MSP430FR2476 LaunchPad™ Development Kit (LP-MSP430FR2476)

The MSP430FR2476 LaunchPad™ development kit is an easy-to-use evaluation module (EVM) for the MSP430FR2476 microcontroller (MCU). The kit contains everything needed to start developing on the ultra-low-power MSP430FRx FRAM microcontroller platform, including onboard debug probe for programming, debugging, and energy measurements. The board also features onboard buttons and LEDs for quick integration of a simple user interface, an onboard temperature sensor, and a CR2032 coin cell battery holder.

Figure 1 shows the LP-MSP430FR2476 LaunchPad development kit.

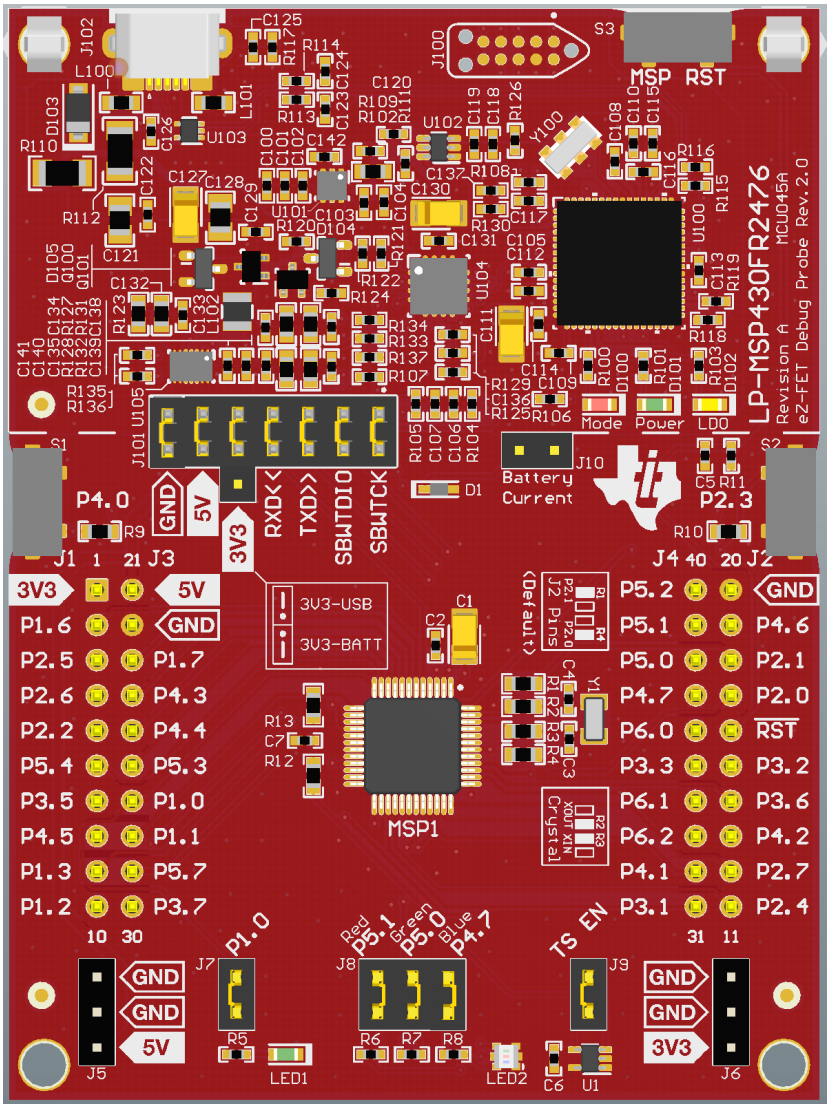


Figure 1. LP-MSP430FR2476 LaunchPad Development Kit

Copyright © 2019, Texas Instruments Incorporated

Contents

1 Getting Started 3
1.1 Introduction 3
1.2 Key Features 3
1.3 What's Included 3
1.4 First Steps: Out-of-Box Experience 3
1.5 Next Steps: Looking Into the Provided Code 4
2 Hardware 4
2.1 Block Diagram 5
2.2 Hardware Features 5
2.3 Power 10
2.4 Measure Current Draw of the MSP430 MCU 11
2.5 Clocking 12
2.6 Using the eZ-FET Debug Probe With a Different Target 12
2.7 BoosterPack Plug-in Module Pinout 12
2.8 Design Files 14
2.9 Hardware Change Log 14
3 Software Examples 14
3.1 Out-of-Box Software Example 14
3.2 Blink LED Example 15
4 Resources 16
4.1 Integrated Development Environments 16
4.2 LaunchPad Development Kit Websites 19
4.3 MSP430Ware and TI Resource Explorer 19
4.4 FRAM Utilities 20
4.5 MSP430FR2476 MCU 20
4.6 Community Resources 20
5 FAQ 21
6 Schematics 22

List of Figures

1 LP-MSP430FR2476 LaunchPad Development Kit 1
2 LP-MSP430FR2476 Overview 4
3 LP-MSP430FR2476 Block Diagram 5
4 MSP430FR2476 Pinout 6
5 eZ-FET Debug Probe 7
6 eZ-FET Isolation Jumper Block Diagram 9
7 Application Backchannel UART in Device Manager 9
8 LP-MSP430FR2476 Power Block Diagram 10
9 BoosterPack Checker Tool 13
10 LaunchPad Development Kit to BoosterPack Plug-in Module Connector Pinout 13
11 TI Resource Explorer Cloud 16
12 CCS Cloud 17
13 Directing the Project>Import Function to the Demo Project 18
14 When CCS Has Found the Project 18
15 Using TI Resource Explorer to Browse LP-MSP430FR2476 in MSP430Ware 19
16 Schematics (1 of 4) 22
17 Schematics (2 of 4) 23
18 Schematics (3 of 4) 24
19 Schematics (4 of 4) 25

