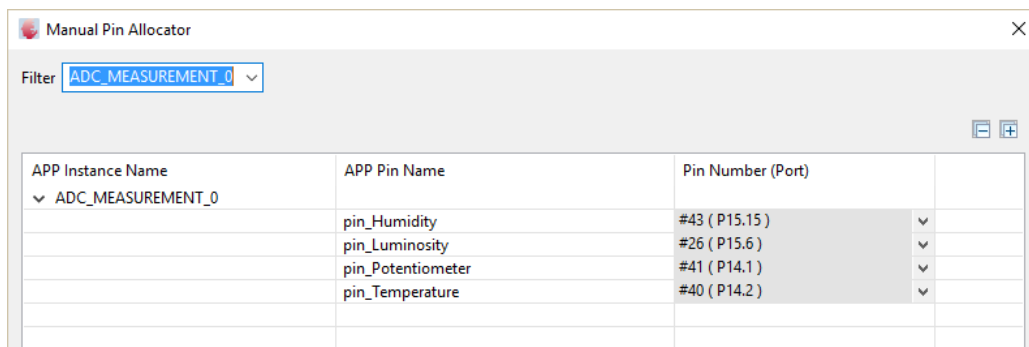


Figura 3.12: Asignación de pines para los diferentes sensores conectados al ADC realizada en este proyecto.

3.4.1.3 Conexión físico de las entradas

Para poder adquirir los datos de los tres sensores analógicos (Temperatura, Humedad y Luminosidad) y del Potenciómetro que se utilizará para la navegación por las opciones del menú (Apartado 3.6), es necesario conectar las salidas de estos sensores a las entradas de los canales del ADC. El potenciómetro integrado en la placa CPU_45A-V3 está directamente conectado al pin 14.1 del Microcontrolador que se corresponde con el canal 1 del módulo 0 del ADC. Para el resto se utiliza la placa de extensión de pines UNI_EXT01-V2 colocada entre la placa de la CPU y la de Entradas/Salidas digitales AUT_IS0-V1. Se ha elegido esta posición porque es una de las placas con menor número de pines reservados y cuyo conector satélite tiene el mayor número de accesos a los canales del ADC. Se han elegido los pines del extensor nº58, 60 y 61 para los sensores de Temperatura, Luminosidad y Humedad respectivamente, los cuales corresponden con los canales 2, 6 y 7 de los módulos 0, 2 y 3 del ADC respectivamente.

El montaje completo del sistema se muestra en la Figura 3.13.

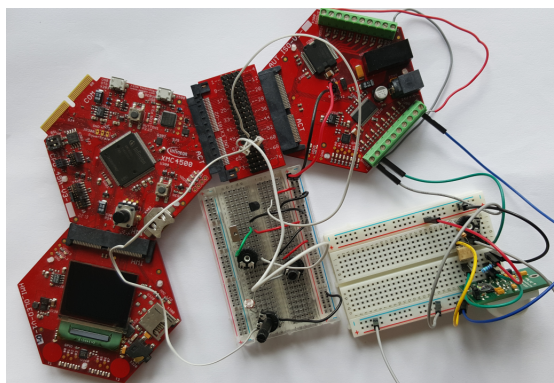


Figura 3.13: Montaje completo del sistema de adquisición de datos.

3.4.2. Tratamiento de los datos mediante Software

Como la resolución de ADC se ha configurado a 12 bits, esto nos permite obtener 4096 valores para el rango de entrada de 0 a 3.3V y apreciar un cambio de voltaje de 0.80567mV. En la Ecuación 9 se muestra de forma general como hallar el valor digital correspondiente a cualquier valor analógico que se pretenda medir:

$$valor\ discreto = \frac{4095}{3300} \cdot valor\ analógico \quad (9)$$

Hay que tener en cuenta esta conversión y la inversa en función de los datos que se quieran almacenar. El almacenamiento de datos se hace en la durante la gestión de la interrupción, y al igual que para los datos digitales, estos se guardan en una estructura dedicada para cada sensor:

- Sensor de Temperatura: En primer lugar, con el objetivo de mostrar por pantalla resultados estables y sin grandes oscilaciones, se ha implementado un miembro en la estructura de datos de este sensor para ir almacenando una serie de valores instantáneos de temperatura; de los cuales se hará la media y se mostrará por pantalla como valor actual.

En segundo lugar, se implementa una cola circular, a la cual se le añade un valor cada vez que se hace la media del conjunto de valores instantáneos. Esta cola se implementa con el objetivo de poder representar en una gráfica los valores que se han ido obteniendo en forma de histórico de medidas.

Por último, cada vez que se extrae una temperatura media, se compara con los valores máximo y mínimo almacenados previamente para poder mostrar los extremos medidos por el sensor posteriormente.

- Sensor de Humedad: La estructura y el almacenamiento de datos del sensor de Humedad es análogo al del sensor de Temperatura, ya que se quieren realizar las mismas representaciones por pantalla. La única diferencia es el tratamiento de datos, ya que para guardar la humedad relativa hay que utilizar la Ecuación 7.
- Sensor de Luminosidad: Las especificaciones de este sensor no recomiendan su uso para realizar una medida exacta de los lux del ambiente por lo que se utilizará este sensor para mostrar por pantalla el nivel relativo de luminosidad respecto a un punto fijado por el usuario. Para esto se ha optado por realizar un proceso de calibración en un punto que el usuario pueda seleccionar como luminosidad máxima o mínima.

