



Escola Politècnica Superior
de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

TREBALL DE FI DE CARRERA

TÍTOL: PROGRAMACIÓN DE UN ALGORITMO DE BAJO CONSUMO PARA MEDIR INTERVALOS DE TIEMPO MEDIANTE EL MSP430

TITULACIÓ: Enginyeria Tècnica de Telecomunicació, especialitat Sistemes de Telecomunicació

AUTORS: Fernando Manzano Rubio y Jean Marc Lloveras Lladó

DIRECTOR: Óscar López Lapeña

DATA: 27 de julio de 2009

Título: PROGRAMACIÓN DE UN ALGORITMO DE BAJO CONSUMO PARA MEDIR INTERVALOS DE TIEMPO MEDIANTE EL MSP430

Autores: Fernando Manzano Rubio y Jean Marc Lloveras Lladó

Director: Óscar López Lapeña

Data: 27 de julio de 2009

Resum

El documento que se dispone a leer se titula "PROGRAMACIÓN DE UN ALGORITMO DE BAJO CONSUMO PARA MEDIR INTERVALOS DE TIEMPO MEDIANTE EL MSP430".

Hoy en día empieza a ser una necesidad la aplicación de procesos controlados por microcontroladores destinados a bajo consumo, sobre todo en el campo de los sistemas autónomos.

Se programará en un microcontrolador un algoritmo destinado a medir tiempos con la máxima resolución, precisión y exactitud posible teniendo en cuenta siempre minimizar al máximo el consumo del sistema.

Previamente se estudiarán los posibles problemas que se encuentran al medir tiempos con un microcontrolador y los métodos más adecuados para ahorrar en el consumo. También se estudiará la técnica de Dynamic Power Management con la cual se implementará el sistema propuesto.

También se estudiarán las características principales que debe tener nuestro microcontrolador para poder llevar a cabo el proyecto, como por ejemplo la posibilidad de trabajar dinámicamente con varios relojes o la posibilidad de activar varios modos de bajo consumo.

Finalmente se mostrarán los resultados obtenidos, se analizará los logros conseguidos con el algoritmo propuesto y se propondrá nuevos caminos para la mejora y ampliación del sistema.

Título: PROGRAMACIÓN DE UN ALGORITMO DE BAJO CONSUMO PARA MEDIR INTERVALOS DE TIEMPO MEDIANTE EL MSP430

Autores: Fernando Manzano Rubio y Jean Marc Lloveras Lladó

Director: Óscar López Lapeña

Data: 27 de julio de 2009

Overview

Later it is presented the final project of degree called "PROGRAMMING OF AN ALGORITHM OF LOW CONSUMPTION TO MEASURE INTERVALS OF TIME BY MEANS OF THE MSP430".

Nowadays it starts being a need the application of processes controlled by microcontrollers destined for low consumption, especially in the field of the autonomous systems.

An algorithm destined to measure times with the maximum resolution, precision and possible accuracy will be programmed in a microcontroller, bearing in mind always to minimize to the maximum the consumption of the system.

Previously it will be studied the possible problems that can be found when measuring times with a microcontroller and the most suitable methods to save in the consumption. It will also be studied the Dynamic Power' Management technique with which the proposed system will be implemented.

It will also be studied the main characteristics required by our microcontroller in order to be able to carry out the project, as for example the possibility of working dynamically with several clocks or the possibility of activating several ways of low consumption.

Finally, the obtained results will be shown, as well as it will be analyzed the achievements attained with the proposed algorithm, and it will be proposed new ways to improve and extend the system.

AGRADECIMIENTOS

A nuestras familias, Isabel, Elena y nuestros amigos por su apoyo incondicional en todo momento.

A Óscar López por habernos dado la posibilidad de hacer este proyecto, por sus correcciones y su guía.

Y a todo aquel que haya participado de una forma u otra en este proyecto.

Gracias.

ÍNDICE

CAPÍTULO 1. INTRODUCCIÓN	1
1.1. ANTECEDENTES	1
1.1.1. Medida de tiempos.....	1
1.1.2. Sistemas de bajo consumo	2
1.2. OBJETIVOS DEL PROYECTO	3
1.2.1. Diseño de un sistema para medida de tiempos basado en un microcontrolador	3
1.3. ESTRUCTURA DE LA MEMORIA.....	4
CAPITULO 2. CONVERSIÓN TIEMPO-DIGITAL.....	5
2.1. SISTEMAS DE MEDIDA.....	5
2.1.1. Contador universal.	5
2.1.2. Medida mediante microcontroladores.....	6
2.2. FUENTES DE INCERTIDUMBRE EN UN SISTEMA BASADO EN MICROCONTROLADORES.....	7
2.2.1. La inestabilidad de la frecuencia de referencia.	7
2.2.2. Error de cuantificación.....	8
2.2.3. Error causado por el ruido de disparo.	10
2.3. INTERFERENCIAS EN LA ALIMENTACIÓN	11
2.4. ACTIVIDAD DE LA CPU	11
CAPÍTULO 3. AHORRO ENERGÉTICO EN MICROCONTROLADORES.....	13
3.1. CONSUMO DE UN MICROCONTROLADOR	13
3.1.1. Potencia disipada por un microcontrolador	13
3.1.2. Relación entre frecuencia de trabajo y tensión de alimentación	14
3.1.3. Compromiso entre el error de cuantificación y la potencia disipada	14
3.2. CONTROL DINÁMICO DE POTENCIA (DPM)	16
3.2.1. Control dinámico de la frecuencia (DFS).....	17
3.2.2. Control dinámico de la tensión (DVS)	18
CAPÍTULO 4. PLANTEAMIENTO DEL PROYECTO	21
4.1. SELECCIÓN DEL MICROCONTROLADOR	21
4.1.1. Criterios de selección.....	21
4.1.2. Microcontroladores de bajo consumo en el mercado.....	22
4.1.3. Selección de los microcontroladores MSP430.....	24
4.2. CRITERIOS PARA EL DISEÑO DE UN ALGORITMO DE MEDIDA.....	28
4.2.1. Detección del evento externo	28
4.2.2. Control dinámico de la frecuencia del reloj.....	28

4.2.3. Control dinámico del modo de funcionamiento de la CPU	29
4.3. ESTRATEGIA DE MEDIDA	30
4.3.1. Descripción del método para hacer la medida	30
CAPÍTULO 5. PROGRAMACIÓN DEL MSP430	39
5.1. ESTRUCTURA DEL PROGRAMA	39
5.2. DESCRIPCIÓN DE LAS INTERRUPCIONES	40
CAPÍTULO 6. RESULTADOS EXPERIMENTALES	43
6.1. PROCEDIMIENTO DE MEDIDA	43
6.2 RESULTADOS SIN CONTROL DINÁMICO DE POTENCIA.....	46
6.2.1 ACLK.....	46
6.2.2 SMCLK (DCO).....	49
6.3 RESULTADOS CON CONTROL DINÁMICO DE POTENCIA	51
CAPÍTULO 7. CONCLUSIONES.....	55
BIBLIOGRAFÍA	57
ACRÓNIMOS.....	59
ANEXO A. APLICACIÓN PANEL SOLAR.....	61
ANEXO B. ESTABILIDAD DCO.....	63
ANEXO C. ALGORITMO DISEÑADO	65
ANEXO D. DATASHEETS.....	71

CAPÍTULO 1. INTRODUCCIÓN

1.1. ANTECEDENTES

1.1.1. Medida de tiempos

El tiempo es una magnitud física común en cualquier proceso físico de la naturaleza. Su medida directa nos proporciona información de la duración de un determinado proceso y se realiza comparándola con otro proceso físico de duración conocida. Así por ejemplo, el tiempo que tarda en recorrer un atleta cien metros puede medirse mediante un cronómetro que cuenta los períodos de un reloj de cuarzo de frecuencia conocida. La exactitud en la medida dependerá de la estabilidad y tolerancia de la frecuencia del reloj de cuarzo, así como de la capacidad para determinar cuándo empieza y termina el evento a medir. Las prestaciones requeridas del sistema de medida dependerán de los objetivos de la medida y de la naturaleza del evento. No es lo mismo cronometrar el tiempo que tarda un glaciar en avanzar un metro que el tiempo que un condensador tarda en cargarse. Pero en todo caso, se obtiene un valor a partir de un dispositivo concebido para medir esta magnitud.

A partir de la medida directa del tiempo se puede obtener indirectamente la medida de otras variables físicas, es lo que se conoce como medidas indirectas. Así por ejemplo, podemos medir la temperatura de un objeto a partir de la duración de una señal. En la figura 1.1 se muestra un posible circuito donde un microcontrolador estima la temperatura a partir del tiempo de carga de un condensador (C). La temperatura determina el valor de la resistencia de carga (R) y consecuentemente, cuánto tiempo tarda en cargarse el condensador.

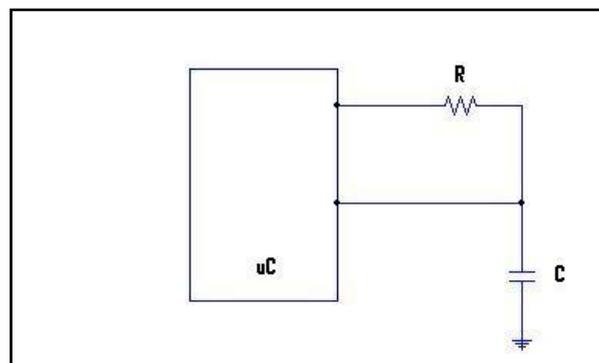


Figura 1.1. Medida indirecta de la temperatura.

1.1.2. Sistemas de bajo consumo

Hoy en día cada vez es más necesario diseñar sistemas con el mayor ahorro energético posible, y sobre todo cuando se trata de sistemas autónomos cuya fuente de energía está limitada. El ejemplo más intuitivo que actualmente está en pleno uso son los sensores autónomos. Los sensores ambientales por ejemplo están colocados en el exterior sin la posibilidad de tener una alimentación continua. Para solucionar este problema es necesario recurrir a varias estrategias para que dicho sistema pueda realizar sus funciones sin que se vea afectado por la falta de energía.

Las posibles fuentes de energía de un sistema autónomo podrían ser varias. A continuación se propone algunos ejemplos:

Pilas no recargables: A pesar de tener como principal inconveniente su capacidad limitada que conduce a tener que sustituirlas periódicamente, es el método más rápido y sencillo de utilizar. En todo caso es la mejor opción hasta hoy en día para sistemas que consuman muy poco y permitan una fácil sustitución de las pilas.

Energía solar: Es una de las fuentes de energía renovable más común hoy en día para sistemas de muy bajo consumo y está ganando protagonismo como fuente de energía en general. Con objeto de minimizar su tamaño y precio, se precisa que el sistema electrónico que alimente sea de muy bajo consumo y se requiere una batería que alimente el sistema en ausencia de luz solar

Energía de los árboles: Se ha descubierto últimamente que los árboles liberan pequeñas cargas eléctricas que serían capaces de alimentar puntualmente a un sensor. Ya está en funcionamiento una red de sensores anti-incendio para bosques los cuales son alimentados exclusivamente por la energía de los árboles. Es un método que también implica el uso de una batería recargable, cuya vida útil es finita.

Células Peltier: Son unos dispositivos termoelectrónicos que se caracterizan por generar una diferencia de potencial eléctrico frente un flujo de calor causado por un diferencial de temperatura entre la dos caras del semiconductor. Se puede obtener energía en sitios donde haya un flujo de calor, como por ejemplo, un ser vivo cuya temperatura es superior al ambiente y radia continuamente calor

Energía a partir de piezoeléctricos: Se puede aprovechar las propiedades de los piezoeléctricos insertándolos en objetos que puedan estar sometidos a deformaciones físicas, como por ejemplo zapatos, mochilas, alfombras...

Aprovechamiento de la energía cinética: Consiste en aprovechar la energía cinética derivada del movimiento de un cuerpo en energía eléctrica. Un ejemplo reciente es el freno regenerativo que se ha instalado en algunos coches de F1 (KERS) que aprovechando sus frenadas se consigue tener en ciertos momentos potencia extra.

No sólo se trata de obtener una fuente de energía continua y suficiente, sino que también es necesario limitar el consumo al máximo. Por lo tanto, un sistema autónomo debe ser capaz de gestionar la energía disponible, consumiendo lo menos posible y operando lo mejor posible.

1.2. OBJETIVOS DEL PROYECTO

1.2.1. Diseño de un sistema para medida de tiempos basado en un microcontrolador

El objetivo del proyecto es el diseño y la implementación de un sistema electrónico basado en un microcontrolador para medir intervalos de tiempo de una señal con mínimo consumo.

Los resultados obtenidos serán útiles para el diseño de nodos sensores autónomos. Estos sistemas utilizan microcontroladores y suelen medir el tiempo como medidas indirectas de otras magnitudes físicas. Convencionalmente, la precisión de la medida supone utilizar relojes de muy alta frecuencia que condicionan el consumo y por tanto, la autonomía del nodo.

El proyecto pretende abordar el compromiso de diseño que existe en sistemas convencionales entre la precisión de la medida y el consumo. A diferencia de los sistemas convencionales, donde se utiliza un único reloj durante toda la medida, este trabajo propone utilizar dos relojes, uno de baja frecuencia y consumo, y el otro de alta frecuencia con un consumo mayor. La idea consiste en aprovechar la precisión del reloj de alta frecuencia y el bajo consumo del de baja.

El sistema propuesto implica utilizar microcontroladores que permitan un control dinámico de la frecuencia, permitiendo el cambio entre diferentes estados de consumo y actividad de procesado. Concretamente, se ha optado por el MSP430 de Texas Instruments que dispone de cinco niveles de consumo diferentes y permite activar o desactivar dinámicamente hasta cuatro relojes.

1.3. ESTRUCTURA DE LA MEMORIA

Este proyecto está enfocado a un contexto concreto, presente hoy en día en el ámbito de la tecnología. Este contexto se describe en el primer capítulo, "Introducción". En él se hace referencia a la utilidad que tiene el hecho de medir períodos de tiempo. Por otro lado se introduce el tema del ahorro energético, pero enfocado a sistemas autónomos. A partir de estas ideas se plantean los objetivos de este proyecto en el siguiente apartado de este capítulo, enfocados a la medida de tiempos con un microcontrolador.

A continuación viene el cuerpo del proyecto. Esta parte se puede dividir en dos bloques bien diferenciados.

Por un lado se encuentra un bloque a nivel teórico que engloba los capítulos 2 "Problemas en la medida de tiempos", 3 "Ahorro energético en microcontroladores" y 4 "Planteamiento del proyecto". En primer lugar se detallan los problemas que se presentan a la hora de tomar medidas de tiempo y sus posibles soluciones. En segundo lugar se explican las técnicas para el ahorro de energía en los microcontroladores. A continuación se hace un planteamiento del proyecto basado en este estudio; este planteamiento consiste en una búsqueda del mejor microcontrolador para realizar la medida, unos criterios a tener en cuenta en el diseño del algoritmo y una explicación a grandes rasgos de la estrategia para realizar la medida.

Por otro lado la memoria se centra en un nivel más práctico que engloba los capítulos 5 "Programación del MSP430" y 6 "Resultados experimentales". En estos capítulos se explica el algoritmo diseñado para hacer la medida, los procedimientos que se llevan a cabo para obtener los resultados y una valoración de éstos. También se hace una comparación entre los resultados obtenidos con el algoritmo diseñado y los resultados obtenidos con otros algoritmos.

Para terminar se explican las conclusiones sacadas del proyecto, entre las cuales están las posibles mejoras del sistema y ampliaciones de cara a un futuro.

CAPITULO 2. CONVERSIÓN TIEMPO-DIGITAL

2.1. SISTEMAS DE MEDIDA

El proceso de medir un período de tiempo mediante un sistema digital se denomina conversión tiempo – digital. Existen básicamente dos opciones para plantear el sistema de medida: el contador universal y el microcontrolador.

2.1.1. Contador universal.

Son unos sistemas electrónicos digitales que miden la frecuencia de una señal de entrada. También pueden dar valores de período, relación de frecuencia entre dos señales entrantes o el intervalo de tiempo entre dos eventos.

El sistema se basa en un reloj de periodo conocido que sirve como base de tiempos en la medida. Se cuenta el número de ciclos de la señal a medir durante el periodo de reloj. De esta forma, se puede calcular la frecuencia dividiendo el número de ciclo, n , por el periodo, t .

$$f = \frac{n}{t} \text{ (Hz)} \quad (2.1)$$

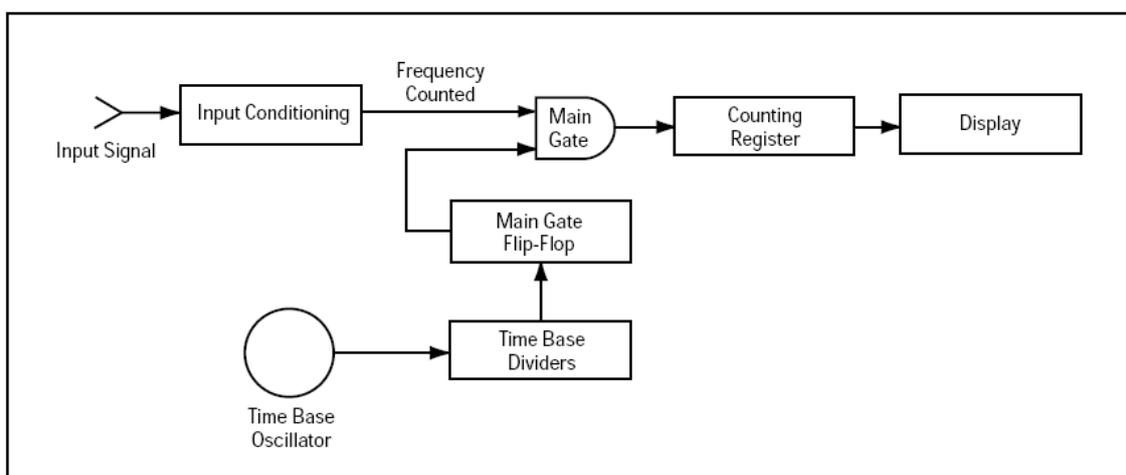


Figura 2.1. Diagrama del contador universal para medidas de frecuencia.

Hay que tener en cuenta que la señal que entra en el dispositivo debe ser adaptada para ser compatible con la circuitería interior. De ahí el bloque de la figura 2.1. en la entrada justo después de entrar la señal. Seguidamente la puerta principal controla el paso de la señal y el registro contador la contabiliza. El tiempo en que la puerta principal pasa de abierta a cerrada es controlado por el Time Base. Por norma general estos aparatos tienen como oscilador un cristal de cuarzo con frecuencias de 1,5 a 10 MHz que tienen como ventaja principal la estabilidad. El Time Base toma como frecuencia de referencia la del oscilador, y con dicha señal cuenta la frecuencia a la cual oscila la señal a medir. El resultado se ve reflejado en el display del dispositivo.

Con esta alternativa se obtienen resultados muy exactos pero en cambio su coste es muy elevado, cosa que hace de esta opción poco recomendable para medidas que no sean de laboratorio.

Algunas aplicaciones industriales podrían ser: medidas de RPM (Revoluciones por minuto) de equipos o medidas de alta precisión para equipos radar.

2.1.2. Medida mediante microcontroladores.

La medida de tiempos a partir de microcontroladores es muy útil para diseños de instrumentación electrónica que requieren conocer con cierta precisión variables de tiempo a un bajo coste.

La mayoría de los microcontroladores existentes en el mercado están provistos de temporizadores. Son contadores de tiempo que están integrados dentro del mismo microcontrolador y dependen de un reloj interno; con los que se puede medir intervalos de tiempo. La señal que se desea medir se conecta a un puerto de entrada del microcontrolador, seguidamente un búfer digital compara la tensión de la entrada con la tensión umbral interna. La CPU tiene la función de detectar los cruces de la señal por ese umbral para abrir o cerrar la puerta que detendrá el incremento del contador. Mientras la puerta está abierta el contador aumenta su valor cada T_s segundos, donde T_s es el período del oscilador de referencia. El valor digital que se obtiene se pasa a un registro interno. En la figura 2.2 puede observarse lo explicado.

