Reglas de diseño de bajo consumo con MCU

Artículo cedido por Renesas Electronics

www.renesas.eu

Autor: David Parsons – Consultor para Renesas Electronics (Europe) GmbH.
Traducción: Óscar Alonso Estradé – Ingeniero de aplicaciones de Renesas Electronics (Europe) GmbH.

Resumen

Las demandas de mercado y gobierno para un entorno más verde y mejor uso de los recursos, tal como hacer baterías más duraderas, significa que la necesidad de funcionamiento de baja potencia y en tiempo de espera sigue aumentando. Por el contrario las demandas para mayores prestaciones y funcionalidades también están al alza, una combinación generalmente no identificada con la disminución de consumo.

Introducción

Éste es el segundo de una serie de cuatro documentos que examinan varias consideraciones para el diseño y operación del bajo consumo. En éste documento nos fijamos en muchos de los "trucos" (técnicas) que permiten a los diseñadores combinar ambas, prestaciones y bajo consumo en sus aplicaciones. Se incluyen ejemplos específicos y referencias que se basan en las familias del RL78 de 16 bits y del RX100 de 32 bits, descritas en el primer documento (que hacen funcionar el sistema idesde un limón!). Cabe señalar que mientras se utilizan dispositivos específicos para la referencia (RL78/L12 y RX111) los principios se pueden aplicar igualmente a las otras familias de los RL78 y RX.

"Trucos" para el bajo consumo

Cómo se indica en el documento 1, el microcontrolador (en adelante MCU) es un factor significativo en el consumo utilizado en las aplicaciones, pero no es siempre la única área a considerar y si bien los requisitos de la aplicación difieren, hay un conjunto general de temas que contribuyen a la potencia utilizada, que estudiamos en este documento.

1. Tiempos de espera del MCU
2. Velocidad de operación del reloj del MCU
3. Fuente de selección del reloj del MCU
4. Operación de periféricos del MCU
5. Uso de pines de entrada /salida del MCU
6. Integración del sistema
7. Opciones de fuentes de alimentación

Estos temas son abordados en detalle a continuación, pero consulte los otros artículos de la serie, ya que cubren algo más a fondo el tema específico asociado con diseño de bajo consumo. Los otros documentos de la serie aparecen referenciados final de este documento.

Modos de espera del MCU

Los modos de espera usados durante periodos de parada o inactivos son normalmente considerados como el método principal para la reducción del consumo medio en cualquier sistema alimentado con baterías.

Muchas familias MCU ofrecen una serie de opciones de bajo consumo porque las aplicaciones requieren diferentes escenarios donde algunos periféricos aún deban funcionar.

(RTC, UART y Timer etc.). La premisa aquí es que la velocidad del reloj principal se reduce o se detiene completamente y que cualquier operación se basa en un reloj de baja velocidad tales como 32 KHz u oscilador interno.

Obviamente el óptimo es donde el sistema está completamente parado y todas las fuentes del reloj detenidas y el MCU está alimentado justo o por encima de la tensión mínima de alimentación, donde el

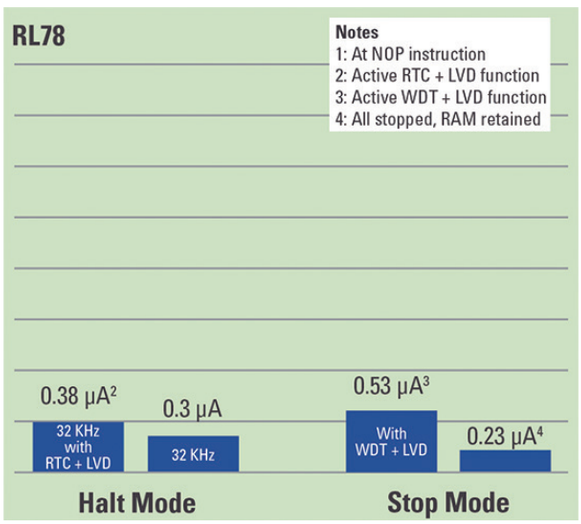

Figura 1. RL78, Consumo de corriente en modo de espera.