Los datos que se necesitan almacenar en la estructura en este caso son el valor analógico de los voltios de entrada (Ecuación 9) y de nuevo, con el objetivo de realizar una representación gráfica con las menores oscilaciones posibles, se recogerán varios valores instantáneos y se trabajará con la media de estos.

Todas las funciones gráficas necesarias para la representación en diferentes formatos de los datos se explican en el Apartado 3.6.

3.5. Interfaz de Usuario

Como se verá en el apartado siguiente, el usuario puede elegir entre las diferentes representaciones de datos mediante un sistema de Menús y SubMenús que se ha desarrollado. En este apartado se describen los elementos de la placa HMI_OLED-V1 y CPU_45A-V3 que se utilizarán para realizar esta selección y navegación por los Menús.

En particular, para poder avanzar y retroceder hacia los diferentes Menús y SubMenús se hará uso de los botones táctiles integrados en las placa HMI_OLED-V1 del *Hexagon Application Kit*, mientras que para seleccionar una u otra opción, se utilizará el Potenciómetro integrado en la CPU_45A-V3. A continuación se describe el funcionamiento y configuración dada para ambos en este proyecto.

3.5.1. Interfaz Humano-Máquina (HMI): Pulsadores Táctiles (*Touch Pads*)

El módulo LEDTS puede medir la capacitancia de hasta 8 pulsadores táctiles utilizando una topología de oscilador a relajación. Esta capacitancia se mide generando oscilaciones en el propio pulsador durante un periodo fijo de tiempo y contándolas. Este módulo también puede controlar hasta 64 LEDs colocados en forma matricial. Con el objetivo de reducir el número de pines necesarios para este tipo de aplicaciones, los sensores táctiles y los LEDs comparten pines. La configuración de las líneas y columnas activas en cada momento la realiza el módulo controlando los pulsadores capacitivos y los LEDs por multiplexación por división de tiempo. Cada una de estas líneas puede compartirse entre 8 LEDs y un pulsador, esta configuración matricial puede observarse en la Figura 3.14. Ciertas funciones como la activación de columnas y la selección y control de funciones se controlan vía Hardware, mientras que otras como la actualización de las líneas de LEDs y la evaluación de los resultados de las medidas del sensor táctil se realizan por Software [13].

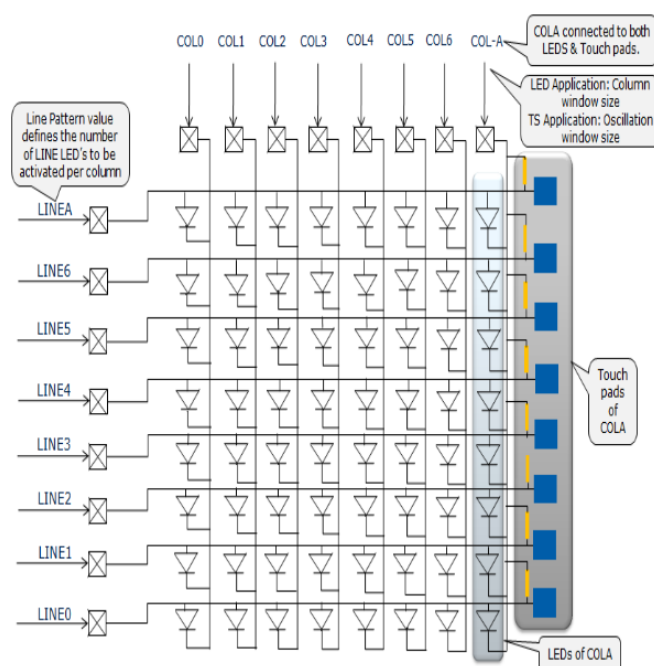


Figura 3.14: Ilustración de la configuración y modos de control de los LEDs y almohadillas táctiles [6].

El único paso a realizar para poder acceder a los datos medidos por los botones, es registrar una función *callback* que nos permita recoger los datos cuando se produzca una interrupción por detección de toque. Durante la ejecución del código de gestión de esta interrupción hay que acceder a los miembros de la estructura de datos donde se guardan, tanto el número de botón que se ha pulsado como el estado de esa pulsación. Solo se almacenará el dato del número del botón pulsado si el estado de la pulsación es válido.

3.5.2. Método de selección de opciones

La función que se ha implementado para saber si el usuario ha decidido avanzar al siguiente menú, retroceder al anterior o desplazarse por el mismo consiste en detectar la pulsación de los botones y el

movimiento del potenciómetro.

Cuando se detecta una pulsación en el botón T1 la función devuelve una variable (P_ESCAPE) que se interpretará como la acción Volver al anterior menú. Si se detecta una pulsación en T2 se interpreta como la acción Avanzar al siguiente menú (P_ENTER).

En cuanto al movimiento del potenciómetro, la función devuelve la variable asociada al movimiento relativo del mismo. P_UP si ha aumentado y P_DOWN si ha disminuido. Cuando no ocurre nada, la función devolverá P_IDLE.