Copyright © 2019, Texas Instruments Incorporated

Trademarks

LaunchPad, BoosterPack, Code Composer Studio, EnergyTrace, MSP430, E2E are trademarks of Texas Instruments.
IAR Embedded Workbench, C-SPY are registered trademarks of IAR Systems.
All other trademarks are the property of their respective owners.

1 Getting Started

1.1 Introduction

The 16-MHz MSP430FR2476 MCU features 64KB of embedded ferroelectric random access memory (FRAM), a nonvolatile memory known for its ultra-low power, high endurance, and high speed write access. Combined with 8KB of on-chip RAM, users have access to 64KB of memory to split between their program and data as required. For example, a data logging application could require a large data memory with relatively small program memory, so the memory can be allocated as required between program and data memory.

Rapid prototyping is simplified by the 40-pin BoosterPack™ plug-in module headers, which support a wide range of available BoosterPack plug-in modules. You can quickly add features like wireless connectivity, graphical displays, environmental sensing, and much more. Design your own BoosterPack plug-in module or choose among many already available from TI and third-party developers.

Free software development tools are also available, such as TI's Eclipse-based Code Composer Studio™ IDE (CCS) and IAR Embedded Workbench® IDE. Both of these IDEs support EnergyTrace™ technology for real-time power profiling and debugging when paired with the MSP430FR2476 LaunchPad development kit.

1.2 Key Features

- MSP430™ ultra-low-power FRAM technology based MSP430FR2476 16-bit MCU
- EnergyTrace technology available for ultra-low-power debugging
- 40-pin LaunchPad development kit standard leveraging the BoosterPack plug-in module ecosystem
- Onboard eZ-FET debug probe
- 2 buttons and 2 LEDs (1 RGB LED) for user interaction
- Temperature sensor for the Out-of-Box Experience demo

1.3 What's Included

1.3.1 Kit Contents

- 1 LP-MSP430FR2476 LaunchPad development kit
- 1 micro USB cable
- 1 quick start guide

1.3.2 Software Examples

- Out-of-Box software

1.4 First Steps: Out-of-Box Experience

An easy way to get started with the EVM is by using its preprogrammed out-of-box code. This code demonstrates some key features of the EVM.

