CE96926 - PSoC® Real-Time Clock Based on Power Line Frequency

Objective

These code examples show how to implement a real-time clock (RTC) using the 50/60-Hz frequency of the mains power line as the time base. PSoC 3, PSoC 4, and PSoC 5LP devices are supported. The example for PSoC 4 BLE also uses a 32-kHz watch crystal oscillator (WCO) that can be used as the secondary time-base in case of a mains power failure.

Overview

An RTC provides time and date information – second, minute, hour, day of the week, day of the month, month, and year. The time and date information are usually updated every second based on a one-second interrupt generated from a 32.768-kHz crystal. However, there can be long-term drifts in the time due to factors such as the temperature and aging of the crystal.

An alternative technique is to use the frequency of the mains power line as a time base. Systems that are wall-powered and need a system clock can make use of this technique. This technique is applicable only in countries where mains power frequency adjustment is done.

Refer to the associated application note AN96667, PSoC Real Time Clock Based on Power Line Frequency for the theory behind this technique.

Two code examples are provided:

- RTC with PSoC 5LP using the Power Line
- RTC with PSoC 4 BLE using the Power Line with Crystal Back-Up

Requirements

Tool: PSoC Creator 3.1 SP1 or higher

Programming Language: C (ARM® GCC 4.8.4 and ARM MDK compilers)

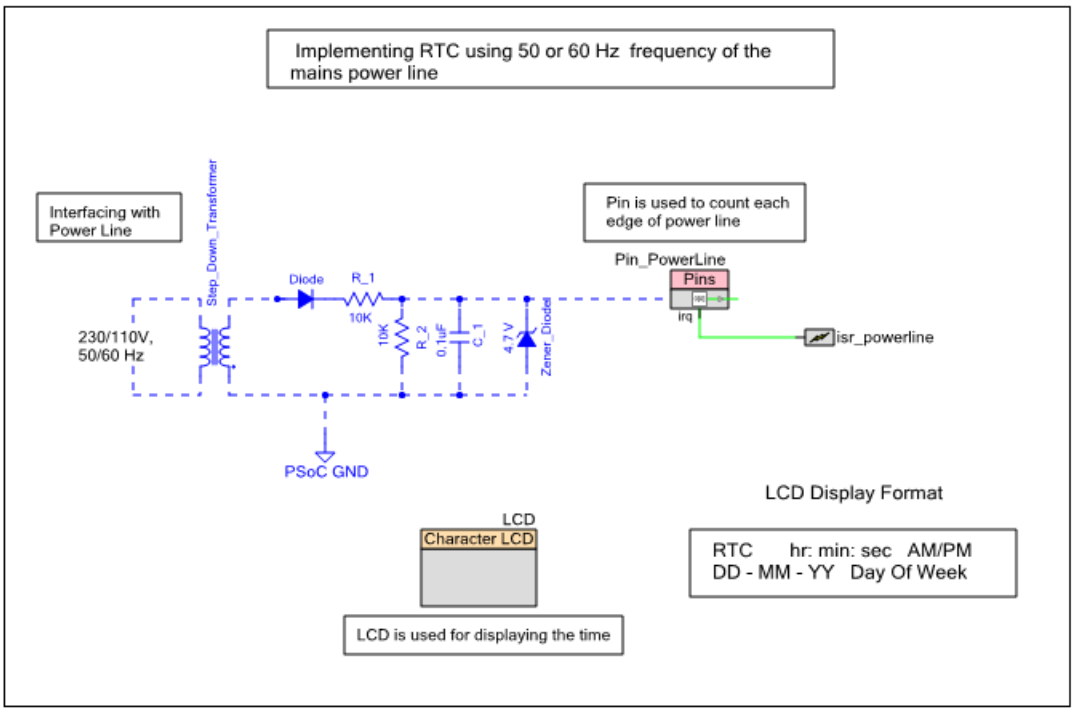
Associated Parts: All PSoC 5LP and PSoC 4 BLE parts

Related Hardware: CY8CKIT-050, CY8CKIT-042-BLE

Design

Figure 1 shows the code example design for PSoC 5LP.

Figure 1. RTC Code Example for PSoC 5LP Device on CY8CKIT-050



The code example features the following:

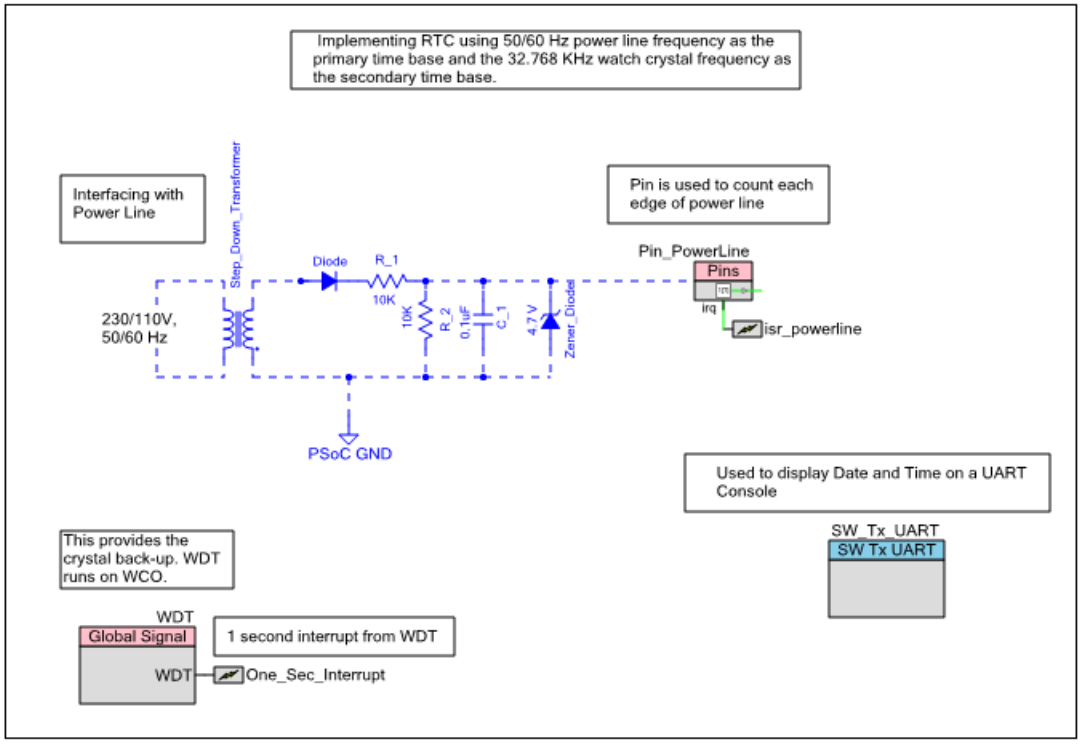
- A one-second event using the interrupt generated from the 50/60-Hz power line
- A character LCD that displays the RTC date and time

Note: This design does not have a crystal or oscillator-based backup to maintain the time in the case of a mains power failure. If the design has a battery backup to power the PSoC device if the mains power fails, then you should implement a crystal-based backup using the existing RTC Component in PSoC Creator.

This design can also be ported to other devices such as PSoC 3, PSoC 4000, PSoC 4100, and PSoC 4200. To implement a crystal backup for the PSoC 4100 and PSoC 4200 series devices, see the code example **CE95915**.

Figure 2 shows the code example design for PSoC 4 BLE.

Figure 2. RTC Code Example for PSoC 4 BLE Devices on CY8CKIT-042-BLE



This code example features the following:

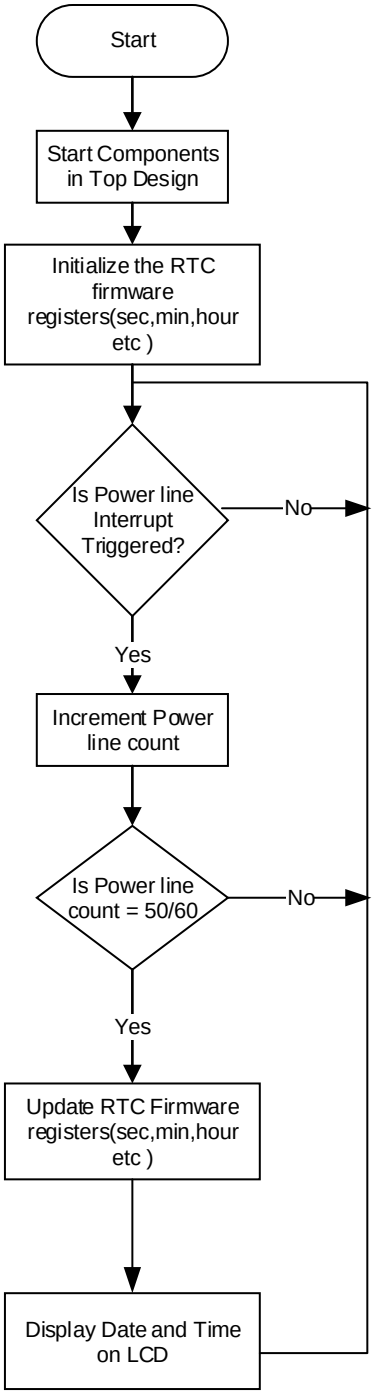
- A one-second event using the interrupt generated from the 50/60-Hz power line
- A watchdog timer (WDT), running on a 32.768-kHz watch crystal oscillator (WCO) that generates a one-second interrupt. This is used as the secondary time base if the main power supply fails.
- Automatic switching between primary and secondary time bases
- UART-based communication with a PC. Because a software-based transmit UART is used, some CPU processing bandwidth is used

Code Design

PSoC 5LP Code Example:

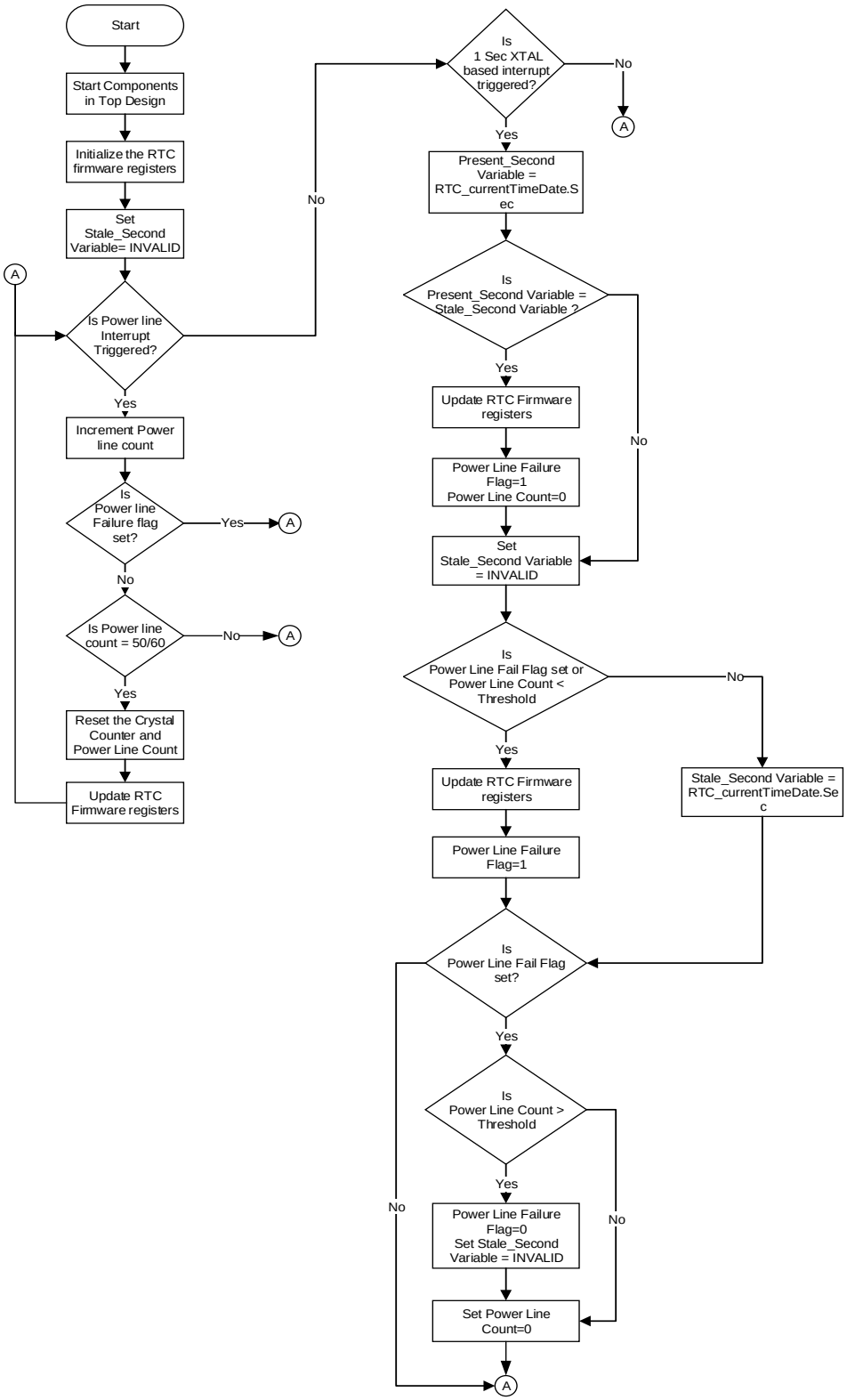
Figure 3 shows the flow chart of the PSoC 5LP code example.

Figure 3. Flow Chart for PSoC 5LP Code Example



PSoC 4 BLE Code Example

Figure 4. Flow Chart for PSoC 4 BLE Code Example



Note: This project can be used in designs that have a battery back up to power the PSoC device in the case of a mains power failure. During a mains power failure, the PSoC 4 BLE device is put to the Deep-Sleep Mode to minimize current consumption. The current consumed by this design in the Deep-Sleep mode is approximately 100 µA.

Design Considerations

Off-kit connections are needed to implement the power-line interface as shown in Figure 5.

The CY8CKIT-050 can be powered using a 12-V AC power adapter. To test the designs, all development kits can be powered using the USB power. Check the “Hardware Setup” section in the user guide for the respective kits to see how the PSoC 4 BLE Development Kit can be powered using a 5-V power adapter.

The PSoC supply voltage in the development kits should be set to 5 V because the power-line interface components include a 4.7-V Zener diode. Also, the wall powering should come directly from the mains power line. The code examples have not been tested with power lines from inverters or UPS systems. This technique is suitable for long-term time accuracy, but not for accurate short time-intervals, for which the direct 32.768-kHz WCO frequency is better suited.

