



UNIVERSIDAD DE JAÉN

Nombre del Centro

SISTEMA AUTÓNOMO DE REGISTRO DE DATOS A MEMORIA USB

Alumno: Jesús Daniel López Rodríguez

Tutor: Prof. D. Luis Miguel Nieto Nieto

Dpto: Ingeniería electrónica y automática



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Electrónica y Automática

Don Luis Miguel Nieto Nieto , tutor del Proyecto Fin de Carrera titulado: Sistema autónomo de registro de datos a memoria USB, que presenta Jesús Daniel López Rodríguez, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, 19 de Junio del 2017

El alumno:

Jesús Daniel López Rodríguez

Los tutores:

Luis Miguel Nieto Nieto

Índice:

1. Introducción.....	6
2. Objetivos	8
3. Antecedentes	10
3.1. Sistemas embebidos	10
3.2. Microcontroladores PIC	12
3.3 PICs de 16 bits	14
3.4 MLA.....	17
4. Herramientas de desarrollo	18
4.1. Placas de desarrollo	19
4.2. MPLAB X IDE V3.51.....	24
4.3. OrCAD Capture	26
4.4. OrCAD Layout	28
5. Diseño de sistema.....	30
5.1. PIC24FJ64B002	33
5.2. LM335	39
5.3. LT1121-5.....	41
5.4. PCB.....	43
5.5. Consumo del sistema	47
5.6. Software	49
5.6.1. Funcionamiento del sistema	51
5.6.2. Máquina de estados	54
5.6.3. Tratamiento de la información	57
Anexo I. Manual de uso del sistema.....	59
Anexo II. Planos	63
Apéndice I. Términos	66
Apéndice II. Referencias de documentos	67
Apéndice III. Referencias de páginas web.....	68
Apéndice IV. Lista de materiales	69

Índice de figuras:

Figura 1. Ejemplo de un sistema embebido.....	10
Figura 2. Microstick II y Microstick plus USB	18
Figura 3. Microstick II	19
Figura 4. Microstick plus USB.....	21
Figura 5. MPLab X IDE y MCC	25
Figura 6. Entorno Orcad Layout	29
Figura 7. Reglas de diseño del sistema.....	45
Figura 8. Cumplimiento de las reglas de diseño	46
Figura 9. Archivo de configuración	59
Figura 10. Archivo de datos tomados generado por el sistema	61
Figura 11. Datos de entrada leídos	62

Índice de diagramas:

Diagrama 1. Resumen del sistema.....	7
Diagrama 2. Familia PIC24FJ128GA310.....	16
Diagrama 3. Microstick II	20
Diagrama 4. Microstick plus USB	22
Diagrama 5. Microstick plus USB conexiones	23
Diagrama 6. Creación del archivo .MAX.....	28
Diagrama 7. Estructura de sistema.....	30
Diagrama 8. Flujo de ejecución	31
Diagrama 9. PIC24FJ64JB002	33
Diagrama 10. Configuración USB HOST	33
Diagrama 11. Comunicación USB	34
Diagrama 12. Funcionamiento módulo RTCC	35
Diagrama 13. Módulo RTCC	36
Diagrama 14. Funcionamiento del módulo ADC	38
Diagrama 15. Esquema de conexiones del LM335.....	39
Diagrama 16. Montaje básico del LM335	39
Diagrama 17. Montaje básico del LT112-X.....	41
Diagrama 18. Esquema de conexiones del LT1121-5	42
Diagrama 19. Esquemático del sistema	43
Diagrama 20. Layout del sistema	44

Diagrama 21. Recursos del sistema.....	49
Diagrama 22. Diagrama de flujo	54
Diagrama 23. Máquina de estados.....	55

Índice de tablas:

Tabla 1. Gama PIC24FJXXGB00X 1.....	14
Tabla 2. Gama PIC24FJXXGB00X 2.....	14
Tabla 3. Prestaciones presentes en las MLA.....	17
Tabla 4. Funciones para tratar ficheros	50

1. Introducción

Se ha realizado el diseño de un sistema embebido que responde a la necesidad de llevar a cabo un registro de datos de temperatura del entorno. De modo que debe de llevarse a cabo un diseño tanto del hardware como del software de un sistema capaz de reunir las siguientes características:

- ❖ Configuración mediante protocolo USB (el hecho de que se necesite la capacidad de escribir hace imprescindible que sea en modo Host).
- ❖ Capacidad para tener conexión con sensores de temperatura por medio de conversión de señales analógicas a digitales.
- ❖ Distintos modos de registro de datos para el posterior volcado a una memoria USB.

Para ello se ha elegido un PIC de 16 bits y placas de desarrollo donde poder probar las distintas versiones preeliminares. Este sistema debe de contar con un sensor de temperatura capaz de captar información por debajo de los -20°C y por encima de 50°C . Esto se debe a la consideración de que pueda funcionar dentro de sistemas de refrigeración o al aire libre en climas cálidos.

El diseño de un sistema de estas características también exige que su uso por sea sencillo y a ser posible intuitivo. Por lo tanto los indicadores luminosos con colores que se suelen asociar verde, correcto y rojo, incorrecto son una opción a tener en cuenta. Igualmente solo los accionadores externos necesarios que permitan llevar a cabo las operaciones. De modo que se va a prescindir de combinaciones de botones para llevar a cabo cualquier proceso.

Se tiene en cuenta que pueda tener una configuración versátil que permita al sistema operar en distintos ámbitos cambiando solamente los parámetros del archivo que se introduce en la inicialización. De modo que aunque el sistema en sí esté dedicado al registro y validación de muestras de temperatura pueda destinarse al control de procesos

Dicha inicialización debe permitir configurar los parámetros:

- ❖ Tiempo de muestreo
- ❖ Fecha y hora
- ❖ Limite superior e inferior de temperatura aceptable

En el manual de uso del sistema deben de especificarse las unidades en las que se introducen los datos y el formato de los mismos. Teniendo a disposición un ejemplo de configuración y con el formato correcto para iniciar el sistema. Además de los pasos a seguir con el hardware del mismo para su adecuado funcionamiento, como por ejemplo la forma de hacer que muestre si se ha registrado algún dato incorrecto o habilitar que la memoria USB se desconecte sin dañar el archivo que contiene los datos almacenados.

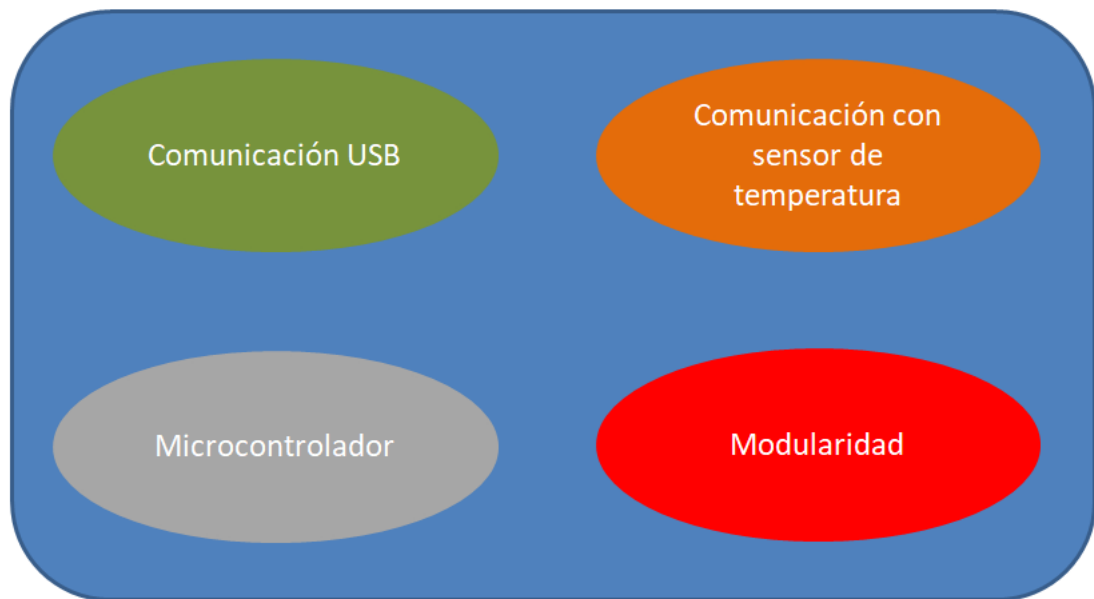


Diagrama 1. Resumen del sistema

2. Objetivos

El objetivo principal es la adquisición de parámetros de temperatura, a la par que se realiza un volcado de los mismos en un dispositivo de almacenamiento externo. Pudiendo el conjunto del sistema operar de forma autónoma durante un tiempo determinado. Además de mostrar si así se desea si alguno de los datos anteriormente adquiridos excede, por exceso o defecto, los límites especificados en la configuración.

Para cumplir con lo anteriormente mencionado debe de cumplirse por medio de un diseño de hardware y software:

- ❖ Registro de eventos con marca de tiempo y volcado a memoria USB.
- ❖ Modo de registro síncrono y asíncrono, programable por agenda o mediante eventos de distintos tipos.
- ❖ Intervalo mínimo de muestreo de 1 segundo.
- ❖ Alimentación autónoma, con capacidad para al menos 1 mes.
- ❖ Interfaz USB para conexión de memoria externa.
- ❖ Puerto de expansión para conexión de sistemas auxiliares de configuración periféricos que amplíen la aplicación de manera modular.
- ❖ Protocolo de comunicación con un sensor de temperatura para adquisición y conversión de la señal en el dispositivo.
- ❖ Capacidad para introducir distintas configuraciones en el dispositivo que alteren el tiempo de muestreo y los rangos de las medidas que se consideran aceptables. Además de los datos de fecha y hora con los que se inicia el sistema.

Dado que el control y seguimiento a lo largo del tiempo de la evolución de la temperatura es una necesidad en diversas tareas este sistema es de gran utilidad en más de un escenario. Se han tenido en mente dos entornos posibles:

- ❖ La necesidad de tener control de la temperatura en el transporte de congelados y la verificación inmediata de que la cadena de frío no se ha roto. Para ello este sistema es perfectamente válido ya que los indicadores luminosos permiten verificar rápidamente que no ha ocurrido ningún evento que comprometa la calidad del envío. De forma análoga se podría trabajar con otros alimentos como fruta, donde es vital que las condiciones en el transporte se preserven para asegurar el buen estado en el que se encuentra a la recepción.
- ❖ Un uso de interés es para el registro de datos de temperatura en el estudio de cuevas y grutas, permitiendo conocer los cambios en la temperatura de las corrientes de aire. Lo cual a posteriori servirá junto con otros conjuntos de datos que no competen a este sistema para estudios de carácter geológico.

En ambos casos el hecho de que sea un sistema autónomo donde los datos tomados se almacenan con su fecha y hora correspondientes además de la validez de los mismos con respecto a los parámetros de inicialización suministrados es de enorme utilidad y fácil uso. Además el hecho de que pueda mostrarse de inmediato el estado del seguimiento facilita la identificación inmediata del estado.

Por otra parte se ha tenido en cuenta la posibilidad de que la persona que manipule el dispositivo tenga algún problema visual como puede ser el daltonismo acromático (distinción de colores en blanco y negro) y el daltonismo tricromático anómalo (distinción de colores pero intercambiando colores), de modo que se emplean dos indicadores luminosos en lugar de uno. Con esto se consigue que a pesar de que la persona no sea capaz de distinguir el color que luce sepa por la ubicación si es el que corresponde al rojo o al verde.

3. Antecedentes

Diseño de un sistema de registro de datos o datalogger, es un dispositivo electrónico encargado de obtener parámetros del entorno por medio de sensores que pueden ser o no propios. Esta información suele almacenarse para usarse posteriormente con el objetivo de realizar un seguimiento o estudio.

3.1. Sistemas embebidos

Suele definirse sistema embebido como “Sistema computacional cuyo hardware y software están diseñados específicamente para realizar la labor de operación, control e instrumentación definida de una manera eficiente” (Heath, Steve 2003). Estos sistemas se alejan de los PC que permiten un amplio rango de tareas para cubrir de forma eficiente y económica una o varias tareas bien definidas.

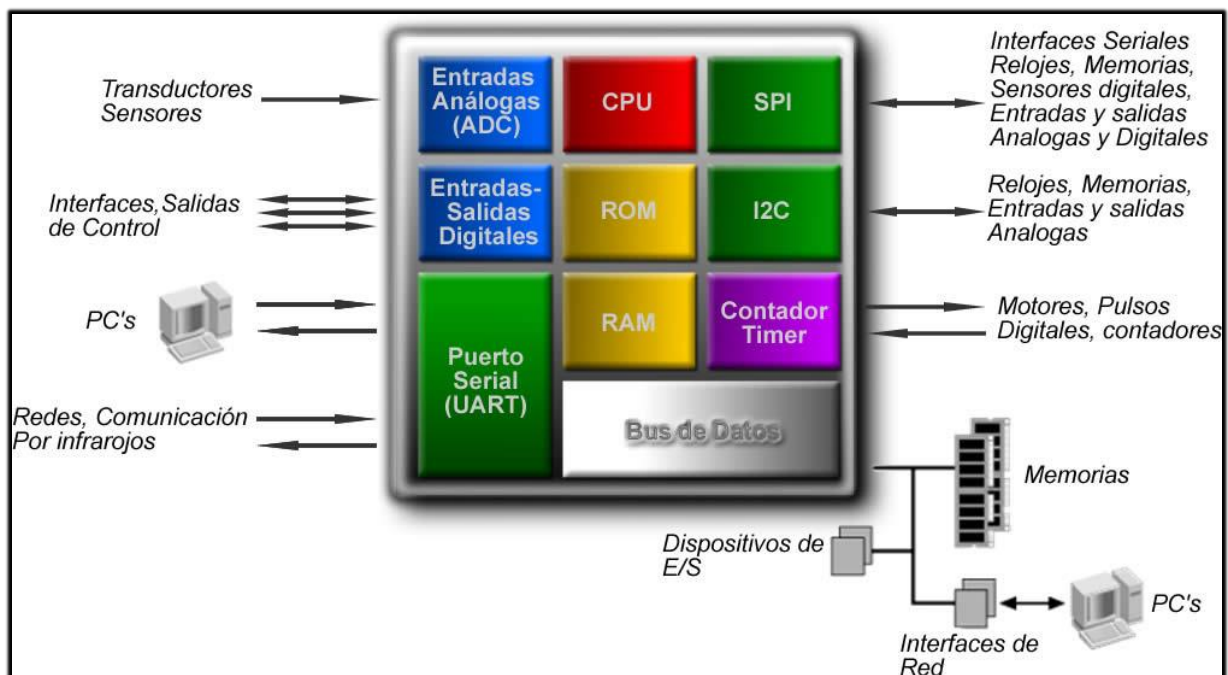


Figura 1. Ejemplo de un sistema embebido

Por lo tanto los sistemas embebidos son diseñados para realizar tareas específicas y este modo satisfacer una necesidad concreta. Suelen emplearse procesadores relativamente pequeños. Existen diversas plataformas que permiten el desarrollo de aplicaciones y prototipos como son: Arduino, Raspberry Pi, Beaglebone o Microchip.

Elementos de un sistema embebido:

- ❖ Principalmente un sistema embebido consta de una unidad de procesamiento como un microprocesador.
- ❖ Habitualmente se tiene además la posibilidad de comunicación por medio de: RS232, I2C, LIN, USB, Wifi, etc...
- ❖ Módulo E/S para la digitalización de señales analógicas además de servir para activación de sistemas externos.
- ❖ Módulo de reloj que genera diferentes pulsos a partir de un oscilador. Debe de cumplir una serie de requisitos como ser capaz de proporcionar una señal estable, tener un consumo adecuado de energía y capacidad para suministrar las señales necesarias.
- ❖ Actuadores en las salidas del sistema para poder ejercer funciones de control.
- ❖ Módulos de conversión ac/dc, dc/ac y filtros entre otros.

3.2. Microcontroladores PIC

Son fabricados por Microchip Technology Inc. y pertenecen a una familia de microcontroladores tipo RISC (Reduced Instruction Set Computing). Derivan del PIC1650 desarrollado por General Instrument. Por medio de un sistema de programación se transfiere el código.

Permiten cubrir una amplia variedad de necesidades. Son sistemas con una arquitectura robusta y flexible con herramientas de desarrollo. Además tienen un bajo coste con funcionalidades como comunicación: I2C, LIN, UART, USB, CAN y Ethernet entre otras.

Para la transferencia del código se emplea circuito llamado programador para volcar directamente el código en el PIC. Algunos de ellos son:

- ❖ PICStart Plus (para aplicaciones de puerto serie y USB)
- ❖ ICD2 (para aplicaciones de puerto serie y USB)
- ❖ Microstick II (para aplicaciones puerto serie y USB)
- ❖ PICKit 3 (solamente para aplicaciones de USB)

En el sistema final el código se carga y aunque es posible que puedan configurarse parámetros del mismo el programa no puede cambiarse, de modo que responde a unas funcionalidades específicas y limitadas. Sin embargo durante el desarrollo del mismo es necesario ir probando y realizando distintas versiones, las cuales se van cargando en el sistema de pruebas, para ello se pueden cargar distintos programas y realizar distintos montajes en estos sistemas de desarrollo.

La arquitectura PIC se considera minimalista y cuenta con estas prestaciones:

- ❖ El área de código y datos están separadas (Arquitectura Harvard).
- ❖ Número de instrucciones reducidas.
- ❖ Un solo acumulador de uso implícito.

- ❖ Una sola pila de hardware para almacenar instrucciones de regreso de funciones.
- ❖ Habitualmente unos 256 bytes de espacio de datos direccionables.
- ❖ Las operaciones duran un tiempo de instrucción, aunque algunas es posible que requieran de varios.
- ❖ No distingue de espacios de memoria y espacios de registros ya que son los mismos para el PIC.

Para desarrollar el software existen diversas herramientas como MPLab X IDE y compiladores como XC16 (para 16 bits) o XC8 (para 8 bits). Además de simuladores como: Proteus, Crocodile Clips, PIC Simulator, etc...

Los simuladores como “Proteus” ofrecen la posibilidad de probar las aproximaciones del sistema si necesidad de realizar montajes o adquirir material. De modo que se convierte en una solución económica a la par que eficaz. Sin embargo y por la propia naturaleza del sistema, de ser funcional y operativo, tienen un uso limitado y llegado a cierto punto hay que pasar a realizar pruebas sobre sistemas hardware que permitan ver la respuesta real.

3.3 PICs de 16 bits

Los microcontroladores de 16 bits ofrecen soluciones más potentes que los de 8 bits pero más económicas que los de 32 bits. Entre otras características toda la gama de 16 bits de Microchip posee un acumulador de 40 bits y la capacidad de funcionar en un modo de bajo consumo.

Product Family	Program Flash Memory (KB)	Pin Count	Intelligent Analog					Waveform Control					Timing and Measurements			Safety and Monitoring									
			ADC (resolution) ¹	DAC (resolution) ²	CVREF	HS Comp	OPA	CCP/ECCP	SCCP	MCCP	PWM	MC PWM	SMPS PWM	IC and OC	PWM Resolution (ns)	8-bit Timer	16-bit Timer	32-bit Timer	RTCC	QEI	LVD	WDT	DMT	CRC	Class B Safety ³
PIC24 Family																									
PIC24FJXXGB00X	32–64	28–44	10			✓				✓			✓	15		✓	✓			✓	✓		✓	L5	

Tabla 1. Gama PIC24FJXXGB00X 1

Product Family	Program Flash Memory (KB)	Pin Count	Communications									User Interface		Secure Data		System Flexibility											
			USB	CAN	UART	LIN	IrDA*	I ² C	SPI	I ² S™	SENT	Parallel Port	CTMU and mTouch* Sensing	LCD (Segments)	GFX	Cryptographic Engine	Secure Key Storage	RNG	Dual Partition Flash	CLC	PPS	PTG	DMA	IDLE, SLEEP and PMD	DOZE	XLP	V _{BAT}
PIC24 Family																											
PIC24FJXXGB00X	32–64	28–44	✓		✓	✓	✓	✓	✓		✓	✓								✓			✓	✓	✓		

Tabla 2. Gama PIC24FJXXGB00X 2

- ❖ Se permite la compatibilidad de gran parte del código entre PIC de la misma gama, dando lugar a la posibilidad de un diseño inicial en un modelo pero pudiendo saltar a otro en caso de que las características del segundo cubran peculiaridades de las que carece el primero.
- ❖ El tamaño reducido de sus encapsulados los hace adecuados para el desarrollo de sistemas compactos. Además de que dentro de la misma familia existen encapsulados capaces de cubrir todas las necesidades, habiendo desde 14 a 144 pines, con un tamaño de 5x5mm.