1.4.1 Connecting to the Computer

Connect the LaunchPad development kit using the included USB cable to a computer. A green power LED should illuminate. For proper operation, drivers are needed. TI recommends that you get the drivers by installing an IDE such as TI's Code Composer Studio IDE or IAR Embedded Workbench IDE. Drivers are also available at www.ti.com/MSPdrivers.

Copyright © 2019, Texas Instruments Incorporated

1.4.2 Running the Out-of-Box Experience (OOBE)

The OOBE of the LP-MSP430FR2476 LaunchPad development kit demonstrates how to implement a simple thermometer by using the internal ADC of the MSP430FR2476 MCU to measure the output voltage from an analog temperature sensor.

When powering up the Out-of-Box demo, the board performs an initial temperature calibration and illuminates RGB LED2 in a green color. Try to heat the temperature sensor U1 by blowing hot air on it to see if you can get LED2 to turn red or using a compressed air can to blow cold air onto the sensor to see LED2 turn blue. LED2 indicates the relative difference between the current temperature and the calibrated measurement.

Press button S1 to recalibrate the temperature to the most recent measurement and send a UART message (at 9600 baud) containing the new calibrated temperature through the backchannel UART. Press button S2 at any time to send a UART message containing the current ambient temperature measurement in Celsius.

1.5 Next Steps: Looking Into the Provided Code

After the EVM features have been explored, the fun can begin. It's time to open an integrated development environment and start editing the code examples. See Section 4 for available IDEs and where to download them.

The quickest way to get started using the LaunchPad development kit is to use TI's cloud development tools. The cloud-based Resource Explorer provides access to all of the examples and resources in MSP430Ware software. Code Composer Studio Cloud is a simple Cloud-based IDE that enables developing and running applications on the LaunchPad development kit.

The out-of-box source code and more code examples are provided and available on the download page. Code is licensed under BSD, and TI encourages reuse and modifications to fit specific needs.

Section 3 describes all functions in detail and provides a project structure to help familiarize you with the code.

With the onboard eZ-FET debug probe, debugging and downloading new code is simple. A connection between the EVM and a PC through the provided USB cable is all that is needed.

2 Hardware

Figure 2 shows an overview of the LP-MSP430FR2476 hardware.

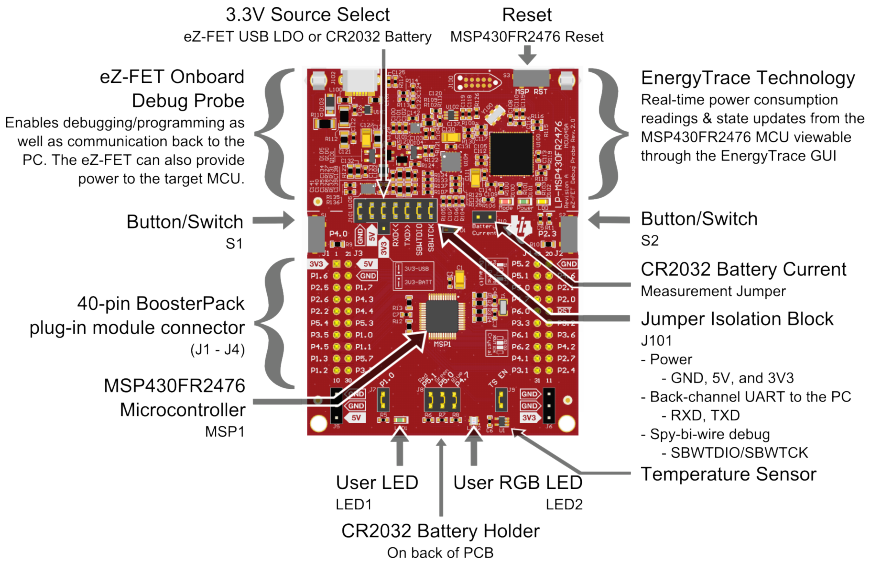


Figure 2. LP-MSP430FR2476 Overview

Copyright © 2019, Texas Instruments Incorporated

2.1 Block Diagram

Figure 3 shows the block diagram.