Hardware Setup

- You need the following hardware for the power-line interface, as

5 shows:

- One 12-0-12 step-down transformer (100-240 V, 500 mA)
- One P-N junction diode
- Two 10 kΩ resistors
- One 0.1 µF capacitor
- One 4.7-V Zener Diode

Figure 5. Power-Line Interface Design

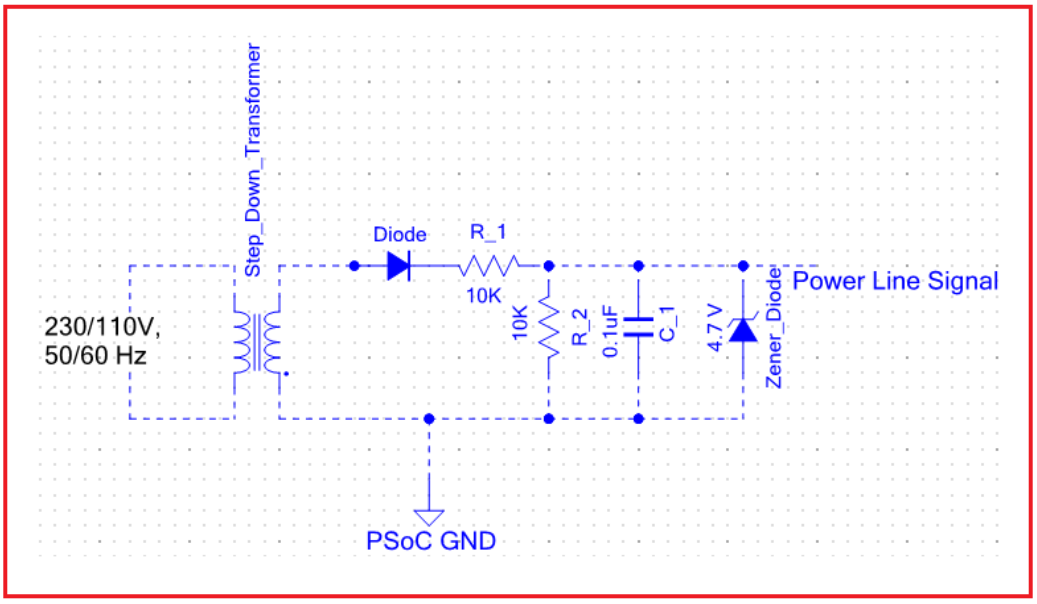
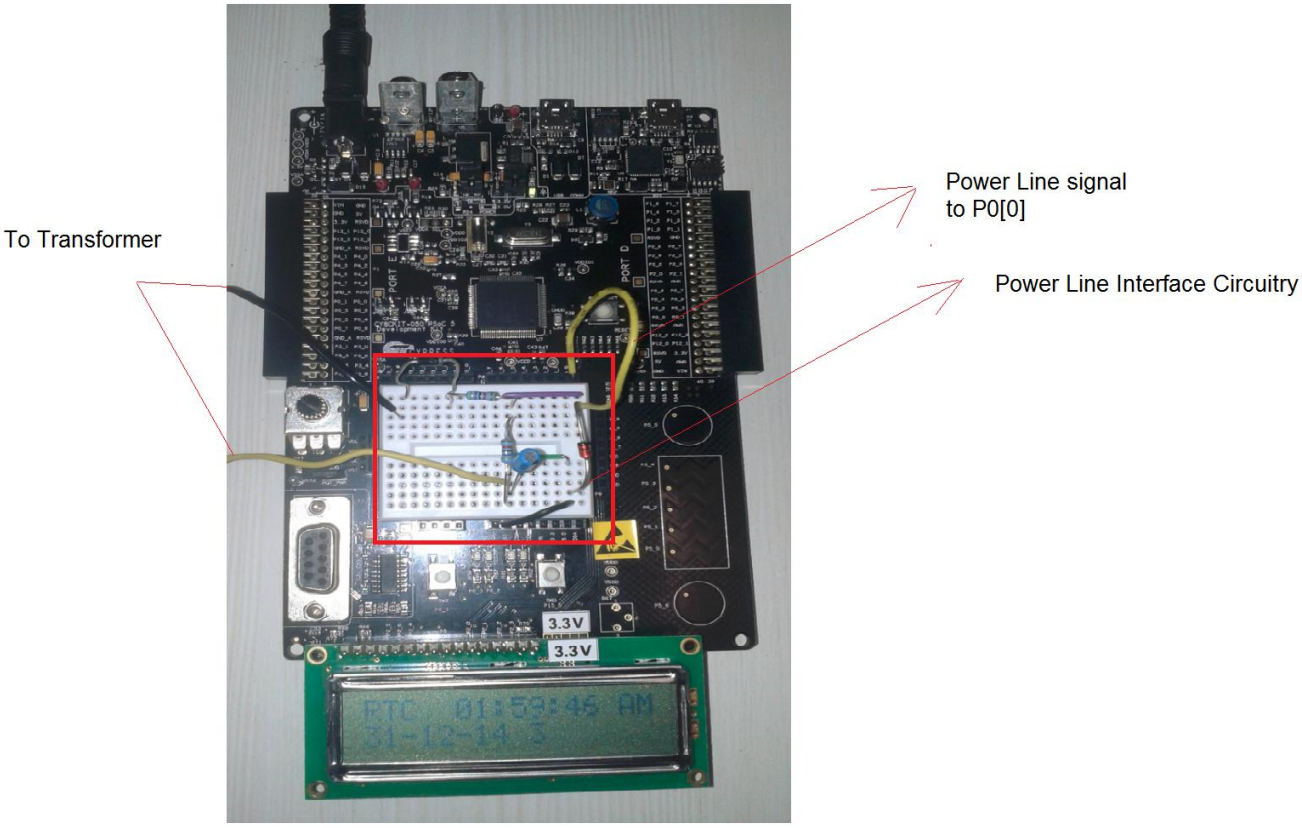