- ❖ La alimentación responde a los estándares de entre de 3V y 5V con posibilidad de consumo en modo Sleep de 10nA.
- ❖ Facilidad para comunicación inalámbrica o cableada, ya sea por medio de módulos Bluetooth, Wifi, Lin, I2C, entre otros. Existe librerías de software libre disponibles para todos ellos.
- ❖ También los módulos ds33 están especialmente diseñados para el control de motores eléctricos. Además cuenta con placas de desarrollo específicas para estos propósitos.
- ❖ Robustos y resistentes al ruido. Además son capaces de funcionar hasta a unos 150°C de temperatura ambiente. Con el objetivo de ser soluciones validas en ambientes industriales o con automóviles. Para aplicaciones donde la seguridad es un aspecto crítico los modelos incluyen un sistema de detección de errores y comprobación de integridad.
- ❖ Se dan recursos para la estimación dela batería con la aplicación libre de “XLP Battery Life Estimator”. Ideal para aplicaciones de bajo consumo. Es capaz de realizar un modelo del consumo en condiciones normales de funcionamiento del y de bajo consumo para poder dimensionar el sistema de alimentación necesario según las exigencias del proyecto.
- ❖ Para las aplicaciones que van a hacer uso de internet se permite el uso de un sistema en encriptación, es un módulo interno a los PIC que pertenecen a la gama GB2 y GB4 de 16 bits. Usan claves generadas a partir de números aleatorios para proteger las aplicaciones.

Las aplicaciones típicas para encriptación suelen ser el acceso a sistemas de control, impresoras, accesorios de fitness, cámaras de seguridad, etc...

- ❖ Hay soporte para el desarrollo de aplicaciones que muestren la información por medio de pantallas LCD e incluso pantallas táctiles. Los PIC24 de las familias GA3, GB4 y GC cuentan con drivers específicos para estas aplicaciones. Sin embargo es posible el desarrollo en el resto de familias de 16 bits.

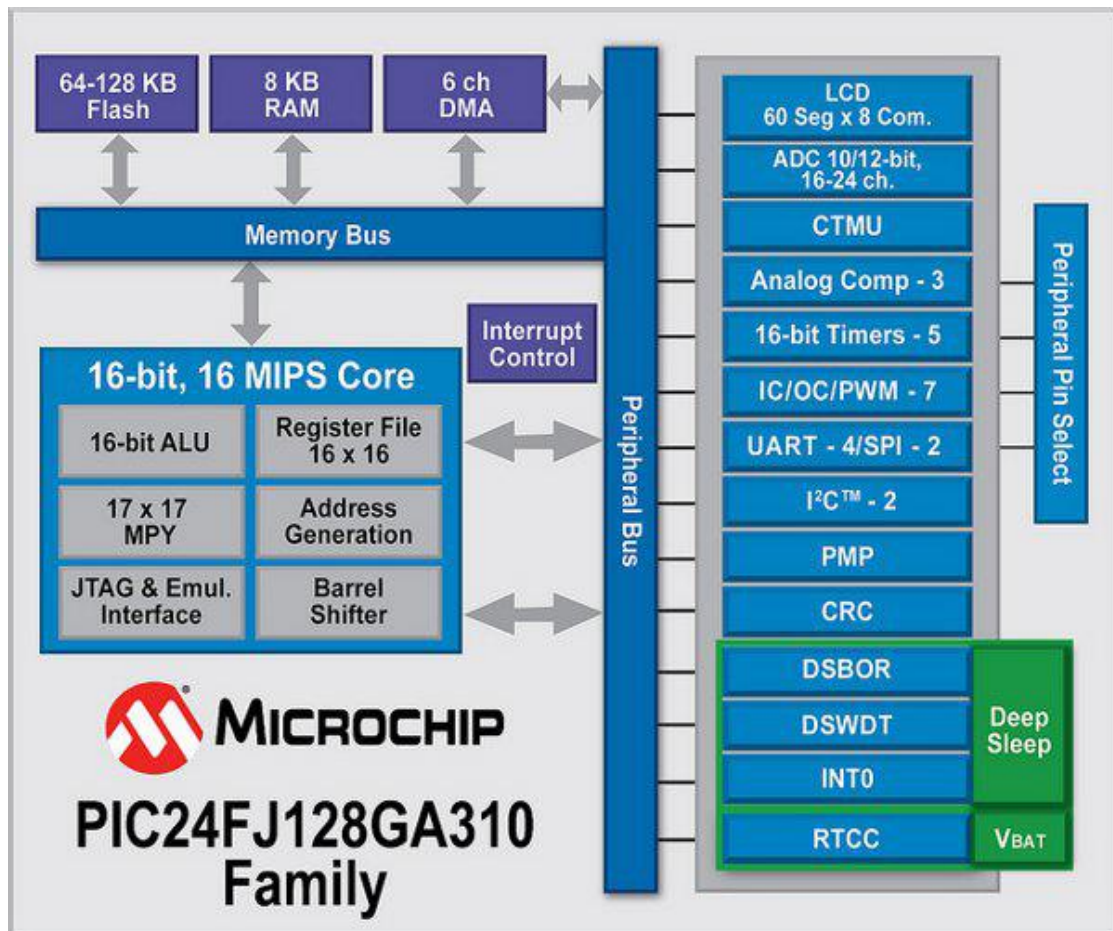


Diagrama 2. Familia PIC24FJ128GA310

Los encapsulados disponibles suelen ser de 28 o 44 pines, siendo de una geometría rectangular los primeros y cuadrada los segundos. Pudiendo encontrarse ambas opciones con los pines para soldadura superficial o atravesando la placa. Están disponibles en QFN, SOIC, SSOP y SPDIP, y QFN and TQFP respectivamente.

3.4 MLA

Las MLA (Microchip Libraries for Applications) son un recurso ofrecido por Microchip para sus PIC, el cual consiste en ejemplos de aplicaciones desarrolladas para una gran variedad de microcontroladores distintos. Su objetivo es que sirvan como ejemplo o base en el desarrollo de aplicaciones posteriores.

Los proyectos están hechos para MPLAB X IDE y MPLAB IDE, y pueden ser compilados por todos los compiladores XC. Además para aplicaciones de 32 bits están disponibles otras versiones de las MLA pero contando con la herramienta software de desarrollo Harmony.

Además de ejemplos que hacen usos de varias librerías distintas y resuelven aplicaciones complejas las MLA incluyen además paquetes de drivers, documentación y utilidades.

Library	Current Version	PIC16F	PIC18F	PIC24 & dsPIC33	PIC32
		(8-bit)	(8-bit)	(16-bit)	(32-bit)
USB	2.17	✓	✓	✓	PIC32MM
Graphics	4.80			✓	
File I/O (Memory Disk Drive)	1.04			✓	
Crypto Software (AES, TDES, RSA, ...)	2.00.02b			✓	
Crypto 16bv1 (Hardware) for PIC24F	0.01.01b			✓	
Crypto Hash (MD5, SHA-1, SHA-2, ...)	1.00			✓	
Smart Card	2.03			✓	
MiWi™ Protocol	5.30		✓	✓	
TCPIP (with Wi-Fi® support only)	5.47		✓	✓	✓
Smartphone	1.04.02			✓	

Tabla 3. Prestaciones presentes en las MLA

También proporcionan archivos explicativos para la comprensión y aplicación de todo el material que contienen. No solo de las aplicaciones y drivers sino de posibles usos para las que han sido pensadas.

4. Herramientas de desarrollo

En apartados sucesivos se expondrán de forma detallada los elementos empleados que hacen posible la realización del sistema. Entre ellos se encuentra una placa de desarrollo Microstick II y otra Microstick plus USB, perteneciente a Microchip. Además del microcontrolador PIC24FJ64GB002 de 16 bits y el sensor DS18B20 que sirve para la toma de medidas de temperatura.

Las placas de desarrollo responden a la necesidad de tener un sistema en el que cargar las distintas versiones del programa en el microcontrolador. No solo para comprobar el funcionamiento individual de los periféricos que van a componer el sistema final sino para disponer de un medio en el que poder depurar los posibles errores. Por lo tanto debe de ser robusto y flexible para proporcionar los elementos necesarios en el desarrollo de sistemas.

La placa Microstick II permite la conexión al PC, mientras que la Microstick plus USB permite el funcionamiento del USB en modo HOST. Aunque están conectadas entre sí, eso no elimina la necesidad de alimentación de cada una de las placas. Sin embargo ambas pueden funcionar de forma independiente la una de la otra o incluso montarse junto con otros modelos compatibles.

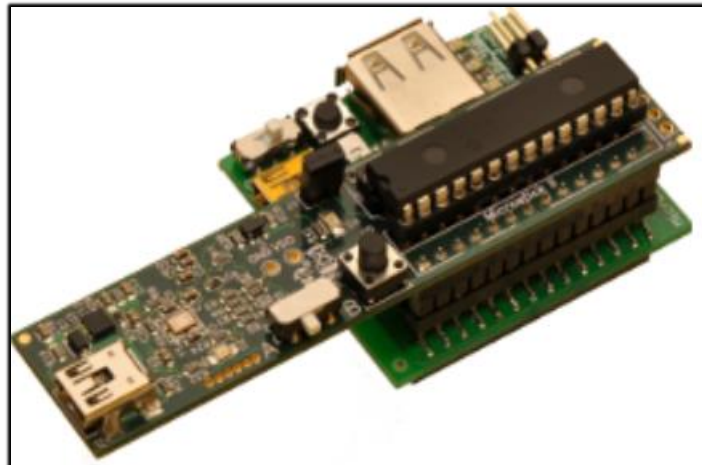


Figura 2. Microstick II y Microstick plus USB

4.1. Placas de desarrollo

Permiten principalmente pasar el código desde un equipo hasta el microcontrolador. Además muchos de estos sistemas sirven para poder desarrollar aplicaciones específicas como Ethernet, pantalla táctil, USB o similares. Aparte de ser un entorno de pruebas seguro desde el que trabajar.

❖ Microstick II:

El sistema de desarrollo se caracteriza por:

- ❖ Precio reducido.
- ❖ Programador USB integrado.
- ❖ Alimentación por medio del USB, no requiere de fuentes externas de alimentación.
- ❖ Tienen soporte con MPLAB.
- ❖ Sencillo intercambio de microcontrolador.
- ❖ Fácil acceso a las señales del dispositivo.
- ❖ Tamaño reducido de 20x76mm, fácil transporte.
- ❖ Código demo gratuito.

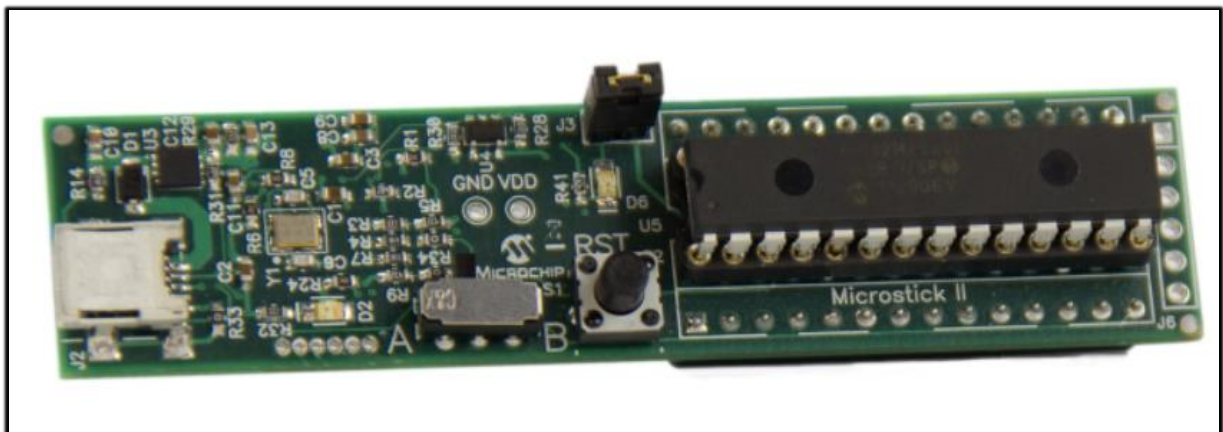


Figura 3. Microstick II

Es una solución económica de uso sencillo para el desarrollo de aplicaciones en 16 y 32 bits. Dentro del proyecto de MPLAB debe de elegirse la placa de desarrollo para que se cargue el código en la misma. Después de cargar el código debe de pulsarse durante un par de segundo el botón de reset de la placa con el fin de tener una correcta inicialización. Algunos de los indicadores luminosos integrados en el sistema pueden ser programados al estar asociados a pines, dando lugar a la posibilidad de emplearlos durante el desarrollo del sistema. Se debe de tener en cuenta que estas mismas conexiones pueden ser origen de errores si se va a usar ese pin como medio para alguna comunicación, de modo que en el caso de que sea posible se debe de abrir por medio del jumper o si no es posible elegir otro pin para la comunicación.

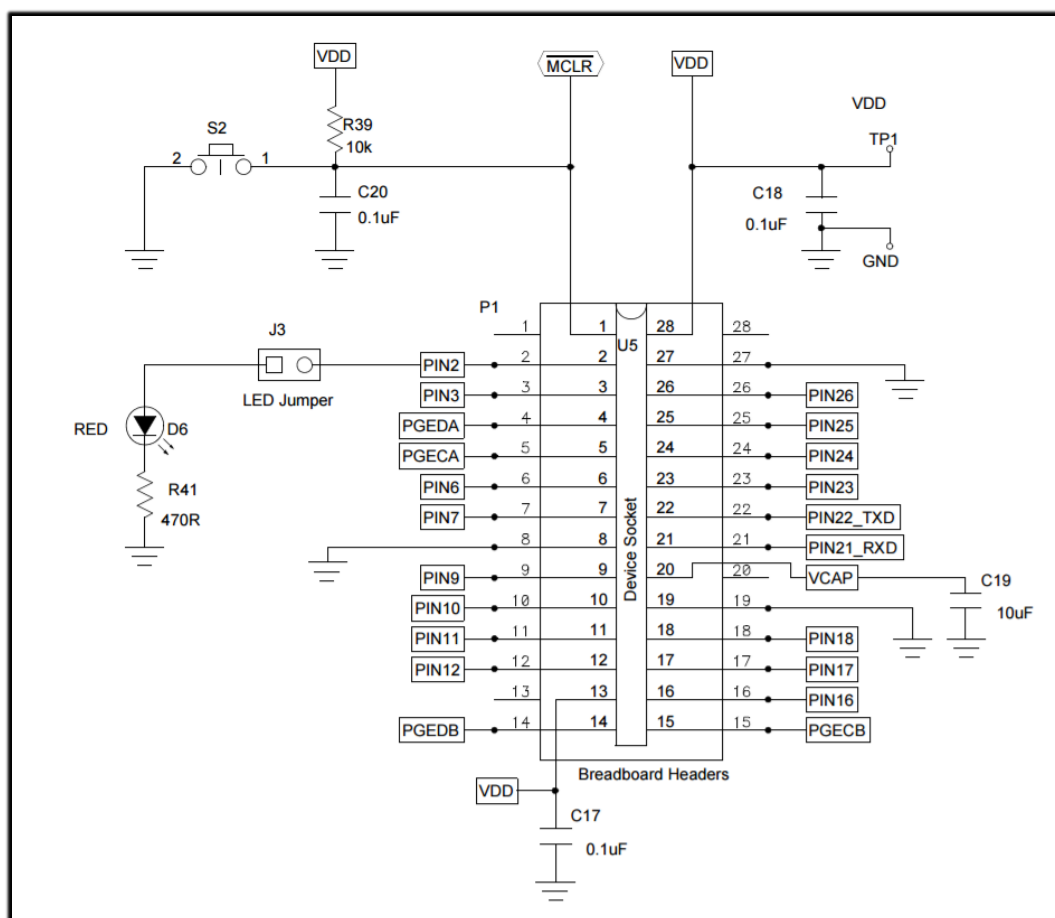


Diagrama 3. Microstick II

Contenido de la placa de desarrollo:

- ❖ Placa Microstick II
- ❖ Cable USB
- ❖ 1 microcontrolador PIC24FJ64GB002
- ❖ 1 microcontrolador PIC24HJ128GP502
- ❖ 1 microcontrolador DsPIC33FJ128MC802
- ❖ 1 microcontrolador PIC32MX250F128B
- ❖ Microstick plus USB:

Es una placa de desarrollo que puede acoplarse a la placa Microstick II. Se compone principalmente de 3 conectores distintos de USB: USB Host A, USB Mini B y USB Micro AB. Además cuenta con leds de estado y botones para la configuración.

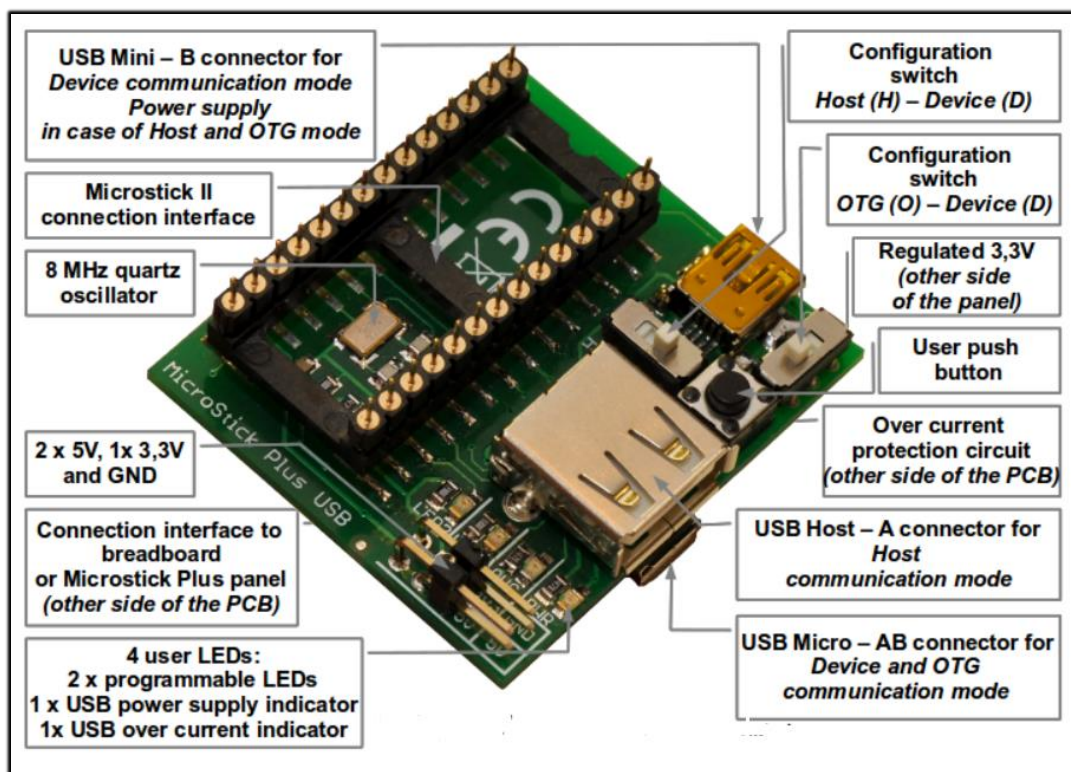


Figura 4. Microstick plus USB

Está pensada principalmente para el desarrollo de aplicaciones USB para microcontroladores de la familia PIC24F y de la familia PIC32MX2. Permite la configuración para la comunicación como USB device, USB Host y USB on the go (es un modo que permite la alternancia entre el modo device y el modo host). Aparte posee una serie de protecciones para poder probar aplicaciones USB de forma segura, incluso en condiciones extremas.

Permite interconectarse con otras placas de desarrollo como la Microstick II y la Microstick plus, siendo su principal limitación para conectarla con otras distintas de estos modelos que el encapsulado de PIC posea una geometría distinta. También el añadido sucesivo de placas de desarrollo limita la capacidad para añadir elementos externos ya que los pines quedan en su mayoría ocupados o simplemente conectados a elementos que generarían errores si se intentan dedicar a otros usos.

