Objective

This code example demonstrates accurate time keeping with PSoC® 6 MCU's real-time clock (RTC), which is synchronized with a current time server such as an iPhone using the BLE current time service (CTS).

Overview

This code example demonstrates accurate time keeping with the RTC of PSoC 6 MCU with BLE Connectivity (PSoC 6 MCU), which also generates alarms (interrupts) at every one minute to show time information on an E-INK display. In addition, a BLE CTS is used to synchronize time and date with a current time server such as an iPhone.

This code example assumes that you are familiar with the PSoC 6 MCU and the PSoC Creator™ Integrated Design Environment (IDE). If you are new to PSoC 6 MCU, you can find introductions in the application note AN210781 – Getting Started with PSoC 6 MCU with Bluetooth Low Energy (BLE) Connectivity.

Requirements

Tool: PSoC Creator 4.2

Programming Language: C (Arm® GCC 5.4.1)

Associated Parts: All PSoC 6 MCUs with BLE Connectivity

Related Hardware: CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit

Design

PSoC 6 MCUs have a fully featured RTC that keeps track of the current time and date independent of the CPU. The RTC is clocked from an accurate 32768-Hz watch crystal oscillator (WCO). The RTC has a programmable alarm feature, which generates interrupts at a specified time and date with the capability to wake up the system from low power modes. In this code example, the alarm feature is used to generate interrupts at one-minute intervals to update an E-INK display with the current time and date. E-INK displays consume no power for display retention; therefore, the power supply of the display is turned off after an update to reduce the average power consumption. See code example CE218133 – PSoC 6 MCU E-INK Display with CapSense.

The BLE Component provides a CTS that allows a GATT time client to get the current time and date information from a GATT time server. iOS devices have built-in BLE time servers that allow BLE GATT time clients to connect to them and extract the time information. This code example utilizes the time server feature of iOS devices to fetch the current time value on establishing a BLE connection and then initializes the RTC with the time information.

Figure 1 shows the functional block diagram of this code example.

Figure 1. BLE Current Time Service Configuration

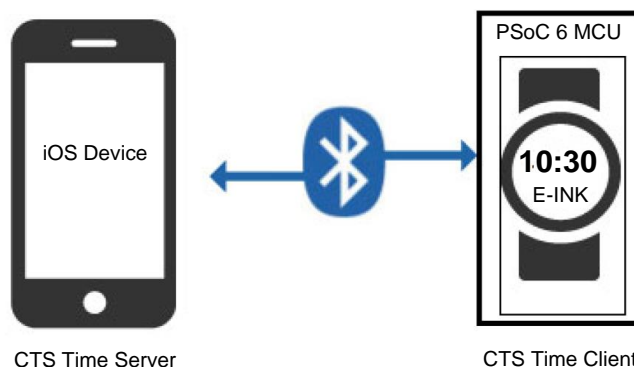CTS Time Server   CTS Time Client

Figure 2 and

Figure 3 show the TopDesign schematic of this code example. In addition to the BLE, RTC, and the E-INK display, this code example includes two LEDs that are used to show BLE status, a multi-counter watchdog timer and associated interrupt that controls LED timing, and a GPIO interrupt that is used to restart BLE advertisement.

