

TI Designs: TIDA-050013

バッテリー駆動アプリケーション用の超小型RGB LEDアニメーション・ソリューションのリファレンス・デザイン

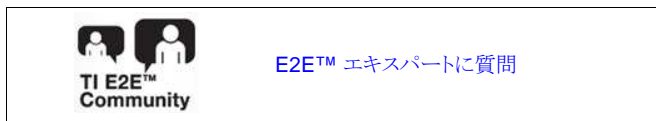


概要

このリファレンス・デザインは、LP5523およびMSP430FR2111デバイスを使用する、バッテリー駆動アプリケーション用の2cm×2cmのRGB LEDアニメーション・ソリューションです。LP5523デバイスはLEDドライバで、プログラム可能な制御エンジンを含み、DSBGAパッケージに搭載されています。LP5523により、非常に小型のPCBに収納しなければならないアプリケーションでも、鮮明な照明パターンが可能になります。このデザインは、制限されたMCUリソースを活用して完全な機能を実現できます。

リソース

- TIDA-050013 デザイン・フォルダ
- LP5523 プロダクト・フォルダ
- MSP430FR2111 プロダクト・フォルダ

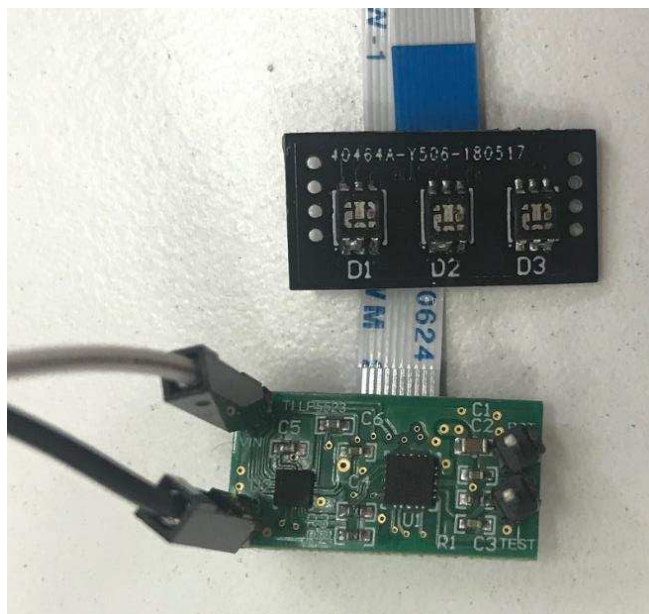
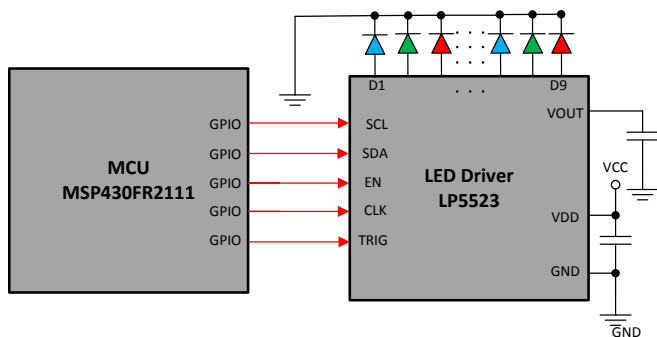


特長

- プリセット照明パターン実行時のマイコン負荷が最小
- LP5523エンジン制御によりシステム消費電流を低減
- スタンバイ・モードでの低い静止電流
- 包括的な診断および保護機能
- 2cm²の小さなPCBサイズに収まるデザイン

アプリケーション

- 電子タバコ
- スマートウォッチ
- 仮想および拡張現実用のヘッドセットおよび眼鏡
- ウェアラブル・カメラ
- スマートIDタグ
- マウス
- 携帯電話



使用許可、知的財産、その他免責事項は、最終ページにあるIMPORTANT NOTICE (重要な注意事項)をご参照くださいますようお願いいたします。

1 System Description

Mobile phones, smart watches, smart ID tags, and e-cigarettes are compact-sized products that benefit from vivid LED animation to catch the consumer's eye. This reference design is an ultra-small RGB LED animation solution for battery-powered applications that makes vivid lighting patterns on a 2-cm² PCB board. The design uses a minimal MCU resource to keep the target design within the limited PCB size. The LP5523 device, a 9-channel RGB or white-LED driver with internal program memory and integrated charge pump, is used to drive 3 RGB LED modules with constant-current control and to run the various lighting patterns autonomously. In this reference design the MSP430FR2111 device loads the engine code to the LP5523 device after power on.

1.1 Key System Specifications

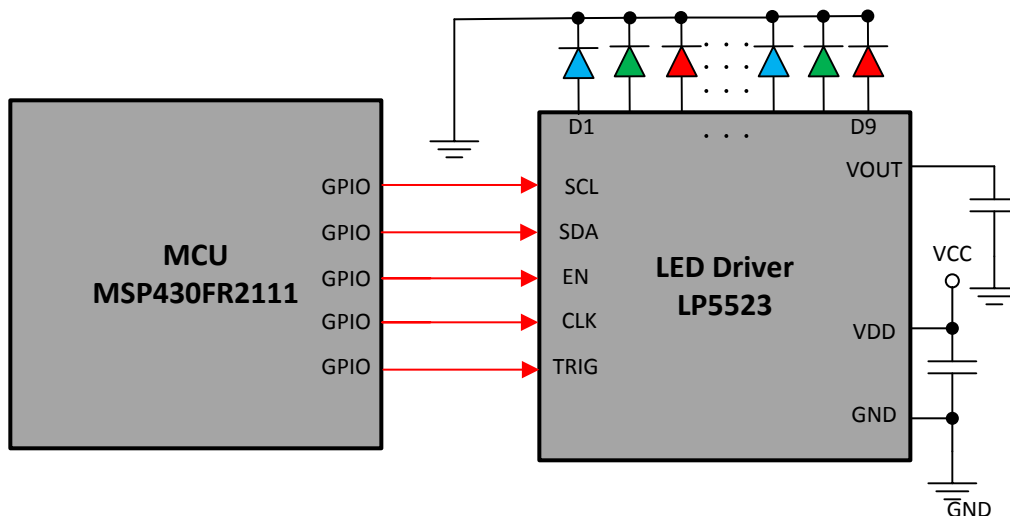
表 1. Key System Specifications

PARAMETER	SPECIFICATIONS
Input voltage range	2.7 V to 5.5 V
Output current	17.5 mA/channel
LED number	3 RGB modules
LED type	ASMT-YTC2-0AA02
Lighting pattern style	3