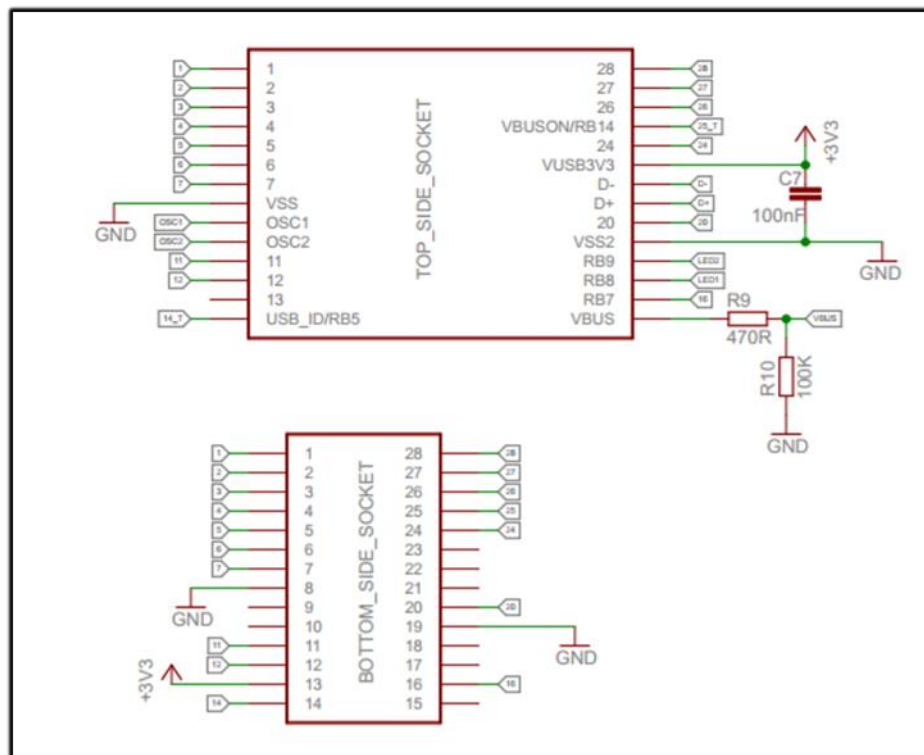


Diagrama 4. Microstick plus USB

Contenido de la placa de desarrollo:

- ❖ Conector USB Mini B para comunicación como device.
- ❖ Conector USB A para comunicación como Host.
- ❖ Conector USB Micro AB para la comunicación como OTG (On-The-Go).
- ❖ Un cristal de cuarzo de 8MHz.
- ❖ Protección para la comunicación por USB.
- ❖ Dos switch de configuración.
- ❖ Dos indicadores de configuración led.
- ❖ Un pulsador de botón.

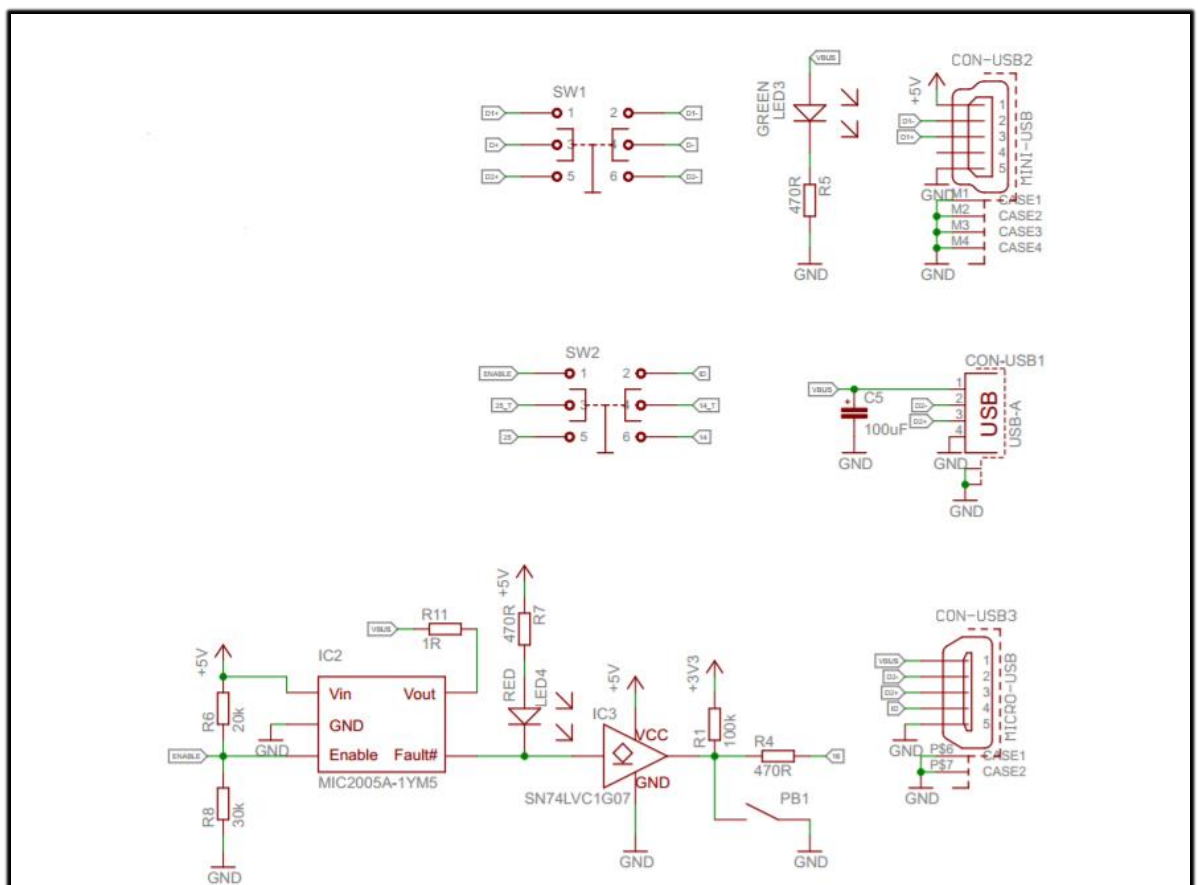


Diagrama 5. Microstick plus USB conexiones

4.2. MPLAB X IDE V3.51

Este entorno de desarrollo se emplea para el desarrollo de la aplicación que debe ir en el PIC. Además permite trabajar con placas de desarrollo y tiene un compilador proporcionado por Microchip para microcontroladores de 16 bits, el XC16.

Además es compatible con otros compiladores y permite la simulación del código seleccionando el modelo de PIC sobre el que se va a trabajar. El entorno permite la creación de proyectos desde cero además de una biblioteca de ejemplos que funcionan sobre una variedad amplia de placas de desarrollo y microcontroladores.

Cuenta con un simulador que permite probar sin necesidad de placa de desarrollo el funcionamiento de la máquina de estados además de otros procesos. Sin embargo el simulador es incapaz de simular comunicación USB, módulo RTCC y otros procesos. De manera que su utilidad se ve limitada una vez se avanza lo suficiente en el desarrollo.

Los proyectos generados en este entorno tienen la ventaja de que pueden guardar distintas configuraciones para microcontroladores distintos, aunque se define uno como estándar. Además permite que dentro de la configuración de ese microcontrolador se especifique el compilador y si se va a cargar en una placa de desarrollo o por el contrario va a realizarse una simulación.

MPLAB X IDE permite de forma rápida conocer cuando se compila el código el origen de los errores en el caso de que estos se produzcan. Generará avisos (warnings) en caso de que no se utilicen variables creadas o se estén llevando a cabo operaciones que el entorno crea que pueden generar fallos, en cambio sí hay errores críticos no lleva a finalizar la compilación. Pero si da de forma detallada un informe de que ha dado lugar a esos fallos, permitiendo identificar y corregir de manera eficiente los fallos.

Al cargar el código tanto en la simulación como en una placa de desarrollo pueden fijarse puntos de ruptura o pausar para observar como procede el programa. Además se cuenta con una herramienta para observar variables y poder ir viendo su evolución. Otra herramienta para depurar código disponible es "*stimulus*" que da la posibilidad de dar entradas lógicas a los pines del micro tanto en simulación como en placa de desarrollo con el fin de probar el código.

Además de las herramientas que trae de base el entorno pueden añadirse otras para añadir funcionalidades como poder comunicar el entorno con proteus o añadir opciones de simulación como la comunicación por USB. Opciones que inicialmente no estaban disponibles.

Partiendo de la herramienta anterior se emplea otra integrada dentro del mismo entorno, MCC (Microchip Code Configurator) que permite una rápida configuración de PIC, además de permitir una base de ciertos protocolos de comunicaciones sobre la que trabajar.

Además de la configuración de los pines del dispositivo, da acceso a la configuración necesaria para que el oscilador permita a comunicación por medio de protocolo USB en modo Host, el funcionamiento del módulo RTCC (Real Time Clock Calendar) del PIC24FJ64GB002 y el módulo ADC para la conversión analógico/digital.

Permite la configuración rápida de pines y la asignación de módulos como el ADC por medio de una interfaz gráfica, además del acceso a los registros de los mismos. Proporcionando además avisos en caso de que alguna de las opciones elegidas no sea viable o que haya que tener en cuenta algún otro requisito tras la generación de los archivos.

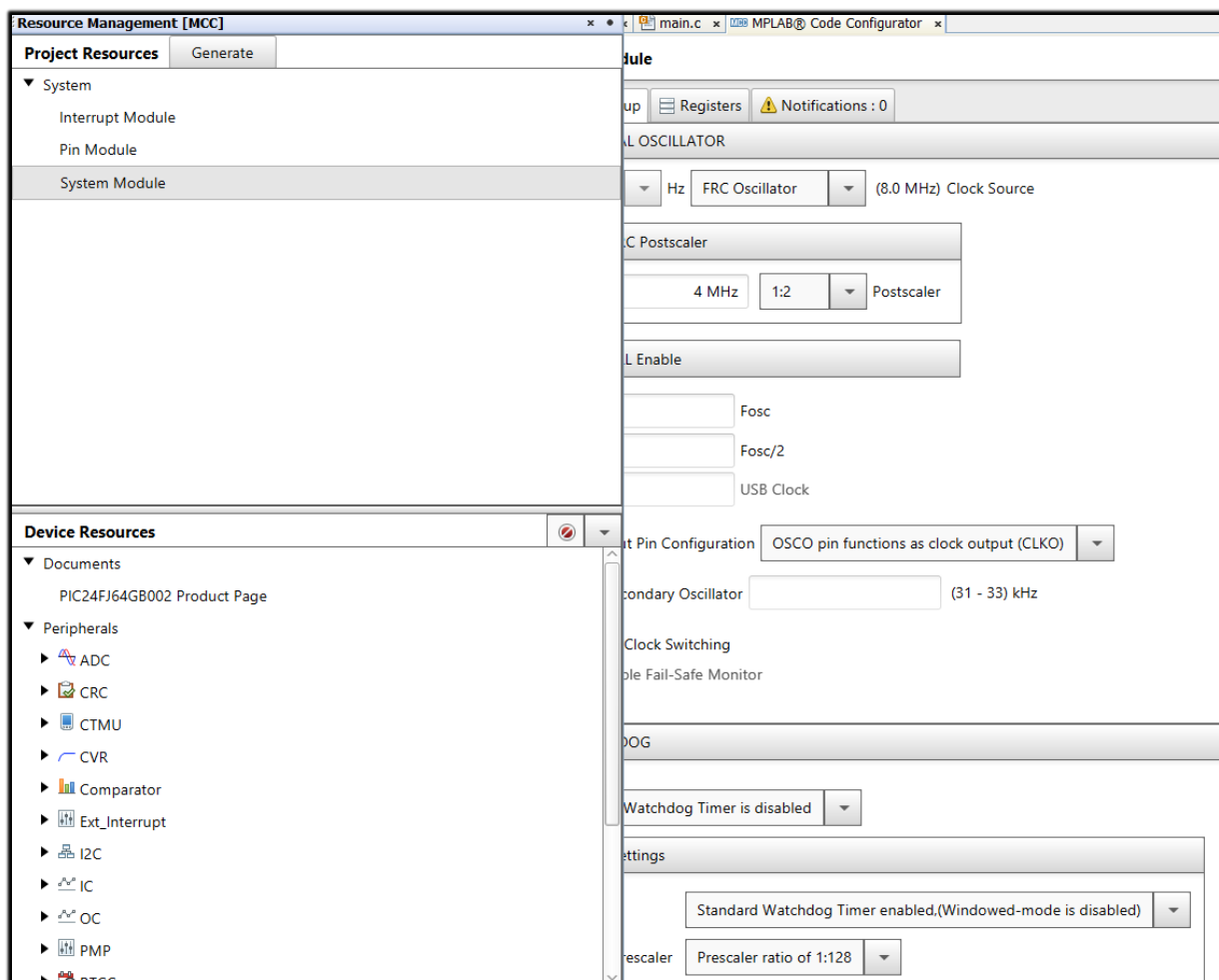


Figura 5. MPLab X IDE y MCC

4.3. OrCAD Capture

Es una herramienta de desarrollo de circuitos por medio de diseños esquemáticos, además de que permite la simulación de los modelos eléctricos. Trabaja con OrCAD CIS para la información de los componentes.

Uno de los puntos de interés de esta herramienta en el desarrollo del proyecto es que permite obtener un archivo “netlist” a partir del cual se puede por medio de OrCAD Layout dar geometría real al sistema. Ya que permite la elección de los encapsulados de los componentes con sus medidas y posición de las conexiones.

Además el entorno permite comprobar que al menos las conexiones se han realizado correctamente. Más aun, posee herramientas para simular el comportamiento del sistema, aunque solo a un nivel eléctrico, no posee capacidad para la simulación de lógica digital. Sin embargo tiene gran utilidad y permite añadir librerías que creadas por fabricantes, además de las propias de OrCAD.

Algunas de las características de Capture son:

- ❖ Permite realizar esquemas del tipo PCB, FPGA o CPLD.
- ❖ Existe un esquema jerárquico que proporciona una edición rápida.
- ❖ Tiene un visor y editor que permiten trabajar con varios proyectos.
- ❖ Los diseños son reutilizables.
- ❖ Hay una relación entre la conectividad visual y la eléctrica.

Algunos de los análisis a los que OrCAD permite someter el circuito por medio de simulación y cálculo son:

- ❖ Análisis AC Sweep/Noise: se emplea para calcular la respuesta en frecuencia del circuito para un rango de frecuencias especificado. El análisis de ruido se realiza a la par que la respuesta en frecuencia. Los elementos que generan ruido son las resistencias y los semiconductores. Con el análisis en Noise se calcula el ruido generado por cada elemento para el rango de frecuencia especificado en análisis AC Sweep, dando como resultado la influencia de la entrada sobre a señal de salida.
- ❖ Análisis DC Sweep: realiza un barrido en torno a una serie de valores de una fuente de entrada independiente, ya sea de intensidad o voltaje, de temperatura, de un parámetro interno del modelo o de un parámetro

global. Se calcula el punto de trabajo para todos y cada uno de los valores que tome la variable que se ha especificado tener en cuenta.

- ❖ **Análisis Transient:** permite el estudio del comportamiento del circuito a lo largo del tiempo. El programa permite determinar el punto de trabajo del circuito, pudiendo sacarlo en un fichero de salida.
- ❖ **Monte Carlo:** Monte Carlo permite conocer qué efectos tiene sobre a señal de salida la variación que se produce en el valor de los componentes a causa de la tolerancia. Primero simula con el valor nominal sujeto a tolerancia y seguidamente con valores aleatorios del elemento.
- ❖ **Worst Case:** lleva a cabo un análisis de sensibilidad y condiciones críticas del circuito, variando solamente un parámetro en cada ejecución.
- ❖ **Save Bias Point:** permite guardar información del punto de trabajo calculado en un fichero distinto del fichero de salida.
- ❖ **Load Bias Point:** permite cargar el archivo de salida de un circuito en otro. Requiere de que previamente se haya realizado el "Save Bias Point".
- ❖ **Análisis de temperatura:** lleva a cabo los análisis que se le indiquen para distintas temperaturas. En el caso de no haber especificado ninguna, el programa toma 27°C por defecto.
- ❖ **Análisis de Fourier:** determina los coeficientes de la serie de Fourier para un ciclo completo de la forma de onda. Se puede especificar la frecuencia fundamental, el número de armónicos y las variables de salida a las que realizar dicha serie.
- ❖ **Análisis paramétrico:** realiza el barrido de una variable en torno a un rango de valores. Seguidamente se simula el circuito para todos esos valores dados anteriormente.

4.4. OrCAD Layout

A partir de la lista de conexiones generada por Capture desde el esquema se puede trabajar con OrCAD Layout para diseñar una placa. El entorno permite elegir en que capa realizar las operaciones. Además tiene herramientas para la conexión de los componentes y el diseño de las pistas.

Lo primero que se debe de hacer con la “*netlist*” obtenida a través de Orcad Capture es asignar a cada uno de los componentes un encapsulado con las medidas que se correspondan. En el caso de no disponer de uno que se ajuste a las especificaciones puede generarse otro desde cero o partiendo de un “*footprint*” ya existente.

La información recogida en la netlist es la siguiente:

- ❖ Nombres de las librerías en la PCB para los componentes.
- ❖ Encapsulado eléctrico.
- ❖ Nombres de los componentes.
- ❖ Nombres de las conexiones.
- ❖ El pin del componente para cada conexión.
- ❖ Información propia de cada conexión, pin y componente.

Los pasos para la creación del archivo .MAX son partir de la “*netlist*” y de emplear una plantilla de placa de OrCAD Layout como “*default.tch*”. De ese “*AutoEco*” se genera el archivo de la placa.

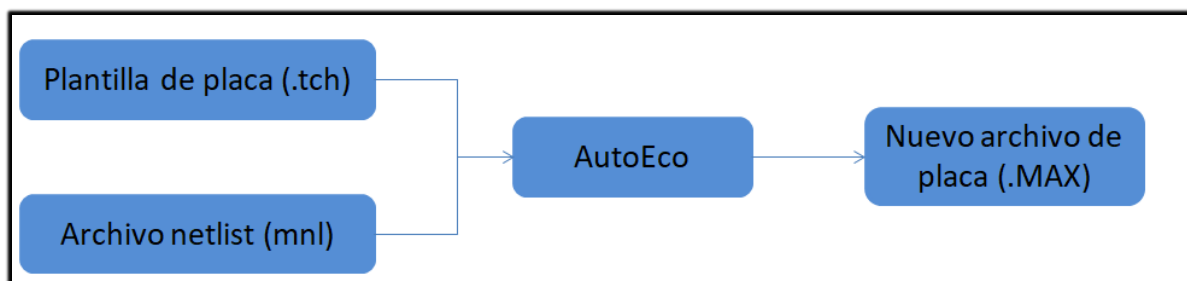


Diagrama 6. Creación del archivo .MAX

El entorno permite distribuir los componentes y seleccionar los atributos de las capas. De modo que para el diseño de las pistas pueden elegirse el número de capas en función de la tecnología de fabricación disponible. Así mismo, es posible seleccionar el ancho de las pistas.

Una vez se define el espacio en el que van a distribuirse las pistas se puede proceder a dibujarlas manualmente o a que lo intente hacer automáticamente el propio programa. En el caso de que el programa lo haga automáticamente hay que comprobar que se han realizado correctamente ya que no mostrará aviso alguno sino es capaz de alcanzar un diseño que satisfaga todas las conexiones.

Para comprobar que el diseño es correcto y cumple otra de las herramientas de entorno permite la verificación de una serie de reglas de diseño, pudiendo elegir cuales tener en cuenta.

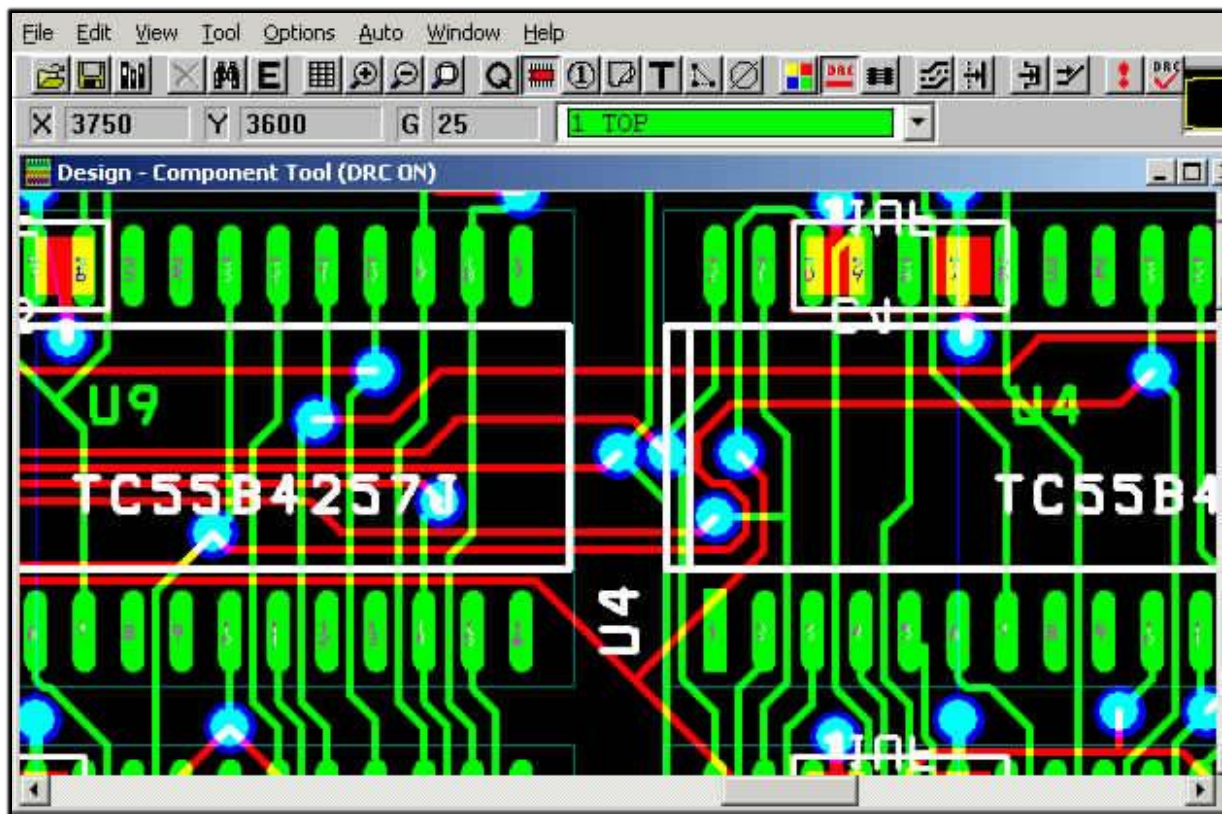


Figura 6. Entorno Orcad Layout

5. Diseño de sistema

Para cumplir con las especificaciones mencionadas se ha elegido un PIC24FJ64B002 ya que cuenta internamente con un módulo RTCC (Real Time Clock Calendar) integrado, permitiendo un diseño final más compacto.

