WICED Studio

CYW20706 and CYW20719 Low-Power Modes

Associated Part Family: CYW20706 and CYW20719

Doc. No.: 002-22800 Rev. **

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
www.cypress.com

Contents

About This Document

This document explains Power Management Unit (PMU) features, Low Power Oscillators (LPOs), and power modes supported in WICED Bluetooth devices CYW20706 and CYW20719. This document will help you design low-power IoT applications in WICED Studio with Cypress's dual-mode Bluetooth devices.

Purpose and Audience

This document is intended for application developers designing and creating low power applications on CYW20706 and CYW20719 devices using Cypress WICED Studio.

Scope

This document explains the functionalities of PMU and various sources of Low Power Oscillators (LPOs), and describes the low power modes supported in CYW20706 and CYW20719. This document does not include the detailed steps to create low power applications on WICED Bluetooth devices. This document might refer to certain terminologies specific to the devices; see the device datasheets for more information.

Acronyms and Abbreviations

In most cases, acronyms and abbreviations are defined on first use.

For a comprehensive list of acronyms and other terms used in Cypress documents, go to www.cypress.com/glossary.

IoT Resources and Technical Support

Cypress provides a wealth of data at http://www.cypress.com/internet-things-iot to help you to select the right IoT device for your design, and quickly and effectively integrate the device into your design. Cypress provides customer access to a wide range of information, including technical documentation, schematic diagrams, product bill of materials, PCB layout information, and software updates. Customers can acquire technical documentation and software from the Cypress Support Community website (http://community.cypress.com/).

1 Power Management Unit (PMU)

The PMU is responsible in lowering the average power consumption of the device. The PMU is mainly a software implementation and has necessary supporting hardware. The PMU module is executed from the context of the lowest priority thread. Whenever the PMU gets the CPU, it tries to put the system into one of the supported low-power mode. The primary tasks of the PMU include:

- Implementing low power modes, transitions, and timings
- Communicating with different blocks and modules to register callbacks for pre-sleep and post-sleep
- Implementing the wake ISR, which wakes up the device at the required time
- Detecting, switching, and calibrating of LPOs

1.1 Low Power Oscillator (LPO)

LPO is a clock that runs in low power modes sourcing some blocks and is responsible for the timing for wakeup. This clock has low power consumption, which makes it ideal for operation in low-power modes. The Bluetooth Specification specifies an accuracy of +-250 ppm for Basic Rate/Enhanced Data Rate (BR/EDR) and +-500 ppm for Bluetooth Low Energy (BLE). Hence, the LPO accuracy should lie within this range if there is an active Bluetooth operation.

There are multiple possible sources for the LPO in both devices. They are:

- Low Power Xtal (LPX): This clock is derived from the external main crystal oscillator, but runs in low power during sleep. LPX typically has a frequency of 1 MHz and provides greater clock accuracy of at least 20 ppm, but consumes relatively higher power compared to other LPO sources. LPX is useful when high accuracy is required without increasing BOM; and it serves as a substitute for the external 32 kHz crystal.
- External LPO clock: Both devices have pins to which an external digital clock or a 32.768 kHz crystal can be connected. The accuracy should be within +-250 ppm as specified by the Bluetooth specification. If an external LPO is used, then the PMU will most likely use this under all conditions when a 20-ppm clock is not required. If a 20-ppm accurate clock is required at any time, then the LPX will be used instead of the external LPO clock.
- Internal LPOs: The internal LPOs belong to the Lean High Land (LHL) domain. There are two internal LPOs: one which has lower accuracy (~500 ppm) and consumes lower power (2-3 uA) and another which has higher accuracy (~50 ppm) and consumes higher power (10 uA). The PMU calibrates these clocks at regular intervals to improve the accuracy, thereby decreasing power consumption.

