

UNIVERSIDAD CARLOS III DE MADRID
ESCUELA POLITÉCNICA SUPERIOR



DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

INGENIERÍA TÉCNICA INDUSTRIAL
ESPECIALIDAD ELECTRÓNICA INDUSTRIAL

PROYECTO FIN DE CARRERA

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA
DE CONTROL DE SEGURIDAD EN
BANCOS DE ENSAYO**

AUTOR: RAFAEL JURADO MOLINERO

TUTOR: D. SANTIAGO MARTÍNEZ DE LA CASA

TUTOR EN EMPRESA: D. MIGUEL VARO DEL RÍO

LEGANÉS, MADRID
NOVIEMBRE 2012

A quien lo sabe.

Índice General

Índice General	V
Lista de Figuras	IX
Lista de Tablas	XIII
Agradecimientos	XV
Resumen	XVII
Abstract	XIX
1 Introducción	1
1.1 Ubicación Espacio Temporal	1
1.1.1 Momento Actual. John Deere Ibérica, S.A.	2
1.1.2 Departamento Product Verification and Validation.....	3
1.1.3 Bancos de Ensayo	3
1.1.4 Control en los Bancos de Ensayo	4
1.1.5 Control por CAN	4
1.2 Motivaciones	5
1.3 Objetivos	6
2 Estudio de las Herramientas	9
2.1 El bus CAN	9
2.1.1 Definición	9
2.1.2 Orígenes y causas de su desarrollo	10

2.1.3	Principios de Funcionamiento Básico.....	10
2.1.4	Elementos que Conforman el CAN-bus	11
2.1.4.1	Cables.....	11
2.1.4.2	Terminadores.....	13
2.1.4.3	Controlador	14
2.1.4.4	Transceiver.....	14
2.1.5	Capas definidas por CAN	15
2.1.5.1	Capa física.....	16
2.1.5.2	Capa de enlace.....	18
2.1.6	Tramas en CAN	19
2.1.6.1	Trama simple o estándar	20
2.1.6.2	Trama extendida.....	22
2.2	Herramientas Hardware	23
2.2.1	Microcontrolador	23
2.2.2	Placa de Desarrollo CEREBOT MX7cK.....	28
2.2.3	Interfaz USB-CAN	37
2.2.4	Relés de Estado Sólido	39
2.2.5	Fuente de Alimentación.....	42
2.3	Herramientas Software.....	44
2.3.1	Entorno de Desarrollo MPLAB IDE.....	44
2.3.2	Programador/Depurador	46
2.3.3	Entorno de Pruebas del Módulo CAN	46
2.3.4	Módulo de seguridad PILZ.....	49
3	Desarrollo del Sistema	51
3.1	Condiciones de partida.....	51
3.1.1	Especificaciones	57
3.2	Programación del Sistema.....	60
3.2.1	Configuración del sistema	61
3.2.1.1	Configuración CAN	62
3.2.1.1.1	Configuración de la Velocidad	62
3.2.1.1.2	Configuración de las Estructuras FIFO.....	64
3.2.1.1.3	Configuración de Filtros y Máscaras	66
3.2.1.1.4	Estructuras de Datos	69
3.2.1.2	Configuración del Reloj	72
3.2.1.3	Configuración de <i>timer1</i>	72
3.2.2	Lectura de los FIFO	73

3.2.3 Comprobación de Latidos.....	78
3.2.4 Gestión de la Entrada.....	80
3.2.5 Watchdog.....	83
4 Conclusiones	87
4.1 Resultados.....	88
4.2 Trabajos Futuros.....	90
4.2.1 Display.....	90
4.2.2 USB.....	91
4.2.3 Ethernet.....	91
4.2.4 CAN.....	92
4.2.5 Otros usos.....	92
5 Presupuesto	93
Bibliografía	97
Hojas de Características	101

Lista de Figuras

2.1: Cableado.....	11
2.2: Tensión diferencial.....	12
2.3: Circuito equivalente.....	13
2.4: Terminadores.....	14
2.5: Controlador.....	14
2.6: Transceiver.....	15
2.7: Niveles OSI y CAN.....	16
2.8: Segmentos de tiempo.....	17
2.9: Filtrado de tramas den CAN.....	18
2.10: Arbitraje en CAN.....	19
2.11: Trama CAN.....	20
2.12: Trama estándar.....	20
2.13: Trama extendida.....	22
2.14: Diagrama de bloques de un PIC32MX7xx de Microchip.....	24
2.15: Nomenclatura de la familia PIC32.....	27
2.16: Placa CEREBOT MX7cK.....	28
2.17: Diagrama de bloques de la placa CEREBOT MX7cK.....	29
2.18: Resistencias de 200Ω para protección contra cortocircuitos.....	32

2.19: Diodos de protección ESD.....	32
2.20: Transceptores, terminadores y conectores de CAN.....	33
2.21: Descripción de los pines del transceptor MCP2551.....	35
2.22: Esquema del transceptor MCP2551.....	35
2.23: Esquema de botones y LED del CEREBOT MX7cK.....	36
2.24: Interfaz NI USB-8473 de National Instruments.....	37
2.25: Conector de la interfaz NI USB-8473.....	38
2.26: Cable de conexión entre NI USB-8473 y placa CEREBOT.....	38
2.27: Serie DEK de relés de Phoenix Contact.....	39
2.28: Esquema del relé DEK-OE-5VDC/24VDC/100kHz.....	40
2.29: Fuente de alimentación para carril DIN, OMRON S8VS03005.....	42
2.30: Entorno gráfico MPLAB v8.83.....	44
2.31: Selección del programador/depurador bajo licencia Microchip.....	45
2.32: Panel de trabajo para el desarrollo de la interfaz en LabVIEW.....	47
2.33: Código de la interfaz para el envío de tramas CAN.....	48
2.34: Código del módulo CANWrite.....	48
2.35: Módulo de seguridad PILZ PNOZ Multi M 1 P.....	49
3.1: Armarios de potencia y de variadores.....	51
3.2: Centralita hidráulica.....	52
3.3: Intercambiador de calor con sensores de temperatura.....	52
3.4: Actuador. Cilindro hidráulico.....	53
3.5: Actuador. Cilindro neumático.....	53
3.6: Sensores de temperatura, resistivo y termopar.....	54
3.7: Sensor de presión.....	54
3.8: Célula de carga.....	55
3.9: Torsiómetro.....	55

3.10: Encoder asociado al eje de un motor.	56
3.11: Arquitectura del sistema de ensayos.	56
3.12: Esquema de un banco de ensayo genérico.	58
3.13: Diagrama de bloques del sistema.	59
3.14: Organigrama general.	61
3.15: Estructuras FIFO en el tratamiento de los datos de CAN.	64
3.16: Filtrado y enmascarado de un mensaje de CAN.	67
3.17: Formato de los datos almacenados en RAM.	69
3.18: Formato CMSGID.	69
3.19: Formato CMSGDATA0.	70
3.20: Formato CMSGDATA1.	70
3.21: Relé KM11A y contactor KM11 del motor M1.	74
3.22: Extracción del <i>FctCode</i>	76
3.23: Interruptor de petición de entrada (“ <i>ABRIR PUERTA</i> ”) en armario de control.	80
3.24: Organigrama de gestión de entrada al banco de ensayo.	82
3.25: Post-escalador del watchdog.	84

Lista de Tablas

2.1: Características principales del PIC32MX7xx.	25
2.2: Características del relé DEK-OE-24VDC/5VDC/100kHz.....	40
2.3: Esquema del relé DEK-OE-24VDC/5VDC/100kHz.	41
2.4: Características del relé DEK-OE-24VDC/5VDC/100kHz.....	41
2.5: Características de la fuente de alimentación OMRON S8VS03005.	43
3.1: Señales de entrada y salida del Sistema de Control.	59
3.2: Time quanta para velocidades de 500 kbps.....	62
3.3: Requerimientos de memoria de los mensajes de CAN.	65
3.4: Bits de configuración del registro CiFLTCOnn.	67
3.5: Entradas y salidas empleadas en la gestión del acceso a la sala.....	81
3.6: Registro WDTCON.....	83
3.7: Registro de configuración DEVCFG1.	84

Agradecimientos

Agradezco a mi tutor, Dr. Santiago Martínez de la Casa, su apoyo, acertado consejo y verdadera voluntad de ayuda.

A mi tutor en John Deere Ibérica, D. Miguel Varo del Río, le doy las gracias por su confianza y por su sentido común. Gracias por compartir conocimientos.

Víctor González y Raúl Sánchez han sido de inestimable ayuda en todo momento. A ambos les agradezco sinceramente su desinteresada colaboración y la dedicación de su tiempo.

Gracias a mi familia por el aporte constante de motivación durante los años de carrera. Gracias de nuevo por compartir los momentos finales, los mejores.

Resumen

El departamento *Product Verification and Validation* de John Deere Ibérica dispone de bancos de ensayo para probar los productos que se diseñan y construyen en las instalaciones de la factoría. El control de los bancos de ensayo está migrando a la arquitectura de bus CAN. Surge así la necesidad de un producto que supervise el funcionamiento, un bus ubicado en condiciones especialmente adversas de humedad, altas temperaturas o vibraciones.

Por otro lado, John Deere Ibérica impulsa la seguridad como pieza clave en el entorno industrial. Es por ello que se requiere un sistema que proporcione el acceso seguro a las salas de ensayo, basándose en las mediciones que los transductores vuelcan en los buses CAN.

El presente Proyecto Fin de Carrera, mediante el análisis de las tramas CAN aporta la supervisión del bus y controla el acceso de las personas a las salas de ensayo bajo condiciones de seguridad.

Para ello se ha realizado un sistema basado en microprocesador montado sobre una placa comercial de desarrollo. El conjunto se conecta a los módulos de seguridad para las maniobras de entrada a la sala, y al bus CAN sin interferir en el trasiego de datos de éste.

Abstract

John Deere Ibérica Product Verification and Validation Department has test benches available to verify the products that are designed and manufactured in the factory facilities. The security in controlling the test benches is currently being transferred to a CAN bus architecture. In this way, it is necessary a product to quantify changes in performance; a bus that may operate in particularly adverse conditions, such as dampness, high temperatures or vibrations, independently of the work site on which it is located.

By the other hand, John Deere Ibérica promotes security as a highly important issue in industrial environments. Therefore, a system that provides safety access to the testing rooms is required, taking into account sensor measurements transmitted onto the CAN buses. The main purpose of this work is both to provide CAN bus supervision or access control to the testing rooms under safety conditions, by analyzing the CAN bus data frames.

Hence, a system based on a microprocessor placed on a commercial development board has been developed. The whole system is not only connected to the safety modules, which manage the entry access to the testing rooms, but also to the CAN bus without interfering the data transmission.

Introducción

1.1 Ubicación Espacio Temporal

Deere & Company es una compañía dedicada principalmente a la construcción de maquinaria. Tiene su origen en Moline, en el estado de Illinois, donde en 1837 John Deere construye el primer arado.

En España, Lanz, empresa alemana destinada a la fabricación de tractores con sede en Mannheim, se establece en Madrid, construyendo su fábrica de Getafe en 1951. El primer tractor saldría de las instalaciones en 1953. Deere & Corp. comienza su expansión mundial en 1956, llegando a controlar la mayoría de las acciones de Lanz en 1962. A finales de 1965, Deere & Corp reestructura el sistema de fabricación, gestión de personal, administración, etc. y la dualidad Lanz – Deere desaparece. A partir de esos momentos, el nombre definitivo de la empresa pasa a ser John Deere. Es así pues, como en Getafe aparece John Deere Ibérica S.A.

Desde los años sesenta hasta nuestros días la evolución ha sido considerable. En cincuenta años han variado los mercados y las necesidades de los mismos. También, por supuesto la tecnología ha experimentado un gran avance; la mecánica se ha visto beneficiada por la introducción de maquinaria regida por control numérico, nuevos

materiales en herramientas, sistemas de medición altamente concisos, permitiendo de este modo precisiones cada vez mayores y más altos niveles de calidad. La electrónica ha colaborado de forma importante tanto en la fabricación como en los controles de calidad. En 1995 salía de la fábrica John Deere en Getafe el último tractor. Se trató de un momento crucial en el que la fábrica hubo de reestructurarse para pasar a fabricar componentes para los productos finales que más tarde habrían de ensamblarse en otras fábricas de la compañía, situadas en Estados Unidos y Alemania.

1.1.1 Momento Actual. John Deere Ibérica, S.A.

John Deere Ibérica en la actualidad es una empresa dedicada al diseño, fabricación, pruebas y venta de componentes para otras fábricas de la multinacional Deere & Corp. Sus principales productos son ejes, engranajes y transmisiones. Las transmisiones engloban desde las pequeñas cajas de simples engranajes a 90º con pocos vatios de potencia a transmitir, hasta cajas de cambios automáticas transmisoras de altas potencias requeridas por vehículos de gran tonelaje.

Los productos finales de la División Agrícola de Deere&Corp abarcan desde variadas series de tractores, cosechadoras destinadas a la recolección de diferentes productos agrícolas, fumigadoras, procesadoras de arbolado (tala, desbroce, corte y almacenaje), hasta equipos para jardines, así como maquinaria para el ocio. La División de Construcción se encarga de productos tales como excavadoras, apisonadoras, taladradoras, etc., John Deere Ibérica es suministradora de ambas divisiones.

Se ha alcanzado un alto grado de automatización en las líneas de producción y se han introducido controles intermedios de calidad y *Poka Yokes* que tratan de imposibilitar la entrega de productos defectuosos.

1.1.2 Departamento Product Verification and Validation

En John Deere Ibérica, los nuevos diseños son transformados en pequeñas series de prototipos que son ensamblados en las propias instalaciones. El departamento PV&V (*Product Verification and Validation*) se encarga de realizar el montaje de los prototipos, de diseñar las pruebas que deben superar éstos y de llevarlas a efecto. Para realizar dicha misión se cuenta con un equipo de ingenieros de pruebas en cargados del diseño, control e interpretación de los resultados de los ensayos. El equipo de mecánicos y electricistas materializan los montajes de los prototipos y su instalación en los bancos de prueba. PV&V se encarga también del diseño de los bancos de prueba que han de reunir las condiciones necesarias para llevar a los componentes que se ensayan a los extremos deseados.

El departamento PV&V cuenta actualmente con diecisiete bancos de prueba, organizados por las potencias de las cajas a ensayar. Así pues, John Deere Ibérica cuenta con dos bancos para potencias ligeras (*Low Gear Boxes Test Benchs*), cuatro para potencias medias (*Medium Gear Boxes Test Benchs*), cuatro bancos *Four Square* y siete bancos de gran potencia (*Heavy Gear Boxes Test Benchs*).

1.1.3 Bancos de Ensayo

A partir de 2005, el departamento PV&V sufre una transformación. De cinco bancos de ensayo se pasa a diecinueve. Los primitivos eran artilugios dotados de poca flexibilidad en lo que a probetas se refiere. Estos primeros bancos de ensayo estaban dotados de potencias medias y bajas, y sistemas de control basados en PLCs.

El concepto de los bancos de ensayo ha cambiado. Los motores hidráulicos se han sustituido por motores eléctricos gracias al desarrollo alcanzado por la electrónica de potencia y las máquinas eléctricas. Las bancadas han sido diseñadas para cumplir con propósitos generales (bancadas universales) de modo que pueden variarse con relativamente escasos costes el tipo y las condiciones de ensayo.

Naturalmente, la modernización se ha visto reflejada en el control de los bancos. Del PLC se ha pasado al PC merced a las considerables ventajas económicas. Las posibilidades de control son mucho mayores debido a la gran potencia que ofrecen los equipos informáticos a precios razonables.

1.1.4 Control en los Bancos de Ensayo

La mayor parte de los bancos de ensayo disponen de control por PC, ya sea comercial o industrial.

El PC contiene la aplicación final de usuario y se conecta por Ethernet a una máquina industrial Linux que se encarga del envío y recogida de datos de todos los sensores dispuestos en el ensayo. Cada probeta de ensayo está instrumentada con sensores de temperatura, velocidad, presión, vibración, posición, fuerza, etc. Por otra parte, es necesario establecer el control sobre los elementos activos del banco, tales como variadores, bombas neumáticas, cilindros hidráulicos, frenos, etc.

1.1.5 Control por CAN

Debido al gran número de sensores y actuadores presentes en un banco de ensayo, se hace necesaria la existencia de numerosas tarjetas de entrada/salida así como de un gran número de cables. PV&V ha optado por realizar estas operaciones empleando el bus CAN.

Como se mencionó previamente, los numerosos sensores y actuadores requieren de gran infraestructura de electrónica y cableado. La aparición del bus CAN se debe, entre otras, a esta razón. Cada uno de los sensores y actuadores empleados en el ensayo se conecta a un bornero, o se conecta directamente al bus CAN, comportándose como un nodo CAN.

1.2 Motivaciones

El departamento PV&V de John Deere Ibérica pretende conseguir un control exhaustivo de sus bancos de ensayo. Es importante resaltar que los ensayos suelen tener larga duración y un funcionamiento autónomo. Las pruebas se realizan habitualmente sin la presencia física de los ingenieros de PV&V. Por esta causa, la supervisión es cedida a los computadores.

Se han dado casos de malos funcionamientos en los sistemas de supervisión. Las pruebas han fallado por problemas en la red de CAN y se han producido pérdidas de materiales y tiempos de ensayo y trabajo.

Por otra parte, antes de realizar un ensayo automático es frecuente la entrada a la sala de pruebas para realizar la comprobación de estado de las probetas, actuadores, etc. Es conveniente realizar inspecciones visuales para detectar fugas, roces, u otros inconvenientes. También es necesario examinar la presencia de ruidos extraños en rodamientos, engranajes, cilindros, etc. Estas comprobaciones requieren el funcionamiento de los actuadores, lo que puede provocar situaciones de peligro para las personas.

La ausencia en el mercado de productos versátiles que realicen la supervisión del bus CAN y estén capacitados para controlar las condiciones de seguridad que se demanden, provoca la motivación del diseño de un Sistema de Control capaz de asumir estas tareas.

La presencia de microcontroladores dotados de altas prestaciones y precios razonables, incita al diseño de un Sistema de Control con el que abordar los trabajos de vigilancia y seguridad. El diseño e implementación del sistema que representa el propósito de este Proyecto Fin de Carrera ha de resultar fructífero en tanto que se profundizará en el conocimiento de los buses de CAN y las condiciones de seguridad.

Son, por tanto, motivaciones para la realización de este PFC, la consecución de entornos de pruebas más fiables, que eviten daños en las probetas e instalaciones, así como la implementación de un sistema capaz de supervisar las condiciones bajo las

cuales los operadores de los ensayos puedan acceder a la sala de pruebas en condiciones seguras.

1.3 Objetivos

Se establecen como objetivos una mejora en la supervisión del bus CAN de los bancos de ensayo y la posibilidad del acceso a la sala de ensayo bajo condiciones de seguridad para las personas.

Para la consecución de los objetivos, el Sistema de Control que se pretende diseñar estará basado en un microcontrolador dotado de módulo de CAN y puertos de entrada/salida disponibles para las maniobras de control y seguridad.

Se procura llevar a cabo la implementación del Sistema de Control en una tarjeta multipropósito comercial que ha de actuar como interfaz entre el microcontrolador y el bus CAN y los elementos de maniobra.

Así pues, el control de la seguridad del banco de ensayo estará a cargo del Sistema de Control objetivo de este Proyecto Fin de Carrera.

Por tanto, los objetivos del Proyecto son los siguientes:

- Desarrollo de un firmware para microcontrolador capaz de supervisar la operatividad del bus CAN y de tomar el control del acceso seguro la sala de ensayo.
- Conexión de una tarjeta multipropósito al módulo de seguridad del armario de control con las siguientes funciones:
 - Suministro de las señales necesarias para las maniobras de apertura y cierre de la puerta de acceso a la sala.
 - Accionamiento de la parada de emergencia del banco de ensayo cuando el bus CAN esté en condiciones de error.

- Lectura de señales proporcionadas por el módulo de seguridad del armario de control.
- Integración de la placa multipropósito en el armario de control del banco de ensayo y anclaje al carril DIN.
- Suministro de la alimentación a la placa multipropósito mediante una fuente de alimentación dotada de sistema de anclaje a carril DIN.
- Conexión de relés de estado sólido entre la placa multipropósito y el módulo de seguridad del armario de control, tanto para entradas como para salidas.
- Conexión de la placa multipropósito a la fuente de señal indicadora de inicio de la actividad del bus CAN del banco de ensayo.

Capítulo 2

Estudio de las Herramientas

2.1 El bus CAN

2.1.1 Definición

El bus CAN (acrónimo de Control Area Network) es un protocolo serie y asíncrono que provee una comunicación segura en entornos de ruido eléctrico.

Se trata de un protocolo basado en el modelo multimaestro, con sistema de transmisión de tipo Broadcast y capacidad de funcionamiento en tiempo real.

La norma ISO 11898, especifica las capas física y de enlace (DLL, Data Link Layer) del bus CAN, que se define como un protocolo de comunicaciones serie que soporta control distribuido en tiempo real y multiplexación para uso en vehículos.

Al estar definidas sólo las capas física y de enlace, se habla de un bus de altas prestaciones, pero de escasa funcionalidad.

2.1.2 Orígenes y causas de su desarrollo

En 1983 comienza un proyecto interno de Robert Bosch GmbH para el desarrollo de una red para vehículos. En febrero de 1986, Bosch presenta el sistema de bus serie Controller Area Network (CAN) en un congreso de la Sociedad de Ingenieros de Automoción (SAE). Constituyó el nacimiento de uno de los protocolos de red más exitosos de la industria.

Hoy en día, la práctica totalidad de los automóviles de turismo fabricados en Europa está equipada con al menos una red CAN. También se utiliza en otros tipos de vehículos, desde trenes hasta barcos, así como en la industria. CAN es tal vez el principal sistema de bus serie en todo el mundo.

Con anterioridad, los fabricantes de automóviles realizaban las conexiones de dispositivos electrónicos en los vehículos utilizando sistemas de cableado punto a punto. A medida que los automóviles iban implementando mayor número de dispositivos electrónicos, los arneses de estos subían en su peso y coste general. Al replazar el cableado por redes de comunicación en los vehículos, se logró reducir el coste asociado al cable, su complejidad y su peso. CAN, un sistema de bus serie de alta integridad destinado a la comunicación de dispositivos inteligentes, emergió como una red estándar para vehículos. La industria automotriz adoptó rápidamente CAN, y en 1993 se convirtió en el estándar internacional ISO 11898. Desde 1994, se han estandarizado varios protocolos de alto nivel a partir de CAN, como CANopen (mantenido por CAN in Automation), DeviceNet (orientado a objetos, mantenido por Open DeviceNet Vendor Association), CANaerospace (aviónica y tecnología aeroespacial), y su uso se ha extendido a otras industrias. [1][2][3][4][7][29]

2.1.3 Principios de Funcionamiento Básico

La información se transmite en forma de mensajes estructurados. Una parte de la estructura de datos, la trama, contiene un identificador, el cual hace referencia al tipo de mensaje que transporta. Como CAN-bus es un sistema broadcast, todos los elementos conectados al bus recibirán el mensaje. Debido a esta razón, se dota al

protocolo de un sistema de filtrado para que los elementos conectados sean capaces de discriminar qué mensajes son de su incumbencia.

Todos los elementos conectados al bus tienen la posibilidad de emitir y de recibir. Puede darse el caso de que dos nodos pretendan transmitir a la vez. Para solucionar las colisiones en el bus, CAN-bus implementa el método CSMA/AMP (Carrier Sense Multiple Access / Arbitration by Message Priority). CSMA/AMP es un algoritmo de arbitraje que resuelve el conflicto evitando la pérdida de información, que decide qué nodo accede al bus en función de la prioridad que porte la trama. Este procedimiento evita la pérdida de datos porque se realiza con anterioridad a la transmisión. Por tanto, la trama que se transmite es aquella con mayor prioridad; el resto de los nodos deben esperar a encontrar el bus libre para transmitir su mensaje.

2.1.4 Elementos que Conforman el CAN-bus

2.1.4.1 Cables

La normativa ISO 11898-2 y 3 especifica cable de par trenzado UTP o STP, si bien es posible realizar transmisiones sin uso de cableado (*wireless*), mediante fibra óptica o incluso empleando la red eléctrica (*powerline*) como medio de transmisión. La figura siguiente muestra un ejemplo de cableado utilizando par trenzado.

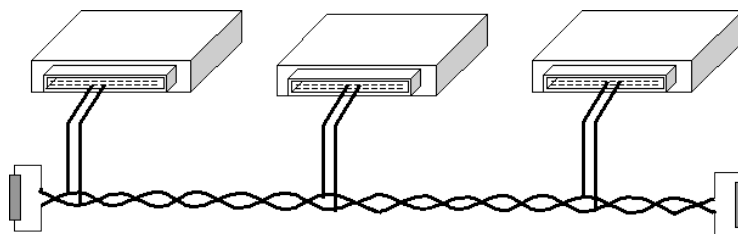


Figura 2.1: Cableado.

Ciñéndose al uso del par trenzado, que permite grandes velocidades de transmisión y valores aceptables de ruido, se dirá que la información se transmite de forma diferencial, tal como muestra la siguiente figura.

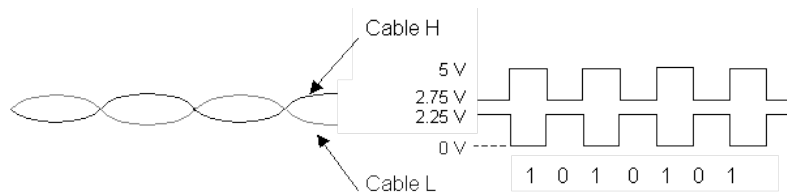


Figura 2.2: Tensión diferencial.

En la figura anterior, en el hilo "H" (high) la tensión puede ser de 5V o de 2.75V, y en el hilo "L" la tensión toma valores de 0V o de 2.75V. La representación de la unidad mínima de información, un bit, se realiza como diferencia de los valores espejados de los hilos "H" y "L".

Este concepto permite la operatividad del bus en los casos en que uno de los hilos se corte o haya derivado a masa.

En efecto, en el caso de que cualquiera de los hilos haya sido cortado, el sistema CAN-bus puede tomar la referencia del otro hilo, del que obtendrá los bits como diferencia de valores de tensión de 2.25V. Similar es el caso de derivación.

La norma ISO 11898 refiere tensiones de -2V en "L" y de +7 en "H".

El dimensionado del cableado debe ser cuidadoso, pues las características eléctricas dependen de la longitud, sección, conductividad, apantallamiento, etc. Una caída de tensión excesiva en el cable puede ocasionar la imposibilidad de comunicación debido a que los nodos no diferencien entre los niveles de tensión adecuados.

Con un diseño adecuado del cableado, se pueden alcanzar buses de hasta 1000 metros.

La siguiente figura muestra un circuito equivalente para el dimensionado del bus e ilustra lo anteriormente mencionado.

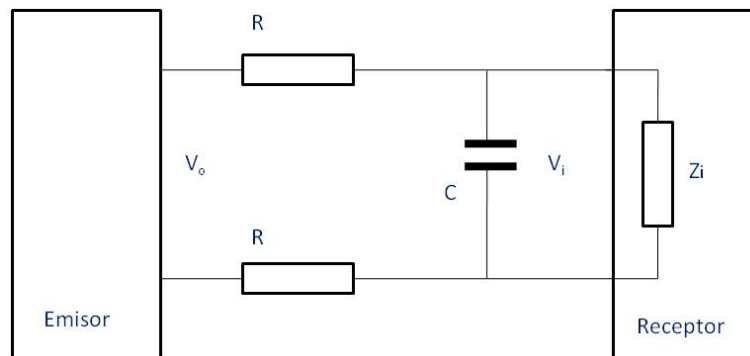


Figura 2.3: Circuito equivalente.

Los parámetros que aparecen en la Figura 1.3, y que se enumeran a continuación, son los responsables de una tasa de transferencia adecuada.

R: Resistencia del cable , siendo $R = \rho \cdot L / S$

C: Capacidad parásita.

V_i : Tensión mínima de entrada requerida por el receptor.

V_o : Tensión de salida del emisor.

Z_i : Impedancia de entrada del nodo receptor

Para buses de pocos metros, se suele despreciar los valores de R y C.

Debido al empleo generalizado del par trenzado, las interferencias electromagnéticas son reducidas, de modo que no es necesario asumir costes añadidos para minimizar los efectos perjudiciales de las EMI.

2.1.4.2 Terminadores

Como se ha mencionado anteriormente, la topología básica es en bus, y para evitar reflexiones de la señal se emplean terminadores en los extremos, según ISO 11898-2 y 3. Estos terminadores son resistencias con un valor típico de 120Ω .

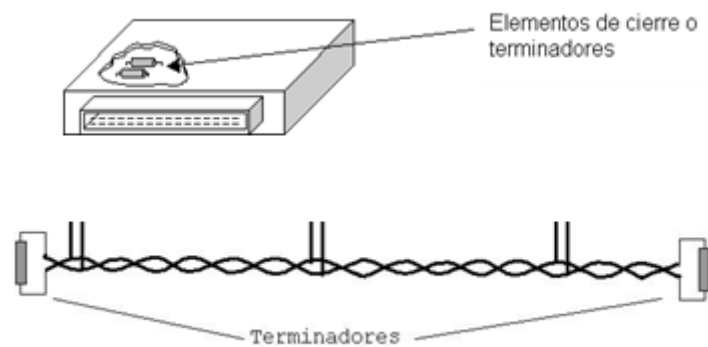


Figura 2.4: Terminadores.

2.1.4.3 Controlador

Es el elemento encargado de la comunicación entre el microprocesador del nodo y el *transceiver*. Acondiciona las señales entre ambos componentes.

Cada uno de los nodos ha de disponer de un controlador. Este elemento trabaja con niveles de tensión muy bajos y es el que determina la velocidad de transmisión de los mensajes, que será más o menos elevada según el compromiso del sistema. El controlador también interviene en la sincronización de los nodos conectados al bus para lograr una correcta emisión y recepción de los mensajes.

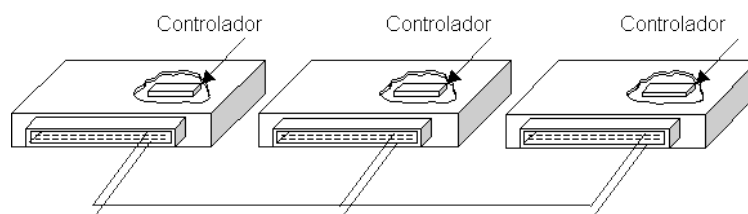


Figura 2.5: Controlador.

2.1.4.4 Transceiver

El *transceiver* o transmisor-receptor (*transceiver* = *transmit and receive*) es un circuito integrado cuya misión es la transmisión y recepción de los datos. También está encargado del acondicionamiento de las señales para su uso por parte del controlador. Para la transmisión, el *transceiver* amplifica la señal recibida del controlador de

manera que los niveles de tensión sean adecuados para el bus. Para la recepción, los niveles de tensión son atenuados hasta los niveles adecuados del controlador.

Como circuito integrado de carácter comercial, suele estar dotado de protecciones frente a cortocircuitos y sobretensiones. Una función adicional del transceptor puede ser también el aislamiento galvánico de un nodo CAN y la línea de bus.

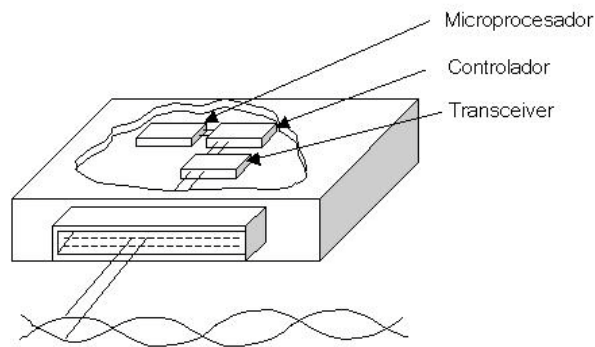


Figura 2.6: Transceiver.

2.1.5 Capas definidas por CAN

En CAN sólo se definen dos capas: física y de enlace.

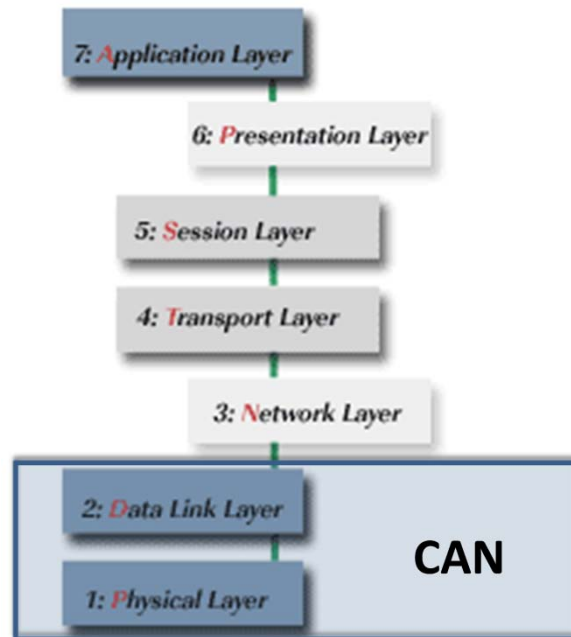


Figura 2.7: Niveles OSI y CAN.