Además da la opción de que el PIC sea funcione como USB Host, lo cual es vital para las operaciones de escritura y lectura exigidas para el sistema. Ambas funcionalidades sumadas a que cuenta con numerosos pines a los que añadir sensores lo hace una solución válida.

Para la adquisición de datos se ha seleccionado un sensor de temperatura LM335, el cual está recomendando para aplicaciones de esta índole. Envía una señal analógica y tiene la referencia de temperatura en kelvin, permitiendo la obtención de temperaturas inferiores a los 0°C sin necesidad de alimentación negativa.

En una PCB diseñada para este propósito se añadirán estos elementos además de las resistencias, condensadores y demás componentes que se requieren para el montaje de un prototipo funcional del sistema.

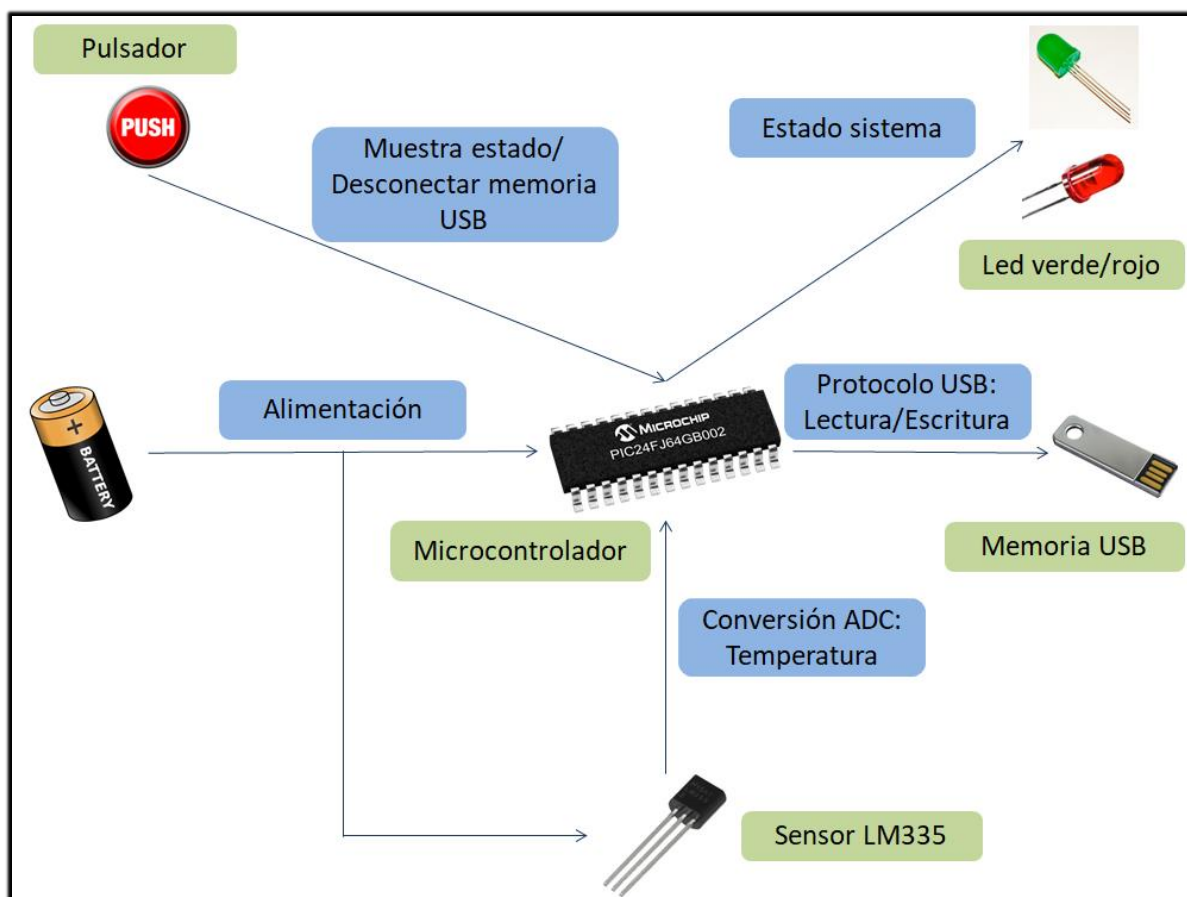


Diagrama 7. Estructura de sistema

El funcionamiento del sistema exige la gestión de varios periféricos. Parte de la solución a la que se ha llegado ha sido planteando el programa como una máquina de estados.

Para el desarrollo del programa se ha requerido de distintas herramientas software como el entorno de MPLAB X IDE V3.51, la herramienta integrada en dicho sistema “Microchip Code Configurator”, las MLA, Orcad PSPICE y Orcad Layout. Desempeñando cada una de ellas una función específica dentro del diseño. Ya que hay que atender tanto a un código para el PIC como a un sistema hardware que lo haga funcionar.

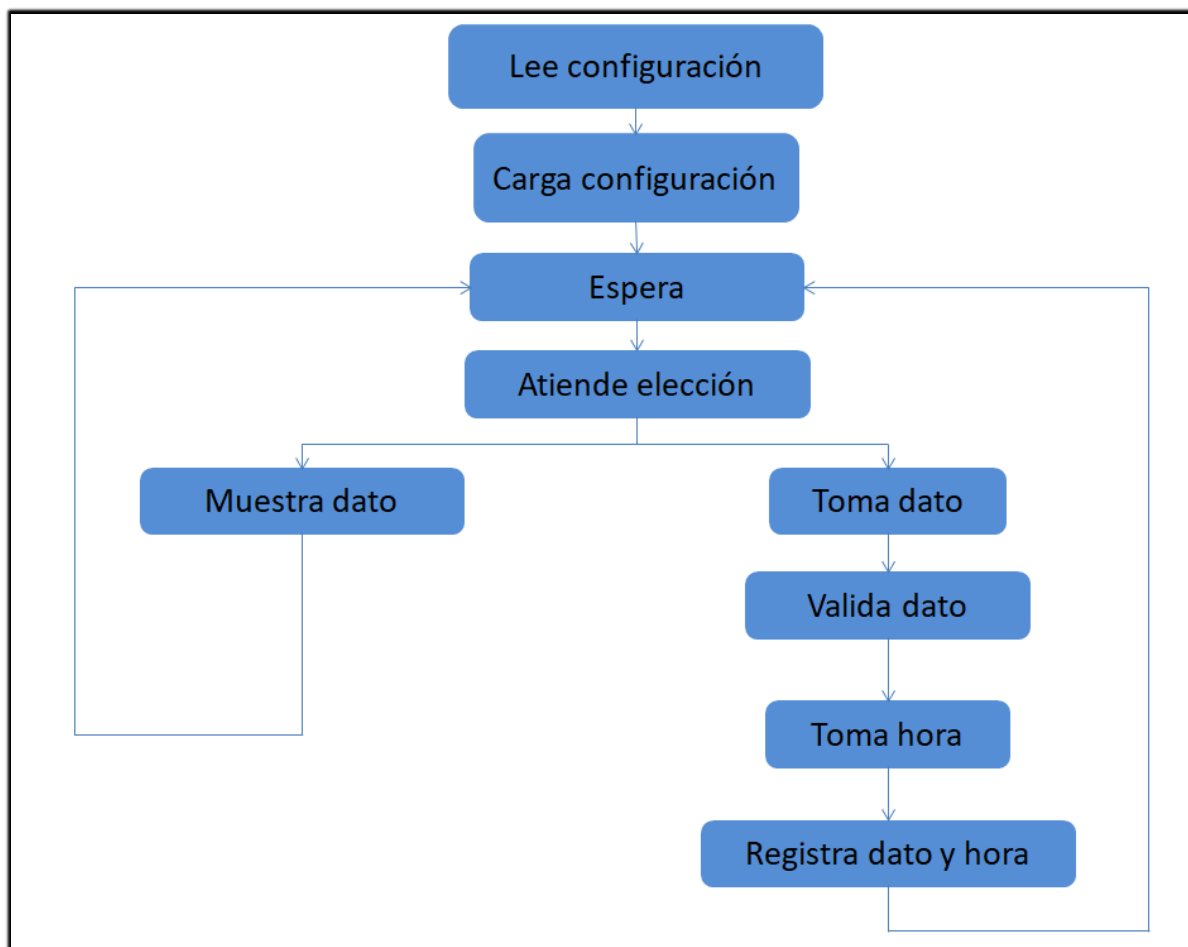


Diagrama 8. Flujo de ejecución

Los programas se organizan en dos bloques, aquellos que aportan una solución software y los que responden a la necesidad de crear un diseño hardware. En el primer bloque se encuentran MPLAB, las MLA y la herramienta MCC del propio entorno de desarrollo, mientras que en el segundo bloque estaría PSPICE y LAYOUT.

El apartado software satisface la comunicación y configuración de los periféricos. Principalmente tomando el ejemplo de comunicación USB Host de las MLA para la detección, escritura y lectura de la memoria USB. La herramienta MCC permite una configuración adecuada del microcontrolador para que el RTCC funcione con el reloj interno y se habilite tanto su alarma como la interrupción que permite despertar al sistema cuando entre en el modo Sleep. Todo esto se desarrolla en el entorno de MPLAB.

El bloque hardware es principalmente el diseño de un esquemático por medio de Orcad PSPICE, para posteriormente dar geometría real con los encapsulados de los que se dispone y los componentes reales.

5.1. PIC24FJ64B002

Es un microcontrolador de la empresa Microchip, y perteneciente a la familia PIC24FJ64GB004. Este modelo concreto tiene 28 pines y cuenta con la posibilidad de usarse como USB host además de incluir un módulo RTCC interno.

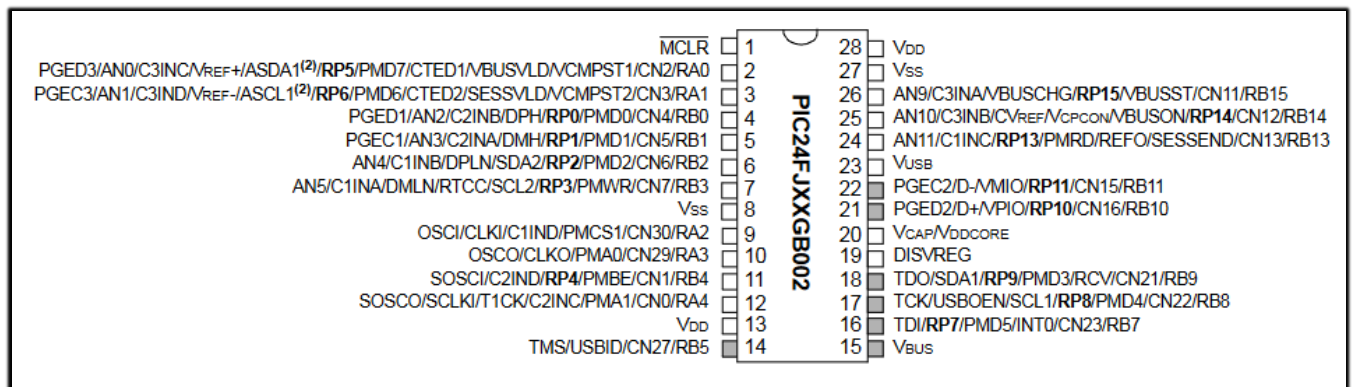


Diagrama 9. PIC24FJ64JB002

❖ Modo USB Host:

Para el funcionamiento de la comunicación por USB se necesita de un cristal externo de 8 MHz. Además de tener en cuenta que se debe de alimentar con 5V. Microchip en la datasheet de la familia de PIC24FJ64GB004 proporciona ejemplos de montajes funcionales.

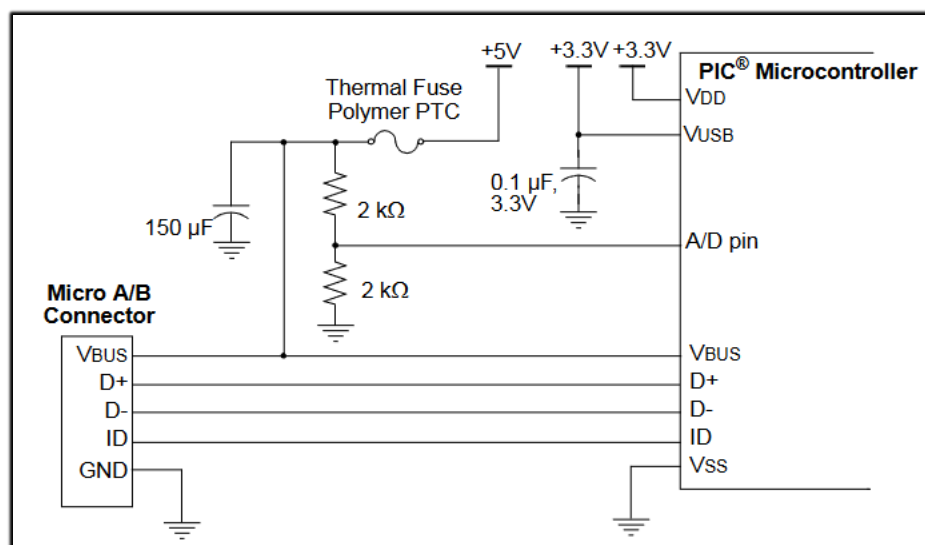


Diagrama 10. Configuración USB HOST

En el modo Host se inicia un protocolo para intercambio de datos. Una vez hecho esto todas las operaciones de escritura, lectura y creación de ficheros de datos están disponibles. También debe de tenerse en cuenta que operaciones como FSfseek permiten mover el puntero dentro de un archivo para acceder justamente a donde se quiere.

Para evitar una mala comunicación se puede forzar a que no haya un cambio de estado hasta que esta no se haya llevado a cabo, sin embargo esta solución puede dar lugar a que el sistema se quede bloqueado. Una solución mejor es hacer que se espere una respuesta por parte del protocolo, y que en función de si detecta o no dispositivo actúe, esta alternativa permite que el sistema no se quede bloqueado pero que a su vez las comunicaciones tanto para lectura como escritura se realicen sin errores posibles.

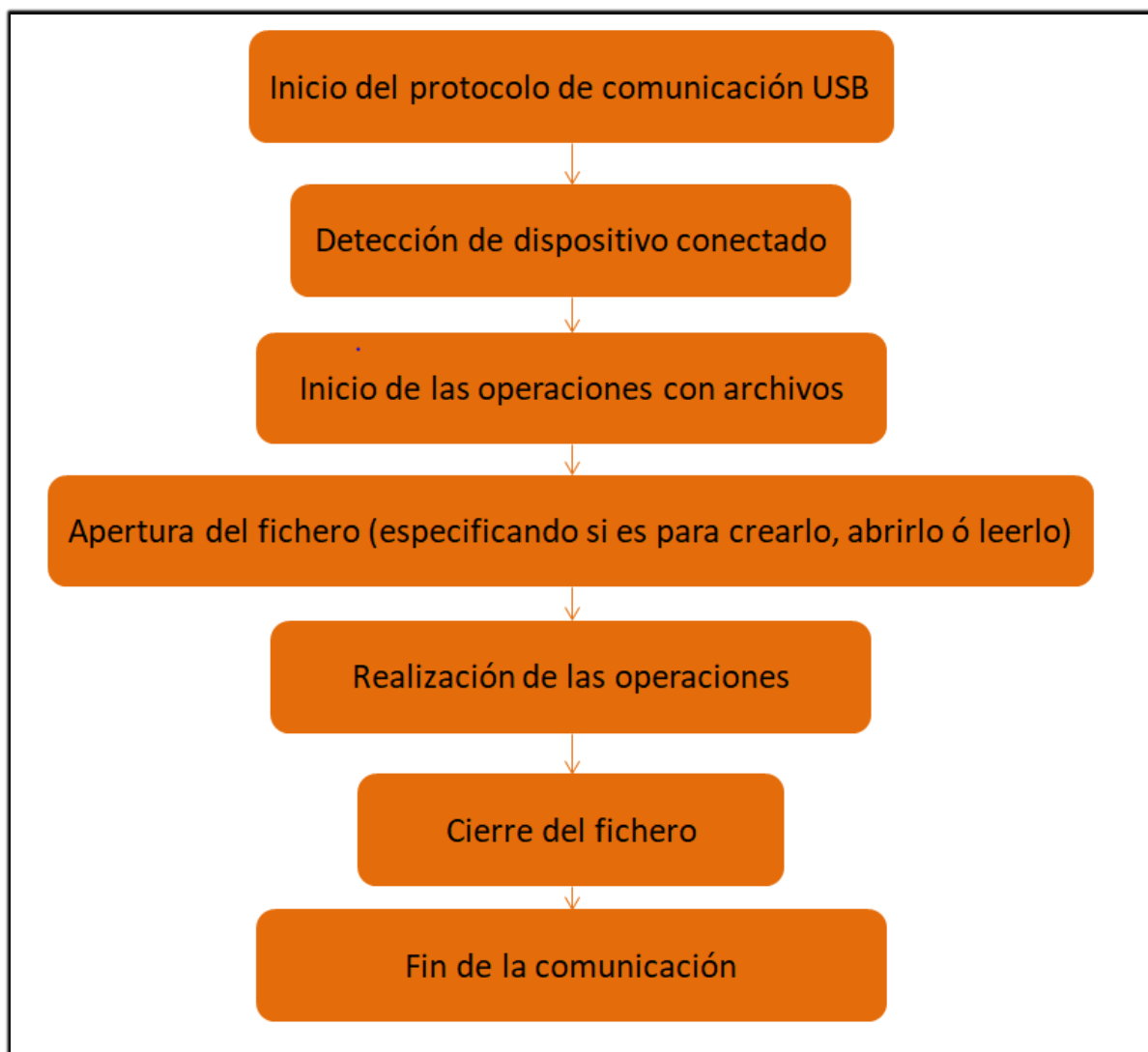


Diagrama 11. Comunicación USB

❖ Módulo RTCC:

Para el módulo RTCC (Real Time Clock Calendar) interno se debe de tener en cuenta si se desea el uso de un oscilador interno o externo. Para este dispositivo se ha decidido aprovechar el propio oscilador interno LPRC a una frecuencia de 32.7 KHz, dado que la datasheet del microcontrolador asegura que este cristal tiene una precisión del 0.5% es una elección válida.

Para la inicialización del reloj deben de introducirse una serie de órdenes para habilitar primero la configuración del RTCC, tanto el propio reloj como la configuración de la alarma. Seguidamente es necesario indicar que va a configurarse y “RCFGCALbits.RTCEN” inicia una secuencia descendente en la que “RTCVAL” va tomando y volcando en su lugar cada parámetro. Excepto el año el resto de los parámetros van a pares: mes y día del mes, día de la semana y hora y minuto y segundo.

Para obtener los valores temporales actuales se tiene que introducir la misma secuencia pero volcando los datos en variables y tratándolos posteriormente para separarlos ya que excepto el año el resto van a pares.