Figure 6 shows the hardware setup required to test the project with the CY8CKIT-050 kit.

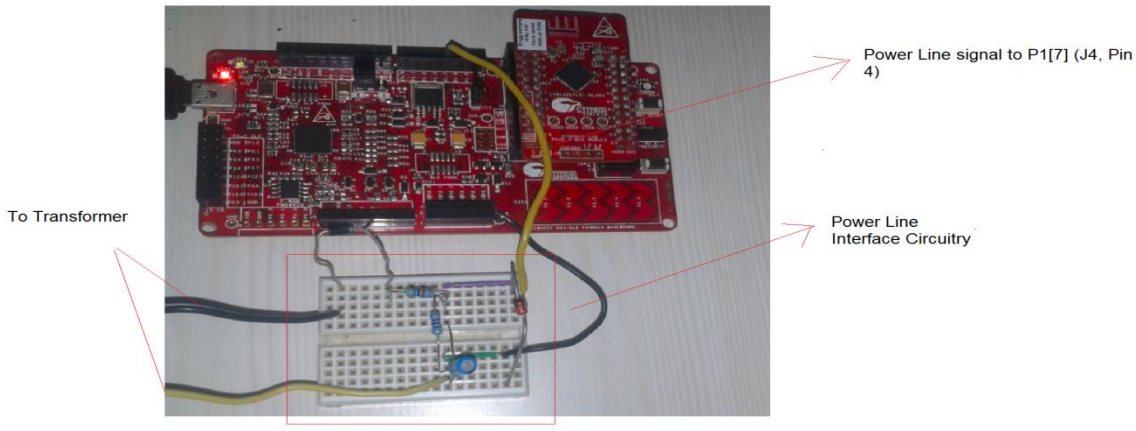
Figure 6. Hardware Setup Diagram for CY8CKIT-050



.

Figure 7 shows the hardware setup required to test the project with the CY8CKIT-042-BLE kit.

Figure 7. Hardware Setup Diagram for CY8CKIT-042-BLE



Make the following connections:

- Connect a wire from the Power Line signal shown in Figure 5 to Pin P0[0] of the CY8CKIT-050 kit. This acts as the Power Line signal to the PSoC GPIO pin.
- Connect a wire from the Power Line signal as shown in Figure 5 to Pin P1[7] of the CY8CKIT-042-BLE (J4, Pin 4 of CY8CKIT-042-BLE). This acts as the Power Line signal to the PSoC GPIO pin. The pin used for transmitting the data over UART (P1[5]) is hardwired to the receiver pin of the PSoC 5LP device in the kit. This enables displaying the time and date on a UART console over the USB-UART Interface.

Note: The CY8CKIT-042-BLE kit can be powered using an external 5-V power adaptor if USB power is not used. To do so, connect the output of the power adaptor to the "VIN" Pin (J1 Header, Pin 1) of the Kit. You can also connect a 5 V Battery to the "BT1" Slot provided at the backside of the Kit. This can be used as the secondary power source if main power fails.

Software Setup

No special software setup is required for this project. All supported compilers can be used with any optimization required.

At the PSoC Creator project's default CPU clock speed (12 MHz for PSoC 4 BLE and 24 MHz for PSoC 5LP), the CPU has enough cycles to support the example. The RTC update ISR uses about 0.03% of the CPU's bandwidth, and the software transmit UART operating at 115,200 baud uses approximately 0.2% of the CPU bandwidth.

Components

Table 1 lists the PSoC Creator Components used in the PSoC 5LP example as well as the hardware resources used by each Component.

Table 1. List of PSoC Creator Components Used by the PSoC 5LP Project

Component	Hardware Resources
Character LCD	1 – uses 7 I/O pins
isr	1 interrupt
Pin	1 I/O pin for the receiving the power line signal

Table 2 lists the PSoC Creator Components used in the PSoC 4 BLE example as well as the hardware resources used by each Component.