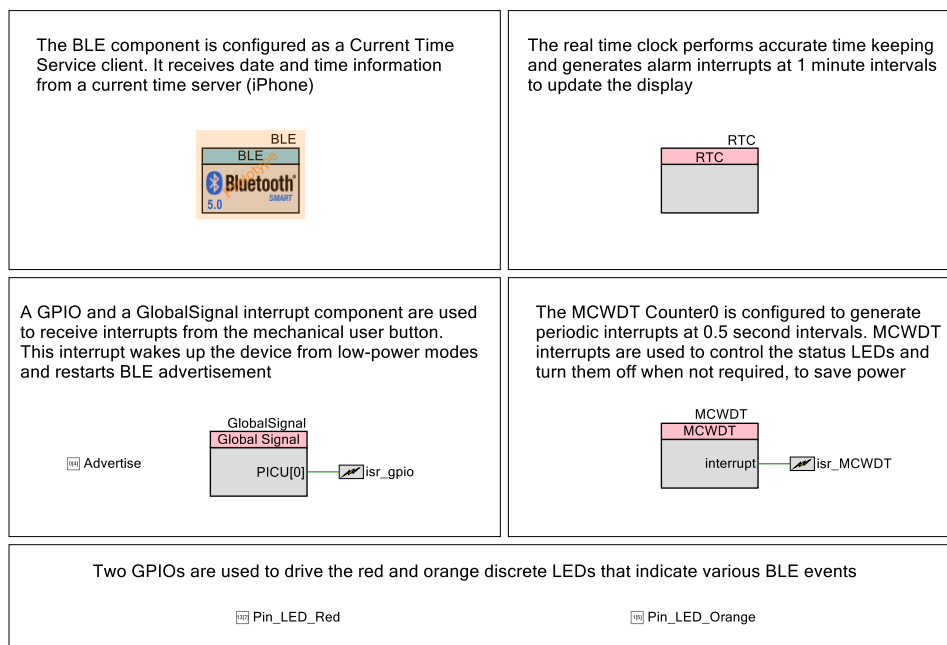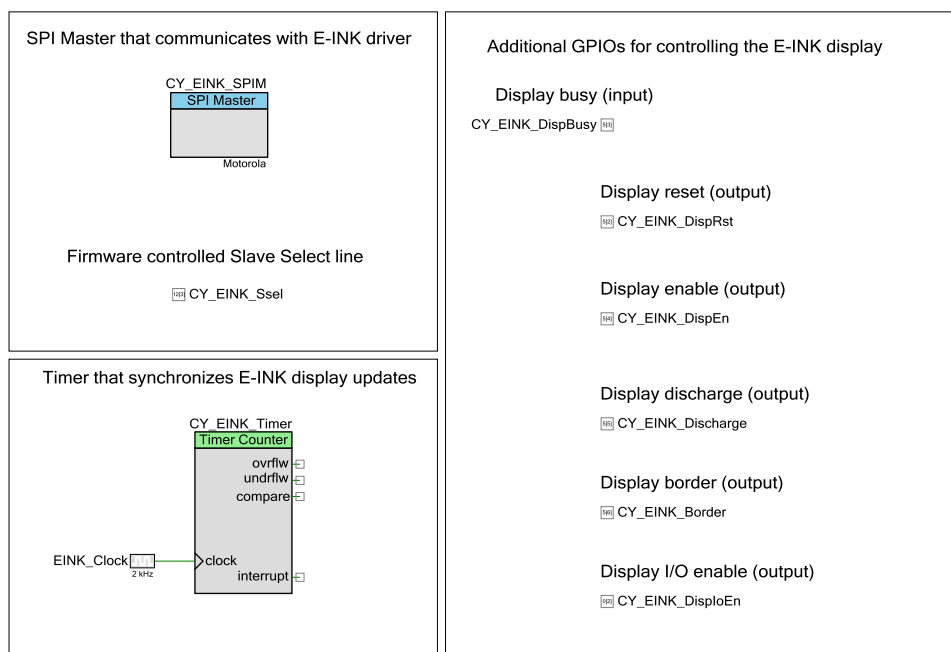Figure 2. TopDesign Schematic: BLE, RTC, Interrupts, and LEDs

The BLE component is configured as a Current Time Service client. It receives date and time information from a current time server (iPhone)

BLE

The real time clock performs accurate time keeping and generates alarm interrupts at 1 minute intervals to update the display

RTC

A GPIO and a GlobalSignal interrupt component are used to receive interrupts from the mechanical user button. This interrupt wakes up the device from low-power modes and restarts BLE advertisement

