Please note that Cypress is an Infineon Technologies Company.

The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content

The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers

Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

www.infineon.com

THIS SPEC IS OBSOLETE

Spec No: 001-15084

Spec Title: CALCULATING BATTERY LIFE IN WIRELESSUSB LS(TM) SYSTEMS - AN4002

Replaced by: NONE

AN4002

Calculating Battery Life in WirelessUSB LS™ Systems

Author: Sai Prashanth Chinnapalli

Associated Project: No
Associated Part Family: CYWUSB6934, CYWUSB6935
Software Version: NA
Related Application Notes: None

To get the latest version of this application note, or the associated project file, please visit http://www.cypress.com/go/AN4002.

A key decision when evaluating WirelessUSB LS™ for use in a design is determining the battery life that is achievable. This application note describes how to calculate the expected battery life in two-way WirelessUSB products.

OBSOLETE

Contents

Introduction

When you evaluate WirelessUSB LS™ for use in a project, product or design, one of the key decision factors will be what battery life will be achievable by the design. This may impact decisions such as what type of batteries to use, and how many cells will be required to power the design.

In other projects, the number and type of batteries may be determined by outside factors. In this case, you must estimate the battery life that is achievable using WirelessUSB, and verify that it meets product requirements.

If you are faced with this task, first refer to the application note Managing Power in WirelessUSB™ LR Systems to gain an overview of the power-related issues in designing WirelessUSB products.

This application note provides guidelines for calculating expected battery life in 2-way WirelessUSB products for engineers who are familiar with these basic power management concepts. Three common WirelessUSB applications are then examined in depth by analyzing three case study examples.

Choosing Cell Type

Although alkaline cells are usually the easiest to use in WirelessUSB applications, all cells involve a trade-off between size, cost, circuit, complexity/cost, and battery life.

OBSOLETE

NiCd and NiMH batteries are not considered in detail in the application note. However, as these battery types are available in AA and AAA form factors, you must consider the impact of an end user attempting to make use of these cells. Although the output voltage of these cells is similar to that of Alkaline cells toward the end of the life of the battery, the initial voltage is somewhat lower– approximately 1.2 V rather than 1.5 V. In some cases, this may have significant implications which you should bear in mind when making battery and power circuit selection decisions.

Coin and button cells, both primary and rechargeable, typically have low maximum output currents, below the peak current requirements of WirelessUSB devices (hereafter referred to as *the Radio*). It is possible to use such batteries to power *the Radio*, but it is necessary to use a capacitor to provide part of the peak current, and this requires specific design techniques that are outside the scope of this application note.

WirelessUSB LS Power Requirements

Before considering battery life, it is essential to consider the current and voltage requirements of the application.

The Radio has a typical peak current when transmitting of 69mA, but the designer must consider the current consumed by other circuit components, including the Microcontroller (MCU), and sensor components.

This current requirement should not be a problem for Alkaline batteries, because they have a relatively low source impedance, and so even towards the end of life this will not cause significant droop.

The Radio has a supply voltage range of 2.7 V–3.6 V. *The Radio* should never be exposed to a voltage of more than 3.9 V, and operation is not guaranteed above the 3.6 V maximum. Cypress guarantees that all devices will operate correctly at 2.7 V, and typical devices will operate a little below this voltage, depending on operating temperature and device-to-device variation. The designer should carefully consider whether or not to detect when the supply voltage falls below 2.7 V, and to cease operating at that point, or whether to continue to operate *the Radio* until correct operation ceases.

Alkaline Cell Usage Basics

When using Alkaline cells, the designer must balance the required battery life with the number of cells and type of the power supply circuit to be used.

Figure 1 and Figure 2 show the discharge characteristic of the Energizer E91 AA and E92 AAA Alkaline cells. Good quality Alkaline cells from other manufacturers typically have similar properties. These cells have a total rated capacity of 2850 mAh and 1250 mAh, respectively. High-power Alkaline cells, with capacities of 3300 mAh or more are available, but will not be considered in this application note, other than to note that in applications where the designer has control over the exact cell being used, they may be an option for extended life.