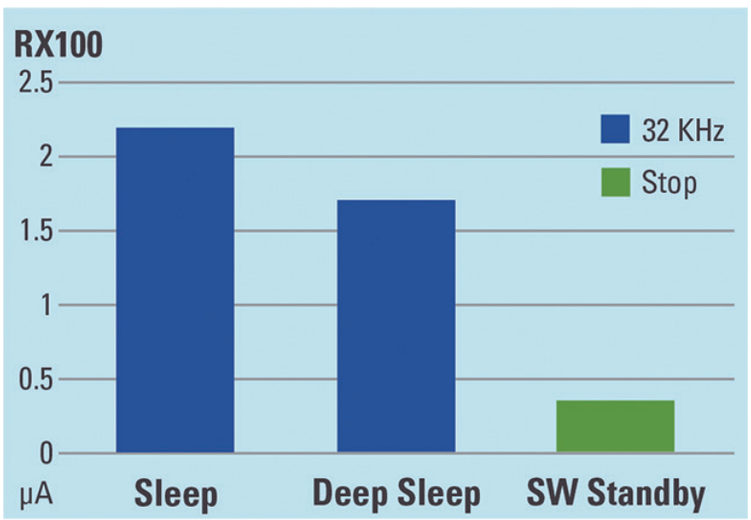

Figura 2. RX100, Consumo de corriente en modo de espera

MCU se mantiene en un estado estático.

En las figuras 1 y 2 se muestran ejemplos de nuestros productos destacados, que muestran la reducción en consumo de energía. Aquí los modos HALT/SLEEP están usando una baja velocidad de reloj de 32 KHz y STOP/ SW Standby, con todos los relojes parados.

Como se indica en el documento 1 los dos modos de bajo consumo del RL78, suspendido (HALT) y parado (STOP) (Mire la figura 1) ofrecen una significativa reducción de energía en cualquier caso, y la función SNOOZE (vinculado al modo de parada) disminuye el número de veces que la CPU necesita ser despertada hasta que se produzca una condición válida de despertar, reduce aún más la corriente media del modo de parada.

Por ejemplo analizando un regulador de calefacción basado en el RL78 en modo de "SNOOZE", que mide la temperatura (conversión ADC), comprueba el resultado y reporta al controlador de unidad HVAC, solamente si la temperatura está fuera del intervalo (ver figura 3).

Mediante el sólo uso del modo "STOP" puede lograrse una corriente promedio de 880 nA, pero mediante operación de "SNOOZE", la corriente media se reduce a 680 nA, una reducción del 25% que puede ser traducido directamente en que la batería dure un 25% más.

Para la familia de MCU RX100 los tres modos de SLEEP, DEEP SLEEP y SOFTWARE STANDBY, tal como se muestra en la figura 2 y se describe en documento 1, se entiende que no son tan bajos como el RL78 pero todavía ofrecen un ahorro de energía importante para una familia MCU altamente integrada y de alto rendimiento.

No debemos ignorar un último factor de las funciones analógicas externas o periféricos digitales, por lo que durante la espera, es importante que estos puedan "desactivarse" ya que algunas funciones analógicas pueden consumir tanta energía como el MCU.

Creo que está claro que mediante el empleo de modos de standby y apagado total del sistema como parte del diseño, se pueden conseguir corrientes promedio muy bajas, maximizando así la duración de la batería.

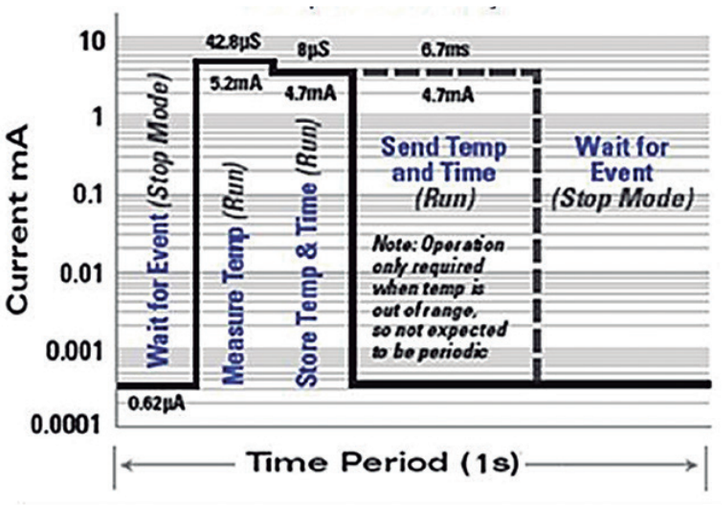

Figura 3. RL78 average consumption example

Velocidad de operación del Reloj del MCU

Una manera simple y tal vez evidente para reducir el consumo de energía que no debe pasarse por alto es hacer funcionar el MCU y cualquier periférico externo a una frecuencia de reloj más baja, permanentemente o durante los períodos de inactividad.