Table 2. List of PSoC Creator Components Used by the PSoC 4 BLE Project

Component	Hardware Resources
SW_Tx_UART	None – Software-based UART does not use any hardware resources
isr	2 interrupts
Pins	1 I/O pin for the receiving the power line signal
Global Signal	WDT Int Global Signal

Parameter Settings

Table 3 lists the parameter settings for each PSoC Creator Component used in the PSoC 5LP code example. Only the parameters that vary from the default values are listed.

Table 3. List of PSoC Creator Component Parameter Settings for PSoC 5LP Example

Component	Non-Default Parameter Settings
Pin	Interrupt = Rising Edge Threshold = LVTTL

Table 4 lists the parameter settings for each PSoC Creator Component used in the PSoC 4 BLE code example. Only the parameters that vary from the default values are listed.

Table 4. List of PSoC Creator Component Parameter Settings for PSoC 4 BLE Example

Component	Non-Default Parameter Settings
Pin	Interrupt = Rising Edge Threshold = LVTTL
Global Signal	Global signal name = WDT Int

Design-Wide Resources

Figure 8 shows the pin assignments for the PSoC 5LP code example.

Figure 8. Pin Assignments for PSoC 5LP Example

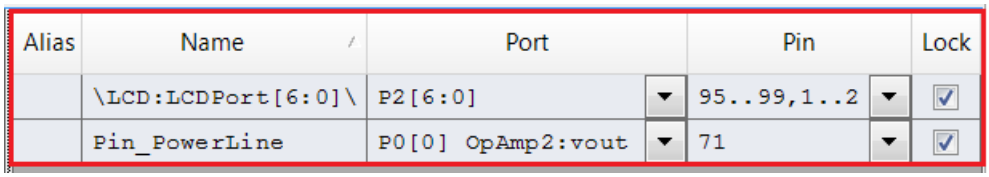


Figure 9 shows the Clock Tree. Enable XTAL 32kHz in the design-wide resource.

Figure 9. PSoC 5LP Clock Tree

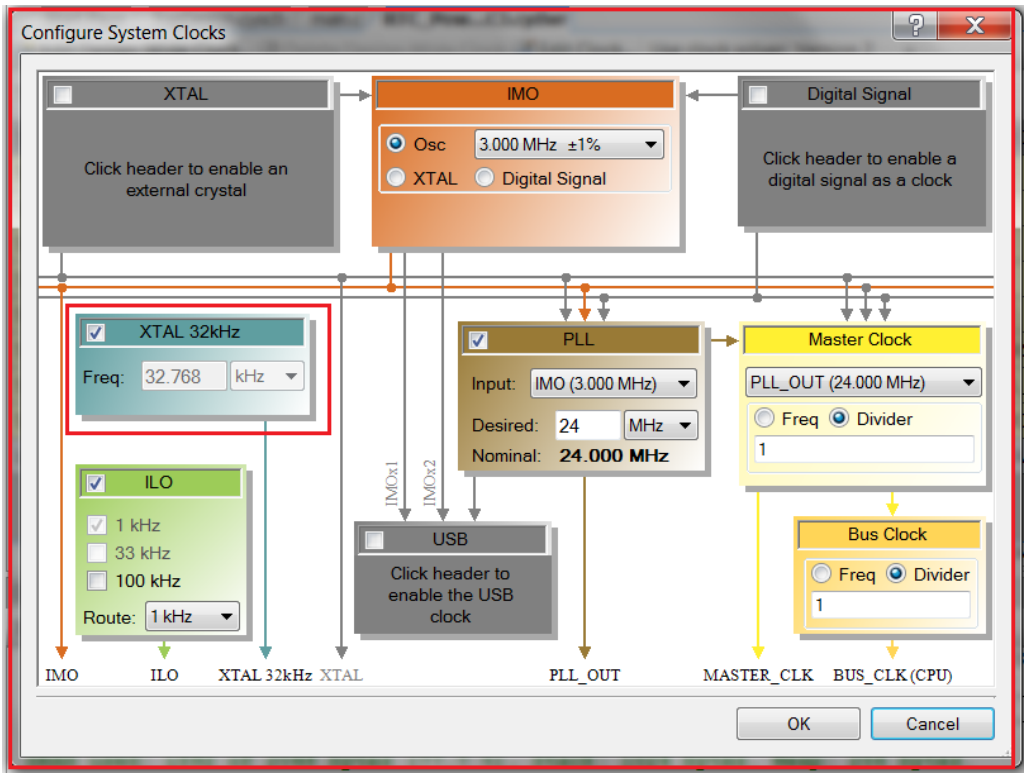
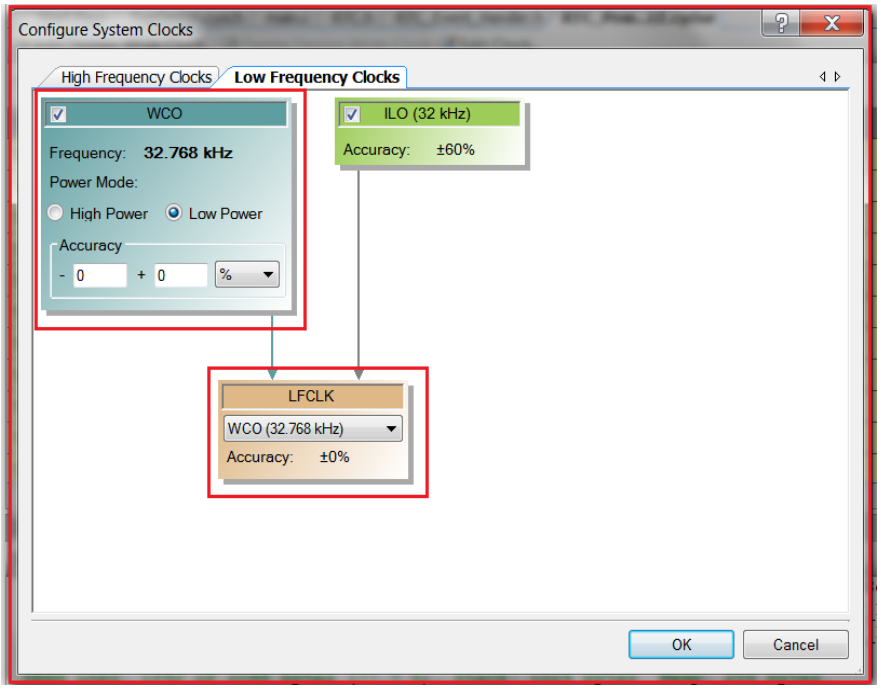