2 System Overview

2.1 Block Diagram

図 1. TIDA-050013 Block Diagram



2.2 Design Considerations

In this reference design, the MSP430FR2111 device loads the engine code to LP5523 device after power on. The LP5523 device, a 9-channel RGB or white-LED driver with internal program memory and integrated charge pump, is used to drive 3 RGB LED modules with constant-current control and to run the various lighting patterns autonomously. The MSP430FR2111 MCU communicates with the LP5523 device through the I²C protocol which is simulated by 2 GPIOs. The LP5523 device only needs a minimum number of components. The SCL and SDA lines must each have a pullup resistor placed somewhere on the line. In typical applications, values of 1.8 k Ω to 4.7 k Ω , with 4.7 k Ω used here. Other values can be used, depending on the bus capacitance, I/O voltage, and the desired communication speed. For the VOUT and VDD capacitors, see [AN-2227 LP5523 Evaluation Kit](#) for the process of obtaining the selected value, which is 1 μ F.

2.3 Highlighted Products

The following highlighted products are used in this reference design. The key features for selecting the devices for this reference design are outlined in the following subsections. For the complete details of the highlighted devices, see their respective product data sheets.

2.3.1 LP5523 Nine-Channel RGB- and White-LED Driver With Internal Program Memory and Integrated Charge Pump – DSBGA Package

The LP5523 device is a 9-channel LED driver designed to produce lighting effects for mobile devices. A high-efficiency charge pump enables LED driving over the full Lithium-ion battery voltage range. The device is equipped with an internal program memory, which allows operation without processor control.

The LP5523 device maintains excellent efficiency over a wide operating range by autonomously selecting the best charge-pump gain based on LED forward voltage requirements. The LP5523 device is able to automatically enter power-save mode when LED outputs are not active, thus lowering idle current consumption down to 10 μ A (typical).

The LP5523 device has an I²C-compatible control interface with four pin-selectable addresses. The device has a flexible general-purpose output (GPO), which can be used as a digital control pin for other devices. The INT pin can be used to notify the processor when a lighting sequence has ended (interrupt function). Also, the device has a trigger input interface, which allows synchronization, for example, between multiple LP5523 devices.

The device requires only four small, low-cost ceramic capacitors. The LP5523 device is available in a tiny 25-pin DSBGA package (0.4-mm pitch).

2.3.2 MSP430FR2111 16-MHz Ultra-low-Power Microcontroller With 4 KB FRAM, 1 KB SRAM, 12 IOs, and 8-Ch 10-Bit ADC

The MSP430FR2000 and MSP430FR21xx devices are part of the MSP430™ microcontroller (MCU) value-line sensing portfolio. This ultra-low-power, low-cost MCU family offers memory sizes from 0.5 KB to 4 KB of FRAM unified memory with several package options including a small 3-mm \times 3-mm VQFN package. The architecture, FRAM, and integrated peripherals, combined with extensive low-power modes, are optimized to achieve extended battery life in portable, battery-powered sensing applications. The MSP430FR2000 and MSP430FR21xx devices offer a migration path for 8-bit designs to gain additional features and functionality from peripheral integration and the data-logging and low-power benefits of FRAM.

Additionally, existing designs using MSP430G2x MCUs can migrate to the MSP430FR2000 and MSP430F21xx family to increase performance and get the benefits of FRAM. The MSP430FR2000 and MSP430FR21xx MCUs feature a powerful 16-bit RISC CPU, 16-bit registers, and a constant generator that contribute to maximum code efficiency. The digitally controlled oscillator (DCO) also allows the device to wake up from low-power modes to active mode typically in less than 10 μ s. The feature set of this MCU can meet the needs of many applications ranging from appliance battery packs and battery monitoring to smoke detectors and fitness accessories.

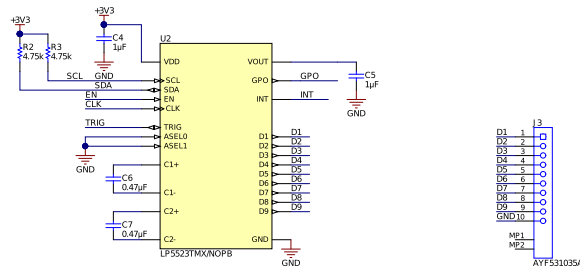
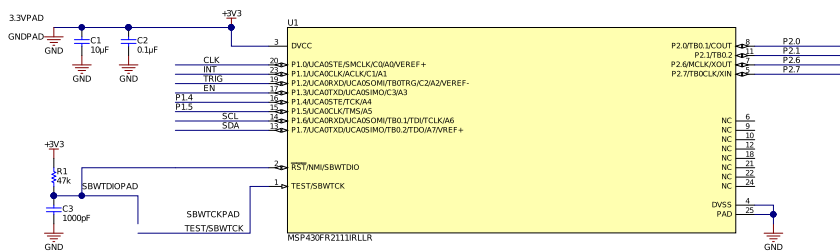
2.4 System Design Theory

One LP5523 device is used to drive 3 RGB LED modules with constant-current control. The MSP430FR2111 device loads the engine code to the LP5523 device after power on and enters the power-save mode. Then the LP5523 device runs the various lighting patterns autonomously.