Reduciendo el reloj del sistema desde el máximo disponible, reducirá la corriente de funcionamiento con resultados que dependen del producto. Por ejemplo, el reloj de sistema de alta velocidad de oscilación interno del RL78 se puede reducir en un número de pasos desde 32 MHz hasta 1 MHz, con cada reducción en la frecuencia se puede disminuir la corriente de funcionamiento en un 22%. Con lo que seleccionando una menor frecuencia de operación se puede tener un impacto en el cálculo de energía. Durante periodos, cuando el sistema no tiene que realizar ningún proceso principal, muchos MCU permiten que el reloj de la CPU cambie por software a menores frecuencias principales o incluso a baja velocidad de reloj (interna o externa).

Utilizando el RL78/L12, como se explicó en el documento 1 demostración del limón) como ejemplo, a velocidad total de operación usando el oscilador interno se consume alrededor de 3 mA (24 MHz @ 3 V), mientras que a 1 MHz el consumo de corriente es reducido en al menos un 30% y menos de 1 mA y a 32 KHz baja a 3.6 μA.

Hay un escenario similar para el RX111, que a velocidad total, el consumo de corriente es de 10 mA (32 MHz @ 3.6 V) mientras que a 1 MHz el consumo se reduce en un 90% hasta 1 mA y a 32 KHz es de 11.5 μA. Con lo que la misma instantánea puede verse en nuestras dos familias de microcontroladores.

Operando con un menor reloj de sistema, es por supuesto algo que afecta al proceso en primer plano, tardando más e impactando en la corriente media, por lo que un análisis de la corriente media a diferentes velocidades de reloj es recomendable.

El consumo medio de corriente se puede calcular en general con la siguiente ecuación:

$$I_{ave} = ((AC1 * AT1) + (AC2 * AT2) + (IC * IT)) / P$$

Donde:

- Active Current 1 (AC1). Consumo de corriente durante el 1er período de actividad.
- Active Current 2 (AC2). Consumo de corriente durante el 2º período de actividad**
- Active Time 1 (AT1). Tiempo necesario para el 1er periodo de actividad.
- Active Time 2 (AT2). Tiempo necesario para el 2º periodo de actividad.

- Inactive Current (IC). Corriente inactiva. El tiempo en inactividad/espera.
- Inactive Time (IT). Periodo total de tiempo inactivo – Total de veces activo.
- Time Period (P). Tiempo total del periodo de repetición.

** Si sólo existe un tiempo activo, entonces simplemente quitar esto de la ecuación y si existen tiempos activos adicionales entonces agregar lo necesario para la ecuación.

Un ejemplo del cálculo anterior es en el que se basa la operación que se muestra en la figura 3.

Iave = ((AC1 * AT1) + (AC2 * AT2) + (IC * IT)) / P

Iave = ((5.2 mA * 42.8 μs) + (4.7mA * 8 μs) + (0.56 μA * 949.2 μs)) / 1
Iave = ((222.56 nA) + (37.6 nA) + (53.2 nA)) / 1

Iave = 313.36 nA

Nota: El evento de enviar temperatura y tiempo no ocurre muy a menudo y no ha sido incluido en los cálculos anteriores.

Cuando este evento ocurre además se utilizan 31.5 μA (4.7 mA*6.7 ms) durante este tiempo.
Si el período de operación ocurre cada 5 minutos (5000 segundos) la corriente media de este evento adicional sería de 6.9 nA (31.5 mA/5000).
Mientras algunos MCU's permiten cambiar el reloj principal por Software, hay que recordar que cambiar el reloj principal dinámicamente puede afectar al funcionamiento de los periféricos, por lo que debe considerarse cuidadosamente para asegurar que los dispositivos externos no se ven afectados y si se emplea durante el tiempo de espera se pueda restaurar la configuración de la operación normal.
Mientras el MCU funciona a frecuencias bajas, esto tiene un marcado efecto sobre la reducción de la potencia utilizada, el método principal de reducción de energía media del sistema es utilizarlo dinámicamente o usar menores frecuencias en conjunto con los modos de espera como se ha comentado anteriormente.

Interna	Externa
3.3 mA @ 24 MHz (3 V)	3.0mA @ 20 MHz (3 V)

Tabla 1. RL78/L12 Comparación de consumo de energía de reloj.

Interna	Externa
10.5 mA @ 32 MHz (3 V)	10.6 mA @ 32 MHz (3 V+PLL)

Tabla 2. Rx111 Comparación de consumo de energía de reloj.

