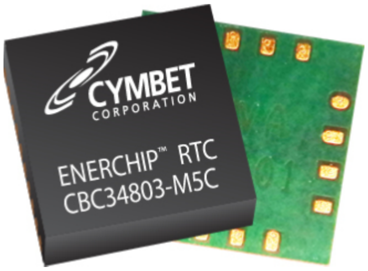
AN-1059

Application Note

Extend Battery Life by Reducing System Power using the EnerChip RTC



Introduction

This Application Note discusses techniques to use the EnerChip RTC CBC34803 or CBC34813 self-powered Real-Time Clocks (RTC) to dramatically reduce system power in microcontroller based applications. This in turn extends battery life significantly. The internal battery in the EnerChip RTC also serves to backup the time in case of a main battery swap or main battery complete discharge. Now batteries can last the life of the product over many years.

Power Reduction Techniques

Typical microcontroller based devices can benefit from the power saving features in many microcontrollers. These features include ways to shut down portions of the chip such as serial ports and other peripherals when they are not being used, changing clock frequencies on the fly, or going into a variety of “sleep” modes that can drastically reduce power by powering down subsystems not currently in use. The most common and generally most effective method is to put the microcontroller and peripherals into sleep with only a wake-up timer running. This mode can often reduce the current of the microcontroller to a few microamps or less of current. Even so, the sleep current integrated over a long time can be a significant power drain. Some microcontrollers can stop all timers and operations but keep a few registers alive to cut the current to only several tens of nanoamps but they don’t have a built-in way to wake up since the internal timers are also suspended. The microcontroller power reduction through sleep is a big benefit but often the sensors or user controls must stay awake which increases the power usage.

This Application Note suggests methods of using an CBC348xx to utilize the lowest power suspend modes of microcontrollers and/or to completely shut power down to peripheral chips for periods of time to **drastically reduce power in systems that need to respond to human-speed time delays.** These techniques can also be used to minimize power in slower, environmental sensor applications. The general techniques and results will be discussed first followed by considerations for systems with higher active power. Lastly, specific registers that are used to set up the CBC348xx to accomplish such power savings are listed. System power savings of factors of ten or even one hundred are possible.

Sleep Power versus Active Power

Typical microcontroller-based devices where energy conservation is an issue spend a large part of their time in a low-power “sleep” state. This is a state where the microcontroller is not running but is waiting for an interrupt from either a sensor or a timer. When one of these interrupts is issued the microcontroller goes to a higher power active mode to process the event and then goes back to sleep. The active mode operation may include processing the sensor data and then may operate an actuator or send a message. Many times a message may be sent via a low-power radio protocol which requires significant processing cycles to correctly operate the protocol stack. The amount of processing depends greatly on the complexity of the protocol.

The average power consumption of the system is the sleep power times the percentage of time the system is asleep plus the active power times the percentage of time the system is active divided by 100.

$$P_{avg} = (P_{sleep} * \%\ \text{time asleep} + P_{active} * \%\ \text{time active})/100$$