3.6. Visualización de los datos

Para la visualización de los datos se utilizará la pantalla PMOLED (Figura 2.4), que se controla al igual que las entradas, mediante el bus SPI. La configuración de esta interfaz ya se fijó anteriormente por lo que el único ajuste del que hay que preocuparse es seleccionar el pin 3.12 como *chip select* (Figura 3.5). La configuración del *clock* establecida en la Figura 3.4 ya es la adecuada para la escritura en la pantalla. El *clock* se configura así porque la API de inicialización de DAVE necesita esta configuración para inicializar la APP encargada de la pantalla (GUI_SEGGERLIBRARY). En todo momento en la ejecución del programa estará establecida esta configuración de *clock*, excepto cuando cuando se esté accediendo a las entradas digitales.

Relacionado con la representación, es necesario indicar que la pantalla tiene una dimensión de 160 píxeles de ancho por 128 píxeles de alto, y que los ejes de coordenadas de píxeles son positivos hacia abajo con el origen de coordenadas en la esquina superior izquierda.

Todas las APIs para la escritura y dibujo están proporcionadas por la empresa Segger [17] y son externas al resto de APIs que proporciona Infineon, es por ello que hay que incluirlas en proyecto y enrutarlas para el *Debugger*.

3.6.1. Navegación por los Menús

Para llevar a cabo la representación de datos por pantalla se han construido dos estructuras: MENU_t y OPCION_t. Los Menús son aquellas pantallas donde es posible elegir entre varias opciones y pulsando el botón de Avanzar, podemos pasar a la representación de otro Menú donde se nos permite elegir entre más opciones, o bien llegar a una pantalla de tipo Opción. Estas pantallas ya muestran el valor que se quería observar, y la única posibilidad disponible en ellas es la de pulsar el botón Volver y regresar a un Menú anterior. La representación de este proceso se muestra en la Figura 3.15.

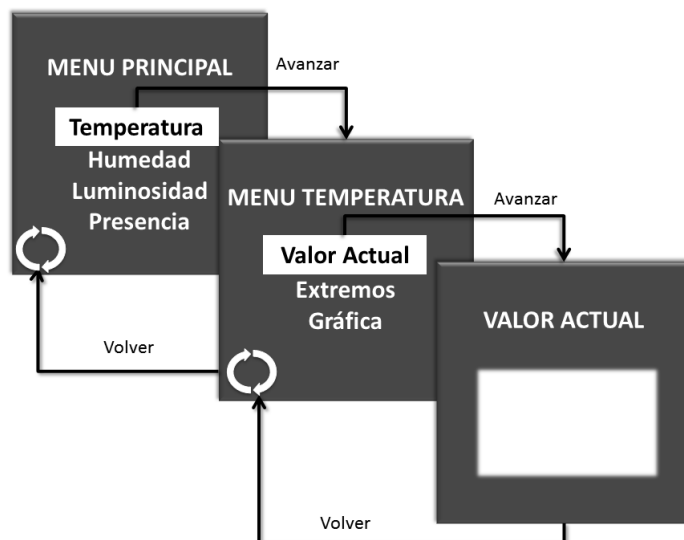


Figura 3.15: Ilustración del salto entre SubMenús y Opciones y representación cíclica de los mismos.

Los miembros más importantes de la estructuras de Menú y Opción, para poder presentar los datos por pantalla y para poder acceder a Submenús más profundos se muestran a continuación:

MENU_t	OPCION_t
/*Punteros a las opciones del menú*/ *Opciones[] /*Señala la opción activa*/ OpcionActual /*Indica si es el menú principal*/ PrimerMenu /*Flag para borrar la pantalla*/ Refresh /*Título del Menú*/ *Titulo /*Nº opciones disponibles*/ numeroOpciones	/*SubMenú dependiente de la opción*/ *SbMenu /*Función con argumento void*/ (*fOpcion)(void *datos) /*Parámetros para la función*/ *Parametros /*Flag para borrar la pantalla*/ Refresh /*Título de la opción*/ *Titulo /*Coordenadas de representación*/ (X,Y)

Dentro de la estructura Opción, hay un elemento que es un puntero a una función. Esta función se implementa para que realice la representación adecuada y cada opción tiene la suya. Solo se implementan funciones para las opciones de los Submenús que vayan a mostrar un resultado por pantalla. La distribución de Menús y SubMenús del programa con sus diferentes opciones de forma jerárquica es la siguiente:

Menú Principal

- Opción Temperatura:
 - SubMenú Temperatura
 - Opción Valor Actual
 - Opción Valores Máximo y Mínimo
 - Opción Representación Gráfica
- Opción Humedad:
 - SubMenú Humedad
 - Opción Valor Actual
 - Opción Valores Máximo y Mínimo
 - Opción Representación Gráfica
- Opción Luminosidad:
 - SubMenú Luminosidad
 - Opción Calibrar
 - ◇ SubMenú Calibrar
 - Opción Calibrar como punto más oscuro
 - Opción Calibrar como punto más claro
 - Opción Representación Gráfica
- Opción Presencia:
 - SubMenú Presencia
 - Opción Valor Actual
 - Opción Valor Total

Cada una de estas será una estructura del tipo que le corresponda y guardará en el miembro correspondiente punteros a la(s) estructura(s) subyacente(s).