Selección de la fuente de Reloj del MCU

La selección del tipo de reloj principal y secundario (si se usa) se refiere a no sólo establecer la velocidad de reloj como hemos comentado anteriormente, sino también al tipo de la fuente de reloj, ya que tal elección, afectará al consumo de corriente.
Ya es común para los fabricantes de MCU proporcionar osciladores de reloj interno, por lo que en muchas aplicaciones la necesidad de un cristal/resonador externo es eliminado, no sólo por el ahorro energía, sino también por el coste.
En los osciladores internos generalmente se distinguen dos categorías, el reloj de sistema de alta velocidad (ej. reloj principal) y el reloj de subsistema de baja velocidad.
Los osciladores principales de sistema internos se han vuelto suficientemente precisos para muchas aplicaciones (≤1%), y pueden incluir una función de "ajuste" permitiendo perfilar y ajustar el oscilador, para mantener la precisión en el rango de temperatura. Los relojes internos de baja velocidad, generalmente no son tan exactos por lo que todavía puede ser necesario utilizar un cristal de 32 KHz o resonador externo, por ejemplo donde se requiera un RTC para registrar tiempo. Si bien es preferible utilizar osciladores internos especialmente para la fuente de reloj del sistema principal, todavía puede haber ocasiones que se requiera una frecuencia específica o donde sea necesaria una mayor precisión para manejar una interfaz USB o Ethernet. Aquí hay pocas opciones de elegir otra que no sea la del cristal de menor consumo o el posible resonador compatible con el fabricante del MCU. Recuerde que el procesador y otros periféricos no tienen porque funcionar a esta frecuencia y pueden además funcionar más despacio.
Una simple comparación entre las diferentes corrientes requeridas por la fuente de reloj del sistema principal mediante el uso de la fuente de reloj interno o externo basados en nuestros productos ejemplo, compara la frecuencia de reloj máxima disponible y asumiendo el funcionamiento normal.
El máximo cristal externo de los RL78 es 20 MHz con lo que racionalizando así ambas fuentes de reloj, el oscilador interno utiliza 137.5 μA/MHz mientras que el externo usa 150 μA/MHz. Así que para comparar, si los dos relojes funcionaran a 16 MHz, entonces el interno consumiría 2.2 mA y el cristal externo 2.4 mA. Pueda parecer que 200 μA no es una gran diferencia, pero todo suma cuando se busca hacer que la batería dure más.
Analizando el mismo escenario para el RX111 (tabla 2), la diferencia es de alrededor de 100 μA. Otra vez no es una gran diferencia, pero todavía son 100 μA que no se consumen de la batería. Un punto a destacar es que usando un cristal externo con el RX111 es necesario utilizar el PLL al considerar velocidades de reloj de sistema por encima de 20 MHz.
Al utilizar osciladores principales internos se ahorra corriente pero otro factor importante es que los osciladores internos tienen tiempos de arranque más rápidos que los cristales o los resonadores, lo que significa que menos tiempo (y energía) se pierde durante el des-

pertar de la espera, donde el reloj se ha detenido. Normalmente los tiempos de despertar de los cristales/resonadores son de alrededor de 2 ms, mientras que el oscilador interno de alta velocidad tiene un despertar de sólo 40 μs (RX111), 50 veces más rápido y ahorrando 50 veces la potencia de arranque utilizada.

Relojes de baja velocidad o "sub" relojes también pueden requerir el uso de un cristal/resonador externo, si el reloj interno es de frecuencia incorrecta o insuficientemente exacto para la aplicación.

La elección del cristal/resonador puede marcar una gran diferencia, especialmente donde está disponible una variedad de opciones de energía en la célula del oscilador.

Es importante comprobar la compatibilidad de un resonador de cristal con el fabricante MCU para asegurar el funcionamiento correcto, especialmente cuando se usa un oscilador en modo de ultra baja potencia (ULP). Por ejemplo la comparación entre el RL78 normal y el "ultra" baja potencia con subclock es que el consumo actual de "ULP" es menos de la mitad de la configuración del oscilador normal (Normal = 380 nA y ULP = 180 nA). Si bien esto no parece mucho, si tenemos en cuenta que las corrientes en espera pueden ser del orden de 560 nA (dependiendo de la configuración y de lo que está en funcionamiento), entonces 200 nA es un 36%, así que se puede ahorrar un 36% más de corriente.

