

## LED Lighting with Remote Control using CC2510 & CC1110

By Jerry Ng

---

### Keywords

- *LED Lighting*
- *Timer*
- *PWM*
- *TPS40211*
- *CC1110Fx*
- *CC1111Fx*
- *CC2510Fx*
- *CC2511Fx*

### 1 Introduction

This note describes a LED lighting design based on CC2510 and TPS40211. CC2510 is 2.4GHz System-on-Chip Low Power RF Transceiver while TPS40211 is a wide input range current mode boost controller, which provides voltage regulation and support 8 LED in series.

CC2510 provides the wireless control function including 16 steps dimmer and various flashing modes. The CC1110 is the sub 1GHz version of the CC2510. Project collateral discussed in this application note can be downloaded from the following URL:  
<http://www.ti.com/lit/SWRA305>.

## Table of Contents

KEYWORDS.....	1
<b>1 INTRODUCTION.....</b>	<b>1</b>
<b>2 ABBREVIATIONS.....</b>	<b>2</b>
<b>3 HARDWARE.....</b>	<b>3</b>
3.1 REMOTE CONTROLLER.....	3
3.2 LED LIGHTING.....	4
<b>4 SOFTWARE.....</b>	<b>5</b>
4.1 IAR WORKSPACE.....	5
4.2 BSP EXTENSION.....	6
4.3 REMOTE CONTROLLER.....	6
4.4 LED LIGHTING.....	7
4.5 FURTHER DEVELOPMENT.....	7
<b>5 SCHEMATIC.....</b>	<b>9</b>
<b>6 REFERENCES.....</b>	<b>10</b>
<b>7 GENERAL INFORMATION.....</b>	<b>11</b>
7.1 DOCUMENT HISTORY.....	11

## 2 Abbreviations

EM	Evaluation module
LED	Light Emitted Diode
PWM	Pulse Width Modulated
SoC	System On Chip
BSP	Board Support Package

## 3 Hardware

LED lighting design contains two parts - the LED lighting board and the remote controller. The LED Lighting board includes the LED driver and the CC2510/CC1110 for control functions. The remote controller detects user inputs and sends commands to the CC2510/CC1110 on the LED lighting board.

### 3.1 Remote Controller

The remote controller uses the existing evaluation board and evaluation module to achieve its function. The SmartRF04EB and CC2510EM are shown in Figure 1 and Figure 2 respectively. SmartRF04EB is a base board for various EM boards. It provides power, interface to other systems and user interface (LCD, LEDs, Buttons and Joystick). By mounting the CC2510EM/CC1110EM to P1 & P2 of SmartRF04EB and loading the proper software, a remote controller for LED lighting is obtained as shown in Figure 3.

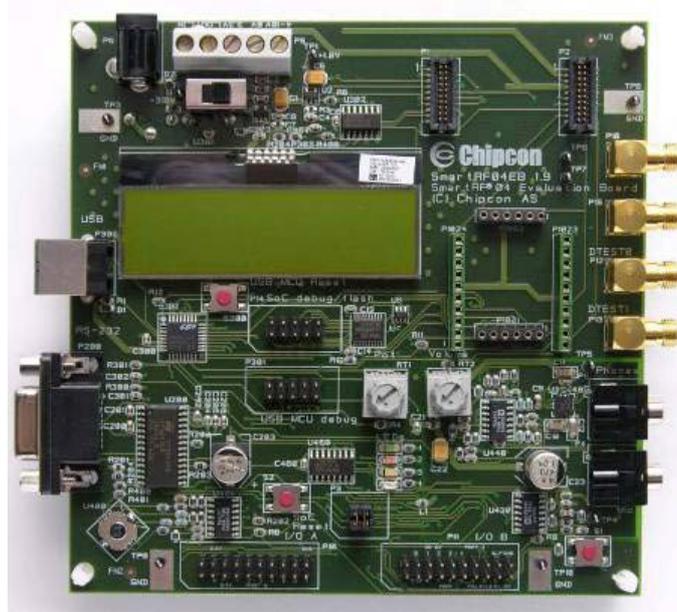


Figure 1. SmartRF04EB



Figure 2. CC2510EM



Figure 3. SmartRF04EB with CC2510EM

## 3.2 LED Lighting

LED lighting board consists of LED board and a CC2510EM/CC1110EM. The CC2510EM/CC1110EM is inserted on the P1 and P2 of LED lighting board (Figure 4).