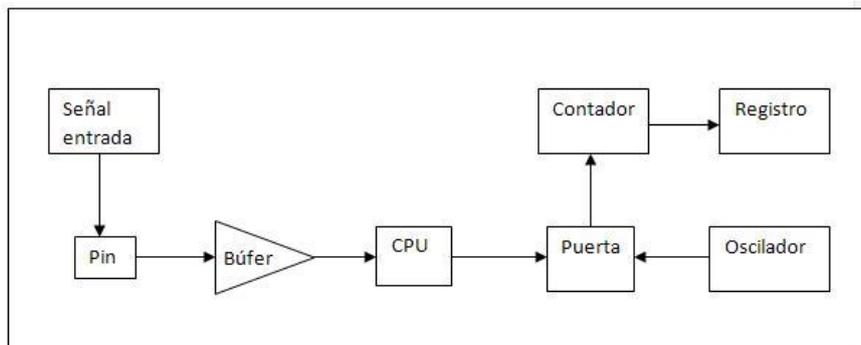


Figura 2.2. Diagrama de bloques de un μC con temporizador interno

Los puertos de entrada de los microcontroladores contienen un búfer ST (Schmitt Trigger Búfer) que permite medir intervalos de tiempo con menor sensibilidad al ruido cuando la velocidad de cambio de la señal es lenta. Estos *buffers* tienen una tensión umbral para el flanco de subida y otra diferente para el flanco de bajada. También en los últimos años se ha optado por incorporar en los microcontroladores comparadores analógicos con los cuales también se puede realizar dicha función.

Con esta opción se permite configurar un sistema de medida de bajo coste con unos resultados menos exactos que los que proporciona un contador universal, pero aún así válidos para aplicaciones industriales y comerciales.

2.2. FUENTES DE INCERTIDUMBRE EN UN SISTEMA BASADO EN MICROCONTROLADORES

Teniendo en cuenta las dos alternativas, hay que tener presente que existen una serie de fuentes de incertidumbre, directas o indirectas, que afectan a la veracidad de la medida. Sin embargo, en el caso de los contadores universales, el efecto de estas fuentes se puede reducir con mayor facilidad siguiendo las opciones de configuración que recomiendan los fabricantes. También, en el manual de operación suele detallarse información que caracteriza el equipo, como por ejemplo el nivel de ruido existente en los puertos; lo que permite determinar qué probabilidad de error aparece en la medida.

A pesar de sus limitaciones, el microcontrolador puede ser una buena opción debido a su reducido coste económico en comparación con el contador universal. Con objeto de alcanzar un diseño con un rendimiento óptimo y conocer las limitaciones de la medida, se hace necesario realizar un estudio pormenorizado de las fuentes de error.

2.2.1. La inestabilidad de la frecuencia de referencia.

Como se ha dicho anteriormente, para incrementar el contador de tiempo de un microcontrolador se utiliza un reloj, el cual está basado en un oscilador. Estos osciladores pueden ser de diferente naturaleza y por lo tanto implicar mayor o menor precisión. En este caso dicha precisión viene dada principalmente por la estabilidad de su frecuencia de oscilación. Mientras que los osciladores implementados con circuitos RC y LC presentan considerables problemas de inestabilidad, los osciladores implementados con cristales de cuarzo son muy estables y los errores de medición que originan pueden normalmente considerarse despreciables.

Los osciladores implementados con *circuitos LC* están formados por una bobina y un condensador en paralelo. Se consiguen las oscilaciones mediante un almacenamiento de energía de la carga eléctrica del condensador y del campo magnético de la bobina. Para estos osciladores la estabilidad a corto plazo se ve afectada por las fluctuaciones en los voltajes de alimentación, mientras que la estabilidad a largo plazo es en función de la temperatura y la edad de los componentes. Para la mayoría de aplicaciones de radiotelecomunicación la estabilidad dada es inadecuada.

Los osciladores implementados con *cristales de cuarzo* son circuitos electrónicos que aprovechan la resonancia mecánica del cuarzo para crear una señal eléctrica con una frecuencia muy precisa. La generación de esta señal eléctrica es debido a la propiedad piezoeléctrica del cuarzo.

En la figura 2.3. se refleja la estabilidad del cristal de cuarzo en función de la temperatura y del corte del cristal.

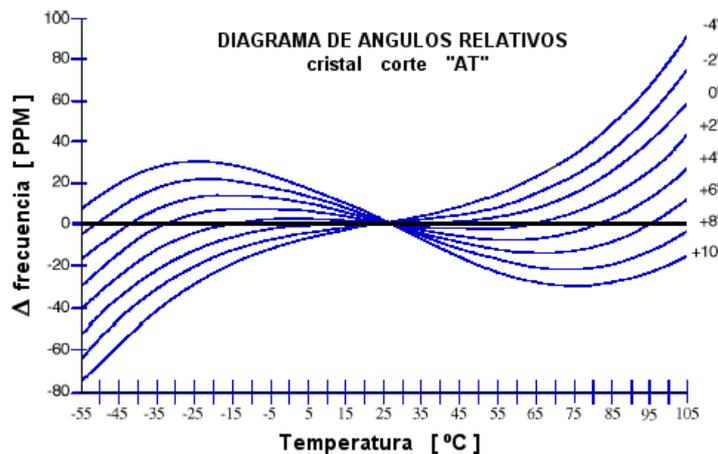


Figura 2.3. Variación de la frecuencia respecto a la temperatura y al corte del cristal¹

2.2.2. Error de cuantificación

En las conversiones de tiempo-digital existen, a grandes rasgos, dos técnicas para detectar un evento externo, y por lo tanto, el momento en que empieza o acaba el período de tiempo a medir: La detección mediante *software* y la detección mediante *hardware*. Según la técnica que se utilice se obtendrá diferentes valores puesto que cada una implica un cierto retardo que no siempre puede ser constante, a esto se le llama error de cuantificación. Todo y que a continuación se pasará a explicar cada técnica y las posibles soluciones para evitar o reducir el error de cuantificación, hay que comentar que este error no sólo viene introducido por los métodos de captura. También viene dado por

¹ Fuente: <http://www.huarpe.com/electronica/osc/oscilador-xtal.html>

el período de la señal del reloj. Cuanta menos frecuencia de reloj más posibilidad habrá de introducir este tipo de error ya que una captura puede caer en un tiempo indeterminado del flanco de la señal del reloj, y si los tiempos son más grandes habrá más posibilidad de introducir error de cuantificación. Se explicará con más detalle en el capítulo 5 en el apartado “Criterios para el diseño de un algoritmo de medida”.

A continuación se explica cada método de captura y las posibles soluciones para evitar o reducir este error de cuantificación.

Por un lado, la técnica del software (llamada *polling*) consiste en detectar el evento externo mediante un bucle de instrucciones en el que existe una instrucción que testea un puerto del microcontrolador. Como se trata de un bucle, la instrucción de testeo se ejecuta periódicamente; por lo tanto, a pesar de haberse producido el evento externo, no se detecta hasta que la CPU ejecuta dicha instrucción. Es entonces cuando el sistema realmente detecta que empieza o acaba la medida y así, la CPU ejecuta el código pertinente para encender o detener el contador. Entonces, dado que el evento externo puede producirse en cualquier momento, el período de tiempo que transcurre hasta que el sistema enciende o apaga el contador puede ser variable, con valor mínimo cero y máximo el tiempo de ejecución del bucle. Pero hay que tener en cuenta que esta técnica es válida tanto para comenzar la medida como para finalizar la medida, así que, si el tiempo de estos dos procesos es igual, el efecto se anularía e implicaría error cero en la medida.

Por otro lado, la técnica del *hardware*, disponible en la mayoría de microcontroladores del mercado, consiste en detectar eventos externos mediante *hardware* y parar el contador en el momento concreto. Hay dos métodos para detectar eventos externos a partir del hardware: *entrada externa de propósito general* y *módulo de captura*. En el primer caso se utiliza un puerto de propósito general del microcontrolador. Cuando éste recibe cierto nivel de tensión a causa de un evento externo, se activa la interrupción asociada al puerto. Cuando la CPU termina de ejecutar la instrucción actual ejecuta las instrucciones asociadas a la interrupción, a lo que se denomina rutina de interrupción, en este caso encender o apagar el contador. En este proceso también hay una demora de tiempo traducida en ciclos de instrucción necesarios para terminar de ejecutar la interrupción actual, se trata de un error de cuantificación que, sumado a los ciclos necesarios para aceptar la interrupción implicaría cierto error en la medida. En el caso del retardo a causa de aceptar la interrupción no se trata de un valor aleatorio sino conocido; y, como en el caso del *polling*, se puede anular si el retardo en el evento de encendido como el retardo en el evento de parada es igual. En el segundo caso, algunos microcontroladores también tienen un periférico adicional llamado *módulo de captura* especialmente concebido para medir intervalos de tiempo. El sistema es parecido al anterior, la diferencia está en que cuando se detecta un evento externo en el puerto predeterminado para realizar la captura, un registro concreto del microcontrolador previamente configurado captura el valor del contador, independientemente del estado de la CPU. Si se utiliza este método tanto para el evento externo en que empieza la medida como para el evento en que termina, el intervalo resultante de la resta de las dos capturas se

ajustaría al valor real con error de cuantificación básicamente igual al periodo de reloj.

2.2.3. Error causado por el ruido de disparo.

El proceso que se está estudiando, la medida de tiempos mediante un microcontrolador, se fundamenta en detectar el cruce de la señal que implica el evento exterior con la tensión umbral existente en el búfer del puerto de entrada del microcontrolador, a este momento se le llama disparo. Debido a la pequeña diferencia de niveles entre la señal y el valor de la tensión umbral en el entorno del instante de disparo, el ruido existente en ambas señales provocará un error en la medida que debe considerarse.

Si existe cierto ruido de disparo, la señal externa se cruza con la tensión umbral del búfer del puerto de entrada antes o después del momento exacto en que debería cruzarse; además, si la señal externa tiene una velocidad de cambio más lenta este efecto se pronuncia más. La figura 2.5 muestra esta explicación.

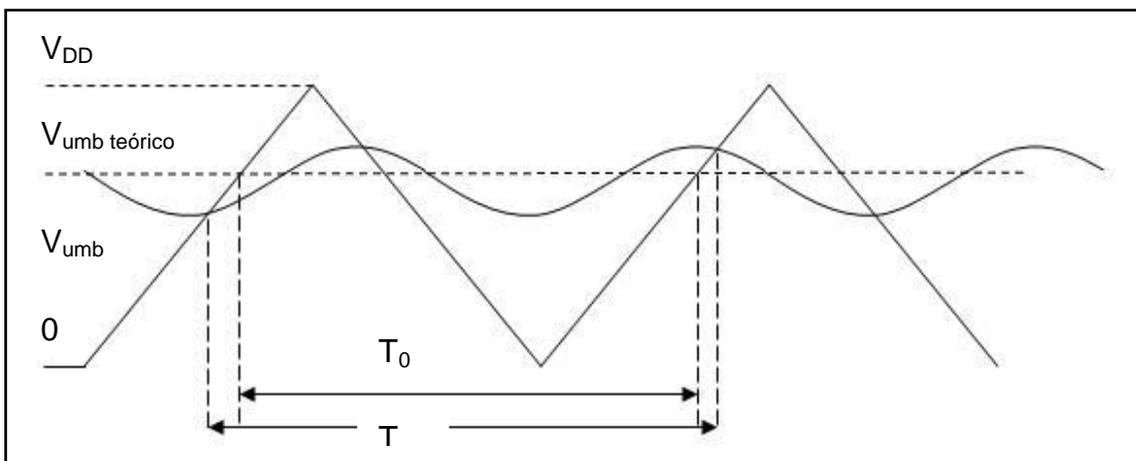


Figura 2.5. El cambio de la tensión umbral desplaza los puntos de disparo. La señal oscilante representa el umbral con el ruido añadido y la señal triangular representa la señal que marca el período a medir.

Por lo tanto, el ruido de disparo depende directamente del ruido de disparo externo (asociado a la señal externa), del ruido de disparo interno (asociado al umbral del puerto de entrada) y de la velocidad de cambio de la señal externa, y sus efectos son un adelanto o retraso en el comienzo o final de la medida de tiempo.

Una de las posibles soluciones para evitar el ruido externo es filtrar lo mejor posible la señal. En el caso del ruido de disparo interno existen tres

componentes principalmente, por un lado las atribuidas al propio microcontrolador, como el ruido térmico y el *ruido debido a la actividad de la CPU*; y por otro lado *el ruido introducido por interferencias en la alimentación*, que proviene del exterior. Estas dos últimas fuentes de ruido son importantes y por lo tanto se detallan en los siguientes apartados, junto a sus posibles soluciones.

2.3. INTERFERENCIAS EN LA ALIMENTACIÓN

Las interferencias o ruido añadidos a la alimentación del microcontrolador influyen directamente en la tensión umbral de los búferes de los puertos de entrada del microcontrolador y consecuentemente, provocan un error en la medida de tiempos. A nivel teórico se puede calcular el error introducido por el ruido gaussiano pero en cambio no se puede calcular los efectos del ruido impulsivo o interferencias sinusoidales.

Una forma de disminuir los efectos de las interferencias en la alimentación es alimentar de forma separada los circuitos digitales y analógicos, los circuitos de procesado y control, y las etapas de potencia.

2.4. ACTIVIDAD DE LA CPU

La actividad interna de la CPU puede provocar ruido de disparo interno. La conmutación de los transistores que forman la CPU en el proceso de ejecución de cada instrucción provoca armónicos de muy alta frecuencia que originan ruido en los niveles umbrales de comparación. Estos efectos provocan que el inicio o parada del contador en la medida de tiempo se produzca antes o después del momento ideal, lo que conlleva un error en la medida.

Muchos microcontroladores actuales tienen la opción de funcionar en diferentes modos. Cada modo implica diferentes periféricos en funcionamiento, incluyendo relojes y la propia CPU; esto conlleva un consumo energético diferente para cada modo, se les denomina modos de bajo consumo. Cuando el modo de bajo consumo incluye apagar la CPU se identifica normalmente por *sleep mode* o *modo reposo* y puede tener más ventajas aparte del ahorro energético. Si la actividad de la CPU, es decir, el hecho de ejecutar instrucciones mientras se produce el evento externo, influye en el ruido de disparo interno y provoca un error en la medida de tiempo, se puede considerar el uso del modo reposo cuando se prevea que se va a producir el evento externo. De este modo se evita la contribución de la CPU en el ruido de disparo interno.

Aunque este método mejoraría mucho el resultado de la medida, exige ciertas prestaciones por parte del microcontrolador. En primer lugar, el microcontrolador ha de contar con interrupciones que permitan despertar al sistema en el momento en que se produzca un evento exterior. En segundo lugar, el contador ha de permanecer activo mientras el sistema permanece en modo reposo; y por último, el tiempo que tarda el sistema en despertar ha de ser suficientemente pequeño como para que esta opción sea buena de cara a mejorar el resultado de la medida. El único inconveniente que aparece en esta situación es que la CPU no puede ejecutar ninguna instrucción mientras está en modo reposo, por lo tanto, hay que adaptar el código de manera que siga siendo efectivo.

Para finalizar, se puede decir que la utilización del *sleep mode* es una opción simple y eficiente para evitar los efectos negativos que la actividad de la CPU tiene sobre la medida de tiempos.

CAPÍTULO 3. AHORRO ENERGÉTICO EN MICROCONTROLADORES

3.1. CONSUMO DE UN MICROCONTROLADOR

Al tratarse de una temática muy enfocada al ahorro de consumo de los microcontroladores, es necesario estudiar el origen de las pérdidas de energía y cómo éstas se pueden minimizar. A continuación se estudia qué factores hay que tener en cuenta para disminuir el consumo de un microcontrolador.

3.1.1. Potencia disipada por un microcontrolador

La potencia que se disipa en un microcontrolador viene dada por la siguiente relación:

$$P \propto C \cdot V_{CC}^2 \cdot F + V_{CC} \cdot I_{LEAKAGE} \quad (3.1)$$

Siendo C la capacidad equivalente de las líneas y dispositivos que formarían el microcontrolador, V_{CC} la tensión de alimentación, F la frecuencia de trabajo e $I_{leakage}$ es la corriente de fugas.

La expresión de la potencia se divide en dos términos. Por un lado se distingue, un consumo de potencia estático básicamente debido a las fugas de los transistores cuando no hay actividad en el microcontrolador. Por otro lado, aparece un consumo de potencia dinámico causado por la conmutación de los transistores. Este último término viene principalmente modelado por el consumo de energía de los condensadores parásitos. Consecuentemente, tiene una dependencia cuadrática respecto a la tensión de alimentación y depende proporcionalmente con la frecuencia.

La expresión de la potencia consumida indica en un principio que el camino a seguir para reducir el consumo energético consiste en disminuir la tensión de alimentación y la frecuencia de trabajo. No obstante, existe un mínimo valor de esta tensión y frecuencia condicionado por la carga de procesamiento del microcontrolador.

Cuanto mayor sea la carga del microcontrolador, se requiere ejecutar un mayor número de instrucciones por unidad de tiempo. Esto significa mayores

frecuencias de reloj y menores tiempos de propagación en las diferentes etapas que forman el microcontrolador.

3.1.2. Relación entre frecuencia de trabajo y tensión de alimentación

Los tiempos de propagación son más pequeños cuanto mayor sea la tensión de alimentación. Por este motivo, es necesario alimentar el microcontrolador a mayor tensión, cuanto más elevada sea la frecuencia de trabajo. Normalmente, los fabricantes de microcontroladores proporcionan una relación entre la frecuencia de trabajo y la tensión mínima de alimentación, véase Fig. 3.1.

Estos valores mínimos en la tensión de alimentación, junto con la frecuencia mínima dada por la carga de trabajo, fijarán el consumo mínimo de potencia por el microcontrolador.

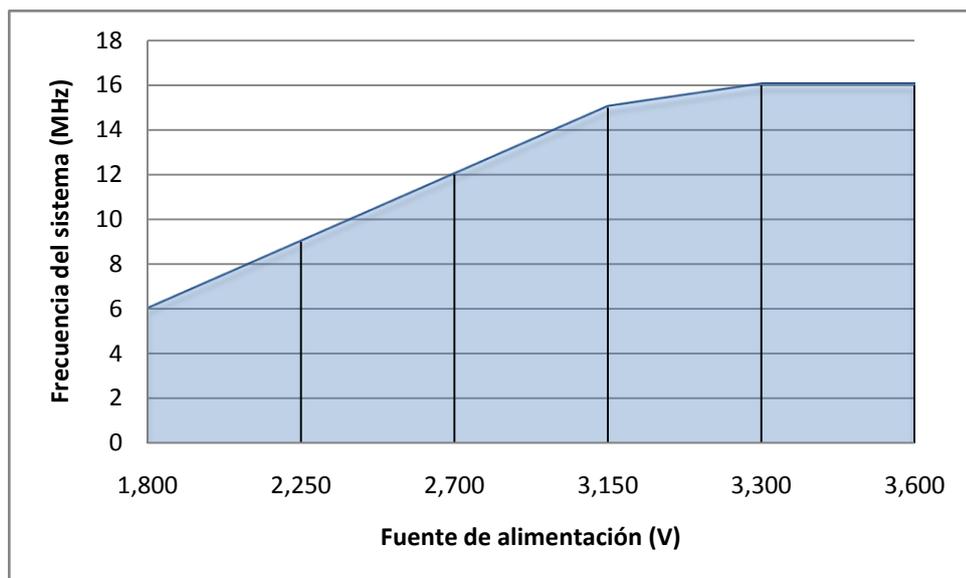


Figura 3.1. Gráfica que relaciona la frecuencia de trabajo con la tensión de alimentación. El microcontrolador utilizado es el MSP430F2111 en modo de funcionamiento normal.

3.1.3. Compromiso entre el error de cuantificación y la potencia disipada

Este apartado se puede enfocar de dos maneras distintas, ambas útiles y muy importantes dependiendo de la aplicación que el microcontrolador en cuestión vaya a tener.

Por un lado se puede hablar, como hasta ahora se ha estado haciendo en el capítulo 3, del reloj del microcontrolador. Pero por otro lado existen los relojes destinados a los contadores de tiempo integrados en el microcontrolador.

Se ha explicado que en el caso del microcontrolador, a mayor frecuencia de trabajo, mayor es la velocidad para realizar una tarea pero también es mayor el consumo. Bien, teniendo en cuenta que el funcionamiento del microcontrolador se basa en un reloj, y que por lo tanto su frecuencia de trabajo y el consumo que ésta implica vienen dados por dicho reloj, entonces se puede entender que para cualquier otro reloj, la función del cual sea o no ser núcleo de un microcontrolador, se puede aplicar la misma teoría. Por lo tanto, se observa que para cualquier contador de tiempo, el cual basa su funcionamiento en un reloj, se puede esperar una relación de potencia disipada similar a la mostrada en la relación (3.1).

A partir de esta analogía, se observa que las consecuencias de controlar el consumo de potencia pueden ser importantes tanto en las prestaciones del microcontrolador como en las prestaciones del contador de tiempo.

- Compromiso en la selección del reloj del microcontrolador.