Operación Periférica del MCU

La operación de la periferia es otra área considerada como "obvia" en la que cualquier periférico no utilizado debería estar apagado para evitar la perdida de energía. Es normal decir que está "apagado" ya que es generalmente el estado por defecto del periférico, pero merece la pena revisar la hoja de datos para asegurarse de ello.

Es importante que para lograr muy bajas corrientes de espera donde está funcionando el reloj secundario, cualquier periférico no requerido, en el estado inactivo, debe suspenderse, a menos que este configurado para operar desde este sub-reloj (15 KHz o 32 KHz) sino, no funcionará correctamente y consumirá energía.

Se necesita un proceso de software adicional para entrar y salir del tiempo de espera, por lo que hay que plantear cuidadosamente como deshabilitar algunos periféricos ya que pueden restablecer sus opciones de configuración.

Esto requiere más tiempo de proceso y consume más energía de ejecución, con lo que es posible que no se apaguen todos los periféricos en el periodo de reposo. Esto debe analizarse durante la fase de diseño para determinar la mejor opción. Obviamente si se paran todos los relojes, entonces no es un problema, ya que tanto el MCU como los periféricos serán forzados a detenerse en un estado estático manteniendo el estado actual de los registros.

Algunos periféricos avanzados como USB pueden requerir un reloj externo independiente que puede continuar funcionando durante los modos de espera. No es lo ideal, por lo que debe consultarse el manual del usuario para ver si se puede detener.

También se debe tener cuidado con el "watchdog" si se está usando como función de seguridad, ya que estos aún funcionan durante los modos de espera. Sin el acceso de "servicio requerido", el watchdog se puede desbordar y causar una interrupción o hardware reset, por lo que debe hacerse una cuidadosa revisión del manual del usuario de la operación del watchdog en modo de espera.

El uso de los modos de espera y parada pueden generalmente configurarse para que el watchdog se atienda a la entrada y a la salida desde el modo de baja energía, evitando así un reinicio precipitado. Es difícil cuantificar el efecto exacto sobre consumo de corriente ya que todas las aplicaciones son diferentes en funcionamiento y espera, aunque cualquier función "analógica" (ADC, DAC, LCD, sensor de temperatura etc.) consumirá más energía que la mayoría de las funciones digitales, pues incluyen drenajes de energía estática (resistencias, referencias etc.) que son independientes de la velocidad de reloj. Para el óptimo bajo consumo, la técnica sugerida es encenderlos, usarlos de manera rápida y apagarlos. Sin embargo debido al tiempo de estabilización de muchos periféricos analógicos, es prudente determinar el mejor momento de habilitarlos y desactivarlos, para asegurarse de que está listo cuando se necesita y si es posible, utilizar el tiempo de estabilización para realizar otras tareas de la CPU. Finalmente hemos considerado periféricos "internos", pero a veces será necesario utilizar funciones externas especialmente dispositivos analógicos tales como sensores. Se recomienda que todos los periféricos externos también puedan ser "apagados" o desactivados cuando no estén en uso.

Uso de Pines I/O del MCU

Esta sección recuerda pequeñas cosas, para evitar el consumo innecesario e inesperado.

Intente evitar bajas unidades de impedancia, ya que esto aumentará las corrientes de conmutación en el controlador de salida.

Asegúrese de que no hay pines flotantes ya que pueden aumentar las corrientes de fuga en el diseño y pueden establecer niveles indeterminados que activen un estado válido de entrada en un pin de entrada.

Deben evitarse resistencias de "Pull up o pull down" a menos que sean absolutamente necesarias. Si usa una resistencia externa, intente establecer el valor lo suficientemente alto para reducir la corriente durante la operación, pero lo suficientemente bajo para asegurar el funcionamiento correcto de la operación.

Por ejemplo una resistencia de pull up interna puede tener un valor tan alto como 100 KΩ, con lo que cualquiera externa puede tener también este valor, mientras que para un drenaje abierto (típicamente usado para la interfaz I2C) tendrá que ser considerablemente menor (tan bajas como 1 KΩ) para mantener correctamente los tiempos de subida y bajada especificadas.

La mayoría de los MCU incluyen resistencias internas de pull up programables en los pines, con lo que se puede usar en caso necesario. Durante el modo de espera hay que probar y asegurar que ningún pin se pone al nivel de pull up/down (es decir, lógica 0 para pull down y lógica 1 para pull up) normalmente en estado inactivo.

Para evitar el uso de resistencias de pull up/down en cualquier pin I/O no usado, puede configurarse como "salida", ya que siempre se establece un nivel definido y no requiere de resistencias de pull up/down.