GlobalSignal
Global Signal
Advertise   PICU[0]   isr_gpio

The MCWDT Counter0 is configured to generate periodic interrupts at 0.5 second intervals. MCWDT interrupts are used to control the status LEDs and turn them off when not required, to save power

MCWDT
interrupt   isr_MCWDT

Two GPIOs are used to drive the red and orange discrete LEDs that indicate various BLE events

Pin_LED_Red   Pin_LED_Orange

Figure 3. TopDesign Schematic: E-INK Display Library

SPI Master that communicates with E-INK driver

CY_EINK_SPIM
SPI Master
Motorola

Firmware controlled Slave Select line

CY_EINK_Ssel

Timer that synchronizes E-INK display updates

CY_EINK_Timer
Timer Counter
ovrflw
undrflw
compare
EINK_Clock 2 kHz   clock
interrupt

Additional GPIOs for controlling the E-INK display

Display busy (input)
CY_EINK_DispBusy

Display reset (output)
CY_EINK_DispRst

Display enable (output)
CY_EINK_DispEn

Display discharge (output)
CY_EINK_Discharge

Display border (output)
CY_EINK_Border

Display I/O enable (output)
CY_EINK_DispIoEn

The code example consists of the following files:

- *main_cm0p.c* contains functions that starts up the BLE controller, starts up CM4, and services BLE stack events.
- *main_cm4.c* contains the main CM4 function, which is the entry point and execution of the firmware application. The main function calls the initializing functions, continuously processes BLE events, updates the E-INK display when required, and enters low power modes to save power if the system conditions permit.
- *ble_application.c/.h* contain all the macros and function definitions related to BLE communication and operation. They include the definition of the event callback function that is registered with the BLE Component at startup. The callback function is used to send BLE-related events from the BLE stack to the application layer for processing. In addition, these files contain CTS specific event handlers.
- *real_time_clock.c/h* contain the functions that initialize, read and write to the RTC.
- *display.c/.h* contain the functions that initialize and refresh the E-INK display[1].
- *screen_contents.c/h* contain the text and background images used by the display module.
- *led.c/.h* contain the functions that control the status LEDs.
- *low_power.c/h* contain functions to make the system enter low-power modes and turn OFF the status LEDs depending on system-level conditions.

Figure 4 shows the firmware flow of this code example.

---

[1] For a detailed list of files included in the E-INK Library, see the code example, CE218133 – PSoC 6 MCU E-INK Display with CapSense.