Dado un sistema de medida basado en un microcontrolador, el cual detecte los eventos externos mediante *polling* o interrupciones de propósito general, si en el caso de reducir su tensión de alimentación, la frecuencia de trabajo del microcontrolador disminuye, el tiempo que tarda la CPU del microcontrolador en ejecutar una instrucción aumentará, por lo tanto, el error de cuantificación ($\epsilon_{\text{cuantificación}}$) en la medida también aumentará. El motivo se ha explicado con detalle anteriormente, pero básicamente es que el evento externo puede llegar en cualquier momento mientras la CPU ejecuta las instrucciones del bucle de testeo, en el caso del *polling*; o cualquier instrucción del programa, en el caso de la interrupción de propósito general. Si el tiempo para ejecutar estas instrucciones es mayor, como la CPU no atenderá al evento externo hasta que no haya finalizado la ejecución de la instrucción actual, el error añadido a la medida también será mayor.

En cambio, si el microcontrolador detecta los eventos externos mediante un módulo de captura, como éste no depende de la CPU, la frecuencia de trabajo, sea alta o baja, no añadirá error de cuantificación y por consecuencia, en la medida.

En el caso del *polling* y de las interrupciones de propósito general es necesario llegar a un compromiso entre el error de cuantificación y la potencia disipada, previendo las prestaciones que necesitarán el sistema y la autonomía disponible para el entorno en que se vaya a utilizar.

- Compromiso en la selección del reloj del temporizador.

Los temporizadores de los microcontroladores están formados por un reloj y un contador que se incrementa síncronamente en cada período. Si el contador se enciende o detiene en mitad de un ciclo del reloj, existirá un intervalo de tiempo de la señal inferior a un periodo de reloj que no se contabilizará en la medida.. Este error supone un error de cuantificación del temporizador y viene

expresado por su resolución. La resolución del contador es el mínimo tiempo que éste puede medir, y viene dado por el período del reloj en que se basa.

Un error de cuantificación pequeño significa una frecuencia de reloj elevada y consecuentemente, un consumo mayor. Existirá por tanto en cada aplicación un compromiso de diseño entre el error de cuantificación y el consumo. Un diseño óptimo implicará determinar cuál es la mínima resolución de medida que se necesita.

3.2. CONTROL DINÁMICO DE POTENCIA (DPM)

En los últimos años se ha investigado mucho en técnicas para diseñar sistemas de bajo consumo y hay principalmente dos grandes objetivos en este ámbito: aumentar la autonomía de los dispositivos portátiles reduciendo su consumo y conseguir diseños de alto rendimiento.

En el primer caso, la intención es prolongar la vida útil del dispositivo pero respetando un mínimo de prestaciones. El motivo principal es la escasez de fuentes de energía ya que en sistemas portátiles los recursos energéticos son limitados.

En el segundo caso, se trata de optimizar el rendimiento del sistema. Lo que se intenta no es que el consumo de potencia final del sistema sea menor, sino que para unas prestaciones concretas el sistema consuma menos, de tal manera que puedan aumentarse sus prestaciones sin que haya limitaciones por parte del entorno, como por ejemplo, la disipación en forma de calor por parte de esa potencia consumida.

El objetivo del bajo consumo se puede abordar de varias formas: En el diseño de un sistema de bajo consumo hay varios puntos a tener en cuenta: la concepción de su arquitectura general, la selección y diseño de los dispositivos que lo integran y la programación de los microcontroladores que controlan el sistema. Entre las técnicas de control se destaca el *Dynamic Power Management* (DPM) ya que puede adaptarse a un gran número de aplicaciones y supone una gran mejora del rendimiento energético.

El DPM es una metodología de diseño para sistemas dinámicamente reconfigurables que cambia el estado de funcionamiento de cada uno de los componentes según sea las prestaciones requeridas en cada momento con el fin de conseguir un consumo mínimo. La premisa fundamental para aplicarla exige que la carga de trabajo del sistema, o de cada uno de sus componentes, no sea uniforme en el tiempo. Es decir, se requiera que existan variaciones de la actividad del sistema que justifiquen su reconfiguración dinámica.

La reconfiguración dinámica del sistema (DPM) puede realizarse de diferentes formas. La alternativa más drástica consistiría en activar o desactivar cada uno de los componentes según la necesidad de ser utilizados por el sistema. No obstante, existen otras alternativas que no llegarían a inactivar completamente

estos dispositivos sino que simplemente reducirían sus prestaciones con objeto de reducir el consumo. En este sentido están orientadas las técnicas *Dynamic Frequency Scalling* (DPM) y *Dynamic Voltage Scalling*(DVS) que adaptan dinámicamente la frecuencia de trabajo y la tensión de alimentación de los componentes.

3.2.1. Control dinámico de la frecuencia (DFS).

El control dinámico de la frecuencia, también llamado *escalado de frecuencia*, se basa en la idea de que una menor velocidad de procesado requiere menos potencia. No obstante, no siempre implica un menor consumo energético. Así por ejemplo, si se calcula la energía consumida por un microcontrolador para ejecutar una determinada rutina, se comprueba que el consumo energético es el mismo sea cual sea la frecuencia de trabajo. Tal y como se muestra en la figura 3.2., la potencia disipada es menor, a menor frecuencia, pero el tiempo de ejecución aumenta de tal forma que consumo final coinciden a ambas frecuencias.

Cabe por tanto preguntarse si esta técnica es útil en este tipo de aplicaciones. En el caso de sistemas autónomos alimentados con baterías, la utilización de FVS conduce a una menor potencia disipada por el microcontrolador y un aumento de la vida útil de la batería. Según la fórmula empírica de Peukert [$C=X/I^\alpha$], la vida útil es mayor si la corriente de descarga es más pequeña.

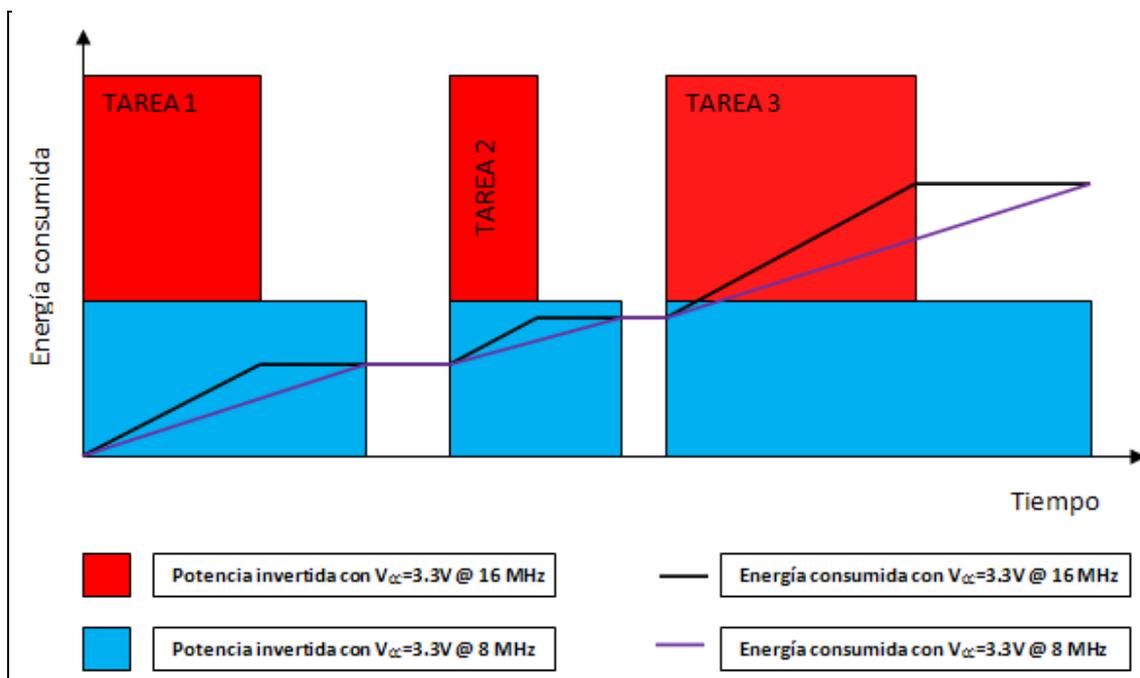


Figura 3.2. Comparación entre tres tareas realizadas sin DFS y con DFS.

Por otra parte, la utilización de esta técnica en el reloj de un temporizador supondría una reducción de la energía consumida ya que no implica un aumento del tiempo de medida. Desgraciadamente, una reducción de la frecuencia supondría un aumento del error de cuantificación. Existirá por tanto, una mínima frecuencia según sea la resolución de la medida requerida y por tanto, un mínimo consumo. No obstante, este consumo mínimo puede reducirse si se varía dinámicamente la frecuencia de reloj durante el proceso de medida. Esta es la técnica que se propone en este proyecto y explicaremos en detalle en el capítulo 4.

El resultado sería un consumo menor para realizar la medida de tiempo sin alargar el tiempo de ejecución de las instrucciones y sin perjudicar la resolución de la medida.

3.2.2. Control dinámico de la tensión (DVS)

El control dinámico de tensión, también llamado *escalado de tensión* es el más efectivo y el que proporciona mejores resultados en el aspecto de la optimización del consumo.

Esta técnica consiste en variar el nivel de tensión según las necesidades que se tengan. En el caso de querer optimizar el rendimiento y querer aprovechar las capacidades de un dispositivo se debe aumentar el nivel de voltaje. El aumento del rendimiento viene ligado al aumento del voltaje de alimentación, ya que al incrementarlo, la frecuencia de trabajo aumenta, con lo cual el sistema es más rápido. Este procedimiento tiene el nombre de *overvolting*. Aunque hay que tener en cuenta que llegado un límite no se consigue ninguna mejora en la velocidad de trabajo. Llegado este punto la mejora puede ser muy poco significativa en relación con el aumento de consumo energético, por lo tanto poco útil.

En cambio, si lo que se pretende es optimizar el consumo y reducir el gasto de energía, el procedimiento a seguir es disminuir la tensión. A esto se le denomina *undervolting*. La consecuencia principal del *undervolting* es la disminución del consumo de potencia disipada por parte del dispositivo ya que la tensión de alimentación es menor. Pero esta bajada de tensión también implica un decremento de la velocidad de trabajo del dispositivo, lo que provoca un incremento en el tiempo que tardará el dispositivo en llevar a cabo una tarea programada. A pesar de esto, debido a que la potencia disipada por el dispositivo depende de la tensión de alimentación al cuadrado como muestra la fórmula 3.1 citada anteriormente, la potencia consumida total se reduce.

En la figura 3.3 se puede observar la comparación entre tres tareas realizadas con *undervolting* y las mismas tareas realizadas sin *undervolting*. Sin aplicar *undervolting*, las tareas se ejecutan más rápidamente pero con un consumo energético máximo. Aplicando *undervolting* el dispositivo tarda más en realizar las tres tareas pero va consumiendo menos energía. Finalmente, se puede

observar que el consumo total del dispositivo para realizar las tres tareas programadas es menor si se ha aplicado *undervolting*.

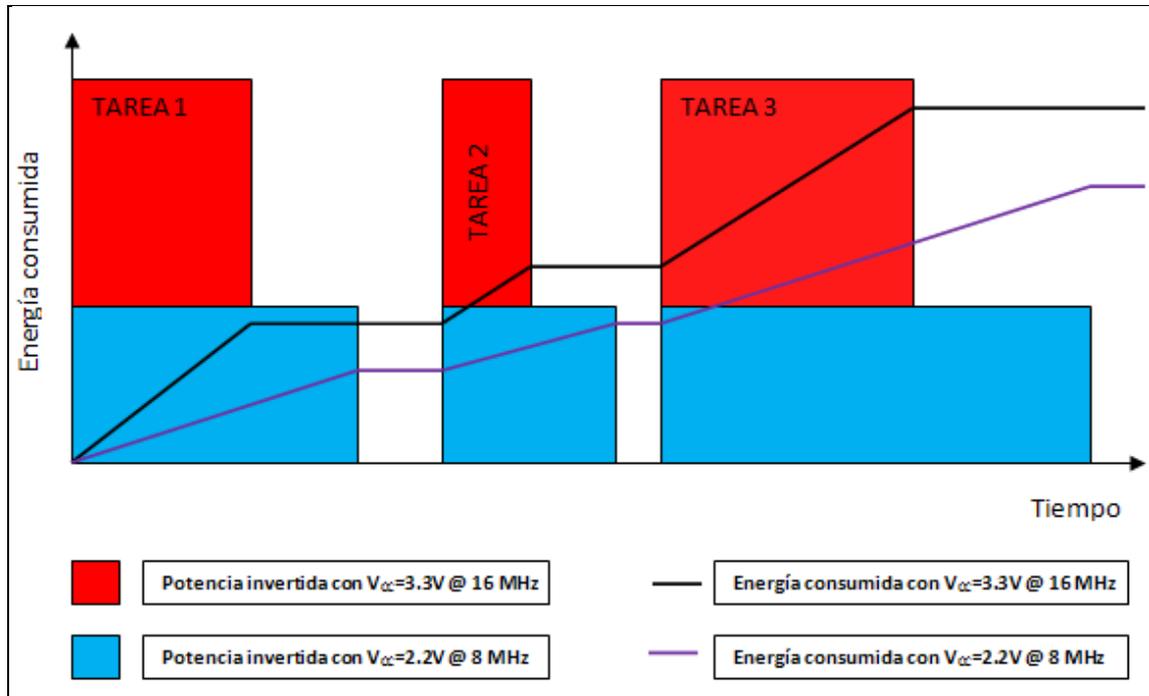


Figura 3.3. Comparación entre tres tareas realizadas sin DVS y con DVS.

Por otro lado, otro efecto debido al undervolting, en este caso, negativo, es un retraso que se añade al funcionamiento del dispositivo. Por ejemplo, en el caso de un microcontrolador, estaríamos hablando de un retardo al realizar una captura, o al habilitar o deshabilitar un puerto. Este retardo se hace más importante cuanto más se baje el voltaje.

Finalmente, la conclusión que se puede extraer de estas observaciones es que dependiendo de la tarea para la que se esté programado el microcontrolador, se debe encontrar un compromiso adecuado entre la tensión de alimentación del dispositivo y el error de cuantificación debido a estos retardos.

CAPÍTULO 4. PLANTEAMIENTO DEL PROYECTO

4.1. SELECCIÓN DEL MICROCONTROLADOR

4.1.1. Criterios de selección.

Existen varios criterios que deben considerarse en la selección microcontrolador para conseguir los objetivos del proyecto. Algunos de ellos ya se han explicado anteriormente como la necesidad de bajo consumo, la posibilidad de trabajar dinámicamente entre diferentes modos de funcionamientos y la disposición de varios relojes.

El bajo consumo es una de las características más importantes que se deben exigir ya que uno de los objetivos del proyecto es el ahorro energético para aplicaciones basadas en baterías. No sólo es importante que el microcontrolador disponga de varios modos de operación sino que también se requiere un consumo mínimo a pleno rendimiento. Para conseguirlo es imprescindible poder gestionar dinámicamente los propios recursos. Es decir, tener la capacidad de utilizar los elementos de microcontrolador siguiendo técnicas de DPM (*Dynamic Power Management*).

Otro de los aspectos a considerar es la disponibilidad de diferentes relojes en el microcontrolador. Se requiere no solamente que tengan un consumo mínimo sino que además puedan activarse o desactivarse en un tiempo lo más corto posible. Cuanto más tarde un reloj en estabilizarse, más tiempo tendrá que permanecer encendido y consecuentemente, supondrá un consumo energético mayor. Desafortunadamente, los relojes con menores tiempos de estabilización, tales como los basados en osciladores RC o tanques resonantes LC, presentan una estabilidad muy pobre. Por otra parte, los relojes de cristal de cuarzo que presentan gran estabilidad necesitan unos tiempos de estabilización muy largos y no resulta rentable activarlos y desactivarlos durante un proceso de medida. Por este motivo, se requiere un microcontrolador que disponga de un reloj de cuarzo de baja frecuencia y consumo que proporcione una medida estable durante la mayor parte del tiempo a medir, y un oscilador RC de alta frecuencia que pueda activarse rápidamente y reduzca el error de cuantificación.

También es importante que el microcontrolador incluya los periféricos que puedan ser necesarios para realizar la medida, como por ejemplo un comparador de histéresis, y que éstos sean fácilmente controlables. Esto reduciría la complejidad del sistema ya que evitaría la necesidad de utilizar otros circuitos integrados.

4.1.2. Microcontroladores de bajo consumo en el mercado

En el mercado actual se puede encontrar mucha variedad de μC con finalidades y características diferentes. En este proyecto se centrará la atención en microcontroladores enfocados al bajo consumo y al control dinámico de frecuencia.

Considerando las necesidades anteriormente expuestas, se describen los diferentes modos de funcionamiento así como las características de los relojes para cada una de las familias de bajo consumo que hay en el mercado.

- *Microchip PIC18F46J11*

Esta familia de microcontroladores de 8 bits proporciona bajo consumo y alto rendimiento, dispone de dos comparadores en un encapsulado reducido. Permite trabajar en cuatro modos de funcionamiento que son el *Run mode* donde todos los componentes funcionan sin restricciones, el *Idle mode* en el que se reduce a más de la mitad el consumo, el *Sleep mode* con la mayoría de las funciones apagadas y por último el *Deep Sleep mode* con corrientes por debajo de los 13 nA. Las prestaciones en cada uno de los modos son las siguientes:

- **Deep Sleep mode:** CPU y periféricos apagados con un consumo de corriente por debajo de los 13 nA.
- **Sleep mode:** CPU y periféricos apagados, SRAM encendida y un consumo de corriente por debajo de los 105 nA.
- **Idle mode:** CPU apagada, periféricos encendidos y corrientes por debajo de los 2,3 μA .
- **Run mode:** Todas las funciones encendidas con corrientes por debajo de los 6,2 μA .

Dispone además de tres relojes de diferentes frecuencias y consumos que pueden ser activados dinámicamente. Dos de estos relojes son de baja frecuencia (32 kHz) y no tienen un consumo elevado, del orden de 1 μA , ni gran estabilidad. El tercero es un reloj de 48 MHz de alta frecuencia

El tiempo de estabilización de un reloj a otro es la suma de dos ciclos del reloj anterior y de tres a cuatro ciclos del reloj actual

- *ATMEL AT91 ARM*

Atmel AT91SAM de 32-bits ARM Embedded Flash MCUs y MPUs está diseñado para el control de sistemas, conectividad inalámbrica, interfaz de usuario de gestión, y es de baja potencia. Ha recibido una amplia aceptación en mercados como terminales POS, en seguridad, automatización de edificios, control industrial, medicina, juegos de PC y periféricos, juguetes y accesorios para teléfono móviles. Proporciona unos resultados de alta calidad con un

consumo limitado. El sistema proporciona acceso a una herramientas de desarrollo, programación Flash, software y soporte técnico en todo el mundo.

Dispone de cinco modos desde el que funciona con todas sus funciones hasta el que está totalmente apagado.

- **OFF mode:** El microcontrolador está apagado a la espera de una interrupción por un cambio en un puerto de entrada. Su consumo en este estado es de alrededor 100nA.
- **Wait mode:** Ningún reloj funciona en este estado pero se tiene un *start-up* que permite despertarse en menos de 6 μ s.
- **Backup mode:** El oscilador de 32kHz encendido, *supply controller* encendido, *zero-power*, *power-on reset* encendido con un consumo de 3,5 μ A.
- **Idle mode:** Sólo está activo el *master clock*.
- **Active mode:** Todos los dispositivos activados con un consumo alrededor de 30 mA.

Pueden utilizarse hasta cuatro relojes: Oscilador RC que permite frecuencias desde 22kHz a 42kHz, *Fast RC oscillator* desde 1,5 MHz a 2,5 MHz, oscilador de cristal a 32kHz y el PLL de 18 MHz a 47 MHz. Estos osciladores proveen a SLCK, MAINCK y PLLCK cosa que permite cambiar de reloj según los intereses del usuario.

- MAXQ 610

Es un microcontrolador de 16-bit de bajo consumo diseñado para aplicaciones de consumo limitado. El MAXQ610 combina un potente microcontrolador RISC de 16-bit y periféricos integrados. Su nuevo modo de bajo consumo permite trabajar con corrientes de 0,2 μ A. Las herramientas para despertar el μ C incluyen interrupciones externas, una interrupción del timer y una interrupción de control de potencia. El microcontrolador trabaja con una alimentación de 1,70V hasta 3,6 V.

Este microcontrolador tiene dos modos de funcionamiento, por un lado el modo normal en que está activo todo, y por otro lado el modo Stop. Este modo mantiene la CPU apagada y guarda su estado y el estado de la memoria. El tiempo que tarda en despertarse se llama *wake up* y puede programarse desde 125 μ s hasta 8 s.

Como opciones que pueden activarse o desactivarse en función de la necesidad (consumo, precisión...) cabe destacar el *Power-Fail Detection*. Esta capacidad resetea el dispositivo en caso de detectar una falta de energía.