Figure 1. Example Cell Discharge Characteristics

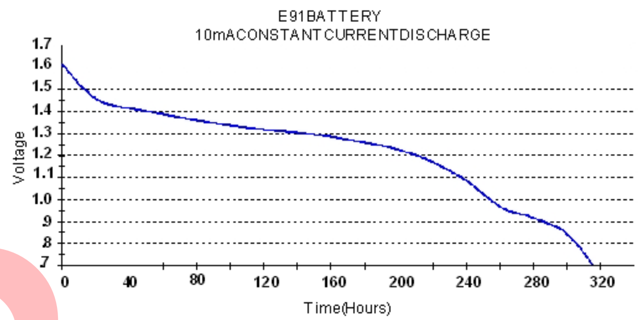
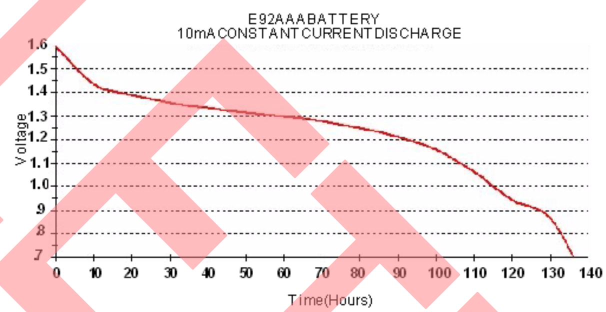


Figure 2. Example AAA Alkaline Cell Discharge



It can be seen from the figures that approximately 25 percent of battery capacity is available above 1.35 V, 75 percent is available above 1.2 V, and 85 percent of capacity is available above 1.1 V.

This leads to several possible power regulation options:

1. *The Radio* may be powered directly from two Alkaline cells. This is generally the lowest cost option, but will require batteries to be changed with most of the capacity unused.
2. *The Radio* may be powered from four or more Alkaline cells through a linear voltage regulator. Using a 3.0 V regulator will allow the full capacity of the battery to be used. A 3.3 V low dropout regulator (LDO) will still allow almost all the battery capacity to be utilized.

3. *The Radio* may be powered from three Alkaline cells through an LDO. Choice of regulation voltage will depend on a trade-off between cost and the amount of the battery capacity the designer is prepared to leave unused. A 2.85 V, 5 percent regulator with a 100-mV drop-out voltage will provide access to 90 percent of the batteries rated capacity. Use of a 3.3 V regulator with a 250-mV drop-out voltage may leave 25 percent of the battery capacity unavailable for use.
4. *The Radio* may be powered from one or two Alkaline cells using a DC-DC boost converter. This is the most costly option, as such converters typically cost $0.30 or more even in high volumes when including the cost of external components. However, this option allows the designer to make use of the entire battery capacity, and is well suited to space-constrained applications.

Calculating Battery Life

The first step in calculating an estimate of battery life is to calculate the average current drawn from the batteries. To calculate the average current, you must determine the following [1]:

- The average current during the transmission of a packet (including both transmit and receive segments of a packet)
- The length of a typical packet
- The frequency with which packets are typically transmitted
- The sleep current.

The Radio is only one component of the current consumption of the device, and in calculating battery life the designer will have to add the average current consumption of the rest of the circuit. However, the process is similar, and will yield an average current which can be added to the average radio current to estimate battery life of the system.

To determine the average current during the transmission of a packet, the designer needs to determine the following:

- Length of time spent in *Standby* mode
- Length of time spent in *Idle* mode
- Length of time spent in *Transmit* mode
- Length of time spent in *Receive* mode
- The current consumed in each mode.

