

EZRADIOPRO[®] LOW DUTY CYCLE MODE OPERATION

1. Introduction

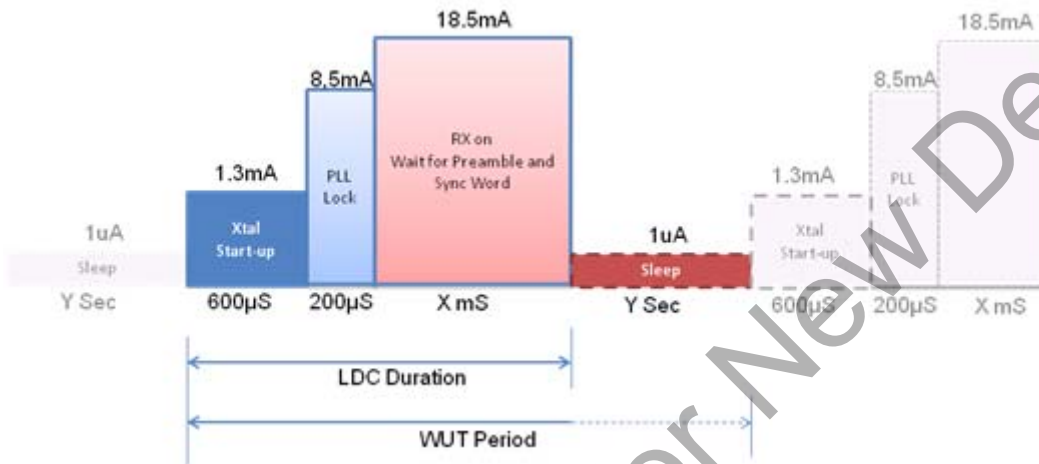


Figure 1. LDC Mode Operation Example

Low duty cycle (LDC) mode is designed to allow low average current polling operation of the Si443x RF receiver by providing “wake on radio” functionality.

In this mode, the receiver periodically wakes itself up to check if a valid signal is being transmitted. If a valid RF preamble is not detected, the chip will return to sleep mode at the end of the “LDC mode duration” and remain in sleep mode until the beginning of the next wake-up period.

If a valid preamble and sync are detected the receiver will remain enabled until the entire packet is received into the FIFO, or until a receive error is detected. This process takes place without intervention from the controlling MCU, which can remain in a very low power stand-by mode. The MCU can be notified, via various configurable interrupts, of the status of the receiver in this mode. Most typically, the MCU will only be interrupted when a valid packet has been received into the FIFO.

The LDC mode makes use of the transceiver’s internal wake-up timer module (WUT). This is based on a 32 kHz oscillator that can either be generated internally or, more accurately, be derived from an external 32.768 kHz crystal. This timer module can be configured to run in sleep mode, and also to generate an interrupt or GPIO signal at wake-up as required.

2. Current Consumption in Low Duty Cycle Mode

2.1. Main Factors

The sleep current, and duty cycle (ratio of sleep mode to active mode) are very significant factors in determining the average current consumption.

The wake-up time (transitioning between sleep and active) can also add to the overall current consumption—even though it is effectively “wasted” current. The EZRadioPRO transceivers minimize this wasted current by transitioning from sleep to active mode very quickly (see Section 2.2).

The Data Rate and Packet Size then dictate the active time of the receiver, when current consumption is at its highest. Using a faster data rate, thereby reducing the active time of the receiver, may therefore be more efficient in terms of battery life, as shown in Figure 2. Note, however, that increasing the data rate may require a corresponding increase in the receiver bandwidth, resulting in slightly reduced receiver sensitivity.



Figure 2. Comparison of Data rate and Corresponding Receive Time

2.2. Transceiver Operating Modes

The various operating modes of the transceiver are outlined in Table 1.

For low duty cycle mode, the lowest current consumption mode used is the $\sim 1 \mu\text{A}$ sleep mode which maintains the wake-up timer and the register contents.