2.1.5.1 Capa física

La norma ISO 11898-2 estandariza la capa física de CAN.

Los controladores de CAN disponen de su propio reloj. Como es lógico, las frecuencias de los distintos controladores conectados al bus pueden diferenciarse ligeramente. Para paliar este fenómeno, los controladores se resincronizan con la corriente de bits. La velocidad de transmisión se ajusta para que sea la misma en todos los dispositivos aunque los periodos producidos por los osciladores no sean exactamente iguales. Este es el concepto del *bit timing*.

El tiempo nominal de un bit se divide en cuatro segmentos de tiempo [5]:

- Segmento de sincronización: Dentro de este segmento se espera un flanco. Según el protocolo CAN, este segmento ocupa un quantum de tiempo.

- Segmento de propagación: Compensa el desfase provocado por los retardos en la propagación del bus o de los transceivers conectados a él.
- Segmentos de fase 1 y 2: Son segmentos de tiempo situados antes y después del punto de muestreo, que pueden ajustar su duración para lograr la sincronización.

El punto de muestreo se coloca típicamente entre el 70% y el 80% del tiempo nominal de bit.

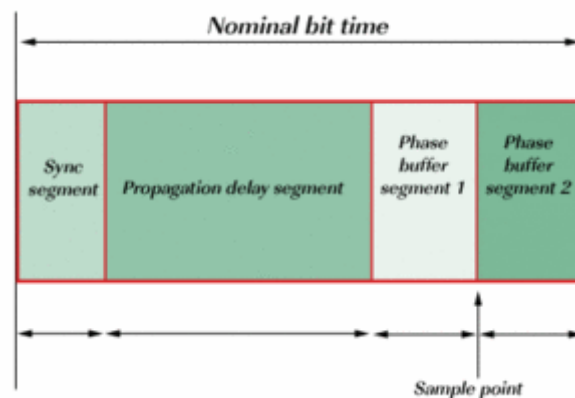


Figura 2.8: Segmentos de tiempo.

El punto de muestreo es aquél en el que se lee el valor del dato (0 ó 1).

Si bien la topología característica es el bus, es posible diseñar otras mediante el uso de:

- Repetidores (*Repeaters*). Añaden un cierto retardo en la comunicación. Se mantiene un único bus lógico.
- Puentes (*Bridges*). Conectan dos redes lógicas separadas en el nivel de enlace. Sólo retransmiten los mensajes que no son locales.
- Pasarelas (*Gateways*). Pueden conectar redes CAN con diferentes protocolos de alto nivel, cuando existen. Se trata de ampliaciones a CAN.

La velocidad máxima de transmisión está definida en 1 Mbps. El retardo de propagación nominal es de 5 ns/m. El número de nodos se limita por la carga eléctrica.

Aparte de ISO 11898, otras normas son aplicables, tales como ISO 11899 (CAN tolerante a fallos), SAE J2441 (CAN de bajos requerimientos), ISO 11992 (CAN punto a punto).

Para aplicaciones industriales se emplean recomendaciones CiA 102, y para automoción, SAE J2284. Hasta la fecha, las transmisiones por fibra óptica no están estandarizadas.

2.1.5.2 Capa de enlace

Según el modelo OSI (*Open System Interconnection*), esta capa se ocupa del direccionamiento físico, de la topología de la red, del acceso al medio, de la detección de errores, de la distribución ordenada de tramas y del control de flujo.

La capa de enlace se sitúa entre las capas física y de red para la creación de sus protocolos básicos; regula la forma de la conexión entre elementos de la red, y determina el paso de tramas.

Los receptores, mediante la aplicación de filtros, decidirán si la trama leída del bus será tratada por ellos. Si la trama pasa el filtro, el nodo tratará la información; en caso contrario, la desechará. La siguiente figura ilustra el mecanismo.

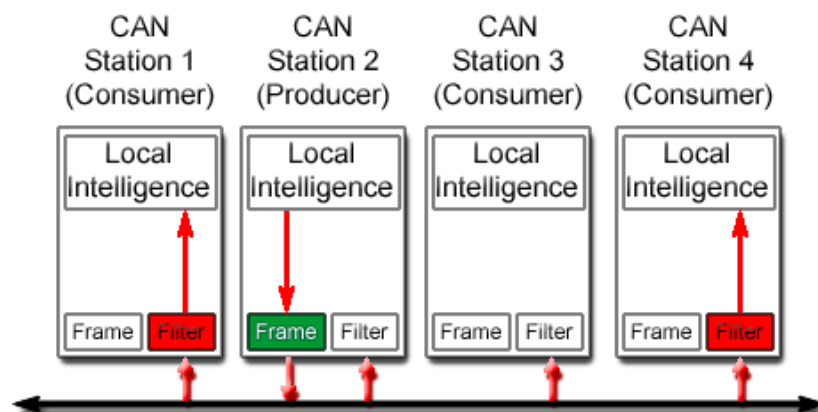


Figura 2.9: Filtrado de tramas den CAN.

En CAN, la capa de enlace es la encargada de llevar a cabo el arbitraje CSMA/AMP (*Carrier Sense Multiple Access / Arbitration by Message Priority*) para el tratamiento de las colisiones.

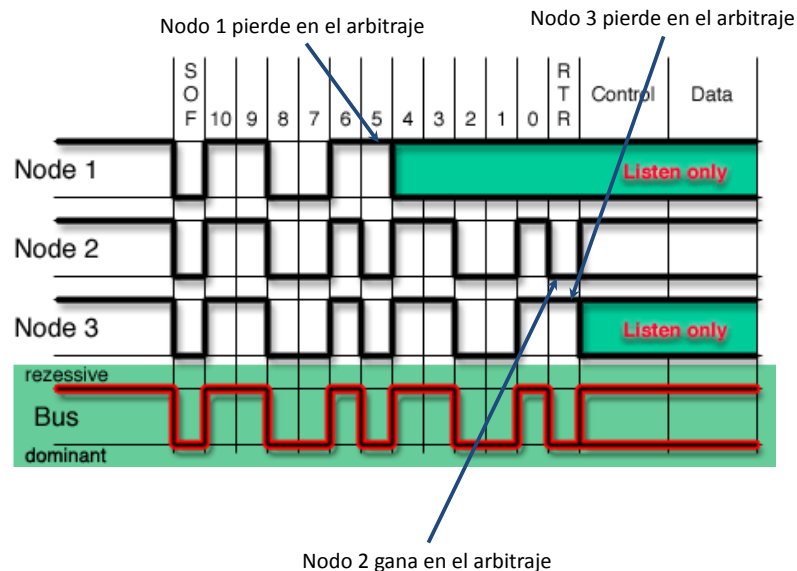


Figura 2.10: Arbitraje en CAN.

En CAN, el bit dominante es el "0", y el recesivo, el "1".

En el ejemplo que muestra la Figura 1.10, la trama emitida por el nodo 2 adquiere el derecho de transmisión por tener una prioridad más alta, marcada por los bits del campo ID.

2.1.6 Tramas en CAN

CAN soporta dos formatos de trama. La diferencia entre ambos formatos es la longitud del campo llamado identificador. La trama base o simple, implementa un identificador

de once bits, mientras que en la trama extendida, el identificador ocupa veintinueve bits [2] [3] [30].



Figura 2.11: Trama CAN.

2.1.6.1 Trama simple o estándar

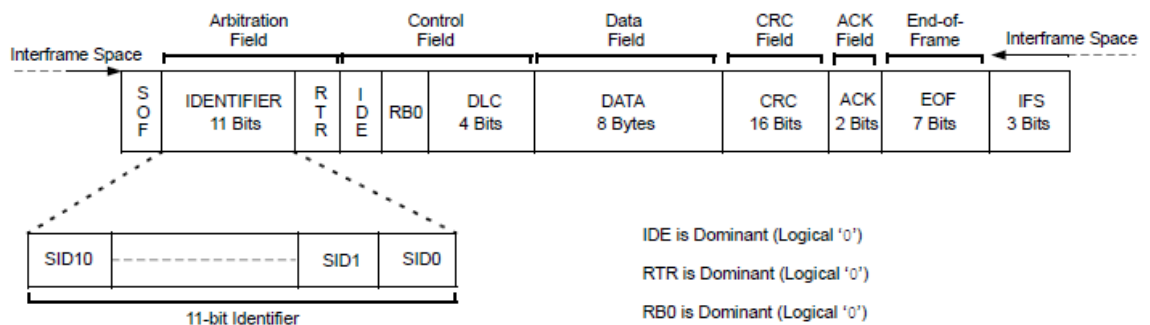


Figura 2.12: Trama estándar.

La trama estándar se compone de los campos que se describen a continuación:

- **SOF (Start Of Frame):** Campo que indica el comienzo de la trama. Consiste en un bit dominante ("0") empleado para la sincronización de los nodos conectados al bus.
- **IDENTIFIER:** En tramas estándar, este campo contiene once bits. Estos bits son empleados para el arbitraje de prioridades.

- RTR (*Remote Transmission Request*): Cuando RTR = 1, se trata de una trama remota para solicitud de datos o para comprobación; no contiene datos. En el caso de RTR = 0, se trata de una trama de datos. Las tramas de datos tienen prioridad a las tramas remotas.
- IDE (*IDentifier Extension*): Si este bit es dominante, es decir, IDE = 0, se trata de una trama estándar, con un campo de arbitraje de once bits. Cuando el valor de IDE es recesivo, IDE = 1, se trata de una trama extendida.
- RB0: Es un bit reservado.
- DLC (*Data Length Code*): Contiene el número de bytes que conforman el campo de datos. En el caso de tramas remotas, el contenido del campo DLC indica el número de bytes que se solicitan.
- DATA: Contiene los datos, que pueden variar entre 0 y 8 bytes, tantos como especifique el campo DLC.
- CRC (*Cyclic Redundant Check*): Es un campo de 16 bits. Los primeros 15 bytes son el resultado de aplicar un algoritmo de redundancia cíclica que debe asegurar la integridad de los datos. El último bit es siempre recesivo ("1"), y es empleado para cerrar el campo CRC.
- ACK (*ACKnowledge*): Este campo lo componen dos bits. Para la transmisión, ambos viajan en estado recesivo ("1"). El nodo que recibe el mensaje correctamente, corroborando el CRC, devuelve el primer bit en estado dominante ("0"). El segundo bit del campo ACK es un simple delimitador en estado recesivo.
- EOF (*End Of Frame*): Indica el final de la trama. El campo EOF contiene 7 bits recesivos.
- IFS (*Intermission Frame Space*): Es el mínimo número de bits, al menos tres, que separa dos tramas consecutivas. Esto permite a los nodos el procesamiento del mensaje antes de recibir uno nuevo. Tras el campo

IFS, el bus permanece en estado libre, estado recesivo, hasta que se produzca una nueva transmisión.

2.1.6.2 Trama extendida

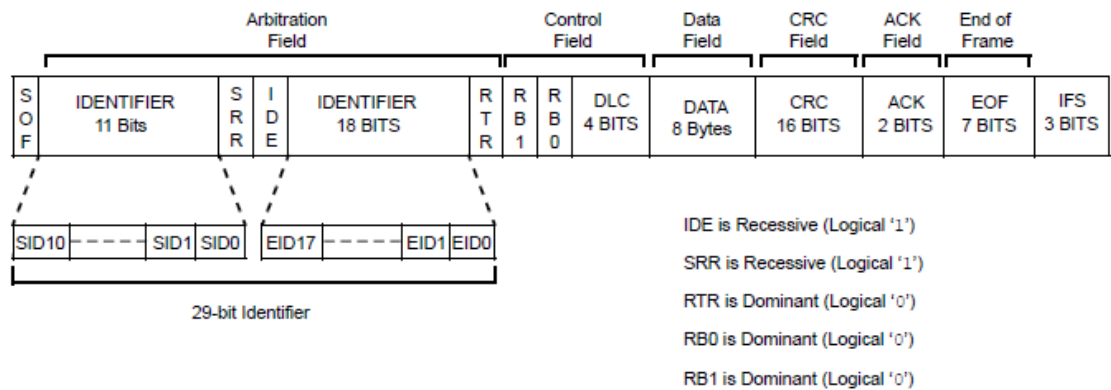


Figura 2.13: Trama extendida.

Una trama extendida se diferencia de una estándar en la longitud del identificador. En las tramas extendidas, el campo *identifier* se compone de 29 bits, los primeros 11 se denominan identificador base, y los 18 siguientes, extensión del identificador. Las tramas extendidas tienen el bit IDE en estado recesivo ("1").

Los dos tipos de trama pueden coexistir en un sistema siempre que todos los nodos soporten la discriminación. La prioridad es más alta para las tramas estándar.

Las tramas extendidas tienen algunas desventajas: el tiempo de latencia de bus es más alto, se requiere un mayor ancho de banda y hay una menor capacidad de detección de errores debido al diseño del polinomio del CRC.

Los controladores de CAN que soportan tramas extendidas, también son capaces de tratar las tramas estándar. Los controladores que sólo soportan tramas estándar, no interpretan las extendidas correctamente. Sin embargo hay controladores de CAN que sólo soportan tramas estándar pero reconocen las tramas extendidas y las ignoran.

CAN 2.0A se refiere a los sistemas que soportan tramas estándar y CAN 2.0B hace referencia a los sistemas que soportan tramas extendidas.

2.2 Herramientas Hardware

2.2.1 Microcontrolador

El microcontrolador elegido para llevar a cabo los objetivos pertenece a la familia de los PIC32 de Microchip Technology Inc. Se trata del PIC32MX795F512L, que satisface las necesidades técnicas del Proyecto, pues puede funcionar a alta frecuencia , dispone de controlador de CAN y las características de memoria FLASH y RAM son adecuadas

El PIC32MX795F512L es un microcontrolador de 32 bits con arquitectura RISC (*Reduced Instruction Set Computer*) [9] [24] [25] [26].

El core es un MIPS32 M4K de 5 etapas de segmentación (*5 stage-pipeline*).

NOTA: MIPS: *Microprocessor without Interlocked Pipeline Stage* –microprocesador sin bloqueos en las etapas de segmentación-.

La frecuencia máxima de trabajo es de 80 MHz.

La figura siguiente muestra el diagrama de bloques de un microcontrolador de la familia PIC32MX7xx.

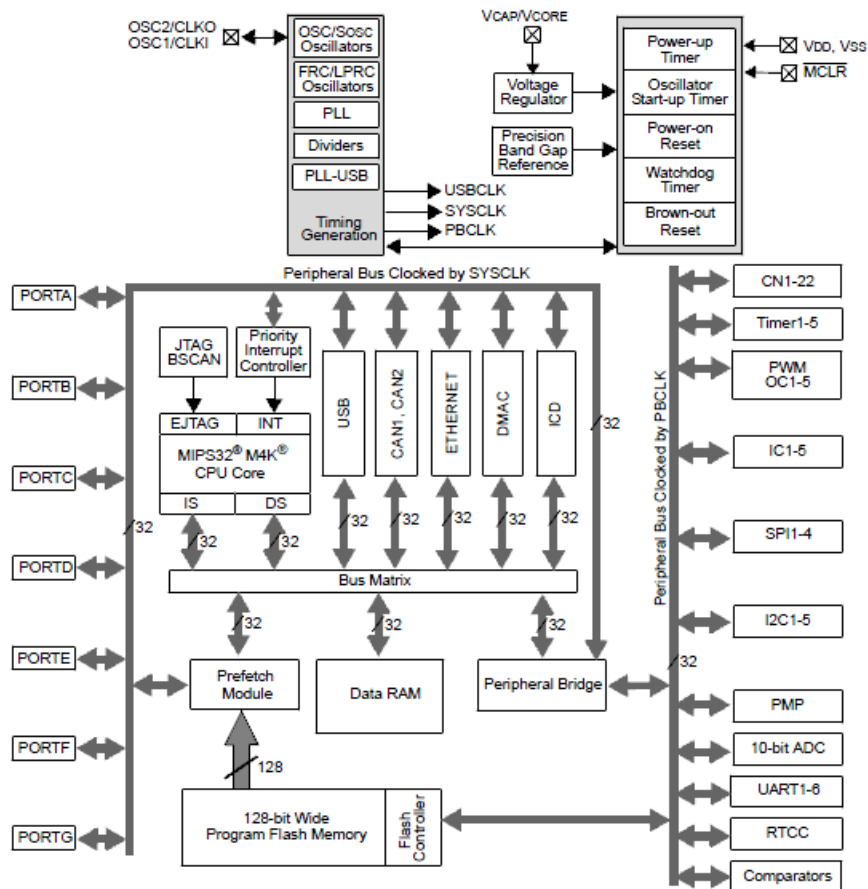


Figura 2.14: Diagrama de bloques de un PIC32MX7xx de Microchip.

Entre las características principales se encuentran:

- Rango de tensiones de funcionamiento de 2.3V a 3.6V.
- 512 Kb de memoria FLASH, con 12 Kb adicionales de *Boot Flash*.
- 128 Kb de memoria SRAM.
- Múltiples vectores de interrupción con prioridad programable individualmente.
- Watchdog Timer configurable con circuito oscilador RC integrado de bajo consumo.

Device	Pins	Program Memory (KB)	Data Memory (KB)	USB	Ethernet	CAN	Timers/Capture/Compare	DMA Channels (Programmable/Dedicated)	UART ^(2,3)	SPI ⁽³⁾	I ² C ^{TM(3)}	10-bit 1 Msps ADC (Channels)	Comparators	PMP/PSP	JTAG	Trace	Packages ⁽⁴⁾
PIC32MX764F128H	64	128 + 12 ⁽¹⁾	32	1	1	1	5/5/5	4/6	6	3	4	16	2	Yes	Yes	No	PT, MR
PIC32MX775F256H	64	256 + 12 ⁽¹⁾	64	1	1	2	5/5/5	8/8	6	3	4	16	2	Yes	Yes	No	PT, MR
PIC32MX775F512H	64	512 + 12 ⁽¹⁾	64	1	1	2	5/5/5	8/8	6	3	4	16	2	Yes	Yes	No	PT, MR
PIC32MX795F512H	64	512 + 12 ⁽¹⁾	128	1	1	2	5/5/5	8/8	6	3	4	16	2	Yes	Yes	No	PT, MR
PIC32MX764F128L	100	128 + 12 ⁽¹⁾	32	1	1	1	5/5/5	4/6	6	4	5	16	2	Yes	Yes	Yes	PT, PF, BG
PIC32MX775F256L	100	256 + 12 ⁽¹⁾	64	1	1	2	5/5/5	8/8	6	4	5	16	2	Yes	Yes	Yes	PT, PF, BG
PIC32MX775F512L	100	512 + 12 ⁽¹⁾	64	1	1	2	5/5/5	8/8	6	4	5	16	2	Yes	Yes	Yes	PT, PF, BG
PIC32MX795F512L	100	512 + 12 ⁽¹⁾	128	1	1	2	5/5/5	8/8	6	4	5	16	2	Yes	Yes	Yes	PT, PF, BG

Tabla 2.1: Características principales del PIC32MX7xx.

Entre las principales características de los periféricos se encuentran las siguientes:

- Operaciones atómicas SET, CLEAR e INVERT en registros de periféricos.
- 8 canales DMA.
- Módulo USB 2.0 full-speed con controlador OTG con canales DMA dedicados.
- Módulo Ethernet 10/100 Mbps con canales DMA dedicados.
- Módulo CAN2.0B con canales DMA dedicados.
- Cristal oscilador de 3 MHz a 25 MHz.
- Osciladores internos de 8 MHz y 32 kHz.

- 6 módulos UART, con soporte para RS-232 y RS-485.
- 4 módulos SPI.
- 5 módulos I2C.
- PLLs separados para los relojes de CPU y USB.
- Parallel Master & Slave Port (PMP/PSP) con hasta 16 líneas de dirección.
- Real-Time Clock and Calendar (RTCC).
- 5 timers de 16 bits.
- 5 entradas de captura.
- 5 salidas Compare/PWM.
- 5 pines de interrupción externa.
- Pines de entrada/salida capaces de variar su estado a 80 MHz.
- Corriente de entrada/salida (*sink/source*) de 18 mA/18mA en todos los pines I/O.
- Salida configurable open-drain en los pines I/O digitales.

Características de depuración:

- Dos interfaces de programación/depuración.
- Interfaz de dos hilos con acceso no intrusivo e intercambio de datos con la aplicación en tiempo real.
- Interfaz de cuatro hilos MIPS JTAG estándar.
- Rastreador no intrusivo de instrucciones basado en hardware.
- IEEE estándar 1149.2.

Características analógicas del PIC32MX7xx:

- Convertidor A/D de 10 bits, 16 canales.
- Ratio de conversión de 1 Msps (*Mega Samples Per Second*).
- Conversión operativa incluso en modos *idle* y *sleep*.
- Dos comparadores analógicos.

Puede encontrarse el *pinout* del microcontrolador PIC32MX795F512L en el Anexo.

La nomenclatura del PIC está descrita por la figura siguiente:

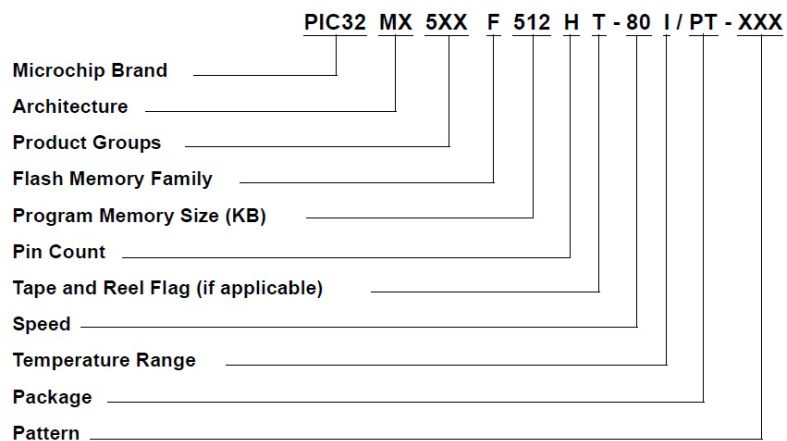


Figura 2.15: Nomenclatura de la familia PIC32.

2.2.2 Placa de Desarrollo CEREBOT MX7cK

CEREBOT MX7Ck es una placa de desarrollo basada en el microcontrolador de Microchip PIC32MX795F512L, y fabricada por Digilent Inc. Se trata de una placa adecuada para el trabajo con las herramientas MPLAB IDE [33].

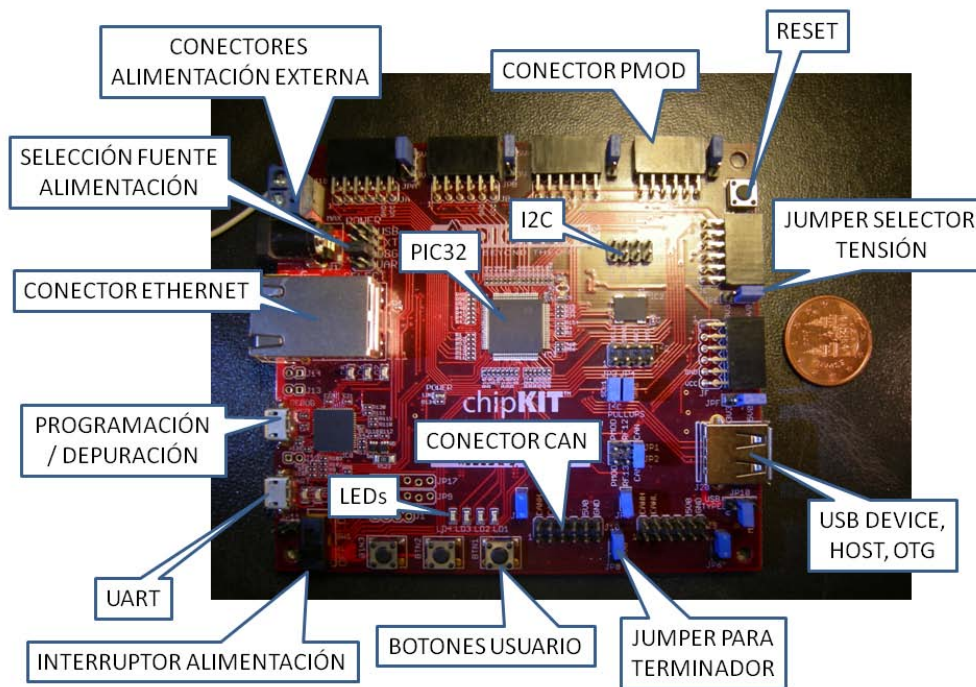


Figura 2.16: Placa CEREBOT MX7cK

Incorpora un sistema integrado de programación y depuración compatible con Microchip MPLAB IDE, de modo que no es necesario ningún hardware adicional.

Entre las principales características del CEREBOT MX7Ck, se incluyen las siguientes:

- Microcontrolador PIC32MX795F512L.
- Soporte para programación y depuración en el entorno de MPLAB IDE.
- Seis conectores *Pmod* para tarjetas periféricas.
- Ethernet 10/100.

- Soporte para USB *Device*, *Host* y *OTG*.
- Dos interfaces de CAN.
- Cuatro LED.
- Varias opciones de alimentación.
- Protección contra cortocircuitos y descargas electrostáticas en pines I/O.

La siguiente figura muestra un diagrama de bloques de la placa CEREBOT MX7cK de Digilent.

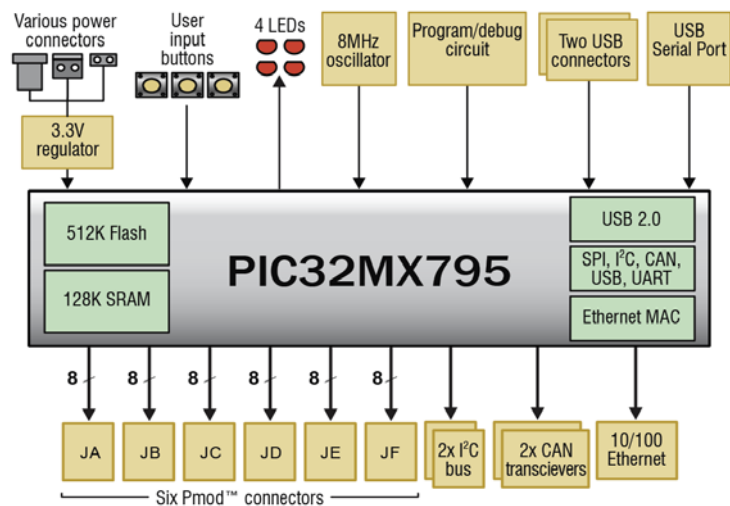


Figura 2.17: Diagrama de bloques de la placa CEREBOT MX7cK.

La alimentación del CEREBOT MX7cK toma efecto al accionar el interruptor principal SW1. El bloque de *jumpers* J3 selecciona la fuente de alimentación empleada. J3 proporciona las siguientes cuatro posibles posiciones:

- USB: La alimentación proviene del conector J19. Se emplea esta opción cuando el CEREBOT MX7Ck es usado para la implementación de un dispositivo USB alimentado
- EXT: La alimentación es suministrada por uno de los conectores de alimentación externa.
- DBG: La alimentación es proviene del conector J15, llamado USB DBG. Es la alimentación típicamente usada en labores de programación y depuración.
- UART: La alimentación es proporcionada por el conector UART USB, J2.

CEREBOT MX7Ck está preparado para una alimentación externa comprendida entre 3.6 VDC y 5 VDC. Cuando la placa trabaja por debajo de los 5 VDC, algunas de las funcionalidades no operarán; tal es el caso de los transceptores de CAN.

El bus VCC5V0 alimenta el regulador principal de tensión de la placa, el voltaje de entrada para el bus USB cuando se usa el CEREBOT MX7Ck como host USB, la alimentación de los transceptores de CAN y la parte de 5VDC de los conectores *Pmod*.

La fuente de la alimentación seleccionada por J3 aparecerá en el bus de alimentación de entrada, llamada VIN en los esquemas. Esta tensión es regulada por IC11 a 3.3 VDC para alimentar el programador/depurador.

Existen dos conectores para alimentar el CEREBOT MX7Ck mediante una fuente de alimentación externa: J17 y J18.

J17 es un conector de tipo coaxial con el alma conectada a la tensión positiva. J18 es un conector que fija los hilos de entrada mediante tornillos. J18 ofrece una mejor fijación del cable de alimentación, frente a la comodidad y rapidez de J17. Los conectores J17 y J18 están conectados en paralelo, y ambos asociados a la posición EXT del bloque de selección J3.

Los transceptores MCP2551 (IC4, IC5) requieren 5 VDC para operar [28].

El microcontrolador PIC32MX795F512L funciona con una tensión de 3.3 VDC, la cual es suministrada por el bus VCC3V3. El regulador principal encargado de esta misión es capaz de suministrar una corriente de 600 mA. El PIC32 consume aproximadamente 85 mA cuando funciona a 80 MHz. Los transceptores de CAN MCP2551 pueden consumir hasta 75 mA cada uno al operar en el bus CAN. El resto de la circuitería consume entre 10 mA y 20 mA. El resto de la corriente, unos 420 mA, están disponibles para los conectores *Pmod* y dispositivos I²C.

Como resumen de lo anteriormente expuesto, en el Anexo de *Hojas de Características* se encuentra el esquema de la alimentación de la placa CEREBOT MX7cK.

El PIC32 puede suministrar o recibir un máximo de 18 mA en cada uno de sus pines de entrada/salida. Sin embargo, para mantener la tensión de salida en el rango de tensión especificado (V_{OL} 0.4V, V_{OH} 2.4V) la corriente por los pines debe ser restringida a 12 mA. La máxima corriente, emitida o recibida por todos los pines de entrada/salida simultáneamente es de 200 mA.

Los pines de entrada/salida del PIC32 son tolerantes a 5VDC. Es seguro aplicar señales de 5VDC directamente en estos pines sin riesgo de dañar el microcontrolador.

Como se mencionó previamente, la placa CEREBOT MX7cK está provista de protección contra cortocircuitos y contra ESD (*Electrostatic Discharge*) mediante la circuitería que se muestra en las figuras siguientes.

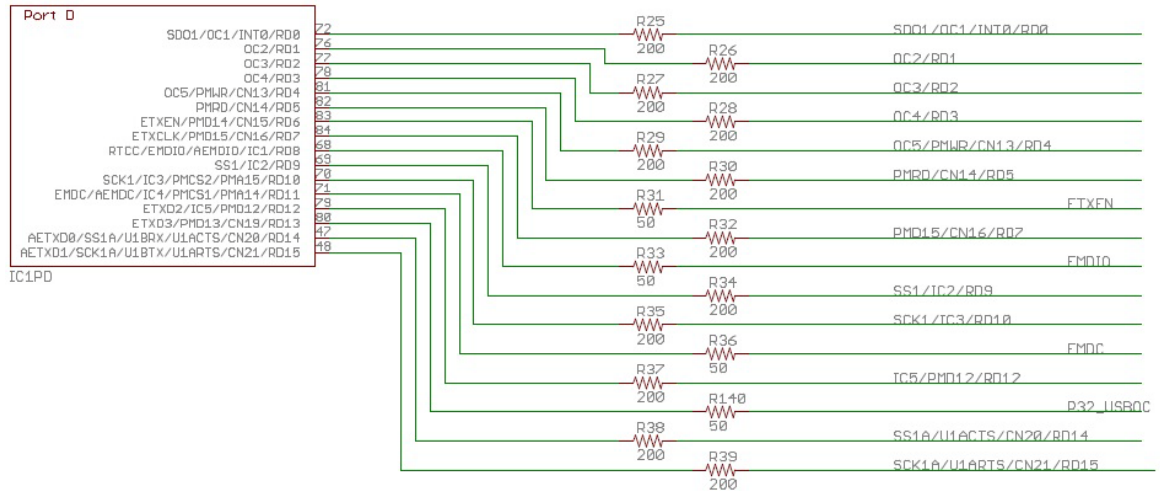


Figura 2.18: Resistencias de 200Ω para protección contra cortocircuitos.

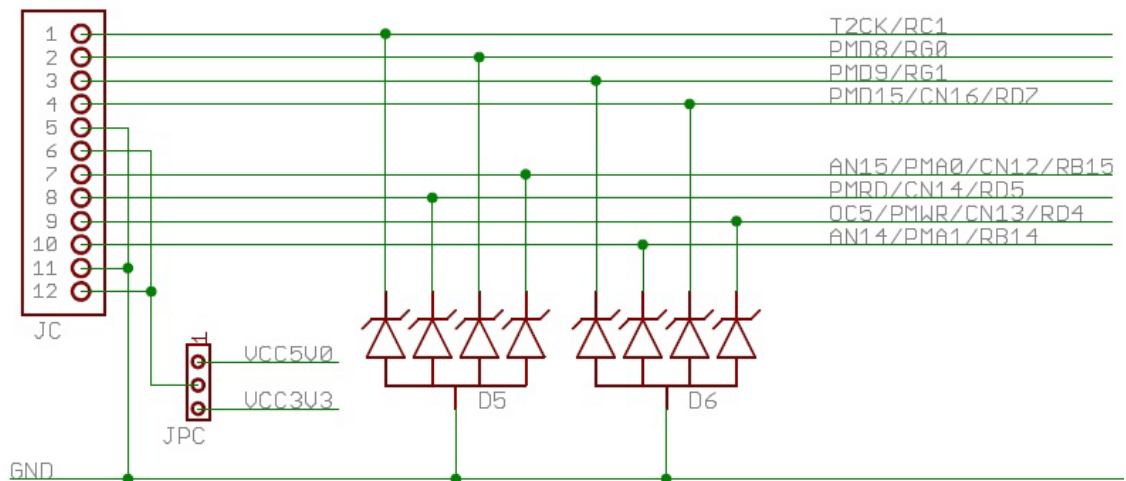


Figura 2.19: Diodos de protección ESD.