Esta familia permite utilizar tres relojes diferentes: dos relojes de 12 MHz, y otro de 20kHz. El tiempo de estabilización de los relojes de alta frecuencia es de:

$$t_{XTAL_RDY} = 8192 * t_{HFXIN}$$

$$\text{Stop-Mode resume time: } t_{ON} = 375 + 8192 * t_{HFXIN}$$

$$\text{Tiempo de wake up: } t_{wake-up} = 1/f_{NANO}$$

P89V52X2

Es una versión de bajo consumo del estándar de 8 bits 80C51. Además del modo de funcionamiento normal, dispone de dos modos de bajo consumo: el *Idle mode* y el *Power Down*.

- **Idle mode:** CPU apagada y los temporizadores, interrupciones y periféricos encendidos.
- **Power Down mode:** Periféricos apagados y las interrupciones sólo se activan mediante detección por niveles.

Contiene 3 temporizadores/contadores de 16 bits y permite la posibilidad de cambiar de reloj por software. Se tienen dos modos de reloj que permiten trabajar a diferentes rangos de frecuencia.

- i. En el modo 12x clock 0- 40 MHz
- ii. En el modo 6x clock 0-20 MHz

- MSP 430

Se trata de un microcontrolador de muy bajo consumo y alto rendimiento que permite cambiar dinámicamente la frecuencia. También tiene la opción de conmutar entre varios modos de consumo.

Los modos que permite usar este microcontrolador de menor a mayor consumo son LPM0, LMP1, LMP2, LMP3 y LMP4.

En cuanto a los relojes se tiene:

Por un lado el ACLK controlado por el oscilador de cuarzo y de muy bajo consumo. Su frecuencia de trabajo es de 32,768 kHz.

Por otro lado se tiene el SMCLK controlado por el DCO, que es un oscilador controlado digitalmente. Su frecuencia de trabajo puede programarse desde 1 MHz hasta 16 MHz. A diferencia del ACLK su consumo es mayor pero proporciona un menor error de cuantificación.

4.1.3. Selección de los microcontroladores MSP430

La familia MSP430 es una familia de microcontroladores del fabricante Texas Instruments (TI) que se implantó hace unos diez años. Las aplicaciones

principales para las que se utilizaba eran dispositivos portátiles, sensores inteligentes, etc.

A finales de los años noventa, la familia MSP430 empezó a coger importancia debido a su gran variedad de posibilidades en aplicaciones, por su precio de compra reducido y por la introducción del JTAG para la depuración y programación de dispositivos basados en memoria flash. Aunque por aquel entonces en este sector del mercado predominaba los productos de Microchip (PIC) y Atmel (AVR).

Los *contras* que se pueden encontrar al elegir esta familia de microcontroladores están directamente relacionados con la dificultad que tiene tomar el liderazgo de un mercado dominado por compañías que ya tienen su prestigio en el sector ganado. Aunque las buenas prestaciones del MSP430 hace que sus competidores lo tengan en cuenta.

El hecho de tener que aprender el funcionamiento de un nuevo microcontrolador a nivel de programación o de su propia arquitectura puede hacer que muchos clientes opten por la comodidad de utilizar un producto ya conocido. Eso supone no tener que adentrarse en el aprendizaje de una nueva arquitectura.

Para medianos y grandes proyectos la elección de la familia MSP430 es una alternativa económica frente a sus adversarios. El diseño de software orientado al MSP430 tiende a ser menos propenso al error y aporta una mejor visión del microcontrolador.

Un ejemplo que muestra la menor susceptibilidad de error es que su arquitectura de 16 bits proporciona una transferencia en un único sentido de los valores de la memoria de 16 bits o de los periféricos a los registros. Esto permite la actualización de datos en las rutinas de interrupción sin la necesidad de cambiar temporalmente las variables mientras se está en el programa principal. Esta ventaja la suele ofrecer el lenguaje ensamblador mientras que en C no se tiene en cuenta.

La facilidad de comprensión del μC viene directamente relacionada con su organización de memoria lineal (Von Neumann). En el pasado siempre se ha tenido la idea de que una arquitectura Harvard aportaba un rendimiento más económico por tener la memoria de datos y la del programa separadas. Este hecho limita el mantenimiento, la portabilidad y el rehúso del programa.

La oferta de kits de TI es otro argumento para escoger esta familia. TI ofrece una amplia gama de kits los cuales en el mercado actual tienen una relación calidad precio muy atractiva.

En cuanto a las principales características que hacen del MSP430 el mejor candidato para este proyecto se describen en el siguiente apartado con precisión.

Características del MSP430:

- Memoria y velocidad de procesamiento sobradamente suficientes.

- Diferentes osciladores:

DCO: oscilador controlado digitalmente, con frecuencia configurable en el margen de 100 kHz a 16 MHz aproximadamente, alta resolución.

Cristal de cuarzo: oscilador de cristal de cuarzo de gran estabilidad, con frecuencia de 32 768 Hz y muy bajo consumo.

VLO: oscilador de muy baja frecuencia, con frecuencia de 12 Hz y mínimo consumo.

- Diferentes relojes:

MCLK: reloj principal configurable para usar el DCO o el cristal de cuarzo.

SMCLK: reloj secundario para periféricos configurable para usar el DCO o el cristal de cuarzo.

ACLK: reloj auxiliar configurable para usar el cristal de cuarzo o el VLO.

- Diferentes modos de funcionamiento (modos de bajo consumo) Figura 4.1 y 4.2:

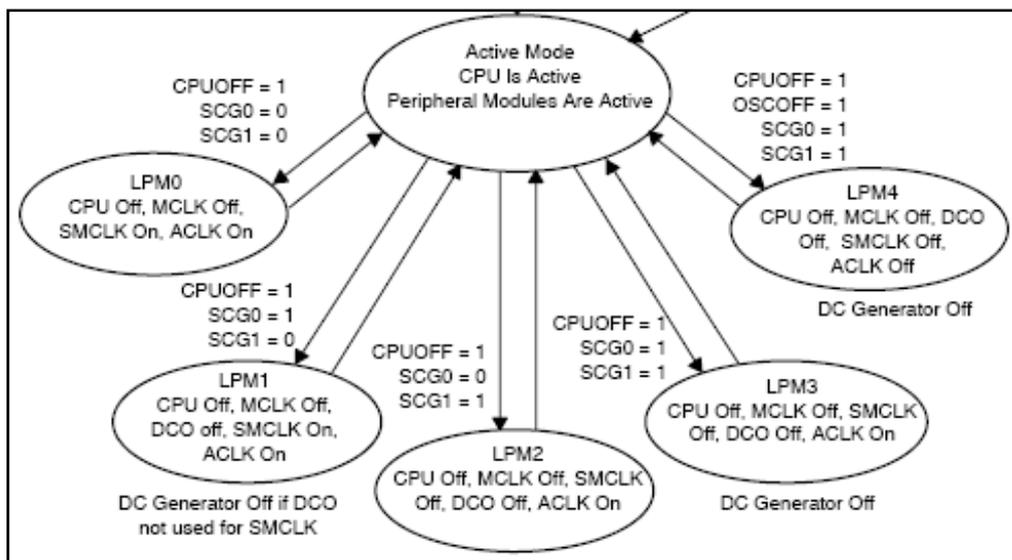


Figura 4.1. Diferentes modos de funcionamiento del MSP430².

² Fuente: Guía de usuario MSP430F2619

LPM0: sólo CPU apagada.

LPM1: CPU y reloj DCO apagados si éste último no se utiliza en el SMCLK.

LPM2: CPU y reloj DCO apagados.

LPM3: CPU, reloj DCO y generador DC apagados.

LPM4: CPU, reloj DCO y reloj ACLK apagados.

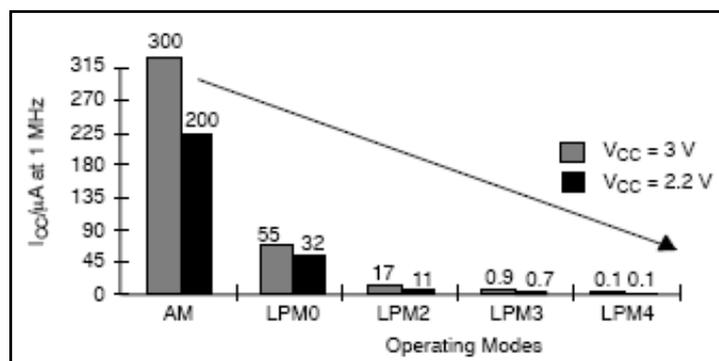


Figura 4.2. Consumo del MSP430 en los diferentes modos de funcionamiento³.

- Corto período de tiempo para estabilizarse el reloj que controla a la CPU, aproximadamente 6 µsegundos, lo que permite una rápida puesta en marcha de ésta.
- Dos temporizadores configurables sobre cualquier reloj del microcontrolador con un total de 9 contadores de tiempo e interrupciones propias y módulos de captura independientes de la CPU, **Timer A** y **Timer B**.
- Un comparador de histéresis integrado totalmente configurable con posibilidad de ser apagado, **Comparador A**; provisto de interrupciones que se activan por cambio de flanco.
- 8 **puertos** con 8 pines de entrada y salida configurables de forma manual.
- **Interrupciones de propósito general** asociadas a los puertos 1 y 2 que se activan por cambio de nivel en el mismo.
- 12 **registros de propósito general** de 16 bits provistos en la CPU para almacenar datos.

³ Fuente: Guía de usuario MSP430F2619

4.2. CRITERIOS PARA EL DISEÑO DE UN ALGORITMO DE MEDIDA

Hasta ahora se han comentado los criterios a seguir para escoger el microcontrolador que llevará a cabo la medida, y se han comparado diversas opciones, llegando a la conclusión de que el MSP430 de Texas Instruments es el mejor candidato.

Dado que el MSP430 tiene un gran abanico de opciones de configuración y periféricos, así como relojes con diferentes frecuencias y distintos modos de funcionamiento; es importante conocer qué ha de tenerse en cuenta para la aplicación a desarrollar, en este caso un contador de tiempo con el mínimo consumo de potencia.

Los elementos principales del MSP430 a controlar en esta aplicación son sus relojes y la CPU. También se dispone de un comparador de histéresis totalmente configurable.

Conociendo las características del microcontrolador a utilizar y los objetivos secundarios de la aplicación a desarrollar se pueden extraer una serie de criterios para diseñar el código que se programará en el microcontrolador.

A partir de ahora se explica qué criterios hay que seguir para que el código a diseñar explote lo mejor posible el MSP430 y éste sea lo más eficiente posible desempeñando su tarea; refiriéndose a consumo de potencia, resolución, exactitud y precisión en la medida.

4.2.1. Detección del evento externo

Se ha explicado que la detección del evento externo puede introducir cierto error dependiendo del método que se utilice. En el método de detección por *polling* o el método de detección por interrupción de propósito general existe un error de cuantificación asociado. En el caso del módulo de captura, éste permite la detección del evento externo inmediatamente sin depender de la CPU, pero sólo realiza una captura del temporizador que está controlando, no puede ejecutar ninguna tarea hasta que la CPU no acepte la interrupción que tiene asociada y previamente habilitada.

Por lo tanto es mejor utilizar el módulo de captura siempre que sea posible para ajustar la medida al valor real. En el caso del MSP430 existen varios módulos de captura disponibles en sus dos temporizadores (Timer A y Timer B).

4.2.2. Control dinámico de la frecuencia del reloj.

El compromiso entre la resolución de la medida y la potencia consumida conduce a utilizar dos relojes de frecuencias diferentes. Concretamente, se

propone utilizar el reloj de cuarzo de 32 678 Hz de bajo consumo, y el reloj DCO con una frecuencia aproximadamente de 16 MHz. Mientras que el primero tiene unos tiempos de activación muy largos y consecuentemente, permanecerá activo durante toda la medida, la rapidez de activación del DCO permitirá activarlo únicamente cuando sea necesario

4.2.3. Control dinámico del modo de funcionamiento de la CPU

Tal y como se ha explicado anteriormente, la actividad de la CPU puede provocar ruido de disparo si está activa en el momento en que se produce el evento externo. Además de este inconveniente, la CPU así como los relojes del microcontrolador tienen un consumo que no siempre es necesario costear. Para evitar esto se puede aprovechar la capacidad del microcontrolador de poder funcionar en diferentes modos. Además, como existen varios modos de funcionamiento en el MSP430, se pueden activar unos u otros en función de las necesidades o circunstancias.

Por ejemplo, para evitar las interferencias por parte de la actividad de la CPU, se puede utilizar el modo LPM0 que sólo apaga la CPU en el momento en que se vaya a recibir el evento externo que determina el inicio del período a medir.

También, durante la mayor parte del tiempo que dura la medida, como no hay ninguna tarea a desempeñar de forma paralela, se puede usar el modo LPM3 que apaga el DCO además de la CPU y mantiene en funcionamiento el reloj auxiliar ACLK con un consumo muy reducido.

Además, aprovechando el corto período de tiempo que el DCO necesita para estabilizarse, la conmutación al LPM0 se puede apurar lo máximo posible al momento en que se vaya a recibir el evento externo que determina el final del período a medir. Esto permitiría un gran ahorro de energía por parte del reloj de alta frecuencia aparte de mantener apagada la CPU para evitar las interferencias que afectarían al detener el contador.

En resumen, el código a diseñar también debería basarse en conmutar entre los modos de funcionamiento LPM0 y LPM3 en momentos concretos del período a medir.

Basándose en estos criterios es más fácil enfocar la programación del MSP430 y diseñar una estrategia para hacer la medida. Esta estrategia combina el control dinámico de la frecuencia con el control dinámico de los modos de funcionamiento de la CPU. En el apartado siguiente se explica con detalle esta estrategia.

4.3. ESTRATEGIA DE MEDIDA

El algoritmo propuesto se ha concebido para medir el semiciclo de una señal cuadrada. A pesar de que la señal no tiene porque ser periódica, se presupone que las variaciones entre un ciclo y el siguiente son suaves. De esta forma, el algoritmo puede predecir cuándo se van a producir los eventos que marcan el inicio y el final de la medida, y así activar los elementos necesarios del microcontrolador instantes antes.

A continuación se explican los diferentes elementos que intervienen en la medida.

Un comparador analógico integrado en el MSP430 se encarga de indicar cuáles son los momentos de inicio y final del período a medir. Para ello dispone de un umbral de tensión. Cuando la señal de entrada cruza ese umbral el comparador genera un flanco en la señal de salida. Cuando la señal vuelve a cruzar por el umbral el comparador genera un flanco en sentido contrario al anterior. Este módulo tiene asociada una interrupción que se puede configurar para activarse en el momento de generarse el flanco de subida o el flanco de bajada.

Un contador interno (Timer A) contabiliza el número de ciclos del reloj auxiliar ACLK de 32 678 Hz. Se trata de un contador de 16 bits, cuyo valor es accesible desde el programa a partir del registro especial TAR. También dispone de módulos de captura e interrupciones que permiten despertar el microcontrolador cuando el TAR alcanza ciertos niveles que pueden ser programados (TACCR0, TACCR1, TACCR2).

Por último, se ha utilizado otro contador (Timer B) para el reloj DCO de 16MHz. Su valor es accesible a partir del registro especial TBR y tiene características similares al Timer A.

4.3.1. Descripción del método para hacer la medida

Debido a la complejidad que supone describir el método que se emplea en el sistema definitivo, la explicación del mismo se desglosará en cinco fases explicadas a nivel general.

La idea es partir de una base muy sencilla e ir añadiendo elementos que mejoren los resultados de la medida, tanto configuraciones como conceptos o nuevos cálculos.

- *Primera fase: Medida con ACLK.*

En este primer paso el objetivo es medir un tiempo mediante el reloj de bajo consumo ACLK.

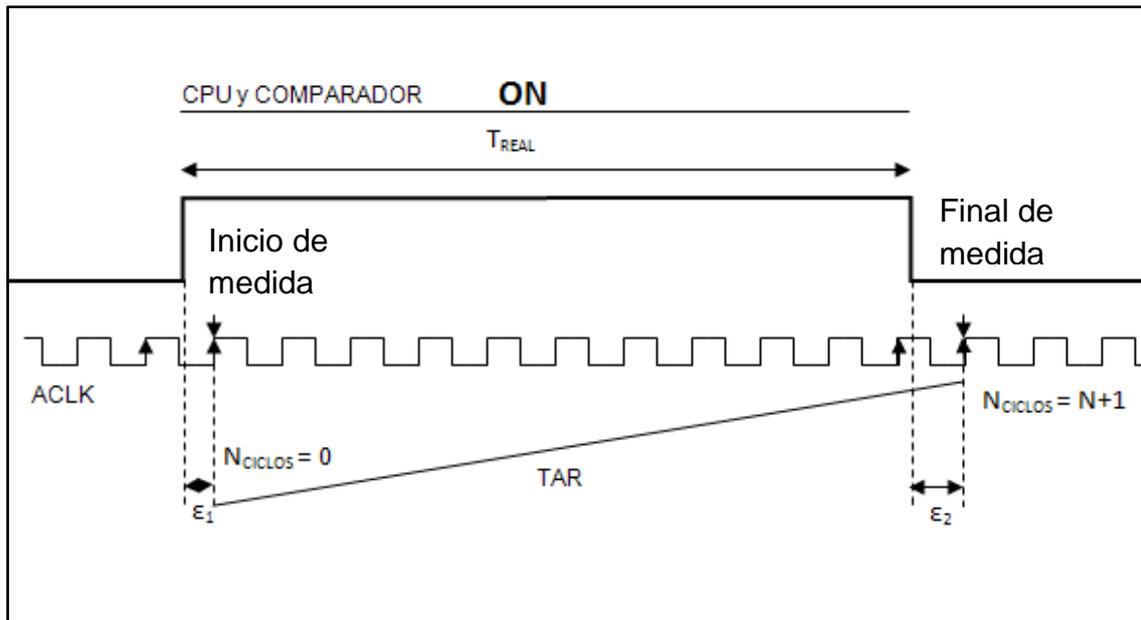


Figura 4.3. Representación de la medida de tiempo sólo con el reloj ACLK sin restricciones de consumo.

Tal y como se muestra en la figura 4.3., para realizar la medida sólo se utiliza el reloj ACLK. Aparte de esto el sistema se encuentra en modo activo (AM) durante todo momento, mantiene la CPU y el comparador encendidos.

La interrupción del comparador se activa con el flanco de subida de éste. Esta interrupción enciende el Timer A; el contador TAR se incrementa hasta que se detecta el flanco de bajada que indica final de medida, en este momento el registro TACCR1 captura el valor del TAR.

Por lo tanto, como resultado de medida se tiene la expresión 4.1.

$$t_{medido} = N_{ciclos} \cdot T_{ACLK} \quad (4.1)$$

Se observa que el inicio de medida no coincide con el flanco de subida del ACLK y que por lo tanto el incremento del TAR no es instantáneo, sino que se pierde cierto tiempo ϵ_1 debido al error de cuantificación del Timer ϵ_q . En el caso del final de la medida pasa algo parecido, se añade cierto tiempo ϵ_2 , también debido al error de cuantificación del Timer. Esto implica que el resultado de la medida tenga un error ϵ_{total} añadido calculado en la expresión 4.2.

$$\varepsilon_{total} = \varepsilon_1 - \varepsilon_2 \quad (4.2)$$

Teniendo en cuenta los objetivos del proyecto, con esta configuración de sistema se observan dos puntos a mejorar:

- El consumo total del sistema se puede disminuir apagando los elementos que no sean necesarios.

- Debido a la frecuencia a la que trabaja el ACLK, el error de cuantificación del Timer A es de alrededor de 30 μ s. Este error es relativamente alto y se puede disminuir utilizando otros relojes que permitan al Timer tener más resolución.

Estos problemas se intentan solucionar en las siguientes fases.

- *Segunda fase: Medida con ACLK en bajo consumo (LPM3).*

En esta fase el objetivo es reducir el consumo. Para ello se apagan los elementos que no son necesarios durante la medida y se encienden en el momento preciso.

Se ha decidido utilizar el modo de bajo consumo LPM3 porque apaga todo el sistema excepto el reloj ACLK que se utiliza durante la medida. Además se apaga el comparador mientras no se espere ningún evento de inicio o final de la medida.