2.4.1 System Schematics

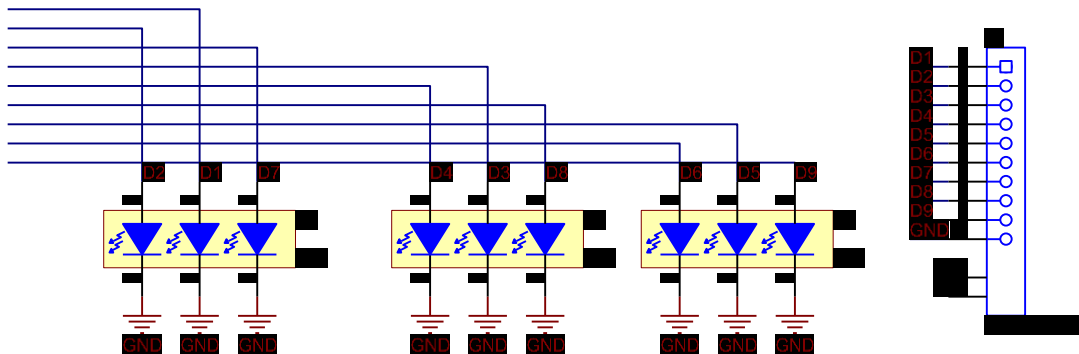
Figure 2 and Figure 3 show the schematics for the LED driver board and LED load board, respectively.

Figure 2. LED Driver Board Schematic



Copyright © 2018, Texas Instruments Incorporated

Figure 3. LED Load Board Schematic



2.4.2 Detailed Design Procedure

To begin the design process, one must decide on a few parameters. The designer must know the following:

- Current of each LED
- LED board layout map
- Lighting pattern

2.4.2.1 LED Current

The maximum current of each LED is 25.5 mA.

2.4.2.2 LED Board Layout Map

Three RGB LED modules are placed in a straight line on the PCB board.

2.4.2.3 Lighting Pattern Design

Firstly, define the LED mapping in the beginning of the engine coding as shown below.

```
; This is a demo for the multicolor breathing. ; This demo to displays a slow ramp of ; all 3
color LED ENG1: dw 0000000100010001b ; ENG2: dw 0000000010001100b ; ENG3: dw 0000000001100010b ;
row1: dw 0000000001001001b ;Map B LED = D1,D4,D7 on the eval. board. row2: dw 0000000010010010b
;Map G LED = D2,D5,D8 on the eval. board. row3: dw 0000000100100100b ;Map R LED = D3,D6,D9 on the
eval. board. row4: dw 0000000011011011b ;Map B LED = D1,D2 on the eval. board. row5: dw
0000000110110110b ;Map G LED = D2,D3 on the eval. board. row6: dw 0000000101101101b ;Map R LED =
D1,D3 on the eval. board. row7: dw 0000000111111111b ;Map all LEDs on the eval. board.
```

2.4.2.3.1 Breathing

During the breathing pattern, all LEDs fade in and out as the same color at the same rate. The sample code is as follows:

```
.segment program1 mux_map_addr row7 set_pwm 0 loop1_0: mux_map_start row1 ;Map the first LED.
mux_ld_end row7 ;End address of the mapping data table. loop1: ramp 2, 200 ;Increase PWM 0-
>PWM200 in 2 second. ramp 2, -255 ;Decrease PWM 200-
>0% in 2 seconds. wait 0.4 ;Wait for 0.4 seconds. ramp 2, 200 ;Increase PWM 0-
>PWM200 in 2 second. ramp 2, -255 ;Decrease PWM 200-
>0% in 2 seconds. wait 0.4 ;Wait for 0.4 seconds. mux_map_next ;Set the next row active in the
mapping table. branch 6,loop1 ;Loop 6 time mux_map_addr row1 ramp 1.5, 200 ;Increase PWM 0-
>PWM200 in 1.5 second. wait 0.4 ;Wait for 0.4 seconds. wait 0.4 ;Wait for 0.4 seconds.
mux_map_addr row2 ramp 3, 200 ;Increase PWM 0->PWM200 in 3 second. ramp 3, -255 ;Decrease PWM 200-
>0% in 3 seconds. ramp 3, 200 ;Increase PWM 0->PWM200 in 3 second. ramp 3, -255 ;Decrease PWM 200-
>0% in 3 seconds. mux_map_addr row1 ramp 1.5, -255 ;Decrease PWM 100->0% in 1.5 seconds. mux_clr
```

2.4.2.3.2 Mixer

This is sample code to display a slow ramp of all 3 colors for each tri-color LED.

```
trigger s{2|3} loop2: mux_map_addr ENG1 ramp 2, 255 ramp 2, -255 ramp 2, -
255 branch 6, loop2 rst .segment program2 trigger w{1} loop3: mux_map_addr ENG2 ramp 2, -
255 ramp 2, 255 ramp 2, -
255 branch 6, loop3 rst .segment program3 trigger w{1} loop4: mux_map_addr ENG3 ramp 2, -
255 ramp 2, -255 ramp 2, 255 branch 6, loop4 rst
```

2.4.2.3.3 Other

For instructions on how to design your own custom lighting pattern, see [LP5523 Nine-Channel RGB- and White-LED Driver With Internal Program Memory and Integrated Charge Pump – DSBGA Package](#).

2.4.3 Uploading The Program to SRAM

The compile tool (Lasm.exe) can be downloaded from the [LP5523 Evaluation Software](#) package on the [product folder](#). The command window or GUI can be used to compile the .scr file. After the compiling, a .hex file appears in the same folder as the .scr file.

```
01 09 00 64 00 92 00 15 00 2A 01 C0 00 3F 01 EA 01 D5 01 FF 9F 89 40 00 9C 03 9C 89 28 C8 21 FF
74 00 28 C8 21 FF 74 00 9D 80 A3 04 9F 83 1E C8 74 00 74 00 9F 84 3E C8 31 FF 3E C8 31 FF 9F 83
19 FF 9D 00 E0 0C 9F 80 20 FF 21 FF 21 FF A3 19 00 00 E0 80 9F 81 21 FF 20 FF 21 FF A3 01 00 00
E0 80 9F 82 21 FF 21 FF 20 FF A3 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 @
0A program1 @ 29 program2 @ 30 program3
```