La resistencia en serie previene daños en los bloques de entrada/salida del microcontrolador en caso de que el pin se conecte por error a VCC o GND, o de que dos salidas sean conectadas entre sí. La protección contra descargas electrostáticas (ESD) es realizada por los diodos (D5 y D6 en la figura anterior).

El PIC32MX795 contiene dos módulos de CAN independientes. Estos dos módulos, en combinación con dos transceptores Microchip MCP2551, permiten a la placa CEREBOT MX7Ck operar hasta en dos buses CAN independientes.

El PIC32MX795 dispone de dos juegos de pines que pueden ser usados para conectar cada módulo de CAN al transceptor. La placa CEREBOT MX7Ck está diseñada para usar el juego alternativo de pines, no el estándar. La selección de los pines alternativos se hace a través de la directiva *pragma*. La siguiente línea debe aparecer en el módulo del programa principal:

```
#pragma config FCANIO = OFF
```

La red CAN requiere que los nodos de los extremos tengan una resistencia de 120Ω . La placa CEREBOT MX7Ck provee de terminadores de 120Ω y jumpers para habilitarlos o deshabilitarlos dependiendo de la situación de la placa en la red CAN. El jumper J5 habilita/deshabilita el terminador de 120Ω del conector CAN1. El jumper J7 habilita/deshabilita el terminador de 120Ω del conector CAN2.

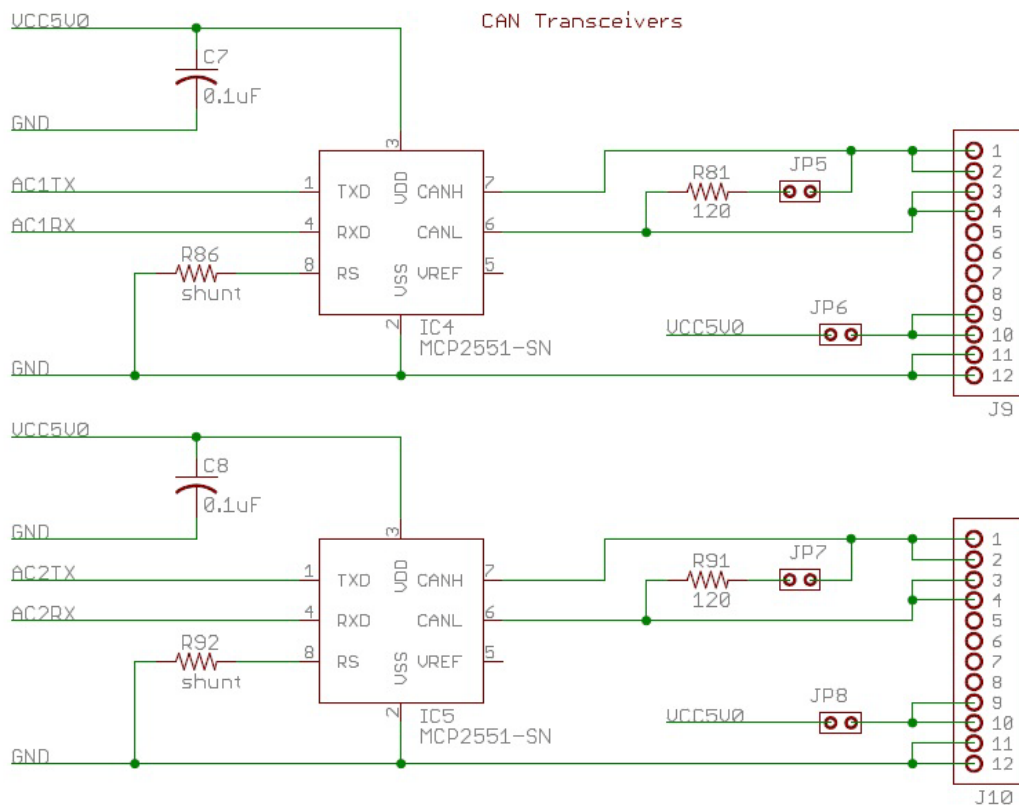


Figura 2.20: Transceptores, terminadores y conectores de CAN.

Es de especial mención el tipo de transceptor de CAN que incorpora la placa CEREBOT MX7Ck. Se trata del MCP2551 fabricado por Microchip, y cuyas características principales se enumeran a continuación:

- Soporta operaciones de 1Mbps.
- Implementa los requerimientos ISO-11898 de capa física.
- Adecuado para sistemas de 12V y 24V.
- Pendientes controladas para reducción de emisiones RFI.
- Detección de fallo de tierra en el pin TXD.
- Un nodo sin alimentación no perturba el bus CAN.
- Operación en *standby* de bajo consumo.
- Protección contra cortocircuitos.
- Protección contra altas tensiones en transitorios.
- Apagado automático como protección térmica.
- Posibilidad de conexión de hasta 112 nodos.
- Alta inmunidad al ruido gracias a la implementación del bus diferencial.
- Rango de temperaturas: Industrial (I), desde -40°C hasta +85°C; Extendido (E), desde -40°C hasta 125°C.

Pin Number	Pin Name	Pin Function
1	TXD	Transmit Data Input
2	VSS	Ground
3	VDD	Supply Voltage
4	RXD	Receive Data Output
5	VREF	Reference Output Voltage
6	CANL	CAN Low-Level Voltage I/O
7	CANH	CAN High-Level Voltage I/O
8	Rs	Slope-Control Input

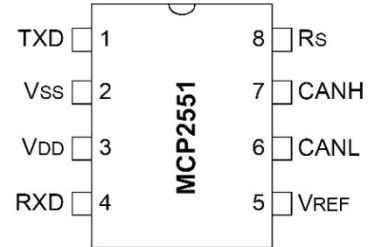


Figura 2.21: Descripción de los pines del transceptor MCP2551.

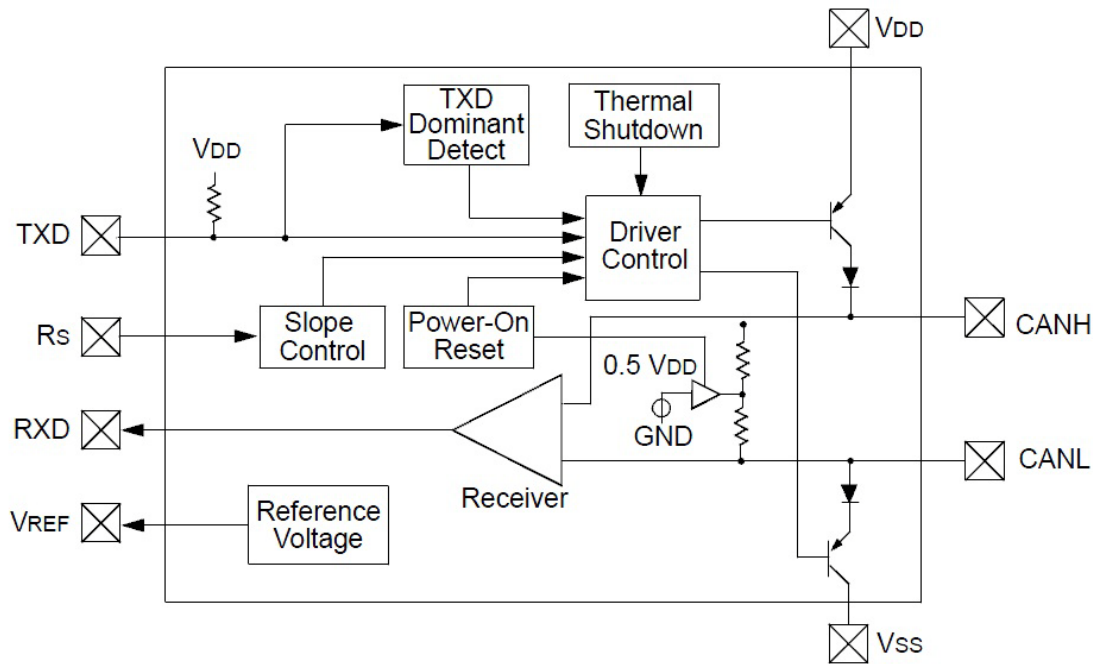


Figura 2.22: Esquema del transceptor MCP2551.

Para su empleo en la ubicación final de trabajo, y como ayuda para la programación y depuración, el CEREBOT MX7cK dispone de cuatro LED como salida y tres botones como entradas del usuario.

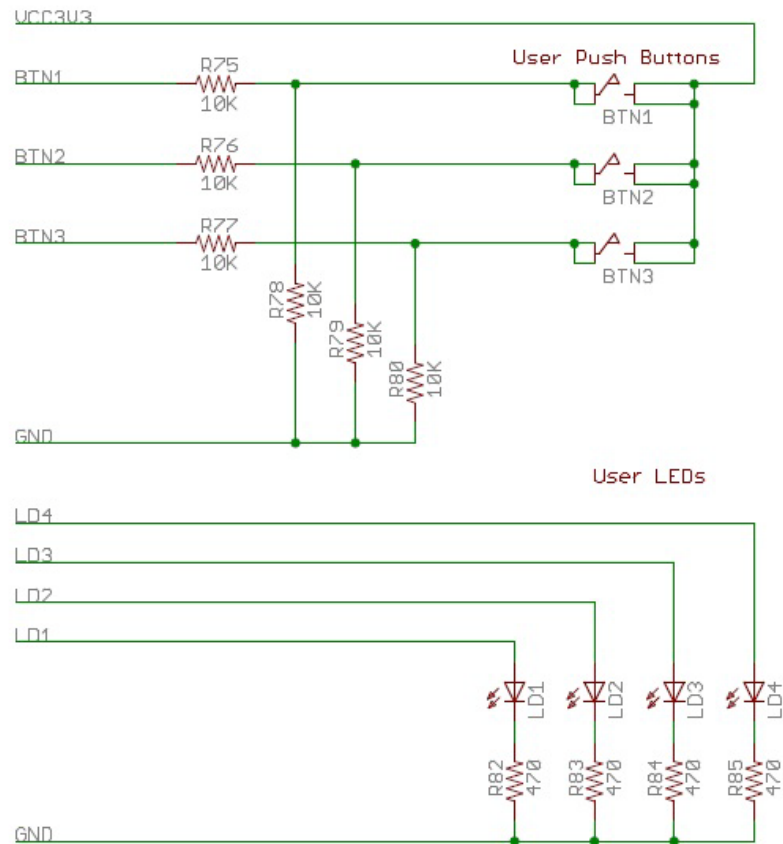


Figura 2.23: Esquema de botones y LED del CEREBOT MX7cK.

Los botones BTN1 y BTN2 están conectados al puerto G [13], bits 6 y 7. El botón BTN3 se está conectado al bit 0 del puerto A. Para realizar la lectura del estado de los botones, los bits 6 y 7 del puerto G y el bit 0 del puerto A deben definirse como entradas estableciendo los bits correspondientes de los registros TRISG y TRISA. El estado del botón se obtiene entonces por la lectura de los registros PORTG y PORTA. Cuando se pulsa un botón, el bit correspondiente será alto ("1"). Los pines utilizados por los LED y los botones no son empleados por ningún conector; están plenamente dedicados. Los cuatro LED están conectados a los bits 12 a 15 del puerto G. El LED1 está conectado a al bit 12, el LED2 al bit 13 y así sucesivamente. Estos cuatro pines, también están dedicados a los LED y no aparecen en ningún conector. Para el uso de los LED es necesario definir los bits 12 a 15 del puerto G como salidas. Poner un "1" en la salida del puerto significa encender el LED.

2.2.3 Interfaz USB-CAN

Para el desarrollo del Sistema de Control es necesario disponer de una herramienta que permita transmitir tramas de CAN que puedan ser leídas por el módulo de CAN del PIC32MX795F512L a través del MCP2551.

En John Deere Ibérica se dispone de varias unidades del interfaz NI USB-8473 de National Instruments [32].



Figura 2.24: Interfaz NI USB-8473 de National Instruments.

Las principales características se resumen a continuación.

- Compatible con Hi-Speed USB, energizado por bus.
- Arbitraje de mensaje CAN 2.0A (11 bits) y CAN 2.0B extendido (29 bits).
- Controlador CAN Philips SJA1000 y capa física ISO 11898.
- Transceptor CAN Philips TJA1041 de alta velocidad.
- Estampa de tiempo del hardware – resolución de 1 μ s.
- Registro 100% de la carga a 1 Mb/s sin caída de marcos.

La salida del NI USB-8473 es un conector Canon DB9 macho, tal como muestra la siguiente figura:

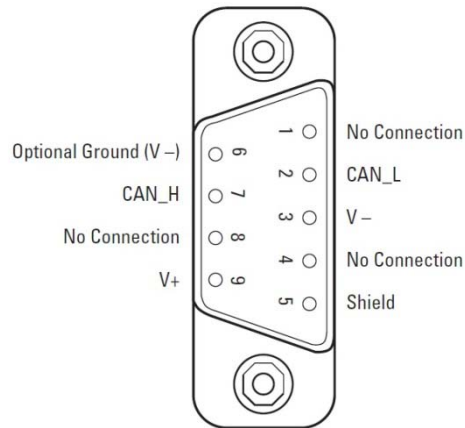


Figura 2.25: Conector de la interfaz NI USB-8473.

Para la conexión del NI USB-8473 a la placa CEREBOT MX7Ck se ha fabricado manualmente un cable con un conector DB9 hembra en uno de los extremos y un conector DIN hembra de doble vía y paso 1/10", como muestra la siguiente figura:



Figura 2.26: Cable de conexión entre NI USB-8473 y placa CEREBOT.

El cable de conexión es de sólo dos hilos: CAN_L y CAN_H.

2.2.4 Relés de Estado Sólido

Tanto las entradas como las salidas del módulo de seguridad PILZ PNOZ Multi M1P son de 24V.

Como se mencionó con anterioridad, las salidas de los puertos del PIC32MX795F512L ofrecen una tensión de 5V [13]. Por esta razón es necesario acondicionar las salidas y entradas del microcontrolador a los requerimientos del módulo PILZ [29] [36]. Se ha recurrido a relés de estado sólido para esta labor [34] [35].

El relé de estado sólido DEK-OE 5DC/24DC/100kHz permite la adaptación de la tensión de salida de 5V del PIC32 a la entrada de 24V del PILZ, y está diseñado para su instalación en carril DIN.



Figura 2.27: Serie DEK de relés de Phoenix Contact.

La versión DEK-OE-5DC/24DC/100KHZ dispone de un opto acoplador de entrada que proporciona el aislamiento galvánico fundamental por tratarse con los puertos de un microcontrolador.

La frecuencia de transmisión es de 100 kHz. El relé posee indicación luminosa y circuito de protección en la entrada y la salida.

La siguiente figura ilustra el esquema electrónico de relé DEK-OE-5VDC/24VDC.

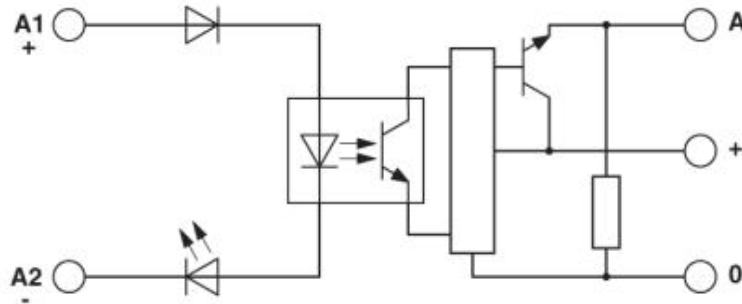


Figura 2.28: Esquema del relé DEK-OE-5VDC/24VDC/100kHz

Las características del relé de estado sólido DEK-OE-5VDC/24VDC/100kHz se enumeran a continuación:

Entrada	
Tensión nominal U_N	5 VDC \pm 20%
Corriente típica a la tensión U_N	7 mA
Frecuencia de transmisión	100 kHz
Tiempo típico de respuesta	1.5 μ s
Tiempo típico de apagado	2 μ s
Salida	
Rango nominal de tensiones	4 VDC ... 30 VDC
Corriente límite	50 mA
Caída de tensión a corriente máxima	\leq 0.5 VDC
Circuito de protección	Diodo protector

Tabla 2.2: Características del relé DEK-OE-24VDC/5VDC/100kHz

Para la adaptación de salida de 24V del módulo PILZ a la entrada del microcontrolador de 5V, se emplean relés de estado sólido de introducción, con contrafase de salida de tensión continua, frecuencia de transferencia de 100 kHz, con indicador luminoso y circuito de protección en circuito de entrada y salida.

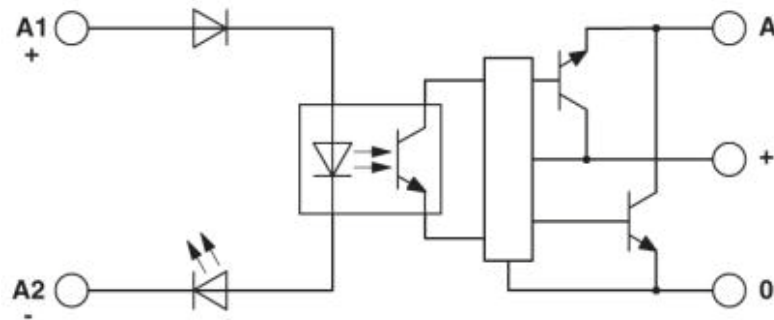


Tabla 2.3: Esquema del relé DEK-OE-24VDC/5VDC/100kHz.

Las características principales del relé de estado sólido DEK-OE-24VDC/5VDC/100kHz se enumeran a continuación:

Entrada	
Tensión nominal U_N	24 VDC
Corriente típica a la tensión U_N	8 mA.
Frecuencia de transmisión	100 kHz
Tiempo típico de respuesta	1 μ s
Tiempo típico de apagado	2 μ s
Circuito de protección	Diodo contra inv. polaridad
Salida	
Rango nominal de tensiones	4 VDC ... 18 VDC
Corriente límite	50 mA.
Caída de tensión a corriente máxima	≤ 1.2 VDC
Circuito de protección	Diodo supresor sobretensión

Tabla 2.4: Características del relé DEK-OE-24VDC/5VDC/100kHz.

El módulo PILZ puede suministrar hasta 2 A en sus salidas, pero se opta por los relés de estado sólido por homogeneizar el conjunto [29].

2.2.5 Fuente de Alimentación

Para las tareas de programación, pruebas y depuración, se emplea normalmente a alimentación proporcionada por el conector de USB de la tarjeta CEREBOT. Para la ubicación definitiva y autónoma, es necesario alimentar la tarjeta con 5 VDC a través del conector J18.

La alimentación de la placa CEREBOT MX7Ck se realiza a través de la fuente S8VS03005 de OMRON.

Se trata de una fuente de alimentación conmutada modular con entrada de tensión alterna universal de bajo perfil para instalación en carril DIN. Su salida es de 5V y 3^a. La tensión de salida es ajustable. Está protegida contra corto circuito y sobrecarga y tiene una alta eficiencia. Posee indicadores de tensión y de estado.



Figura 2.29: Fuente de alimentación para carril DIN, OMRON S8VS03005.

La siguiente tabla enumera las características esenciales de la fuente de alimentación.

Tensión de salida	5V
Corriente máxima de salida	4A
Rango de tensión de salida	5..5.1V
Tensión de salida para LED ON	3V
Eficiencia típica	74%
Carga mínima	0
Rizado	10mV
Tensión de entrada nominal	100..240VAC
Rango de tensiones de entrada	90..264VAC
Protección por sobrecarga	105..160% de U_N

Tabla 2.5: Características de la fuente de alimentación OMRON S8VS03005.

2.3 Herramientas Software

2.3.1 Entorno de Desarrollo MPLAB IDE

MPLAB es una interfaz gráfica creada para Microchip y otros sistemas de desarrollo. Soporta compiladores, emuladores y depuradores de Microchip así como muchas otras herramientas de diversos fabricantes.

La versión empleada es MPLAB IDE v8.83, de libre distribución [10] [11] [12].

Proporciona un fácil manejo del proyecto de desarrollo con un entorno intuitivo. Ofrece herramientas de programación y depuración, volcado de datos desde el microcontrolador, puntos de ruptura (hasta seis *breakpoints* en la versión gratuita).

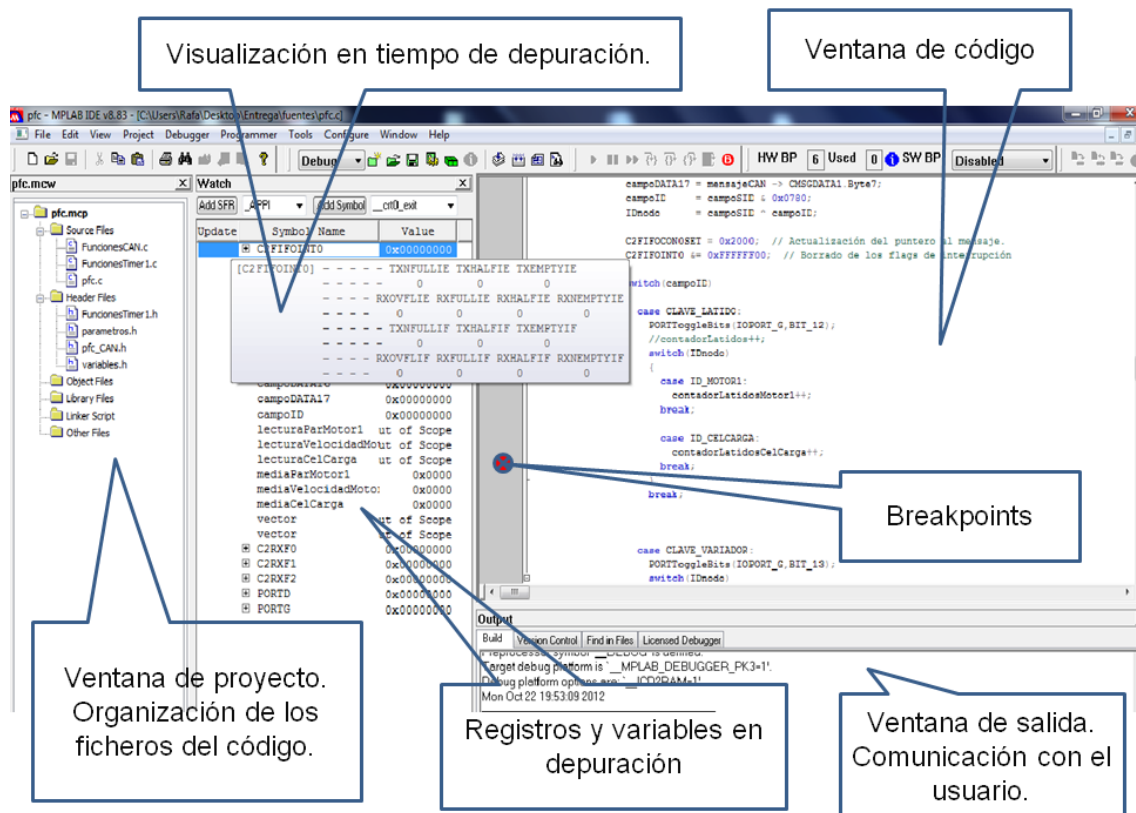


Figura 2.30: Entorno gráfico MPLAB v8.83.

Cuando se crea un nuevo proyecto, es imprescindible seleccionar adecuadamente el microcontrolador, asegurando la selección del dispositivo PIC32MX795F512L.

Para el empleo del programador y depurador del CEREBOT MX7cK,, es necesario realizar la selección

Debugger→Select Tool→Licensed Debugger

tal como ilustra la Figura.

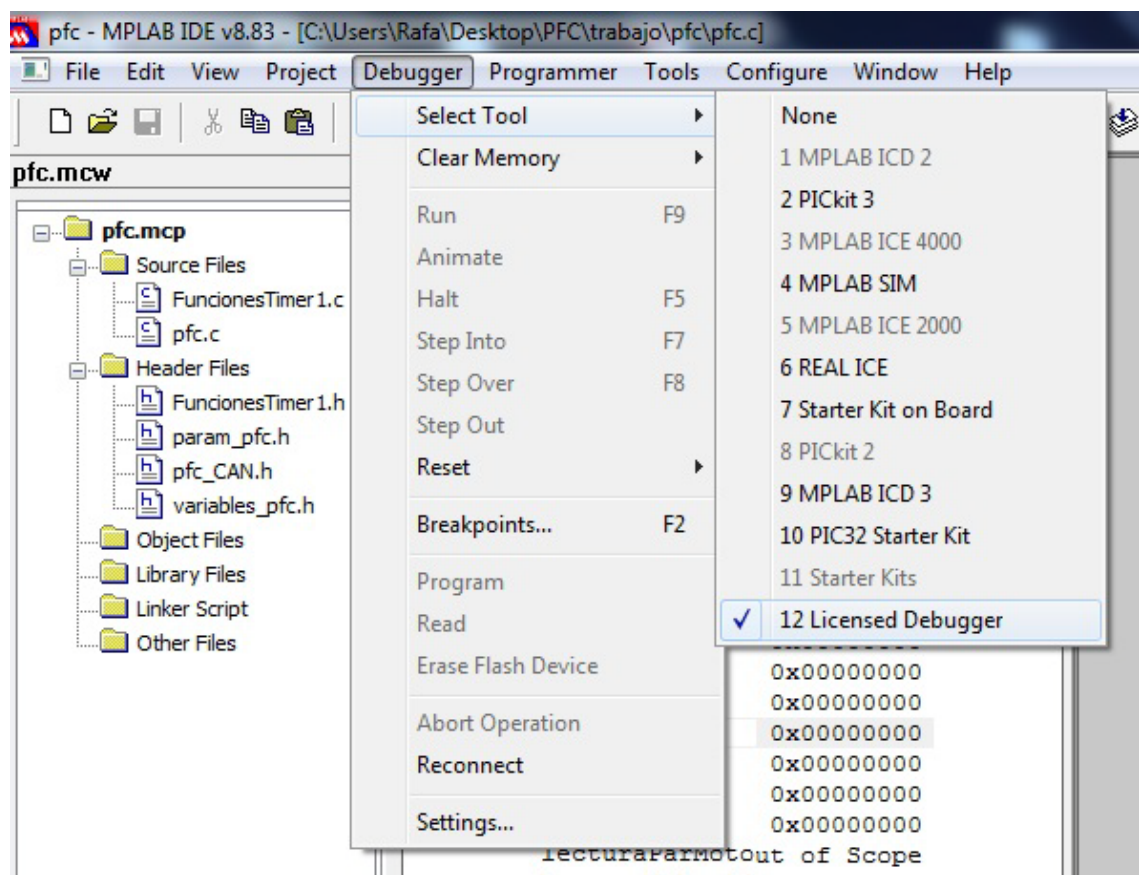


Figura 2.31: Selección del programador/depurador bajo licencia Microchip.

2.3.2 Programador/Depurador

La programación y depuración son tareas encargadas a la placa CEREBOT MX7CK, la cual es perfectamente compatible con MPLAB.

Cuando se usa MPLAB (versiones 8.63 y posteriores) con la tarjeta CEREBOT MX7cK, la programación y la depuración del firmware del microcontrolador PIC32MX795F512L se realiza empleando un circuito programador y depurador embebido con licencia de Microchip [10] [12] [19].

El depurador bajo licencia usa el conector USB (micro-B) J15 etiquetado como *DEBUG* en la placa CEREBOT. Cuando MPLAB IDE detecta este depurador, automáticamente examina el número de versión del firmware y ofrece una actualización en caso de que sea necesaria para la versión de MPLAB que se esté utilizando [21] [22] [23].

Para la programación y depuración del PIC32, la interfaz emplea dos pines del microcontrolador. El PIC32 dispone de dos pares de pines alternativos para la interfaz: PGC1/PGD1 y PGC2/PGD2. Por defecto se emplea el par PGC2/PGD2, pero debido a conflictos con los pines del microcontrolador, la placa CEREBOT MX7CK está diseñada para el empleo del par PGC1/PGD1. Por esta razón es necesario seleccionar el uso del par PGC1/PGD1 para la interfaz de programación y depuración. La selección se realiza mediante la directiva *pragma*. La siguiente línea debe ser escrita para configurar al microcontrolador para el uso de un depurador bajo licencia.

```
#pragma config ICSEL = ICS_PGx1
```

2.3.3 Entorno de Pruebas del Módulo CAN

LabVIEW 2011 de 32 bit es la herramienta empleada para la programación del módulo de lectura y escritura de CAN. La estructura del programa es simple al igual que su interfaz gráfica. Se compone de una función de lectura de las tramas almacenadas en el hardware (módulo *USB to CAN –NI8473-* de National Instruments), una función de escritura de tramas y varias funciones auxiliares para el filtrado y la grabación de las

tramas leídas. Las funciones de lectura y escritura de CAN se han desarrollado con las librerías Frame API (Measurement I/O >> CAN) de LabVIEW. Todas las funciones se ejecutan en un bucle de control del módulo Real-Time de LabVIEW. Esto mejora las propiedades del sistema en términos de tiempo real aunque no las asegura por completo ya que se trabaja con un sistema bajo Windows [27].

Para la ejecución del programa no es necesario tener instalado LabVIEW 2011. Se han desarrollado un instalador y un ejecutable con el módulo Application Builder de LabVIEW y el programa se puede lanzar en cualquier máquina con Windows 32 bits.

La programación en LabVIEW es fundamentalmente gráfica. El código generado en el diagrama de bloques genera automáticamente controladores e indicadores en el panel frontal (interfaz gráfica). En la siguiente figura se muestra la interfaz gráfica del programa principal. En la parte izquierda se muestran las tramas recibidas y en la parte izquierda las tramas a escribir. Mediante una serie de botones pueden borrarse el registro de tramas, filtrarlas, comenzar la escritura, grabar, etc.

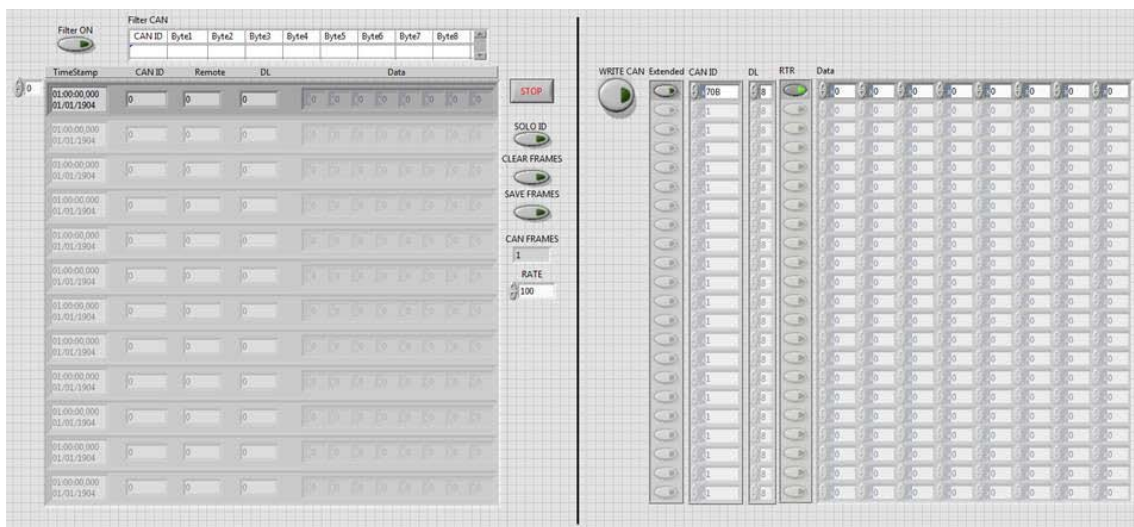


Figura 2.32: Panel de trabajo para el desarrollo de la interfaz en LabVIEW.

El código es el mostrado en la Figura 2.21. Se inicia con la configuración del USB2CAN (Open CAN) y la apertura de un nuevo archivo de grabación. Dentro del bucle, se encuentran las diferentes funciones de escritura y lectura, así como otras

auxiliares. Finalmente, se cierran la sesión del módulo USB2CAN y el archivo de grabación de tramas.