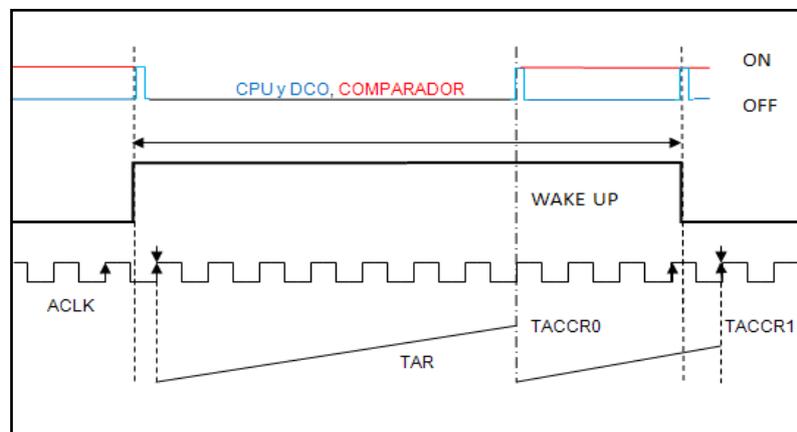


Figura 4.4. Representación de la medida de tiempo con el uso del modo de bajo consumo LPM3 (CPU y DCO OFF).

En la figura 4.4. se puede observar cómo se procede en esta fase. Las medidas se realizan con el sistema en bajo consumo. Este modo se mantiene durante todo momento debido a que la ejecución de las interrupciones programadas activa automáticamente la CPU cuando es necesario.

En el caso del comparador se apaga desde que comienza la medida hasta que se activa la interrupción del registro de comparación TACCR0. Esta interrupción vuelve a encender el comparador para poder detectar el final de la medida. Al tratarse el período a medir de valores que varían suavemente se puede predecir el final del período con cierta seguridad y así poder inicializar el TACCR0.

Se observa que el contador TAR se reinicia una vez ha alcanzado el valor del TACCR0. A partir de aquí el final de medida es igual que en la fase anterior a diferencia de que el valor capturado por el TACCR1 será menor. En la interrupción asociada a éste registro se calcula el valor de TACCR0 de la medida siguiente de esta manera 4.3.

$$TACCR0' = TACCR0 + TACCR1 - 5 \quad (4.3)$$

En la expresión se observa un 5, este valor es el margen que se coloca para aceptar las pequeñas variaciones del período a medir.

A continuación, en la tercera fase se propone una solución al problema del error de cuantificación presentado al principio.

- *Tercera fase: Introducción del DCO en el sistema.*

Como ya se ha comentado anteriormente, el error de cuantificación debido al ACLK es relativamente alto. En esta fase se propone terminar la medida con el DCO configurado con una frecuencia más alta que la del ACLK. De este modo se pretende conseguir la resolución de este oscilador⁴. Cabe destacar que el resto de configuraciones permanecen igual que en el caso anterior excepto el modo de funcionamiento. En este caso se incluye el LPM0 en un momento concreto que se menciona a continuación.

En la figura 4.5. se muestra el procedimiento para llevar a cabo esta fase.

⁴ La opción de utilizar únicamente el DCO para realizar la medida se descarta debido a su elevado consumo.

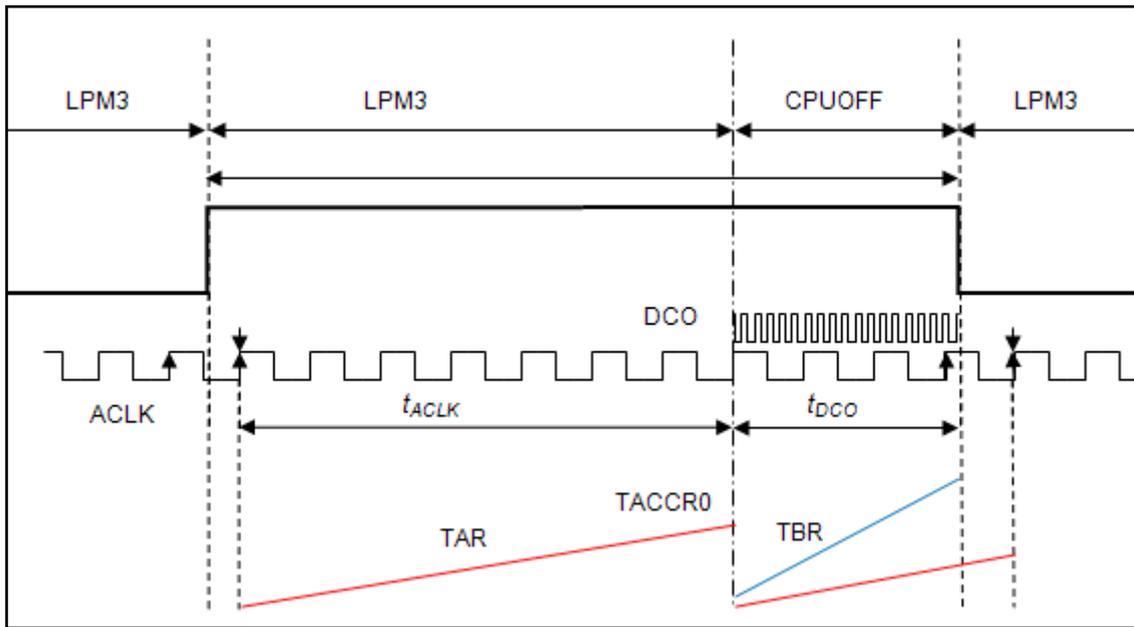


Figura 4.5. Representación de la medida de tiempo del reloj ACLK combinado con el reloj DCO.

En esta figura se ha incorporado el Timer B, el cual utiliza el DCO para incrementar el contador TBR. Este contador mide el tiempo transcurrido desde el momento de encender el comparador en la interrupción del TACCR0 hasta que finaliza el período a medir. En este momento el registro TBCCR0 captura su valor a causa del flanco del comparador.

Durante este nuevo margen de tiempo medido el estado del dispositivo ya no es el LPM3 sino el LPM0. El motivo es que en el primer caso se apaga el DCO, por lo tanto no se podría mantener activo durante la medida del Timer B y por eso se activa el modo LPM0 que sólo apaga la CPU.

El resultado de la medida se describe mediante la siguiente expresión 4.4.

$$t_{medido} = t_{ACLK} + t_{DCO} = N_{ciclos\ ACLK} \cdot T_{ACLK} + N_{ciclos\ DCO} \cdot T_{DCO} \quad (4.4)$$

Se observa que el nuevo margen de tiempo medido viene dado por la combinación del ACLK y el DCO. Esto implica una mejora en el resultado porque el DCO se ajusta mejor al momento en que llega el flanco del comparador, por lo tanto implica menor error de cuantificación. Pero a pesar de esto, hay que tener en cuenta que este oscilador necesita cierto tiempo de

estabilización, y que el hecho de empezarlo a utilizar al momento de encenderlo puede provocar medidas erróneas.

En la cuarta fase se propone una solución para este problema.

A parte de esto, también se puede observar que el error de cuantificación del inicio de la medida continúa sin ser eliminado, pero eso se tratará en la última fase.

- *Cuarta fase: Estabilización del DCO.*

En esta fase se tiene en cuenta el tiempo de estabilización del DCO. Para ello se provoca una espera equivalente a tres ciclos del ACLK desde que se enciende el DCO hasta que se comienza a utilizar para medir. De esta manera se asegura un tiempo superior al tiempo que necesita para estabilizarse (alrededor de 6 μs).

La siguiente figura (figura 4.6) muestra este proceso.

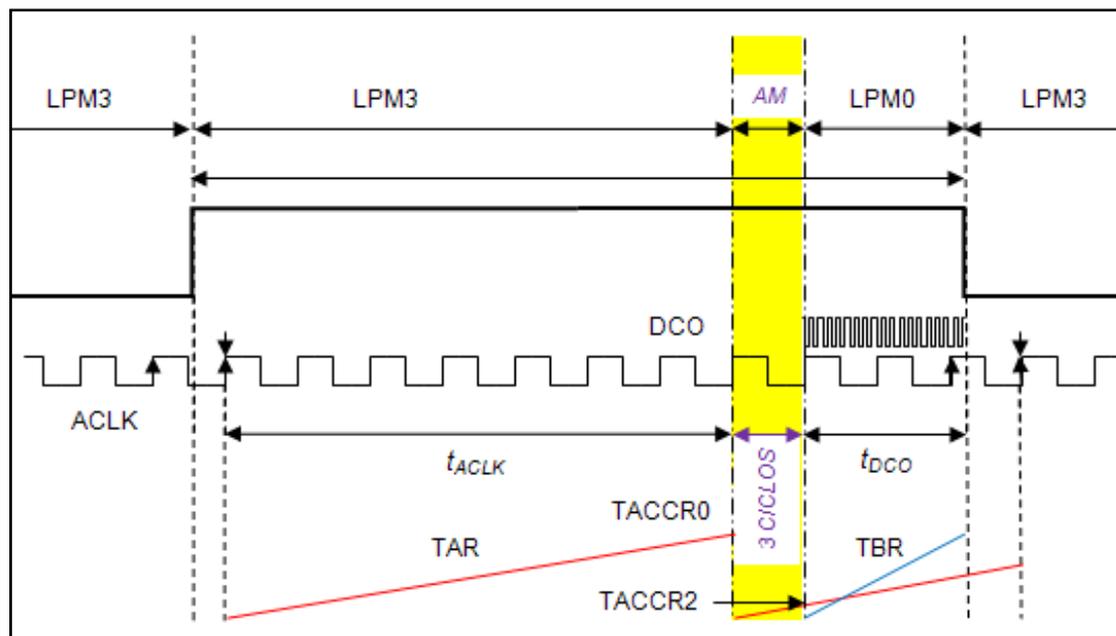


Figura 4.6. Medida de tiempo teniendo en cuenta la estabilización del DCO.

En esta fase se habilita una segunda interrupción a partir del registro de comparación TACCR2 inicializado con un valor de tres ciclos. La interrupción original TACCR0 despierta el sistema conmutando al modo activo. Tres ciclos después se ejecuta la interrupción del TACCR2 que pone en marcha el Timer B. Este reloj termina la medida en el momento que finaliza el período a medir.

Con esto, el resultado de la medida se obtiene de la suma del tiempo medido por el ACLK, los tres ciclos de espera de estabilización del DCO y el tiempo medido por el DCO. La siguiente expresión (4.5) muestra este resultado.

$$t_{medido} = t_{ACLK} + 3 \text{ CICLOS } ACLK + t_{DCO} \quad (4.5)$$

Esta expresión aún no tiene en cuenta el error de cuantificación que introduce el ACLK al comenzar la medida. Esto provoca que se aplique una nueva técnica para compensar este error.

A continuación, en la quinta fase, se explica esta técnica.

- *Quinta fase: Cálculo de t_p .*

Este tiempo compensa el error de cuantificación ε_1 presente en el inicio de la medida. Para poder calcularlo es necesario realizar un par de medidas aparte de la medida principal.

En la figura 4.7 se muestra un esquema general de las medidas tomadas.

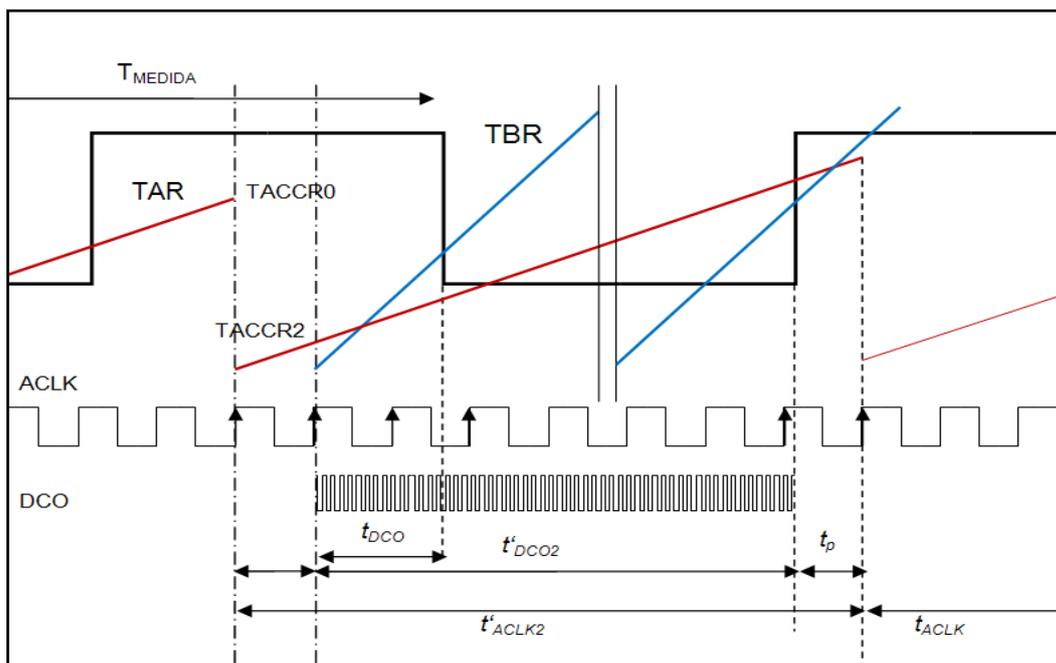


Figura 4.7. Medida del tiempo incluyendo el cálculo de t_p .

Por lo tanto, la expresión del resultado final será la siguiente (Expresión 4.7).

$$t_{medido} = t_{ACLK} + 3 \text{ CICLOS } ACLK + t_{DCO} + t_p \quad (4.7)$$

CAPÍTULO 5. PROGRAMACIÓN DEL MSP430

5.1. ESTRUCTURA DEL PROGRAMA

El algoritmo programado consta de tres partes bien diferenciadas: configuración general, programa principal y las interrupciones.

- *Configuración general*

La parte de configuración tiene como objetivo preparar el dispositivo para el proceso principal que se haya programado. En este caso la configuración abarca los periféricos, los módulos de los Timers, y el programa principal.

Como periféricos está el Comparator A, al cual se le configura el nivel de tensión de referencia y los pines de entrada y salida.

También en este apartado se incluye la configuración de los puertos que según la necesidad que se requiera se pueden poner de salida o de entrada. Se debe tener en cuenta que los pines no utilizados deben dejarse en configuración de salida ya que en caso contrario afecta al consumo aumentándolo.

Como módulos están el Timer A y el Timer B. En los dos casos se configura el reloj de origen y el registro de captura. En el primer caso el reloj origen que se toma es el ACLK. Este reloj ya tiene asociado automáticamente el oscilador de cuarzo del microcontrolador, que funciona a una frecuencia de 32 768 Hz. En el caso del Timer B, se configura para usar el reloj SMCLK, que previamente se ha configurado para basarse con el oscilador controlado digitalmente (DCO) con una frecuencia aproximada de 16 MHz.

El registro de captura del Timer A se configura en modo captura ya que interesa capturar el valor del registro del contador en el momento que se produzca un evento. También se sincroniza la señal del reloj con la señal de entrada para evitar que se pierdan ciclos de reloj en el contador. Otro aspecto que se configura es que la captura se produzca en el flanco de bajada. Se ha elegido de esta manera ya que por problemas de umbrales la medida se hace mejor con el flanco de bajada que con el de subida. Finalmente se configura como fuente de captura la salida del comparador.

En el registro de captura del Timer B se realiza la misma configuración que en el anterior debido a que las necesidades son las mismas, a pesar de que son relojes diferentes.

Por último en la configuración del programa hay dos cosas que configurar. En primer lugar se configuran los registros de propósito general, donde se van a guardar los valores medidos por los relojes. Y en segundo lugar se configura el encendido del Timer B tres ciclos más tarde a partir de cuándo la CPU se enciende.

Programa principal

En cuanto al programa principal se trata de una parte en la cual se activan o se desactivan registros a la espera de una interrupción.

Hay una inicialización del momento en el que debe despertar el microcontrolador, ese es un valor aproximado teniendo en cuenta la medida que se va a realizar. Este valor sólo sirve para empezar, en cuanto el programa ya está en funcionamiento ya no se tiene en cuenta.

A continuación, se prepara el comparador para la medida, encendiéndolo y habilitando su interrupción. Seguidamente, se apaga la CPU y se espera con un bucle infinito a que lleguen los eventos externos.

Para terminar, y debido a su importancia, se explicarán las interrupciones en el apartado siguiente.

5.2. DESCRIPCIÓN DE LAS INTERRUPCIONES

En este algoritmo se han creado cuatro interrupciones las cuales van a encargarse de gestionar la medida y de tomarla. A continuación se pasa a detallar cada una de ellas.

Interrupción 1: Se activa la interrupción en el momento en que se detecta un flanco de subida del comparador. En ese momento se captura el valor del contador del Timer A y del Timer B. Estos dos valores representan el final de la medida y posteriormente servirán para calcular el valor t_p anteriormente explicado. Se pone a cero el contador del Timer A para comenzar la medida de la manera en que se ha descrito en el capítulo 4 (Estrategia de medida). Seguidamente el Timer A empieza a contar en *UP mode*, lo cual quiere decir que llegado a un valor predeterminado se activa la interrupción asociada. Por último, se habilita la interrupción 2, se apaga el comparador y se entra en el modo de bajo consumo 3 (LPM3).

Interrupción 2: Se activa cuando el registro contador del Timer A llega a TACCR0. En esta interrupción se conmuta al modo activo, se habilita la interrupción 3 y se deshabilita la interrupción 1 para evitar que se active en momentos no deseados.

Interrupción 3: Se activa tres ciclos de ACLK después de despertar, para pasar a encender el Timer B en el que el DCO ya estará estable y listo para la medida. El valor del contador del Timer A que hace activar la interrupción es TACCR2.

Primeramente se pone a cero el contador del Timer B para empezar la medida y se enciende el Timer B. Se deshabilita la interrupción 2 para evitar que se active en momentos no deseados, se habilita la interrupción 4 y se apaga la CPU para que las interferencias generadas no afecten a la medida.

Interrupción 4: Se activa cuando el registro de captura TACCR1 recibe el flanco de bajada del comparador, justo después de capturar el valor del contador.

Los valores capturados por el TACCR1 y el TBCCR0 que servirán para el resultado de la medida final se guardarán en los registros de propósito general. A continuación, se sumará el TACCR1 al TACCR0 y se restará siete al todo. Finalmente se guardará los cuatro valores que forman el resultado final de la medida en la dirección de la memoria RAM que apunta su puntero correspondiente.

En la figura 5.1. se puede ver un diagrama de flujo del algoritmo programado.

Por último es conveniente comentar la utilidad de la interrupción del TAIV que se encuentra a parte de las cuatro explicadas anteriormente. Tanto las interrupciones TACCR1 como TACCR2 tienen el mismo vector de interrupción, factor que provoca que, en caso de que se utilicen simultáneamente, sería necesaria una gestión de éstas. Así que la función de esta interrupción es reconocer cual es la interrupción que realmente se ha activado para ejecutar el código correspondiente.

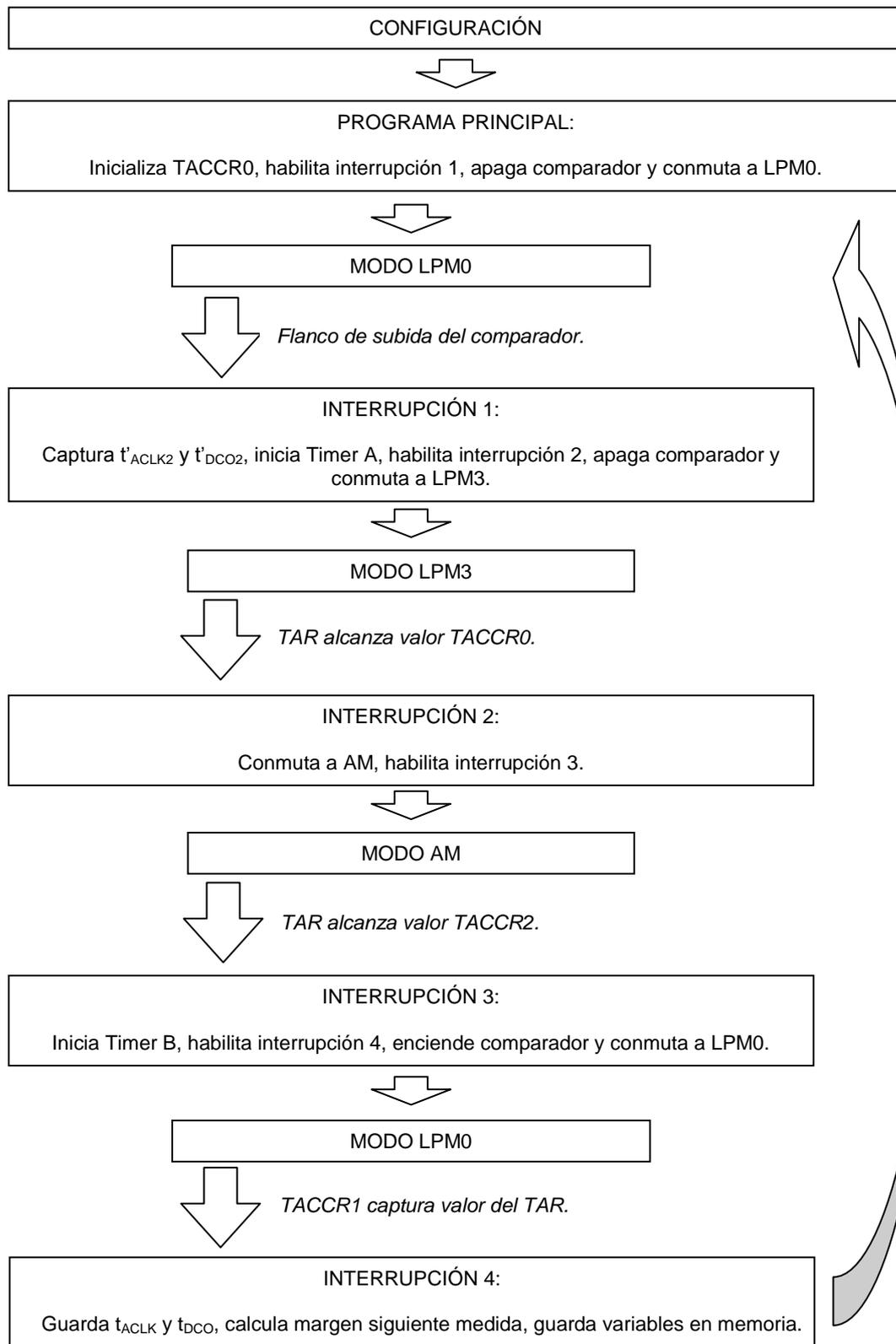


Figura 5.1. Diagrama de flujo del programa.