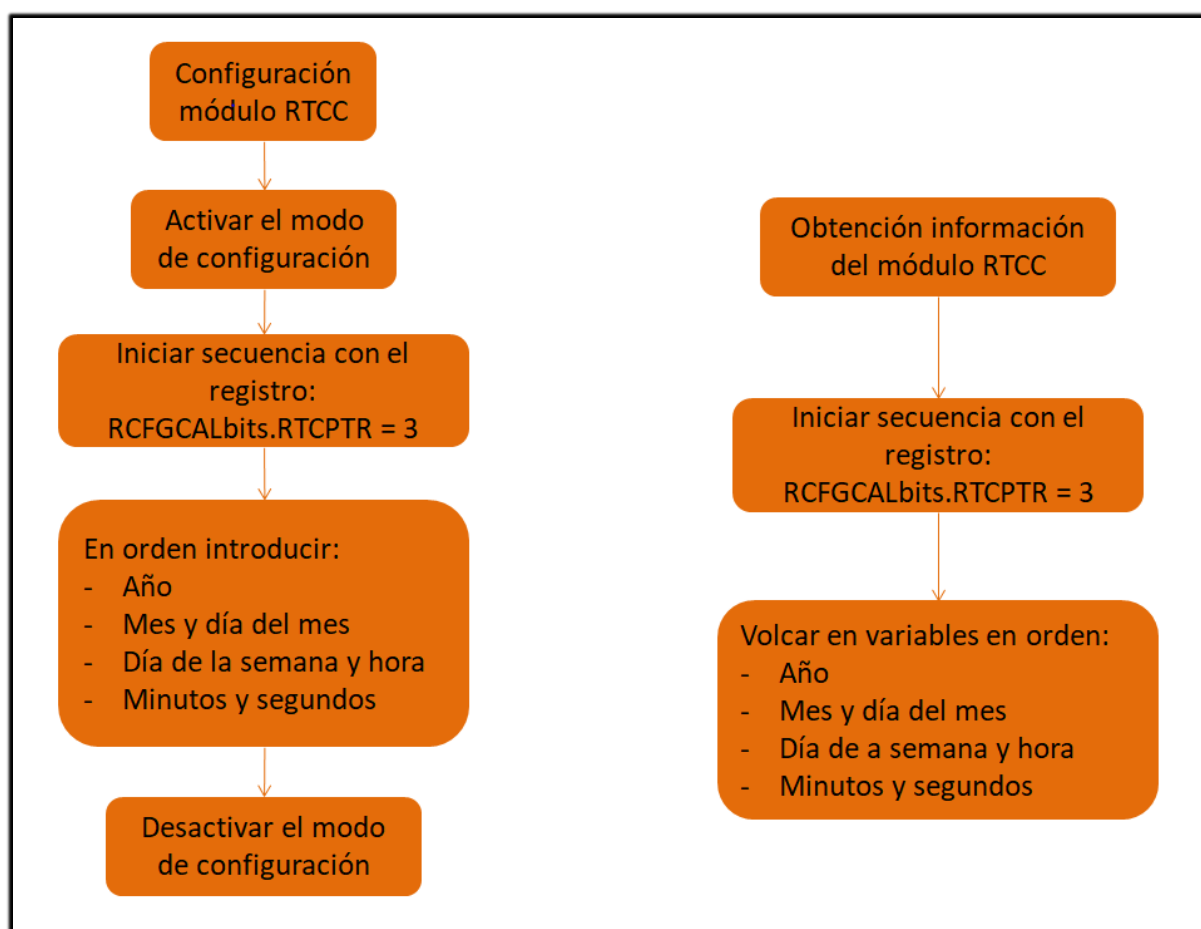


Diagrama 12. Funcionamiento módulo RTCC

Es posible cargar distintas configuraciones en el módulo RTCC, pero para ello debe de tenerse en cuenta la exigencia de llevar a cabo un protocolo con los registros que habiliten el cambio. Sin embargo tiene que tenerse en cuenta que si se produce una vez iniciado hay que resetear todos los flag o comenzará a producir errores debidos antiguos.

La opción más sencilla y directa para por una reinicialización del módulo a través de las propias opciones que ofrece. Debe tenerse en cuenta en el caso de que se desea seguir usando el sistema sin hacer un reset pero cambiando los parámetros de fecha y hora. Esta opción se contempla ya que es posible que los datos iniciales no sean correctos y halla que reintroducirlos para corregirlos.

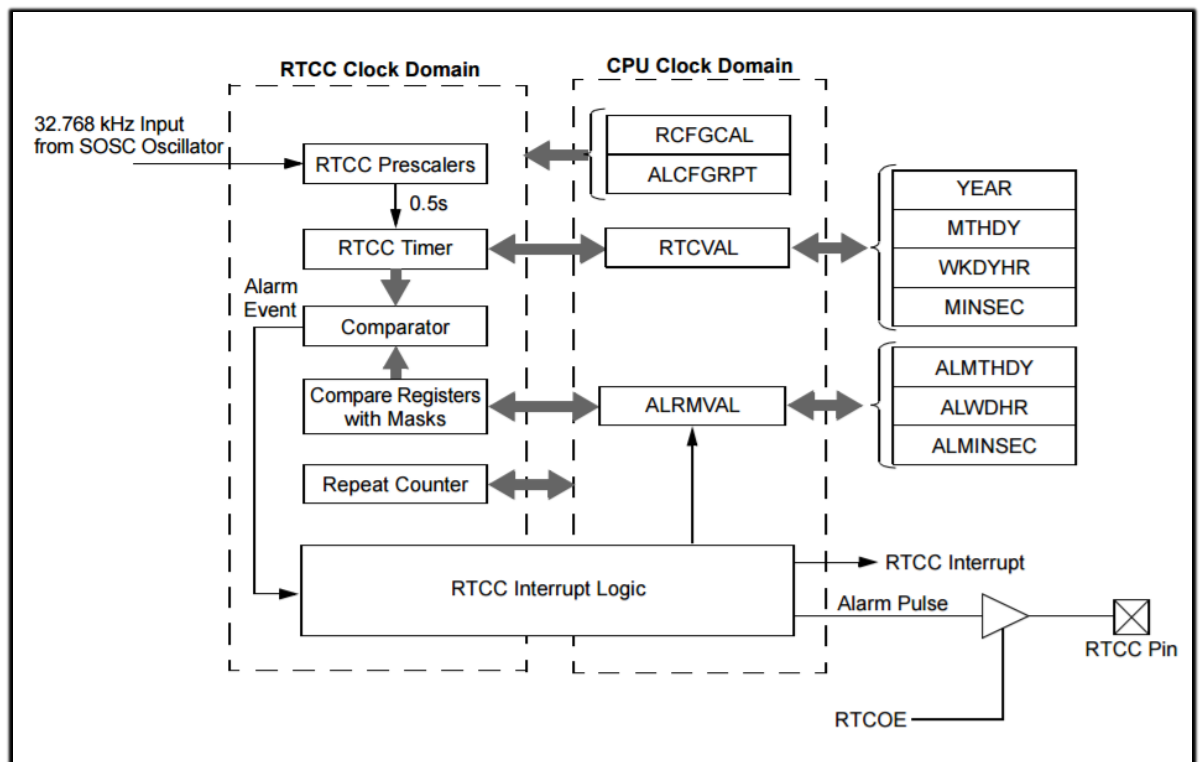


Diagrama 13. Módulo RTCC

❖ **Módulo ADC:**

Algunos pines permiten la opción de usar un módulo de conversión de señales analógico/digitales. El cual convierte a valores comprendidos de 0 hasta 2023, esto junto con la información de la tensión de la señal de entrada permite por medio de una ecuación conocer el valor de entrada. Es útil para por ejemplo sensores de temperatura que suministran la información de este modo.

La configuración de un pin para el funcionamiento de este módulo requiere de algunos registros que habiliten dicho pin para tal funcionamiento, además de que el hardware del propio microcontrolador posea ese módulo como es el caso del PIC24FJ64GB002 y el resto de su familia.

Las características del módulo ADC son las siguientes:

- ❖ Velocidad de conversión de 500 ksps
- ❖ Hasta 16 canales analógicos de entrada
- ❖ Tensión de entrada referenciada en el pin de entrada
- ❖ Buffer de conversión de palabras de 16bits
- ❖ Modo automático de escaneado del canal
- ❖ Funcionamiento durante modo de bajo consumo y ahorro de energía
- ❖ Capacidad para la selección de del origen den pulso
- ❖ Amplificación de las muestras y retención unipolar

Para el funcionamiento del módulo ADC se requiere una inicialización inicial del módulo seguida de un comando que dé comienzo a la conversión. Hay que tener en cuenta que la conversión tarda un tiempo, de modo que la exigencia inmediata de un valor por parte del módulo ADC dará en la mayoría de los casos un dato erróneo. Para evitar eso existen dos estrategias, una de ellas es esperar un tiempo de varios milisegundos con el fin de dar un margen amplio a la conversión, el con un “while” esperar a que un flag que indique el proceso de conversión ha acabado se ponga a nivel alto.

En el caso de que se requiera asegurar que el dato obtenido es correcto y no se ha visto alterado se recomienda hacer varias muestras sucesivas en el menor periodo de tiempo posible (lo cual hace que la segunda forma de conversión sea la más adecuada y eficaz en este caso)

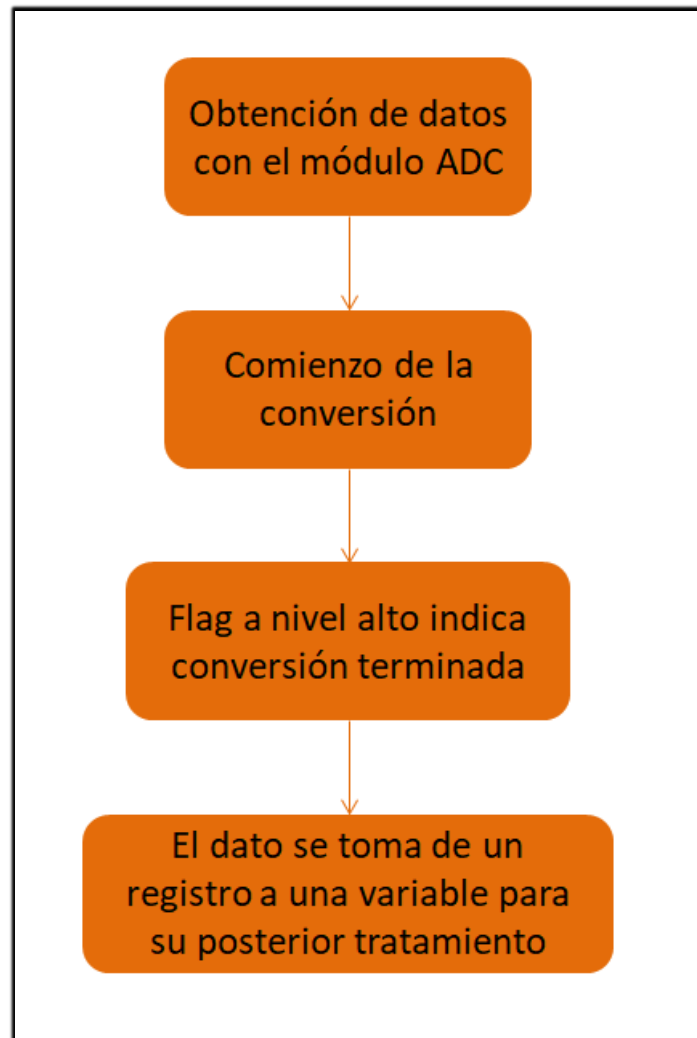


Diagrama 14. Funcionamiento del módulo ADC

5.2. LM335

Es un sensor de temperatura que envía una señal analógica de 0 a 255. Resulta fácil de calibrar y presenta un circuito integrado muy compacto. La relación de la tensión con la temperatura es de 10mV/K . Puede funcionar con una corriente de $400\mu\text{A}$ hasta 5mA sin mostrar ningún cambio en su funcionamiento. Con un calibrado a 25°C presenta un error inferior a 1°C . Al contrario que otros sensores el LM335 presenta una salida lineal.

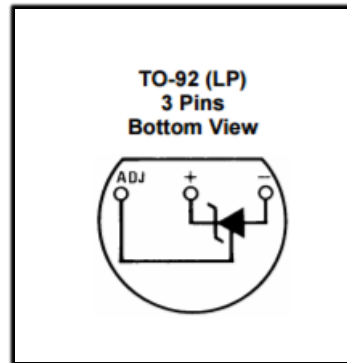


Diagrama 15. Esquema de conexiones del LM335

Como sensor tiene multitud de aplicaciones ya que cuenta con un rango de operación de entre -40°C a 100°C . Dada su baja impedancia y linealidad es fácil de usar. El hecho de que la escala esté en kelvin permite la obtención de temperaturas por debajo de los 0°C sin necesidad de alimentación negativa.

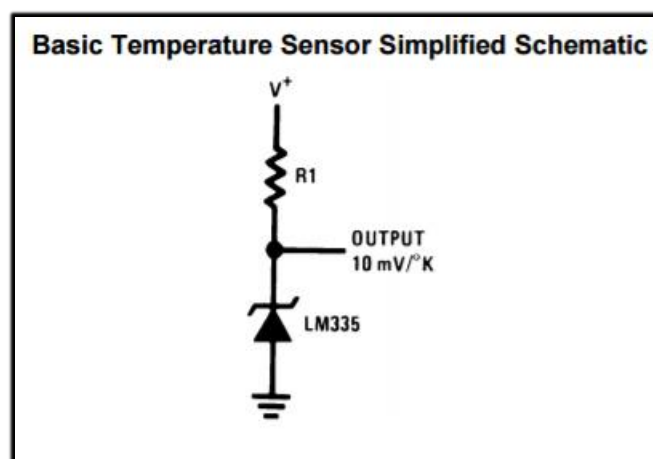


Diagrama 16. Montaje básico del LM335

Características:

- ❖ Calibrado en una escala de Kelvin
- ❖ Tiene una precisión inicial de 1 grado kelvin
- ❖ La impedancia dinámica es menor a 1Ω
- ❖ Posee un calibrado sencillo por medio de una resistencia de al menos $10K\Omega$ conectada a la patilla ADJ y otra de al menos $1K\Omega$ desde la alimentación.
- ❖ Bajo precio
- ❖ Amplio rango de operación de -40°C hasta 100°C

El calibrado recomendado en el caso de que se vaya a emplear un conversor analógico digital asociado a un pin es realizar el ajuste a $-273'015^{\circ}\text{C}$, lo cual se corresponde a una tensión de entrada de 0V. La expresión matemática vendría dada por:

$$V_{outT} = V_{outT_0} \times \frac{T}{T_0}$$

Un montaje recomendado por el fabricante sería de emplear una resistencia de $1k\Omega$ como R1 y alimentar en sensor con 5V, respetando que la corriente se encuentre siempre dentro de los límites.

5.3. LT1121-5

Es un regulador que proporciona una alimentación de 5V con una corriente de 150mA. Está diseñado para alimentarse de una batería y operar con corrientes de entre $30\mu\text{A}$ y $16\mu\text{A}$. Otra de sus ventajas es que es capaz de trabajar con pequeñas capacidades a la salida. Es estable con capacidades de $1\mu\text{F}$ a $100\mu\text{F}$.

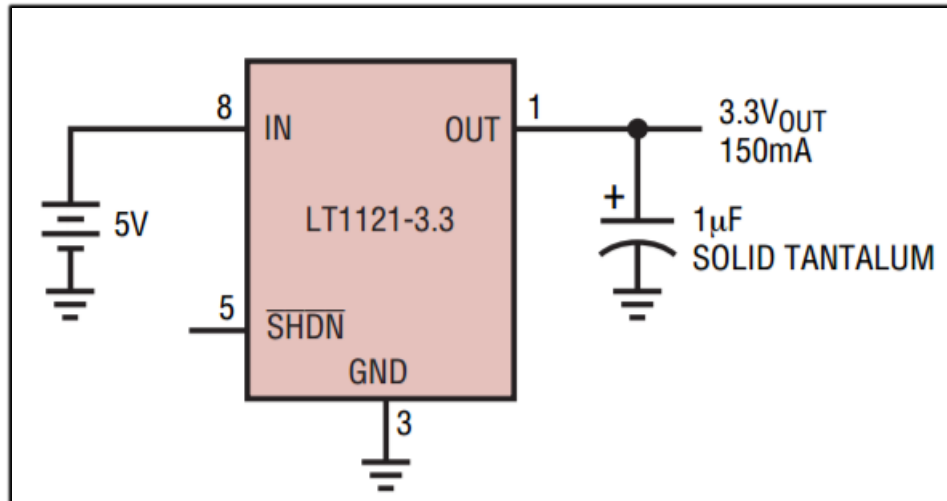


Diagrama 17. Montaje básico del LT112-X

Posee las siguientes características:

- ❖ Caída de tensión de 0.4V
- ❖ Salida de corriente de 150mA
- ❖ Corriente en estado inactivo de $30\mu\text{A}$
- ❖ No requiere de diodos de protección
- ❖ Limitación térmica
- ❖ Estable con una capacidad de $0.33\mu\text{F}$
- ❖ Con una entrada baja no hay corriente inversa

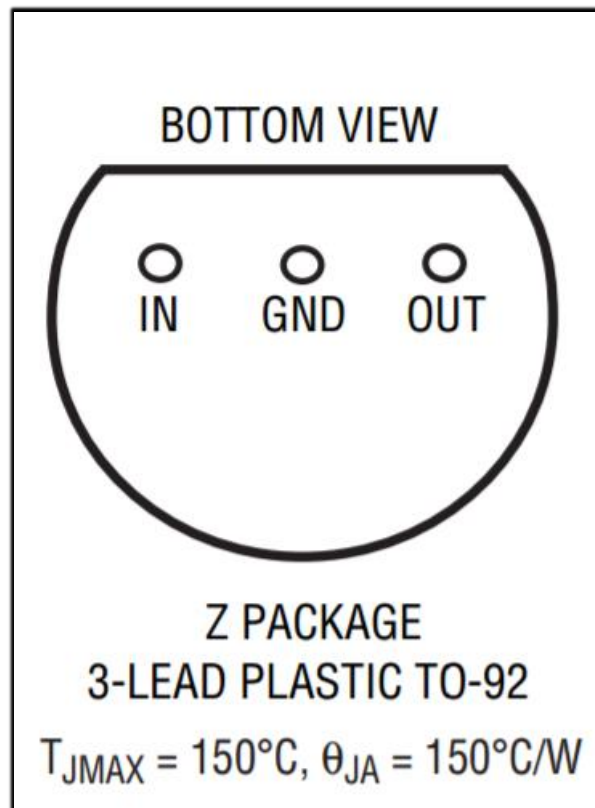


Diagrama 18. Esquema de conexiones del LT1121-5

Input pin: El pin de entrada está diseñado para funcionar con tensión inversa respecto a tierra, de modo que no hay riesgo de dañar el dispositivo si la batería se conecta al revés por error. Por otro lado a la salida solo puede obtenerse tensión directa.

Output pin: El pin de salida proporciona la energía a la carga. Según la información proporcionada por el fabricante se recomienda un condensador de $1\mu F$.

Es capaz de funcionar en un rango de $0^{\circ}C$ a $125^{\circ}C$ con una tensión de entrada que va desde $-30V$ a $36V$. Puede mantenerse almacenado a una temperatura de que varía de los $-65^{\circ}C$ a $150^{\circ}C$. Se puede adquirir en 8 pines y 3 pines, siendo el de tres pines en elegido para este proyecto dado que no se requieren las opciones que ofrece el modelo de 8.

5.4. PCB

Para el diseño hardware del sistema se han seguido los montajes recomendados por los distintos fabricantes de los componentes que lo forman. Dado que los montajes de elementos como el sensor de temperatura, cristal de 8Mhz, botón de reset, pulsador, led verde y rojo y USB han sido previamente probados en el sistema de pruebas se ha seguido ese mismo esquema pero con un diseño propio. Dadas las exigencias del software se ha tenido que diseñar teniendo presente a que pines va conectado cada elemento.

Primero se parte de un diseño esquemático realizado en Orcad Capture y del cual se puede obtener posteriormente una netlist.