Figure 4. Firmware Flow
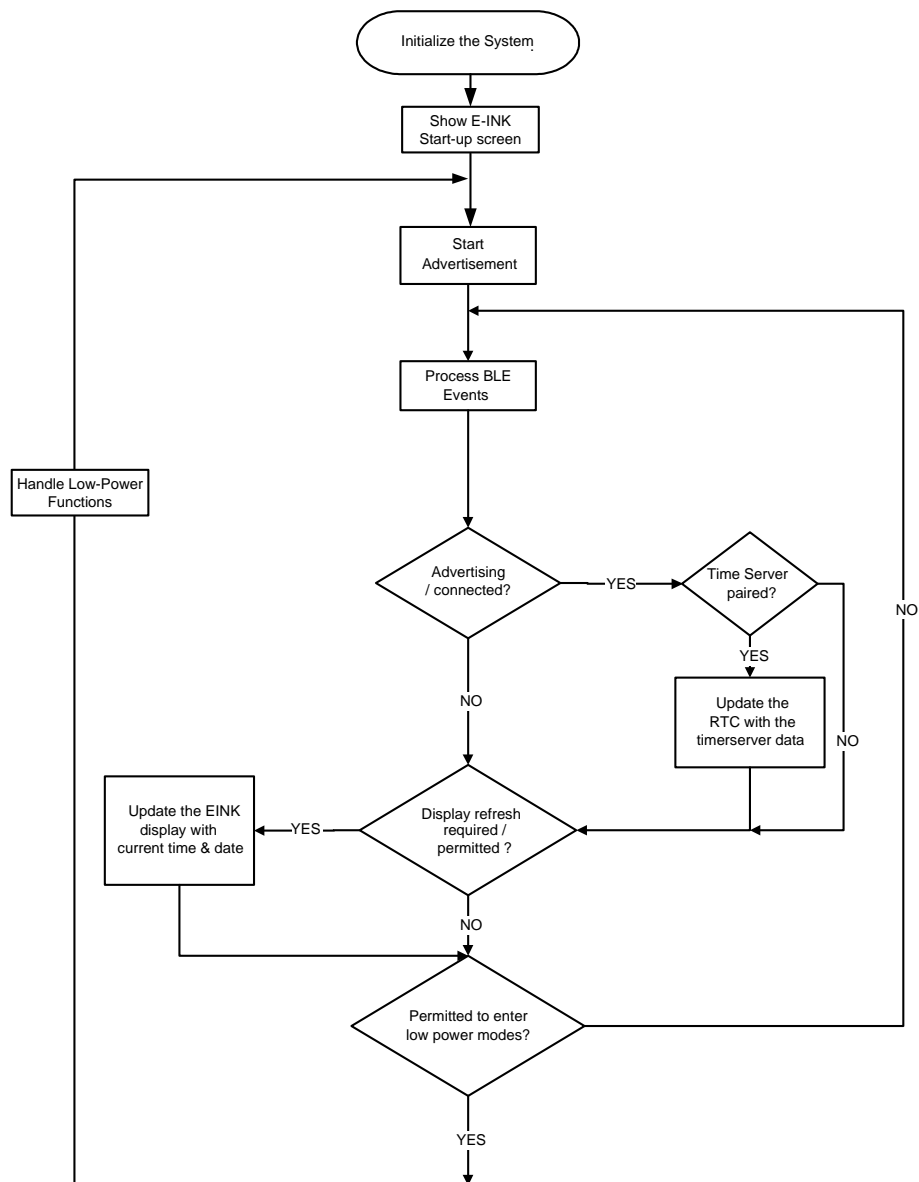
Hardware Setup

Set the switches and jumpers on the Pioneer Board as shown in Table 1.

Table 1. Switch and Jumper Selection

Switch/Jumper	Position	Location
SW5	3.3 V	Front
SW6	PSoC 6 BLE	Back
SW7	V_{DDD} /KitProg2	Back
J8	Installed	Back