The average packet transmission current can then be calculated by multiplying the time spent in each mode by the current drawn in each mode, and then dividing that sum by the total packet transmission time. This value then needs to be corrected for packet transmissions where a valid handshake is not received. In short range applications (up to ten meters), this will be much less than 1 percent of packets. However, when a device is designed to work close to the limits of range, or in a very high interference environment, this adjustment may be significant, and will require empirical data from tests of the device in the application environment.

The Managing Power in WirelessUSB™ LR Systems application note explains the mechanics of packet transmission and timing, so that topic will not be addressed in detail here. Some example calculations are shown in the case studies later in this document.

The average current resulting from packet transmission can then be calculated by multiplying the average transmission current by the transmission duty cycle. In some cases calculating duty cycle may be straightforward. One example is a wireless sensor which transmits four bytes every second. In other cases, this may require complex modeling of the usage of the device. Examples of such applications are covered in the case studies.

An alternative method is to determine the total charge required to transmit a packet, by multiplying the time spent in each mode by the current drawn in each mode. The resulting value, in Amp-seconds (As) is multiplied by the time spent transmitting during each second to provide the average current attributable to packet transmission.

In many cases, the resulting average current from data transmission will be very much greater than the sleep current, in which case *the Radio* sleep current can be safely ignored, as it will have negligible impact on the overall battery life.

Generally, in cases where the sleep current is significant, the designer should take care to take account of other small quiescent currents in the circuit. Capacitor leakage currents, battery self-discharge rates and voltage regulator quiescent currents will all typically be greater than the sleep current of *the Radio*.

Having obtained a value for average overall system current, the battery life can then be estimated. Factors affecting the accuracy of the estimate may include:

- Battery capacity varies with current drawn. Alternating between drawing 69 mA, and drawing a few microAmps will result in a somewhat lower battery capacity than constantly drawing the same average current. This effect can be mitigated by using a capacitor to provide some of the peak current during packet transmission.

[1] These values may be found in the data sheet. The values used in the calculations in this application note are approximate.

OBSOLETE

- Device to device variation in current drawn, not only in *the Radio*, but also the voltage regulator, MCU, and other circuit components may be significant.
- A usage model is just that – a model. The accuracy of the battery life estimate can never be better than the reliability of the model used.

In many cases it is advisable to calculate the battery life for several different battery options. The impact of quiescent currents, voltage regulator efficiencies and other factors are often greater or less than may be expected at first sight. In order to make the right choice of battery and power management architecture, it is recommended that several of the battery combinations discussed earlier are evaluated, so that the trade-off between battery life and cost or size can be made on the basis of the best data available, rather than assumptions.

The following three case studies all use the Radio in the same modes and use the same protocol:

- 2-way protocol
- 62.5-kbps DDR mode
- 1–1 system.

Case Study 1: Wireless Keyboard

As noted in the previous section, one of the most important factors influencing the accuracy of a battery life estimate is the usage model. In this case study, the following usage model will be considered [2].

- Four keystrokes per second, four hours per day, five days per week.
- Each keystroke is of 200-ms duration.
- A packet is transmitted on both key-up and key-down events.
- A "keep alive" is transmitted every 100 ms during the time a key is pressed, so that the receiver can detect if the RF link is lost, and in that unlikely case insert a "key up" event, to prevent a "stuck key" state being transmitted to the PC.
- When only one key is pressed, or on a "key-up" a 3-byte packet is transmitted.
- When two keys are pressed (for example a modifier key and a character key), a 4-byte packet is transmitted.
- 90% of transmissions result from single keystrokes, and 10% of packets report two keys simultaneously pressed.
- Cases where more than two keys are simultaneously pressed are judged sufficiently rare to be omitted from the calculation.
- In this application environment, the retransmission rate due to data errors will be well below 1 percent, and so this factor will be discounted.

The first stage is to calculate the average current during a transmission for each packet type.