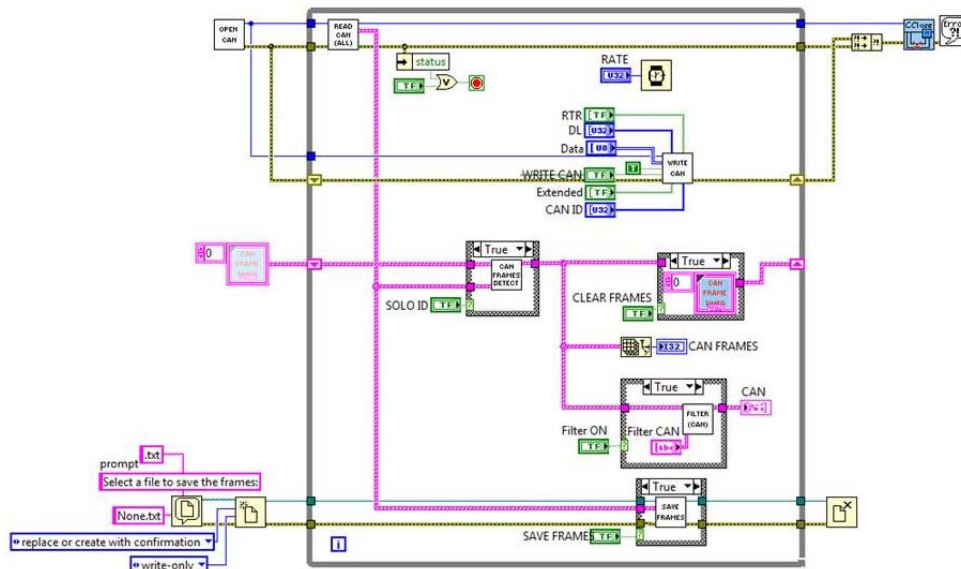


Figura 2.33: Código de la interfaz para el envío de tramas CAN.

El código del módulo *CANWriter* se muestra en la figura inferior. Se trata de una función de escritura y de una función de comprobación de estado para asegurar que el módulo recibe el *acknowledge* (ACK) o señal de confirmación.

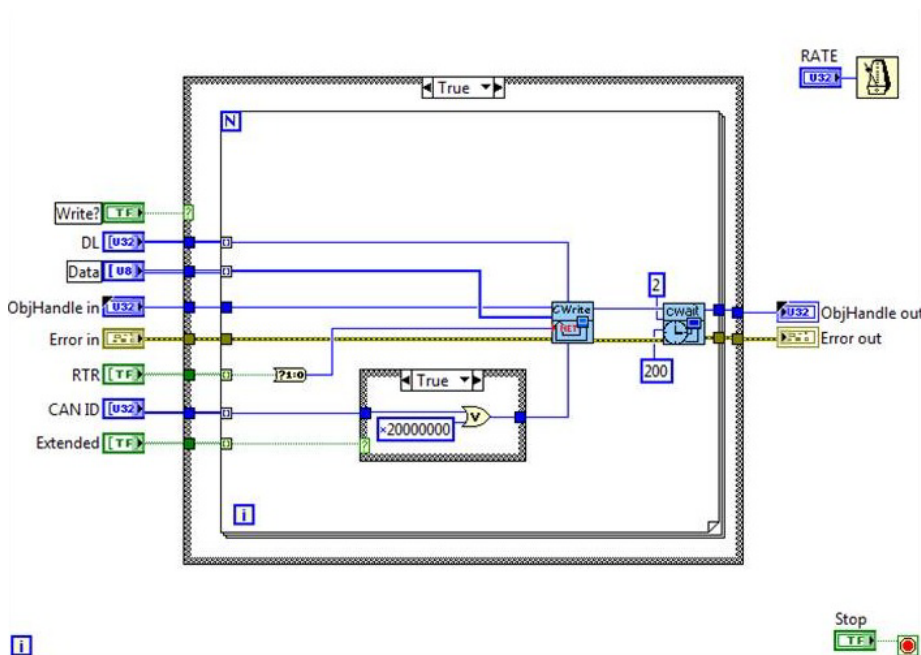


Figura 2.34: Código del módulo CANWrite.

2.3.4 Módulo de seguridad PILZ

La programación del módulo de seguridad PILZ PNOZ Multi M1P se lleva a cabo bajo un entorno gráfico [29].

Deben activarse las salidas hacia la placa CEREBOT MX7Ck con las señales de petición de acceso a la sala y de estado del anclaje de la puerta de acceso. Como entradas, deben habilitarse las señales de *Parada de Emergencia* y de *Apertura de Puerta*.



Figura 2.35: Módulo de seguridad PILZ PNOZ Multi M 1 P

La programación del módulo PILZ es ajena a este Proyecto.

Desarrollo del Sistema

3.1 Condiciones de partida

En el interior de la sala del banco existen numerosos elementos que deben ser controlados, tanto actuadores como sensores.

En cuanto a los actuadores, se cuenta con motores eléctricos gobernados por variadores alimentados por una fuente de alimentación que, a su vez, recibe la energía de un centro de transformación ubicado en las propias instalaciones de la factoría.

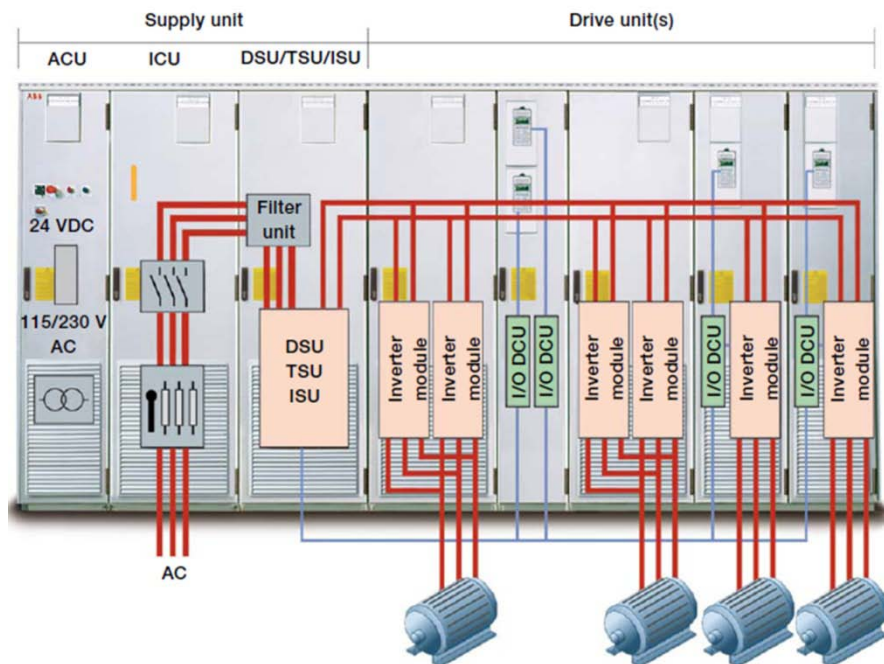


Figura 3.1: Armarios de potencia y de variadores.

Otro caso de actuador es la centralita hidráulica, que se encarga de proporcionar diferentes presiones a subsiguientes actuadores presentes en el ensayo. Estos actuadores pueden estar situados en la propia pieza a probar, o pueden formar parte de los sistemas de accionamiento necesarios para el ensayo.



Figura 3.2: Centralita hidráulica.

Las centralitas cuentan con su sistema propio de control, de modo que el suministro de caudales presurizados se comanda mediante el accionamiento remoto de las electroválvulas. El encendido y apagado de las bombas de las centralitas está automatizado, así como la refrigeración del aceite en el intercambiador de calor.

Las centralitas hidráulicas están dotadas de numerosos sensores, principalmente de presión. También disponen de sensores (generalmente termopares o PT100) que controlan la temperatura del aceite en el circuito, en el intercambiador de calor, en el tanque, etc.



Figura 3.3: Intercambiador de calor con sensores de temperatura.



Figura 3.4: Actuador. Cilindro hidráulico.

Según el tipo de ensayo que se realice, es posible disponer de accionamientos neumáticos, que reciben el aire comprimido de alguna de las subestaciones de la factoría. El funcionamiento se consigue comandando electroválvulas del sistema neumático del ensayo. Se controla el estado de los actuadores mediante transductores de posición, presión y células de carga, entre otros.



Figura 3.5: Actuador. Cilindro neumático.

Los ejemplos de sensores son múltiples. Existen sensores de temperatura que monitorizan las piezas a ensayar así como el estado térmico de otros sistemas de la prueba.



Figura 3.6: Sensores de temperatura, resistivo y termopar.

Los sensores de presión proporcionan la lectura del interior de las probetas y las presiones del aceite o del aire comprimido.



Figura 3.7: Sensor de presión.

Se cuenta también con células de carga para la lectura de fuerzas aplicadas.



Figura 3.8: Célula de carga.

Los torsiómetros (comúnmente disponen de salida CAN) se ocupan de la lectura de los valores del par a que se someten dos elementos entre sí.



Figura 3.9: Torsiómetro.

Existen *encoders* en los motores, encargados de proporcionar el ángulo de giro y la velocidad.



Figura 3.10: Encoder asociado al eje de un motor.

El control y la visualización del estado de los elementos anteriores se realizan como muestra la siguiente figura:

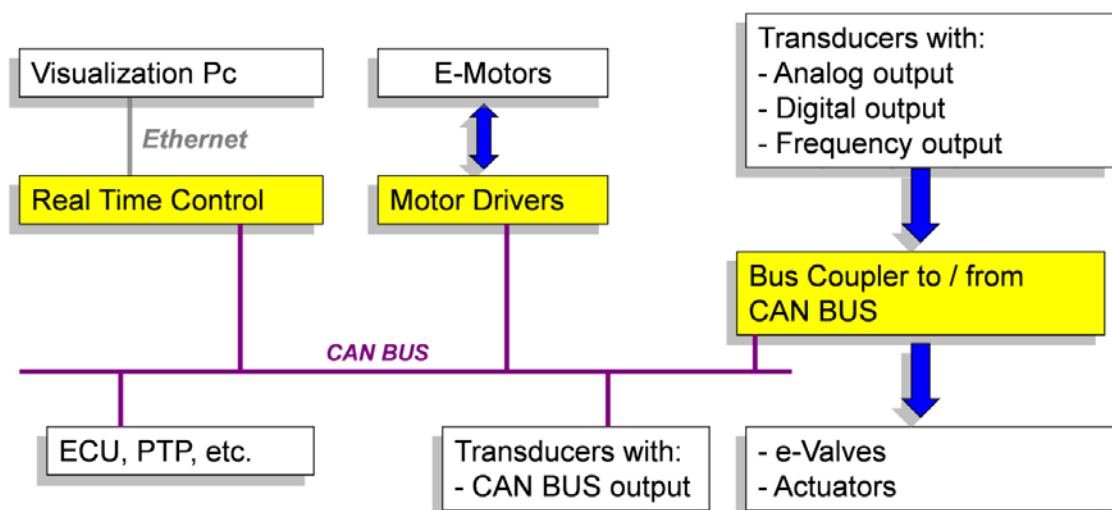


Figura 3.11: Arquitectura del sistema de ensayos.

El PC de visualización genera un ensayo automático y lo envía por Ethernet a un computador de tiempo real, que a su vez compone los mensajes necesarios y los transmite al CAN-bus de modo que los actuadores seleccionados realizan su función cuando su controlador de CAN identifica la trama.

La recepción por parte del PC opera de manera similar. La señal acondicionada de los sensores es transformada en tramas reconocibles por CAN y es recogida del bus por el controlador de CAN del computador de tiempo real. El computador de tiempo real pasa los valores leídos al PC de visualización para su almacenamiento y para que el operador pueda observar la evolución del ensayo. El PC de visualización también se encarga de manejar las alarmas del ensayo, esto es, cuando los valores leídos alcanzan unas condiciones deseadas o cuando se ha producido una rotura o error, la prueba finaliza.

Los bancos de ensayo de John Deere Ibérica están migrando a sistemas de control basados en CAN. El objetivo del departamento PV&V es realizar los ensayos de la manera más autónoma posible. Es por esta razón que el control es necesario en todos los niveles.

Puede suceder que debido a las condiciones adversas de las pruebas (humedad, temperatura, vibraciones...) y también a su, generalmente, larga duración, las comunicaciones fallen. Un fallo en el bus CAN supone un descontrol sobre el ensayo, con el consiguiente riesgo de pérdidas económicas, temporales, etc.

3.1.1 Especificaciones

El siguiente esquema muestra el esquema de un banco de ensayo genérico de John Deere Ibérica.



Figura 3.12: Esquema de un banco de ensayo genérico.

Se ha de realizar la implementación de un sistema que se encargue de dos misiones fundamentales.

La primera misión es la de vigilar que los nodos del bus CAN están emitiendo tramas de latido (*heartbeat*), asegurando así que los nodos están en funcionamiento y en condiciones de comunicarse con el bus.

El segundo cometido consiste en la determinación de la concesión del permiso de entrada a las personas a la sala de ensayo con seguridad. Para evitar el acceso a la sala en condiciones de peligro, el sistema ha de comprobar las señales correspondientes a los transductores asociados a los elementos sometidos a especiales normas de seguridad, tales como velocidades, pares de torsión o fuerzas.

Para conectar el sistema al bus, ha de tenerse en cuenta la imposibilidad de interferencia con los datos que circulan por el bus CAN, esto es, es necesario que el Sistema de Control, objetivo de este Proyecto Fin de Carrera, sea transparente para el control propio del banco, creado con anterioridad.

Las maniobras tales como apertura de puerta o activación de parada de emergencia, así como las lecturas de estados de los actuadores (anclaje de puertas, petición de acceso a la sala) se deben realizar a través del módulo de seguridad PILZ PNOZ Multi M1P presente en el armario de control del exterior de la sala de ensayo.

El Sistema de Control se conectará al módulo de seguridad PILZ a través de sus puertos de entrada/salida.

El Sistema de Control tendrá como entradas y salidas las siguientes señales:

ENTRADAS		SALIDAS	
DESDE MÓDULO PILZ	DESDE VARIADOR	HACIA MÓDULO PILZ	
Anclaje de la puerta de acceso	Activación de tráfico CAN	Apertura de puerta	
Petición de entrada a la sala de ensayo		Parada de emergencia	

Tabla 3.1: Señales de entrada y salida del Sistema de Control.

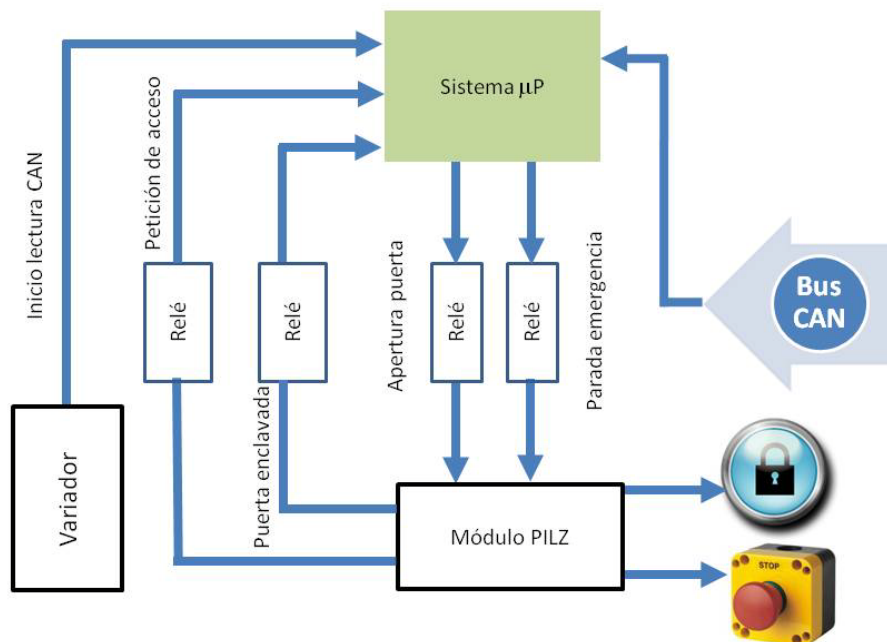


Figura 3.13: Diagrama de bloques del sistema.

Debido a las pretensiones de empleo futuro de la plataforma hardware, es conveniente que el microcontrolador disponga de otras posibilidades, tales como módulo ADC, comparador analógico o incluso generador de ondas PWM, si bien estas características quedan fuera del alcance de este Proyecto.

El banco de pruebas elegido para realizar este Proyecto es el llamado *Heavy Gear Boxes Test Bench 5*, HGBTB5.

Las condiciones del ensayo programado en el banco de ensayos HGBTB5 para los meses siguientes a la realización de este Proyecto hacen que las variables que ha de controlar el Sistema de Control sean las siguientes:

- Velocidad del motor M1
- Par asociado al motor M1
- Fuerza de la célula de carga

Los elementos anteriormente citados son los considerados de riesgo, tanto para el propio ensayo como para las personas.

3.2 Programación del Sistema

Las funciones que realiza el firmware implementado se explican a continuación, basándose en el organigrama representado a continuación..

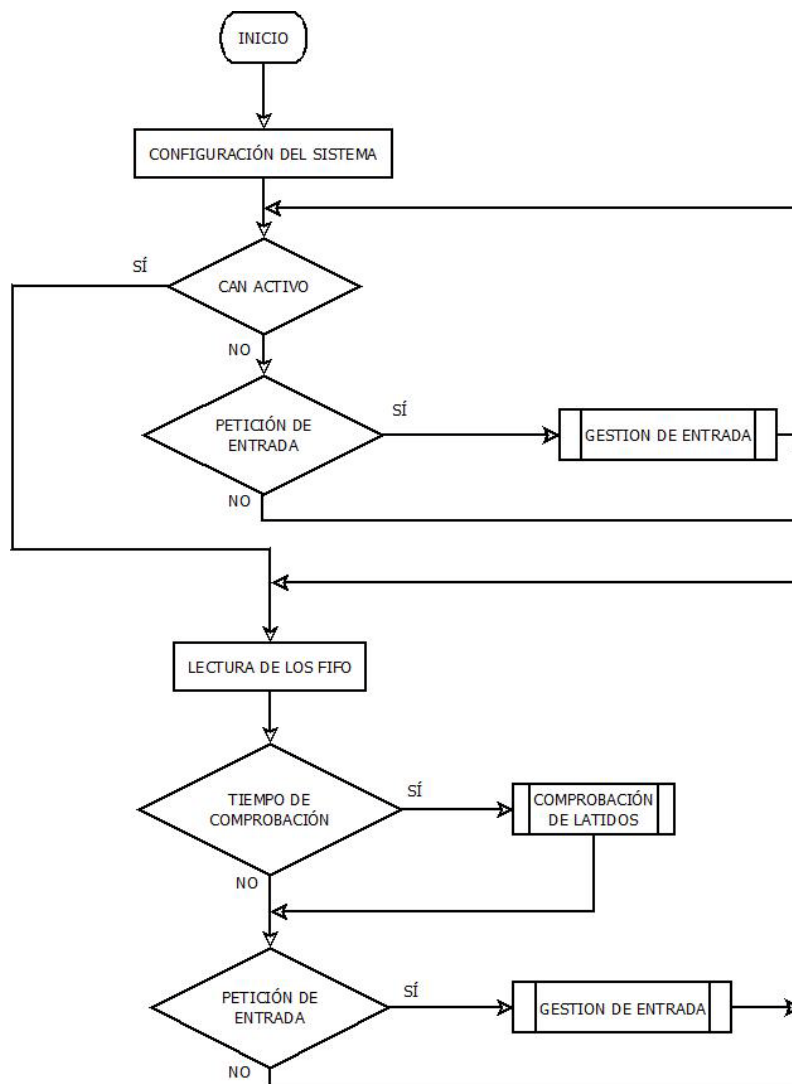


Figura 3.14: Organigrama general.

El organigrama de la figura anterior representa el esqueleto del funcionamiento del programa cargado en el PIC32 en el que se encuentran los bloques que se describen a continuación.

3.2.1 Configuración del sistema

En el bloque de configuración se definen e inicializan las variables y estructuras empleadas por el código. Además se configuran los periféricos de la placa CEREBOT

MX7cK, tales como reloj, alimentación, interrupciones, módulo de CAN y *pinout* del microcontrolador [19].

3.2.1.1 Configuración CAN

Los siguientes apartados refieren las diversas secciones de la configuración del módulo de CAN del PIC32.

3.2.1.1.1 Configuración de la Velocidad

De los datos tabulados para una velocidad del bus CAN de 500 kbps (Figura 3), se determina un número de 10 cuantos de tiempo por bit [37].

T1	T2	BTQ	SP%
6	4	10	60
6	4	10	60
6	4	10	60
6	4	10	60
7	3	10	70
7	3	10	70
7	3	10	70
7	3	10	70
8	2	10	80
8	2	10	80
8	2	10	80
8	2	10	80

Tabla 3.2: Time quanta para velocidades de 500 kbps.

Con estos datos, el fabricante Microchip facilita el proceso de determinación de los valores de configuración.

$$F_{TQ} = N \cdot F_{BAUD}$$

donde

F_{TQ} : Frecuencia de los time quanta de un bit de CAN.

N : Número de time quanta en un bit.

F_{BAUD} : Velocidad del bus.

En este caso concreto, dado que el bus CAN funciona a 500 kbps y se establecen 10 quanta de tiempo por bit, se obtiene un resultado para la frecuencia de los quantos de tiempo de

$$F_{TQ} = 5 \cdot 10^6 \text{ baudios}$$

Para la obtención del valor del preescalado [5], Microchip [20] proporciona la ecuación siguiente:

$$CiCFG \langle BRP \rangle = \frac{F_{SYS}}{2F_{TQ}} - 1$$

donde

F_{SYS} : Frecuencia del reloj de trabajo del microcontrolador PIC32.

F_{TQ} : Frecuencia de los time quanta de un bit de CAN.

Teniendo en cuenta que el PIC32 está trabajando con un reloj de 80 MHz y que el valor anteriormente hallado de F_{TQ} es de $5 \cdot 10^6$ baudios, se obtiene un valor para el preescalado de

$$CiCFG \langle BRP \rangle = 7$$

3.2.1.1.2 Configuración de las Estructuras FIFO

El módulo de CAN del PIC32 no destina una zona específica de memoria para el almacenamiento de las tramas accesible para el programador, ya sea de transmisión o de recepción. Se emplean zonas de RAM del microcontrolador que deben ser configuradas. Así pues, la memoria de CAN se organiza obligatoriamente en estructuras FIFO, de las que es necesario configurar su tipo (lectura o escritura), número y profundidad [15] [20].

Hay disponibles 32 estructuras FIFO con una profundidad máxima de 32 mensajes de CAN y mínima de un mensaje.

La siguiente figura ilustra la organización.

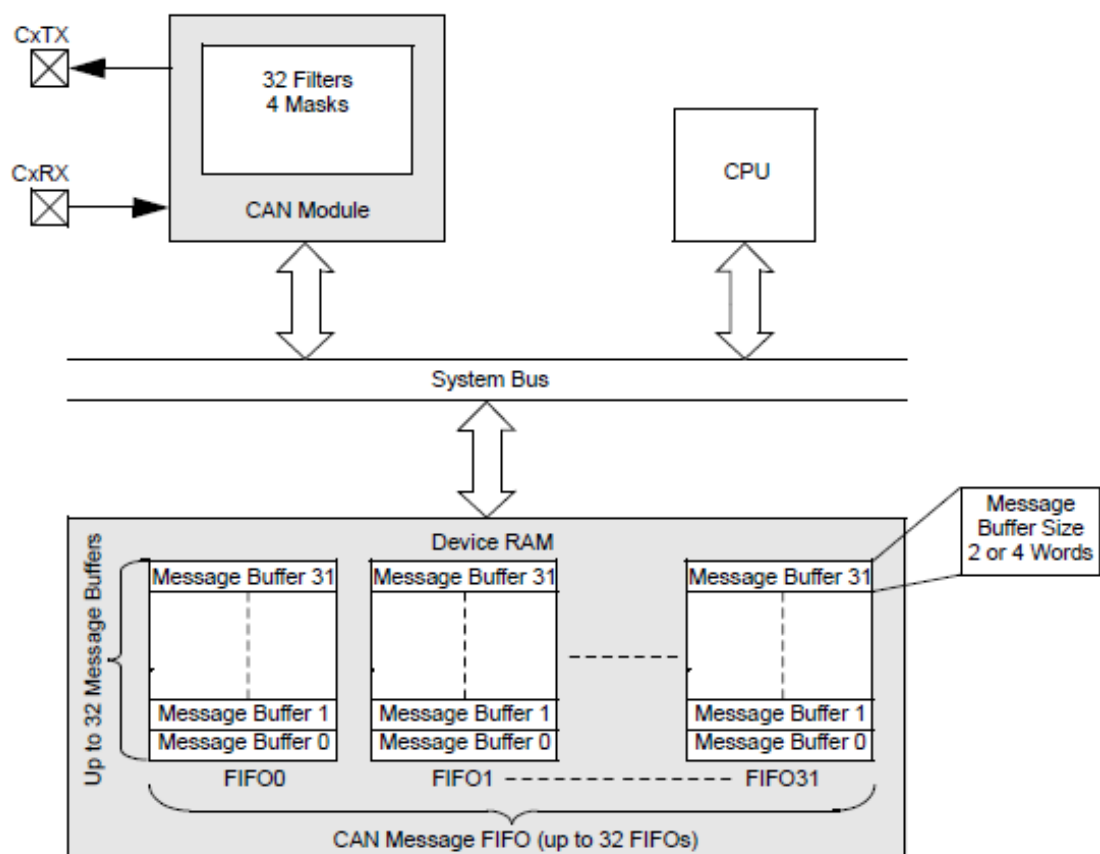


Figura 3.15: Estructuras FIFO en el tratamiento de los datos de CAN.

Cada FIFO lleva asociados cuatro registros del PIC32.

CiFIFOCON_n: Registro de control.

CiFIFOINT_n: Registro de interrupciones asociadas a los FIFO.

CiFIFOUA_n: Registro de direcciones para el usuario.

CiFIFOCI_n: Registro de índice de mensajes.

El módulo CAN sólo necesita la dirección del primer mensaje en el FIFO. El registro CiFIFOB_A (*Base Address*) apuntará a la dirección de este buffer de mensajes. El módulo CAN calcula automáticamente las direcciones de los mensajes, basándose en la configuración de cada FIFO, los cuales ocuparán posiciones consecutivas de RAM, asegurando que no queden huecos.

El registro CiFIFOB_A proporciona la dirección del mensaje de CAN que la aplicación va a usar (tanto para transmisión como para recepción), y el registro CiFIFOCI_n suministra el correspondiente índice al siguiente buffer de mensaje.

Para almacenar los mensajes de CAN, los requerimientos son:

Transmisión de mensaje	4 words (16 bytes)
Recepción de mensaje	4 words (16 bytes)
Recepción de sólo datos del mensaje	2 words (8 bytes)

Tabla 3.3: Requerimientos de memoria de los mensajes de CAN.

Dado que el PIC32 es un microcontrolador de 32 bits, cada palabra (*word*) ocupa cuatro bytes.

La configuración de los FIFO se realiza en cinco pasos:

- 1.- Reserva de memoria.
- 2.- Puesta del módulo CAN en modo configuración (OPMOD<2:0> = 0x100).
- 3.- Escritura del registro de direcciones de base CiFIFOBA con la dirección física del mensaje primero del FIFO.
- 4.- Escritura del tamaño del FIFO, CiFIFOCONn<FSIZE> (bits 20:16).
- 5.- Selección del tipo de FIFO: transmisión o recepción. El bit TXEN del registro CiFIFOCONn será un "1" para un FIFO de transmisión o un "0" para un FIFO de recepción, que es el caso que nos ocupa.

Realizados estos pasos, se puede establecer el modo de operación del módulo de CAN.

La configuración de los FIFO se realiza mediante la función

```
void ConfigurarFIFO(void)
```

3.2.1.1.3 Configuración de Filtros y Máscaras

Para realizar este bloque de configuración se emplea la función

```
void ConfigurarFiltroMascara(void)
```

En esta función se configuran los filtros y máscaras que se emplean para recibir las tramas deseadas en las estructuras FIFO. En este caso, se han definido tres filtros y tres máscaras que delimitan la aceptación de los mensajes CAN y la ajustan al máximo posible para evitar un trasiego innecesario de datos del módulo de CAN a la memoria del microcontrolador.

Para que un mensaje sea de interés para cualquier módulo de CAN, ha de pasar por un proceso de filtrado y enmascarado, tal como muestra la siguiente figura.

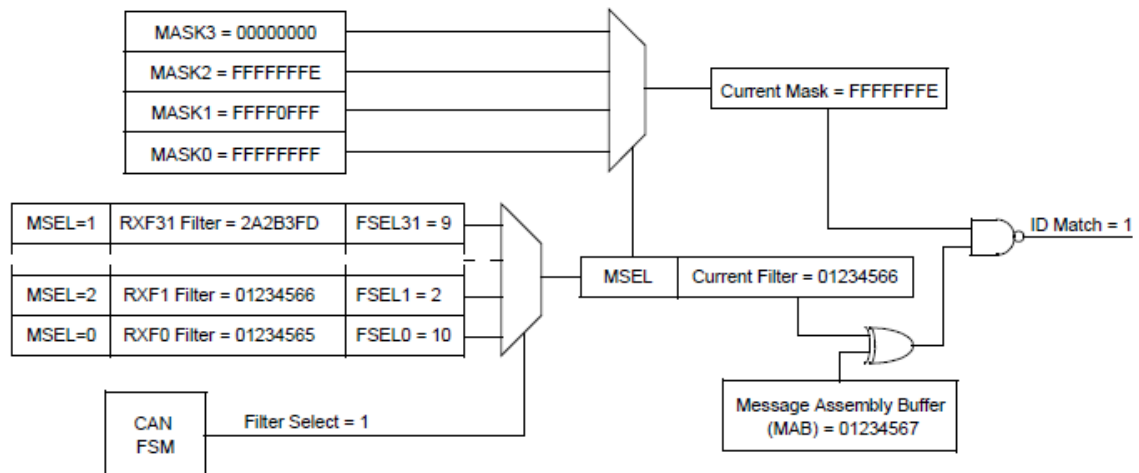


Figura 3.16: Filtrado y enmascaramiento de un mensaje de CAN.

La trama de CAN obtenida por el módulo a través del transceptor será enfrentada a los filtros definidos. Si pasa alguno de ellos, el resultado será enfrentado a su vez a la máscara asociada al filtro en el registro de configuración CiFIFOCON_n. Si la coincidencia ocurre, el módulo de CAN pasará el contenido del MAB (*Message Assembly Buffer*) al FIFO definido por los bits FSEL_{n<4:0>} del registro CiFLTCON_n.

El proceso de filtrado consiste en una operación de XOR (ó exclusiva) entre el mensaje y el filtro. El resultado de esta operación pasa a la operación de enmascaramiento, consistente en una operación lógica NAND. Si el resultado contiene algún cero, no se ha superado el proceso y al mensaje no será trasladado a la memoria. En caso contrario, es decir, si todos los bits del resultado del enmascaramiento son "1", el mensaje es aceptado y el módulo CAN lo traslada al FIFO correspondiente [20].

La configuración de los filtros se realiza en los registros CiFLTCON_n. Dentro de dicho registro se encuentran los siguientes bits de configuración:

FSEL0	Indica el FIFO de almacenamiento para los mensajes que superen el filtro 0
MSEL0	Selecciona la máscara 0 para el filtro 0
FSEL1	Indica el FIFO de almacenamiento para los mensajes que superen el filtro 1
MSEL1	Selecciona la máscara 1 para el filtro 1
FSEL2	Indica el FIFO de almacenaje para los mensajes que superen el filtro 2
MSEL2	Selecciona la máscara 2 para el filtro 2

Tabla 3.4: Bits de configuración del registro CiFLTCON_n.

Las operaciones de filtro y enmascaramiento pueden ser realizadas sólo con el campo SID o con el total del mensaje CAN. En este Proyecto se ha tenido en cuenta sólo el campo SID. Esta configuración se realiza escribiendo un "0" en el bit EXID del registro CiRXFn.

Concluida la configuración de las estructuras de memoria y los filtros y máscaras, ya puede establecerse el modo de operación del módulo CAN. Como se refirió con anterioridad, se pretende que el sistema no interfiera en el bus CAN del banco de ensayos. Para garantizar esta premisa, el modo de operación del módulo de CAN será *listen only mode*. Este modo de operación es una variante del modo de funcionamiento normal (*Normal Operation Mode*). Si está activado el modo *listen only mode*, el módulo CAN estará presente en el bus, pero de manera pasiva. Podrán recibirse mensajes, pero no transmitirlos. No se generarán los reconocimientos (ACK). Es el modo de funcionamiento indicado como monitor del bus, pues nunca influye en los datos.

La configuración como *listen only mode* se realiza escribiendo 0x011 en los bits REQOP del registro CiCON.

En el archivo cabeceras\parametros.h se define el modo de funcionamiento del módulo CAN:

```
#define MODO_CAN      7 // Listen ALL MESSAGES
```

Las siguientes líneas de código pertenecientes al fichero fuentes\FuncionesCAN.c realizan la configuración:

```
CiCONbits.REQOP = MODO_CAN;  
  
while(C2CONbits.OPMOD != MODO_CAN);
```

3.2.1.1.4 Estructuras de Datos

Es de resaltar el especial tratamiento de los datos manejados por el módulo CAN del PIC32, ya que tanto la recepción como la transmisión al transceptor se realiza mediante unas estructuras de datos fijas [9] [20].