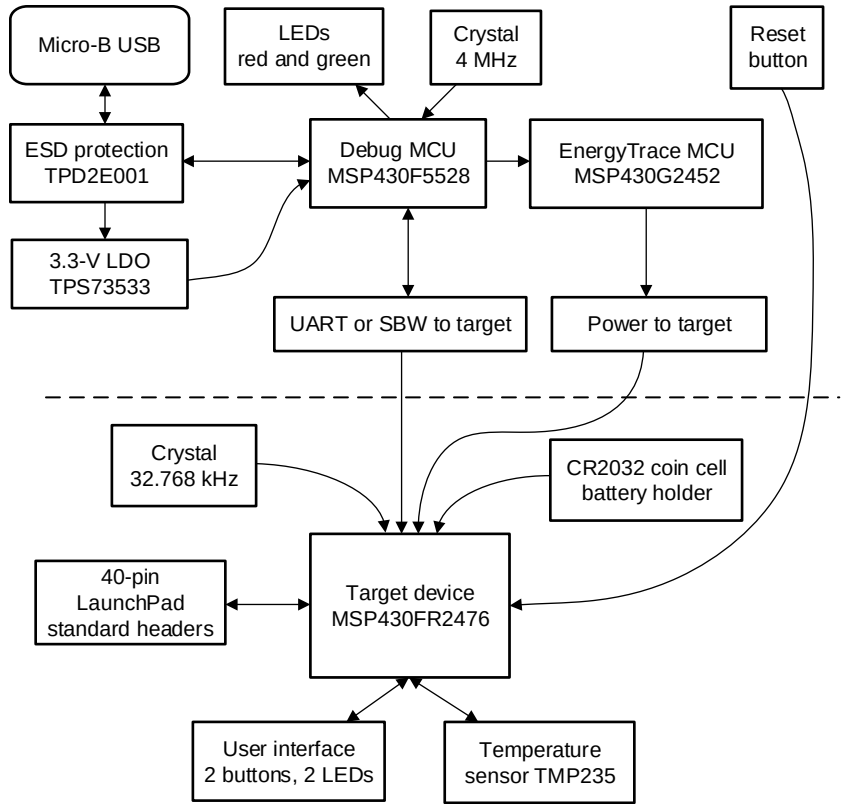


Figure 3. LP-MSP430FR2476 Block Diagram

2.2 Hardware Features

2.2.1 MSP430FR2476 MCU

The MSP430FR2476 is an ultra-low-power MSP430FRx FRAM-based microcontroller (MCU), which offer extended data logging and security capabilities. The MSP430FR2476 offers the small LQFP package (7 mm × 7 mm) in the FRAM microcontroller portfolio, combined with a variety of integrated peripherals and ultra-low power consumption. FRAM is a cutting edge memory technology, combining the best features of flash and RAM into one nonvolatile memory. For more information on FRAM, visit www.ti.com/fram.

Device features include:

- 1.8-V to 3.6-V operation
- 16-bit RISC architecture up to 16-MHz system clock and 8-MHz FRAM access
- 64KB of program FRAM, 512 bytes of information FRAM, and 8KB of RAM
- 12-channel 12-bit ADC
- One enhanced comparator with integrated 6-bit DAC as reference voltage
- Four 16-bit timers with three capture/compare registers (Timer_B3)
- One 16-bit timer with seven capture/compare registers (Timer_B7)
- One 16-bit counter-only RTC
- 16-bit cyclic redundancy check (CRC)
- 43 GPIOs