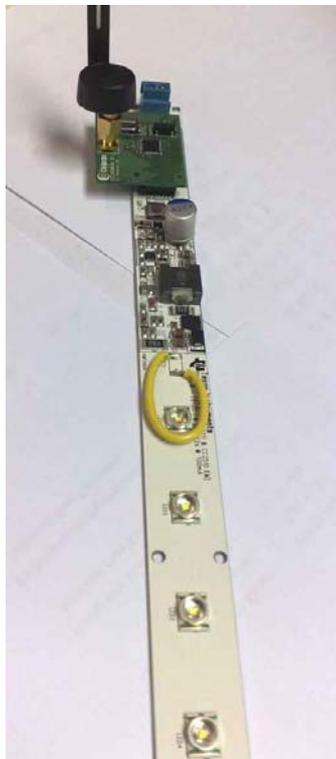


Figure 4. LED lighting board with CC2510EM

## 4 Software

The wireless communication protocol on this system is based on SimpliciTI 1.1.0. SimpliciTI is a simple network protocol aimed at small low data rate, low duty cycle RF networks with limited number of nodes. SimpliciTI is free and can be downloading from TI web. For more details, please refer to TI web below:

<http://focus.ti.com/docs/toolsw/folders/print/simpliciti.html>

The code is based on Simple Peer-to-Peer sample applications in SimpliciTI. The original code can download from TI web and contains documentation and flow chart for this example. IAR version 7.50 was used for the sample code. Currently, commands are sent from the remote controller to LED lighting board with the LED lighting board returning an acknowledge.

### 4.1 IAR Workspace

IAR workspace is located in the following folder:

C:\Texas Instruments\LED\_lighting\ide\led\_lighting.eww

Four workspaces (led\_cc2510, led\_cc1110, srf04\_cc2510, srf04\_cc1110) are included in the package as show in Figure 5. Workspaces led\_cc2510 and srf04\_cc2510 are the LED lighting and remote controller projects for CC2510 respectively. Workspaces led\_cc1110 and srf04\_cc1110 are the LED lighting and remote controller projects for CC1110 respectively. The individual project can be opened directly on the corresponding folder. For example, the led\_cc2510 project can be open in the folder.