Then, copy the .hex file to the array table and upload the data to the SRAM by the below coding.

```
uint8_t WR_ENGINE_LOAD[2] = {0x01,LP5523_LOAD_ENG1|LP5523_LOAD_ENG2|LP5523_LOAD_ENG3}; uint8_t
WR_ENGINE_STARTADD[4] = {0x4C,0x0A,0x29,0x30}; uint8_t WR_ENGINE_SRAMP1D[2] = {0x4f,0x00};
uint8_t WR_ENGINE_SRAMP2D[2] = {0x4f,0x01}; uint8_t WR_ENGINE_SRAMP3D[2] = {0x4f,0x02}; uint8_t
WR_ENGINE_SRAMP4D[2] = {0x4f,0x03}; uint8_t WR_ENGINE_SRAMP5D[2] = {0x4f,0x04}; uint8_t
WR_ENGINE_SRAMP6D[2] = {0x4f,0x05}; uint8_t WR_ENGINE_SRAMP1[33] =
{0x50,0x01,0x09,0x00,0x64,0x00,0x92,0x00,0x15,0x00,0x2A,0x01,0xC0,0x00,0x3F,0x01,0xEA,0x01,0xD5,0x01,
0x00,0xFF,0x9F,0x89,0x40,0x00,0x9C,0x03,0x9C,0x89,0x28,0xC8,0x21,0xFF}; uint8_t
WR_ENGINE_SRAMP2[33] =
{0x50,0x74,0x00,0x28,0xC8,0x21,0xFF,0x74,0x00,0x9D,0x80,0xA3,0x04,0x9F,0x83,0x1E,0xC8,0x74,0x00,0x
74,0x00,0x9F,0x84,0x3E,0xC8,0x31,0xFF,0x3E,0xC8,0x31,0xFF,0x9F,0x83}; uint8_t
WR_ENGINE_SRAMP3[33] =
{0x50,0x19,0xFF,0x9D,0x00,0xE0,0x0C,0x9F,0x80,0x20,0xFF,0x21,0xFF,0x21,0xFF,0xA3,0x19,0x00,0x00,0x
E0,0x80,0x9F,0x81,0x21,0xFF,0x20,0xFF,0x21,0xFF,0xA3,0x01,0x00,0x00}; uint8_t
WR_ENGINE_SRAMP4[33] =
{0x50,0xE0,0x80,0x9F,0x82,0x21,0xFF,0x21,0xFF,0x20,0xFF,0xA3,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}; uint8_t
WR_ENGINE_SRAMP5[33] =
{0x50,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}; uint8_t
WR_ENGINE_SRAMP6[33] =
{0x50,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}; uint8_t WR_ENGINE_HOLD[2] =
{0x01,0x00|LP5523_ENABLE}; uint8_t WR_ENGINE_RUN[2] = {0x01,0x2A}; uint8_t WR_ENGINE_EXEC[2] =
{0x00,0x2A|LP5523_ENABLE};
```

3 Hardware, Software, Testing Requirements, and Test Results

3.1 Required Hardware and Software

The hardware boards include one LED driver board and one LED load board, and the two boards are connected with the flexible flat cable.

3.1.1 Hardware

The following hardware is required for the testing.

- A personal computer with Windows® OS
- A 3.3-V dc power supply
- An MSP430RF2311 LaunchPad™

After downloading the firmware from the [TIDA-0510013 product folder](#), connect a 3.3-V dc supply to the LED board input connector (VIN pad). Then use the USB port to load the firmware to the MSP430RF2111 device with the MSP430RF2311 LaunchPad development kit through the Code Composer Studio™ integrated development environment (IDE). Then the LED driver and LED load boards can run the pattern.

3.1.2 Software

Install the [Code Composer Studio \(IDE\)](#) in the PC.

3.2 Testing and Results

3.2.1 Test Setup

Download the sample code at TIDA-050013. Open the project in the Code Composer Studio IDE and download to the MSP430RF2111 device.

Connect the MSP430RF2311 LaunchPad development kit with the LED driver board as shown in [Figure 4](#) and [Figure 5](#). Connect RST (MSP430RF2111 pin 2 on the LED driver board) with red wire to SRWTDI0 (MSP430RF2311 LaunchPad development kit). Connect TEST (MSP430RF2111 pin 1 on the LED driver board) with orange wire to SRWTCK (MSP430RF2311 LaunchPad development kit). Connect the two board GNDs together with blue wire and connect the two board VCCs (3.3 V) together with black wire.