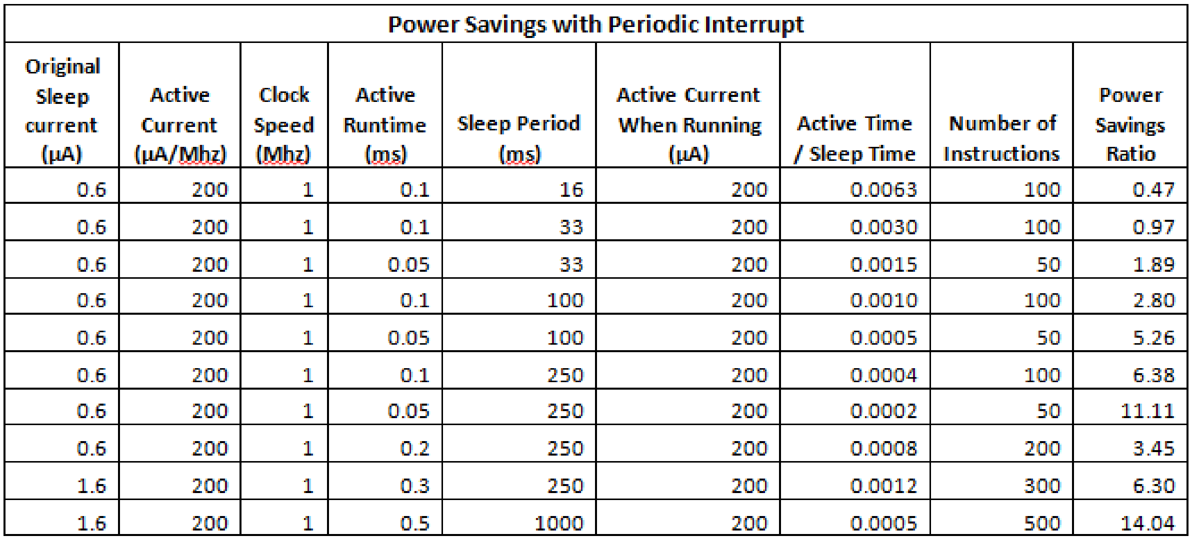
Minimizing the largest of these terms provides the greatest power savings. In some cases the active power term is much larger than the sleep power term either because the power per event is large or the active power events happen very often.

Average Power Consumption when Mostly in Sleep State

When the system that has a large sleep power compared to its active power is asleep, there is an opportunity to reduce power by placing the microcontroller in its lowest power mode while using the CBC348xx RTC timer functions to provide periodic wake-ups to the microcontroller and associated circuitry. This way the entire system is totally asleep for the majority of time and the microcontroller is only awakened for short periods to determine if it needs to service a sensor or switch. The CBC348xx is configured to automatically wake the system at regular periods for a finite time and then the microcontroller goes back to sleep. The CBC348xx can also be configured to completely turn off power to the sensors and/or microcontroller by using its internal 1Ω pull-down switch. If the microcontroller determines during one of its waking intervals that it needs to service something then it quickly commands the CBC348xx to not automatically shut it down until further commanded. This technique can reduce the system power greatly since the CBC348xx only requires 36nA of current to manage the timing functions and the rest of the system can go to its lowest power mode.

The average system power is a function of the time the system needs to run compared to the time it is asleep. If the microcontroller/system is awakened too often the power savings will be minimal. A simple metric of the possible savings is to add the sleep current (since it is always present) to the active current times the ratio of the active time divided by the sleep time. Table 1 below shows some examples of different power savings that can be achieved with different sleep vs. run times. Column one is the Original Sleep Current of the system without using the CBC348xx. The Original Sleep Current includes the sleep power of the microcontroller with a timer running plus any sensor current. The Power Savings Ratio is the Active Current times the ratio of active/sleep times plus the 36nA CBC348xx current compared to the Original Sleep Current in column one. The Number of Instructions column shows how many instructions the microcontroller can execute in the period of time listed in the Active Runtime column. For sake of reference it takes about 28 I2C clocks at 400kHz or about 70µs to write to a single register in the CBC348xx. Make sure to write to the register to disable the timer before the CBC348xx automatically switches the microcontroller/system power off.

Table 1: Combining Sleep Power and Active Power to Compare Power Savings



Power Savings with Periodic Interrupt

Original Sleep current (µA)	Active Current (µA/Mhz)	Clock Speed (Mhz)	Active Runtime (ms)	Sleep Period (ms)	Active Current When Running (µA)	Active Time / Sleep Time	Number of Instructions	Power Savings Ratio
0.6	200	1	0.1	16	200	0.0063	100	0.47
0.6	200	1	0.1	33	200	0.0030	100	0.97
0.6	200	1	0.05	33	200	0.0015	50	1.89
0.6	200	1	0.1	100	200	0.0010	100	2.80
0.6	200	1	0.05	100	200	0.0005	50	5.26
0.6	200	1	0.1	250	200	0.0004	100	6.38
0.6	200	1	0.05	250	200	0.0002	50	11.11
0.6	200	1	0.2	250	200	0.0008	200	3.45
1.6	200	1	0.3	250	200	0.0012	300	6.30
1.6	200	1	0.5	1000	200	0.0005	500	14.04