1.2 Tasks of PMU in LPOs

- **Detection**: PMU detects external digital LPO or external crystal during boot. CYW20706 supports auto-detection of an external LPO. The PMU detects the presence or absence of an external crystal or external digital clock. On the other hand, CYW20719 does not support auto-detection. An entry in the *patch.cgs* file[1] determines if an external LPO source is present. In WICED, the entry is present by default, so an external LPO source is expected. If an external LPO source is absent, remove the below code entry from the *patch.cgs* file.

```
ENTRY "Data" = "g_aon_flags"
{
  "Address" = 0x0028067c
  "Data" =
   COMMENTED_BYTES
     {
        <hex>
        81
     } END_COMMENTED_BYTES
}
```

[1] File path: *\20719-B1_Bluetooth\WICED\internal\20719B1\patches*

- **Switching**: PMU switches between different LPOs. The active peripherals can request a clock with a ppm value, and the PMU will select the LPO source based on the required PPM. The PMU might also turn OFF unused LPO sources to save power. The usual preference for the LPO source is in the order: External LPO clock > Internal LPO > LPX. Although it is possible for the WICED Bluetooth devices to enter low power modes using internal LPOs, it is recommended to use an external crystal.
- **Calibration:** PMU calibrates the internal LPO at regular intervals to improve the accuracy, and thereby decreases power consumption. The calibration is performed against the external high accuracy (20 ppm) clock. PMU calibrates the LPO at every 500 ms when active. If in low power mode, the calibration is done when the chip wakes up. The calibration takes place only if the internal LPOs are selected as the source.

1.3 PMU Architecture

The PMU module executes from the context of the lowest priority thread. PMU interacts with the other modules to determine if it can put the system to sleep. PMU collects necessary information to decide whether go to sleep and the duration. The PMU manages the states of the clocks and the LDO voltages in the low power modes.

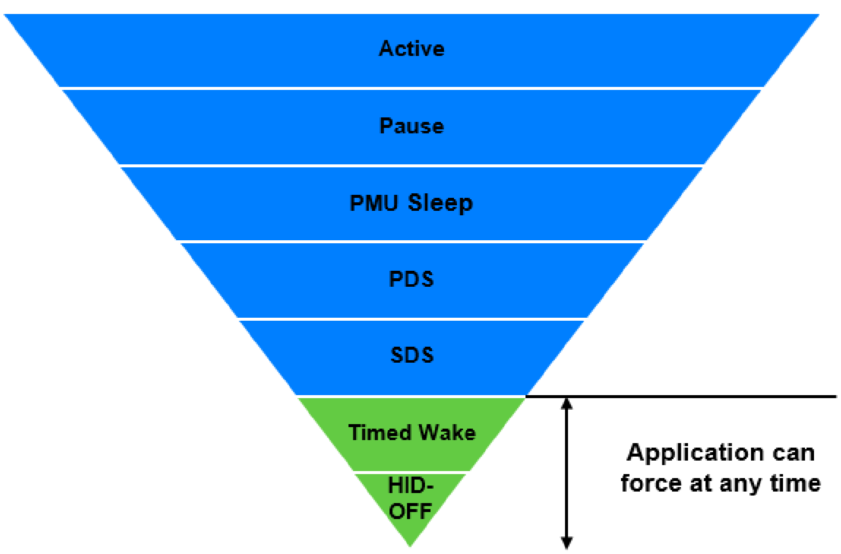
The differences between the different power modes lies in

- the blocks that are turned OFF
- the duration for which the system goes to sleep
- the accuracy of the internal clock while sleeping
- the wakeup sequences

2 Power Modes

WICED Bluetooth devices support different power modes. However, it is important to note that all the devices do not support all the modes.

- **Active**: Active mode is the normal operating mode in which all peripherals are available and the CPU is active.
- **Pause**: In this mode, the CPU is in Wait for Interrupt (WFI) and the HCLK, which is the high frequency clock derived from the main crystal oscillator, is running at a lower clock speed. Other clocks are active and the state of the entire chip is retained. Pause mode is chosen when the other lower power modes are not possible.
- **PMU Sleep**: In this mode, CPU is in WFI and the HCLK is not running. The PMU determines if other clocks can be turned off and does accordingly. State of the entire chip is retained, the internal LDOs run at a lower voltage (voltage is managed by the PMU), and SRAM is retained.
- **Power Down Sleep (PDS)**: This mode is an extension of the PMU Sleep wherein most of the peripherals such as UART, and SPI are turned OFF. The entire memory is retained, and on wakeup the execution resumes from where it was paused.
- **Shut Down Sleep (SDS)**: Everything is turned OFF except LHL, RTC, and LPO. The device can come out of this mode either due to Bluetooth activity or a LHL interrupt. This mode makes use of micro Bluetooth Core Scheduler (uBCS), to be read as micro-BCS) which is a compressed scheduler different from the regular BCS. Before going into this mode, the application can store some bytes of data into the Always On RAM (AON). When the device comes out of this mode, the data from AON is restored. After waking from SDS, the application will start from the beginning (warmboot) and has to restore its state based on information stored in AON. In the SDS mode, a single Bluetooth task with no data activity, such as an ACL connection, BLE connection, or BLE advertisement can be performed. If there is data activity during these tasks, the system will undergo full boot and normal BCS will be called. This mode is available in CYW20719, but not in CYW20706.
- **Timed-Wake**: The device can enter this mode asynchronously, that is. the application can force the device into this mode at any time without asking the permission from other blocks. LHL, RTC, and LPO are the only active blocks. A timer that runs off the LPO is used to wake the device up after a pre-determined fixed time.
- **HID-OFF**[2]: This mode is similar to Timed-Wake, but in HID-OFF mode even the LPO and RTC are turned OFF. So, the only wakeup source is a LHL interrupt.