The following timings and currents will be used [3]:

- 2 ms @ 2 mA Standby mode = 4 µAs
- 400 µs @ 30 mA *Idle* mode = 12 µAs
- 400 µs @ 57 mA *Receive* mode = 23 µAs
- 200 µs + 128 us/byte @ 69 mA *Transmit* mode
  - 2-byte packets = 32 µAs
  - 3-byte packets = 40 µAs
  - 4-byte packets = 48 µAs

The charge used during each transmission is therefore as follows:

- Keep alive = 71 µAs
- Single Key press = 79 µAs
- Two keys pressed = 87 µAs

Each keystroke includes the following sequence of events:

1. 3-byte or 4-byte packet on key make
2. Keep alive after 100 ms
3. 3-byte packet on key release.

Given the 10% dual key press model, the average charge per keystroke = (0.9 * 3 byte packet) + (0.1 * 4 byte packet) + keep alive + 3 byte packet. This produces an average value of 230 u as per keystroke.

This can be converted to an average current by multiplying this value by the number of keystrokes per second. Four key presses per second therefore gives an average current during keyboard operation, resulting from packet transmission of 0.92 mA.

[2] This usage model is regarded as being typical for a business user. This model will yield pessimistic estimates of battery life if applied to home use.

[3] Note that these values represent approximate values using optimized, application-specific firmware. More general purpose, less highly optimized firmware running on a slower MCU or more highly optimized firmware running on a faster MCU may differ significantly.

OBSOLETE

Four hours per day, five days per week is equivalent to a 12% duty cycle. The long term average radio current is therefore 110 µA. As this greatly exceeds the *Sleep* mode current (1 µA typical), *the Radio* sleep current will be discounted.

This case study will assume the use of an MCU with a 3-mA active current, a 100-µA sleep current with a wake-up timer running, and a 10-µA deep sleep current. It will be assumed that while a key is pressed, the MCU is awake for 20% of the time, scanning the key matrix and managing radio transmissions, and in Sleep mode with its wake-up timer running for the remaining 80% of the time. When no key is pressed, it is in deep Sleep mode.

During the four hours per day of keyboard usage, the average MCU current is therefore 550 µA. As the keyboard duty cycle is 12%, and the MCU draws 10 µA during the remaining 88% of the time, the long term average MCU current is therefore 75 µA.

An allowance of 5 µA will be made for leakages and quiescent currents other than from the power supply, giving a total average current of 190 µA.

The different battery options can now therefore be considered:

- Three AA cells used with a 3.0 V LDO having a 10-µA quiescent current will be able to access approximately 85% of the 2850-mAh battery capacity. This option therefore yields a battery life estimate of 504 days, which is over 16 months.
- Using a 3.3 V regulator with a 300-mV dropout voltage leaves an additional 10% of the battery capacity unused, providing a battery life estimate of just under 15 months.
- Three AAA cells (with a capacity of 1250 mAh) can therefore provide a life of 1250/2850 of this period which is 221 days—a little over seven months using a 3.0 V LDO, or six months with a lower cost, higher dropout voltage 3.3 V regulator.
- Two AA cells powering *the Radio* directly [4], assuming 25% of capacity available to 1.35 V provides a battery life of 155 days, or five months.
- Two AAA cells, using an 85% efficient boost converter with a 100-µA quiescent current. In this case all 2850 mAh is available. As the life-average voltage of the cells combined is approximately 2.4 V [5], using a 3.0 V regulation voltage, the battery current will be approximately 1.5 times the 3.0 V current. The average battery current is therefore 100 + (1.5*190) = 385 µA. The 1250-mAh capacity of the batteries therefore provides a life of approximately 3250 hours, or 183 days, which is approximately six months.
- Single AA cell, using an 85% efficient boost converter with a 100-µA quiescent current. In this case 2850 mAh is available. As the life-average voltage of the cell is approximately 1.2 V, using a 3.0 V regulation voltage, the battery current will be approximately three times the 3.0 V current. The average battery current is therefore 100 + (3*190) = 670 µA. The 2850-mAh capacity of the battery therefore provides a life of 177 days, which is a approximately six months.