Figure 4. Wire Connections on The LED Driver Board

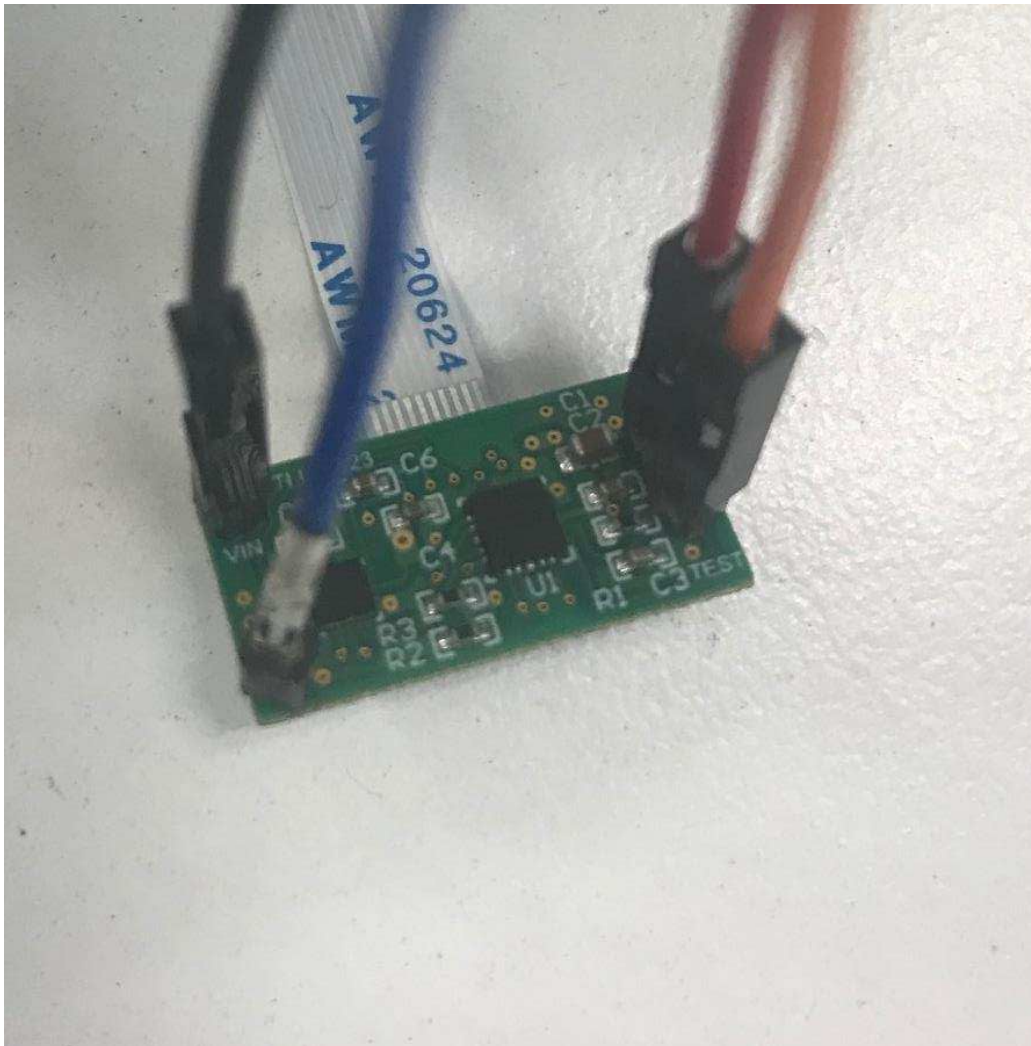
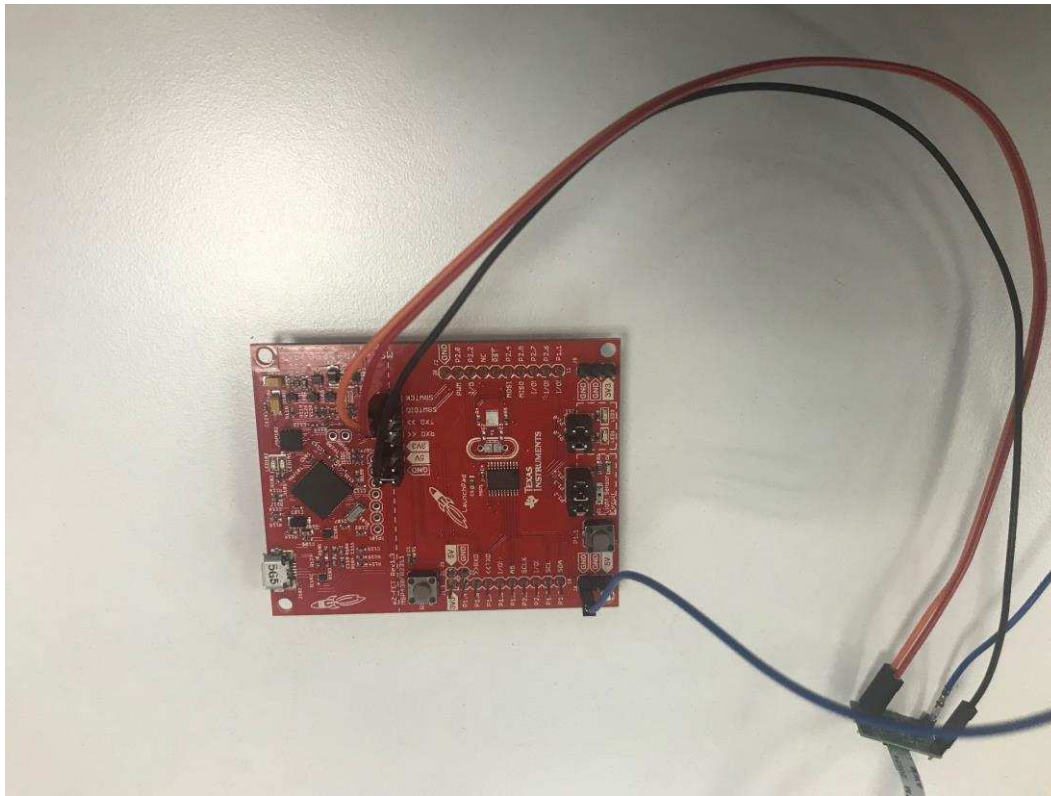