Figure 5. Hardware Setup
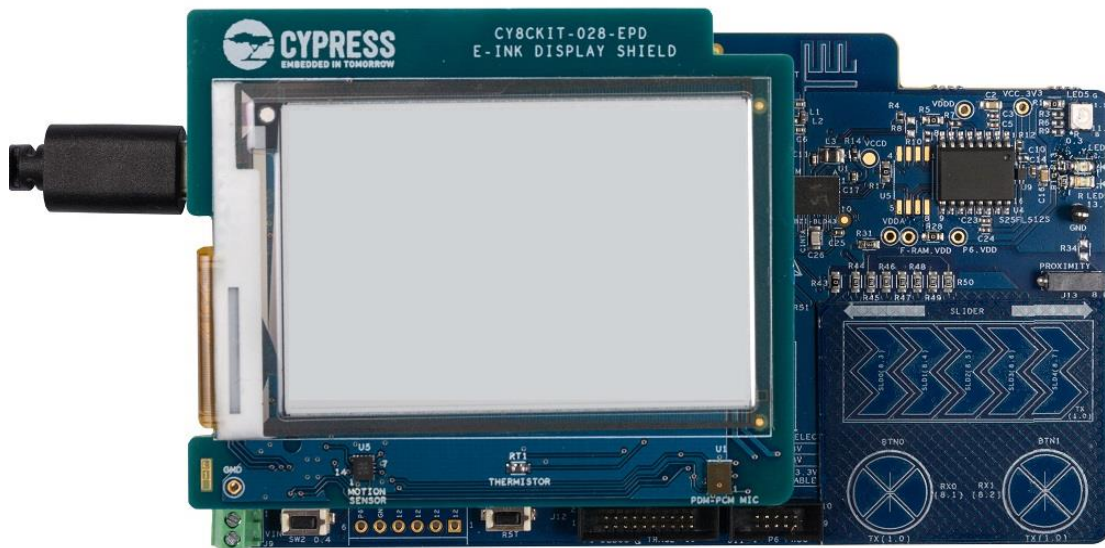
Software Setup

Install the CY8CKIT-62-BLE PSoC 6 BLE Pioneer Kit software, which contains all the required software to evaluate this code example. No additional software setup is required.

PSoC Creator Components

See the PSoC Creator project for details of PSoC Component configurations and system wide resource settings.

Operation

To verify the code example using an iOS device, follow these steps:

Note: This code example requires an iOS device with iOS 8 or a later version to evaluate. Android devices do not support the Current Time Service.

1. Power the Pioneer Board through the USB connector **J10**.

2. Program the Pioneer Board with the *CE220186 _RTC_CTS* project. See the Pioneer Kit guide for details on how to program firmware into the device.

   After programming successfully, the E-INK display will refresh and show the default time and date, and the instructions to use this project. BLE will start advertising with an advertising timeout of 20 seconds. The orange LED (**LED8**) remains ON during this period to indicate the BLE advertising state.

Figure 6. BLE Advertising

3. If BLE advertisement has timed out (**LED8** is OFF), press **SW2** to restart advertisement.

4. Open *Settings* on your iOS device, and select *Bluetooth*. From the *Bluetooth* settings, turn ON Bluetooth as Figure 7 shows.

Figure 7. Turning ON Bluetooth
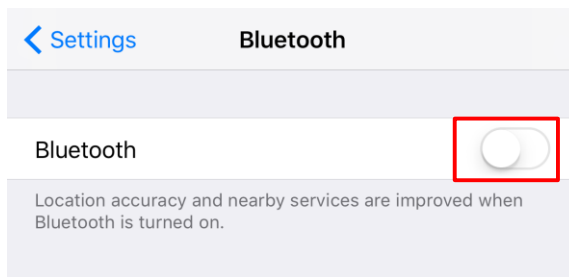
5. After Bluetooth is turned ON, the application will automatically search for available devices and will list them. Select the **RTC CTS** device as shown in Figure 8. A successful connection is indicated by **LED8** continuously blinking at half second intervals.

Figure 8. Selecting the Device

6. When connected, the PSoC 6 MCU sends a pairing request to the iOS device. Accept the pairing request as Figure 9 shows.

Figure 9. PSoC 6 MCU Pairing Request
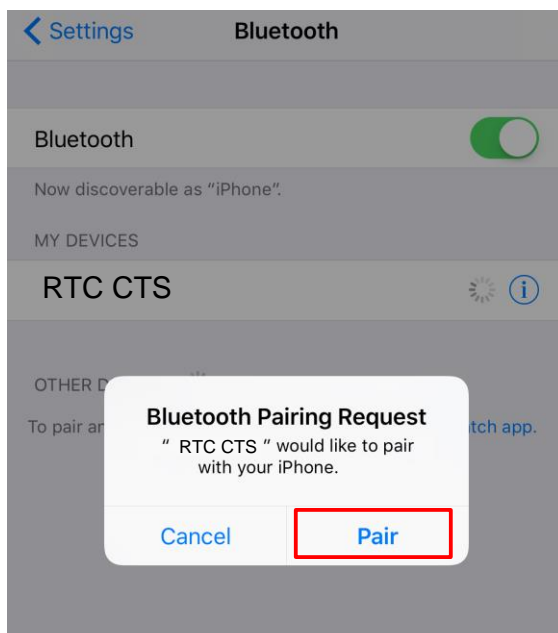
7. Upon pairing, PSoC 6 MCU receives the current time and date from the iOS device and updates the RTC accordingly. The E-INK display will refresh to show the updated time and date.