With this information, the designer is equipped to make a decision about which batteries best suit the application. The most cost conscious designer may find five-month life acceptable, and therefore choose to power the keyboard directly from two AA cells, believing that in many cases *the Radio* may operate somewhat below 2.7 V, in which case five months may be an underestimate. The designer building a very space-constrained product may select a single AA cell, and be prepared to accept the small cost premium for the boost converter, and be happy with a six-month life. An engineer seeking the longest possible battery life may choose to use three AA cells to achieve 16 months of battery life.

Case Study 2: Wireless Optical Mouse

Cypress has conducted research to establish a mouse usage model, based on logging of usage for a variety of different business PC users. Although wide variations were found between the extremes, the majority of users logged fit the following model quite well.

Using this model, the mouse is in active use for 1 hour per day. “active use” means that the mouse has been moved within the last 10 seconds. Agilent optical sensors feature an automatic power saving mode that significantly reduces current consumption after one second since the last motion was detected. The use model provides for 75% of the use period to be spent in *Active* mode. During the active period, mouse motion is detected in 20% of sampling intervals.

It should be remembered that a typical business user will get much lower battery life than a typical home user simply because the typical business user is actively using their mouse for many more hours per week than a typical home user. Therefore the battery life calculated in this example is likely to be significantly lower than that which a typical home user would experience.

[4] Use of a capacitor to supply some of the peak current is assumed. This is necessary to flatten the discharge curve; periodic 69-mA discharging of the battery will reduce the available capacity at 1.35 V.

[5] An alternative calculation is to use the amount of energy in the battery. A typical AA cell stores 3.4 Wh. Dividing this number by the number of mAh gives the average voltage used here.

SOFT

Obsolete

Before proceeding to calculate battery life, it is necessary to examine how power is typically managed in wireless optical mice. This is particularly important because the current consumed by the optical sensing circuit dominates the overall average active current. The power consumption patterns of a typical 27-MHz wireless optical mouse will therefore be used as a reference.

27-MHz mice typically read the optical sensor, and transmit a data packet once every 20 ms during active mode, because that is the time taken to transmit a data packet using this type of low throughput radio.

After one second in *Active* mode, the optical sensor automatically changes into a power saving mode, as described above. After 10s, the mouse MCU begins powering the optical sensor up and down on a reducing duty cycle over time, in order to reduce current consumption while maintaining the ability to detect movement. As the duty cycle reduces, the responsiveness of the mouse decreases, requiring more rapid movement of the mouse to cause motion to be recognized. After a protracted period of inactivity, the MCU turns off the optical sensor, and goes into a deep *Sleep* mode, from which it can only be awoken by pressing a key.

Table 1 shows the times spent in each mode by a typical wireless mouse, and the typical overall 3.3 V current consumption [6].

Table 1. 27-MHz Mouse Power Consumption

Mode	Time/day (hrs)	Average Icc
Active, radio transmitting	0.15	22 mA
Active	0.60	16 mA
Semi-Active	0.25	10 mA
Standby	7.00	1.5 mA
Sleep [7]	16.0	10 µA

It can therefore be seen that this 27-MHz radio has an average current consumption of 6 mA while transmitting. It should be noted that the largest contributor to overall average current will be the current drawn during the time spent in *Standby* mode.

A WirelessUSB 2-way mouse implementation will send 4-byte packets when the mouse is moving and no button is pressed, and 5-byte packets if a button is pressed or if scroll wheel movement is detected. The usage model provides for 10% of packets to be 5-byte packets, and 90% to be 4 bytes long.

Using the calculation from the keyboard case study, charge usage to transmit each packet type is 87 µAs for a 4-byte packet and 95 µAs for a 5-byte packet. Given the 90/10% split, this provides an 88uAs average charge per packet.