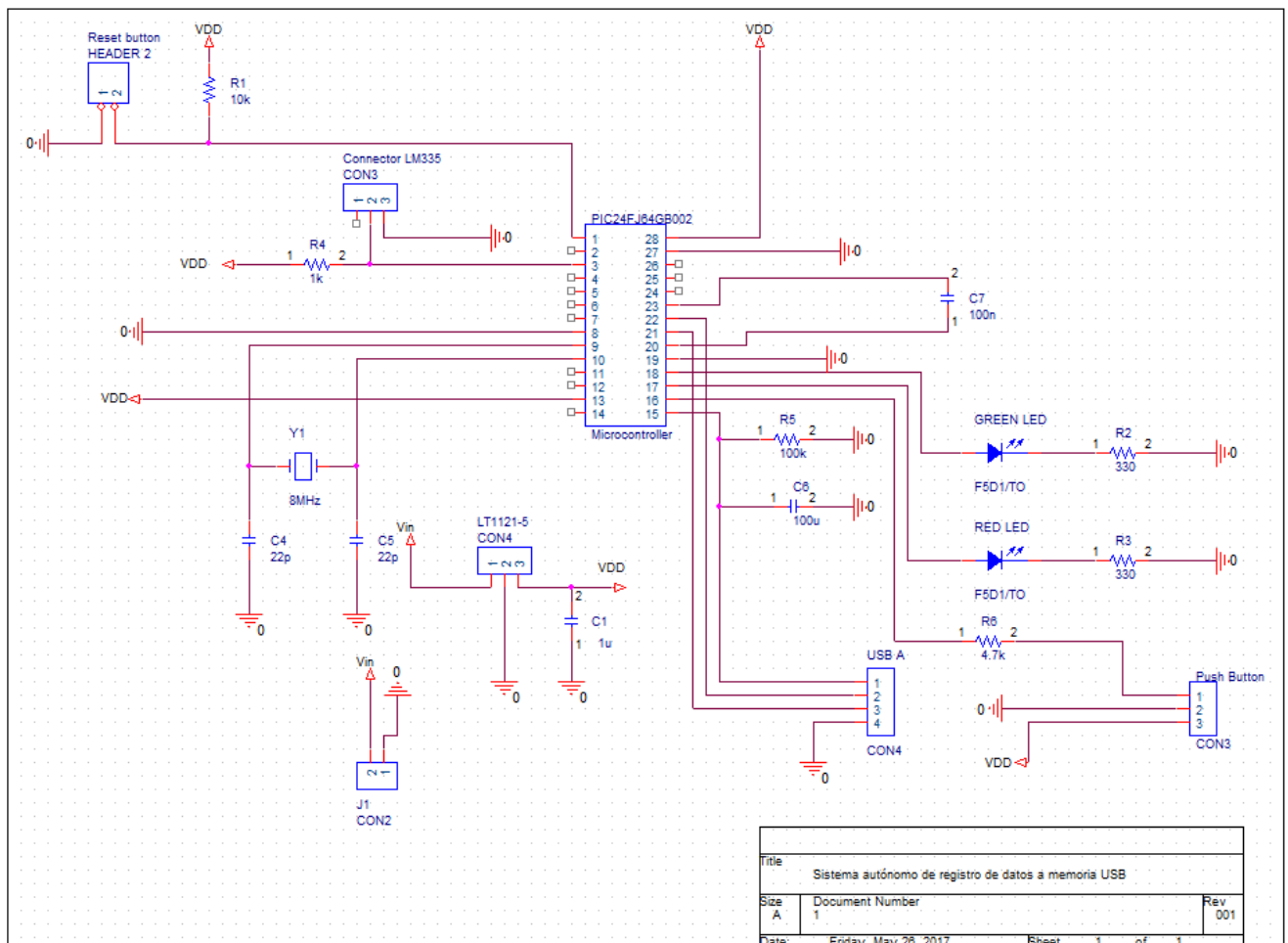


Diagrama 19. Esquemático del sistema

El esquemático solo tiene la relación de las conexiones de los componentes y los valores de los mismos, pero carece de geometría real y distribución. Para poder trasladar este esquemático a una placa real se emplea la netlist de Capture y se crea un archivo “.MAX” con Orcad Layout.

Lo primero es asignar a cada componente una geometría de las que se encuentran en las librerías. Dado que alguno de los componentes es posible que no tenga una que se ajuste de forma adecuada se procede a crearle un footprint.

Después de esto el entorno de Layout permite la ubicación de los componentes y señala las interconexiones que existen entre los distintos pines, pero aún no están las pistas hechas. Para ello se crea un cuadrado que sería el espacio de la paca y se colocan todos los componentes en su interior. Se han realizado varias tentativas hasta conseguir un diseño compacto.

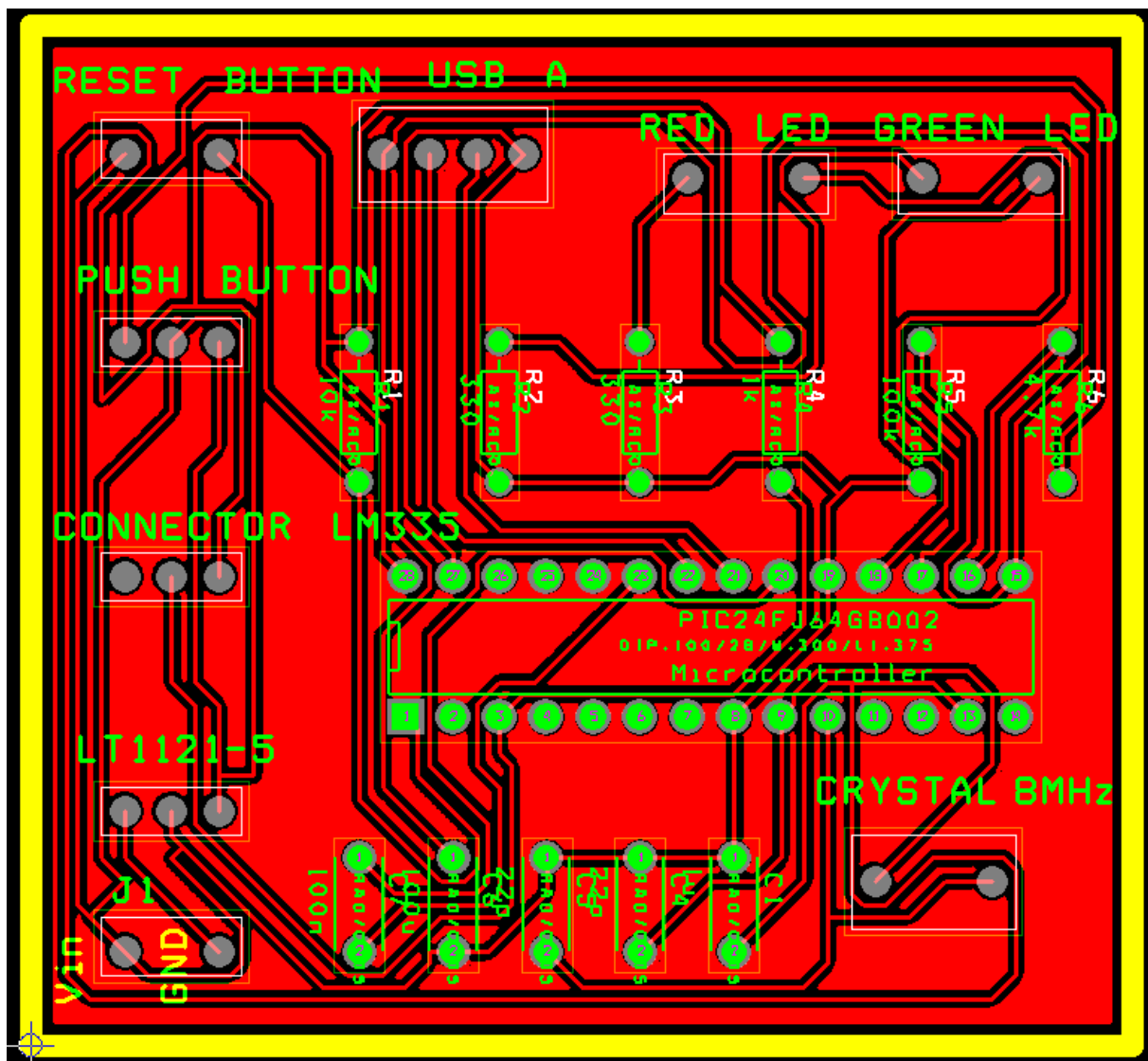


Diagrama 20. Layout del sistema

Tras tener este diseño se comprueba que cumple con una serie de reglas, Orcad Layout es capaz de hacerlo. Se han comprobado todas las reglas que el programa permite como el espacio entre componentes, el conexionado, el espacio entre vías y el posicionamiento de las vías entre otros.

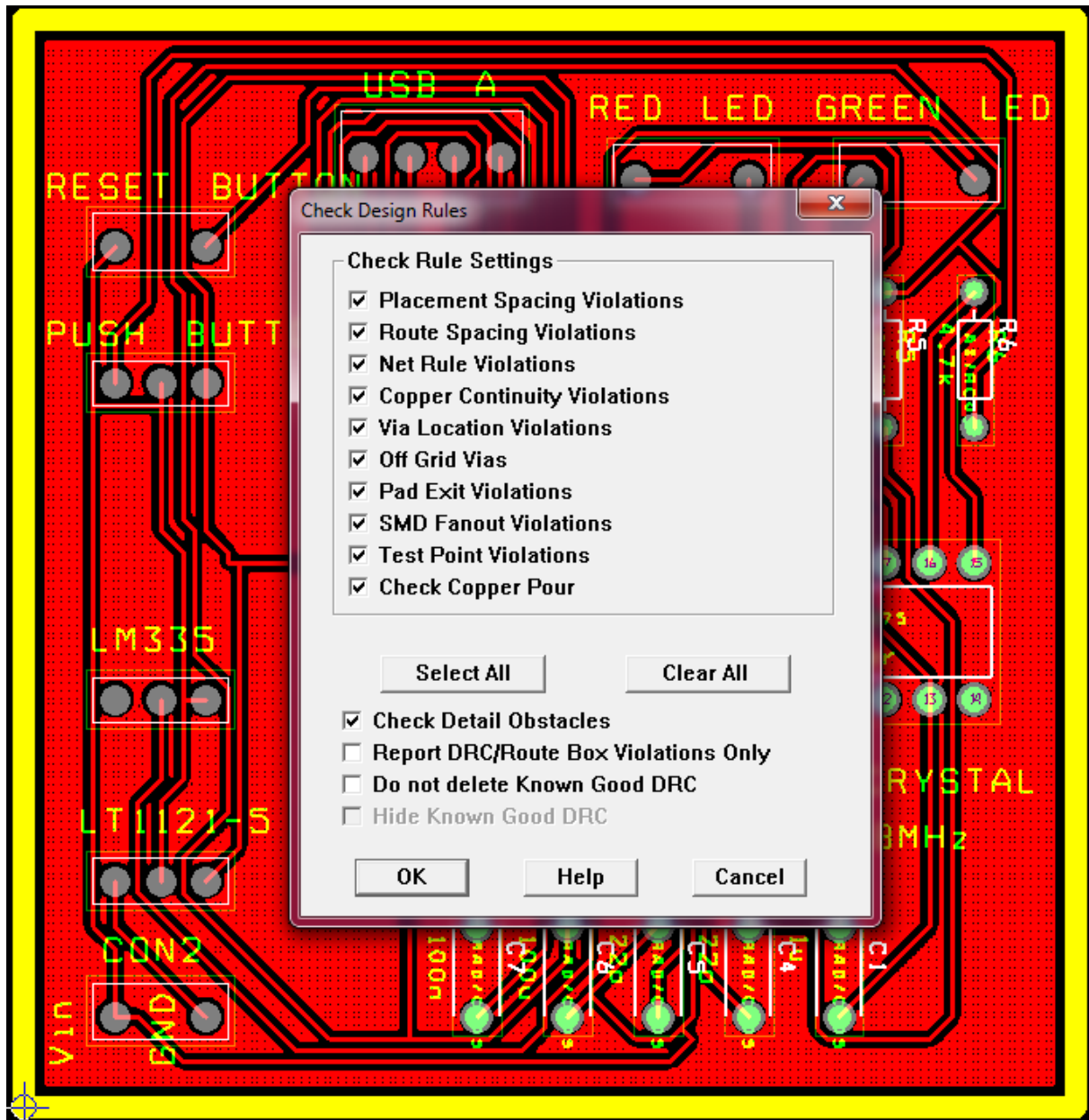


Figura 7. Reglas de diseño del sistema

Para la distribución de los componentes se ha tenido en cuenta algunos factores que el programa no tiene en cuenta. Uno de ellos es que el cristal y sus condensadores estén lo más cerca posible del microcontrolador para evitar imprecisiones no deseadas a causa de la distancia. El regulador y su condensador deben de estar también cerca del PIC.

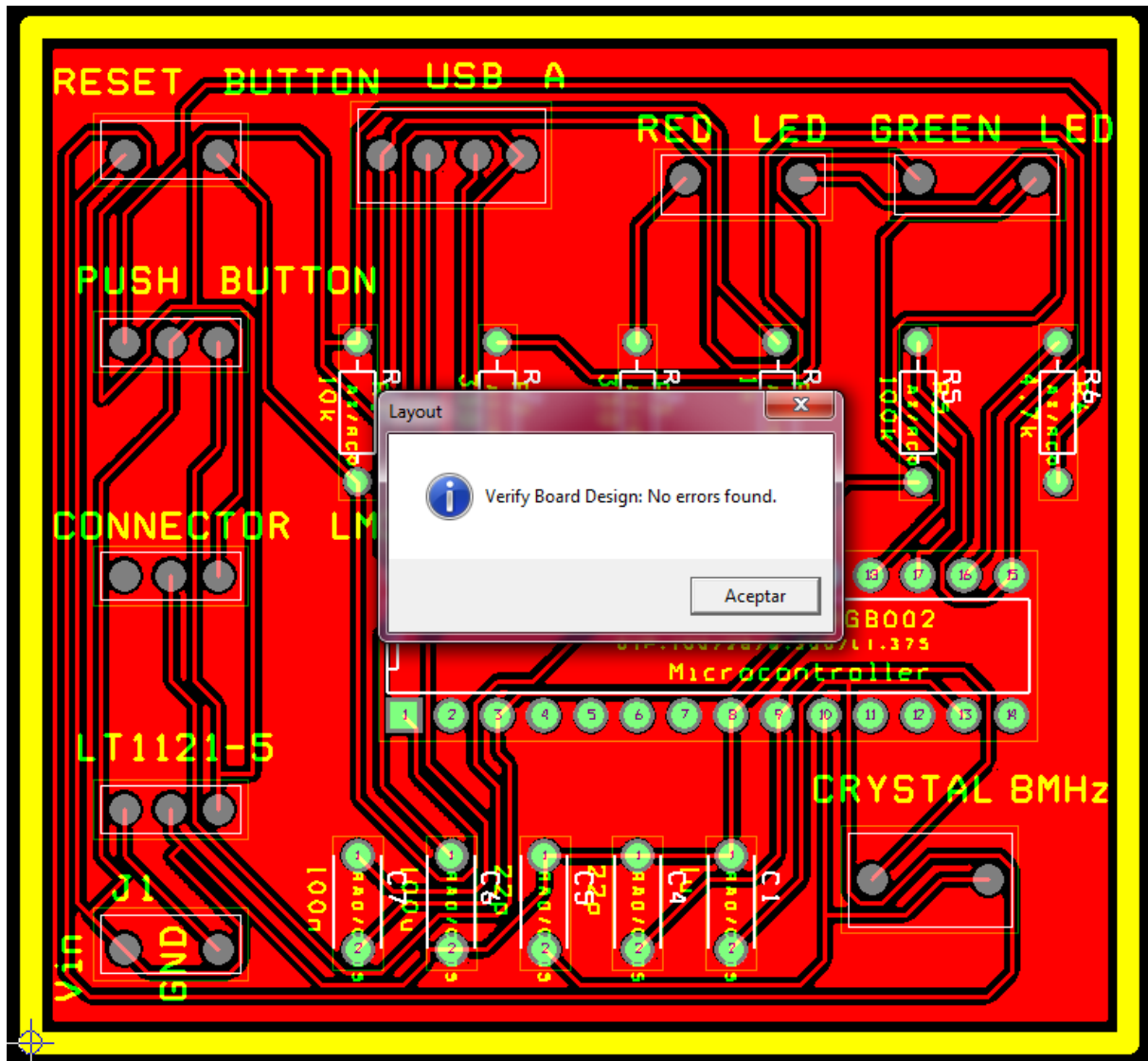


Figura 8. Cumplimiento de las reglas de diseño

Finalmente se ha atendido a una distribución de los elementos que permita una manipulación cómoda del dispositivo. Como es disponer de ambos led próximo entre sí en una esquina para poder ver ambos a la vez, el conector al que va la alimentación está cerca del borde para que el elemento que se conecte no tenga que pasar por encima del resto de la placa y el conector USB se encuentra también cerca del borde y a cierta distancia de otros elementos para sea físicamente posible conectar y desconectar una memoria USB sean cuales sean sus dimensiones sin que otros componentes sean un estorbo en el caso de que tenga un ancho anómalo.

5.5. Consumo del sistema

Se calculan las necesidades energéticas de acuerdo al caso crítico que es la configuración del sistema para tomar muestras en el periodo mínimo, un segundo. De modo que se tiene en consideración el consumo de:

- ❖ Sensor LM335: según su datasheet el sensor funciona con una corriente de hasta 5mA y la tensión de alimentación elegida ha sido de 5V.

$$P_{LM335} = V_{LM335} \times I_{LM335} = 5V \times 5mA = 0.025W$$

$$E_{LM335} = P_{LM335} \times t = 0.025W \times 1h = 25mWh$$

La potencia instantánea consumida por el sensor LM335 es de 25mW.

La energía consumida por el sensor LM335 es de 25mWh.

- ❖ Comunicación USB en modo HOST: la comunicación USB tiene su mayor consumo durante las operaciones de lectura y escritura. Para calcular su consumo en estos momentos se emplea la fórmula dada por el fabricante en el documento de características del PIC24FJ64GB002.

$$I_{USB} = \frac{40mA \times V_{USB} \times P_{zero} \times P_{in} \times L_{cable}}{5V \times 5m} + I_{PULLUP}$$

I_{usb} : Corriente que requiere la comunicación USB para funcionar en modo HOST.

P_{zero} : Porcentaje del tráfico de los bits enviados por el PIC que tienen un valor de cero. Dado que se desea conocer el consumo en el caso más desfavorable se toma como el 100%.

P_{in} : Porcentaje del total de la banda de comunicación del bus que se usa. Caso crítico, de modo que 100%.

L_{cable} : Longitud en metros del cable USB. Para poder emplear la máxima velocidad de comunicación no debe de superar los 5m.

I_{pullup} : Corriente nominal con la resistencia de “pullup” que suministra el pin, 25mA.

$$I_{USB} = \frac{40mA \times 5V \times 1 \times 1 \times 5m}{5V \times 5m} + 25mA = 65mA$$

$$P_{USB} = V_{USB} \times I_{USB} = 5V \times 65mA = 0.325W$$

$$E_{USB} = P_{USB} \times t = 0.325W \times 1h = 325mWh$$

La potencia instantánea consumida por al comunicación USB en modo HOST es de 325mW.

La energía consumida por la comunicación USB en modo HOST es de 325mWh.

- ❖ PIC24FJ64GB002: 5V, 250mA. Esto abarca el pleno funcionamiento del microcontrolador con el módulo ADC y el módulo RTCC.

$$P_{PIC24FJ64GB002} = V_{PIC24FJ64GB002} \times I_{PIC24FJ64GB002} = 5V \times 250mA = 1.25W$$

$$E_{PIC24FJ64GB002} = P_{PIC24FJ64GB002} \times t = 1.25W \times 1h = 1.25Wh$$

La potencia instantánea consumida por el PIC24FJ64GB002 y los módulos RTCC y ADC es de 1.25W.

La energía consumida por el PIC24FJ64GB002 y los módulos RTCC y ADC es de 1.25Wh.

El total de energía consumida por el sistema en el caso crítico es:

$$E_{total} = E_{LM335} + E_{USB} + E_{PIC24FJ64GB002} = 25mWh + 325mWh + 1.25Wh = 1.6Wh$$

Para que la batería garantice el funcionamiento del sistema durante al menos un mes, suponiendo el mes más largo, debe de tener una carga de:

$$1 \text{ mes} = 31 \text{ días} = 31 \times 24 \text{ horas} = 744 \text{ horas}$$

$$Consumo_{mes} = 1.6Wh \times 744 h = 1.190,4W$$

5.6. Software

La estructuración del programa responde a una máquina de estados. Los estados pueden dividirse en partes según el tipo de proceso que realizan: comunicación por medio del protocolo USB en modo host, módulo RTCC, módulo ADC y sensor de temperatura. Las tareas de la adquisición de los parámetros de configuración se realizan tomando los datos a través de la comunicación por medio del protocolo USB.

Se basa en que el PIC24FJ64B002 actúe como host para poder escribir y leer en la memoria. Así tenemos una memoria USB como dispositivo de almacenamiento donde ir volcando todos los datos tomados por medio del sensor. Por otro lado hay que gestionar el módulo de reloj calendario que tiene en su interior el PIC. Con él se consigue, configurándolo previamente, poder obtener la información temporal. Además del módulo ADC para convertir la señal analógica que suministra el sensor y tomar datos de temperatura del entorno.

Para ello se parte de los ejemplos de USB HOST de las MLA (Microchip Legacy Applications) y para el RTCC (Real Time Clock Calendar) de la herramienta MCC (Microchip Code Configurator).