Los mensajes aceptados por el módulo CAN son almacenados en el buffer de memoria empleando el formato mostrado en la Figura 3.5.

Address Offset	Name	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0	
0x00	MSGSID	31:24	MSGTS<15:8>							
		23:16	MSGTS<7:0>							
		15:8	FILHIT<4:0>				SID<10:8>			
		7:0	SID<7:0>							
0x04	MSGEID	31:24	---	---	SRR	IDE	EID<17:14>			
		23:16	EID<13:6>							
		15:8	EID<5:0>					RTR	RB1	
		7:0	---	---	---	RB0	DLC<3:0>			
0x08	MSGDATA0	31:24	Receive Buffer Data Byte 3							
		23:16	Receive Buffer Data Byte 2							
		15:8	Receive Buffer Data Byte 1							
		7:0	Receive Buffer Data Byte 0							
0x0C	MSGDATA1	31:24	Receive Buffer Data Byte 7							
		23:16	Receive Buffer Data Byte 6							
		15:8	Receive Buffer Data Byte 5							
		7:0	Receive Buffer Data Byte 4							

Figura 3.17: Formato de los datos almacenados en RAM.

El módulo de CAN utiliza los formatos que ilustran las siguientes figuras. Así, la Figura 3.6 muestra la estructura que contiene el SID.

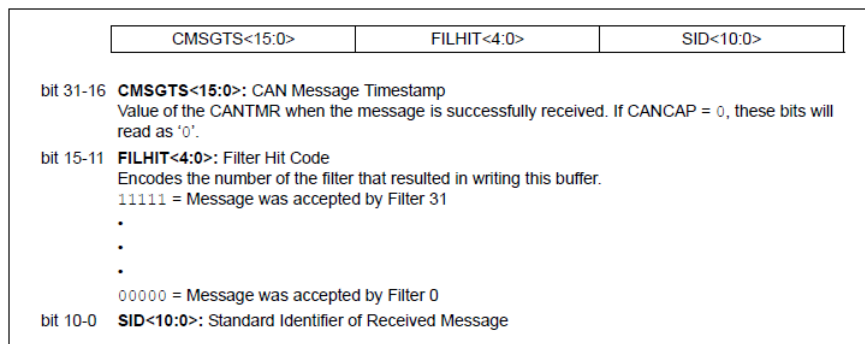


Figura 3.18: Formato MSGSID.

Las dos figuras siguientes ilustran los campos que contienen los bits de datos de la trama.

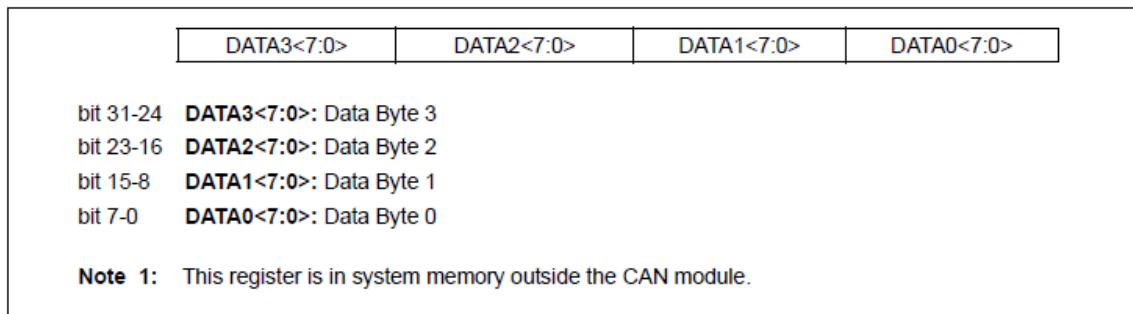


Figura 3.19: Formato CMSGDATA0.

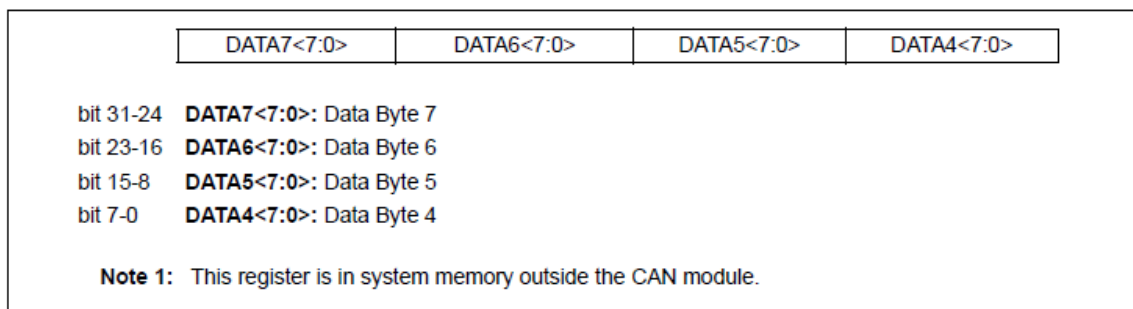


Figura 3.20: Formato CMSGDATA1.

La aplicación crea una estructura C llamada `msgRXCAN` en la que se definen los campos de las figuras anteriores y que servirá para rellenarse con los datos de las tramas de CAN aceptadas y almacenadas por tanto en las estructuras FIFO. A continuación se muestra la codificación en C.

```

typedef struct
{
    unsigned SID:11;                // Once bits para Standard Identifier.
    unsigned FILHIT:5 ;            // Filtro que hace capturar la trama.
    unsigned CMSGTS:16;           // Timestamp.
}rxcmsgsid;

/* Creación del tipo de datos CMSGEID */
typedef struct
{
    unsigned DLC:4;                // Longitud de los datos.
    unsigned RB0:1;                // Bit reservado, recesivo.
    unsigned :3;                   // No se usan. Mantiene formato del registro HW.
    unsigned RB1:1;                // Bit reservado, recesivo.
    unsigned RTR:1;                // Trama remota, RTR=1. Trama datos, RTR=0.
    unsigned EID:18;               // Extended Identifier bits.
    unsigned IDE:1;                // Estandar, IDE=0, Extendida, IDE=1
    unsigned SRR:1;                // SRR=1, trama remota extendida.
    unsigned :2;                   // No se usan. Mantiene formato del registro HW.
}rxcmsgeid;

/* Creación del tipo de datos CMSGDATA0 */
typedef struct
{
    unsigned Byte0:8; ;            // Datos.
    unsigned Byte1:8;
    unsigned Byte2:8;
    unsigned Byte3:8;
}rxcmsgdata0;

/* Creación del tipo de datos CMSGDATA1 */
typedef struct
{
    unsigned Byte4:8; ;            // Datos.
    unsigned Byte5:8;
    unsigned Byte6:8;
    unsigned Byte7:8;
}rxcmsgdata1;

/* Principal estructura de datos */
typedef union unionMsgRXCAN
{
    struct
    {
        rxcmsgsid CMSGSID;
        rxcmsgeid CMSGEID;
        rxcmsgdata0 CMSGDATA0;
        rxcmsgdata1 CMSGDATA1;
    }
    ;int messageWord[4];
}msgRXCAN;

```

3.2.1.2 Configuración del Reloj

Se emplea el reloj interno del microcontrolador, y se especifica un rendimiento de 80 MHz. La configuración del reloj se realiza en el módulo principal del código fuentes\pfc.c, mediante el uso de directivas *pragma* y de macros pertenecientes a la librería p32xxxx.h [].

```
#pragma config PLLMUL=MUL_20,FPLLIDIV=DIV_2,FPLLODIV=DIV_1,FWDTEN=OFF
#define SYS_FREQ (80000000L)
SYSTEMConfig(SYS_FREQ, SYS_CFG_WAIT_STATES | SYS_CFG_PCACHE);
```

3.2.1.3 Configuración de *timer1*

La carga del *timer1*, así como la definición de la rutina de atención a la interrupción están codificadas en fuentes\funcionesTimer1.c.

```
void InicializarTimer1(void)
{
    T1CON = 0x0;           // Basic Timer Operation
    PR1 = TIMER1_PERIOD;  // Periodo definido en FuncionesTimer.h
    IFS0CLR = 0x10;       // Borrar flag de interrupción.
    IEC0SET = 0x10;       // Timer 1: interrupciones habilitadas.
    IPC1bits.T1IP = 4;    // Timer 1: prioridad 4
    unSegundoVencido = FALSE; // Inicializacion del aviso de un segundo.
    contadorMilisegundos = 0; // Inicialización del contador de milisegundos.
    T1CONSET = 0x8000;    // Puesta en marcha del timer.
}
```


La rutina de atención a la interrupción se dispara cada milisegundo, al alcanzarse la cuenta de 0x1f40 [18]. Esta es su codificación:

```
void __attribute__((vector(4), interrupt(ipl4), nomips16)) Timer1InterruptHandler(void)
{
    IFS0CLR = 0x10;           // Borrar el flag de la interrupción.
    contadorMilisegundos++;   // Incrementar el contador de milisegundos.
    if(contadorMilisegundos == 0x2710)
    {
        unSegundoVencido = TRUE; // Ha pasado un segundo.
        contadorMilisegundos = 0; // Reset del contador de milisegundos.
    }
}
```

3.2.2 Lectura de los FIFO

El Sistema de Control comienza la lectura del bus CAN cuando se recibe la señal correspondiente a través del relé KM11A, en serie con el contactor del motor M1.

El relé KM11A se activa cuando, desde el armario de control, se pulsa el botón “Rearme” y, posteriormente, desde la aplicación del PC se comanda el “Control Manual”.

Se aprovecha una de las salidas del relé (conectada a 5V) para activar la señal que lee el bit 0 del puerto D (RD0) del microcontrolador, a través del pin 2 del conector *pmod* JD de la tarjeta CEREBOT MX7cK.

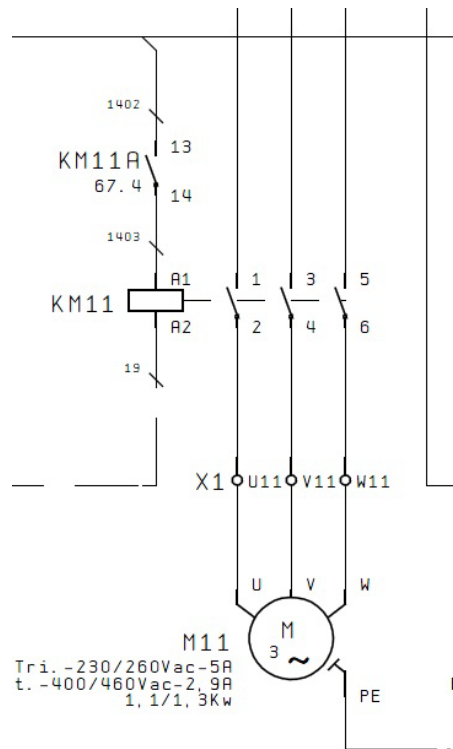


Figura 3.21: Relé KM11A y contactor KM11 del motor M1.

Se puede acceder al mensaje de CAN por dos vías distintas: mediante interrupciones o por *polling* de los flags correspondientes. La aplicación emplea el método de espera activa (*polling*) sobre el flag RXNEMPTYIF del registro CiFIFOINTn. Dicho flag se activa cuando el FIFO que ha almacenado un mensaje no está vacío [20].

Se define la variable mensajeCAN como un puntero a la estructura msgRXCAN.

Al detectar la activación del flag RXNEMPTYIF, se obtiene un puntero al buffer correspondiente:

$$\text{mensajeCAN} = (\text{msgRXCAN} *) (\text{PA_TO_KVA1}(\text{CiFIFOUn}))$$

A partir de este momento, mensajeCAN contiene la información del mensaje de CAN. Definidas las variables campoSID, campoDLC, campoDATA00, campoDATA01, campoDATA02, campoDATA03, campoDATA14, campoDATA15, campoDATA16 y campoDATA17, la trama de CAN es accesible mediante las sentencias:

```
campoSID = mensajeCAN -> CMSGSID.SID;
campoDLC = mensajeCAN -> CMSGEID.DLC;
campoDATA00 = mensajeCAN -> CMSGDATA0.Byte0;
campoDATA01 = mensajeCAN -> CMSGDATA0.Byte1;
campoDATA02 = mensajeCAN -> CMSGDATA0.Byte2;
campoDATA03 = mensajeCAN -> CMSGDATA0.Byte3;
campoDATA14 = mensajeCAN -> CMSGDATA1.Byte4;
campoDATA15 = mensajeCAN -> CMSGDATA1.Byte5;
campoDATA16 = mensajeCAN -> CMSGDATA1.Byte6;
campoDATA17 = mensajeCAN -> CMSGDATA1.Byte7;
```

Cada uno de los FIFO implicados en el almacenamiento de mensajes tendrá su correspondiente puntero al mensaje, de modo que se hará *polling* a los flags RXNEMPTYIF.del siguiente modo:

```
if(C2FIFOINT0bits.RXNEMPTYIF == 1) // Polling al flag: FIFO no vacío.
{
    WDTCON = WDT_CLR; // Notificación para watchdog

    // Obtener la dirección del buffer
    mensajeCAN = (msgRXCAN *)(PA_TO_KVA1(C2FIFOUA0));

    // Carga de los campos.
    campoSID = mensajeCAN -> CMSGSID.SID;
    campoDLC = mensajeCAN -> CMSGEID.DLC;
    campoDATA00 = mensajeCAN -> CMSGDATA0.Byte0;
    campoDATA01 = mensajeCAN -> CMSGDATA0.Byte1;
```

```

campoDATA02 = mensajeCAN -> MSGDATA0.Byte2;
campoDATA03 = mensajeCAN -> MSGDATA0.Byte3;
campoDATA14 = mensajeCAN -> MSGDATA1.Byte4;
campoDATA15 = mensajeCAN -> MSGDATA1.Byte5;
campoDATA16 = mensajeCAN -> MSGDATA1.Byte6;
campoDATA17 = mensajeCAN -> MSGDATA1.Byte7;

campoID    = campoSID & 0x0780; // Extracción del "comando" FctCode.
IDnodo     = campoSID ^ campoID; // Extracción del ID del nodo.

C2FIFOCON0SET = 0x2000; // Actualización del puntero al mensaje.
}

```

En las tramas estándar de CAN, que son las que nos ocupan, el campo SID, (*cobId*), ocupa once bits. Los cuatro primeros conforman el *FctCode* [6] [7] [8]:

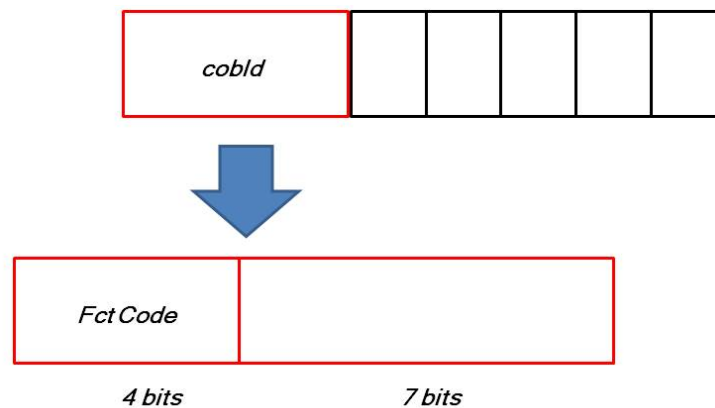


Figura 3.22: Extracción del *FctCode*.

Para hallar el *FctCode*, se realiza una operación AND entre el *cobId* y la máscara 0x780, extrayendo así los cuatro bits de mayor peso, esto es,

$$\text{FctCode} = 0x780 \& \text{cobId}$$

En la lectura de tramas *heartbeat*, no hay datos implicados, sólo se necesita conocer el ID del nodo que emitió el latido. Así pues, para conocer el ID del nodo, basta someter el campo SID a una operación lógica XOR con el comando de latido 0x700, resultando así el número de nodo.

```
campoSID = mensajeCAN -> MSGSID.SID;
```

```
campoID = campoSID & 0x0780;
```

```
IDnodo = campoSID ^ campoID;
```

El firmware escucha tanto los latidos como los valores de los elementos críticos para la seguridad para las personas.

Cada vez que se recibe del bus CAN el valor de la medida de un valor crítico, se almacena en su vector correspondiente.

```
int lecturaVelocidadMotor1[MAX_MEDIDAS];
```

```
int lecturaParMotor1[MAX_MEDIDAS];
```

```
int lecturaCelCarga[MAX_MEDIDAS];
```

La carga de los valores leídos del bus CAN se realiza como sigue:

```
lecturaVelocidadMotor1[indiceMotor1] = (campoDATA01<<8 | campoDATA00);
```

...

```
lecturaParMotor1[indiceMotor1] = (campoDATA03<<8 | campoDATA02);
```

...

```
lecturaCelCarga[indiceCelCarga] = (campoDATA01<<8 | campoDATA00);
```

Se observa en las anteriores líneas de código cómo se invierte el orden de los bytes del campo de datos de la trama CAN.

Para posteriores consultas relativas al acceso a la sala, cada vez que se cumple un número de medidas de los valores críticos, se calcula la media de los mismos.

```
mediaVelocidadMotor1 = Media(lecturaVelocidadMotor1,MAX_MEDIDAS);
```

```
mediaParMotor1 = Media(lecturaParMotor1,MAX_MEDIDAS);
```

```
mediaCelCarga = Media(lecturaCelCarga,MAX_MEDIDAS);
```

`MAX_MEDIDAS` es un parámetro definido en `cabeceras\parametros.h` que hace referencia al número de medidas que hay que tomar para calcular la media.

3.2.3 Comprobación de Latidos

Los nodos del bus CAN están configurados para emitir un *heartbeat* (latido, cobld 0x700) con una frecuencia determinada por el maestro del bus CAN en la configuración inicial.

Cada vez que se recibe una trama, se comprueba si representa un latido. Identificado éste, se extrae el nodo que la emitió y se actualiza un contador de latidos asociados al nodo. El firmware contabiliza los latidos correspondientes a los elementos sometidos a condiciones de seguridad del ensayo.

Cuando el *timer1* ha contado un tiempo igual a un segundo, se realiza la comprobación de latidos de cada nodo.

Si el número de latidos por segundo es el esperado, los contadores se resetean y comienza una nueva cuenta. En caso contrario, se activa la parada de emergencia a través de RD4. El número de latidos por segundo está definido en el fichero `cabeceras\parametros.h`.

A continuación se escriben las líneas de código pertenecientes al archivo fuentes\pfc.c que expresan lo mencionado en el anterior párrafo.

```
if(HaPasadoUnSegundo() == TRUE)

{

    WDTCON = WDT_CLR;                // Notificación para watchdog

    PORTToggleBits(IOPORT_G,BIT_15); // Cambiar LED: comprobación visual

    if ((contadorLatidosMotor1 < LATIDOS_POR_SEGUNDO) || // M1 tiene veloc y par.

        (contadorLatidosCelCarga < LATIDOS_POR_SEGUNDO)) // Cel.Carga: fuerza.

        paradaDeEmergencia = 1;      // Elementos críticos sin latidos necesarios.

    else

        ResetearContadores();        // Nueva cuenta, contadores a cero.
```

Cuando se activa la parada de emergencia, el banco de ensayo está detenido y el bus CAN está inactivo. El programa entra en el bloque correspondiente, es decir, está a la espera de una petición de entrada o del rearme del banco y su consecuente puesta en marcha.

3.2.4 Gestión de la Entrada

Para solicitar el paso a la sala de ensayo, el usuario debe accionar el selector “ABRIR PUERTA” del armario eléctrico de control mostrado en la siguiente figura.



Figura 3.23: Interruptor de petición de entrada (“ABRIR PUERTA”) en armario de control.

Cuando una persona intenta acceder a la sala, pueden darse dos situaciones distintas.

Si el banco está parado, con el variador inactivo y, por tanto, el bus CAN parado, todos los actuadores están en reposo, de modo que no existe riesgo físico para las personas. En esta situación, el firmware del PIC32 concede siempre el acceso a la sala mediante la señalización correspondiente al módulo de seguridad PILZ.

Si el banco está en funcionamiento, esto es, con el variador en marcha y, consecuentemente, el bus CAN activo, pueden ocurrir dos situaciones en función de la seguridad para las personas. La condición de seguridad se determina por el cálculo cíclico del valor de los parámetros críticos de seguridad, a saber, velocidad del motor, par torsor del motor y fuerza en la célula de carga. Con los valores que proporcionan al bus CAN los transductores de los actuadores críticos, se calcula una media aritmética empleando la función

```
int Media(int vector[],char numero)
```


Las medias de los valores críticos son comparadas con los valores de seguridad definidos en el fichero `cabeceras\parametros.h`:

```
// Valores límite
#define VMAX_MOTOR1 500
#define PARMAX_MOTOR1 200
#define FMAX_CELCARGA 5000
```

Así pues, cuando el banco está en marcha, sólo se concederá el acceso en condiciones seguras para las personas. En caso contrario el firmware obviará la petición.

Puede ocurrir que, una vez concedido el acceso a la sala, los valores críticos varíen hasta la transgresión de los límites de seguridad. En este caso, el firmware del Sistema de Control activa la parada de emergencia, deteniendo por tanto el ensayo.

Las entradas serán leídas por *polling* del puerto D.

Para llevar a cabo las labores de este bloque de código se emplean las entradas y salidas de la siguiente tabla.

ENTRADAS			SALIDAS		
Referencia	PIC32	CEREBOT	Referencia	PIC32	CEREBOT
CAN ACTIVO	RD0	JD02	APERTURA PUERTA	RD3	JD09
PUERTA ENCLAVADA	RD1	JD07	PARADA DE EMERGENCIA	RD4	JC09
PETICIÓN DE ENTRADA	RD2	JD08			

Tabla 3.5: Entradas y salidas empleadas en la gestión del acceso a la sala.

El funcionamiento del módulo de la gestión de entrada se realiza por llamadas a la función siguiente:

BOOL GestionarEntrada(BOOL estadoBusCAN)

Nótese que el valor de retorno de la función es de tipo BOOL. Los valores que pueden devolverse son SEGURO ("0") o EMERGENCIA ("1"). Estos valores son empleados en el código para tomar las acciones pertinentes.

El funcionamiento del código queda resumido gráficamente en el organigrama de la figura siguiente.

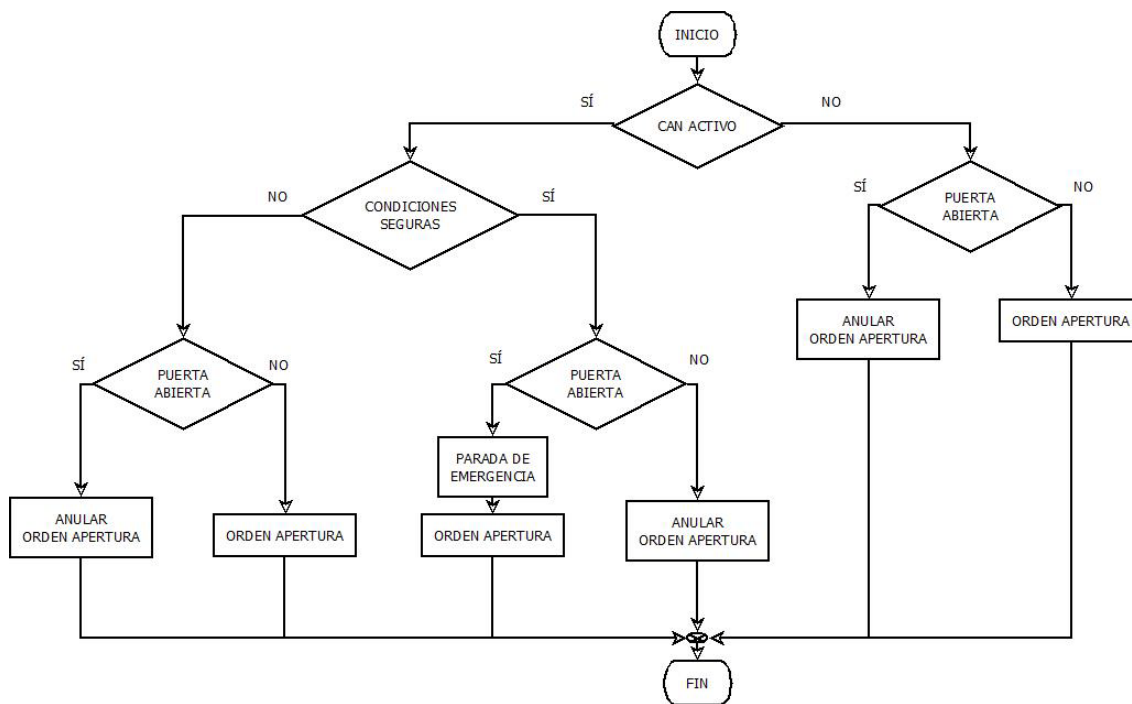


Figura 3.24: Organigrama de gestión de entrada al banco de ensayo.

3.2.5 Watchdog

El *watchdog* es un mecanismo de seguridad que provoca un reset del microcontrolador cuando se ha bloqueado. Consiste en un temporizador cuyo contador se decrementa hasta alcanzar el cero o hasta que una rutina o instrucción reinicie el temporizador.

Cuando expira el periodo de time-out del watchdog, automáticamente se genera un reset del microcontrolador. La ejecución del código salta al vector de reset del dispositivo y los valores de reset serán escritos en los registros y periféricos [14].

En el supuesto caso de la ocurrencia de un time-out del periodo del watchdog, el Sistema de Control inicia el flujo normal del programa, esto es, realiza su configuración y tras el periodo de estabilización de la alimentación y los relojes, espera la activación del 0 del puerto D para comenzar con la lectura del bus CAN y de los eventos asociados a la sala de ensayo para actuar en consecuencia.

Por defecto, el watchdog del microcontrolador está activado. Es necesario el refresco del bit WDTCLR para evitar el salto al vector de reset. WDTCLR es el bit 0 del registro WDTCON, tal como aparecen en la tabla siguiente [16].

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0 —	U-0 —	U-0 —	U-0 —	U-0 —	U-0 —	U-0 —	U-0 —
23:16	U-0 —	U-0 —	U-0 —	U-0 —	U-0 —	U-0 —	U-0 —	U-0 —
15:8	R/W-0 ON(1,2)	U-0 —	U-0 —	U-0 —	U-0 —	U-0 —	U-0 —	U-0 —
7:0	U-0 —	R-y	R-y	R-y	R-y	R-y	R/W-0	R/W-0 WDTWEN WDTCLR

Tabla 3.6: Registro WDTCON.

El refresco se realiza mediante la escritura a “1” del bit WDTCLR, esto es, la instrucción

$$\text{WDTCON} = \text{WDT_CLR}$$

u otra similar hace que se resetee el timer del watchdog y comience una nueva cuenta.

La fuente del reloj del watchdog es el oscilador interno LPRC (*Low Power RC*) que tiene una frecuencia nominal de 32 kHz. Esto crea un periodo nominal de 1 ms cuando no se ha seleccionado un post-escalado.

El módulo de watchdog dispone de cinco bits para seleccionar el post-escalador, lo que permite generar una gran variedad de periodos de time-out. El post-escalador provee unos ratios de división desde 1:1 hasta 1:1048576, es decir, es posible la obtención de periodos desde 1 ms hasta 1048,576 s.

La selección del post-escalador se realiza en los bits de configuración del microcontrolador, según la tabla y la figura siguientes.

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	r-1 —	r-1 —	r-1 —	r-1 —	r-1 —	r-1 —	r-1 —	r-1 —
23:16	R/P FWDTEN	r-1 —	r-1 —	R/P —	R/P —	R/P —	R/P —	R/P —
15:8	R/P FCKSM<1:0>	R/P —	R/P FPBDIV<1:0>	R/P —	r-1 —	R/P OSCIOFNC	R/P POSCMOD<1:0>	R/P —
7:0	R/P IESO	r-1 —	R/P FSOSCEN	r-1 —	r-1 —	R/P —	R/P —	R/P FNOSC<2:0>

Tabla 3.7: Registro de configuración DEVCFG1.

Address	Value	Field	Category
1FC0_2FF0	FFFFFFF	USERID	
1FC0_2FF4	FFFFFFF	FSRSSEL	SRS Select
		FMIEN	Ethernet RMII/MII Enable
		FETHIO	Ethernet I/O Pin Select
		FCANIO	CAN I/O Pin Select
		FUSBIDIO	USB USID Selection
		FVBUSONIO	USB VBUS ON Selection
		FPLLIDIV	PLL Input Divider
		FPLLMUL	PLL Multiplier
		FUPLLDIV	USB PLL Input Divider
		FUPLLEN	USB PLL Enable
1FC0_2FF8	FFFFFFF	FPLLIDIV	System PLL Output Clock Divider
		FNOSC	Oscillator Selection Bits
		FSOSCEN	Secondary Oscillator Enable
		IESO	Internal/External Switch Over
		POSCMOD	Primary Oscillator Configuration
		OSCIOFNC	CLKO Output Signal Active on the OSCO Pin
		FPBDIV	Peripheral Clock Divisor
		FCKSM	Clock Switching and Monitor Selection
		WDTPS	Watchdog Timer Postscaler
		FWDTEN	Watchdog Timer Enable
1FC0_2FFC	7FFFFFF	DEBUG	Background Debugger Enable
		ICSESEL	ICE/ICD Comm Channel Select
		FWP	Program Flash Write Protect
		BWP	Boot Flash Write Protect bit
		CP	Code Protect

Figura 3.25: Post-escalador del watchdog.

El periodo de time-out del watchdog se calcula como sigue:

$$T_{watchdog} = 1ms \cdot 2^{Post-escalador}$$

Para el firmware del Sistema de Control se ha seleccionado un periodo de time-out del watchdog igual a 0,512s, por lo que el valor del post-escalador es

$$WDTPS<4:0> = 0x01001$$

ya que

$$T_{watchdog} = 1ms \cdot 2^9$$

Capítulo 4

Conclusiones

En este capítulo se refiere la consecución de los objetivos propuestos.

Se ha desarrollado e implementado un firmware que cumple con las expectativas iniciales. El programa implementado hace que el Sistema de Control realice la supervisión de la actividad del bus CAN y proporcione con sus acciones las condiciones de seguridad para que las personas accedan al banco de ensayo. Por tanto, el Sistema de Control implementado toma el control de la seguridad.

La conexión de la placa multipropósito al módulo de seguridad se ha llevado a efecto con un resultado satisfactorio. El Sistema de Control obtiene correctamente la lectura de las señales del módulo de seguridad, y a su vez proporciona adecuadamente las señales de maniobra de apertura de puerta de acceso a la sala de ensayo. De igual modo, el Sistema de Control proporciona correctamente la señal de parada de emergencia cuando las condiciones lo requieren.

Se ha conectado con éxito la placa multipropósito a una fuente de alimentación con anclaje en carril DIN.

Se ha llevado a cabo la conexión de los relés de estado sólido tanto a la placa multipropósito como al módulo de seguridad, con resultado plenamente satisfactorio.

Por retrasos en la petición y entrega de materiales, no se dispone del soporte para el anclaje de la placa multipropósito al carril DIN, de modo que la ubicación de la tarjeta Cerebot no es definitiva.

Se obtiene correctamente la señal de inicio de actividad del bus CAN.

El Sistema de Control muestra un funcionamiento que responde a las expectativas. Las pruebas realizadas confirman la fiabilidad del hardware del CEREBOT MX7cK en su conjunto.

En cuanto al firmware, las tramas que recoge son las esperadas acordes con productos del mercado (CANAlizer). El tratamiento que realiza el Sistema de Control con los datos es correcto, esto es, las medidas son fiables y las acciones ulteriores son realizadas sin error.

El Sistema de Control implementado es un producto inexistente en el mercado. Esa es la principal virtud que presenta.

4.1 Resultados

Gracias a la realización de este Proyecto Fin de Carrera se ha podido conocer más acerca del sistema CAN del banco de ensayo H5. Se ha determinado la velocidad del bus CAN. Se ha confirmado la presencia del protocolo *heartbeat* y se ha podido comprobar su frecuencia. El estudio previo de las tramas del bus CAN ha servido para conocer mejor el modo de envío de los datos en las tramas, dependiendo del tipo de nodo del que se trate.

También ha sido de utilidad el estudio previo de las entradas y salidas que emplea el Sistema de Control, pues de este modo se ha profundizado en el sistema de seguridad del banco de ensayo y en sus módulos. Tal es el caso del módulo de seguridad PILZ.

Es importante resaltar que el Sistema de Control no pierde tramas del bus. Para la determinación del número de tramas presentes en el bus en un determinado tiempo, se conectan en paralelo al bus CAN el Sistema de Control y la herramienta CANAlizer. Ambos sistemas están leyendo del bus hasta que transcurrido aproximadamente un segundo se pulsa manualmente la parada de emergencia. En este punto, el Sistema de Control y el CANAlizer dejan de leer pues el bus queda inactivo. Las tramas

recogidas por el CANAlizer son pasadas a una hoja de cálculo para su filtrado y cuenta. El número de tramas recogidas por el Sistema de Control queda registrado en unos contadores de prueba introducidos en el firmware. Los números son coincidentes.