At the start of the WUT period, the device automatically transitions through the ready and tuning modes and into to receive mode over an $800 \mu\text{s}$ period.

Table 1. Si443x Transceiver Operating Modes

Mode	I_{VDD}	Time to TX/RX	Description
Shutdown	15 nA	16.8 ms	Lowest power mode. Register contents lost.
Standby	450 nA	800 μs	Lowest power IDLE state. Registers maintained and can be accessed.
Sleep	1 μA	800 μs	As standby mode plus wake-up timer.
Sensor	1 μA	800 μs	As sleep mode plus optional low-battery detect and temperature sensor.
Xtal Start	1.3 mA	(Takes 600 μs)	Crystal osc starting up.
Ready	800 μA	200 μs	Crystal osc enabled. Reduces start-up time.
Tuning	8.5 mA	(Takes 200 μs)	PLL enabled.
Receive	18.5 mA	200 μs from Tx	A built-in sequencer automatically takes care of the transitions between the different states—e.g., crystal start-up, PLL lock, PA ramp, etc
Transmit	30 mA (+13 dBm)	200 μs from Rx	

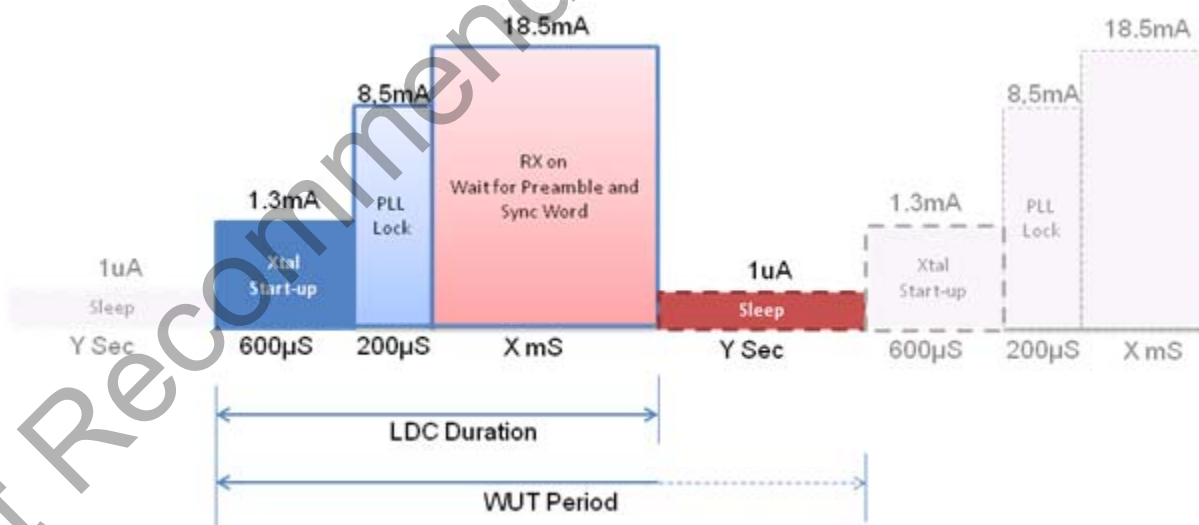


Figure 3. Current Consumption Profile during Polling

3. Low Duty Cycle Mode Operation

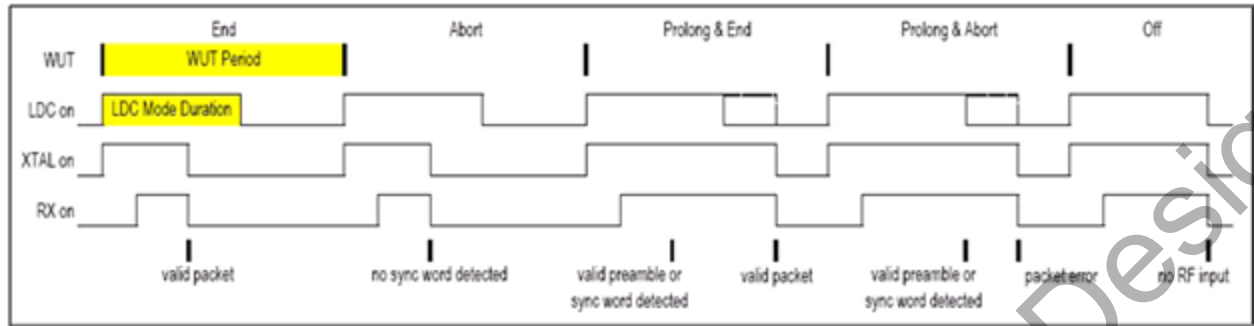


Figure 4. LDC Mode Outcomes

Figure 4 shows the different scenarios that can occur during the wake-up period when using low duty cycle mode. Those scenarios are as follows:

- **END**—A valid preamble and sync word are detected, and a valid packet is received, within the period of the LDC Mode Duration. The RX is then disabled following the successful packet reception, in advance of the LDC timeout.
- **ABORT**—A valid preamble is detected, but no valid sync word is received after the preamble detect threshold is reached, so the RX is disabled before the LDC timeout (see Section 4.4).