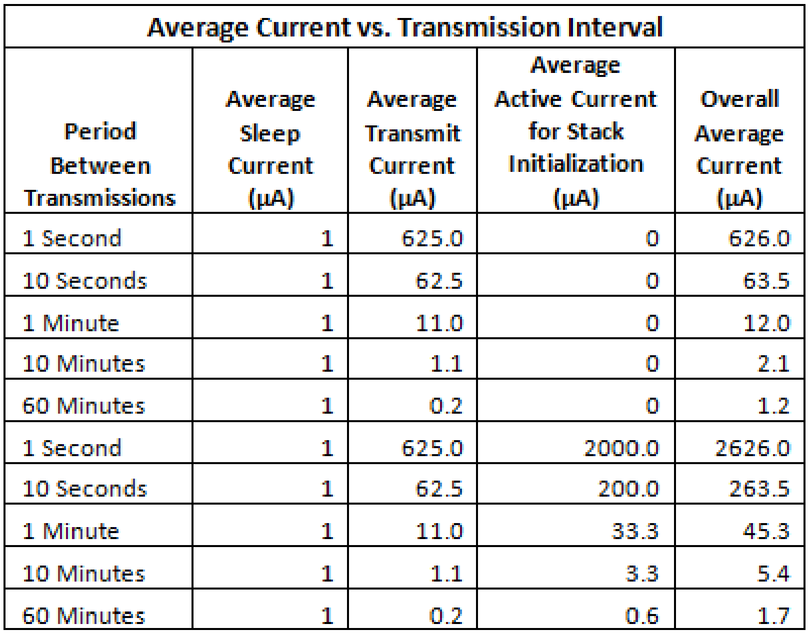
Notice that the Power Savings Ratio is only a benefit if it is over 1.0. This table shows that the longer sleep periods have the best ratios. With higher Original Sleep Currents the benefits are also magnified. The next to the last line shows a Power Savings Ratio of 6.30 with over 300 instructions executed per wake-up. The system in this example had an Original Sleep Current of 1.6µA for the microcontroller current with internal sleep timer and an external sensor. **A 6.30 Power Savings Ratio means 6.3 times more battery life in a battery powered system.** The last line shows a one-second sleep time associated with a slower, environmental sensor. Notice the large 14.04 Power Savings Ratio in this example. These examples show that a long battery life extension can easily be achieved using this technique.

Radio Power and Software Stack Concerns

The power savings technique described in the previous section works well with systems where the sleep power is a large or significant contributor to the average power. In systems where a radio is used, the power for the radio and the power used by the microcontroller to run the software stack can be large. Many radio standards are complicated and use a sophisticated stack to manage the protocol. These stack implementations often have significant runtimes to initialize their internal memory structures. Often these stacks are supplied by the radio chip vendor and therefore are attractive since they save software development time. If the stack needs a long time to initialize, then the previous power savings technique only makes sense if the radio operates very infrequently.

A typical radio application requires 25mA for 25ms, or 625µA-seconds of charge, for each transmission. A simple stack may take essentially no time to initialize but a sophisticated one may take upwards of 400ms of processing time at 5mA, resulting in a drain of 2000µA-seconds to initialize. Table 2 shows the effect of the radio current and the stack initialization current for this example at various reporting intervals.

Table 2: Effect of Active Power with Various Reporting Intervals.



Average Current vs. Transmission Interval				
Period Between Transmissions	Average Sleep Current (µA)	Average Transmit Current (µA)	Average Active Current for Stack Initialization (µA)	Overall Average Current (µA)
1 Second	1	625.0	0	626.0
10 Seconds	1	62.5	0	63.5
1 Minute	1	11.0	0	12.0
10 Minutes	1	1.1	0	2.1
60 Minutes	1	0.2	0	1.2
1 Second	1	625.0	2000.0	2626.0
10 Seconds	1	62.5	200.0	263.5
1 Minute	1	11.0	33.3	45.3
10 Minutes	1	1.1	3.3	5.4
60 Minutes	1	0.2	0.6	1.7

Design Technique: Notice the effects on overall average current both with transmission period and with stack initialization. This highlights the importance of implementation: use a simple stack initialization, or a microcontroller that has a low power mode to retain the initialization in RAM at low power, or both.