La función para representar se llama de forma recursiva según se avanza hacia submenús mas profundos, la forma esquemática de funcionamiento de esta función se muestra en la Figura 3.16. Esta función tiene

dos argumentos de entrada: la función para capturar los datos (Figura 3.7, ya que para adquirir los datos analógicos la conversión se realiza en segundo plano hasta que salta la interrupción) y el puntero del Menu Principal. Esta es la única función presente en el *main* del programa (además de las de inicialización de las APPs), que es desde donde se le llama. Al ejecutarse en un ciclo infinito conseguimos que el programa esté en funcionamiento bien hasta que se reinicie, o bien hasta que se desconecte el Microcontrolador de la alimentación.

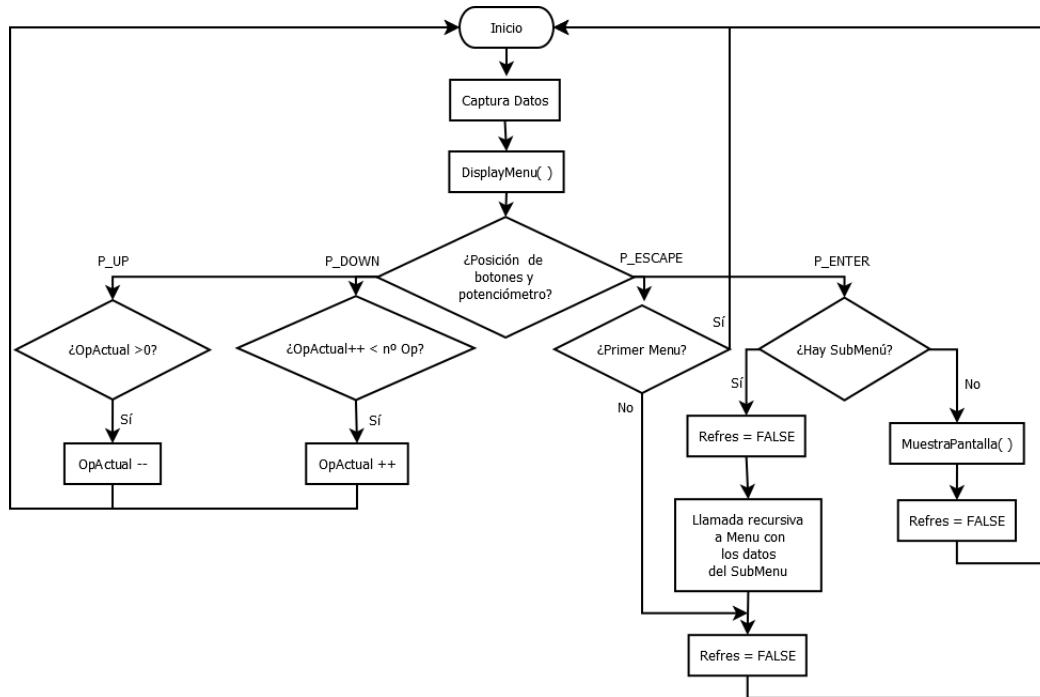


Figura 3.16: Diagrama de flujo de la rutina encargada de la representación por pantalla de los Menús y Opciones [Menu()].

La función DisplayMenu se encarga únicamente de «dibujar» el menú correspondiente a la estructura que se le ha pasado, la cual incluye ciertos parámetros gráficos. Esta función borra la pantalla si así se lo indica el flag Refresh, escribe los títulos de las opciones disponibles en el display con fuente normal y escribe la opción que marca la variable de opción seleccionada en fuente inversa.

Por otra parte, la función MuestraPantalla se encarga de que se ejecute la función asociada a cada opción que esta guardada en la estructura, el diagrama de flujo de dicha función se muestra en la Figura 3.17. Cada vez que se llama a esta función se devuelve el valor de los botones, para dar la posibilidad de retornar al Menú anterior desde ahí.

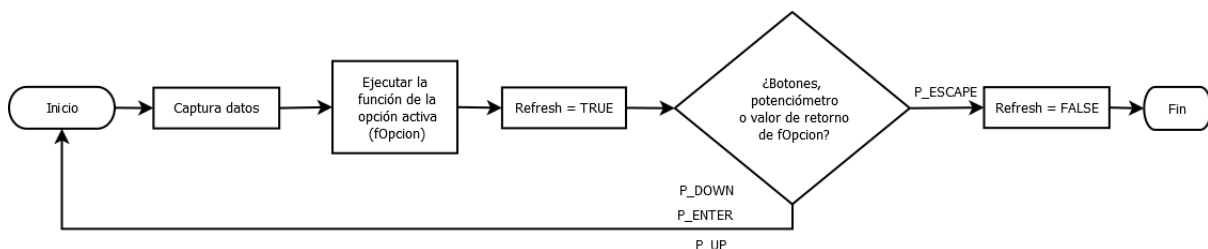


Figura 3.17: Diagrama de flujo de la rutina encargada de que se ejecuten las funciones asociadas a las Opciones [MuestraPantalla()].