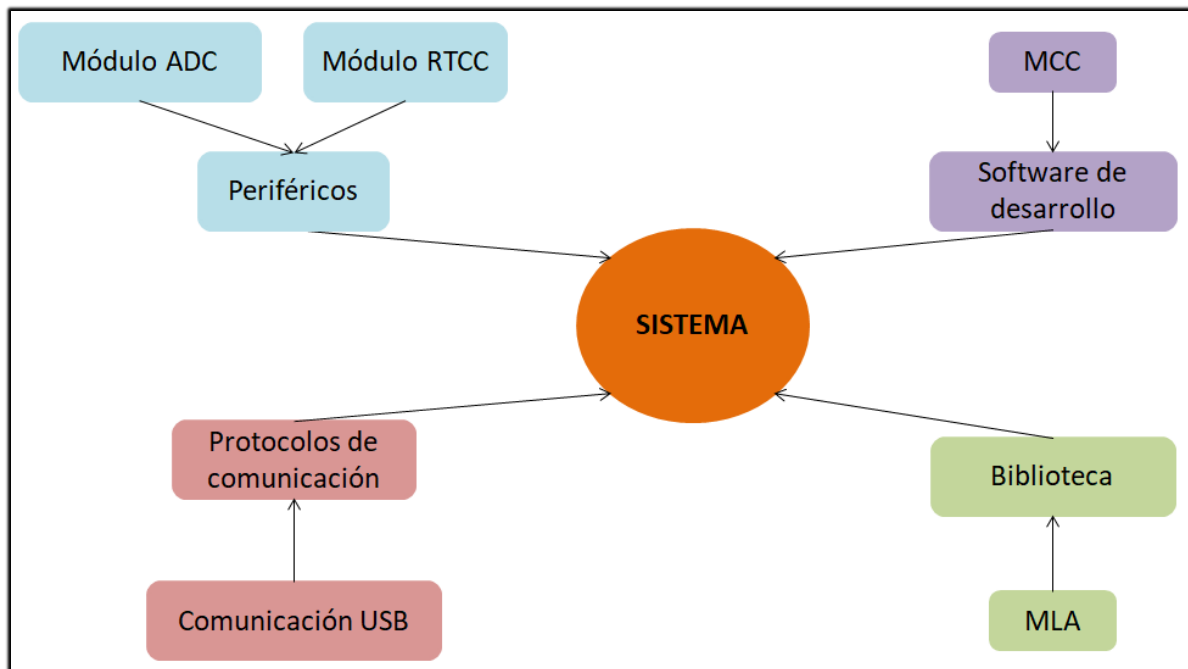


Diagrama 21. Recursos del sistema

Para trabajar con la escritura y lectura de datos además de se usan las funciones creadas en las MLA que recogen todas las necesidades de creación de ficheros, acceso a los mismos, además de la mencionada lectura y escritura.

Función	Descripción
FSInit	Inicia la comunicación por USB y permite el tratamiento de los ficheros.
FSfclose	Cierra el fichero abierto. Es necesario para que la información que contiene no se vea dañada y el dispositivo pueda retirarse de forma segura.
FSfopen	Abre un archivo, puede darse la opción de que en el caso de que no exista se cree uno con el nombre especificado.
FSfread	Lee del archivo abierto anteriormente los caracteres que se le especifican.
FSfwrite	Escribe los caracteres en el archivo abierto, debe introducirse cuantos y de que peso se van a escribir.
FSfseek	Sitúa en puntero en el lugar especificado.

Tabla 4. Funciones para tratar ficheros

5.6.1. Funcionamiento del sistema

1. Detección y lectura de la memoria USB para la adquisición de los datos de configuración. Para ellos se realizan lecturas sucesivas en las que se van volcando los datos a las variables de tipo “*string*” correspondientes.

Esto se consigue de forma sencilla al quedarse el puntero justamente por donde acaba de leer la vez anterior siempre y cuando no se cierre el archivo. Por una cuestión de seguridad y para poder asegurar que la lectura se ha llevado a cabo correctamente, la información tomada se comprueba que es del tipo que debe ser cada uno.

Simplemente esto es comparar que la cadena de caracteres de por ejemplo el año son 4 caracteres (por ejemplo 2017) y así sucesivamente. En el caso de que alguno de estos parámetros no se corresponda con el tamaño que debe se procede a realizar nuevamente la lectura, esta tarea se repetiría de forma iterativa hasta que todos coincidan.

Más que por un problema del protocolo de comunicación en sí, este tipo de errores suelen deberse a una mala conexión del dispositivo por parte del usuario. Pero añade un punto de seguridad para la correcta inicialización del sistema.

2. Seguidamente estos datos se tratan para convertirlos a hexadecimal y poder usarlos en el dispositivo. Esta tarea se desarrolla en varios estados que tratan un dato concreto. Dicha distribución responde a la necesidad de tener una estructura legible y eficiente a la que el propio sistema pueda volver en caso de que algún dato no sea leído de forma correcta.

Con el objetivo de evitar volcar en el reloj calendario información defectuosa debida a una mala lectura si los datos contienen caracteres inadecuados se procedería a volver a la lectura para evitar configuraciones indeseadas. Se parte de un correcto archivo de configuración en la memoria conectada.

3. Posteriormente estos datos son los que se usan para dar valores a la inicialización del reloj calendario y límites de temperatura del sensor. Estas tareas se realizan en estados diferentes, los cuales al formar parte de la configuración del sistema no se vuelve a ir salvo que se proceda a una nueva configuración posterior por medio de un reset.

4. Una vez establecida esta inicialización el sistema va a un estado desde el cual se va a volver de manera iterativa tras almacenar información o mostrarla. Este estado será un punto recurrente del sistema al que siempre se acabará volviendo después de que se lleven a cabo una serie de tareas. El estado posterior a él vendrá determinado por que ocurra al sistema.
5. En el caso de que no suceda ninguna eventualidad desde el estado intermedio al que siempre se vuelve se procedería tras un tiempo dado en la configuración (tiempo de muestreo) a volver a tomar una muestra de temperatura. Sin embargo si se pulsa el botón del sistema lo que ocurrirá es que se irá a otro estado en el cual se mostrará el estado del sistema.
6. Si se pasa a un estado de tomar temperatura tras el tiempo de muestreo dado que no se ha pulsado en ningún momento el botón ocurre lo siguiente:
 - a. Tomar un dato de temperatura por medio del sensor. Siguiendo el protocolo requerido por el mismo. Este dato para poder escribirse debe de convertirse en una cadena de caracteres.
 - b. Comprueba que el dato obtenido se encuentra entre los límites dados por la configuración. En caso de no encontrarse contenido por dichos límites se registra la incidencia. Esto será posteriormente usado para mostrar el estado del registro de datos llevado a cabo por el dispositivo.
 - c. Se toma del reloj toda la información que en la configuración se ha especificado que debe aparecer en cada registro. Estos datos deben convertirse a cadenas de caracteres para poder escribirlos.
 - d. Se procede a escribir en la memoria USB tanto los parámetros temporales como el dato de temperatura, además de la validez del mismo.
 - e. Al estado intermedio donde tras el tiempo de muestreo exigido se procederá a repetir todo lo anterior en el caso de que siga sin pulsarse el botón.

7. Si por el contrario se ha accionado el botón el sistema parte a mostrar el estado de los datos tomados y la validez de los mismos de con indicadores luminosos. Además mientras se mantenga pulsado es seguro retirar la memoria USB sin temor de que pueda dañarse el fichero que contiene la información tomada hasta el momento.
8. Si es para ver cómo va el registro de datos se pasa a un estado donde en función del valor de una variable se muestra encendido un led verde o rojo. Indicando el verde que todos los datos tomados hasta el momento se encuentran dentro de los límites dados en la configuración y rojo si alguno de los datos no está contenido dentro de esos límites. De forma iterativa se comprueba si el accionamiento sigue activo hasta que este no sea accionado. En ese momento el sistema vuelve al estado que lo pone en bajo consumo, activando justo antes las interrupciones.
9. Igualmente mientras se muestra el estado de los datos registrados la memoria USB está lista para ser desconectada. Es importante señalar que dependiendo de las especificaciones de la configuración a partir de aquí el sistema puede o bien tomar una nueva configuración cuando detecte una memoria o seguir con el funcionamiento anterior.
10. En el caso de que la memoria quede desconectada del sistema este puede seguir tomando datos que no van a poder ser volcados a falta de memoria. Sin embargo la validación seguirá funcionando de modo que se puede seguir usando el sistema para comprobar si ha habido eventualidades aunque ya resulte imposible conocer el momento de las mismas.

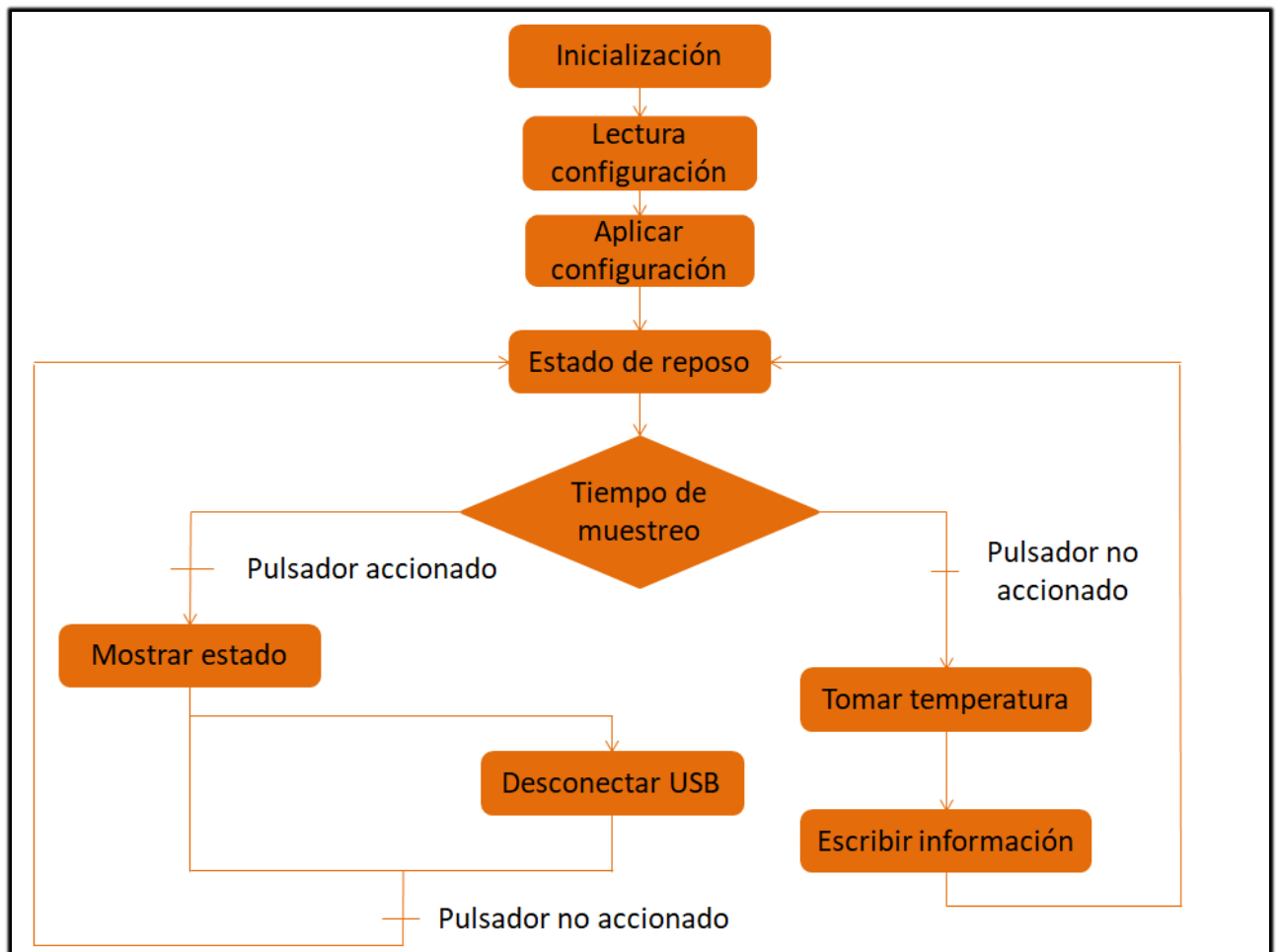


Diagrama 22. Diagrama de flujo

5.6.2. Máquina de estados

Una máquina de estados es considerado aquel sistema con entradas y salidas en el que las salidas son el resultado no solo de las entradas actuales, sino además de las anteriores. De modo que existe una respuesta según los eventos pasados, y no solo los que ocurren a cada momento.

De manera que una máquina de estados consta de una serie de estados que hacen de intermediario entre las entradas y las salidas del sistema, además el historial de señales de entrada resulta decisivo para la máquina. Dando lugar a que la salida dependa del estado y las entradas actuales.

Habitualmente una máquina de estados se representa con un diagrama de estados, aunque es posible representarla con un diagrama de flujo.

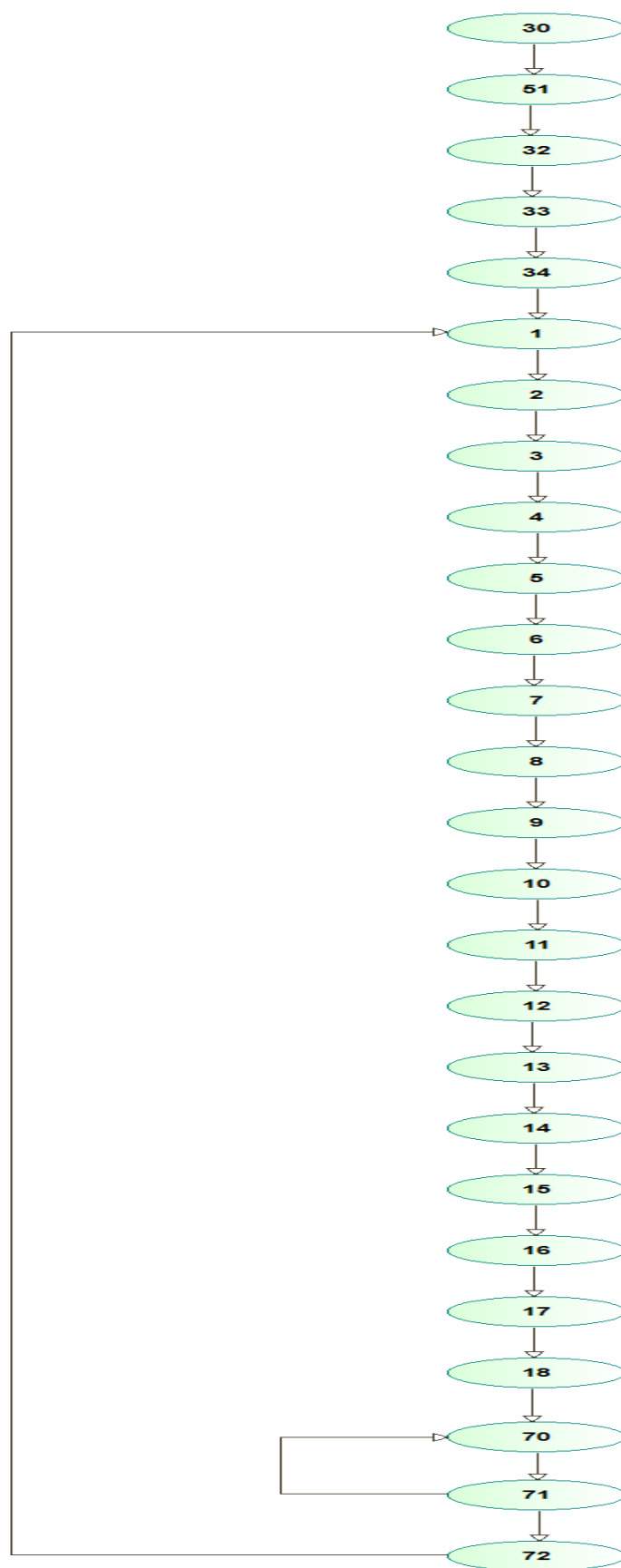


Diagrama 23. Máquina de estados

Estados de la máquina de estados:

Estado	Descripción
1	Adquisición de la información temporal a través del módulo RTCC.
2	Creación de la cadena de caracteres que va a escribirse en el año a partir del valor del año tomado del módulo RTCC.
3	Creación de la cadena de caracteres que va a escribirse en el mes a partir del valor del mes tomado del módulo RTCC.
4	Creación de la cadena de caracteres que va a escribirse en el día de la semana a partir del valor tomado del módulo RTCC.
5	Creación de la cadena de caracteres que va a escribirse en el día del mes a partir del valor tomado del módulo RTCC.
6	Creación de la cadena de caracteres que va a escribirse en la hora a partir del valor tomado del módulo RTCC.
7	Creación de la cadena de caracteres que va a escribirse en los minutos a partir del valor tomado del módulo RTCC.
8	Creación de la cadena de caracteres que va a escribirse en los segundos a partir del valor tomado del módulo RTCC.
9	Adquisición del dato de temperatura y conversión de milivoltios a grados kelvin.
10	Tratamiento de los datos de temperatura.
11	Creación de la cadena de caracteres que va a escribirse en las centenas de la temperatura a partir de los datos tomados y tratados.
12	Creación de la cadena de caracteres que va a escribirse en las decenas de la temperatura a partir de los datos tomados y tratados.
13	Creación de la cadena de caracteres que va a escribirse en las unidades de la temperatura a partir de los datos tomados y tratados.
14	Creación de la cadena de caracteres que va a escribirse en las decimas de la temperatura a partir de los datos tomados y tratados.
15	Creación de la cadena de caracteres que va a escribirse en las centésimas de la temperatura a partir de los datos tomados y tratados.
16	Conversión de los límites de cadena de caracteres a números.
17	Comparación de la temperatura adquirida con los límites para escribir si el dato es correcto o incorrecto.
18	Escritura en el archivo de registro de los parámetros recopilados.
30	Inicia la comunicación USB para crear el archivo de registro y confirmar que hay un dispositivo conectado.
32	Tratamiento de los parámetros referentes al tiempo de muestreo. Se convierten de cadenas de caracteres a números.
33	Convierte los datos temporales destinados a cargarse en el módulo RTCC de cadenas de caracteres a números.
34	Configuración del módulo RTCC con los datos tomados del archivo de configuración y puesta en marcha.
51	Lee la configuración de la memoria USB y crea y escribe el archivo "entrada.txt" en el que se reflejan los parámetros tomados tal cual fueron leídos.
70	Estado que comprueba si el botón ha sido pulsado entre cada segundo de la espera para tomar la siguiente muestra.

71	Espera de un segundo exacto para los tiempos de muestreo.
72	Contador para las esperas de los tiempos de muestreo.

5.6.3. Tratamiento de la información

El sistema diseñado realiza una serie de operaciones para el tratamiento de los datos leídos de la memoria USB. Las cadenas de caracteres se convierten en variables numéricas del tipo “*int*”. Estas variables se emplean para configurar el módulo RTCC, servir como valores de comparación para la verificación de los límites de temperatura del sensor o los tiempos de muestreo.

Para generar las cadenas de caracteres con la información del módulo RTCC que van a volcarse en la memoria USB se sigue el siguiente procedimiento:

- ❖ Se vuelca el valor del parámetro en una variable y en el caso de ser necesario se aplica una máscara para separar en dos variables distintas. Por ejemplo cuando se toman los minutos y segundos vienen en una misma variable y seguidamente se dividen en dos.
- ❖ Se realiza la tarea de ir realizando operaciones de comparación con la variable hasta que coincida para asignar a otra variable la cadena de caracteres con el texto que se corresponde con el parámetro temporal de ese número.
- ❖ Este proceso se repite cada vez que se va a realizar una escritura.

La temperatura se obtiene con el siguiente procedimiento:

- ❖ Tras la conversión ADC de la entrada del sensor y el posterior tratamiento matemático de la señal para convertir los milivoltios a kelvin y grados centígrados.
- ❖ Se realiza un proceso muy similar al de los datos temporales pero con la temperatura en grados centígrados.
- ❖ Para la comparación con los límites superior e inferior se procede a expresar tanto la medida como los límites en milikelvin y según

proceda se introduce una cadena de caracteres u otra en la variable de validación que posteriormente se vuelca en la memoria.

- ❖ En el caso de que se realice un registro de variable incorrecta se pone a nivel alto un “*flag*”, de modo que cuando se solicite conocer el estado del sistema lucirá el led rojo.

Durante la espera entre muestra y muestra a cada segundo se consulta si el botón para mostrar el estado del sistema ha sido accionado. En caso afirmativo el sistema entra en un bucle de espera mientras se muestra por medio de los leds el estado hasta que deja de pulsarse el botón.

Anexo I. Manual de uso del sistema

Para el correcto funcionamiento del sistema desarrollado hay que tener en cuenta que debe de introducirse un archivo de configuración que recoja los datos necesarios para su correcto funcionamiento. En caso de que tener una ausencia de alguno de estos datos el sistema está preparado para responder de distintas formas, asegurando siempre un funcionamiento correcto dentro de las limitaciones que la falta de una configuración correcta le permita.

El archivo de configuración puede crearse con cualquier bloc de notas que genere un archivo con el formato “.txt”. El formato que se muestra en la “Figura 8” solo tiene como parte útil la primera línea. Sin embargo se recomienda escribir el resto de líneas como forma de tener controlado que es lo que se está introduciendo y sus equivalencias respectivas. Esto viene por el hecho de que el sistema solo va a tomar los datos de la primera línea, y el resto sirve como referencia para el usuario.