Figure 10. Display Update

8. After synchronizing time and date, the red LED (LED9) will turn ON for three seconds to indicate a disconnect event.

Figure 11. Disconnect Indication
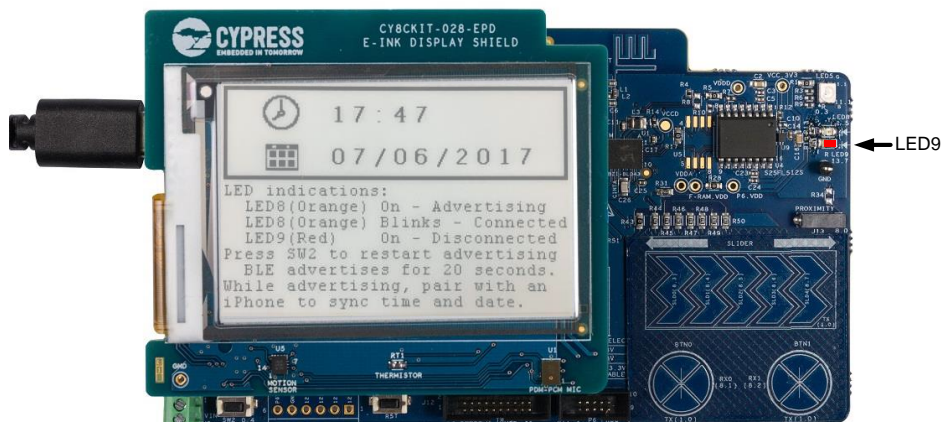
9. The display will keep updating at one-minute intervals. Press **SW2** to restart the advertisement, if required.

Components

Table 2. List of PSoC Creator Components

Component	Instance Name	Function
BLE	BLE	The BLE Component is configured as a current-time-service client. It can receive date and time information from a current time server such as an iPhone.
RTC	RTC	The real-time clock performs accurate time keeping and generates alarm interrupts at 1-minute intervals to update the display.
MCWDT	MCWDT	The MCWDT Counter0 is configured to generate periodic interrupts at 0.5 second intervals. MCWDT interrupts are used to control the status LEDs and turn them off when not required, to save power.
Digital Output Pin	Pin_LED_Red Pin_LED_Orange	These GPIOs are configured as firmware controlled digital output pins that control status LEDs.
Digital Input Pin	Advertise	This pin is configured as a digital input pin that is used to generate interrupts when the user button (**SW2**) is pressed.
Global Signal Reference	GlobalSignal	The Global Signal Component is configured to extract interrupts from **Advertise** pin.

Note: See the code example CE218133 – PSoC 6 MCU E-INK Display with CapSense for more details on components used by E-INK library. See the PSoC Creator project for more details of PSoC Component configurations and design wide resource settings.

Related Documents

Application Notes	
AN210781 Getting Started with PSoC 6 MCU with Bluetooth Low Energy (BLE) Connectivity	Describes PSoC 63 with Bluetooth Low Energy (BLE) Connectivity and how to build your first PSoC Creator project
PSoC Creator Component Datasheets	
Bluetooth Low Energy	Facilitates designing applications requiring BLE connectivity
CapSense	Provides guidelines to use the CapSense component
Device Documentation	
PSoC 6 MCU: PSoC 63 with BLE Datasheet	PSoC 6 MCU: PSoC 63 with BLE Architecture Technical Reference Manual
Development Kit (DVK) Documentation	
CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit	

Document History

Document Title: CE220186 – PSoC 6 MCU with BLE Connectivity: RTC with Current Time Service

Document Number: 002-20186

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	5840916	NIDH	08/23/2017	Initial public release version
*A	6005873	NIDH	12/13/2017	Updated template and minor text edits. Updated project to PSoC Creator 4.2 Beta.

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