A partir del Menú Principal, cuyo aspecto en el display de la plataforma se muestra en la Figura 3.18, se puede seleccionar uno de los cuatro sensores disponibles. A partir de ahí, se mostrará otro Menú que permitirá elegir los datos que se quieren representar por pantalla. A continuación se muestra el proceso de

representado de estos valores y el resultado final sobre la plataforma.

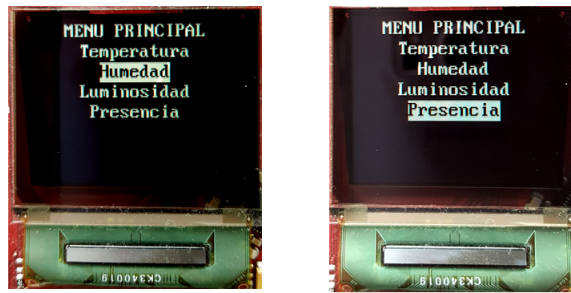


Figura 3.18: Diferentes vistas del Menú Principal

3.6.2. Display de Opciones para los Sensores de Temperatura y Humedad

El aspecto del display cuando se selecciona la Opción de Temperatura y Humedad se muestra en la Figura 3.19, donde se puede elegir entre ver el valor actual del sensor, los valores máximo y mínimo que ha medido el sensor y una representación gráfica de los datos almacenados en la cola circular. Para cada una de las opciones el proceso de representado y resultado final es el que sigue.



Figura 3.19: Vista de los SubMenús de Temperatura (izq.) y Humedad (dcha.)

3.6.2.1 Valor Actual

Para la representación del valor actual (media de un conjunto de valores instantáneos) se sigue el diagrama de la Figura 3.20. Se utilizarán varias funciones de la librería Segger [17] para la representación de este valor, entre ellas se encuentran la de borrado de pantalla `GUI_Clear` que borra la pantalla completa, `GUI_SetFont` para seleccionar un tamaño diferente de fuente al establecido por defecto y `GUI_DispStringHCenterAt` y `GUI_DispDecMin` para escribir una *string* centrada en un punto y un valor decimal respectivamente. El resultado final del display se muestra en la Figura 3.21.

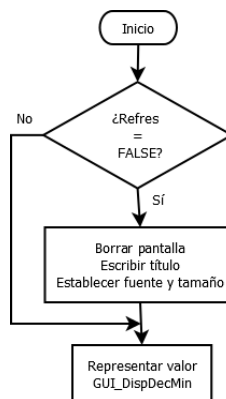


Figura 3.20: Diagrama de flujo de la rutina implementada para la Opción Valor Actual.

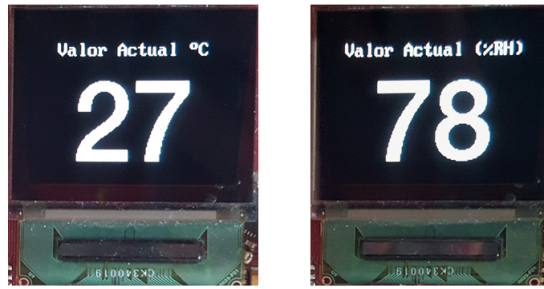


Figura 3.21: Representación de la opción Valor Actual para Temperatura (izq.) y Humedad (dcha.)

3.6.2.2 Valores Mínimo y Máximo

El programa para representar estos dos valores es muy similar al de la Figura 3.20, la única diferencia es que al mostrarse por pantalla tanto el valor máximo como el mínimo, las coordenadas de representación varían y el tamaño de fuente también. El resultado se muestra en la Figura 3.22.



Figura 3.22: Representación de la opción Extremos para Temperatura (izq.) y Humedad (dcha.)

3.6.2.3 Representación gráfica

En este caso hay que hacer uso de la cola en la que se han ido almacenando los valores, pero estos valores no pueden utilizarse directamente en la función que aporta Segger para dibujar gráficas `GUI_DrawGraph`; cuyos argumentos de entrada son: la matriz de valores para la representación en píxeles referenciados a las coordenadas del primer punto, el número de valores a representar y las coordenadas (X,Y) del primer punto.

Como la cola guarda los valores en grados, hay que traducirlos a píxeles. Para ello se ha realizado una función que utiliza una regresión lineal y extrapola los píxeles asignados a dicha temperatura. Se han asignado valores fijos a los extremos de la pantalla, en concreto 24°C y 32°C para la temperatura y $74\% \text{RH}$ y $90\% \text{RH}$ para la humedad, que se encuentran en el rango que se quiere medir. El programa está preparado para que el reescalado sea automático si se cambian estos límites. Para facilitar la visualización se han dibujado también unas líneas indicando los valores máximo y mínimo medido durante el transcurso del programa.

Una vez se ha llenado el display con la representación gráfica se borra la pantalla hasta el nivel del título y siempre que tengamos un nuevo valor en la cola se vuelve a representar. Este proceso se ha seguido con la intención de tener un histórico de valores que se va actualizando con los valores actuales, y de observar las variaciones en el rango seleccionado. El resultado de esta representación gráfica se muestra en la Figura 3.23.