Using the same 20-ms update rate as the 27-MHz mouse, requires 50 packets per second when continuous motion is detected, giving an average active current in this state of 4.4 mA.

The WirelessUSB mouse reference design eliminates the *Standby* mode by using an inexpensive mechanical motion sensor. This enables the mouse to go into *Sleep* mode at the end of the semi-active mode. The mechanical motion sensor immediately wakes the MCU if the mouse is moved, which in turn places the optical sensor into *Active* mode, and eliminates the need to press a button to wake from *Sleep* mode.

[6] These currents assume that the mouse is operated on a white surface.

[7] This excludes the quiescent current of the boost converter, the effect of which greatly increases battery current in this state.

For the purposes of this calculation, it will be assumed that the MCU has the same current consumption, and is used in the same way as in the 27-MHz wireless mouse. As the sleep current of *the Radio* is not significantly different from the inactive current of the 27-MHz transmitter, this will lead to current consumption in the given modes as shown in Table 2.

Table 2. WirelessUSB‰ Mouse Current Consumption

Mode	Hrs/day	Average Icc
Active, radio transmitting	0.15	20.4 mA
Active	0.60	16 mA
Semi-Active	0.25	10 mA
Sleep [8]	23.0	10 µA

The average current consumed by the WirelessUSB mouse in active and semi-active mode is therefore 15.6 mA. As the mouse is in active or semi-active mode for only five hours per week, the duty cycle is approximately 3%. The contribution to overall average current from active and semi-active modes is therefore 470 µA. The sleep current is dependent on the battery configuration used, and so is calculated separately for each of the options examined below.

Mechanical considerations will limit the choice of batteries in a mouse application. The battery life for each of the options is therefore as follows:

- Two AA cells (this is the most common option used in wireless mice). Because a typical optical mouse sensor requires a 3.3 V supply voltage, this option requires the use of a DC-DC boost converter. A 90% efficient converter, with a 120 µA quiescent current is assumed. Given a life-average battery voltage 1.2 V [9], the average battery current in active modes is approximately 50% higher than the 3.3 V consumption. This gives an average current resulting from the active mode of 705 µA. Applying the 97% duty cycle to the 135-µA sleep current at battery voltage adds 130 µA, giving an overall average current of 835 µA. Using a DC-DC boost converter allows use of the entire battery capacity, so 2850 mAh AA cells will give a battery life of 142 days, which is approximately 20 weeks, or almost five months.
- Three AAA cells occupy a similar volume as a pair of AA cells. In this case, a 3.3V linear regulator would be used giving access to 75% of the 1250-mAh capacity. Adding the 470uA average current resulting from the non sleep time to the 10uA sleep current resulting from the 97% sleep duty cycle gives an overall average current of 480 uA. The 937mAh of available battery capacity therefore gives a battery life of 81 days.

The clear choice in this case is therefore a pair of AA cells.

These calculations assumed that the full available throughput WirelessUSB was not taken advantage of to improve on the 20ms update rate of the 27-MHz mouse. Many designers may wish to use the increased throughput offered by WirelessUSB to improve the responsiveness of the mouse.

Reducing the update interval to 10 ms effectively doubles *the Radio* current in active mode to 8.8mA. This adds approximately 660 uA to the 15.6-mA non-*Sleep* mode average current, giving 16.3 mA. After applying the 3% duty cycle factor, this translates into an increase in the overall average current resulting from the non-*Sleep* modes to 490 µA. This 10-µA increase in overall average battery current, reduces overall battery life in the 2xAA case by just two days, to 140 days.