Copyright © 2019, Texas Instruments Incorporated

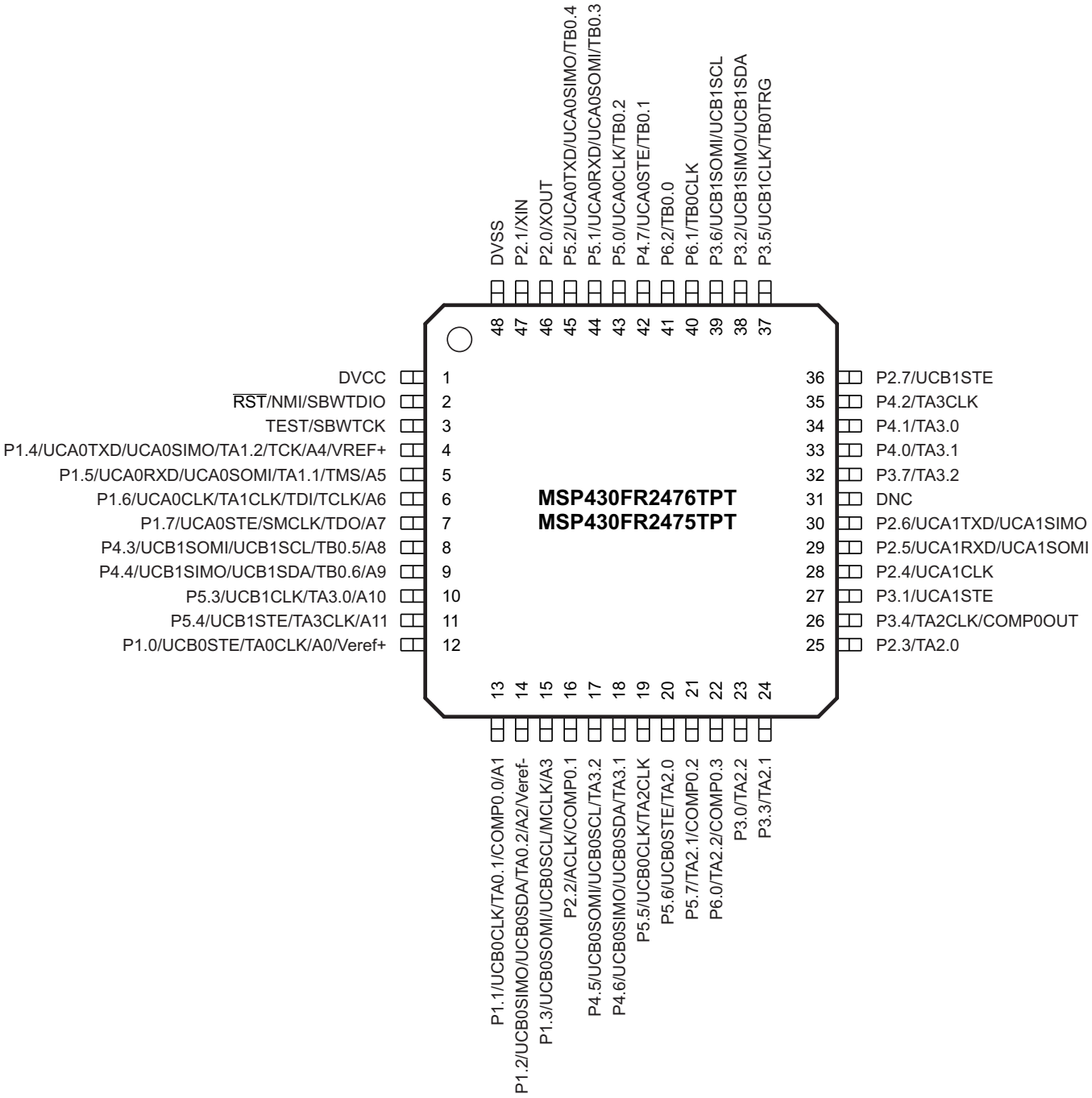


Figure 4. MSP430FR2476 Pinout

Copyright © 2019, Texas Instruments Incorporated

2.2.2 eZ-FET Onboard Debug Probe With EnergyTrace™ Technology

To keep development easy and cost effective, TI's LaunchPad development kits integrate an onboard debug probe, which eliminates the need for expensive programmers. The LP-MSP430FR2476 has the eZ-FET debug probe (see Figure 5), which is a simple and low-cost debugger that supports all MSP430™ device derivatives.