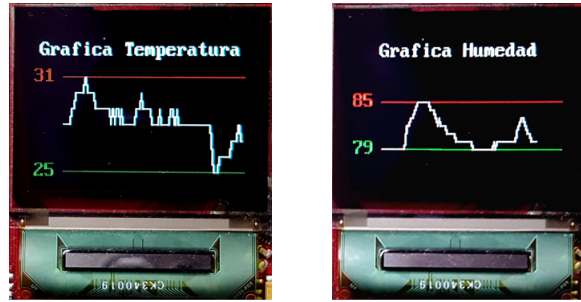


Figura 3.23: Representación de la opción Gráfica para Temperatura (izq.) y Humedad (dcha.)

3.6.3. Display de Opciones para el Sensor de Luminosidad

El aspecto del display cuando se selecciona la Opción de Luminosidad se muestra en la Figura 3.24, donde se puede elegir entre calibrar el sensor a partir de un punto estándar de luminosidad (que queda en manos del usuario decidir si será el punto más oscuro o más claro que se quiere medir); o ver la representación a partir de los datos obtenidos en la calibración. Para cada una de las opciones el proceso de representado y resultado final es el que sigue.

3.6.3.1 Calibrado

Una vez se selecciona esta opción aparece en pantalla el SubMenú asociado a ella, que se muestra en la Figura 3.24. Aquí es posible realizar una calibración ajustada al punto estándar de luminosidad que se quiera medir. El programa guardará los datos de voltaje necesarios dependiendo de la opción elegida, para realizar una representación por pantalla acorde con ella. Las etapas seguidas para realizar esa calibración son las siguientes:

1. Se muestra por pantalla un SubMenú (Figura 3.24) para que el usuario decida si quiere que el nivel actual de luminosidad del ambiente se establezca como el máximo o como el mínimo que se va a detectar, es decir, si el sensor se encuentra en el punto más claro o más oscuro respectivamente.

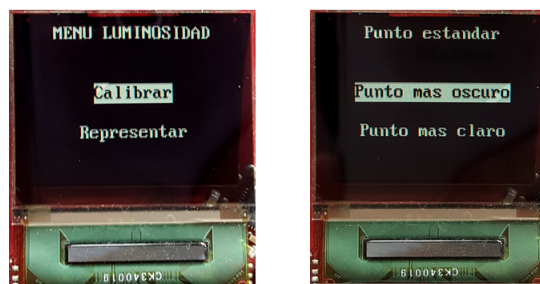


Figura 3.24: Vista del SubMenú de Luminosidad (izq.) y del SubMenú Calibrado (dcha.).

2. Dependiendo de la opción seleccionada, se mostrará por pantalla un mensaje de advertencia en el caso de que la diferencia de voltaje de salida entre el nivel de luminosidad del ambiente y la opción seleccionada sea muy pequeño. Se ha decidido que esta diferencia tiene que ser de al menos 0.515625V, que es el suficiente para distinguir 10 niveles diferentes con una resolución de 6 bits, la mitad de resolución que tiene el ADC. Esto nos aporta un rango suficientemente amplio de valores como para poder mostrar por pantalla una medida estable.

Si se ha seleccionado el punto más oscuro, el sistema mostrará una señal de advertencia como la de la Figura 3.25(izq.) si el voltaje que se está midiendo es superior al que obtenemos para 20kΩ, que es el máximo valor del rango que aporta el fabricante para el sensor en máxima luminosidad.

Si se ha seleccionado el punto más claro, el sistema mostrará una señal de advertencia como la de la Figura 3.25(dcha.) si la diferencia entre el voltaje que está midiendo y el que proporcionaría el sensor para máxima oscuridad no supera los 0.515625V.

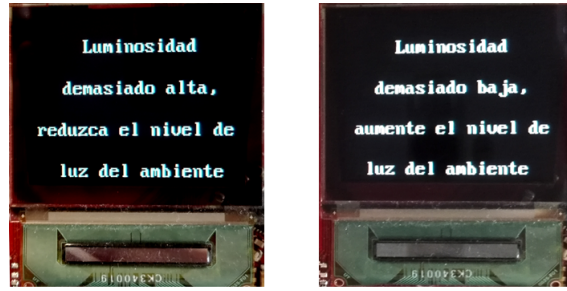


Figura 3.25: Señal de advertencia para valores fuera de rango en el punto más oscuro (izq.) y en el punto más claro (dcha.).

3. Si no se ha producido ninguna señal de advertencia o una vez se haya solucionado, se almacenarán en la estructura de datos del sensor de Luminosidad los valores máximo y mínimo de voltaje con los que trabajará la opción de Representación.

Si se ha seleccionado el punto más oscuro para la calibración, se guardará como voltaje máximo el que se calcula que se obtendrá para máxima luminosidad ($R_{LDR}=5k\Omega$) y como voltaje mínimo el que se ha medido durante la calibración.

Si se ha seleccionado el punto más claro para la calibración, se guardará como voltaje máximo el que se ha medido durante la calibración y como voltaje mínimo el que se calcula que se obtendrá para mínima luminosidad ($R_{LDR}=20M\Omega$).

Una vez finalizado este proceso la función devuelve el valor P_SCAPE, por lo que se retornará directamente al Menú de Opciones (Figura 3.24).