A la vez que se determina la coincidencia del número de tramas, se realiza el análisis del tipo de tramas recibidas por los sistemas. Gracias al filtrado de la hoja de cálculo (el bus CAN contiene datos que el Sistema de Control desecha por no ser de su interés) puede verificarse de nuevo la coincidencia. Además se comprueba que los valores leídos por el PC de aplicación coinciden con las medias aritméticas calculadas por el firmware del Sistema de Control.

El sistema implementado lee correctamente y a tiempo las tramas del bus CAN. Se ha comprobado que no ocurre *overflow* en las estructuras FIFO. Esta prueba se ha realizado en primera instancia por la activación de los LED (a modo de contador digital visual) de la tarjeta CEREBOT MX7cK en caso de ocurrencia de desbordamiento. El *overflow* no llega a producirse en un periodo de una hora, con el banco de ensayo en funcionamiento normal.

Se realizó una prueba más restrictiva para detectar *overflow*, bajando la profundidad de los FIFO hasta sólo un mensaje de CAN. En el firmware del Sistema de Control se coloca un bucle infinito en caso de ocurrencia de desbordamiento, mostrando un código luminoso conocido. De nuevo se verificó que no se produce *overflow* en las lecturas.

La placa del Sistema de Control es capaz de suministrar la corriente a los relés de estado sólido. Se comprobó que eran activados dos relés simultáneamente y con una frecuencia de 1 kHz. No se observó comportamiento anormal, ni en términos de hardware, como un exceso de calor, ni en términos de software, pues el tiempo perdido en el manejo de los puertos, uno por cada relé, no influyó en la lectura correcta de las tramas de CAN.

Se comprobó la activación de los dos relés en modo continuo, detectando un consumo medio unitario de 6.3 mA. El Sistema de Control es por tanto fiable en el suministro de corriente.

El módulo *watchdog* del PIC32 no puede ser probado en depuración. La prueba se realizó por tanto modificando el firmware del microcontrolador para que no realizara los refrescos correspondientes. Se verificó el correcto funcionamiento del watchdog al comprobar que continuamente salta al vector de reset, de modo que el Sistema de Control es seguro por sí mismo.

Es significativo destacar la resistencia de la placa CEREBOT MX7cK ante cortes y puestas en tensión. El Sistema de Control es robusto, fiable y seguro.

4.2 Trabajos Futuros

La gran variedad de periféricos y módulos de la familia PIC32, y en concreto del PIC32MX795F512L hace que los usos a los que se destine sean numerosos. A su vez la placa CEREBOT MX7cK proporciona el soporte adecuado para el potencial del microcontrolador.

Algunos de los posibles trabajos futuros son enumerados a continuación.

4.2.1 Display

La presencia de puertos libres en el PIC32 hace posible la implementación de un *display* en el sistema [26]. Es recomendable la utilización del *Parallel Master Port* disponible en este modelo de microcontrolador [17] para el manejo de una pequeña pantalla LCD en la que mostrar el status del sistema, los mensajes consecuentes a las peticiones de entrada, el tiempo de funcionamiento o cualquier otro tiempo que quiera visualizarse mediante el RTCC (*Real Time Clock and Calendar*) etc.

4.2.2 USB

Gracias al módulo USB, los parámetros del firmware del Sistema de Control podrían modificarse sin necesidad de una recompilación.

La instalación de un conector USB en el exterior del armario de control haría viable la interacción de un usuario con el firmware. De este modo, no sólo la parametrización sería posible, sino una observación in situ de cualquiera de los valores que quisieran monitorizarse.

Microchip dispone de librerías y pilas para la programación del módulo USB [38]. Sería necesario, lógicamente, realizar la programación de una interfaz para PC.

En este caso, el tiempo dedicado a la interfaz USB sería objeto de estudio, pues habría de determinarse en qué condiciones el bus CAN puede seguir siendo monitorizado.

4.2.3 Ethernet

La presencia de un módulo de Ethernet en el PIC32MX795F512L [9] hace posible el envío de resultados, estado o cualquier dato deseable del banco de ensayo (incluso del propio Sistema de Control) a través de la red hasta una aplicación para la visualización por el usuario controlador.

Aprovechando la red Ethernet extendida por la factoría de John Deere Ibérica, el ingeniero de mantenimiento podría controlar el estado de los ensayos desde un PC remoto sin necesidad de acudir a la ubicación del banco de pruebas.

Aparte de la codificación del firmware, debería programarse una aplicación para PC, y de nuevo, debería estudiarse el tiempo de dedicación al módulo Ethernet frente a la monitorización del bus CAN.

4.2.4 CAN

Tras el estudio necesario de tiempos de proceso, el Sistema de Control podría emplearse para el control simultáneo de dos bancos de ensayo. Esta posibilidad implica el uso del segundo módulo CAN presente en el PIC32MX795F512L.

4.2.5 Otros usos

El conjunto puede ser empleado para innumerables propósitos, tales como: calibración de sensores y actuadores, medición de tensiones, generador de ondas, generación de ensayos automáticos, controladores de pequeños ensayos, reguladores PID...

Hay posibilidades del PIC32 que no se han utilizado en este proyecto [9], como los módulos ADC, comparadores, etc.

Capítulo 5

Presupuesto

El presupuesto global asociado al proyecto realizado se desglosa a continuación:



UNIVERSIDAD CARLOS III DE MADRID Escuela Politécnica Superior

PRESUPUESTO DE PROYECTO

1.- Autor:

Rafael Jurado
Molinero

2.- Departamento:

Ingeniería de Sistemas
y Automática

3.- Descripción del Proyecto:

**Diseño e implementación de un sistema
de control de seguridad en bancos de
ensayo.**

- Título

- Duración (meses) 6

Tasa de costes

Indirectos:

20%

4.- Presupuesto total del Proyecto (valores en Euros):

19.409,00 Euros

5.- Desglose presupuestario (costes directos)

PERSONAL

Apellidos y nombre	N.I.F. (no rellenar - solo a título informativo)	Categoría	Dedicación (hombres mes) ^{a)}	Coste hombre mes	Coste (€)	Firma de conformidad	
Jurado Molinero, Rafael		Ingeniero Senior	6	4.289,54	0,00		
		Ingeniero		2.694,39	0,00		
					16.166,34		
					0,00		
					0,00		
Hombres mes 6				Total	16.166,34		

^{a)} 1 Hombre mes = 131,25 horas. Máximo anual de dedicación de 12 hombres mes (1575 horas)

Máximo anual para PDI de la Universidad Carlos III de Madrid de 8,8 hombres mes (1.155 horas)

EQUIPOS

Descripción	Coste (Euro)	% Uso dedicado proyecto	Dedicación (meses)	Periodo de depreciación	Coste imputable ^{d)}
Cerebot	90,00	100	5	60	7,50
Soporte PCB carril DIN	38,28	100	0	60	0,00
Fuente 5VDC	56,30	100	0	60	0,19
Relé 5/24	58,38	100	0	60	0,19
Relé 24/5	57,80	100	0	60	0,19
					0,00
Total					8,07

^{d)} Fórmula de cálculo de la Amortización:

$$\frac{A}{B} \times C \times D$$

A = nº de meses desde la fecha de facturación en que el equipo es utilizado

B = periodo de depreciación (60 meses)

C = coste del equipo (sin IVA)

D = % del uso que se dedica al proyecto (habitualmente 100%)

**6.- Resumen de
costes**

- - -

Presupuesto Costes Totales	Presupuest o Costes Totales
Personal	16.166
Amortización	8
Subcontratación de tareas	0
Costes de funcionamiento	0
Costes Indirectos	3.235
Total	19.409

Bibliografía

CAN

[1] *CAN Bus Performance Analysis*.

Dr. K.H. Tang, R.T. McLaughlin, Eva D'Souza. University of Warwick.

[2] *Bosch Controller Area Network protocol specification, Revision 2.0*.

Robert Bosch GmbH, 1991.

[3] *Controller Area Network (CAN). Basics*.

Keith Pazul. Microchip Technologies, 2005.

[4] *El bus CAN*.

Departamento de Ingeniería de Sistemas y Automática. Universidad de Oviedo.

[5] *Understanding Microchip's CAN Module Bit Timing (AN754)*.

Pat Richards, Microchip Technologies, 2005.

CANOpen

[6] *CANOpen Memento*.

Francis Dupin. LIVIC – Laboratoire de recherche sur les Interactions Véhicule - Infrastructure - Conducteur.2006.

[7] *CAN y CANOpen*.

Raquel Sánchez Díaz. Departamento de Informática y Automática. Facultad de Ciencias. Universidad de Salamanca.

[8] *CANopen Object Dictionary Reference*.

Advanced Motor Controls.

Microchip

[9] PIC32MX5XX/6XX/7XX Family Data Sheet High-Performance, USB, CAN and Ethernet 32-bit Flash Microcontrollers.

[10] PIC32 Peripheral Libraries for MPLAB C32 Compiler.

[11] MPLAB® C Compiler for PIC32 MCUs user's Guide.

[12] MPLAB® IDE User's Guide with MPLAB Editor and MPLAB SIM Simulator.

[13] Section 4. Memory Organisation.

[14] Section 5. Flash Programming.

[15] Section 8. Interrupts.

[16] Section 9. Watchdog Timer and Power-up Timer.

- [17] Section 13. Parallel Master Port (PMP).
- [18] Section 14. Timers.
- [19] Section 32. Configuration.
- [20] Section 34. Controller Area Network (CAN).

Digilent

- [21] 502-223.Cerebot MX7cK reference manual.
- [22] 500-223.Cerebot MX7cK schematics.
- [23] DSD-0000330. Demonstration projects for the Cerebot MX7cK.

Microprocesadores

- [24] *Arquitectura de Computadores*.
M. Morris Mano, Prentice Hall Hispanoamericana. HISPANOAMERICANA
- [25] *Beyond the Hype: MIPS® - the Processor for MCUs*.
MIPS Technologies, Inc., 2010.
- [26] *Programming 32-bit Microcontrollers in C.Exploring the PIC32*.
Lucio Di Jasio. Elsevier, 2008.

Otros

- [27] *LabVIEW User Manual*.
National Instruments, 2003
- [28] *An In-depth Look at the MCP2551 (AN739)*.
Pat Richards, Microchip Technologies, 2003.
- [29] *Technical Catalogue. PNOZmulti Product Range*.
Pilz GmbH & Co., 2004.

Páginas Web

- [30] CiA (CAN in Automation)
<http://www.can-cia.org/>
Última fecha de revisión: 20/10/2012

- [31] CANopen: High level protocol for CAN-bus
<http://www.nikhef.nl/pub/departments/ct/po/doc/CANopen30.pdf>
Última fecha de revisión: 20/10/2012
- [32] National Instruments.
<http://sine.ni.com/nips/cds/view/p/lang/es/nid/203384>
Última fecha de revisión: 20/10/2012
- [33] Digilent.
<http://www.digilentinc.com/>
Última fecha de revisión: 20/10/2012
- [34] Phoenix Contact.
<http://www.phoenixcontact.com>
Última fecha de revisión: 20/10/2012
- [35] RS-Amidata.
<http://es.rs-online.com>
Última fecha de revisión: 20/10/2012
- [36] Pilz International.
<https://shop.pilz.com/eshop/b2b/publicinit.do?category=A0011B0019>
Última fecha de revisión: 20/10/2012
- [37] KVaser. Advanced CAN solutions.
<http://www.kvaser.com/>
Última fecha de revisión: 20/10/2012
- [38] Microchip Technologies.
<http://www.Microchip.com>
Última fecha de revisión: 20/10/2012

Anexo A

Hojas de Características

A continuación se incluyen las hojas de características que constituyen fragmentos de los catálogos de los equipos y componentes utilizados en este Proyecto.



PIC32MX5XX/6XX/7XX

High-Performance, USB, CAN and Ethernet 32-bit Flash Microcontrollers

High-Performance 32-bit RISC CPU:

- MIPS32® M4K® 32-bit core with 5-stage pipeline
- 80 MHz maximum frequency
- 1.56 DMIPS/MHz (Dhrystone 2.1) performance at zero Wait state Flash access
- Single-cycle multiply and high-performance divide unit
- MIPS16e® mode for up to 40% smaller code size
- Two sets of 32 core register files (32-bit) to reduce interrupt latency
- Prefetch Cache module to speed execution from Flash

Microcontroller Features:

- Operating voltage range of 2.3V to 3.6V
- 64K to 512K Flash memory (plus an additional 12 KB of Boot Flash)
- 16K to 128K SRAM memory
- Pin-compatible with most PIC24/dsPIC® DSC devices
- Multiple power management modes
- Multiple interrupt vectors with individually programmable priority
- Fail-Safe Clock Monitor mode
- Configurable Watchdog Timer with on-chip Low-Power RC oscillator for reliable operation

Peripheral Features:

- Atomic SET, CLEAR and INVERT operation on select peripheral registers
- Up to 8-channels of hardware DMA with automatic data size detection
- USB 2.0-compliant full-speed device and On-The-Go (OTG) controller:
 - Dedicated DMA channels
- 10/100 Mbps Ethernet MAC with MII and RMII interface:
 - Dedicated DMA channels
- CAN module:
 - 2.0B Active with DeviceNet™ addressing support
 - Dedicated DMA channels
- 3 MHz to 25 MHz crystal oscillator

Peripheral Features (Continued):

- Internal 8 MHz and 32 kHz oscillators
- Six UART modules with:
 - RS-232, RS-485 and LIN support
 - IrDA® with on-chip hardware encoder and decoder
- Up to four SPI modules
- Up to five I²C™ modules
- Separate PLLs for CPU and USB clocks
- Parallel Master and Slave Port (PMP/PSP) with 8-bit and 16-bit data, and up to 16 address lines
- Hardware Real-Time Clock and Calendar (RTCC)
- Five 16-bit Timers/Counters (two 16-bit pairs combine to create two 32-bit timers)
- Five Capture inputs
- Five Compare/PWM outputs
- Five external interrupt pins
- High-speed I/O pins capable of toggling at up to 80 MHz
- High-current sink/source (18 mA/18 mA) on all I/O pins
- Configurable open-drain output on digital I/O pins

Debug Features:

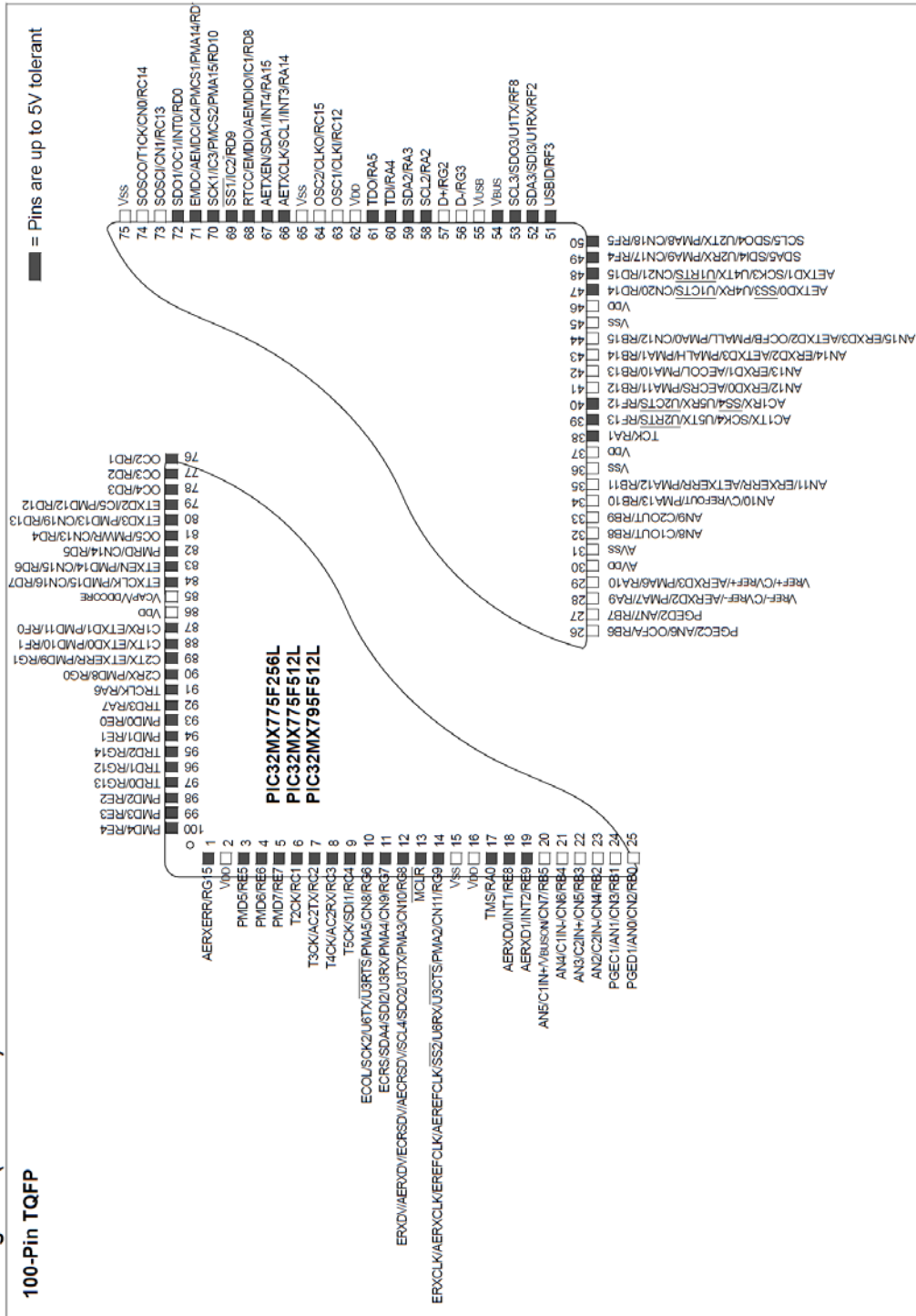
- Two programming and debugging Interfaces:
 - 2-wire interface with unintrusive access and real-time data exchange with application
 - 4-wire MIPS® standard enhanced Joint Test Action Group (JTAG) interface
- Unintrusive hardware-based instruction trace
- IEEE Standard 1149.2 compatible (JTAG) boundary scan

Analog Features:

- Up to 16-channel, 10-bit Analog-to-Digital Converter:
 - 1 Msps conversion rate
 - Conversion available during Sleep and Idle
- Two Analog Comparators

PIC32MX5XX/6XX/7XX

Pin Diagrams (Continued)



PIC32MX5XX/6XX/7XX

TABLE 6: PIN NAMES: PIC32MX775F256L, PIC32MX775F512L AND PIC32MX795F512L DEVICES

Pin Number	Full Pin Name	Pin Number	Full Pin Name
A1	PMD4/RE4	E8	AETXEN/SDA1/INT4/RA15
A2	PMD3/RE3	E9	RTCC/EMDIO/AEMDIO/IC1/RD8
A3	TRD0/RG13	E10	SS1/IC2/RD9
A4	PMD0/RE0	E11	AETXCLK/SCL1/INT3/RA14
A5	C2RX/PMD8/RG0	F1	MCLR
A6	C1TX/ETXD0/PMD10/RF1	F2	ERXDV/AERXDV/ECRSDV/AECRSDV/SCL4/SDO2/ U3TX/PMA3/CN10/RG8
A7	VDD	F3	ERXCLK/AERXCLK/EREFCLK/AEREFCLK/SS2/U6RX/ U3CTS/PMA2/CN11/RG9
A8	VSS	F4	ECRS/SDA4/SDI2/U3RX/PMA4/CN9/RG7
A9	ETXD2/IC5/PMD12/RD12	F5	VSS
A10	OC3/RD2	F6	No Connect (NC)
A11	OC2/RD1	F7	No Connect (NC)
B1	No Connect (NC)	F8	VDD
B2	AERXERR/RG15	F9	OSC1/CLKI/RC12
B3	PMD2/RE2	F10	VSS
B4	PMD1/RE1	F11	OSC2/CLKO/RC15
B5	TRD3/RA7	G1	AERXD0/INT1/RE8
B6	C1RX/ETXD1/PMD11/RF0	G2	AERXD1/INT2/RE9
B7	VCAP/CORE	G3	TMS/RA0
B8	PMRD/CN14/RD5	G4	No Connect (NC)
B9	OC4/RD3	G5	VDD
B10	VSS	G6	VSS
B11	SOSCO/T1CK/CN0/RC14	G7	VSS
C1	PMD6/RE6	G8	No Connect (NC)
C2	VDD	G9	TDO/RA5
C3	TRD1/RG12	G10	SDA2/RA3
C4	TRD2/RG14	G11	TDI/RA4
C5	TRCLK/RA6	H1	AN5/C1IN+/VBUSON/CN7/RB5
C6	No Connect (NC)	H2	AN4/C1IN-/CN6/RB4
C7	ETXCLK/PMD15/CN16/RD7	H3	VSS
C8	OC5/PMWR/CN13/RD4	H4	VDD
C9	VDD	H5	No Connect (NC)
C10	SOSCI/CN1/RC13	H6	VDD
C11	EMDC/AEMDC/IC4/PMCS1/PMA14/RD11	H7	No Connect (NC)
D1	T2CK/RC1	H8	VBUS
D2	PMD7/RE7	H9	VUSB
D3	PMD5/RE5	H10	D+/RG2
D4	VSS	H11	SCL2/RA2
D5	VSS	J1	AN3/C2IN+/CN5/RB3
D6	No Connect (NC)	J2	AN2/C2IN-/CN4/RB2
D7	ETXEN/PMD14/CN15/RD6	J3	PGED2/AN7/RB7
D8	ETXD3/PMD13/CN19/RD13	J4	AVDD
D9	SDO1/OC1/INT0/RD0	J5	AN11/ERXERR/AETXERR/PMA12/RB11
D10	No Connect (NC)	J6	TCK/RA1
D11	SCK1/IC3/PMCS2/PMA15/RD10	J7	AN12/ERXD0/AECRS/PMA11/RB12
E1	T5CK/SD11/RC4	J8	No Connect (NC)
E2	T4CK/AC2RX/RC3	J9	No Connect (NC)
E3	ECOL/SCK2/U6TX/U3RTS/PMA5/CN8/RG6	J10	SCL3/SDO3/U1TX/RF8
E4	T3CK/AC2TX/RC2	J11	D-/RG3
E5	VDD	K1	PGEC1/AN1/CN3/RB1
E6	C2TX/ETXERR/PMD9/RG1	K2	PGED1/AN0/CN2/RB0
E7	VSS	K3	VREF+/CVREF+/AERXD3/PMA6/RA10

PIC32MX5XX/6XX/7XX

TABLE 6: PIN NAMES: PIC32MX775F256L, PIC32MX775F512L AND PIC32MX795F512L DEVICES (CONTINUED)

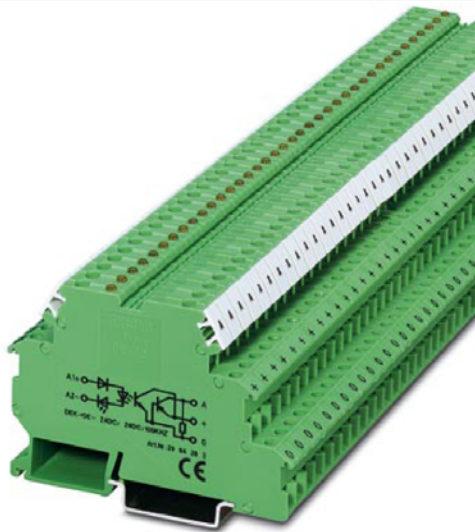
Pin Number	Full Pin Name	Pin Number	Full Pin Name
K4	AN8/C1OUT/RB8	L3	AVss
K5	No Connect (NC)	L4	AN9/C2OUT/RB9
K6	AC1RX/SS4/U5RX/U2CTS/RF12	L5	AN10/CVREFOUT/PMA13/RB10
K7	AN14/ERXD2/AETXD3/PMALH/PMA1/RB14	L6	AC1TX/SCK4/U5TX/U2RTS/RF13
K8	VDD	L7	AN13/ERXD1/AECOL/PMA10/RB13
K9	AETXD1/SCK3/U4TX/U1RTS/CN21/RD15	L8	AN15/ERXD3/AETXD2/OCFB/PMALU/PMA0/CN12/RB15
K10	USBID/RF3	L9	AETXD0/SS3/U4RX/U1CTS/CN20/RD14
K11	SDA3/SDI3/U1RX/RF2	L10	SDA5/SDI4/U2RX/PMA9/CN17/RF4
L1	PGEC2/AN6/OCFA/RB6	L11	SCL5/SDO4/U2TX/PMA8/CN18/RF5
L2	VREF-/CVREF-/AERXD2/PMA7/RA9		

DEK-OE- 5DC/24DC/100KHZ



Please note that the data given here has been taken from the online catalog. For comprehensive information and data, please refer to the user documentation at <http://www.download.phoenixcontact.com>. The General Terms and Conditions of Use apply to Internet downloads.

► Extract from the online catalog



Input optocoupler, transmission frequency 100 kHz, with light indicator and protection circuit in input and output circuit, input: 5 V DC, output: 4 - 30 V DC/ 50 mA

The illustration shows the version DEK-OE- 24DC/24DC/100KHZ

Order No.	2964270
Ord designation	DEK-OE- 5DC/24DC/100KHZ
EAN	4017918107406
Pack	10 Pcs.
Customs tariff	85364190
Catalog page information	Page 114 (IF-2005)

► Technical data

Input data

Nominal input voltage U_N	5 V DC $\pm 20\%$
Switching threshold "0" signal in reference to U_N	
Switching threshold "1" signal in reference to U_N	> 0.8
Typical input current at U_N	7 mA
Transmission frequency	100 kHz
Typical response time	1.5 μ s
Typical turn-off time	2 μ s

DEK-OE- 5DC/24DC/100KHZ

**Output data**

Output nominal voltage range	4 V DC ... 30 V DC
Limiting continuous current	50 mA
Quiescent current	4.3 mA
Voltage drop at max. limiting continuous current	≤ 0.5 V DC
Protective circuitry	Surge protection Suppressor diode

Connection data

Type of connection	Screw connection
Stripping length	8 mm
Screw thread	M 3
Conductor cross section, rigid min.	0.2 mm ²
Conductor cross section, rigid max.	4 mm ²
Conductor cross section flexible min.	0.2 mm ²
Conductor cross section, flexible max.	2.5 mm ²
Conductor cross section AWG/kcmil min.	24
Conductor cross section AWG/kcmil max	12

General data

Length	80 mm
Width	6.2 mm
Height	56 mm
Test voltage input/output	2.5 kV AC
Ambient temperature (operation)	-20 °C ... 60 °C
Installation position	Any
Assembly instructions	In rows with zero spacing
Operating mode	100% operating factor
Standards/regulations	IEC 60664
Standards/regulations	IEC 60664 A
Standards/regulations	DIN VDE 0110
Contamination class	2
Surge voltage category	II



Datos básicos

DEK-OE- 24DC/ 5DC/100KHZ-G

Código de artículo: 2964364

La figura muestra la variante DEK-OE- 24DC/24DC/100KHZ-G




<http://catalog.phoenixcontact.net/phoenix/treeViewClick.do?UID=2964364>

Relé de estado sólido de introducción, con contrafase de salida de tensión continua, frecuencia de transferencia de 100 kHz, con indicador luminoso y circuito de protección en circuito de entrada y salida, entrada: 24 V DC, salida: 4 - 18 V DC/50 mA



Datos mercantiles

EAN	 4 017918 107437
sales group	G120
Paquete	10 Pcs.
Arancel	85364190
Peso bruto en unidades	0,02525 KG
Peso neto por unidad	0,02383 KG
Página del catálogo	Página 151 (IF-2011)

Observaciones acerca del producto

Conforme a WEEE/RoHS desde:
02.01.2006



Tenga en cuenta que los datos indicados aquí proceden del catálogo en línea. Los datos completos se encuentran en la documentación para el usuario en <http://www.download.phoenixcontact.es> Son válidas las condiciones generales de uso de las descargas por Internet.

Datos técnicos

Datos de entrada

Tensión nominal de entrada U_N	24 V DC
Margen de tensión de entrada referido a U_N	0,8 ... 1,2
Umbral de conmutación Señal "0" referido a U_N	$\leq 0,4$
Umbral de conmutación Señal "1" referido a U_N	$\geq 0,8$

DEK-OE- 24DC/ 5DC/100KHZ-G Código de artículo: 2964364
<http://catalog.phoenixcontact.net/phoenix/treeViewClick.do?UID=2964364>

Corriente de entrada típica a U_N	8 mA
Tiempo de conexión típico	1 μ s
Tiempo de desconexión típico	2 μ s
Indicación de la tensión de servicio	LED amarillo
Denominación de la protección	Prot. contra inversión de polaridad
	Prot. contra sobretensiones
Circuito de protección/componente de protección	Diodo contra inv. de polaridad
Frecuencia de transmisión	100 kHz

Datos de salida

Gama de tensión de salida	4 V DC ... 18 V DC
Corriente constante límite	50 mA
Corriente de reposo	8,5 mA
Caída de tensión con corriente constante límite máxima	$\leq 1,2$ V DC
Circuito de salida	3 conductores contrafase, con masa
Denominación de la protección	Prot. contra sobretensiones
Circuito de protección/componente de protección	Diodo supresor

Datos de conexión

Tipo de conexión	Conexión por tornillo
Longitud a desaislar	8 mm
Rosca de tornillo	M3
Sección de conductor rígido mín.	0,2 mm ²
Sección de conductor rígido máx.	4 mm ²
Sección de conductor flexible mín.	0,2 mm ²
Sección de conductor flexible máx.	2,5 mm ²
Sección de conductor AWG/kcmil mín.	24
Sección de conductor AWG/kcmil máx.	12

Datos generales

Anchura	6,2 mm
Altura	80 mm
Profundidad	56 mm
Tensión de prueba entrada/salida	2,5 kV AC
Temperatura ambiente (servicio)	-20 °C ... 60 °C
Temperatura ambiente (almacenamiento / transporte)	-20 °C ... 70 °C

P06012 010102

entrelec

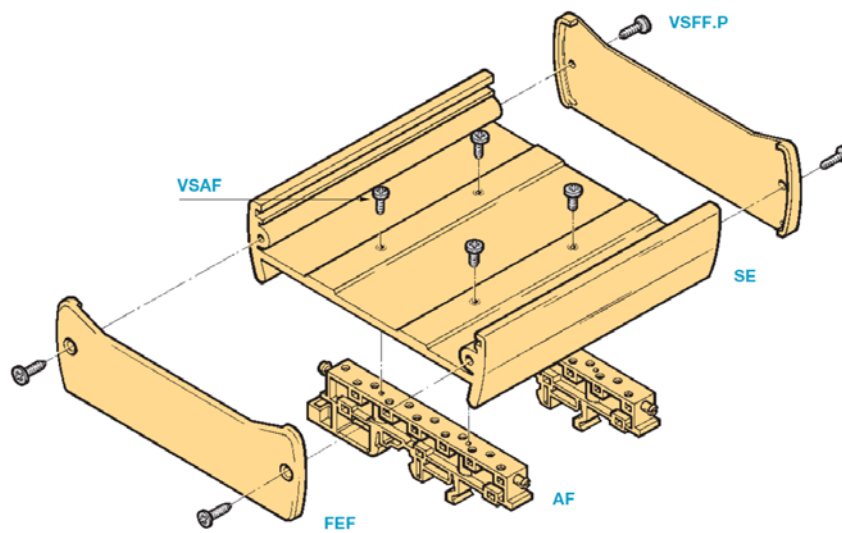
PRINTED CIRCUIT BOARD HOLDER

"Series 20 000"

Features

The extruded PVC series 20 000 holders allow:

- 2 PCB width: 71 mm or 100 mm
- PCB length: 60 mm to 1000 mm
- PCB thickness: 1,6 mm
- Assembly snaps onto DIN 1 or DIN 3 standard rails (EN 50035 and 50022)



Complete assembly includes:

- 1 body 71 or 100 mm
- 2 end plates
- 4 screws for end plates
- 2 feet for length 60 to 250 mm
- 3 feet for length 250 to 500 mm
- 4 feet for length 500 to 750 mm
- 5 feet for length 750 to 1000 mm
- 2 screws per foot

Actual PCB width	71 mm	2.795"	100 mm	3.937"	Std packaging
Body L = 1000 mm	SE 71	0103868.06	SE 100	0103885.10	1 m
End plate	FEF 71	0103724.25	FEF 100	0103687.21	10
Screws for end plate	VSFF.P	0178100.13	VSFF.P1	0178100.13	
	Pozidriv		Pozidriv		
Foot	AF 50	0103323.23	AF 76	0103324.24	10
Screw for foot	VSAF	0174582.16	VSAF	0174582.16	100

Note: drilling for screws VSAF are not done.