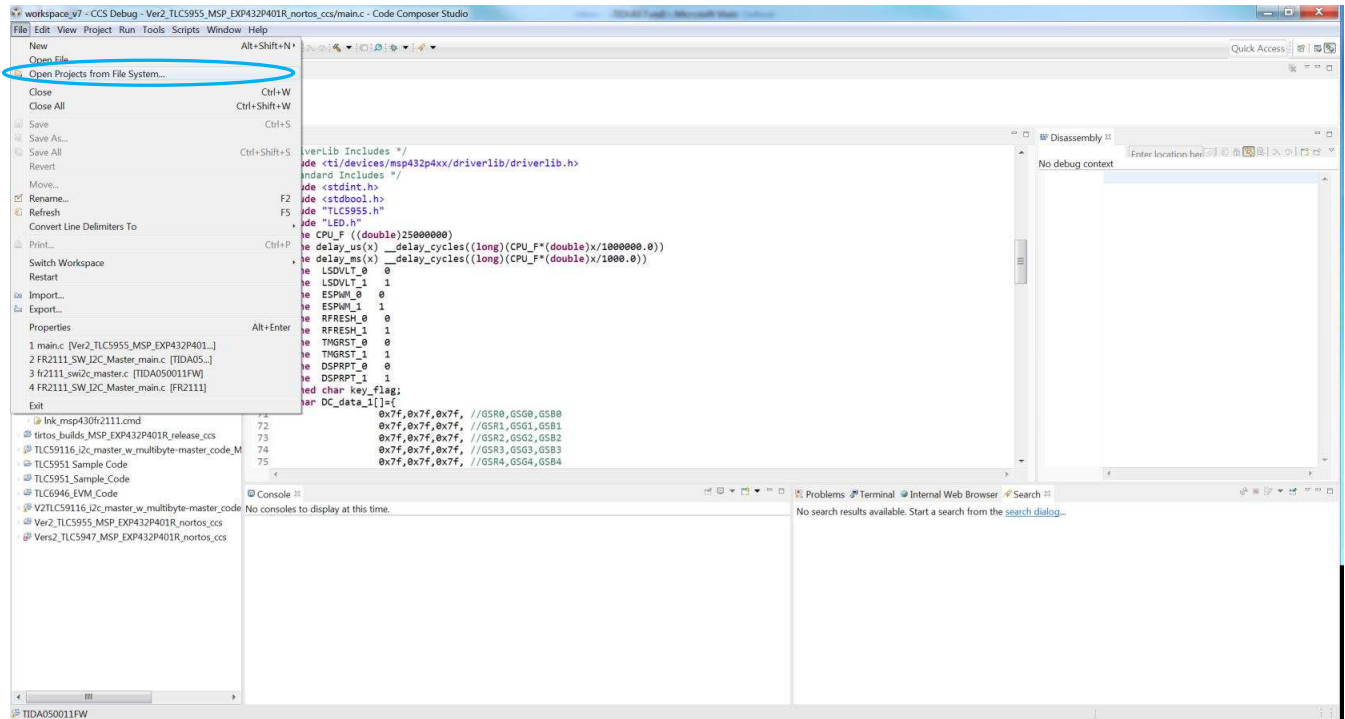
図 5. Wire Connections on The MSP430RF2311 LaunchPad Development Kit



After opening the Code Composer Studio integrated development environment (IDE), load the sample code to the MSP430FR2111. The code then runs in the MSP430FR2111.

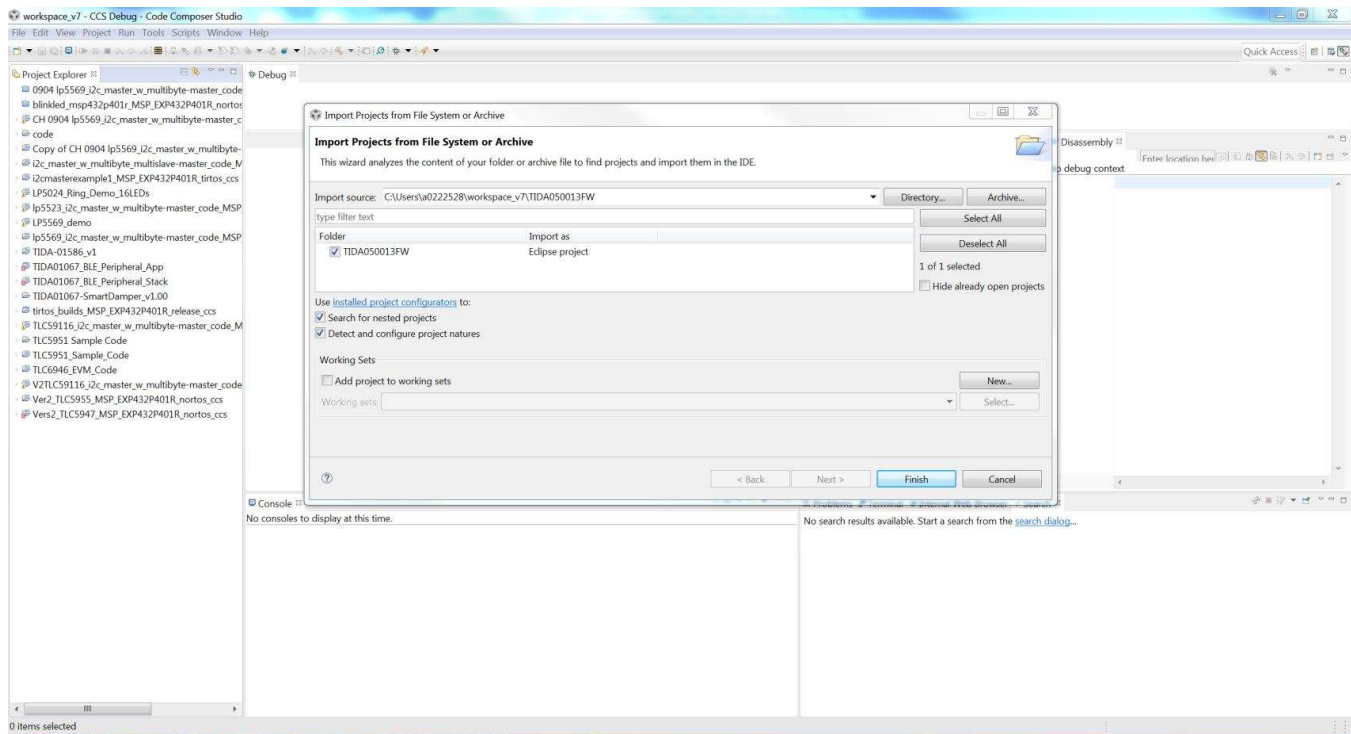
Open the project through the *Open Projects from File System* drop-down menu under *File*.