- **PROLONG and END**—A valid preamble, or preamble followed by sync word is detected during the LDC Mode Duration. The RX on-time is then prolonged, and in this case, a valid packet is received by the packet handler.
- **PROLONG and ABORT**—As above, a valid preamble, or preamble followed by sync word is detected during the LDC Mode Duration. The RX on-time is then prolonged, and in this case, the packet handler generates a packet error.
- **OFF**—No RF input is detected by the receiver, and the receiver is turned off at the end of the LDC Mode Duration.

4. Low Duty Cycle Mode Implementation

4.1. Calculating the Register Settings

The WUT period is set in conjunction with the LDC mode duration.

14	R/W	Wake-Up Timer Period 1	Reserved	Reserved	Reserved	wtr[4]	wtr[3]	wtr[2]	wtr[1]	wtr[0]	03h
15	R/W	Wake-Up Timer Period 2	wtm[15]	wtm[14]	wtm[13]	wtm[12]	wtm[11]	wtm[10]	wtm[9]	wtm[8]	00h
16	R/W	Wake-Up Timer Period 3	wtm[7]	wtm[6]	wtm[5]	wtm[4]	wtm[3]	wtm[2]	wtm[1]	wtm[0]	01h
17	R	Wake-Up Timer Value 1	wtv[15]	wtv[14]	wtv[13]	wtv[12]	wtv[11]	wtv[10]	wtv[9]	wtv[8]	—
18	R	Wake-Up Timer Value 2	wtv[7]	wtv[6]	wtv[5]	wtv[4]	wtv[3]	wtv[2]	wtv[1]	wtv[0]	—
19	R/W	Low-Duty Cycle Mode Duration	ldc[7]	ldc[6]	ldc[5]	ldc[4]	ldc[3]	ldc[2]	ldc[1]	ldc[0]	00h

Figure 5. LDC Mode Register Overview

The R (exponent) value (wtr[4:0] set in Register 14h) is shared between the WUT and the LDC mode duration. The value of R can be 0, and is limited to a maximum of decimal 20 (10100).

The M (mantissa) value (wtm[15:0] set in Registers 15h and 16h) is only used to define the WUT, and must be set to a minimum of decimal 1.

The LDC value (ldc[7:0] set in Register 19h) must be smaller than the specified M value, and must be set to a minimum of decimal 1.

The current 16-bit wake-up timer value can be read from Registers 17h and 18h.