Los datos temporales se agrupan en dos menos el año, esto se hace por simplicidad, ya que el módulo RTCC del sistema lee los datos de esa misma manera. El hecho de crear el título en mayúscula daría lugar a una apertura incorrecta del archivo y por lo tanto una incorrecta inicialización.

```
2017050706174025303273201

Año:                2017
Mes y día del mes:   0507
Día de la semana y hora: 0617
Minutos y segundos:  4025
Límite superior:     303(kelvin)
Límite inferior:     273(kelvin)
Tiempo de muestreo:  20
Escala de tiempos:   1(segundos)
```

Figura 9. Archivo de configuración

En el caso de querer guardar en la misma memoria registros anteriores debe de alterarse el nombre con el que fue creado o el nuevo registro lo sustituirá. Lo mismo ocurrirá con el archivo que recoge que es lo que ha leído el sistema.

El fichero que el sistema leerá para tomar la configuración debe de llamarse “config”. Los datos que recoge son los siguientes:

- ❖ Año: El sistema está limitado por el propio módulo RTCC interno del microcontrolador para fechas de 2.000 a 2.099.
- ❖ Mes y día del mes: Los meses se cuentan de 01 a 12 y los días de 01 a 31. Se debe de tener en cuenta que el módulo RTCC al que va esta configuración reconoce cuando un mes no tiene más de 30 días o el caso excepcional de Febrero. En caso de que se le suministre una información imposible se producirán anomalías en la fecha suministrada.
- ❖ Día de la semana y hora: Los días de la semana va de 0, siendo el lunes, a 6, siendo el domingo. La hora está configurada a 24 horas de modo que no hay que especificar si son de la mañana o la tarde.
- ❖ Minutos y segundos: Los minutos y segundos van de manera convencional de 0 a 59.
- ❖ Tiempo de muestreo: se introduce un tiempo de muestreo que puede ser de segundos, minutos u horas. Debe de tenerse en cuenta que el total del tiempo de muestreo se introduce en la misma unidad y no por partes. El tiempo de muestreo mínimo es de 1 segundo.
- ❖ Límite superior de temperatura: viene dado por el propio límite del sensor de 100°C para el calibrado realizado. La conversión a decimal se hace desde la medida en grados kelvin.
- ❖ Límite inferior de temperatura: viene dado por el límite inferior del propio sensor que es de -40°C para el calibrado realizado. La conversión a decimal se hace desde la medida en grados kelvin.
- ❖ Escala de tiempos: Se especifica con un 1 si se trata de segundos, 2 minutos o 3 horas el tiempo de muestreo especificado.

Aparte de la configuración para iniciar el sistema debe de accionarse durante unos breves instantes el pulsador de reset para que el sistema se inicie correctamente y en el caso de que hubiese sido usado con anterioridad no conserve

configuraciones antiguas. De igual forma si lo que se ha hecho es retirar la memoria USB y se desea volver a insertarla para que continúe funcionando lo único que hay que hacer es volver a insertarla mientras se pulsa el botón que permite mostrar el estado del sistema, de este modo no habrá posibilidad que se pueda corromper la información del archivo de registros.

Cuando se desee retirar la memoria USB del sistema lo único que hace falta es pulsar el botón destinado para este fin. El cual además de permitir que retirar la memoria de forma segura hace que uno de los dos leds de sistema se encienda, permitiendo conocer el estado del sistema. En caso de que los datos registrados se encuentren dentro de los límites dados en la configuración se encenderá el led verde, si por el contrario alguno de los datos tomados se encuentra por encima o por debajo de dichos límites será el led rojo el que se ilumine.

07/mayo/2017	-	domingo	-	17:40:25	-	temperatura: 027.20 celsius	-	correcto
07/mayo/2017	-	domingo	-	17:40:26	-	temperatura: 028.04 celsius	-	correcto
07/mayo/2017	-	domingo	-	17:40:27	-	temperatura: 028.22 celsius	-	correcto
07/mayo/2017	-	domingo	-	17:40:28	-	temperatura: 028.50 celsius	-	correcto
07/mayo/2017	-	domingo	-	17:40:29	-	temperatura: 028.50 celsius	-	correcto
07/mayo/2017	-	domingo	-	17:40:30	-	temperatura: 028.77 celsius	-	correcto
07/mayo/2017	-	domingo	-	17:40:31	-	temperatura: 028.98 celsius	-	correcto
07/mayo/2017	-	domingo	-	17:40:32	-	temperatura: 029.10 celsius	-	correcto
07/mayo/2017	-	domingo	-	17:40:33	-	temperatura: 029.25 celsius	-	correcto
07/mayo/2017	-	domingo	-	17:40:33	-	temperatura: 029.45 celsius	-	correcto
07/mayo/2017	-	domingo	-	17:40:34	-	temperatura: 029.50 celsius	-	correcto
07/mayo/2017	-	domingo	-	17:40:35	-	temperatura: 029.64 celsius	-	correcto
07/mayo/2017	-	domingo	-	17:40:36	-	temperatura: 029.78 celsius	-	correcto
07/mayo/2017	-	domingo	-	17:40:37	-	temperatura: 030.00 celsius	-	correcto
07/mayo/2017	-	domingo	-	17:40:38	-	temperatura: 030.60 celsius	-	correcto
07/mayo/2017	-	domingo	-	17:40:39	-	temperatura: 031.45 celsius	-	incorrecto
07/mayo/2017	-	domingo	-	17:40:40	-	temperatura: 031.67 celsius	-	incorrecto
07/mayo/2017	-	domingo	-	17:40:41	-	temperatura: 032.22 celsius	-	incorrecto
07/mayo/2017	-	domingo	-	17:40:42	-	temperatura: 032.34 celsius	-	incorrecto
07/mayo/2017	-	domingo	-	17:40:42	-	temperatura: 033.44 celsius	-	incorrecto
07/mayo/2017	-	domingo	-	17:40:44	-	temperatura: 033.46 celsius	-	incorrecto

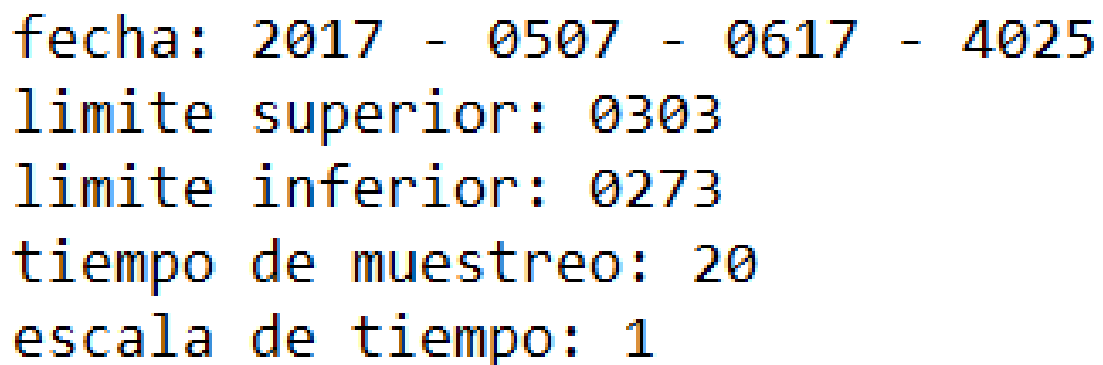
Figura 10. Archivo de datos tomados generado por el sistema

Además en la memoria se habrá generado un archivo .txt llamado "REGISTRO", en el cual estarán todos los datos de temperatura tomados por el sistema, en que momento fueron tomados y si con correctos o no. Este fichero es de gran interés ya que es lo que permite tener un seguimiento exhaustivo de las eventualidades acontecidas al sistema. En este archivo se muestran todos los datos temporales en el momento en el que se adquirió la temperatura, la propia temperatura y la unidad en la que se expresa, además de si el dato en cuestión es correcto o no de acuerdo con los parámetros establecidos.

El hecho de que cualquiera de las medidas de temperatura fuera incorrecta, aparte de reflejarse en el fichero se vería en que el led rojo se encendería al pulsar el botón.

Finalmente hay otro fichero que el sistema genera por una cuestión de seguridad y que sirve como verificación de que todos los datos tomados en la configuración han sido leídos correctamente. Documento de comprobación lo genera el sistema y se llama “entrada.txt”, él se copian todos los parámetros de configuración. En caso de querer conocer cuál es la última configuración insertada en el sistema este archivo sirve para tal fin.

El archivo lo que tiene es directamente la lectura que realiza el sistema cuando toma la configuración y tal cual la vuelva en el nuevo archivo “entrada.txt”. De modo que aún no ha convertido la variable de una cadena de caracteres a valores numéricos con los que sí puede trabajar el módulo RTCC para la hora o los límites de temperatura para las comprobaciones.

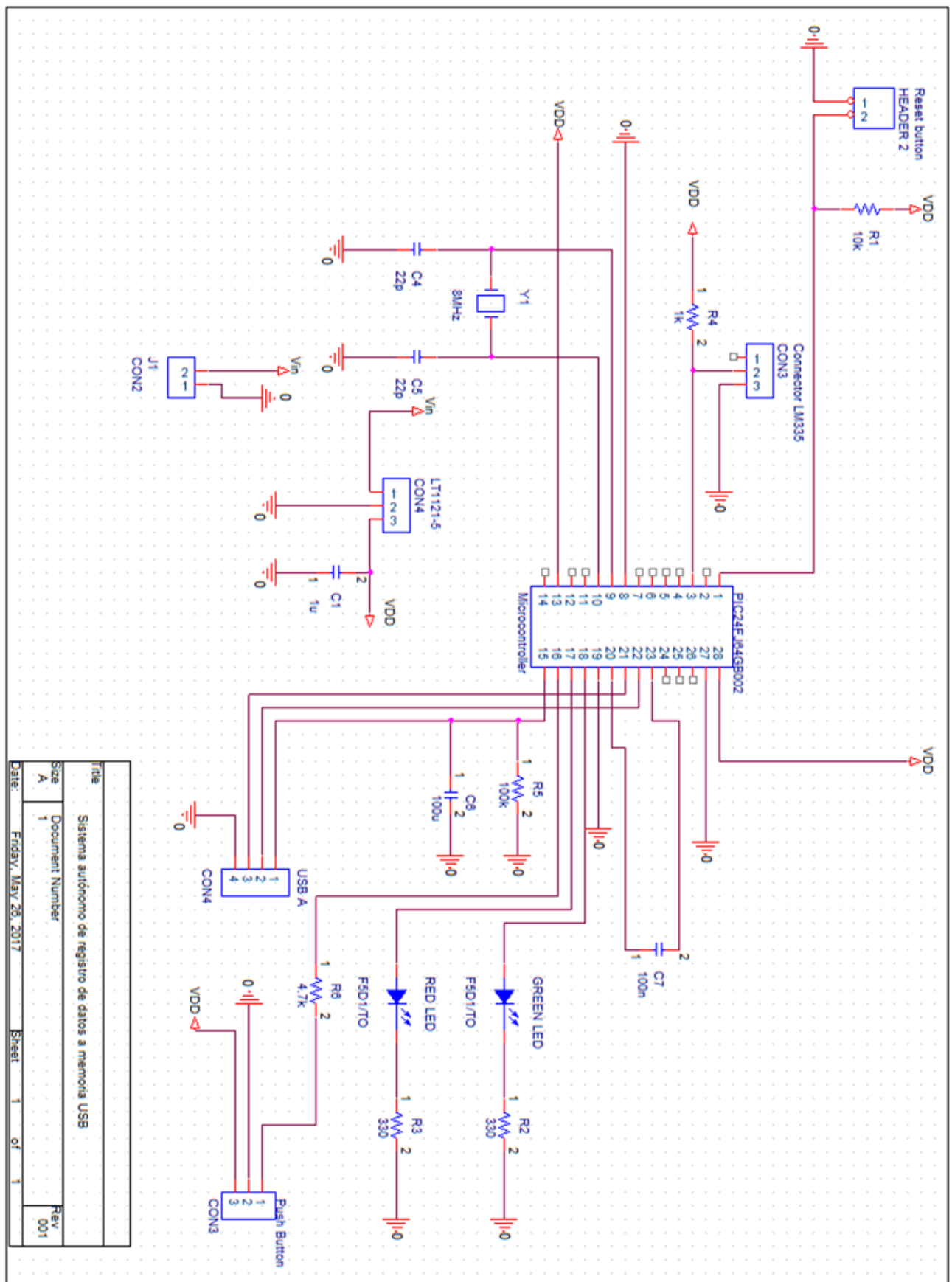


```
fecha: 2017 - 0507 - 0617 - 4025
limite superior: 0303
limite inferior: 0273
tiempo de muestreo: 20
escala de tiempo: 1
```

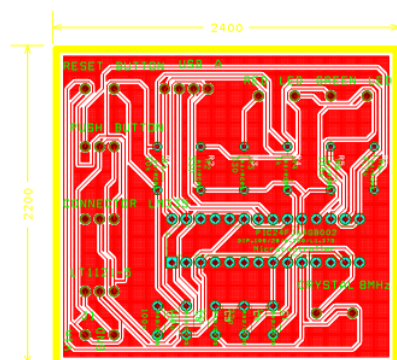
Figura 11. Datos de entrada leídos

Como recordatorio, se señala nuevamente que los archivos “*entrada.txt*” y “*registro.txt*” se verán sustituidos por otros si el sistema se vuelve a iniciar tras un reset.

Anexo II. Planos



Layout:

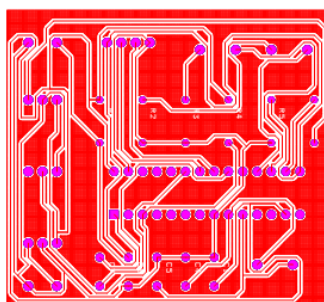


DRILL CHART				
SYM	DIAM	TOL	QTY	NOTE
×	0.028		12	
+	0.034		38	
◇	0.038		23	
TOTAL			73	

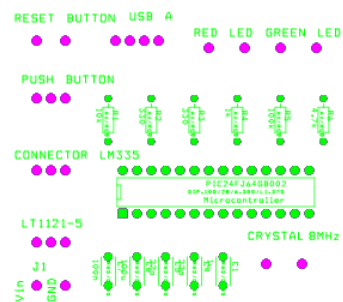
Drills:



Bottom:



Componentes:



Apéndice I. Términos

- ❖ **MCC:** Microchip Code Configurator. Herramienta generadora de código que forma parte del entorno de MPLAB X IDE.
- ❖ **MLA:** Microchip Legacy Applications. Ejemplos funcionales suministrados por Microchip de distintas aplicaciones.
- ❖ **QFN:** Quad-Flat-No-Leads.
- ❖ **SOIC:** Small Outline Integrated Circuit.
- ❖ **SSOP:** Shrink Small Outline Package.
- ❖ **SPDIP:** Shrink Plastic Dual In-line Package.
- ❖ **TQFP:** Thin Quad Flat Package.
- ❖ **ADC:** Analog to Digital Converter. Conversión de datos analógicos a digitales.
- ❖ **RTCC:** Real Time Clock Calendar. Reloj calendario.

Apéndice II. Referencias de documentos

Microchip Technology Inc. (2010). *PIC24FJ64GB004 Family Data Sheet* [On-line]. Disponible: <http://ww1.microchip.com/downloads/en/DeviceDoc/39940d.pdf> [18/05/2017].

Microchip Technology Inc. (2010). *Section 27. USB On-The-Go* [On-line]. Disponible: <http://ww1.microchip.com/downloads/en/DeviceDoc/39721b.pdf> [18/05/2017].

Microchip Technology Inc. (2007). *Section 29. Real-Time Clock Calendar (RTCC)* [On-line]. Disponible: <http://ww1.microchip.com/downloads/en/DeviceDoc/39696b.pdf> [18/05/2017].

Microchip Technology Inc. (2011). *Microstick II Information Sheet* [On-line]. Disponible: <http://ww1.microchip.com/downloads/en/DeviceDoc/51951B.pdf> [18/05/2017].

Microchip Technology Inc. (2013-2014). *MPLAB Code Configurator User's Guide* [On-line]. Disponible: <http://ww1.microchip.com/downloads/en/DeviceDoc/40001725B.pdf> [18/05/2017].

Microchip Technology Inc. (2016). *16-bit Microcontrollers and Digital Signal Controllers* [On-line]. Disponible: <http://ww1.microchip.com/downloads/en/DeviceDoc/00001032p.pdf> [18/05/2017].

Microchip Technology Inc. (2016). *MPLAB® XC16 C Compiler User's Guide* [On-line]. Disponible: <http://ww1.microchip.com/downloads/en/DeviceDoc/50002071F.pdf> [18/05/2017].

Linear Technology (1994). *LT1121/LT1121-3.3/LT1121-5 Micropower Low Dropout Regulators with Shutdown* [On-line]. Disponible: <http://cds.linear.com/docs/en/datasheet/1121fg.pdf> [18/05/2017].

Texas Instruments (2015). *LMx35, LMx35A Precision Temperature Sensors* [On-line]. Disponible: <http://www.ti.com/lit/ds/symlink/lm335.pdf> [18/05/2017].

Apéndice III. Referencias de páginas web

2017 Microchip Technology, Inc. "Analog to Digital Converter (ADC)". Internet: <http://microchipdeveloper.com/8bit:emr-adc>, [18/05/2017].

Microchip Technology, Inc. "6-bit PIC24 MCUs and dsPIC[®] DSCs". Internet: <http://www.microchip.com/design-centers/16-bit>, [18/05/2017].

Wikipedia. "Microcontrolador PIC". Internet: https://es.wikipedia.org/wiki/Microcontrolador_PIC, [18/05/2017].

Wikipedia. "Máquina de estados". Internet: https://es.wikipedia.org/wiki/M%C3%A1quina_de_estados, [08/06/2017].

Apéndice IV. Lista de materiales

Cantidad	Descripción	Precio unitario	Precio total	Código
1	Terminal de rosca para PCB 2 vías, 2.54mm	0.608€	0.608€	220-4260
1	Terminal de rosca para PCB 3 vías, 2.54mm	0.888€	0.888€	220-4276
1	Placa de cobre para PCB, AE16, Doble cara, FR4, grosor 35µm, 100 x 160 x 1.6mm	4.16€	4.16€	219-2139
1	Regulador LDO, 150mA 5V, TO-92 3 pines, entrada 6 to 20V	2.042€	2.042€	822-6847
1	Sensor de temperatura LM335AZ/NOPB, encapsulado TO-92 3 pines, interfaz analógico	1.478€	1.478€	761-5634
1	PIC16 bits 8kb RAM, 64kb Flash, SPDIP 28 pines 32MHz USB, PIC24FJ64GB002	3.95€	3.95€	703-7602
1	Unidad de cristal 8MHz, ±32ppm, HC-49-4H-SMX, 2 pin	1.49€	1.49€	130-8838
2	Interruptor de botón pulsador en miniatura negro	2.84€	5.68€	133-6502
1	Led Kingbright, Montaje en orificio pasante, Verde, 568nm, 20mcd, 2V, 60°C, 5mm (T-1 3/4)	0.10€	0.10€	228-6004
1	Led Kingbright, Montaje en orificio pasante, Rojo, 660nm, 5mcd, 2.25V, 60°C, 5mm (T-1 3/4)	0.172€	0.172€	228-5988
2	Resistencia fija de película de carbono, TE Connectivity, 330Ω, ±5%, 1W, Axial, Serie CFR100	0.042€	0.084€	131-801
1	Resistencia fija de película fina, Vishay, 1kΩ, ±1%, 0.4W, Axial, Serie MBA0204	0.14€	0.14€	157-446
1	Resistencia fija de película de carbono, TE Connectivity, 10kΩ, ±5%, 0.5W, Axial, Serie CFR50	0.04€	0.04€	132-731
1	Resistencia fija de película de carbono, TE Connectivity, 100kΩ, ±5%, 1W, Axial, Serie CFR100	0.086€	0.086€	131-918
1	Resistencia fija de película metálica, TE Connectivity, 4.7kΩ, ±1%, 0.6W, Axial, Serie	0.048€	0.048€	148-663

	LR1			
1	Zócalo USB tipo A TE Connectivity 292303-7, hembra, 1 puerto, Ángulo de 90°, montaje en PCB, Versión 2.0	2.52€	2.52€	909-7864
2	Condensador cerámico monocapa SLCC Murata, 22pF, $\pm 5\%$, 6.300V dc, orificio pasante, SL dieléctrico	0.434€	0.434	721-5101
1	Condensador de poliéster PET WIMA, 100nF, $\pm 10\%$, 100Vdc, 63Vac, orificio pasante	0.362€	0.362€	108-2700
1	Condensador de poliéster PET KEMET, 1 μ F, $\pm 5\%$, 40Vdc, 63Vac, orificio pasante	0.65€	0.65€	823-998
1	Condensador electrolítico de aluminio Yageo SK063M0100B5S-1012, 100 μ F, $\pm 20\%$, 63V dc, serie SK	0.228€	0.228€	440-6575
	Total materiales		25.594€	

Todos los códigos de referencia pertenecen a RS-Online y han sido consultados en el 29/05/2017.