3.6.3.2 Representación

En esta opción se representará en forma de barras el aumento o disminución de la luminosidad en función del punto estándar que se haya escogido previamente. Si se ha elegido el estándar como punto más claro, según se reduzca la luminosidad disminuirá el número de barras desde 10 hasta 0; y si se ha elegido el punto más oscuro las barras aumentarán desde 0 hasta un máximo de 10. Se le han añadido diferentes colores a la representación para facilitar su visualización.

Para poder hacer este tipo de representación se utiliza el valor entero de la Ecuación 10 para calcular el número de barras que hay que dibujar por pantalla. El resultado final se muestra en la Figura 3.26.

$$n^{\circ}barras = 10 - \frac{V_{maximo} - V_{actual}}{V_{maximo} - V_{minimo}} \cdot 10 \quad (10)$$



Figura 3.26: Representación de la escala de Luminosidad para tres valores diferentes. De izq. a dcha. valores ascendentes de Luminosidad.

3.6.4. Display de Opciones para el Sensor de Presencia

El aspecto del display cuando se selecciona la Opción de Presencia se muestra en la Figura 3.27, donde se puede elegir entre ver el valor actual del sensor o el valor acumulado de activaciones. Para cada una de las opciones el proceso de representado y resultado final es el que sigue.

3.6.4.1 Valor Actual

En este caso, dentro de la estructura de datos se ha guardado un array con los colores que se utilizarán para representar los estados de presencia y no presencia, a los cuales se accederá utilizando como índice el valor digital registrado. La función de la librería Segger que se utilizará será `GUI_DrawFillCircle` cuyos argumentos son de entrada son las coordenadas (X,Y) y el radio del círculo. Para establecer el color, antes de dibujar el círculo es necesario utilizar `GUI_SetColor` pasando como argumento el color que se quiere que tenga el círculo. El resultado de esta representación puede verse en la Figura 3.27.

3.6.4.2 Valor Total

Aquí se trata de representar el valor total acumulado de veces en las que el sensor se ha activado y por lo tanto ha detectado una nueva presencia. El proceso de representación por pantalla es el mismo que se utiliza para mostrar el valor actual de la temperatura (Apartado 3.6.2). El resultado final se muestra en la Figura 3.27.



Figura 3.27: Vista del SubMenú Luminosidad (sup. izq.). Representaciones de la opción Valor Total (sup. dcha.) y opción Valor Actual cuando no hay detección (inf. izq.) y cuando sí hay detección (inf. dcha.).

3.7. Esquema general del programa

Finalmente, todas las rutinas descritas a lo largo de los apartados anteriores se engloban en un esquema general de programa (*main*) cuya estructura se muestra en la Figura 3.28.

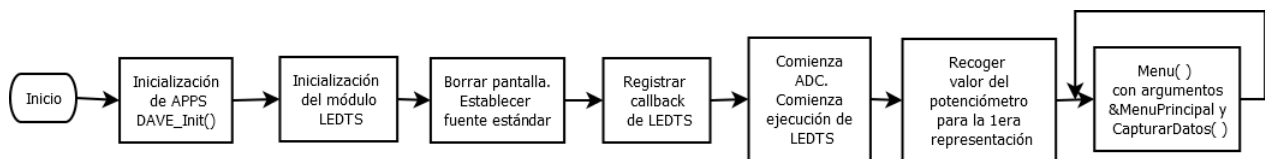


Figura 3.28: Diagrama de flujo del *main*.

En la Figura 3.29 se puede ver la estructura de archivos del Software desarrollado:

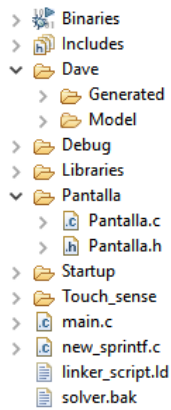


Figura 3.29: Estructura general de archivos del proyecto en el entorno DAVE.

Donde «Pantalla.c» y «Pantalla.h» son los archivos donde se han definido las funciones para almacenar, tratar y representar los datos. En la carpeta Touch_sense se almacena el código que ha sido necesario añadir para solucionar la falta de soporte del módulo LEDTS en el cambio de una versión a otra de DAVE. El resto de archivos son generados por el DAVE a través de las configuraciones que previamente se le indican y archivos para la depuración, también generados automáticamente. El tamaño total del programa es de 450KB.

4. Conclusiones

Tras el desarrollo de este trabajo, que ha consistido en el estudio de la arquitectura y las características principales de un microcontrolador actual y de su entorno de desarrollo de Software para la implementación de un sistema de adquisición de datos; se han podido extraer las siguientes conclusiones:

Durante el transcurso del trabajo se han ido completando varias fases de aprendizaje que se pueden resumir como sigue:

1. Estudio de la arquitectura del Microcontrolador, profundizando en aquellos aspectos que resultan de especial interés para el desarrollo de este proyecto.
2. Conocimiento de las características físicas del Kit de Desarrollo de Hardware en el que va integrado el Microcontrolador y planificación de los recursos necesarios para este proyecto, intentando aprovechar al máximo las características que ofrece.
3. Familiarización con el Entorno de Desarrollo de Software creado específicamente para la programación de microcontroladores de esta familia.
4. Aprendizaje de las estructuras básicas del lenguaje C para el desarrollo de la aplicación con esta plataforma.
5. Diseño e implementación de una red de sensores con la adaptación necesaria a las características del Microcontrolador.