Example System Configurations

The system can be configured to cut power to the sensors and microcontroller as shown in Figure 1 and Figure 2. The additional FET in Figure 2 allows the VCC bus to be switched instead of VSS. The PSW/nIRQ2 output includes a selectable 1Ω FET switch that greatly reduces drops in the switched bus.

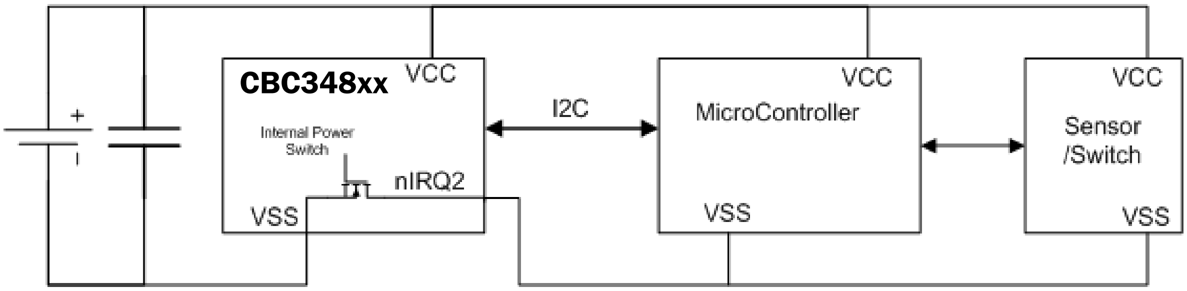


Figure 1: Switched Ground Configuration.

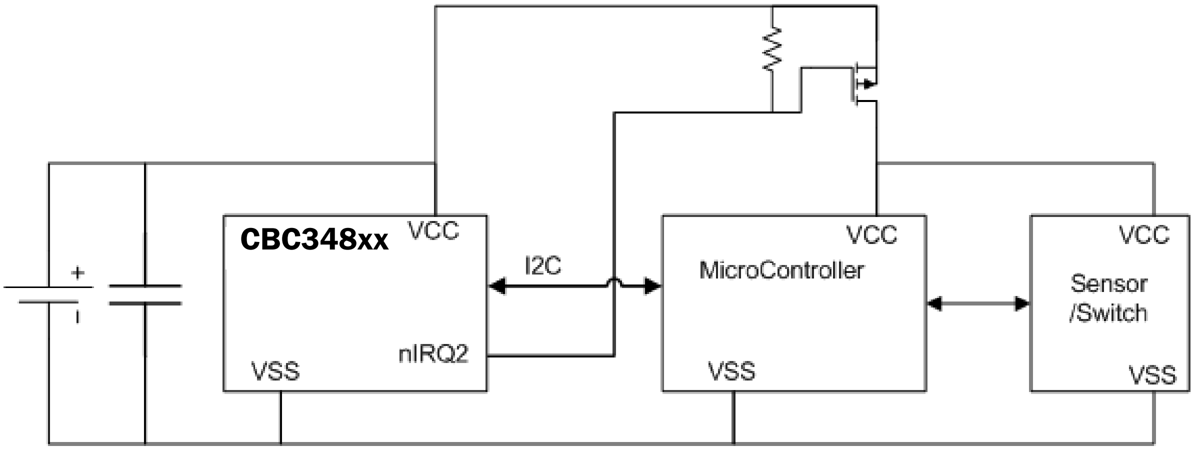


Figure 2: Switched VCC Configuration.

It is also possible to not switch the power bus, but instead put the microcontroller in deep sleep, and wake it periodically with an interrupt as shown in Figure 3. This works well with microcontrollers that have higher internal sleep timer currents but very low current when clocks are disabled and only some memory is retained.

Design Technique: Firmware must be designed to avoid large memory re-initialization routines on startup either from reset or from the interrupt.