Figure 10 shows the pin assignments for the PSoC 4 BLE code example. Enable the 32.768-kHz WCO in the design-wide resource. Figure 11 shows the Clock Tree.

Figure 10. Pin Assignments for PSoC 4 BLE Example

Alias	Name	Port	Pin	Lock
	\SW_Tx_UART:Tx\	P1[5] OA3:vplus, TCPWM2:line_out_compl, SCB0:uart_tx, SCB0:i2c_scl, SCB0:spi_miso	33	☑
	Pin_PowerLine	P1[7] OA3:vplus_alt, TCPWM3:line_out_compl, SCB0:uart_cts, SCB0:spi_clk	35	☑

Figure 11. PSoC 4 BLE Clock Tree

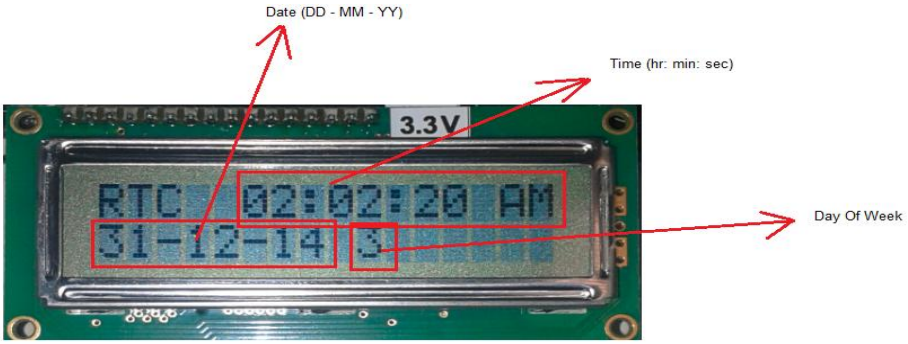


Operation

PSoC 5LP Code Example

1. Set the suitable Power Line frequency for the define POWERLINE_FREQ_HZ in main.c file of the Code Example. The value is 50u by default. Change this to 60u if power line frequency is 60 Hz.
2. Make the necessary hardware connections. See Figure 5.
3. Build the project in PSoC Creator.
4. Program the PSoC 5LP project into the CY8CKIT-050.
5. Observe the date and time getting displayed on the Character LCD.
   Figure 12 shows the output on the Character LCD. The LCD is powered from the 3.3 V On-Board Regulator. Hence it can be safely used even if PSoC is powered at 5V.

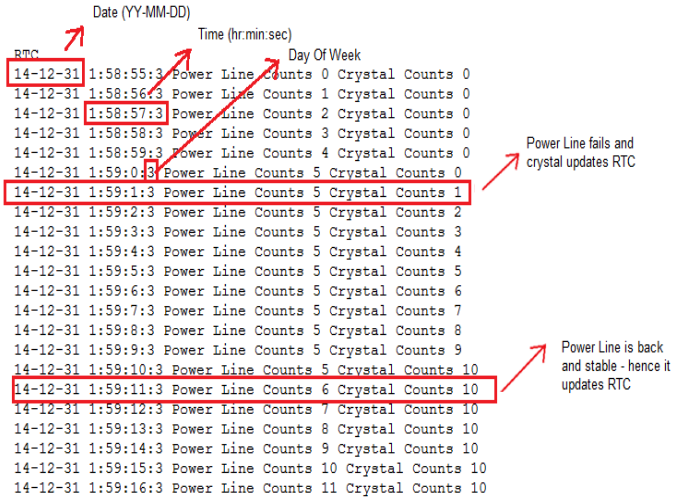
Figure 12. Date and Time Shown in the LCD for PSoC 5LP Code Example