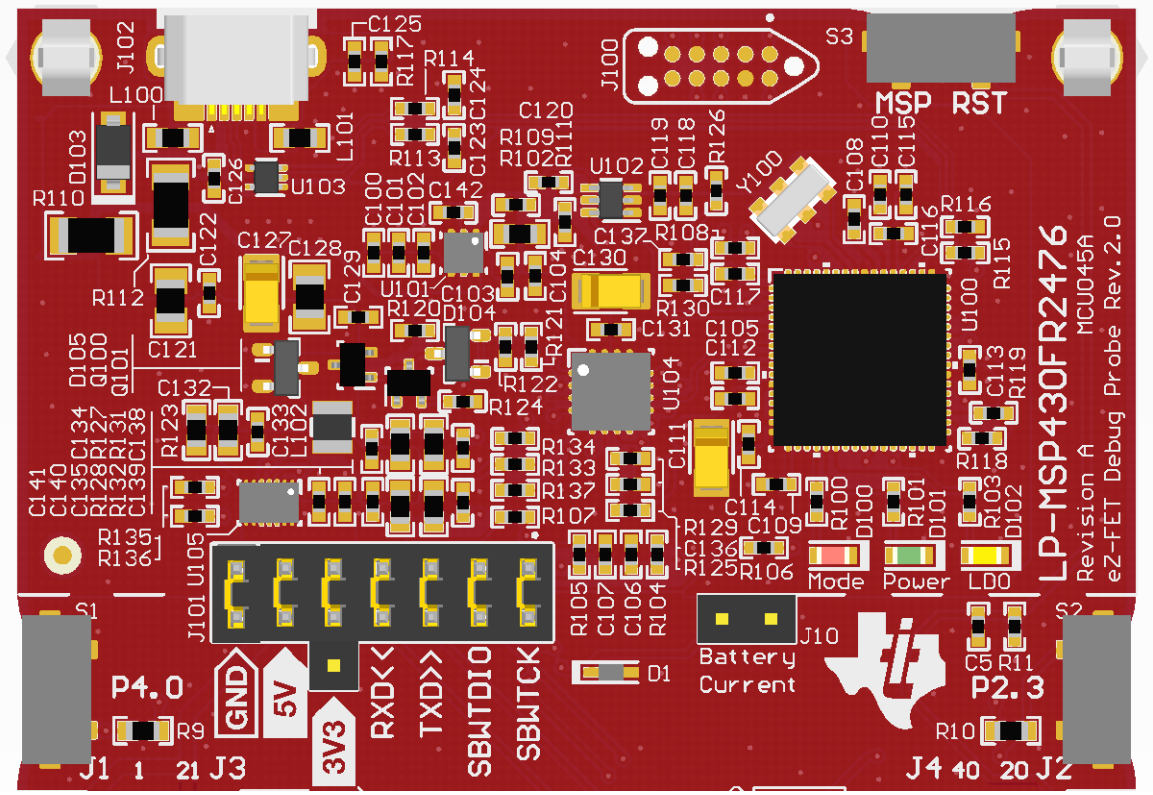


Figure 5. eZ-FET Debug Probe

The LP-MSP430FR2476 LaunchPad development kit features EnergyTrace technology but does not have support for EnergyTrace++ technology. The EnergyTrace technology functionality varies across the MSP portfolio (see Table 1).

Table 1. EnergyTrace Technology

Features	EnergyTrace Technology	EnergyTrace++ Technology
Current monitoring	✓	✓
CPU state		✓
Peripheral and system state		✓
Devices supported	All MSP430 MCUs	MSP430FR59xx and MSP430FR69xx MCUs
Development tool required	MSP-FET or eZ-FET	MSP-FET or eZ-FET

The dotted line through J101 shown in Figure 5 divides the eZ-FET debug probe from the target area. The signals that cross this line can be disconnected by jumpers on J101, the isolation jumper block. For more details on the isolation jumper block, see Section 2.2.3.

The eZ-FET also provides a backchannel UART-over-USB connection with the host, which can be very useful during debugging and for easy communication with a PC. For more details on this connection, see Section 2.2.4.

The eZ-FET hardware schematics are available in Section 6 and in the hardware design files download page. For more information on the software and the debugger, visit the eZ-FET wiki.