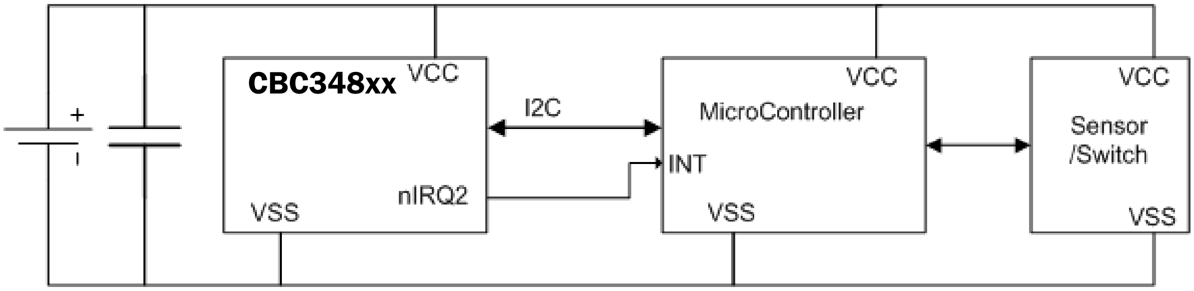


Figure 3: Interrupt-Only Configuration.

Important CBC348xx Registers to Consider

Control1 Register: PWR2 -> When 1, the PSW/nIRQ register is driven by an approximately 1Ω pull-down. Set this to control power to the system.

Control1 Register: OUTB -> Set to inactive level (1) of PSW/nIRQ2 pin.

Control2 Register: OUT2S -> Set this field to 0x100 to send timer interrupt to PSW/nIRQ2 pin.

Countdown Timer Control Register -> Configures timer operation. TE turns timer on/off. TM, TRPT = 1.

Countdown Timer Register -> Set to decimal 32 to divide the 128Hz RC clock down to 4Hz.

Timer Initial Value Register -> Set to decimal 32 as above for auto-reload of 4Hz period.

Oscillator Control Register: OSEL-> Set to 1 to use 128Hz RC oscillator for timers.

Oscillator Status Register: LKO2 -> Set to 0 to unlock OSEL.

Pseudo-Code Examples

Please contact the Cymbet Applications group for code examples.

Ordering Information

EnerChip RTC Part Number	Description	Notes
CBC34803-M5C	EnerChip RTC in 5mm x 5mm x 1.4mm 16-QFN Land Grid Array	Shipped in Tube
CBC34803-M5C-TR1 CBC34803-M5C-TR5	EnerChip RTC in 5mm x 5mm x 1.4mm 16-QFN Land Grid Array	Tape-and-Reel - 1000 pcs (TR1) or 5000 pcs (TR5) per reel
CBC34813-M5C	EnerChip RTC in 5mm x 5mm x 1.4mm 16-QFN Land Grid Array	Shipped in Tube
CBC34813-M5C-TR1 CBC34813-M5C-TR5	EnerChip RTC in 5mm x 5mm x 1.4mm 16-QFN Land Grid Array	Tape-and-Reel - 1000 pcs (TR1) or 5000 pcs (TR5) per reel
CBC-EVAL-12-34803	EnerChip RTC Evaluation Kit	USB based Eval Kit with CBC34803 tab board
CBC-EVAL-12-34813	EnerChip RTC Evaluation Kit	USB based Eval Kit with CBC34813 tab board

U.S. Patent No. 8,144,508. Additional U.S. and Foreign Patents Pending.

Disclaimer of Warranties; As Is

The information provided in this data sheet is provided “As Is” and Cymbet Corporation disclaims all representations or warranties of any kind, express or implied, relating to this data sheet and the Cymbet EnerChip product described herein, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, non-infringement, title, or any warranties arising out of course of dealing, course of performance, or usage of trade. Cymbet EnerChip products are not authorized for use in life critical applications. Users shall confirm suitability of the Cymbet EnerChip product in any products or applications in which the Cymbet EnerChip product is adopted for use and are solely responsible for all legal, regulatory, and safety-related requirements concerning their products and applications and any use of the Cymbet EnerChip product described herein in any such product or applications.