PSoC 4 BLE Code Example

1. Set the suitable Power Line frequency for the define POWERLINE_FREQ_HZ in main.c file of the Code Example. The value is 50u by default. Change this to 60u if power line frequency is 60 Hz.
2. Make the necessary hardware connections. See Figure 5.
3. Build the project in PSoC Creator.
4. Program the project into the CY8C4247LQI-BL483 device, which is the default device in the CY8CKIT-042 BLE Pioneer Kit.
5. Configure any UART console (such as Hyperterminal or Coolterm) for a baud rate of 115,200.
6. Select the COM port that is enumerated as “KitProg USB-UART” in the Device Manager of the PC.
7. Connect the console and reset the device by pressing SW1 (Reset Switch). Verify that the time is updated once per second, as Figure 13 shows. Figure 13 also shows the crystal updating the RTC when the mains power line fails and switching back to the mains power line when it is back and stable).

Figure 13. Time shown in the console for PSoC 4 BLE Code Example



Note: The CY8CKIT-042 BLE kit has an option to insert a coin-cell battery. If the main power supply fails, then the PSoC 4 BLE device can run on this coin cell. While testing the code example, we assume that the kit is always powered with the USB so that we can observe the date and time on the serial console. The power failure can be simulated by removing the connection between the Power Line Signal and the GPIO pin.

Related Documents

Table 5 lists relevant application notes, code examples, knowledge base articles, device datasheets, and Component datasheets.

Table 5. Related Documents

Application Notes		
AN54181	Getting Started with PSoC 3	AN54181 introduces you to PSoC® 3, an 8051-based programmable system-on-chip.
AN77759	Getting Started with PSoC 5LP	AN77759 introduces you to PSoC® 5LP, an ARM® Cortex®-M3-based programmable system-on-chip.
AN54460	PSoC 3 and PSoC 5LP Interrupts	AN54460 explains the interrupt architecture in PSoC® 3 and PSoC 5LP, and its configuration in PSoC Creator™ IDE with the help of code examples.
AN60631	PSoC 3 and PSoC 5LP Clocking Resources	AN60631 describes PSoC 3 and PSoC 5LP's oscillators and clock sources, phase-locked loop (PLL), and clock distribution network.
AN54439	PSoC 3 and PSoC 5LP External Crystal Oscillators	AN54439 describes how to configure hardware and firmware for PSoC 3 or PSoC 5LP using the integrated oscillator subsystems and external crystal or ceramic resonators.
AN79953	Getting Started with PSoC 4	Introduces user to PSoC® 4, an ARM® Cortex™-M0 based programmable system-on-chip.
AN90799	PSoC 4 Interrupts	Explains the interrupt architecture in PSoC 4 and its configuration in PSoC Creator IDE
AN91267	Getting started with PSoC 4 BLE	AN91267 introduces you to PSoC® 4 BLE, an ARM® Cortex®-M0 based Programmable System-on-Chip (PSoC) that integrates a Bluetooth Low Energy (BLE) radio system.
Code Examples		
RtcDesign	Example Project for RTC Implementation in PSoC 3 and PSoC 5LP using the available RTC Component	
CE95915	Example Project for Implementing an RTC with PSoC® 4100 / PSoC 4200 Devices with an external crystal	
Knowledge Base Articles		
Implementing a 32-kHz ECO Interface with PSoC® 4100 / PSoC 4200 – KBA95848	How can we implement a 32.768 kHz ECO interface with PSoC 4100 and PSoC 4200 series devices?	
PSoC Creator Component Datasheets		
Character LCD	Implements the industry standard Hitachi HD44780 LCD display driver chip protocol	
Interrupt	Defines hardware-triggered interrupts	
Software Transmit UART	Provides an 8-bit RS-232 data-format compliant serial transmitter through software bit-banging	
Pins	Controls interface with physical I/O port pins	
Global Signal Reference	Allows access to device level global signals	
Device Documentation		
PSoC 3 Datasheets	PSoC 3 Technical Reference Manuals	
PSoC 5LP Datasheets	PSoC 5LP Technical Reference Manuals	
PSoC 4 Datasheets	PSoC 4 Technical Reference Manuals	
Development Kit (DVK) Documentation		
PSoC 3 Kits		
PSoC 5LP Kits		
PSoC 4 Kits		

Document History

Document Title: CE96926 – PSoC® Real-Time Clock Based on Power Line Frequency

Document Number: 001-96926

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	4709131	BMAH	04/01/2015	New code example
*A	5735367	AESATMP9	05/18/2017	Updated logo and copyright.

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

Cypress Developer Community

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

Technical Support

cypress.com/support

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

©Cypress Semiconductor Corporation, 2015-2017. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.