Un beneficio colateral que ofrece la baja impedancia hacia el mundo exterior es que ayuda con la inmunidad al ruido.

Integración del Sistema

Los interfaces para periféricos externos son uno de los factores a considerar en los sistemas grandes, ya que todos estos requieren de control y señales de datos, por lo que todos estos pines de I/O consumirán una corriente de conmutación definida por la ecuación:

½ C * V2 * f (Carga capacitiva * Conmutacion Voltage2 * Conmutacion de frecuencia).

Por integración nos referimos a una familia MCU que tiene algunos o todos los periféricos integrados en el chip, incluyendo funciones como USB, Ethernet y E2ROM (utilizando datos en Flash) más un número creciente de funciones analógicas tales como, sensor de temperatura analógico, comparador y amplificador de ganancia programable etc.

Los beneficios de proporcionar periféricos en un chip aparte del ahorro de costes, es que mientras el periférico todavía consume energía (cuando se usa) las interfaces de datos e interfaces de control usan corrientes de conmutación mucho más bajas (menor voltaje (V2) y menor capacidades (C)) con corrientes no muy altas en los pines I/O usados y las funciones pueden apagarse fácilmente cuando no son necesarias.

Opciones de suministro de energía

La elección de la fuente de alimentación parece ser una elección sencilla, "cuanto más bajo el voltaje más baja la potencia" y hasta cierto punto esto es cierto, pero no necesariamente como usted podría pensar. Muchos MCU actuales incluyen un regulador interno que bajan el voltaje de funcionamiento interno y ofrecen un consumo de energía constante en la mayor parte del rango de operación del voltaje, haciendo funcionar al dispositivo en 3 V o 5 V (en algunos casos hasta 1.8 V) sin ninguna diferencia en el consumo del MCU.

Obviamente hacer funcionar el sistema a 3 V puede marcar la diferencia en la potencia total utilizada, pero se trata de una opción del diseñador y de las interfaces fuera del MCU.

Sin embargo, muchos dispositivos ahora operan por debajo de voltajes muy bajos, por ejemplo el RL78/L12 puede operar a 1.6 V y el RX111 a 1.8 V, haciendo posible la operación a bajo voltaje. Con esta amplia gama de posibles voltajes, se extiende la duración de la batería antes de que sea necesario cargarla o reemplazarla. Sólo hay que tener en cuenta que la frecuencia máxima del reloj es normalmente reducida cuando se opera a voltajes muy bajos, así que tenga cuidado cuando ajuste la frecuencia de reloj en un rango de alimentación de 3 V hasta 1.8 V para asegurar que la frecuencia del reloj principal es adecuada para el rango de tensión usado.

La frecuencia del oscilador interno de alta velocidad del RL78/L12 a 1.6 V puede llegar hasta 4 MHz, la frecuencia del oscilador interno máximo del RX111 de 1.8 V es de 8 MHz.

Es posible reducir el suministro de energía durante tiempos de espera donde la tensión mínima se utiliza para mantener la configuración de los registros y el contenido de la RAM. Sin embargo esto puede ser un procedimiento complicado y se debe tener cuidado en la secuencia de "disminución de consumo" y "encendido de consumo" del sistema, para que no afecte a ningún periférico externo conectado al MCU. Lo más probable es que esto también afecte al tiempo de "apagado" y "despertar" del sistema.

Conclusión

El tema de este documento ha sido buscar los "trucos" que ayudan a reducir el consumo de energía. Mientras que las secciones anteriores proporcionan una visión general donde cada aplicación puede tener diversos requisitos, es recomendable usar un "Perfil de uso de energía" en el diseño, para analizar el consumo actual y resaltar las áreas que no cumplen sus objetivos de diseño o donde hay consumo inesperado y permitan la implementación analizada y ajustarla a la vida de la batería usada.

Para más información se recomienda leer los otros documentos que se destacan en esta serie y visitar el centro de diseño de recursos de Renesas.

Documento 1: Diseño con la fuerza de un limón

Un ejemplo de lo que puede lograrse con el producto adecuado y los modos de operación.

Documento 3: Reducir el reloj frente al modo de espera del MCU, para diseños de bajo consumo.

Reducir la velocidad del reloj del MCU durante los tiempos de funcionamiento e inactividad y los efectos de combinarlos con tiempos de espera.

Documento 4: Maximiza la duración de tu batería.

Análisis de sistemas que están diseñados para funcionar en largos períodos de tiempo de espera.