The wake-up time (for Rev B1 silicon) is determined by the formula below:

$$WUT = 4 \times M \times 2^R / 32.768 \text{ ms}$$

The time of the LDC mode duration is determined by the formula below:

$$TLDC = LDC[7:0] \times [4 \times 2^R / 32.768] \text{ ms}$$

4.2. Example Register Calculation

An example of these register settings (to set a 1 second wake-up time, and a low duty cycle mode duration of 10 milliseconds) is shown below:

Wake Up Timer Period 1 [wtr4:0] = 0x00

Wake Up Timer Period 2 [wtm15:8] = 0x20

Wake Up Timer Period 3 [wtm7:0] = 0x00

Low Duty Cycle Mode Duration [ldc7:0] = 0x52

$$WUT = 4 \times M \times 2^R / 32.768 \text{ ms}$$

$$WUT = 4 \times 8192 \times 2^0 / 32.768 \text{ ms} = 1 \text{ sec}$$

$$TLDC = LDC [7:0] \times [4 \times 2^R / 32.768] \text{ ms}$$

$$TLDC = 82 \times [4 \times 2^0 / 32.768] \text{ ms} = 10 \text{ ms}$$

4.3. Wake-Up Timer Period and Low Duty Cycle Mode Duration

It is clear that, to achieve a low average system current using the LDC mode, the LDC Mode Duration should be a fraction of the total Wake-Up Timer period.

The selection of suitable timing parameters for using LDC mode, however, depends heavily on the design of the user's system, particularly the RF protocol.

The following sections provide example implementations of LDC mode.

4.3.1. Unsynchronized Master / Slave Network

Figure 6 is an example of a line-powered “master” node that needs to initiate communication with one of a number of battery powered “slave” nodes. In order to save cost, complexity, and power consumption, the individual slave nodes do not maintain an accurate time-base.

In the example in Figure 6, the master node cannot predict when the slave node will wake-up and listen for an incoming message. The master node should therefore repeat its packet transmission for the entire length of the slave's wake-up timer period. Each slave node should then listen for a preamble for a period of at least 1.5 packet times to ensure overlap with a valid message from the master.

Since the LDC duration is longer than the packet time in this case, it implies the use of a short packet to initiate communication while minimizing slave current consumption. If required, a larger data transfer can take place when the relevant slave responds to the master's initial message.

Referring to the example calculation in Section 4.2:

- The master node will repeat its short (5 ms) message for a period of 1 second.
- Each slave node will wake-up every 1 second, and listen for a period of 10 ms (2 packet times).