P06013 010102

entretec

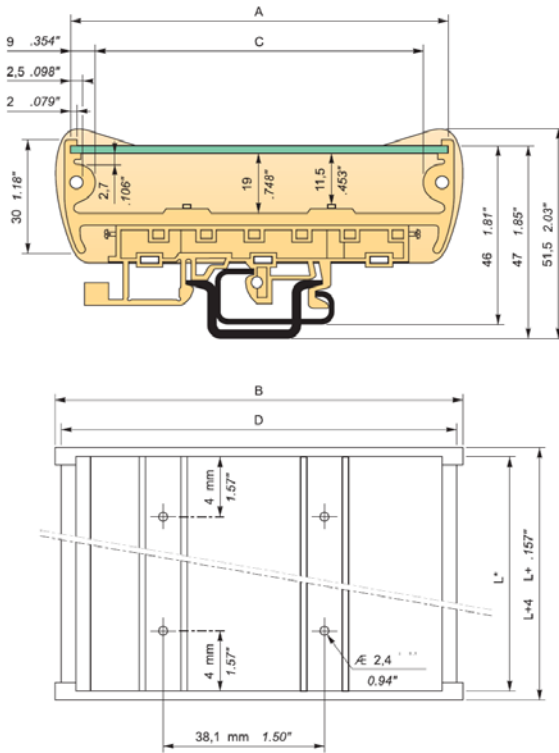
Material specification

Body: (delivered in 1000 mm length)
PVC rigid extruded class M2 (orange)

End plate and Mounting foot:
Molded polyamide UL 94 V2 (orange)

Dimensions

Holder

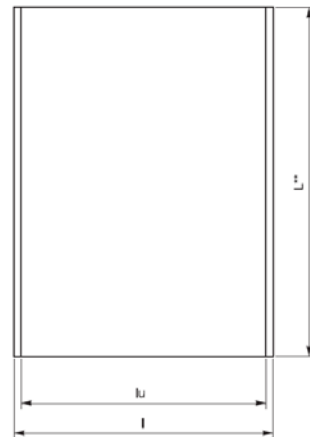


Drill from bottom in center of grooves.

*Rail length margin L: + 0,5 .020"
+ 0,2 .008"

	SE 71		SE 100	
	mm	inch.	mm	inch.
A	71	2.79"	100	3.94"
B	78	3.07"	107	4.21"
C	57,8	2.28"	86,5	2.95"
D	75	2.96"	104	4.09"

PCB



	SE 71		SE 100	
	mm	inch.	mm	inch.
l	71	2.79"	100	3.94"
lu (1)	66	2.60"	95	3.74"
lu (2)	57,8	2.28"	86,5	2.95"

(1): For component pins less than 2 mm.
(2): For component pins longer than 2 mm.

** PCB length margin L: 0 .000"
- 0,2 -.008"

CAN and LIN Interfaces for Hi-Speed USB

NI USB-8472, NI USB-8472s, NI USB-8473, NI USB-8473s, NI USB-8476, NI USB-8476s,

- 1-port interfaces for high-speed CAN, low-speed/fault-tolerant CAN, and LIN
- Optional hardware synchronization
- Hi-Speed USB, bus-powered
- 500 V digital isolation
- Bus error logging
- Hardware timestamping – 1 μ s resolution

CAN Interfaces

- Philips SJA1000 CAN controller
- Transmit/receive 100 percent bus load at 1 Mb/s
- ISO 11898 compliance for standard (11-bit) and extended (29-bit) arbitration IDs
- Software-selectable termination for low-speed/fault-tolerant CAN
- J1939 compliance

LIN Interfaces

- Atmel ATA6620 transceiver
- LIN 1.3/2.0 and J2602 compliance
- Software-selectable master/slave termination

Operating Systems

- Windows Vista/XP/2000

Recommended Software

- LabVIEW
- LabWindows™/CVI
- Visual C++ 6.0
- Visual Basic 6.0
- Borland C/C++

Application Software (included)

- CAN/LIN bus monitoring and logging utility

Driver Software (included)

- NI-CAN



Model	Physical Layer	Transceivers	Ports	Max Transfer Rate (kb/s)	Hardware Sync	Software Termination	API
USB-8472	Low-speed/fault-tolerant CAN	TJA1054AT	1	125	–	✓	Frame
USB-8472s	Low-speed/fault-tolerant CAN	TJA1054AT	1	125	✓	✓	Frame
USB-8473	High-speed CAN	TJA1041	1	1,000	–	–	Frame
USB-8473s	High-speed CAN	TJA1041	1	1,000	✓	–	Frame
USB-8476	LIN	ATA6620	1	20	–	–	Frame
USB-8476s	LIN	ATA6620	1	20	✓	✓	Frame

Table 1. NI USB-847x Selection Guide

Overview

NI USB-847x devices provide Hi-Speed USB interfaces for Controller Area Network (CAN) and Local Interconnect Network (LIN) monitoring, logging, and testing. With high-speed CAN, low-speed/fault-tolerant CAN, and LIN interfaces featuring optional hardware synchronization, you can use two or more USB-847x devices together to interface to a wide variety of CAN and LIN networks.

The USB-847x interfaces are ideal for many types of applications, including:

- In-vehicle network monitoring and logging
- Bus load monitoring
- Device validation with synchronized data acquisition
- CAN device development and test
- CAN and LIN data correlation with external measurements

The convenient all-in-one design features a captive 2 m USB cable and built-in transceiver, requiring no extra cables or accessories to get applications running quickly.

With hardware timestamping, you can log messages with microsecond-accurate timestamps for reconstructing network events and correlating data across synchronized devices. All USB-847x interfaces use an industry-standard 9-pin male D-Sub (DB9) connector to interface to CAN and LIN buses.

Hi-Speed USB connectivity and onboard frame buffering ensure no dropped frames, even under 100 percent bus loads.

CAN Interfaces

USB-847x CAN interfaces feature the industry-standard Philips SJA1000 CAN controller, which implements ISO 11898 CAN functionality. The SJA1000 offers additional features to aid in system development, including listen-only mode, sleep/wakeup mode, error counter access, and self-reception (echo) mode. USB-847x CAN interfaces recognize standard (11-bit) and extended (29-bit) arbitration IDs and are compatible with J1939 networks.



CAN and LIN Interfaces for Hi-Speed USB

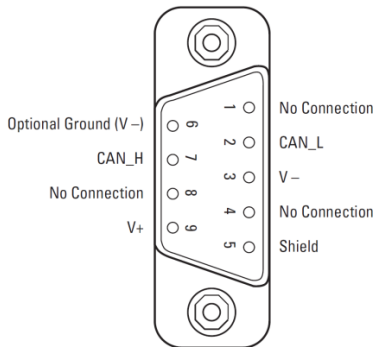


Figure 1. CAN DB9 Connector (USB CAN Modules)

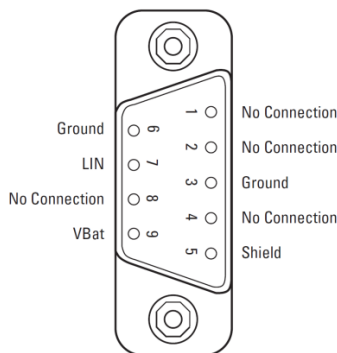


Figure 2. LIN DB9 Connector (USB LIN Modules)

Low-speed/fault-tolerant USB-847x CAN interfaces offer software-selectable termination for compatibility with all low-speed network configurations.

Onboard Physical Layer

The CAN physical layer connects the CAN controller to the physical bus wires. USB-847x CAN interfaces are available with high-speed and low-speed/fault-tolerant physical layers. All USB-847x devices have onboard physical layer transceivers and require no external dongles. The physical layers are internally powered via a DC-to-DC converter and electrically isolated up to 500 V.

LIN Interfaces

NI USB-8476 LIN interfaces, featuring the Atmel ATA6620 LIN transceiver, are compliant with the LIN 1.3/2.0 and J2602 specifications and offer software-selectable master/slave termination. The ATA6620

features baud rates up to 20 kb/s and offers advanced power management with a low-power sleep mode.

Hardware Synchronization

Many automotive applications demand tight integration of CAN, LIN, analog, and digital measurements. In applications such as CAN device development and validation, you need to synchronize different measurements to correlate data. You can program this synchronization in software, but OS latency and clock drift between the different onboard oscillators introduce unacceptable delays for certain automotive test applications.

NI offers USB-847x CAN and LIN interfaces with an optional hardware synchronization feature, with which the USB interfaces can share timing and triggering signals with other USB-847x interfaces, as well as data acquisition, vision, and motion devices, to achieve true hardware synchronization. Determinism is maintained between the trigger signal and the desired response because timing and triggering signals are handled in hardware. The host PC software interacts only to retrieve the data once it is acquired or to write new data.

NI-CAN Software

National Instruments provides NI-CAN driver software for Windows Vista/XP/2000. NI-CAN includes a detailed API and examples for NI LabVIEW and LabWindows/CVI, Microsoft Visual C/C++ 6.0 and Visual Basic 6.0, and Borland C/C++. USB-847x interfaces use the NI-CAN Frame API for frame-level access to messages on CAN and LIN networks. The NI-CAN driver software also includes the Bus Monitor and NI Spy utilities to aid in application development.

NI-CAN Frame API

The NI-CAN Frame API offers a powerful interface for accessing NI CAN hardware. The API enables full access to all traffic on CAN and LIN buses. It is a powerful tool for implementing a variety of applications, including CAN and LIN frame-level access, challenge response protocols, remote frame handling, and advanced synchronization.

Bus Monitor

To quickly monitor all CAN and LIN bus traffic, use Bus Monitor, a utility that offers an easy-to-use interface that displays all CAN and LIN frames on the network, and log the traffic to disk. Bus Monitor provides options to control, display, and view bus statistics.

CAN and LIN Interfaces for Hi-Speed USB

NI Spy

NI Spy is a utility for monitoring NI-CAN API calls made by applications without recompiling or rebuilding. It is ideal for testing application functionality, troubleshooting problems, and verifying communication with CAN devices. NI Spy dynamically captures and displays all NI-CAN API calls made by applications running on the computer.

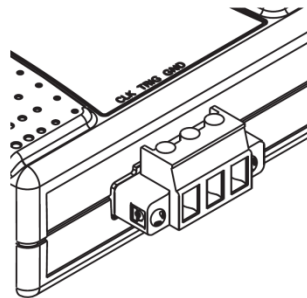


Figure 3. COMBICON Synchronization Connector for USB-847x Devices

Ordering Information

NI USB-8472

1-port, low-speed, USB CAN interface779790-01

NI USB-8472s

1-port, low-speed, USB CAN interface
with synchronization779791-01

NI USB-8473

1-port, high-speed, USB CAN interface779792-01

NI USB-8473s

1-port, high-speed, USB CAN interface
with synchronization779793-01

NI USB-8476

1-port, USB LIN interface779794-01

NI USB-8476s

1-port, USB LIN interface with synchronization779795-01

CAN Device Simulator

U.S. 120 VAC779189-01

Euro 240 VAC779189-02

Japan 100 VAC779189-07

Cables and Brackets

USB cable strain relief bracket777550-01

CAN single-termination, high-speed cable, 2 m192017-02

Specifications

USB Power Requirements

USB-8472(s), USB-8473(s)	250 mA, 5 VDC
USB-8476(s).....	200 mA, 5 VDC

Hardware Synchronization

Input clocks	1, 10, and 20 MHz
Output clock	1 MHz
Trigger lines.....	1 input/output
Clock lines.....	1 input/output

Examples included for sync to NI-DAQmx, LIN, CAN, and RTSI bus.

Supported Baud Rates

USB-8472(s).....	5,000 to 125,000 baud
USB-8473(s).....	40,000 to 1,000,000 baud
USB-8476(s).....	2,400 to 20,000 baud

Safety

Galvanic CAN channel to USB isolation	500 V
Withstand.....	2 s maximum

Dimensions

Nonsynchronized versions	7.87 by 6.35 by 2.54 cm (3.1 by 2.5 by 1.0 in.)
Synchronized versions	7.87 by 7.11 by 2.54 cm (3.1 by 2.8 by 1.0 in.)
Cable length.....	2 m
I/O connector	9-pin male D-Sub, optional 3-pin COMBICON for synchronization

Operating Environment

Ambient temperature.....	0 to 55 °C
Relative humidity	10 to 90%, noncondensing (tested in accordance with IEC-60068-2-1, IEC-60068-2-2, IEC-60068-2-56)

BUY NOW!

For complete product specifications, pricing, and accessory information, call 800 813 3693 (U.S.) or go to ni.com/can.

Cerebot MX7cK™ Board Reference Manual

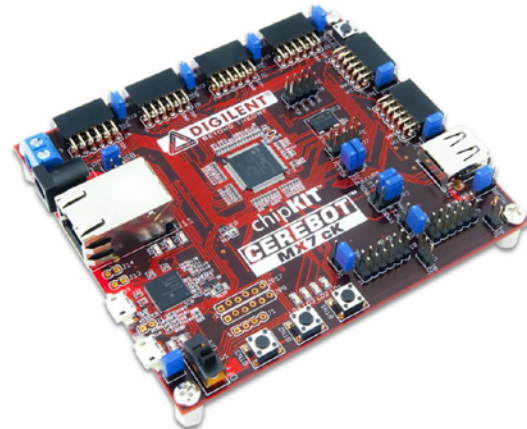
Revision: January 3, 2012
 Note: This document applies to REV B of the board.



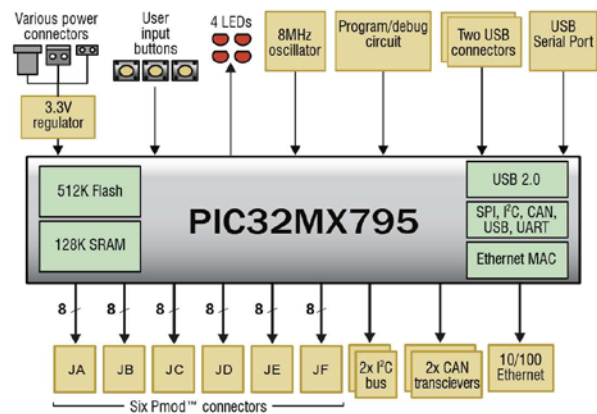
1300 Henley Court | Pullman, WA 99163
 (509) 334 6306 Voice and Fax

Overview

The Cerebot MX7cK is a microcontroller development board based on the Microchip PIC32MX795F512L, a member of the 32-bit PIC32 microcontroller family. It is compatible with Diligent's line of Pmod™ peripheral modules, and is suitable for use with the Microchip MPLAB® IDE tools. The Cerebot MX7cK is also compatible for use with the chipKIT™ MPIDE development environment. ChipKIT and MPIDE is a PIC32 based system compatible with many existing Arduino™ code examples, reference materials and other resources.



The Cerebot MX7cK is designed to be easy to use and suitable for use by anyone from beginners to advanced users for experimenting with embedded control and network communications application. A built in programming/debugging circuit compatible with the Microchip MPLAB® IDE is provided, so no additional hardware is required for use with MPLAB. The kit contains everything needed to start developing embedded applications using either the MPLAB® IDE or the MPIDE.



Cerebot MX7cK Circuit Diagram

The Cerebot MX7cK provides 52 I/O pins that support a number of peripheral functions, such as UART, SPI and I²C™ ports as well as five pulse width modulated outputs and five external interrupt inputs. Its network and communications features also include a 10/100 Ethernet interface, Full Speed USB 2.0 OTG interface, and dual CAN network interfaces. Ten of the I/O pins can be used as analog inputs in addition to their use as digital inputs and outputs.

The Cerebot MX7cK can be powered in various ways via USB, or using an external AC-DC power adapter.

Appendix A: Connector Descriptions and Jumper Settings

Label	Function
JA-JF	<p>Pmod Connectors</p> <p>These connectors provide access to the I/O pins on the PIC32MX795 microcontroller. Digilent Pmod peripheral modules can be attached to these connectors. These connectors can be used for general access to I/O pins on the PIC32MX795 microcontroller.</p>
JPA – JPF	<p>Pmod header power select</p> <p>Any of the Pmod connectors can provide either regulated or unregulated power. To use regulated power, place the jumper block over the center pin and the pin marked 3V3. To use unregulated power, place the jumper block over the center pin and the pin marked 5V0.</p>
J1	<p>USB Serial converter auxiliary signals</p> <p>This connector can be used to access the auxiliary RS232 handshaking signals not used on the Cerebot MX4cK board.</p>
J2	<p>USB Serial converter (UART) connector</p> <p>This USB micro-AB connector is used to connect the FT232R serial converter to a USB port on the user PC.</p>
J3	<p>Power supply source select</p> <p>This jumper is used to select the source of main board power.</p> <p>Place the shorting block in the, “USB” position to have the board powered from the USB device connector, J19.</p> <p>Place the shorting block in the, “EXT” position to have the board powered from one of the external power connectors, J17 or J18.</p> <p>Place the shorting block in the, “DBG” position to have the board powered from the debug USB connector, J15.</p> <p>Place the shorting block in the “UART” position to have the board powered from the USB serial converter connector, J2</p>
J7	<p>I2C1 daisy chain connector</p> <p>This connector provides access to the I2C signals, power and ground for I2C1.</p>
J8	<p>I2C2 daisy chain connector</p> <p>This connector provides access to the I2C signals, power and ground for I2C2.</p>
J9	<p>CAN #1 Connector</p> <p>This connector is used to access the signals for CAN #1.</p>
J10	<p>CAN #2 Connector</p> <p>This connector is used to access the signals for CAN #2.</p>
J11	<p>Ethernet Connector</p>

Cerebot MX7cK Reference Manual



	This connector provides access to the 10/100 Ethernet port.
J12- J14	Do Not Use.
J15	Debug USB Connector This connector is used to connect the on-board programming and debug circuit to the PC for use with the MPLAB IDE.
J17	External Power Connector This is a 2.5mm x 5.5mm, center positive, coax power connector used to provide external power to the board. The optional Diligent 5V Switching Power Supply is connected here.
J18	External Power Connector This is a screw terminal connector used to provide external power to the board. Be sure to observe proper polarity (marked near the connector) when providing power via this connector, or damage to the board and/or connected devices may result.
J19	USB Device / OTG Connector This is a USB micro-AB connector. It is used when using the PIC32MX795 microcontroller to implement a USB device or OTG Host/Device.
J20	USB Host Connector This is a standard sized USB type A connector. This connector is used to connect USB devices to the board when using the PIC32MX795 microcontroller to implement an embedded USB host.
JP1 & JP2	CAN or Pmod Select These jumpers select microcontroller signals RF12 and RF13 for use with CAN #1 or Pmod connector JF. Place these jumpers in the CAN position to use CAN #1. Place the jumpers in the PMOD position to use then with Pmod connector JF.
JP3 & JP4	Pull-up enable for I2C1 These two jumpers are used to enable/disable the pull-up resistors on I2C1. Insert shorting blocks on these two jumpers to enable the pull-up resistors. Remove the shorting blocks to disable the pull-up resistors. Only a single device on the I ² C bus should have the pull-up resistors enabled.
JP5	CAN #1 Termination This jumper is used to enable/disable the 120 ohm termination resistor for CAN #1. Insert the shorting block to enable the termination resistor, remove it to disable the termination resistor.
JP6	CAN #1 5V0 Enable This jumper is used to enable/disable providing 5V to the CAN #1 connector. Insert the shorting block to connect the board 5V0 supply to pins 9 & 10 of CAN #1 connector. Remove the shorting block to disconnect the 5V0 supply.
JP7	CAN #2 Termination This jumper is used to enable/disable the 120 ohm termination resistor for CAN #2. Insert the shorting block to enable the termination resistor, remove it to disable the termination resistor.

Cerebot MX7cK Reference Manual



JP8	CAN #1 5V0 Enable This jumper is used to enable/disable providing 5V to the CAN #1 connector. Insert the shorting block to connect the board 5V0 supply to pins 9 & 10 of CAN #1 connector. Remove the shorting block to disconnect the 5V0 supply.
JP9	Do Not Use
JP10	USB host power select This jumper is used to select which host connector is powered when host power is enabled. Place the shorting block in the "MICRO" position to supply power to the USB micro-AB OTG Connector, J19. Place the shorting block in the "A" position to supply power to the USB type A Host Connector, J20.
JP11	USB Serial converter reset disconnect This is used to connect/disconnect the USB serial converter reset circuit from the PIC32 MCLR pin. The shorting block must be in place on this jumper when using the chipKIT MPIDE development tools. Remove the shorting block if the USB serial converter is interfering with proper operation of the licensed debugger circuit.
JP17	Do Not Use

Appendix B: Example of Configuration Values

The following example illustrates setting the configuration values in the PIC32 microcontroller on the Cerebot MX7cK. The microcontroller configuration should be done in a single source file in the project, and is typically done in the 'main' project source file. This example sets all configuration values to valid values for the Cerebot MX7cK board. It sets the system clock for processor operation at 80Mhz, and the peripheral bus at 10Mhz.

```

/* ----- */
/*          PIC32 Configuration Settings          */
/* ----- */

/* Oscillator Settings
*/
#pragma config FNOSC    = PRIPLL    // Oscillator selection
#pragma config POSCMOD = EC        // Primary oscillator mode
#pragma config FPLLIDIV = DIV_2    // PLL input divider
#pragma config FPLLMUL  = MUL_20   // PLL multiplier
#pragma config FPLLODIV = DIV_1    // PLL output divider
#pragma config FPBDIV   = DIV_8     // Peripheral bus clock divider
#pragma config FSOCEN   = OFF       // Secondary oscillator enable

/* Clock control settings
*/
#pragma config IESO     = OFF        // Internal/external clock switchover
#pragma config FCKSM    = CSDCMD    // Clock switching (CSx)/Clock monitor (CMx)
#pragma config OSCIOFNC = OFF       // Clock output on OSCO pin enable

/* USB Settings
*/
#pragma config UPLEN    = ON         // USB PLL enable
#pragma config UPLLIDIV = DIV_2     // USB PLL input divider
#pragma config FVBUSONIO = OFF      // VBUS pin control
#pragma config FUSBIDIO = OFF       // USBID pin control

/* Other Peripheral Device settings
*/
#pragma config FWDTEN   = OFF        // Watchdog timer enable
#pragma config WDTPS    = PS1024    // Watchdog timer post-scaler
#pragma config FSRSEL   = PRIORITY_7 // SRS interrupt priority
#pragma config FCANIO   = OFF        // Standard/alternate CAN pin select
#pragma config FETHIO   = ON         // Standard/alternate ETH pin select
#pragma config FMIEN    = OFF        // MII/RMII select (OFF=RMII)

/* Code Protection settings
*/
#pragma config CP       = OFF        // Code protection
#pragma config BWP      = OFF        // Boot flash write protect
#pragma config PWP      = OFF        // Program flash write protect

/* Debug settings
*/
#pragma config ICESEL   = ICS_PGx1  // ICE pin selection

```

Appendix C: Connector and Jumper Block Pinout Tables

Arranged by Microcontroller Pin Number

PIC32 Pin #	Connector Pin	chipKIT Pin #	MCU Port Bit	PIC32 Signal Name	Notes
1	LD4	54	RG15	AERXERR/RG15	LED4
3	JB-08	14	RE05	PMD5/RE5	
4	JB-09	15	RE06	PMD6/RE6	
5	JB-10	16	RE07	PMD7/RE7	
6	JC-01	17	RC01	T2CK/RC1	
7	N/A	N/A	RC02	T3CK/AC2TX/RC2	CAN2
8	N/A	N/A	RC03	T4CK/AC2RX/RC3	CAN2
9	JD-03	27	RC04	T5CK/SDI1/RC4	
10	BTN1	48	RG06	.../SCK2A/U2BTX/U2ARTS/PMA5/CN8/RG6	BTN1
11	BTN2	49	RG07	.../SDA2A/SDI2A/U2ARX/PMA4/CN9/RG7	BTN2
12	N/A	N/A	RG08	.../SCL2A/SDO2A/U2ATX/PMA3/CN10/RG8	Ethernet PHY
14	N/A	N/A	RG09	.../SS2A/U2BRX/U2ACTS/PMA2/CN11/RG9	Ethernet PHY
17	BTN3	50	RA00	TMS/RA0	BTN3
18	JE-07	37	RE08	AERXD0/INT1/RE8	
19	JF-07	45	RE09	AERXD1/INT2/RE9	
20	N/A	59	RB05	AN5/C1IN+/VBUSON/CN7/RB5	USB VBUSON
21	JA-03	3	RB04	AN4/C1IN-/CN6/RB4	
22	JA-02	2	RB03	AN3/C2IN+/CN5/RB3	
23	JA-01	1	RB02	AN2/C2IN-/CN4/RB2	
24	N/A	N/A	RB01	PGEC1/AN1/CN3/RB1	debug PGD
25	N/A	N/A	RB00	PGED1/AN0/CN2/RB0	debug PGC
26	JA-04	4	RB06	PGEC2/AN6/OCFA/RB6	
27	JA-07	5	RB07	PGED2/AN7/RB7	
28	JE-09	39	RA09	Vref-/CVref-/AERXD2/PMA7/RA9	
29	JE-10	40	RA10	Vref+/CVref+/AERXD3/PMA6/RA10	
32	JA-08	6	RB08	AN8/C1OUT/RB8	
33	JA-09	7	RB09	AN9/C2OUT/RB9	
34	JA-10	8	RB10	CVrefout/PMA13/AN10/RB10	
35	N/A	N/A	RB11	AN11/ERXERR/AETXERR/PMA12/RB11	Ethernet PHY
38	JF-08	46	RA01	TCK/RA1	
39	JF-04	44	RF13	AC1TX/SCK3A/U3BTX/U3ARTS/RF13	also CAN1 (JP-2)
40	JF-01	41	RF12	AC1RX/SS3A/U3BRX/U3ACTS/RF12	also CAN1 (JP-1)
41	N/A	N/A	RB12	AN12/ERXD0/AECRS/PMA11/RB12	Ethernet PHY
42	N/A	N/A	RB13	AN13/ERXD1/AECOL/PMA10/RB13	Ethernet PHY
43	JC-10	24	RB14	AN14/ERXD2/AETXD3/PMALH/PMA1/RB14	
44	JC-07	21	RB15	AN15/.../OCFB/PMALL/PMA0/CN12/RB15	
47	JE-01	33	RD14	AETXD0/SS1A/U1BRX/U1ACTS/CN20/RD14	
48	JE-04	36	RD15	AETXD1/SCK1A/U1BTX/U1ARTS/CN21/RD15	

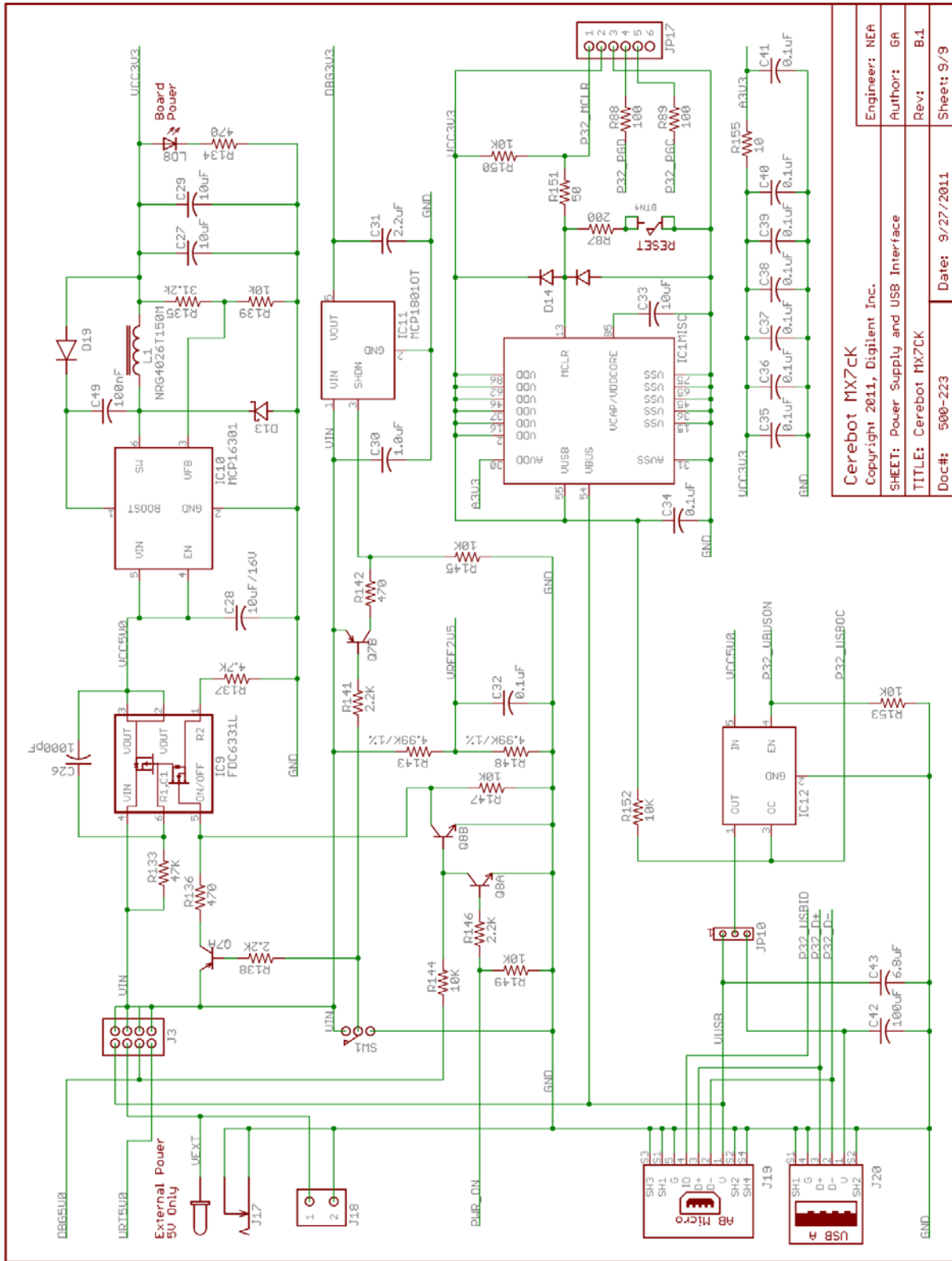
Cerebot MX7cK Reference Manual



49	JF-03	43	RF04	SDA3A/SDI3A/U3ARX/PMA9/CN17/RF4	
50	JF-02	42	RF05	SCL3A/SDO3A/U3ATX/PMA8/CN18/RF5	
51	N/A	N/A	RF03	USBID/RF3	USBID (USB-4)
52	JE-03	35	RF02	SDA1A/SDI1A/U1ARX/RF2	
53	JE-02	34	RF08	SCL1A/SDO1A/U1ATX/RF8	
56	N/A	N/A	RG03	D-/RG3	D- (USB-2)
57	N/A	N/A	RG02	D+/RG2	D+ (USB-3)
58	J7-1,J7-2	55	RA02	SCL2/RA2	I2C2
59	J7-3,J7-4	56	RA03	SDA2/RA3	I2C2
60	JF-09	47	RA04	TDI/RA4	
61	JF-10	48	RA05	TDO/RA5	
63	N/A	N/A	RC12	OSC1/CLKI/RC12	Primary Oscillator
64	N/A	N/A	RC15	OSC2/CLKO/RC15	Primary Oscillator
66	J8-1,J8-2	57	RA14	AETXCLK/SCL1/INT3/RA14	I2C1
67	J8-3,J8-4	58	RA15	AETXEN/SDA1/INT4/RA15	I2C1
68	N/A	N/A	RD08	RTCC/EMDIO/AEMDIO/IC1/RD8	Ethernet PHY
69	JD-01	25	RD09	SS1/IC2/RD9	
70	JD-04	28	RD10	SCK1/IC3/PMCS2/PMA15/RD10	
71	N/A	N/A	RD11	EMDC/AEMDC/IC4/PMCS1/PMA14/RD11	Ethernet PHY
72	JD-02	26	RD00	SDO1/OC1/INT0/RD0	
73	N/A	N/A	RC13	SOSCI/CN1/RC13	Secondary Oscillator
74	N/A	N/A	RC14	SOSCO/T1CK/CN0/RC14	Secondary Oscillator
76	JD-07	29	RD01	OC2/RD1	
77	JD-08	30	RD02	OC3/RD2	
78	JD-09	31	RD03	OC4/RD3	
79	JD-10	32	RD12	ETXD2/IC5/PMD12/RD12	
80	N/A	60	RD13	ETXD3/PMD13/CN19/RD13	USB Overcurrent
81	JC-09	23	RD04	OC5/PMWR/CN13/RD4	
82	JC-08	22	RD05	PMRD/CN14/RD5	
83	N/A	N/A	RD06	ETXEN/PMD14/CN15/RD6	Ethernet PHY
84	JC-04	20	RD07	ETXCLK/PMD15/CN16/RD7	
87	N/A	N/A	RF00	C1RX/ETXD1/PMD11/RF0	Ethernet PHY
88	N/A	N/A	RF01	C1TX/ETXD0/PMD10/RF1	Ethernet PHY
89	JC-03	19	RG01	C2TX/ETXERR/PMD9/RG1	
90	JC-02	18	RG00	C2RX/PMD8/RG0	
91	N/A	N/A	RA06	TRCLK/RA6	Ethernet PHY Reset
92	JE-08	38	RA07	TRD3/RA7	
93	JB-01	9	RE00	PMD0/RE0	
94	JB-02	10	RE01	PMD1/RE1	
95	LD3	53	RG14	TRD2/RG14	LED3
96	LD1	51	RG12	TRD1/RG12	LED1
97	LD2	52	RG13	TRD0/RG13	LED2
98	JB-03	11	RE02	PMD2/RE2	

Cerebot MX7cK Reference Manual

99	JB-04	12	RE03	PMD3/RE3
100	JB-07	13	RE04	PMD4/RE4



Cerebot MX7ck	
Copyright 2011, Digilent Inc.	Engineer: NEA
SHEET: Power Supply and USB Interface	Author: GA
TITLE: Cerebot MX7CK	Rev: B.1
Doc#: 500-223	Date: 9/27/2011
	Sheet: 9/9

OMRON

Switch Mode Power Supply S8VS (15/30/60/90/120/180/240-W Models)

15/30-W Models

Compact, Thin Power Supplies That Mount Just About Anywhere to Contribute to Control Panel Downsizing

- Compact, thin size: 22.5 × 85 × 96.5 mm (W × H × D).
- Three mounting directions (standard, horizontal, facing horizontal).
- Mounting directly onto the panel is possible.
- Safety standards:
UL508/60950-1/1604, CSA C22.2 No. 14/60950-1/213,
EN50178 (= VDE0160), EN60950-1 (= VDE0805).