[2] There is a small difference between SDS and HIDOFF. The PMU decides and controls when the device can be put into SDS mode. On the other hand, device can be put into HIDOFF asynchronously. In CYW20719, there is support for SDS but not for HIDOFF. However, developers need not worry as the difference in current consumption is negligible.

2.1 Power Modes Supported by CYW20706 and CYW20719

Table 2-1 lists the power modes supported by both CYW20706 and CYW20719.

	Active	Pause	PMU Sleep	PDS	SDS	Timed Wake and HID-OFF
CYW20706	✓		✓	✓		✓
CYW20719	✓	✓	✓	✓	✓	

Table 2-1. Supported Power Modes

3 WICED APIs

The PMU functionality and low power modes might seem complex, but WICED offers drivers that take care of low power operation. You will have to call a few functions and be able to observe the low power consumption.

3.1 CYW20706

The header file *wiced_power_save.h* contains the APIs related to low power for CYW20706.

- `wiced_sleep_config()`: Use this API to enable or disable the low power operation of the device. PMU will determine whether the system can go to low power, and will put the device to the appropriate low power state. The application will resume on waking from sleep.
- `wiced_power_save_start()`: Use this API to put the device into HID-OFF or Timed-Wake. The wakeup source should be passed as an argument. The wakeup source can be a LHL GPIO (`WICED_WAKE_SOURCE_GPIO`) interrupt, timer interrupt (`WICED_WAKE_SOURCE_TIMEOUT`), or both. Application will be restarted on exiting out of the power save mode. If the application wants to save any state before going into power save, it can use the `wiced_power_save_store_state()` API. The application can use the `wiced_power_save_retrieve_state()` API to retrieve the state on coming back from power save.

See the CYW20706 - BLE Low Power Beacon example which demonstrates low power operation in CYW20706.

3.2 CYW20719

The header file *wiced_sleep.h* contains the APIs related to low power operation of CYW20719.

- `wiced_sleep_configure()`: Use this API to enable the low power operation of the device. The parameter to be passed to this API contains a callback that will be called by the PMU to poll for sleep permission. In the callback, the application has to return one of these values based on the requirements:
  - `WICED_SLEEP_NOT_ALLOWED` – The application can return this value if it does not want the device to go to Sleep mode.
  - `WICED_SLEEP_ALLOWED_WITHOUT_SHUTDOWN` -The application can return this value if low power is allowed; but if the device should not enter SDS. This means that the lowest power mode that the device can enter is PDS. This value should be passed if data exchange over Bluetooth is expected and entering SDS will be irrelevant.
  - `WICED_SLEEP_ALLOWED_WITH_SHUTDOWN` – When this value is returned, the device can enter any of the low power modes including SDS.

See the low_power_sensor example included in WICED. The example demonstrates how low power can be configured in CYW20719.

4 Conclusion

WICED Bluetooth devices support ultra-low power operation which is essential for many IoT applications. Although architecturally complex, WICED Studio enables easy usage of APIs to configure these devices for low power operation. Examples of low power applications are included in the SDK and you can easily adopt them in your projects.

Document Revision History

Document Title: CYW20706 and CYW20719 Low-Power Modes

Document Number: 002-22800

Revision	ECN	Issue Date	Description of Change
**	6062087	01/18/2018	Initial release

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

Arm® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6 MCU

Cypress Developer Community

Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

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