CAPÍTULO 6. RESULTADOS EXPERIMENTALES

6.1. PROCEDIMIENTO DE MEDIDA

Antes de pasar a explicar el procedimiento de medida es conveniente tener presente la señal de entrada que se ha utilizado en todo momento, también denominado como condiciones de medida.

La señal de entrada consiste en una señal cuadrada generada por un generador de gran precisión, con una amplitud de pico de 2 V, con una asimetría del 20% y la magnitud de período fija a 4 ms. Estas características implican que el período a medir sea de 3.2 ms.

Se ha procedido a hacer dos tipos de medidas, por un lado las de tiempo, que permitirán conocer las prestaciones del sistema y por otro las de consumo.

A continuación se explicará el procedimiento que se ha seguido para la obtención de las medidas de tiempo.

Medidas de tiempo

El medio por el que se comunica con el microcontrolador es mediante el programa IAR, con el cual se programa el algoritmo y con el que se podrán leer las medidas en memoria. La alimentación al microcontrolador para la entrada de una señal cuadrada viene directamente del ordenador mediante USB.

Las medidas no se toman de igual manera con cualquier señal de entrada debido a que no es lo mismo realizar pruebas con una señal cuadrada a la entrada, que con una señal rampa (debido a que la rampa introduce un ruido de disparo externo como se explico en el apartado 3.2).

En el caso actual en el que se utiliza una señal cuadrada, la medida se realizará como indica la Figura 6.1.

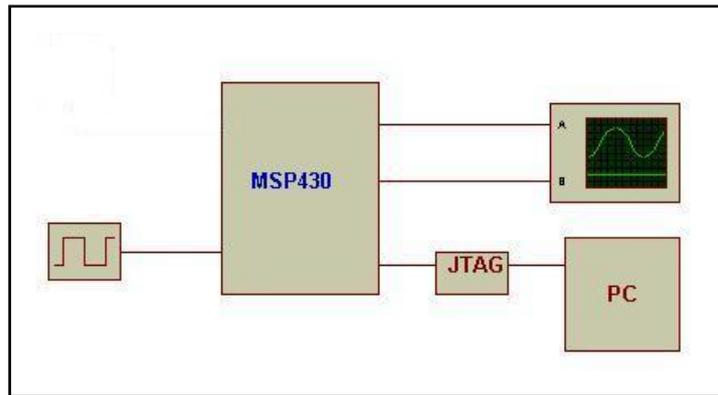


Figura 6.1. Esquema del montaje para las medidas utilizando una señal cuadrada.

En la entrada del microcontrolador se inserta la señal cuadrada. En la salida correspondiente se conecta el osciloscopio por el cual se visualizará el comportamiento del sistema programado. Por un lado se visualiza la salida del comparador (una señal cuadrada invertida) y por otro, la salida de un puerto de señalización. Este puerto tiene un nivel alto cuando el sistema conmuta del modo reposo al modo activo y vuelve a tener un nivel bajo cuando finaliza la medida. En la figura 6.2 se muestra lo explicado.

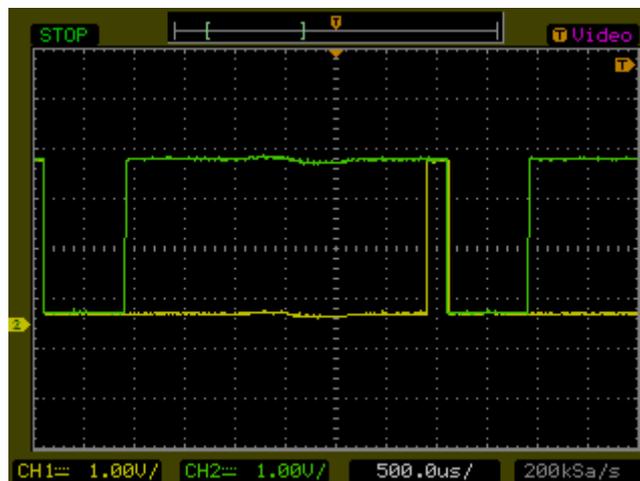


Figura 6.2. Señal de testeo y del comparador observadas en el osciloscopio.

Por último el ordenador mediante el JTAG alimentará al MSP430.

Como se ha indicado anteriormente, el sistema realiza medidas repetidamente mientras está encendido y recibe la señal cuadrada. Para posteriormente caracterizar el sistema, las variables que forman la medida se guardarán en la memoria. Para ello hay que tener en cuenta el espacio total de memoria y la

frecuencia a la que se guardarán los datos. De este modo se calcula el tiempo que el sistema debe permanecer encendido para obtener el número máximo de medidas. Para caracterizar el sistema se obtendrán aproximadamente mil medidas.

En cuanto esas medidas se obtienen, se introducen en el algoritmo explicado anteriormente (apartado 4.3.1) y se generan los resultados que más adelante se analizarán.

Medidas de consumo

Para las medidas de consumo se ha diseñado un circuito externo, al que llamaremos circuito conversor, que permite visualizar en el osciloscopio el voltaje proporcional a la corriente consumida.

Utilizando las características del TL074 se ha diseñado el circuito de la siguiente manera: El Amplificador Operacional se conectaría al MSP430 mediante un *jumper* que interrumpe la alimentación de éste. El pin positivo del A.O. se conecta al pin que recibe los 3 V del *jumper* y el pin negativo del A.O. se conecta al pin que va al MSP430. Debido a la naturaleza propia del TL074, la tensión del terminal positivo se refleja en el negativo.

La corriente que consume el MSP430 pasa a través del A.O. y mediante una resistencia de 100 kΩ conectada entre el pin de salida y el pin negativo se refleja una tensión proporcional a la corriente consumida.

Para obtener la corriente de consumo aplicaremos la fórmula

$$I_c = \frac{V_o - V_1}{R_1} \quad (6.1)$$

A continuación se muestra en la Figura 6.3 un esquema del circuito.

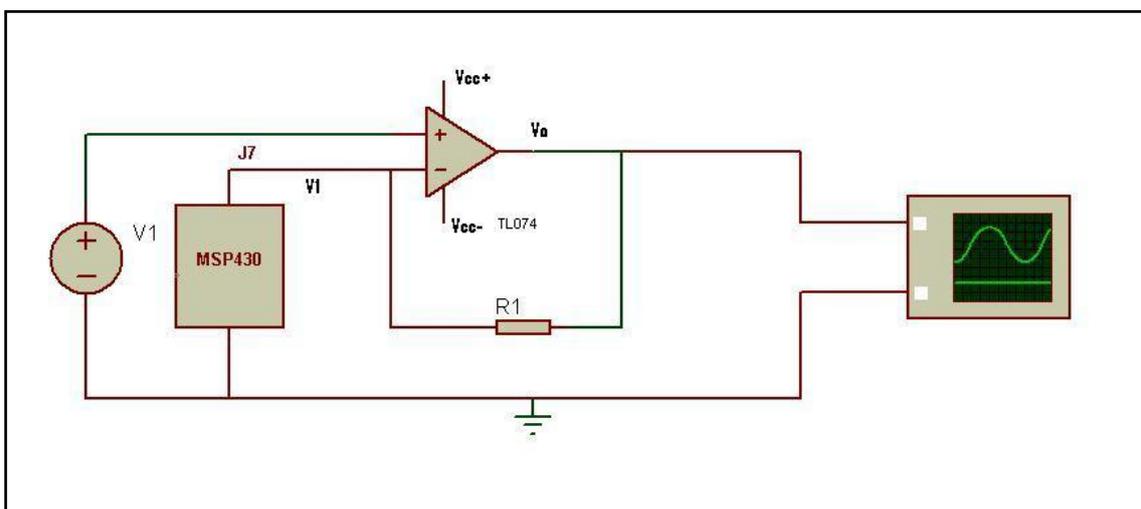


Figura 6.3. Circuito diseñado para la visualización del consumo en V.

El consumo en cada estado se puede obtener aproximadamente mediante la fórmula explicada en el apartado anterior. A pesar de esto, resulta más fácil comparar las prestaciones de los sistemas si se mide la potencia media de cada uno a través de la corriente media, medida con un amperímetro de gran precisión. Por eso también se ha procedido a realizar estas medidas.

A través del mismo *jumper* en el que se ha hecho la medida anterior, se ha medido la intensidad directamente con un amperímetro de alta precisión obteniendo la intensidad media de consumo. Con este valor y sabiendo que la alimentación es de 3 V, se podrá calcular la potencia media con la siguiente fórmula.

$$P = V \cdot I \quad (6.2)$$

6.2 RESULTADOS SIN CONTROL DINÁMICO DE POTENCIA

6.2.1 ACLK

En este apartado se exponen los resultados de la medida hecha con el sistema basado en el reloj ACLK, el cual utiliza el oscilador de cuarzo con una frecuencia de 32 768 Hz.

En primer lugar, en la figura 6.4 se puede observar que los resultados oscilan de forma periódica entre dos medidas únicamente, 3.204 ms y 3.235 ms. El motivo de estos dos valores únicos es, por un lado, que el generador no es 100% exacto y tampoco 100% preciso, por lo que el período generado no tiene siempre la misma duración; y por otro lado, la incapacidad del contador para aproximarse más al valor de tiempo real del período generado por el generador, esto es debido al gran error de cuantificación que causa la baja frecuencia del oscilador de cuarzo. Además, que estos valores varíen de forma periódica es consecuencia de que la frecuencia de la señal a medir no es un múltiplo exacto de la señal del oscilador que realiza la medida.

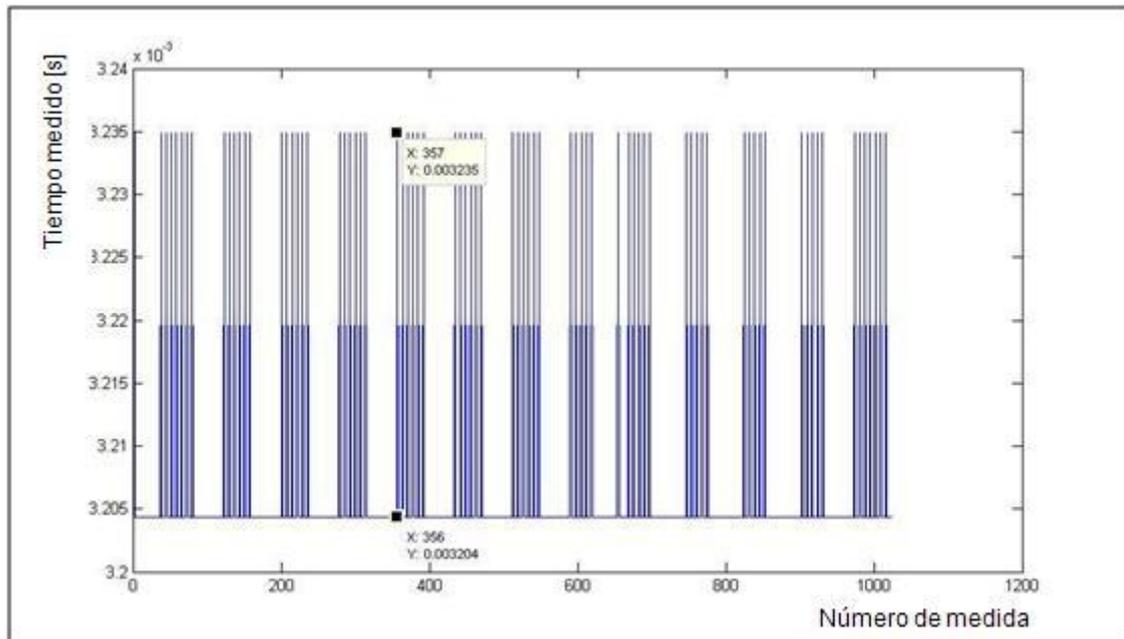


Figura 6.4. Resultado de medidas realizadas con el ACLK.

De estas medidas se puede extraer un gráfico que indica la probabilidad que existe de obtener un resultado u otro (figura 6.5).

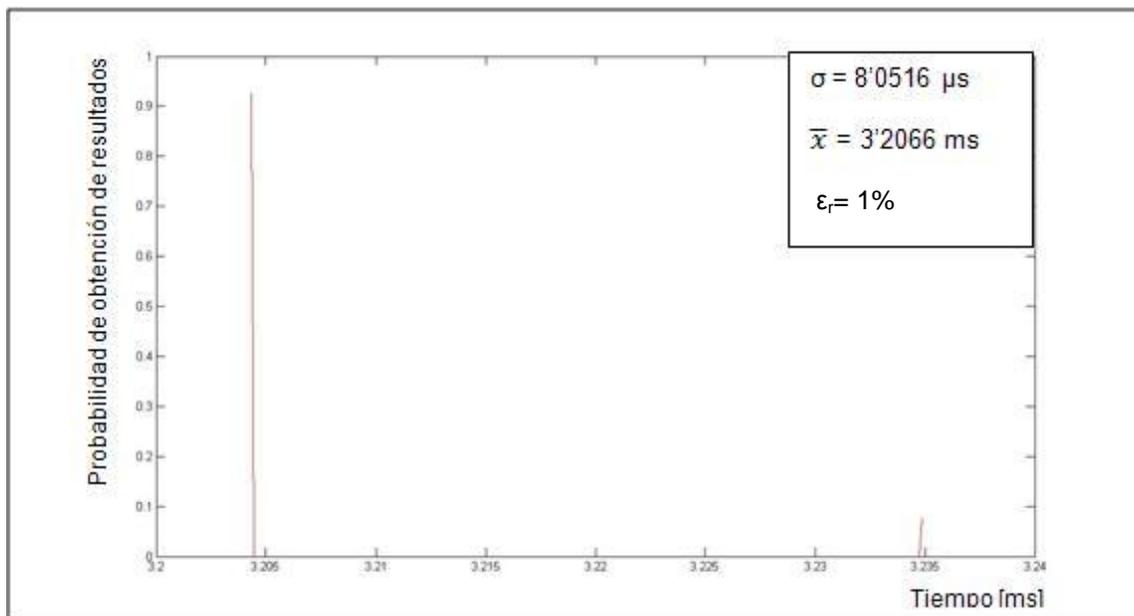


Figura 6.5. Gráfico de probabilidad del sistema basado en el ACLK.

Este gráfico muestra de forma más clara lo afirmado anteriormente. Además, se observa que la probabilidad de obtener un resultado de medida de 3.204 ms es mucho mayor que la probabilidad de obtener un resultado de 3.235 ms, esto significa que el valor real medio está más próximo al primer resultado que al segundo. Además, este valor real medio no corresponde con el valor de 3.2 ms esperado (explicado en el apartado anterior), sino que debe de estar ligeramente por encima de los 3.204 ms, porque de lo contrario se observaría cierta probabilidad de obtener un valor por debajo de éste en lugar de por encima. El motivo de esto es que el generador no sea 100% exacto y por lo tanto no genera una señal de 3.2 ms de semiciclo sino ligeramente diferente. También se puede observar la desviación estándar σ , la media \bar{x} y el error relativo ϵ_r , variables que caracterizan mejor el sistema y que permitirán compararlo con los dos sistemas siguientes con más detalle.

Para caracterizar el sistema a nivel de consumo energético se obtienen las gráficas de la figura 6.6.

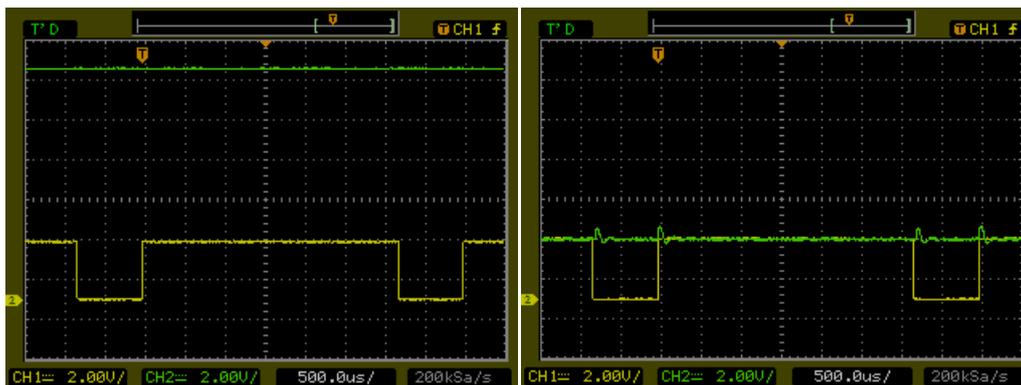


Figura 6.6. Gráficas del sistema en modo activo y en modo de bajo consumo.

Ambas gráficas muestran en amarillo la señal generada por el comparador y en verde la señal que proviene del circuito conversor explicado anteriormente. La primera gráfica se ha obtenido con el sistema configurado para permanecer siempre encendido y la segunda se ha obtenido con el sistema configurado en bajo consumo durante todo momento.

La primera gráfica muestra un nivel constante de corriente debido a que se mantiene el modo activo en todo momento.

En la segunda gráfica también se observa un solo nivel de corriente y un pico cada flanco de la señal generada por el comparador. Estos picos reflejan el incremento de consumo correspondiente a la ejecución de cada interrupción por parte de la CPU; deberían alcanzar el valor del nivel reflejado en la primera gráfica pero debido al *slew-rate* del A.O. que forma el circuito conversor y a la corta duración de la interrupción éste no llega a alcanzarse.

En general se puede apreciar que el consumo ha disminuido muchísimo por el simple hecho de pasar a otro modo de funcionamiento. Si se mide con un multímetro de alta precisión la corriente media que consume el sistema con cada configuración se obtiene 9.15158 mA en el primer caso y 0.10259 mA en el segundo caso, que corresponde a una potencia de 27.45474 mW y 0.30777 mW respectivamente. Con esta medida se puede observar que las gráficas mostradas son totalmente coherentes con la configuración en cada caso: en el caso de pasar al modo de bajo consumo disminuye bastante el nivel de tensión del circuito conversor y el consumo medio también disminuye bastante.

6.2.2 SMCLK (DCO)

En este apartado se exponen los resultados de la medida hecha con el sistema basado en el reloj SMCLK, el cual utiliza el oscilador controlado digitalmente con una frecuencia de 16 MHz aproximadamente.

En este caso, la figura 6.7 muestra que el número de resultados diferentes ha aumentado respecto al sistema anterior, y estos oscilan entre los 3.204 ms y los 3.209 ms aproximadamente. El motivo de que ahora el margen de medidas sea menor vuelve a ser por el oscilador del reloj, que en este caso tiene una frecuencia de 16 MHz, lo que implica un error de cuantificación mucho más pequeño y que el resultado se aproxime mejor al valor real del generador. Como en el caso anterior se producen diferentes resultados debido a la precisión del generador; pero en este caso son muchos en lugar de dos a causa de la mayor resolución del reloj basado en el DCO, que permite detectar mejor el valor real. También hay que decir que la inestabilidad del oscilador DCO, mayor que la del cristal de cuarzo, influye en la variedad de estas medidas⁵.