It can therefore be seen that the battery life of wireless optical mice is relatively unaffected by changes in radio current. This leads to two conclusions: firstly, it is recommended that designers take full advantage of the high WirelessUSB throughput by increasing the update rate to 8 ms, matching the typical low speed USB polling rate. Secondly, it is recommended that designers seeking to maximize battery life investigate methods of reducing the power consumed by the rest of the design and in lower power modes when *the Radio* is not in use. The Cypress WirelessUSB Mouse/Keyboard Reference Design includes several such optimizations.

Case Study 3: Wireless Sensor

This case study examines power consumption in a much lower duty cycle application. The sensor must transmit a 5-byte data packet every minute.

Using the previously calculated value of 95 µAs for a 5-byte packet, the average current consumption resulting from data transmissions will be 1.6 µA. To this is added the *Sleep* mode current of 1 µA; as the duty cycle is so low, it is not necessary to apply a duty cycle factor to this value, given the approximate nature of battery life estimates. The overall average radio current is therefore 2.6 µA.

[8] This excludes the quiescent current of the boost converter, the effect of which greatly increases battery current in this state.

[9] An alternative calculation is to use the amount of energy in the battery. A typical AA cell stores 3.4 Wh. Dividing this number by the number of mAh gives the average voltage used here.

OBSOLETE

For the purposes of this example, we will assume that the MCU used has a very low power *Sleep* mode, even with its wake-up timer running, such as the Cypress PSoC™. This MCU family has a 3-µA sleep current with its wake-up timer enabled. In this example, the MCU wakes up and takes sensor measurements for 100 ms every minute, during which time it transmits data using *the Radio*.

The PSoC active current is 5 mA, and it will be assumed that other circuits consume 1 mA while the MCU is active, but are not powered at other times. Given a duty cycle of 0.1/60, the average mcu current resulting from the time in active mode is 10 µA. To this is added the 3-µA *Sleep* mode current, giving 13 µA overall average mcu current.

This case study highlights two important points about low duty-cycle, long service duration WirelessUSB applications. Firstly, *the Radio* current is generally significantly less than the overall system current. Secondly, the sleep and quiescent currents of the system are greater than the power consumed by data transmission. Designers can therefore specify WirelessUSB for such applications, confident that *the Radio* will not be the dominant factor is determining battery life.

Summary

In many applications, the current consumed by WirelessUSB device itself is not the dominant factor in the battery life of the system. Case studies 2 and 3 are good illustrations of such applications.

In cases where *the Radio* is the main consumer of power, the low Idle and Sleep mode currents of these devices enables long battery life, and the current consumed in the active modes is just one of several factors determining battery life.

About the Author

Name:	Sai Prashanth Chinnapalli.
Title:	Applications Engineer Staff

OBSOLETE

Document History

Document Title: Calculating Battery Life in WirelessUSB LS™ Systems - AN4002

Document Number: 001-15084

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	1013602	SPM	04/26/2007	New application note.
*A	1767565	CSAI	08/19/2008	Added Spec No. and new disclaimer; updated copyright date.
*B	3187175	CSAI	03/02/2011	Updated title. Updated in new template.
*C	4390888	CSAI	05/27/2014	Updated in new template. Completing Sunset Review.
*D	5740123	ANKC	05/23/2017	Obsoleting the AN

OBSOLETE

OBSOLETE

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

Automotive	cypress.com/go/automotive
Clocks & Buffers	cypress.com/go/clocks
Interface	cypress.com/go/interface
Lighting & Power Control	cypress.com/go/powerpsoc cypress.com/go/plc
Memory	cypress.com/go/memory
PSoC	cypress.com/go/psoc
Touch Sensing	cypress.com/go/touch
USB Controllers	cypress.com/go/usb
Wireless/RF	cypress.com/go/wireless

PSoC® Solutions

psoc.cypress.com/solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP

Cypress Developer Community

Community | Forums | Blogs | Video | Training

Technical Support

cypress.com/go/support

WirelessUSB and WirelessUSB LS are trademarks of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com

© Cypress Semiconductor Corporation, 2007-2017. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

This Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.