C:\Texas Instruments\LED\_lighting\ide\LED\_cc2510\iar\led.eww

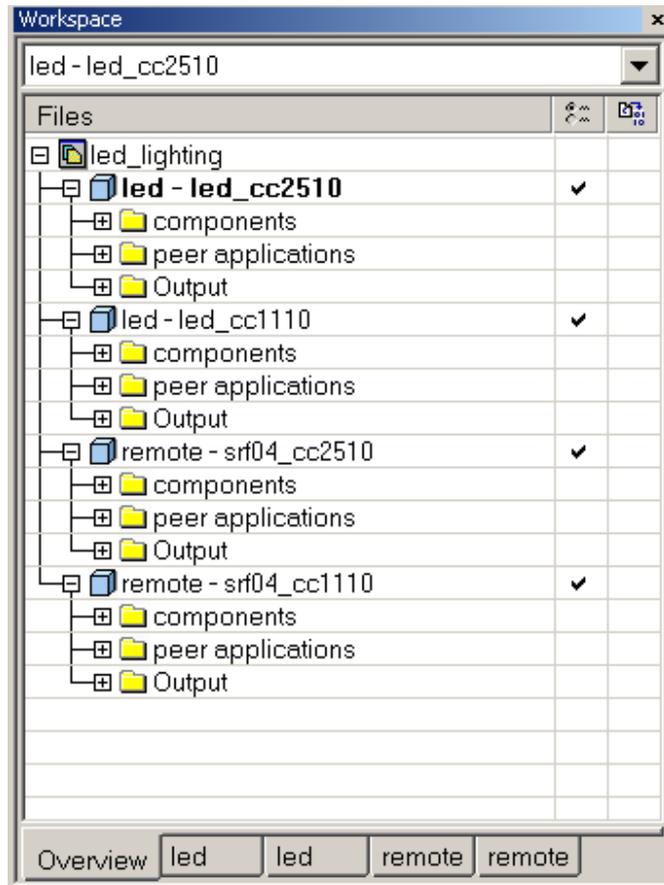


Figure 5. LED lighting IAR workspace

## 4.2 BSP Extension

SimpliciTI structured the board related code (BSP – Board Support Package) in Component -> bsp folder within the workspace. Different BSP will be compiled and linked when using different board (such as SmartRF04EB and SmartRF05EB). To extend the support on LED lighting project, extended package were added to the Component->bsp-> extended folder for this purpose as shown in Figure 6. This folder contains two file bsp\_extended.c and bsp\_extended.h for the corresponding code and header file. In the remote controller workspace, joystick on the SmartRF04EB was used as an input device. The corresponding code and header are put in the extended files within the remote workspace. In the LED workspace, LED lighting board is used instead of SmartRF board. The supporting code and header are put in the extended files within the LED workspace.

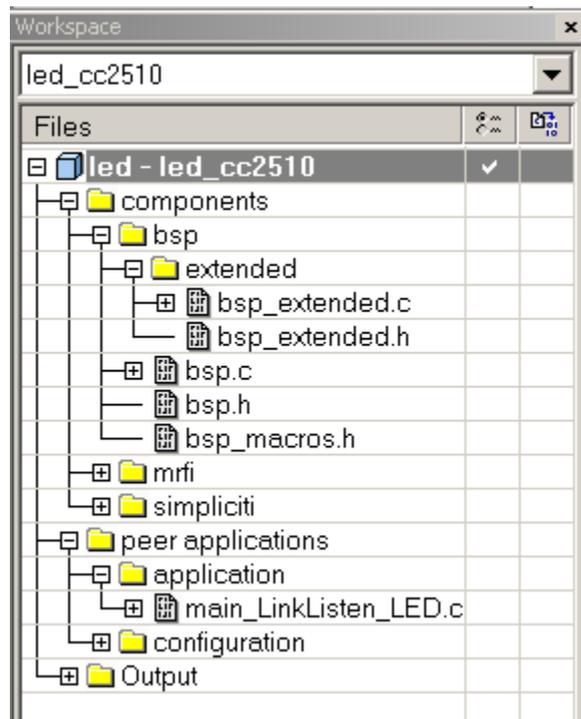


Figure 6. BSP Extended package in led\_cc2510 workspace

## 4.3 Remote Controller

The remote controller project is based on SimpliciTI Simple Peer-to-Peer LinkTo example. Main() is located in the peer applications->application->main\_LinkTo\_Remote.c. The code starts with initializations on the BSP and extended BSP. The extended BSP initialize the joystick functions. Then, it will initialize the SimpliciTI stack. To build up the link between remote and LED, button S1 has to be pushed so that remote can send LinkTo command to the LinkListen device. After link successfully, the program detects the user inputs and send the corresponding command to LED.

Joystick code is located in bsp\_extended.c. The detection of joystick is based on the ADC value from I/O port 0 pin 6. Details of Smart04EB hardware can be found on the Development Kit User Manual (SWRU134A). The usage of ADC for CC2510 and CC1110 can be found on datasheet (SWRS055 and SWRS033).

Details of each function are listed below:

- Button S1: Toggle ON/OFF
- Joystick Left/Right: Dimming control (16 steps)
- Joystick Up/Down: Flashing toggle in the following mode.
- i) Slow flashing – 1.5 sec on and 1.5 sec off.
  - ii) Fast flashing – 0.5 sec on and 0.5 sec off
  - iii) Slow flashing for 6 sec then Fast flashing for 4 sec
  - iv) Fast flashing for 2 sec then pause for 2 sec
  - v) Fast flashing for 3 sec at 10/10, 8/10, 6/10, 4/10, 2/10 duty. Total cycling time is 15 sec.

## 4.4 LED Lighting

The LED lighting project is based on SimpliciTI Simple Peer-to-Peer LinkListen example. Main() is located in the peer applications->application->main\_LinkListen\_LED.c. The code starts with initializations on the BSP and PWM. It then initializes the SimpliciTI stack. To build up the link between LED and remote, button S1 has to be pushed so that LinkTo command from remote can be accepted. After link successfully, the program will wait for commands received from remote. The call back function sRxCallback() will be called when message received from remote. It in turn passes the message to BSP\_toggleLED() for controlling the LED functions including toggle on/off, dimmer and flashing. Function BSP\_toggleLED() is located in Component->bsp->extended->bsp\_extended.c within the LED workspace.