図 6. Open The Project Menu



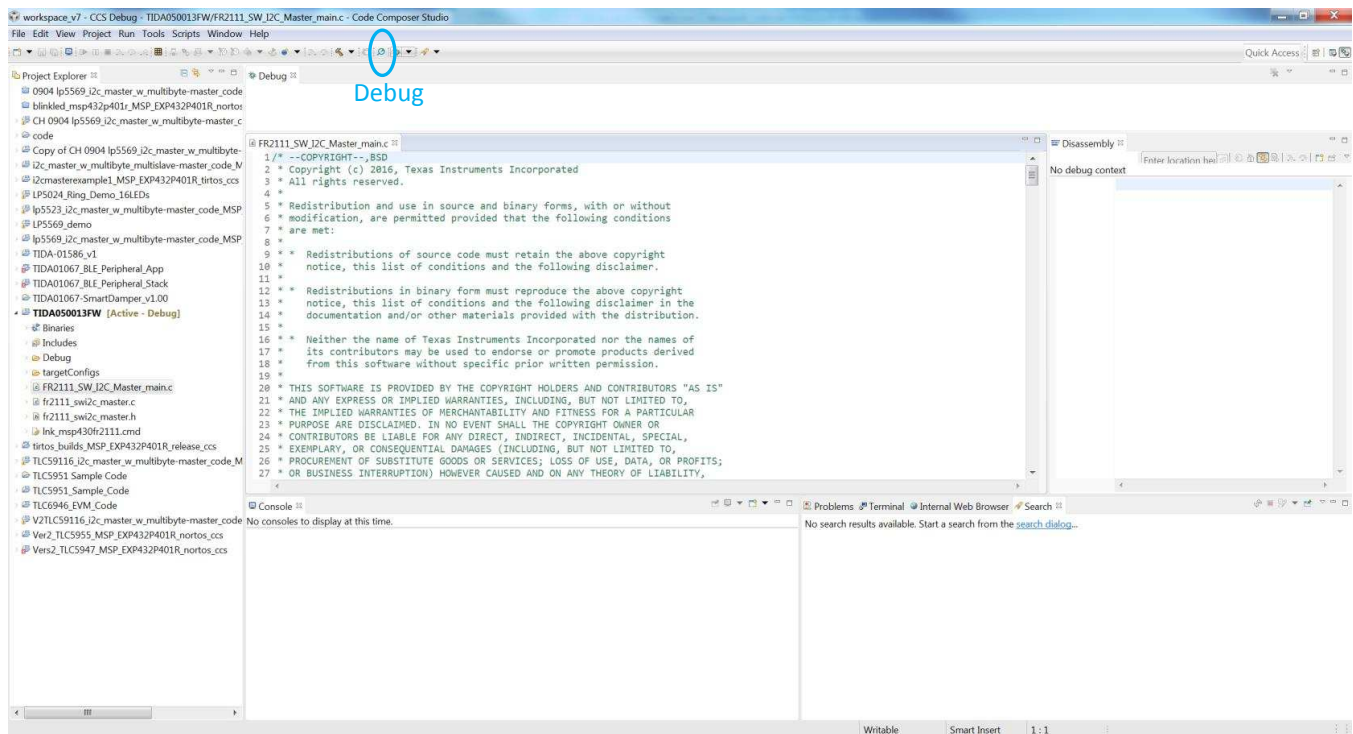
Open the TIDA050013FW folder.

図 7. Open Firmware Project



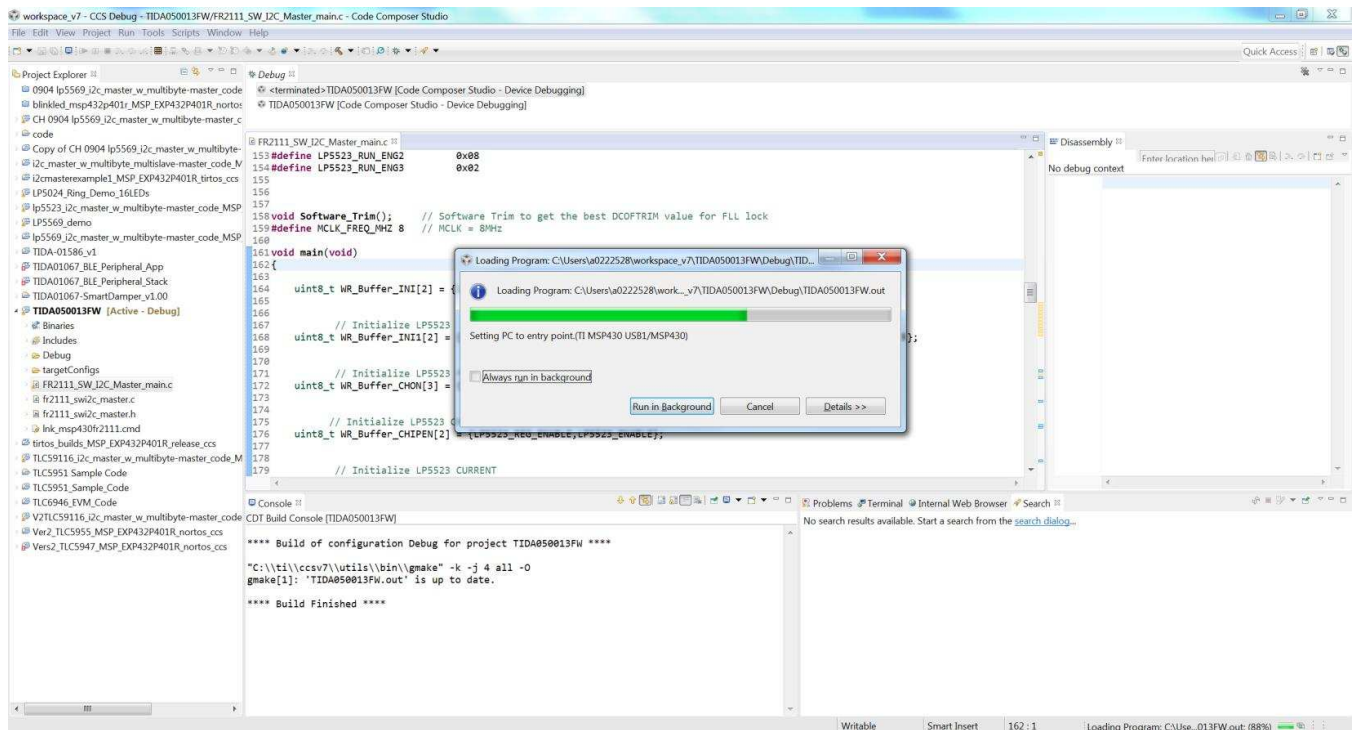
Click the Debug button to load the firmware to the MSP430FR2111 device.