60/90/120/180/240-W Models

New Models with Total Run Time Monitor in Addition to Models with Maintenance Forecast Monitor

- Compact size: 40 × 95 mm (W × H) (60-W Models).
- Status displayed on 3-digit, 7-segment display.
- Safety standards:
UL508/60950, CSA C22.2 No. 14/60950,
EN50178 (= VDE0160), EN60950 (= VDE0805).



Features Common to All Models

- Mount to DIN Rail.
- Lead-free solder.



Model Number Structure

Model Number Legend

S8VS-

--	--	--	--	--	--

1. Power Ratings

- 015: 15 W
- 030: 30 W
- 060: 60 W
- 090: 90 W
- 120: 120 W
- 180: 180 W
- 240: 240 W

2. Output voltage

- 05: 5 V
- 12: 12 V
- 24: 24 V

3. Configuration

- 15-W, 30-W Models**
None: Standard

60-W Models

- None: Standard
- A: With maintenance forecast monitor
- B: With total run time monitor

- 90-W, 120-W, 180-W, 240-W Models**
None: Standard

- A: With maintenance forecast monitor and undervoltage alarm (transistor (sinking))
- B: With total run time monitor and undervoltage alarm (transistor (sinking))
- AP: With maintenance forecast monitor and undervoltage alarm (transistor (sourcing))
- BP: With total run time monitor and undervoltage alarm (transistor (sourcing))

Ordering Information

Power ratings	Input Voltage	Output voltage	Output current	Alarm output	Model number	
15 W	100 to 240 VAC	5 V	2.0 A	---	S8VS-01505 (See note 1.)	
		12 V	1.2 A	---	S8VS-01512	
		24 V	0.65 A	---	S8VS-01524	
30 W		5 V	4.0 A	---	S8VS-03005 (See note 2.)	
		12 V	2.5 A	---	S8VS-03012	
		24 V	1.3 A	---	S8VS-03024	
60 W		24 V	2.5 A	---	S8VS-06024	
90 W		24 V	2.5 A	3.75 A	---	S8VS-09024
					Sinking	S8VS-09024A
					Sourcing	S8VS-09024AP
					Sinking	S8VS-09024B
120 W		24 V	2.5 A	5 A	Sourcing	S8VS-09024BP
	---				S8VS-12024	
	Sinking				S8VS-12024A	
	Sourcing				S8VS-12024AP	
180 W	24 V	2.5 A	7.5 A	Sinking	S8VS-12024B	
				Sinking	S8VS-12024BP	
				Sourcing	S8VS-18024	
				Sinking	S8VS-18024A	
240 W	24 V	2.5 A	10 A	Sourcing	S8VS-18024AP	
				Sinking	S8VS-18024B	
				Sourcing	S8VS-18024BP	
				Sinking	S8VS-24024	
240 W	24 V	2.5 A	10 A	---	S8VS-24024	
				Sinking	S8VS-24024A	
				Sourcing	S8VS-24024AP	
				Sinking	S8VS-24024B	
240 W	24 V	2.5 A	10 A	Sourcing	S8VS-24024BP	
				Sinking	S8VS-24024BP	

Note: 1. The output capacity of the S8VS-01505 is 10 W.
2. The output capacity of the S8VS-03005 is 20 W.



Specifications

■ Ratings/Characteristics

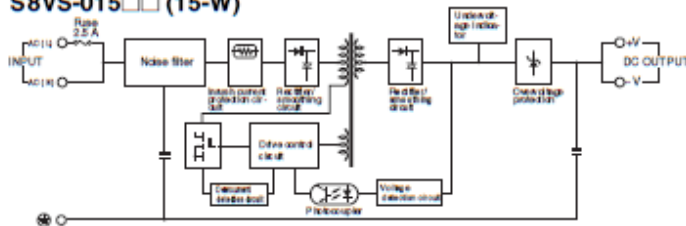
Item	Power ratings		15 W	30 W			
	Type		Standard	Standard			
Efficiency (typical)	5-V models	72% min. (78% typ.)		70% min. (76% typ.)			
	12-V models	74% min. (79% typ.)		76% min. (83% typ.)			
	24-V models	77% min. (81% typ.)		80% min. (85% typ.)			
Input	Voltage	100 to 240 VAC (85 to 264 VAC)					
	Frequency	50/60 Hz (47 to 430 Hz)					
	Current	100 V input	0.48 A max.	0.9 A max.			
		200 V input	0.35 A max.	0.6 A max.			
		230 V input	5 V: (0.14 A typ.), 12 V: 0.19 A typ.)	5 V: (0.27 A typ.), 12 V: 0.37 A typ.)			
	Power factor	---					
	Harmonic current emissions	Conforms to EN61000-3-2					
	Leakage current	100 V input	0.5 mA max.				
		200 V input	1.0 mA max.				
		230 V input	5 V/12 V: 0.30 mA typ.)	5 V/12 V: 0.32 mA typ.)			
Inrush current (See note 1.)	100 V input	25 A max. (20 A typ.) (for a cold start at 25°C)					
	200 V input	50 A max. (40 A typ.) (for a cold start at 25°C)					
	230 V input	5 V/12 V: 24 V: (50 A typ.) (See note 5.)	5 V/12 V: 24 V: (40 A typ.) (See note 5.)				
Output	Voltage adjustment range (See note 2.)	-10% to 15% (with VADJ) (guaranteed)					
	Ripple	2.0% (p-p) max. (at rated input/output voltage)					
		<table border="0"> <tr> <td>to 200 Hz measuring</td> <td>5 V: (0.70% p-p) typ.), 12 V: (0.48% p-p) typ.), 24 V: (0.25% p-p) typ.)</td> <td>5 V: (0.70% p-p) typ.), 12 V: (0.52% p-p) typ.), 24 V: (0.19% p-p) typ.)</td> </tr> <tr> <td>to 100 Hz measuring</td> <td>5 V: (0.85% p-p) typ.), 12 V: (0.58% p-p) typ.), 24 V: (0.32% p-p) typ.)</td> <td>5 V: (0.80% p-p) typ.), 12 V: (0.58% p-p) typ.), 24 V: (0.21% p-p) typ.)</td> </tr> </table>	to 200 Hz measuring	5 V: (0.70% p-p) typ.), 12 V: (0.48% p-p) typ.), 24 V: (0.25% p-p) typ.)	5 V: (0.70% p-p) typ.), 12 V: (0.52% p-p) typ.), 24 V: (0.19% p-p) typ.)	to 100 Hz measuring	5 V: (0.85% p-p) typ.), 12 V: (0.58% p-p) typ.), 24 V: (0.32% p-p) typ.)
	to 200 Hz measuring	5 V: (0.70% p-p) typ.), 12 V: (0.48% p-p) typ.), 24 V: (0.25% p-p) typ.)	5 V: (0.70% p-p) typ.), 12 V: (0.52% p-p) typ.), 24 V: (0.19% p-p) typ.)				
	to 100 Hz measuring	5 V: (0.85% p-p) typ.), 12 V: (0.58% p-p) typ.), 24 V: (0.32% p-p) typ.)	5 V: (0.80% p-p) typ.), 12 V: (0.58% p-p) typ.), 24 V: (0.21% p-p) typ.)				
	Input variation influence	0.8% max. (at 85 to 264 VAC input, 100% load)					
	Load variation influence (rated input voltage)	2.0% max. (5 V), 1.5% max. (12 V, 24 V), (with rated input, 0 to 100% load)					
	Temperature variation influence	0.05%/°C max.					
	Start up time (See note 1 and 7.)	100 ms max. (at rated input/output voltage)					
	Hold time (See note 1.)	20 ms min. (at rated input/output voltage)					
at 100% load		5 V: (328 ms typ.), 12 V: (251 ms typ.), 24 V: (243 ms typ.)					
Additional functions	Overload protection (See note 1.)	105% to 160% of rated load current, voltage drop, automatic reset		105% to 160% of rated load current, voltage drop, intermittent operation, automatic reset			
	Overvoltage protection (See note 1.)	Yes (a zener diode clamp) (See note 3.)		Yes (See note 4.)			
	Output voltage indication	No					
	Output current indication	No					
	Peak-hold current indication	No					
	Maintenance forecast monitor indication	No					
	Maintenance forecast monitor output	No					
	Total runtime monitor indication	No					
	Total runtime monitor output	No					
	Undervoltage alarm indication	Yes (color: red)					
	Undervoltage alarm output	No					
	Parallel operation	No					
	Series operation	Models with 24-V output: Possible for up to 2 Power Supplies (with external diode) Models with 5- or 12-V output: Not possible					
	Other	Operating ambient temperature	Refer to the derating curve in Engineering Data. (with no icing or condensation)				
		Storage temperature	-25 to 85°C				
		Operating ambient humidity	25% to 85% (Storage humidity: 25% to 90%)				
Dielectric strength		3.0 kVAC for 1 min. (between all inputs and outputs; detection current: 20 mA)					
		2.0 kVAC for 1 min. (between all inputs and PE terminals; detection current: 20 mA)					
		1.0 kVAC for 1 min. (between all outputs and PE terminals; detection current: 20 mA)					
Insulation resistance		100 MΩ min. (between all outputs and all inputs/ PE terminals) at 500 VDC					
Vibration resistance		10 to 16 Hz, 0.375-mm single amplitude for 2 h each in X, Y, and Z directions					
Shock resistance		10 to 150 Hz, 0.35-mm single amplitude (5 G max.) for 30 min. each in X, Y, and Z directions					
Output indicator		Yes (color: green)					
EMC		Conducted Emissions	Conforms to EN61004-3 EN55011 Class B and based on FCC Class A				
		Radiated Emissions	Conforms to EN61004-3 EN55011 Class B				
EMS		Conforms to EN61004-3 high severity levels					
Approved standards		UL: UL508 (Listing, Class 2: Per UL1310), UL8060-1, UL1604 (Class (D) Division 2) cUL: CSA C22.2 No. 14 (Class 2), No. 60950-1, No. 213 (Class (D) Division 2) EMVDE: EN55013-1/VD03r02, EN60950-1 (µVDE05) SELV: (EN60950/EN6176/UL 60950-1) According to VDE0106P100, IP20					
Weight	180 g max.		180 g max.				

- Note:**
1. Refer to the Engineering Data section on page 17 for details.
 2. If the VADJ adjuster is turned, the voltage will increase by more than +15% of the voltage adjustment range. When adjusting the output voltage, confirm the actual output voltage from the Power Supply and be sure that the load is not damaged.
 3. The overvoltage protection of the S8VS-015 uses a zener diode clamp. If the internal feedback circuit is destroyed by any chance, the load may be destroyed by the clamped output voltage (up to 150% of the rated output voltage).
 4. To reset the protection, turn OFF the power supply for three minutes or longer and then turn the power supply back ON.
 5. The typical values indicate the values for an input condition of 230 VAC. All items are measured at a frequency of 50 Hz.
 6. The inrush current circuit is not a filter for voltage specifications. Therefore, the typical values are the data values for 24-V models.
 7. The circuit forms are different, so the start up time is shorter only when using a 15-W power rating.

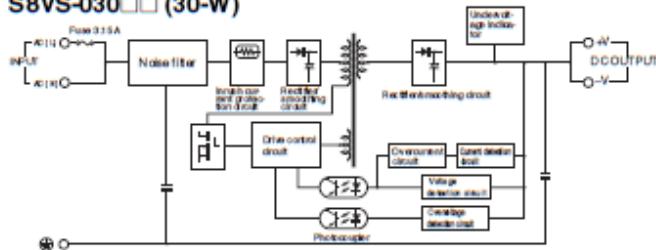
Connections

■ Block Diagrams

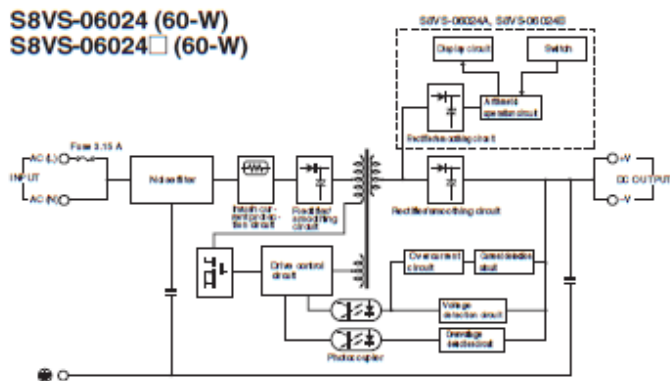
S8VS-015□□ (15-W)



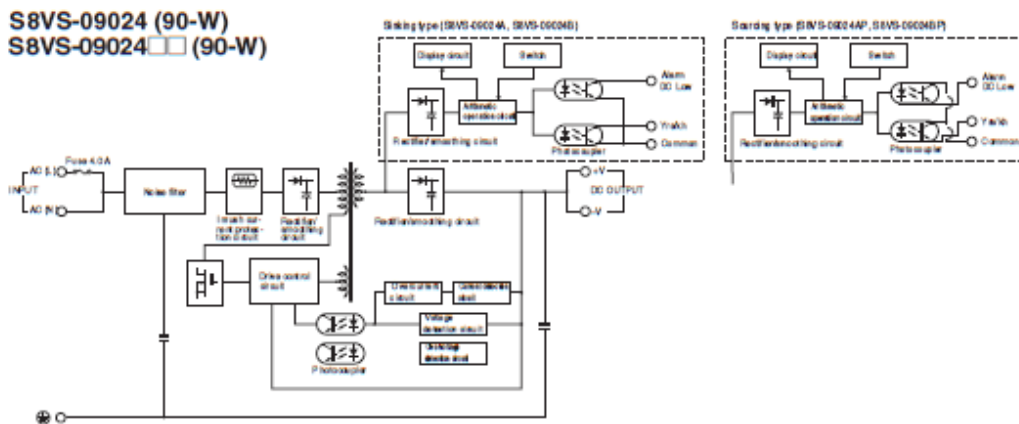
S8VS-030□□ (30-W)



S8VS-06024 (60-W) S8VS-06024□ (60-W)



S8VS-09024 (90-W) S8VS-09024□□ (90-W)



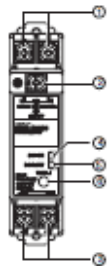


Construction and Nomenclature (15-W, 30-W Models)

■ Nomenclature

15-W, 30-W Models

S8VS-015□□/S8VS-030□□



No.	Name	Function
1	AC Input terminals (L), (N)	Connect the input lines to these terminals. (See note 1.)
2	Protective Earth terminal (PE)	Connect the ground line to this terminal. (See note 2.)
3	DC Output terminals (-V), (+V)	Connect the load lines to these terminals.
4	Output indicator (DC ON: Green)	Lights while a direct current (DC) output is ON.
5	Undervoltage indicator (DC LOW: Red)	Lights when a drop is detected in the output voltage.
6	Output voltage adjuster (V.ADJ)	Use to adjust the voltage.

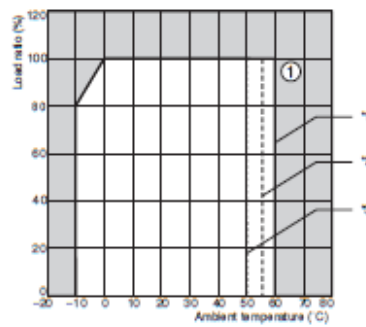
- Note:** 1. The fuse is located on the (L) side. It is NOT user-replaceable.
 2. This is the protective earth terminal specified in the safety standards. Always ground this terminal.

Note: The S8VS-01505 is shown above.

Engineering Data (15-W, 30-W Models)

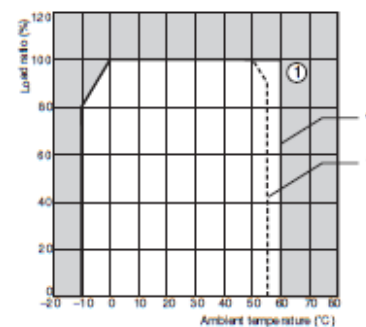
■ Derating Curve

S8VS-015□□



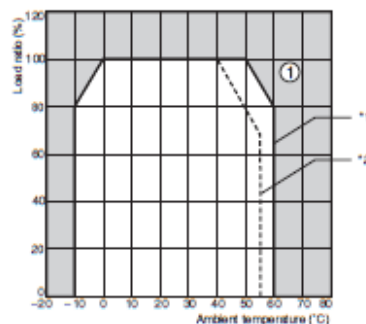
- 1* Standard mounting
 2* Horizontal mounting
 3* Mounting facing horizontally

S8VS-03024



- 1* Standard mounting
 2* Horizontal mounting/mounting facing horizontally

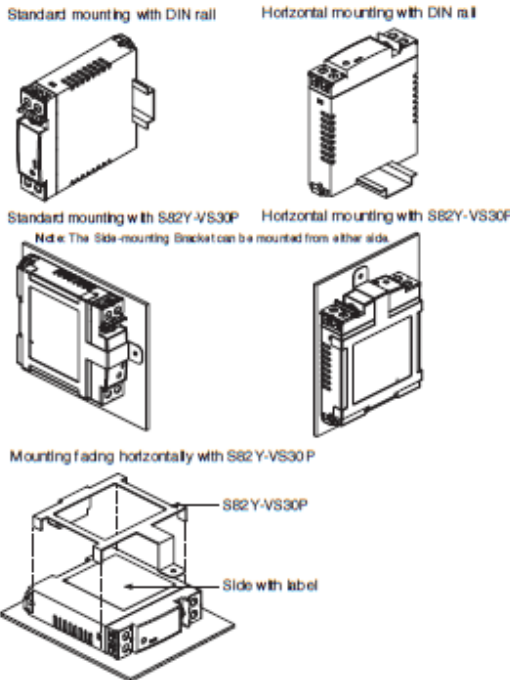
S8VS-03005/S8VS-03012



- 1* Standard mounting
 2* Horizontal mounting/mounting facing horizontally

- Note:** 1. Internal parts may occasionally deteriorate or be damaged. Do not use the Power Supply in areas outside the derating curve (i.e., the area shown by shading ① in the above graph).
 2. If there is a derating problem, use forced air-cooling.
 3. Provide a space of at least 20 mm when using standard mounting and horizontal mounting. If 20mm is not available, make sure that the space is at least 10 mm. In this case, reduce the corresponding derating curve by 5°C.
 4. When mounting Power Supplies facing horizontally in a vertical stack, provide a space of at least 75 mm in between the Power Supplies. If 75 mm is not available, reduce the corresponding derating curve by 1°C for every 5-mm reduction in space. A space of at least 125 mm, however, must be provided. In this case, reduce the corresponding derating curve by 10°C.

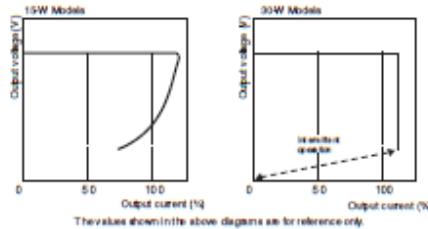
■ Mounting



- Note:**
1. Improper mounting will interfere with heat dissipation and may occasionally result in deterioration or damage of internal parts. Use the product within the derating curve for the mounting direction that is used. Do not use the Power Supply mounted in any way not shown above.
 2. Use a mounting bracket (S82Y-VS30P, sold separately) when the Product is mounted facing horizontally.
 3. Heat dissipation will be adversely affected. When the Product is mounted facing horizontally, always place the side with the label facing upward.
 4. Use PFP-M End Plates on the top and bottom of the Power Supply when mounting facing horizontally on a DIN rail.

■ Overload Protection

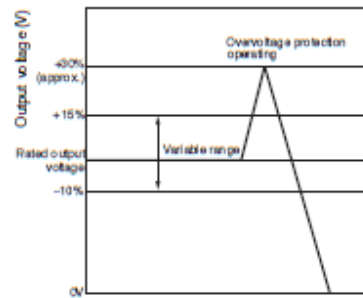
The Power Supply is provided with an overload protection function that protects the power supply from possible damage by overcurrent. When the output current rises above 105% min. of the rated current, the protection function is triggered, decreasing the output voltage. When the output current falls within the rated range, the overload protection function is automatically cleared.



- Note:**
1. Internal parts may occasionally deteriorate or be damaged if a short-circuited or overcurrent state continues during operation.
 2. Internal parts may possibly deteriorate or be damaged if the Power Supply is used for applications with frequent inrush current or overloading at the load end. Do not use the Power Supply for such applications.

■ Overvoltage Protection

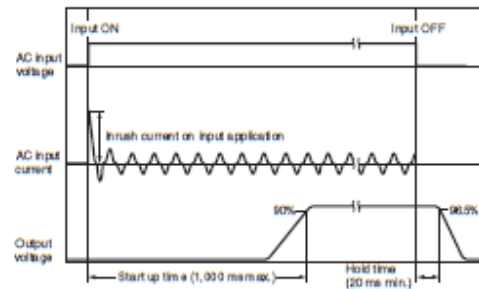
Consider the possibility of an overvoltage and design the system so that the load will not be subjected to an excessive voltage even if the feedback circuit in the Power Supply fails. When an excessive voltage that is approximately 130% of the rated voltage or more is output, the output voltage is shut OFF. Reset the Power Supply by turning it OFF for at least three minutes and then turning it back ON again.



The values shown in the above diagram is for reference only.

- Note:**
1. Do not turn ON the power again until the cause of the overvoltage has been removed.
 2. The overvoltage protection of the S8VS-015□□ uses a zener diode clamp. The output voltage will be clamped at approx. 140% or higher of the rated output voltage (approx. 140% to 190%). If the internal feedback circuit is destroyed by any chance, the load may be destroyed by the clamped output voltage (approx. 140% to 190% of the rated output voltage). The power Supply will not restart if the output is turned OFF by the overvoltage protection operation. If this occurs, replace the Power Supply.

■ Inrush Current, Start Up Time, Output Hold Time



■ Undervoltage Alarm Indication

LED (DC LOW red) lights to warn of output voltage drop. Detection voltage is set to approx. 80% (75 to 90%) of the rated output voltage.

Note: This function monitors the voltage at the power supply output terminals. To check actual voltage, measure voltage on the load side.

■ Reference Values

Item	Value
Reliability (MTBF)	15 W: 610,800 hrs, 30 W: 656,400 hrs
Life expectancy	10 yrs. min.

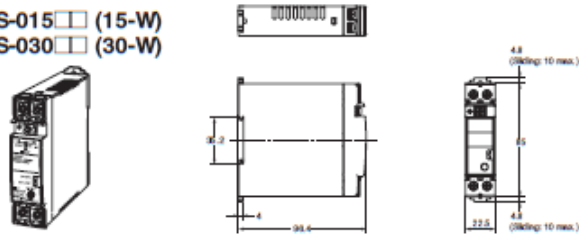
Note: Refer to page 15 for definitions of MTBF and life expectancy.



Dimensions

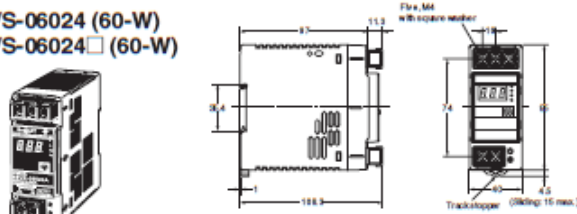
Note: All units are in millimeters unless otherwise indicated.

S8VS-015□□ (15-W)
S8VS-030□□ (30-W)



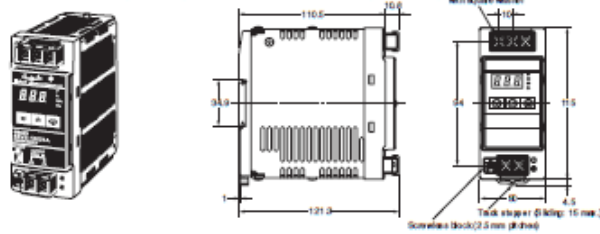
Note: The illustration is the S8VS-03024 Model.

S8VS-06024 (60-W)
S8VS-06024□ (60-W)



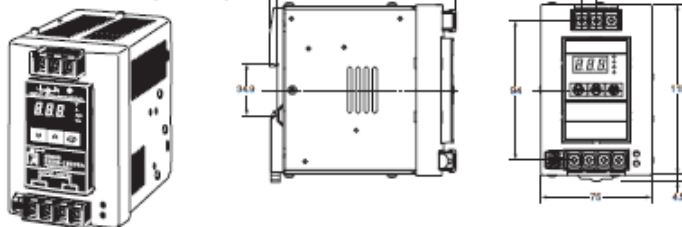
Note: The illustration is the S8VS-06024A Model.

S8VS-09024 (90-W)/S8VS-12024 (120-W)
S8VS-09024□□ (90-W)/S8VS-12024□□ (120-W)



Note: The illustration is the S8VS-12024A Model.

S8VS-18024 (180-W)
S8VS-18024□□ (180-W)





Note: The illustration is the S8VS-18024A Model.





Safety Precautions


CAUTION

Minor electric shock, fire, or Product failure may occasionally occur. Do not disassemble, modify, or repair the Product or touch the interior of the Product. 

Minor burns may occasionally occur. Do not touch the Product while power is being supplied or immediately after power is turned OFF. 

Fire may occasionally occur. Tighten terminal screws to the specified torque (15 and 30 W Models: 0.8 to 1.0 N·m, 60, 90, 120, 180, and 240 W Models: 1.08 N·m). 

Minor injury due to electric shock may occasionally occur. Do not touch the terminals while power is being supplied. Always close the terminal cover after wiring. 

Minor electric shock, fire, or Product failure may occasionally occur. Do not allow any pieces of metal or conductors or any shavings or cuttings resulting from installation work to enter the Product. 

■ Precautions for Safe Use

Mounting

Take adequate measures to ensure proper heat dissipation to increase the long-term reliability of the product. Be sure to allow convection in the atmosphere around devices when mounting. Do not use in locations where the ambient temperature exceeds the range of the derating curve.

When cutting out holes for mounting, make sure that cuttings do not enter the interior of the products.



- *1. Convection of air
 - *2. 20 mm min.
- If 20 mm is not available, however, at least 10 mm must be provided.

(15-W and 30-W Models)

Improper mounting will interfere with heat dissipation and may occasionally result in deterioration or damage of internal parts. Use the product within the derating curve for the mounting direction that is used.

Use a mounting bracket when the product is mounted facing horizontally.

Heat dissipation will be adversely affected. When the product is mounted facing horizontally, always place the side with the label facing upward.

Always provide a space of 20 mm even when mounting horizontally or facing horizontally. If a space of 20 mm is not available, at least 10 mm must be provided. When mounting Power Supplies facing horizontally in a vertical stack, provide a space of at least 75 mm in between the Power Supplies. For details, refer to *Derating Curve* on page 8.

(60-W, 90-W, 120-W, 180-W and 240-W Models)

Improper mounting will interfere with heat dissipation and may occasionally result in deterioration or damage of internal parts. Use the standard mounting method only.

Wiring

Connect the ground completely. A protective earthing terminal stipulated in safety standards is used. Electric shock or malfunction may occur if the ground is not connected completely.

Minor fire may possibly occur. Ensure that input and output terminals are wired correctly.

Do not apply more than 100 N force to the terminal block when tightening it.

Be sure to remove the sheet covering the product for machining before power-ON so that it does not interfere with heat dissipation.

Use the following material for the wires to be connected to the S8VS to prevent smoking or ignition caused by abnormal loads.

Recommended Wire Type

15-W and 30-W Models

Model	Stranded wire	Solid wire
S8VS-03005	AWG18 to 14 (0.9 to 2.0 mm ²)	AWG18 to 16 (0.9 to 1.1 mm ²)
Other models	AWG20 to 14 (0.5 to 2.0 mm ²)	AWG20 to 16 (0.5 to 1.1 mm ²)

60-W, 90-W, 120-W, 180-W and 240-W Models

Model	Recommended wire size	
	For screw terminal	For alarm output terminal
S8VS-06024□	AWG14 to 20 (Cross section 0.517 to 2.081mm ²)	---
S8VS-09024□□ S8VS-12024□□ S8VS-18024□□ S8VS-24024□□	AWG14 to 18 (Cross section 0.823 to 2.081mm ²)	AWG18 to 28 (Cross section 0.081 to 0.823mm ²)

Installation Environment

Do not use the Power Supply in locations subject to shocks or vibrations. In particular, install the Power Supply as far away as possible from contactors or other devices that are a vibration source.

Install the Power Supply well away from any sources of strong, high-frequency noise and surge.

Operating Life

The life of a Power Supply is determined by the life of the electrolytic capacitors used inside. Here, Arrhenius's Law applies, i.e., the life will be cut in half for each rise of 10°C or the life will be doubled for each drop of 10°C. The life of the Power Supply can thus be increased by reducing its internal temperature.

Ambient Operating and Storage Environments

Store the Power Supply at a temperature of -25 to 65°C and a humidity of -25% to 90%.

Do not use the Power Supply in areas outside the derating curve otherwise, internal parts may occasionally deteriorate or be damaged.

Use the Power Supply at a humidity of 25% to 85%.

Technical Details

PNOZmulti modular safety system

Electrical data			
Supply voltage (U_b)	24 VDC	Off time during self test	< 300 μ s
Voltage tolerance	85 ... 120%	Galvanic isolation	Yes
Residual ripple U_b	+/- 5%	Short circuit protection	Yes
Power consumption at U_b without load	Max. 8 W + 2.5 W per expansion module	Switch-off delay	< 30 ms
Times		Residual current at "0"	< 0.5 mA
Switch-on delay	5 s (after U_b is applied)	Signal level at "1"	U_b -0.5 VDC at 2 A
Simultaneity channel 1/2/3	3 s, two-hand control device, 0.5 s	Status display	LED
Supply interruption before de-energisation	Min. 20 ms	Max. capacitive load	2 μ F
Reaction times PNOZ ms1p		Relay outputs	
$f \geq 100$ Hz	10 ms + switch-off delay PNOZ m1p	Number	See unit-specific data
$f < 100$ Hz	1/f + 10 ms + switch-off delay PNOZ m1p	for EN 954-1, 12/96, Cat. 4	See unit-specific data
		for EN 954-1, 12/96, Cat. 2	See unit-specific data
Inputs		Switching capability	
Number	See unit-specific data	in accordance with EN 60947-4-1, 02/01	AC1: 240 V/6 A/1440 VA
Voltage and current	24 VDC/8 mA	in accordance with EN 60947-5-1, 11/97	DC1: 24 V/6 A/144 W
Galvanic isolation	No		AC15: 230 V/3 A/690 VA
Cascading input	500 VAC		DC13: 24 V/3 A/72 W
Signal level at "0"	-3 ... +5 VDC	Contact fuse protection to EN 60947-5-1, 08/00	
Signal level at "1"	15 ... 30 VDC	Blow-out fuse	6 A quick or 6 A slow
Input delay	0.6 ... 4 ms	Circuit breaker 24 VDC	6 A (characteristic C)
Status display	LED	Switch-off delay	50 ms
Pulsed outputs		Status display	LED
Number	See unit-specific data	Auxiliary outputs	
Voltage and current	24 VDC/0.5 A	Number	See unit-specific data
Off time during self test	< 5 ms	Voltage and current	24 VDC, max. 0.5 A
Galvanic isolation	No	External supply voltage (U_e)	24 VDC
Short circuit protection	Yes	Voltage tolerance	85 ... 120 %
Status display	LED	Galvanic isolation	Yes
Semiconductor outputs		Short circuit protection	Yes
Number	See unit-specific data	Residual current at "0"	< 0.5 mA
for EN 954-1, 12/96, Cat. 4		Signal level at "1"	U_b -0.5 VDC at 0.5 A
for EN 954-1, 12/96, Cat. 3		Status display	LED
Switching capability	See unit-specific data		
External supply voltage (U_e)	24 VDC/max. 2 A/max. 48 W		
Voltage tolerance	85 ... 120 %		

pilz
more than automation
safe automation

2.9

2

Status 03/04

Pilz GmbH & Co. KG, Safe Automation, Felix-Wankel-Str. 2, 73760 Ostfildern, Germany, Telephone: +49 711 3409-0, Telefax: +49 711 3409-133, E-Mail: pilz.gmbh@pilz.de