Two pins (DIS and PWM) are used to control the LED lighting as shown in the LED board schematic Figure 7. DIS is connected to CCxx10 I/O Port 0 pin 0 while PWM is connected to I/O port 1 pin 1. DIS is an active low signal controlling the on/off and flashing of the LED lighting. PWM is the pulse width modulated signal supporting the dimmer functions. Dimmer function is implemented by using Timer 1 in PWM mode while Timer 4 is used for flashing function. It can highly reduce the complexity of the control code and the processing time on software by using the timer peripheral.

Function BSP\_toggleLED() controls the LED lighting according to the message received. It controls the on/off, dimmer and flashing. When on/off message received, the port 1 pin 1 (DIS) is clear/set accordingly while timer 1 and timer 4 need to be reset. Note that PWM pin is directly controlled by timer 1. The pin is reset by clear the timer 1 counter and run the timer for a short period. The reset of the PWM pin will then trigger by zero count on timer 1. Timer 4 is disable or stopped when on/off message received. Additionally, the variable sLEDstate (bit 0-3: LED dimmer level, bit 4-6: LED flash mode, bit 7: LED on/off) is reset which store the current state of the LED.

When dimmer message is received, the dimmer level is up or down accordingly and the LED is forced on. The dimmer level is controlled by setting the timer 1 counter which in turn sets the pulse width of the PWM signal. Please refer to DN115 Using PWM for Dimmer Function in LED Lighting (SWRU227) for details on setting timer in PWM mode.

When flashing message is received, the flashing mode is up/down accordingly and is stored in the variable sLEDstate. The LED is forced on and timer 4 is enabled and started to run. The flashing management is done in the timer 4 interrupt routine bsp\_Timer4ISR. The period of timer 4 interrupt is set to around 2.5ms. To support different flashing modes, variables (sFlashCount1, sFlashCount2) are used for counting purpose. For example, to implement flash mode 2 (0.5s on then 0.5s off), sFlashCount1 is count from 0 to 400. When sFlashCount1 reaches 200 ( $2.5\text{ms} \times 200 = 0.5\text{s}$ ), LED is turn off. When sFlashCount1 reaches 400 ( $2.5\text{ms} \times 400 = 1\text{s}$ ), LED is turn on and sFlashCount1 is reset to 0.

## 4.5 Further Development

# ***Application Note AN080***

Suggestions for further development are listed below:

- By using the timer peripheral for dimmer function, the number of dimmer level is only limited by the resolution of the timer. CCxx10 has one 16-bit timer and three 8-bit timers. Thus, in theory there is 65535 and 255 level for 16-bit and 8-bit timer respectively. Although one may not be able to distinguish the difference in brightness. On the other hand, one can implement a smooth transition among dimmer levels with further divide each step in 4-8 time. Instead of go from one step to another, the brightness can be changed slowly in 4-8 step before go to the final brightness by setting the timer counter.
- The flashing code is also structured in a flexible way. One can implement its own flash mode by using the sFlashCount as a general counter. It can also combine with dimmer together. Flashing mode 5 is the example for combining flashing and dimmer. Additionally, CCxx10 have one 16-bit timer, two 8-bit timers (can run in PWM mode) and 21 GPIO. One can use in wireless signage applications with maximum three parts the signage has dimmer capabilities while the GPIO can control the on/off functions.
- The LED lighting project is based on SimpliciTI simple peer-to-peer. One can use different network configuration for different applications. There are three more sample applications in SimpliciTI including polling with AP, cascading end device and AP as a data Hub. They can be used as a starting point for other applications.



# ***Application Note AN080***

## **6 References**

- [1] CC2510Fx/CC2511Fx Data Sheet (SWRS055)
- [2] CC1110DK/CC2510DK -- Development Kit User Manual (Rev. A)(SWRU134A)

# ***Application Note AN080***

## **7 General Information**

### **7.1 Document History**

<b>Revision</b>	<b>Date</b>	<b>Description/Changes</b>
SWRA305	2009.09.15	Initial release.

## IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to TI's Terms of Sale ([www.ti.com/legal/termsofsale.html](http://www.ti.com/legal/termsofsale.html)) or other applicable terms available either on [ti.com](http://ti.com) or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2019, Texas Instruments Incorporated