8. Using Debug to Load The Program



The firmware loading is completed.

図 9. Finishing The Program Loading



3.2.2 Test Results

All the effects appear as shown in the video on the [product folder](#).

4 Design Files

4.1 Schematics

To download the schematics, see the design files at [TIDA-050013](#).

4.2 Bill of Materials

To download the bill of materials (BOM), see the design files at [TIDA-050013](#).

4.3 PCB Layout Recommendations

Place capacitors as close as possible to the LP5523 device to minimize the current loops. External components for the MSP430FR2111 device that are shown in the schematic must be placed as close as possible to the respective device pins to avoid long traces, because they add additional parasitic capacitance, inductance, and resistance on the signal. Avoid routing analog input signals close to a high-frequency pin (for example, a high-frequency PWM), because the high-frequency switching can be coupled into the analog signal.

4.3.1 Layout Prints

To download the layout prints, see the design files at [TIDA-050013](#).

4.4 Altium Project

To download the Altium Designer® project files, see the design files at [TIDA-050013](#).

4.5 Gerber Files

To download the Gerber files, see the design files at [TIDA-050013](#).

4.6 Assembly Drawings

To download the assembly drawings, see the design files at [TIDA-050013](#).

5 Software Files

To download the software files, see the design files at [TIDA-050013](#).

6 Related Documentation

1. [LP5523 Nine-Channel RGB- and White-LED Driver With Internal Program Memory and Integrated Charge Pump – DSBGA Package](#)
2. [AN-2227 LP5523 Evaluation Kit](#)
3. [MSP430FR21xx, MSP430FR2000 Mixed-Signal Microcontrollers](#)

6.1 商標

E2E, MSP430, LaunchPad, Code Composer Studio are trademarks of Texas Instruments. Altium Designer is a registered trademark of Altium LLC or its affiliated companies. Windows is a registered trademark of Microsoft Corporation. すべての商標および登録商標はそれぞれの所有者に帰属します。

7 About the Author

XING SU is an application engineer at Texas Instruments for the LED Driver product group.

重要なお知らせと免責事項

TI は、技術データと信頼性データ(データシートを含みます)、設計リソース(リファレンス・デザインを含みます)、アプリケーションや設計に関する各種アドバイス、Web ツール、安全性情報、その他のリソースを、欠陥が存在する可能性のある「現状のまま」提供しており、商品性および特定目的に対する適合性の黙示保証、第三者の知的財産権の非侵害保証を含むいかなる保証も、明示的または黙示的にかかわらず拒否します。

これらのリソースは、TI 製品を使用する設計の経験を積んだ開発者への提供を意図したものです。(1) お客様のアプリケーションに適した TI 製品の選定、(2) お客様のアプリケーションの設計、検証、試験、(3) お客様のアプリケーションが適用される各種規格や、その他のあらゆる安全性、セキュリティ、またはその他の要件を満たしていることを確実にする責任を、お客様のみが単独で負うものとします。上記の各種リソースは、予告なく変更される可能性があります。これらのリソースは、リソースで説明されている TI 製品を使用するアプリケーションの開発の目的でのみ、TI はその使用をお客様に許諾します。これらのリソースに関して、他の目的で複製することや掲載することは禁止されています。TI や第三者の知的財産権のライセンスが付与されている訳ではありません。お客様は、これらのリソースを自身で使用した結果発生するあらゆる申し立て、損害、費用、損失、責任について、TI およびその代理人を完全に補償するものとし、TI は一切の責任を拒否します。

TI の製品は、TI の販売条件 (www.tij.co.jp/ja-jp/legal/termsofsale.html)、または ti.com やかかる TI 製品の関連資料などのいずれかを通じて提供する適用可能な条項の下で提供されています。TI がこれらのリソースを提供することは、適用される TI の保証または他の保証の放棄の拡大や変更を意味するものではありません。

Copyright © 2018, Texas Instruments Incorporated
日本語版 日本テキサス・インスツルメンツ株式会社

重要なお知らせと免責事項

TI は、技術データと信頼性データ(データシートを含みます)、設計リソース(リファレンス・デザインを含みます)、アプリケーションや設計に関する各種アドバイス、Web ツール、安全性情報、その他のリソースを、欠陥が存在する可能性のある「現状のまま」提供しており、商品性および特定目的に対する適合性の黙示保証、第三者の知的財産権の非侵害保証を含むいかなる保証も、明示的または黙示的にかかわらず拒否します。

これらのリソースは、TI 製品を使用する設計の経験を積んだ開発者への提供を意図したものです。(1) お客様のアプリケーションに適した TI 製品の選定、(2) お客様のアプリケーションの設計、検証、試験、(3) お客様のアプリケーションが適用される各種規格や、その他のあらゆる安全性、セキュリティ、またはその他の要件を満たしていることを確実にする責任を、お客様のみが単独で負うものとします。上記の各種リソースは、予告なく変更される可能性があります。これらのリソースは、リソースで説明されている TI 製品を使用するアプリケーションの開発の目的でのみ、TI はその使用をお客様に許諾します。これらのリソースに関して、他の目的で複製することや掲載することは禁止されています。TI や第三者の知的財産権のライセンスが付与されている訳ではありません。お客様は、これらのリソースを自身で使用した結果発生するあらゆる申し立て、損害、費用、損失、責任について、TI およびその代理人を完全に補償するものとし、TI は一切の責任を拒否します。

TI の製品は、TI の販売条件 (www.tij.co.jp/ja-jp/legal/termsofsale.html)、または ti.com やかかる TI 製品の関連資料などのいずれかを通じて提供する適用可能な条項の下で提供されています。TI がこれらのリソースを提供することは、適用される TI の保証または他の保証の放棄の拡大や変更を意味するものではありません。

Copyright © 2018, Texas Instruments Incorporated
日本語版 日本テキサス・インスツルメンツ株式会社