Por otra parte, se ha abordado el diseño de rutinas y funciones de bajo nivel para la adquisición, almacenamiento y tratamiento de los datos, y utilizado librerías externas para su representación por pantalla.

Se ha abordado también la solución de problemas de distinta índole que fueron surgiendo durante el transcurso del proyecto. Desde problemas de diseño e implementación del programa hasta problemas derivados del Hardware del sistema que tuvieron que ser solucionados a través de la depuración de la ejecución del programa a nivel de registros, del análisis de otros ejemplos y de la búsqueda en manuales y material de ayuda.

De la misma forma, se han solucionado los problemas surgidos a raíz del cambio de versión de la plataforma DAVE para la aplicación de control de los botones táctiles. La falta de soporte de este periférico para la nueva versión se ha solucionado adaptando el código implementado en la versión anterior de DAVE.

Se ha podido observar una ausencia de ejemplos por parte de los desarrolladores de Infineon para la ayuda a la comprensión del funcionamiento de ciertas APPs, para ser este un sistema diferente de programación, con mucha generación de código automática. En sí, esta forma de unión de APPs supone una forma muy interesante para la configuración de las múltiples funcionalidades del Microcontrolador, pero puede llegar a resultar algo hermético si se quiere modificar desde el programa esta configuración.

También, observando los resultados obtenidos, se han visto posibles puntos de mejora para este sistema:

- Existe la posibilidad de realizar conversiones analógico-digital con un control mayor sobre el tiempo, para poder dibujar representaciones gráficas con una relación temporal exacta.
- El trabajo realizado con el sensor de luminosidad queda distorsionado por el ruido e interferencias que se introducen. Con una configuración de menor ruido o un sensor de mayor precisión sería posible realizar medidas más significativas de luminosidad.
- Conociendo el funcionamiento de las entradas digitales, es prácticamente inmediata la posibilidad de implementar un programa de control de las salidas digitales. Esto permite utilizarlas para diversas aplicaciones de actuación (enfocadas sobre todo al ámbito industrial por los rangos de voltaje en los que se mueven).

Por último, se puede intuir que este es un dispositivo que tiene un gran potencial para desarrollar aplicaciones complejas debido a sus numerosos periféricos que integra y a las múltiples opciones de configuración que estos ofrecen.

5. Referencias

- [1] List of common microcontrollers. [Online]. Available: https://en.wikipedia.org/wiki/List_of_common_microcontrollers#Hyperstone
- [2] Microcontroladores y sus aplicaciones. [Online]. Available: <https://microcontroladoresv.wordpress.com/about/>
- [3] Reduced instruction set computing. [Online]. Available: https://en.wikipedia.org/wiki/Reduced_instruction_set_computing
- [4] H. U. V. González. (2009, Junio) Microprocesadores y microcontroladores. Universidad Nacional Abierta y a Distancia - UNAD. Bogotá, Colombia. [Online]. Available: http://datateca.unad.edu.co/contenidos/309696/ECBTI_MPyMC_V_02/index.html
- [5] Infineon. Dave (infineon). [Online]. Available: [https://en.wikipedia.org/wiki/DAvE_\(Infineon\)](https://en.wikipedia.org/wiki/DAvE_(Infineon))
- [6] *Help - DAVETM: DAVETM User Manual*, Infineon Technologies AG.
- [7] *KIT XMC4X UNI_EXT01_001*, Infineon Technologies AG. [Online]. Available: http://www.infineon.com/cms/en/product/evaluation-boards/KIT_XMC4X_UNI_EXT01_001/productType.html?productType=db3a304433b8a4100133daf9e260112c
- [8] *XMC4000: Advanced Microcontrollers for Industrial Solutions*, Infineon Technologies AG.
- [9] *AutoManual I/O Card: Board User's Manual*, Infineon Technologies AG, Febrero 2012.
- [10] *Ethernet/CAN/RS485: Board User's Manual*, Infineon Technologies AG, Febrero 2012.
- [11] *Standard Human Machine Interface Card: Board User's Manual*, Infineon Technologies AG, Febrero 2012.
- [12] *CPU Board XMC4500 General Purpose: Board User's Manual*, Infineon Technologies AG, Octubre 2014.
- [13] *XMC4500. Microcontroller Series for Industrial Applications. XMC4000 family.*, Infineon Technologies AG, Julio 2014.
- [14] I. A. M. José María Angulo Usategui, *Microcontroladores PIC. Diseño práctico de aplicaciones.* Mc Graw Hill, 2003.
- [15] *Events 2.0. Technical Guide.*, Libellium Comunicaciones Distribuidas S.L.
- [16] *PIR sensor (555-28027)*, Parallax Inc., hoja de características.
- [17] *emWin: Graphic Library with Graphical User Interface*, SEGGER Microcontroller GmbH and Co. KG, Octubre 2015.
- [18] *808H5V5 Humidity Transmitter*, Sencera Co., hoja de características.
- [19] *Optocoupler, Phototransistor Output, with Base Connection*, Vishay Semiconductors.
- [20] Microcontroladores. Wikipedia. [Online]. Available: <https://es.wikipedia.org/wiki/Microcontrolador>