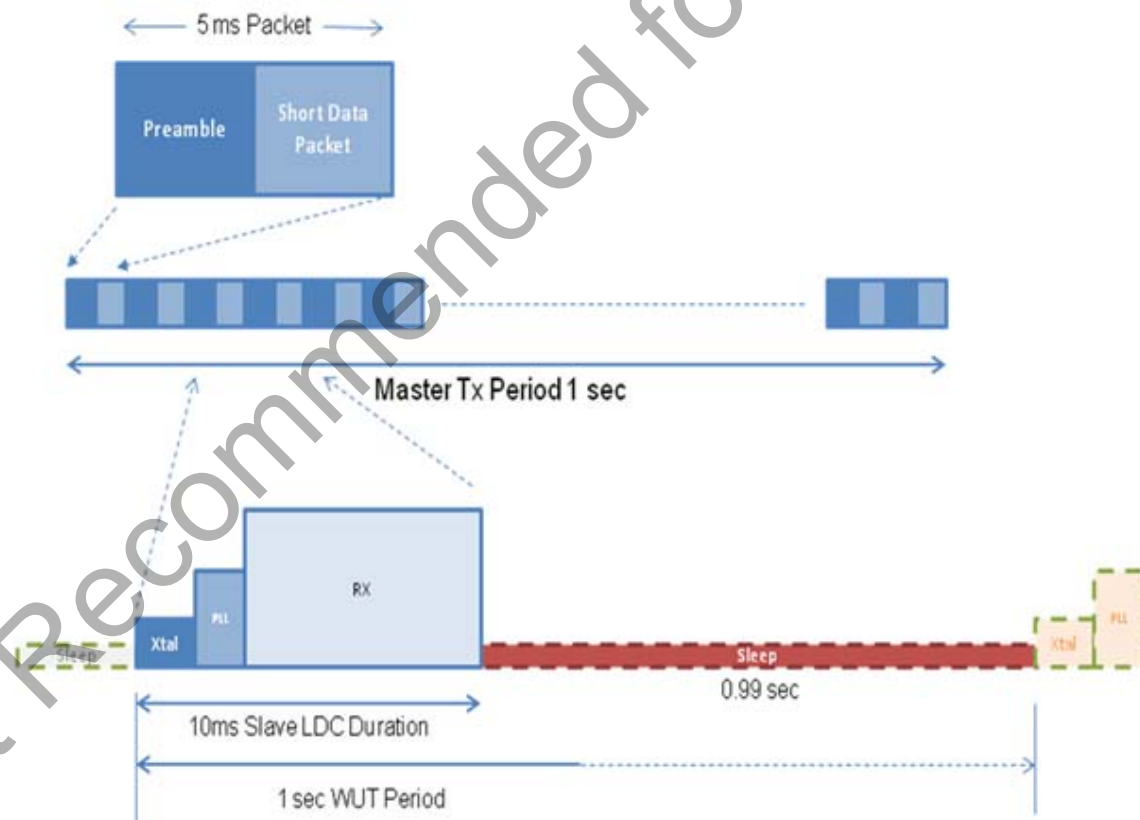


Figure 6. Example LDC Mode Timing

An example of this type of Master/Slave network is shown in the accompanying software example described in Section “5. Firmware Example Overview” .

4.3.2. Synchronized Peer Network

In this second example, the nodes within the network use synchronized communication. As such, a node within the network knows when to expect a valid message.

Due to the fact that the LDC Duration will be automatically prolonged until the end of a valid message (see Section "3. Low Duty Cycle Mode Operation"), the Duration can be set to a period just longer than the expected preamble plus sync word length. This LDC Duration should, however, be set to allow for any cumulative error in the synchronization between the nodes that could occur between valid messages being sent. This prolonged LDC Duration allows the system to send long data packets, while still benefitting from the reduced power consumption of LDC mode between messages.

The Wake-Up Timer period can be chosen to suit the power consumption and latency requirements of the system in question.

4.4. Wake-up Timer and LDC Mode Control Bits

enldm (Reg 08h, Bit 2)—Enable Low Duty Cycle Mode

If this bit is set, then Low Duty Cycle Mode is enabled. The wake-up frequency should be set in the Wake-Up Timer Period Registers (Registers 14h, 15h and 16h), and the minimum ON time should be set in the Low Duty Cycle Mode Duration register (Register 19h). The FIFO mode should also be enabled (dmod 1:0 - Register 71, Bits 4 and 5). It is **not** necessary to separately enable the Wake-Up Timer (enwt—Register 07h, Bit 5).

rxmpk (Reg 08, Bit 4)—RX Multi Packet

To ensure reliable operation with all possible receiver parameters, this bit should be set. In the unlikely event of a false preamble detection, enabling this bit prevents the receiver from incorrectly aborting the LDC operation (see Section "3. Low Duty Cycle Mode Operation" on page 4). Instead, it keeps the receiver active until a valid packet is received or until the LDC Mode Timeout expires.

X32ksel (Reg 07, Bit 4)—Internal / External Time Base

If this bit is cleared (default) then the Wake-Up Timer uses the internal RC oscillator as a clock source. If this bit is set, then the Wake-Up Timer uses the external 32.768 kHz crystal.

5. Firmware Example Overview

The “AN585SW.zip” file includes demonstration firmware that implements low duty cycle mode operation, as described in Section “4.3.1. Unsynchronized Master / Slave Network” .