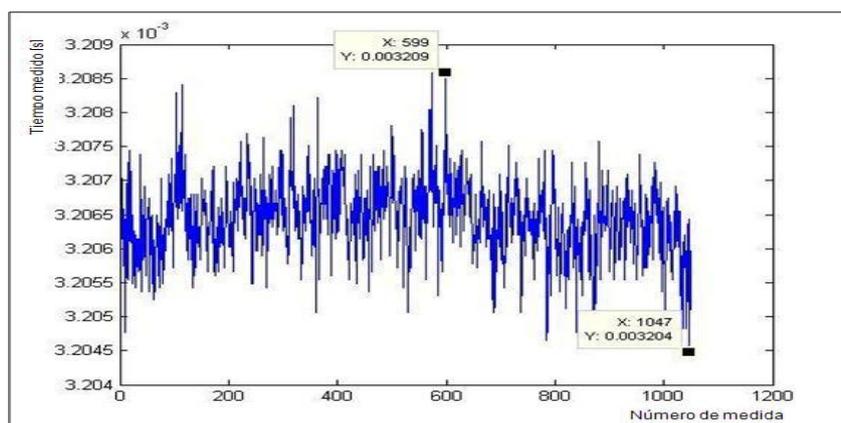


Figura 6.7. Resultado de medidas realizadas con el DCO.

⁵ En el anexo C se hace una comparación entre la estabilidad del oscilador de cuarzo y la estabilidad del DCO.

Como en el caso anterior, de estas medidas se puede extraer un gráfico que indica la probabilidad que existe de obtener un resultado u otro (figura 6.8).

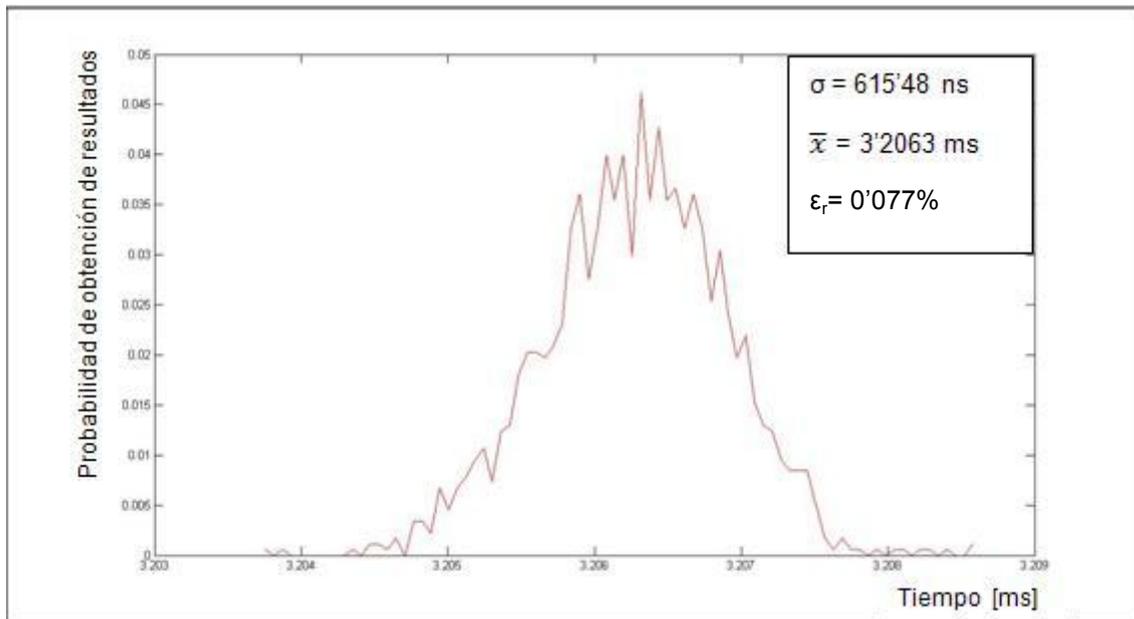


Figura 6.8 Gráfico de probabilidad del sistema basado en el SMCLK (DCO).

En este caso se puede apreciar una forma más común, más esperada en general. El gráfico muestra un margen de valores cuya probabilidad de obtenerse como resultado de medida disminuye a medida que se aleja del valor medio o la media representada en la figura. Esta media corresponde con el valor real medio, generado por el generador, y debería coincidir con el valor esperado propuesto al principio del capítulo (3.2 ms); el motivo de que esto no ocurra vuelve a ser que el generador no sea 100% exacto. También se muestra la desviación estándar y el error relativo. Tal y como se esperaba la desviación estándar se ha reducido debido al aumento de resolución del sistema, aunque el valor que tiene se justifica por la estabilidad del DCO principalmente. Esta desviación menor, en el sistema anterior implica un error relativo también menor.

Como en el sistema anterior, para caracterizar el sistema a nivel de consumo energético se obtienen las gráficas de la figura 6.9.

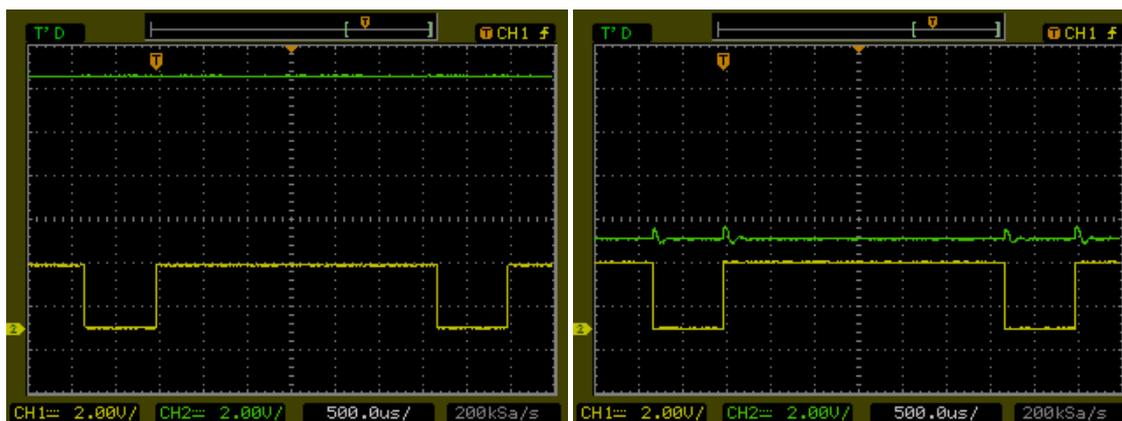


Figura 6.9 Gráficas del sistema en modo activo y en modo de bajo consumo

Ambas gráficas muestran las mismas señales que en el apartado anterior y en las mismas condiciones.

La primera gráfica aparece aproximadamente igual que en el caso del ACLK porque se ha obtenido con el sistema configurado para permanecer siempre encendido lo que implica un consumo similar ya que está todo activo.

En la segunda gráfica se observa diferencia respecto al sistema basado en el ACLK. En este caso el nivel de corriente es mayor debido a que sólo se apaga la CPU porque el DCO se está utilizando para hacer la medida. También se observan los picos de consumo debidos a la ejecución de las interrupciones, como ya se ha explicado en el sistema anterior.

Se puede apreciar que el consumo baja mucho cuando se apaga la CPU, pero es mayor que usando el ACLK. Las medidas de corriente media demuestran esto, se obtienen 9.17735 mA en el primer caso y 1.21218 mA en el segundo caso, correspondientes a 27.53205 mW y a 3.63654 mW respectivamente.

6.3 RESULTADOS CON CONTROL DINÁMICO DE POTENCIA

6.3.1 MÉTODO PROPUESTO: ACLK + DCO

En este apartado se exponen los resultados de la medida hecha con el sistema basado en la combinación del reloj ACLK y del reloj DCO.

En la figura 6.10 se aprecia que estos resultados varían entre 3.188 ms y 3.19 ms aproximadamente, por lo tanto se ha superado el error de cuantificación que tenía el sistema basado en el oscilador de cuarzo, y se ha obtenido el menor error de cuantificación que proporciona el oscilador DCO. Por otro lado, el gráfico presenta un aspecto similar al gráfico que representaba los resultados del sistema basado en el DCO, esto es debido a la estabilidad del oscilador DCO junto a la precisión del generador como ya se había comentado.

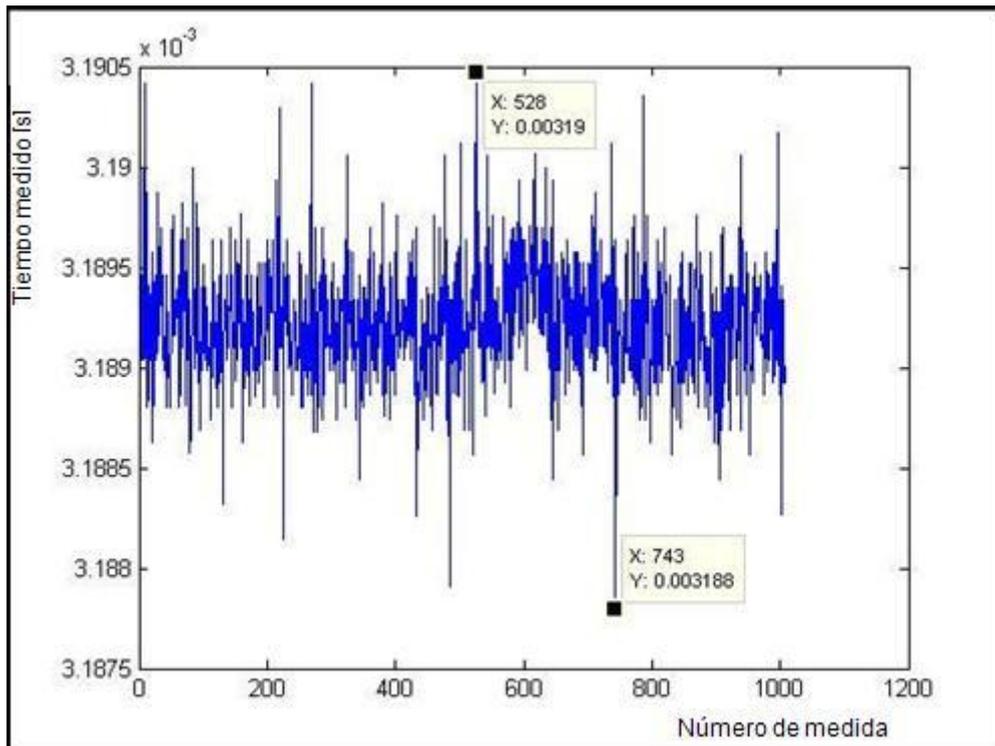


Figura 6.10 Resultado de medidas realizadas con el sistema combinado

De igual modo que en los apartados anteriores, el siguiente gráfico (figura 6.11) muestra la probabilidad de obtener un resultado u otro utilizando este método.

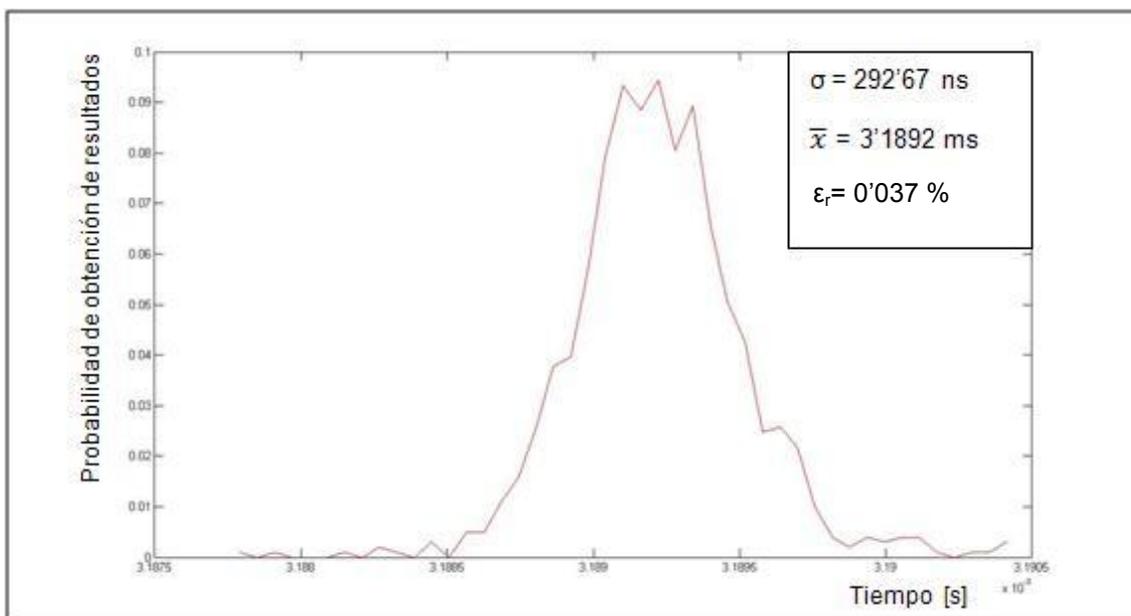


Figura 6.11 Gráfico de probabilidad del sistema propuesto

En este gráfico se aprecia una forma similar al caso anterior. El motivo vuelve a ser el mismo, que la resolución del DCO permite obtener varios valores debido a la precisión no exacta del generador. Se sigue observando un valor medio cuya probabilidad es mayor a la del resto de valores. Este valor, correspondiente a la media calculada y mostrada, debería coincidir con el valor medio del caso anterior; aparte del motivo de la exactitud del generador, el motivo por el cual esto no ocurre es que la mayor complejidad de código en este sistema implica cierto retardo al obtener la medida; de este modo los resultados obtenidos son ligeramente inferiores y por lo tanto el valor medio también. En el gráfico también se muestran los valores de desviación estándar y error relativo. Se observa que éstos han disminuido bastante respecto al sistema anterior, esto es debido a que ahora el sistema basa la medida principalmente en el oscilador de cuarzo que tiene muy buena estabilidad, por lo tanto la estabilidad general ha mejorado.

En el aspecto referente a consumos de energía se presenta la siguiente figura (figura 6.12), que muestra un gráfico del sistema en pleno funcionamiento y un segundo gráfico que muestra de forma ampliada el momento en que termina la medida y comienza la siguiente.

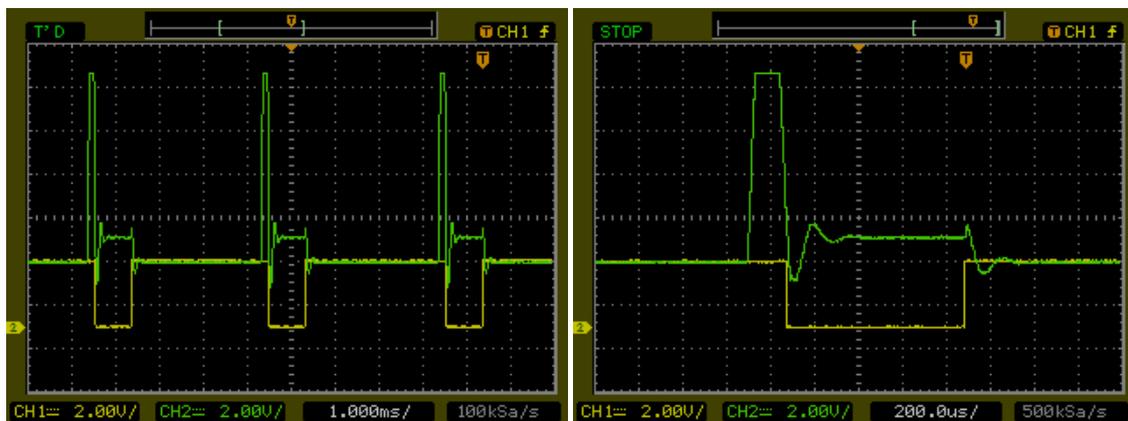


Figura 6.12 Gráficos del sistema en plena medida

En el primer gráfico se observan tres niveles de corriente diferentes. Se contempla que la mayor parte de tiempo el sistema consume muy poco, esto ocurre desde que comienza el período a medir hasta que termina, el motivo es que el sistema está funcionando en el modo de bajo consumo LPM3, en que la CPU y DCO se mantienen apagadas. En el segundo gráfico se pueden apreciar con más detalle los otros dos niveles. El primero se produce cuando el período a medir va a terminar, el motivo es que el sistema pasa a modo activo automáticamente para poder ejecutar una serie de interrupciones consecutivas programadas en este momento; esto implica el máximo consumo puesto que el sistema está activo completamente. A continuación este nivel cae de nuevo a

un tercer nivel que se mantiene durante el semiciclo no perteneciente al período a medir, esto es debido a que la CPU se apaga pero se mantiene encendido el reloj basado en el DCO para poder calcular el tiempo t_p descrito en el capítulo anterior.

Con esta figura se ha podido observar que el consumo del sistema que combina los dos relojes parece ser menor que el consumo del sistema basado en el DCO.

Respecto a la corriente media medida en el sistema combinado son 0.6114 mA, correspondientes a una potencia media de 1.8342 mW. Con esta medida se demuestra de forma clara que el sistema combinado permite disminuir el consumo de corriente respecto al sistema basado en el DCO.

Y para terminar, la figura siguiente (figura 6.13) muestra una gráfica que indica el consumo medio de corriente del dispositivo en función de diferentes períodos de tiempo medidos.

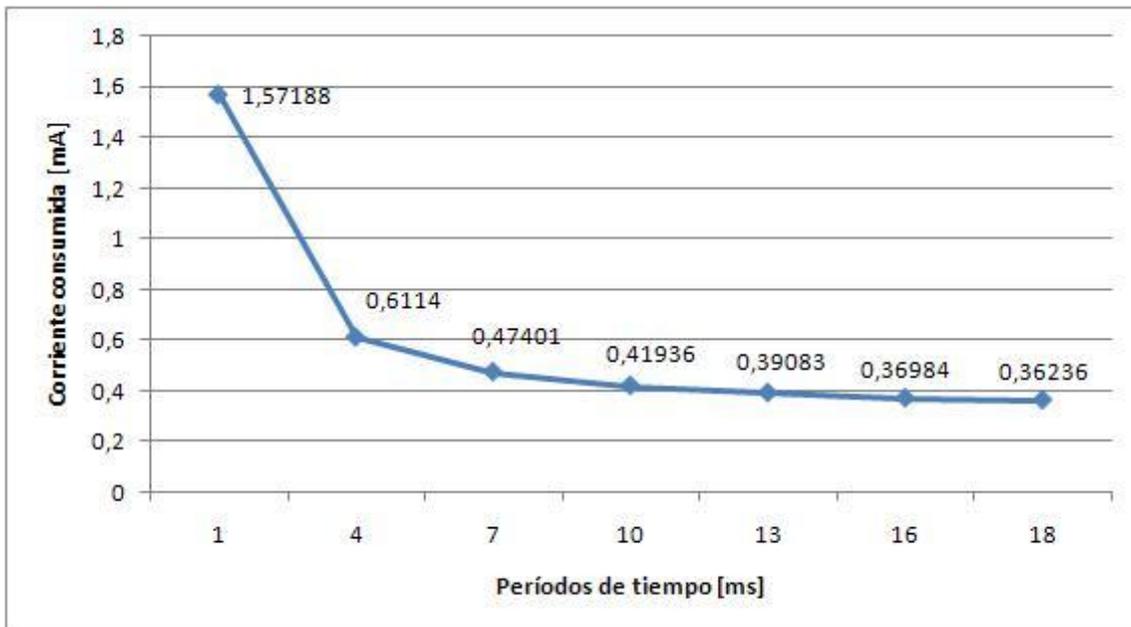


Figura 6.13 Corriente consumida en función de los períodos de tiempo medidos.

La forma de esta gráfica se justifica principalmente con la señal que se introduce para determinar los períodos de tiempo a medir. Esta señal tiene una asimetría del 20 %, por lo tanto el período a medir se corresponde con el 80 % restante. Como la mayor parte de la medida se realiza con el reloj ACLK y el sistema en bajo consumo, cuanto mayor sea el período de tiempo a medir más se aproximará el consumo medio al consumo del ACLK, correspondiente a 0.25463 mA tal y como se ha medido anteriormente.

CAPÍTULO 7. CONCLUSIONES

Los sistemas autónomos que se instalan hoy en día deben tener un consumo muy reducido a causa de la falta de fuentes de energía. A raíz de esta necesidad ha surgido este proyecto que tiene como pilar principal el bajo consumo.

Así pues, después de haber analizado los resultados obtenidos y de haberlos comparado entre sí, se han obtenido las siguientes conclusiones.

En el proceso de medida de tiempos existe un compromiso entre el consumo energético y la precisión de la medida. Es un factor que se ha tenido en cuenta durante todo el proyecto y con el cual se ha llegado al sistema final propuesto.

Se ha demostrado que con un sistema híbrido entre los dos osciladores de los que se dispone en el MSP430, se consigue un consumo muy parecido a cuando sólo está en funcionamiento el ACLK y se llega a tener una resolución igual a la que proporciona el oscilador DCO. Eso hace que de la unión de los dos relojes se haya heredado sus ventajas dando un resultado de bajo consumo y pequeño error de cuantificación.

También se ha podido ver que la diferencia de consumo entre el oscilador de cuarzo y el DCO no era tan acentuada como se preveía, todo y que éste último tiene un consumo más elevado. En consecuencia se ha extraído de los resultados que una estrategia muy importante a seguir en el ámbito del ahorro de energía es el uso adecuado de la CPU. Una gran disminución del consumo del sistema se ha conseguido al apagar la CPU en momentos en los que no hacía falta su actividad.

En este último ámbito del ahorro de consumo, se puede afirmar que la ventaja que el MSP430 tiene de poder conmutar entre sus relojes y entre modos de consumo, lo hacen una opción muy interesante para la implementación de sistemas de alto rendimiento y muy bajo consumo.

Desde el punto de vista medioambiental este proyecto tiene mucha relación debido a que el objetivo principal es conseguir un mínimo consumo energético; esto le hace más propicio a basar su alimentación en fuentes de energía renovables como por ejemplo podría ser el caso de la energía solar.

7.1. Mejoras y ampliaciones del proyecto

Por otro lado, de cara a la mejora y ampliación del proyecto hay varias propuestas:

El sistema tiene la limitación de medir tiempos mayores de 2 segundos debido a limitaciones del propio hardware. Teniendo en cuenta que el registro contador

del Timer A se incrementa a una frecuencia de 32,768 Hz, y que éste es de 16 bits, se obtiene que el máximo tiempo a medir es de 2 segundos. Por otro lado, refiriéndose al valor mínimo que el sistema puede medir, teóricamente sería de aproximadamente 30 μ s, equivalente a un incremento del contador del Timer A.

Una posible mejora sería ampliar este rango de medida desplazando el límite inferior o el superior dependiendo de la aplicación que se quiera desarrollar con el sistema. Esto implicaría utilizar una combinación de varios registros para contabilizar el número de desbordamientos producidos por el contador del Timer A en el caso de querer aumentar el límite superior; o una configuración diferente del algoritmo programado en el MSP430 para poder utilizar el Timer B basado en el DCO sin depender del Timer A en el caso de querer disminuir el límite inferior, de este modo se podrían medir períodos de tiempo desde 60 ns, equivalentes a un incremento del contador del Timer B.

Otra limitación del algoritmo diseñado está en la necesidad de programar el sistema en función del período a medir, es decir, actualmente hay que inicializar el valor de uno de los registros que intervienen en el proceso de medida para que el sistema enganche con el período a medir. Una mejora podría ser la autoinicialización de este registro una vez recibe la señal a medir. Esto haría el sistema mucho más flexible y robusto.

Otro aspecto a mejorar es la posibilidad de poder medir señales que no sean sólo cuadradas como es el actual caso. En el caso de introducir una señal rampa muchos problemas que no se encuentran en la señal cuadrada (el ruido de disparo por ejemplo) aparecen, con lo cual el diseño de un algoritmo que pueda soportar ese tipo de señal sería un avance muy importante.

BIBLIOGRAFÍA

- [1] Luca Benini and Giovanni De Micheli, "Dynamic Power Management, Design Techniques and CAD Tools", Kluwer Academic Publishers 1998
- [2] Ferran Reverter y Ramón Pallàs Areny, "Circuitos de interfaz directa sensor-microcontrolador", Marcombo ediciones técnicas 2005
- [3] Michael Bay, "MSP430 Power Solutions, Portable Power Applications Manager" MSP430 Advanced Technical Conference 2008
- [4] Trevor Pering, Tom Burd, and Robert Brodesen "Dynamic Voltage Scaling and the desing of a Low Power Microprocessor System" University of California Berkeley
- [5] Agilent Technologies "Application Note 200 Electronic Counter Series, Fundamental of the Electronic Counters" 1997
- [6] Prof. Rafael Martin Lamaison "Energía Solar Fotovoltaica" 2004

Fuentes de Internet:

www.ti.com

<http://www.huarpe.com/electronica/osc/oscilador-xtal.html>

http://en.wikipedia.org/wiki/Crystal_oscillator

http://es.wikipedia.org/wiki/Oscilador_LC

http://en.wikipedia.org/wiki/Dynamic_voltage_scaling

<http://dynamicpower.sourceforge.net/>

<http://atmel.com/products/AT91/>

<http://mspgcc.sourceforge.net/manual/book1.html>

ACRÓNIMOS

ACLK Auxiliar Clock

CA Comparador A

CPU Central Processing Unit

DCO Digital Controlled Oscillator

DFS Dynamic Frequency Scaling

DPM Dynamic Power Management

DVS Dynamic Voltage Management

EEPROM Electrically-Erasable Programmable Read-Only Memory

JTAG Joint Test Action Group

KERS Kinetic Energy Recovery System

LPM Low Power Mode

MCLK Main Clock

MCU Microcontroller

MPU Multiple Process Unit

POS Point of Sale

RAM Random Access Memory

RISC Reduced Instruction Set Computer

RPM Revoluciones Por Minuto

SMCLK Sub-Main Clock

ST Schmitt Trigger

TAR Timer A Register

TBR Timer B Register

TACCR Timer A Capture/Compare Register

TACCTL Timer A Capture/Compare Control

TI Texas Instruments

VLO Very Low Power Oscillator

ANEXO A. APLICACIÓN PANEL SOLAR

Un panel solar se encarga de transformar la energía solar que capta en energía eléctrica. Dependiendo de la tensión eléctrica a la que se entregue esa energía a una carga la intensidad de la corriente será mayor o menor, obteniendo así una relación corriente – tensión y por lo tanto una relación potencia – tensión. Esta última relación tiene un punto máximo a determinada tensión que implica la entrega máxima de potencia y a su vez el funcionamiento óptimo del panel solar. Así que basta con mantener de algún modo cierta tensión en la salida del panel solar para extraer la máxima potencia de éste.

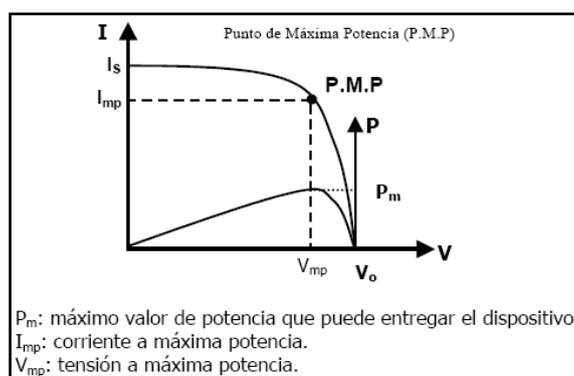


Figura B.1. Curva característica de un panel solar⁶.

El problema es que esta tensión varía en función de la luz que recibe la placa, por lo tanto es necesario algún dispositivo que se encargue a la vez de medir directa o indirectamente la potencia que se está entregando y en función de eso mantenga la tensión óptima. La técnica empleada consiste en colocar un condensador entre los bornes del panel solar de tal modo que se cargue cuando el panel recibe luz. En este caso un módulo del dispositivo se encarga de medir cuánto tiempo tarda en cargarse el condensador, y otro módulo se encarga de descargarlo y cargarlo consecutivamente entorno al valor de tensión óptimo del panel solar. Conociendo el tiempo que tarda en cargarse el condensador y la relación que existe con el resto de parámetros que lo caracterizan se puede saber la relación energía – tensión que está recibiendo y por lo tanto en qué momento volverlo a descargar. De este modo es la carga del condensador quien fija una tensión en la salida del panel solar, de forma variable y dependiendo de la velocidad de carga del condensador.

⁶ Fuente: “Energía Solar Fotovoltaica”, de Prof. Rafael Martin Lamaison.

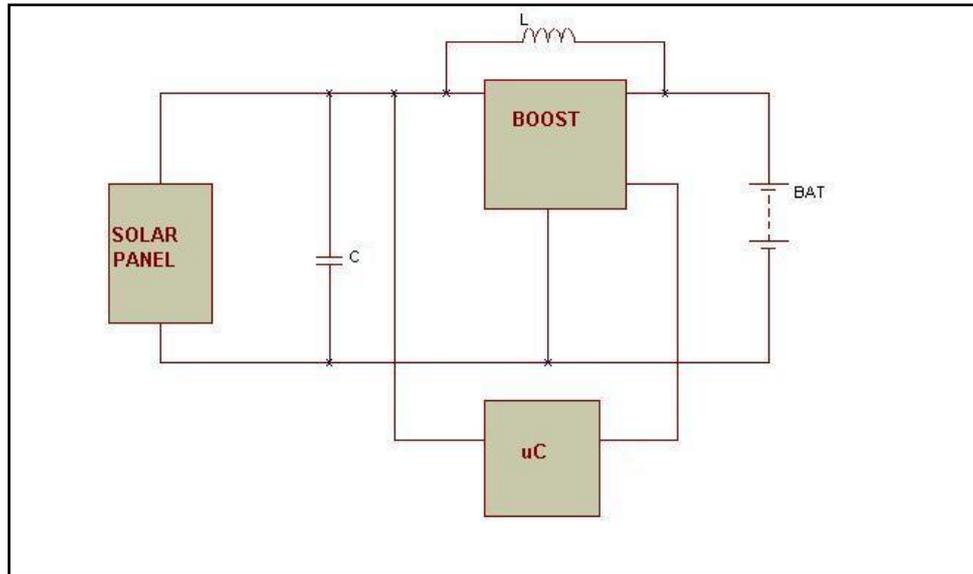


Figura B.2. Diagrama de bloques del sistema.

Actualmente existen dispositivos diseñados para medir la potencia que entrega el panel y de este modo mantener en la salida una tensión lo más cercana posible a la tensión óptima forzando así que la entrega de energía sea máxima. De esta manera se consigue un incremento de energía a pesar del consumo energético del dispositivo. El problema se encuentra en paneles solares pequeños en los que el incremento de energía conseguido no compensa con el consumo del dispositivo; en estos casos se requiere un rediseño de dicho dispositivo con el único objetivo de bajar el consumo.

ANEXO B. ESTABILIDAD DCO

En este anexo se estudia la estabilidad del oscilador DCO.

En apartados anteriores la estabilidad del DCO se caracteriza por ser mucho peor en comparación con la estabilidad del oscilador de cuarzo. Para demostrar esto se programa un algoritmo en el microcontrolador que mide repetidamente con el Timer B un período de tiempo generado por el Timer A. Cada timer está programado para utilizar el oscilador de cuarzo y el oscilador DCO respectivamente.

En primer lugar se inicializa un registro de comparación con el valor equivalente al tiempo inicial, por ejemplo un 2 y un segundo registro de comparación con el valor equivalente al tiempo final, por ejemplo un 100. El período a medir por el Timer B será el tiempo comprendido entre ambos valores.

Por otro lado se habilita un registro de captura del Timer B que captura el valor del contador TBR en el momento que se manda la orden.

A continuación se habilita la interrupción de cada registro que se ejecutará cuando el contador del Timer A alcance los valores inicializados.

La interrupción del primer registro de comparación únicamente enciende el Timer B. La segunda interrupción provoca la captura del TBR y guarda el valor capturado en un espacio de la memoria y vuelve a iniciar el contador del Timer A.

Una vez se han llenado varios espacios de memoria con resultados se representan en el gráfico que muestra la siguiente figura.

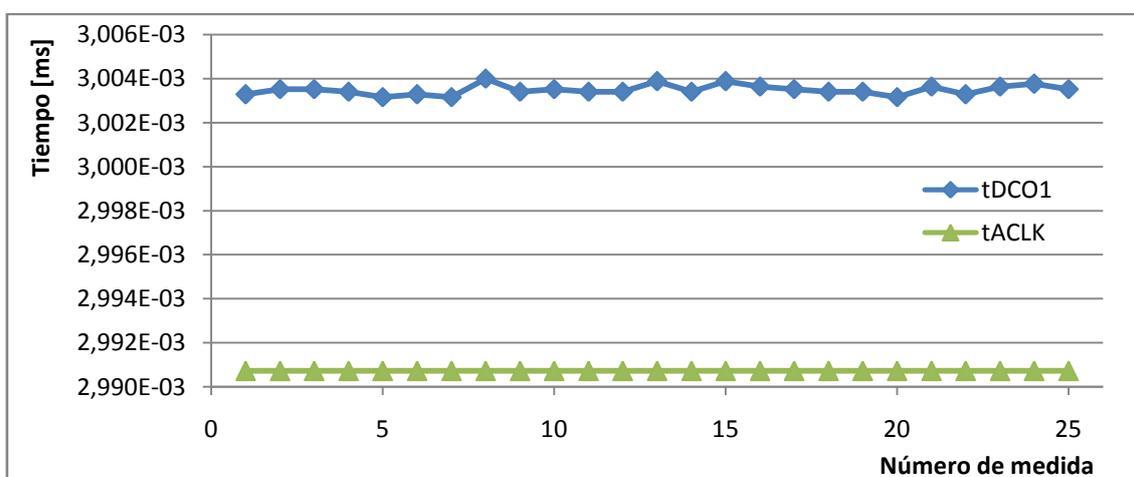


Figura C.1. Medida de un período mediante el oscilador de cuarzo y el oscilador DCO.

En este gráfico se muestra que para un período concreto que se repite frecuentemente el Timer B ofrece diferentes resultados. Estos resultados oscilan en un margen de tiempo equivalente a un par de ciclos del oscilador DCO arriba o abajo.

Con este estudio se demuestra que la estabilidad del DCO tiene cierta importancia en comparación con la del oscilador de cuarzo. A pesar de esto, esta variación no es un problema para el resultado final de las medidas realizadas en el proyecto ya que se trata de una variación relativamente pequeña.

```

; programa
    mov.w    #1100h,R10           ; 1a dirección donde guarda Tdco
    mov.w    #64h,&TACCR0        ; ponemos un 100 en TACCR0
    mov.w    #02h,&TACCR1        ; ponemos un 2 en TACCR1
    bis.w    #TACLR,&TACTL        ; ponemos a 0 el TAR
    bis.w    #MC0,&TACTL         ; encendemos timer A
    bis.w    #CCIE,&TACCTL1      ; interrupción TACCR1 habilitada
    bis.w    #CCIE,&TACCTL0      ; interrupción TACCR0 habilitada
    jmp      $

; interrupciones
int1  bis.w  #TBCLR,&TBCTL        ; ponemos a 0 el TBR
      bis.w  #MC1,&TBCTL         ; encendemos timer B
      bic.w  #CCIFG,&TACCTL1     ; bajamos bandera
      reti

int2  xor.w  #CCIS0,&TBCCTL0     ; captura manual del timer B
      bic.w  #MC0,&TACTL         ; paramos timer A
      bic.w  #MC1,&TBCTL         ; paramos timer B
      mov.w  TBCCR0,0(R10)       ; guardamos Tdco en el puntero
      inc.w  R10                 ; incrementamos puntero
      bis.w  #TACLR,&TACTL        ; ponemos a 0 el TAR
      bis.w  #MC0,&TACTL         ; encendemos timer A
      reti

```


; configuramos el ComparadorA

bis.b #CAREF0,&CACTL1 ; VCAREFF = 0.25*Vcc

bis.b #P2CA1,&CACTL2 ; pin CA1 terminal -

bis.b #CAF,&CACTL2 ; habilitamos el filtro RC de la salida

; configuramos el timer A

bis.w #100h,&TACTL ; clock origen ACLK

; configuramos el timer B

bis.w #200h,&TBCTL ; clock origen SMCLK

bic.w #100h,&TBCTL

; configuramos el registro de captura del Timer A

bis.w #CAP,&TACCTL1 ; modo captura

bis.w #CM1,&TACCTL1 ; captura en flanco de bajada

bis.w #SCS,&TACCTL1 ; sincronizamos señal de entrada

bis.w #CCIS0,&TACCTL1 ; captura salida del comparador CCI1B

; configuramos el registro de captura del Timer B

bis.w #CAP,&TBCCTL0 ; modo captura

bis.w #CM1,&TBCCTL0 ; captura en flanco de bajada

bis.w #SCS,&TBCCTL0 ; sincronizamos señal de entrada

; configuramos el programa

mov.w #1100h,R10 ; 1a dirección donde guarda Tdco2

mov.w #1900h,R11 ; 1a dirección donde guarda Taclk2

mov.w #2100h,R12 ; 1a dirección donde guarda Taclk

mov.w #2900h,R13 ; 1a dirección donde guarda Tdco

mov.w #03h,&TACCR2 ; TimerB se enciende 3 ciclos después
de CPU ON

; código principal

mov.w #40h,&TACCR0 ; inicializamos con valor aproximado al
momento de despertar (40h)

bis.b #CAIE,&CACTL1 ; habilitamos interrupción del comparador

bis.b #CAON,&CACTL1 ; encendemos el comparador

```
    bis.w  #cpuoff,SR          ; apagamos CPU para que no afecte a la
medida
```

```
    nop                        ; esperamos a que llegue el flanco de subida
```

```
; inicio del programa
```

```
    jmp    $
```

```
; código interrupciones en orden de ejecución:
```

```
; salta cuando llega el flanco de subida del comparador
```

```
int1  mov.w  TAR,R6           ; capturamos Taclk2
```

```
    mov.w  TBR,R5           ; capturamos Tdco2
```

```
    sub.w  R6,TAR           ; resta TAR - R6 y lo coloca en TAR
```

```
    bis.w  #MC0,&TACTL       ; UP mode, Timer A empieza a contar
```

```
    bic.w  #CCIFG,&TACCTL0   ; bajamos bandera TACCR0
```

```
    bis.w  #CCIE,&TACCTL0   ; interrupción TACCR0 habilitada
```

```
    bic.b  #CAON,&CACTL1    ; apagamos el comparador
```

```
    bis.w  #lpm3,0(SP)      ; entramos en bajo consumo 3
```

```
    reti
```

```
; salta cuando el Timer A llega al valor de TACCR0
```

```
int2  bic.w  #cpuoff+scg1+scg0,0(SP) ; volvemos al estado activo
```

```
    bis.b  #00010000b,&P1OUT ; nivel alto en P1.4 out
```

```
    bic.w  #CCIFG,&TACCTL2   ; bajamos bandera TACCR2
```

```
    bis.w  #CCIE,&TACCTL2   ; interrupción TACCR2 habilitada
```

```
    bic.w  #CCIE,&TACCTL0   ; interrupción TACCR0 deshabilitada
```

```
    reti
```

```
; salta 3 ciclos ACLK después de despertar para encender el TimerB en
momento estable del DCO (TACCR2)
```

```
CCIFG_2 bis.w  #TBCLR,&TBCTL   ; ponemos a 0 el TBR
```

```
    bis.w  #MC1,&TBCTL       ; encendemos timer B
```

```
    bic.w  #CCIFG,&TACCTL2   ; bajamos bandera TACCR2
```

```
    bis.w  #CCIE,&TACCTL1   ; interrupción TACCR1 habilitada
```

```
    bic.w  #CCIE,&TACCTL2   ; interrupción TACCR2 deshabilitada
```

```

    bis.b  #CAON,&CACTL1      ; encendemos el comparador
    bis.w  #cpuoff,0(SP)      ; apagamos CPU para que no afecte a la
medida
    reti

; salta cuando llega el flanco de bajada del comparador (TACCR1)
CCIFG_1 bic.b  #00010000b,&P1OUT    ; nivel bajo en P1.4 out
    bic.w  #CCIFG,&TACCTL1      ; bajamos bandera TACCR1
    mov.w  TACCR0,R7           ; capturamos Tack
    mov.w  TBCCR0,R8           ; capturamos Tdco
    add.w  TACCR1,&TACCR0      ; suma
    sub.w  #05h,&TACCR0        ; restamos 5
    mov.w  R5,0(R10)           ; guardamos Tdco2 en el puntero
    mov.w  R6,0(R11)           ; guardamos Tack2 en el puntero
    mov.w  R7,0(R12)           ; guardamos Tack en el puntero
    mov.w  R8,0(R13)           ; guardamos Tdco en el puntero
    inc.w  R10                  ; incrementamos puntero
    inc.w  R11                  ; incrementamos puntero
    inc.w  R12                  ; incrementamos puntero
    inc.w  R13                  ; incrementamos puntero
    reti

; controlador interrupciones TAIV
int3  add.w  &TAIV,PC          ; según la interrupción que haya saltado
(TACCR1 o TACCR2)
    reti
    jmp    CCIFG_1
    jmp    CCIFG_2
    reti
    reti

; vectores interrupciones
    ORG    0FFF6h              ; interrupción comparador

```

```
DC16  int1
ORG   0FFF2h      ; interrupción TACCRO
DC16  int2
ORG   0FFF0h      ; interrupción TAIV
DC16  int3
ORG   0FFFEh      ; RESET
DC16  RESET
END
```


ANEXO D. DATASHEETS