Both Master and Slave firmware use compile time options to set:

- Master-to-Slave Data Rate (select between 9600 bps, 50 kbps and 100 kbps)
- Net ID and Node ID

The master firmware:

- Runs on the Silicon Labs SDBC-DK3 motherboard and Si4431 test card.
- Transmits 3 different data packets to 3 different Slave nodes in response to button presses.
- Runs a state machine that Transmits on a button press and then waits approximately 1 second to Receive a 10-byte acknowledgement from the Slave node.
- Shows a successful acknowledgement with the corresponding LED on the SDBC board.
- Defines the Transmit baud rate based on the compile-time Data Rate selection.
- Receives the acknowledgement using the default (40 kbps) Receiver settings.

The slave firmware:

- Runs on the Silicon Labs Si4431 EZLink module board and MSC-DBSB11 peripheral board.
- Using LDC mode, looks for a valid received data packet with its own Net and Node IDs.
- Runs a state machine to receive valid data packets in LDC mode and then transmit a 10-byte acknowledgement in response to its own Node ID.
- Configures LDC mode with a timeout that is 2 received data-packets long, based on the compile-time data rate selection.
- Defines the receiver modem parameters (Receiver bandwidth, Clock recovery and AFC) based on the compile-time data rate selection.
- Transmits the acknowledgement using the default (40 kbps) transmit data rate.

6. Limitations

The low duty cycle mode does not provide for dynamic changes in the operating frequency and modem settings, and as such, is only directly suitable for a single-channel RF system. For a multi-channel implementation, such as Silicon Labs' EZMacPRO, the wake-up timer can be used separately to provide a periodic wake-up signal to the MCU controlling the system. The MCU can then adjust the radio parameters as required at each wake-up in order to check multiple channels, thereby allowing the transceiver to act as a "system supervisor" to minimize total current consumption.

Some ultra low-power MCUs can support sleep mode currents—including real time clock operation—that are lower than the 1 μA sleep current of the Si443x transceivers. For example, Silicon Labs' C8051F9xx devices use 600 nA in sleep mode with the smaRTClock peripheral enabled. If a system design is based on one of these low-power MCUs, and the system uses an extended RF wake-up time (in the region of 30 seconds or more), then it may be more power efficient overall to use the MCU to put the transceiver into its 15 nA shutdown mode and reboot it at each RF wake-up.

7. Conclusion

The low duty cycle mode of the EZRadioPRO transceivers is a simple and effective way to reduce the current consumption of an RF transceiver system by autonomously waking up the receiver to look for transmitted packets, while the MCU remains in its lowest power shutdown mode.



Simplify Studio

One-click access to MCU tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!

www.silabs.com/simplify



MCU Portfolio
www.silabs.com/mcu



SW/HW
www.silabs.com/simplify



Quality
www.silabs.com/quality



Support and Community
community.silabs.com

Disclaimer

Silicon Laboratories intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Laboratories products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Laboratories reserves the right to make changes without further notice and limitation to product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Silicon Laboratories shall have no liability for the consequences of use of the information supplied herein. This document does not imply or express copyright licenses granted hereunder to design or fabricate any integrated circuits. The products must not be used within any Life Support System without the specific written consent of Silicon Laboratories. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Laboratories products are generally not intended for military applications. Silicon Laboratories products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons.

Trademark Information

Silicon Laboratories Inc., Silicon Laboratories, Silicon Labs, SiLabs and the Silicon Labs logo, CMEMS®, EFM, EFM32, EFR, Energy Micro, Energy Micro logo and combinations thereof, "the world's most energy friendly microcontrollers", Ember®, EZLink®, EZMac®, EZRadio®, EZRadioPRO®, DSPLL®, ISOmodem®, Precision32®, ProSLIC®, SiPHY®, USBXpress® and others are trademarks or registered trademarks of Silicon Laboratories Inc. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. All other products or brand names mentioned herein are trademarks of their respective holders.



Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

<http://www.silabs.com>