Copyright © 2019, Texas Instruments Incorporated

2.2.3 Debug Probe Connection: Isolation Jumper Block

The isolation jumper block at jumper J101 lets you connect or disconnect signals that cross from the eZ-FET domain into the MSP430FR2476 target domain. This includes eZ-FET Spy-Bi-Wire signals, application UART signals, and 3.3-V and 5-V power.

Reasons to open these connections:

- To remove any and all influence from the eZ-FET debug probe for high accuracy target power measurements
- To control 3-V and 5-V power flow between the eZ-FET and target domains
- To expose the target MCU pins for uses other than onboard debugging and application UART communication
- To expose the programming and UART interface of the eZ-FET to use it for devices other than the onboard MCU
- To select the CR2032 battery holder as the 3-V source

Table 2. Isolation Block Connections

Jumper	Description
GND	Ground
5V	5-V VBUS from USB
3V3	3.3-V rail, derived from VBUS in the eZ-FET domain or 3V rail derived from CR2032 battery
RXD <<	Backchannel UART: The target FR2476 receives data through this signal. The arrows indicate the direction of the signal.
TXD >>	Backchannel UART: The target FR2476 sends data through this signal. The arrows indicate the direction of the signal.
SBW $\overline{\text{RST}}$	Spy-Bi-Wire debug: SBWTDIO data signal. This pin also functions as the $\overline{\text{RST}}$ signal (active low).
SBW TST	Spy-Bi-Wire debug: SBWTCK clock signal. This pin also functions as the TST signal.

Copyright © 2019, Texas Instruments Incorporated

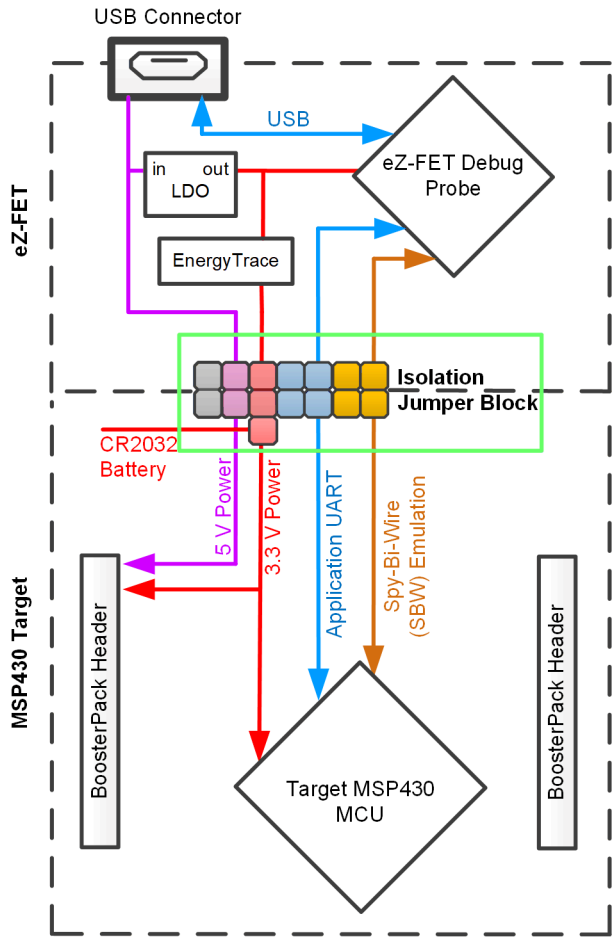


Figure 6. eZ-FET Isolation Jumper Block Diagram

2.2.4 Application (or Backchannel) UART

The backchannel UART allows communication with the USB host that is not part of the target application's main functionality. This is very useful during development and also provides a communication channel to the host PC. Data from this communication channel can be used to create a graphical user interface (GUI) or other programs on the PC that communicate with the LaunchPad development kit.

Figure 6 shows the pathway of the backchannel UART. The backchannel UART is the UART on eUSCI_A0.

A virtual COM port for the application backchannel UART is generated on the host when the LaunchPad development kit enumerates on the host. You can use any PC application that interfaces with COM ports, including terminal applications like Hyperterminal or Docklight, to open this port and communicate with the target application. You need to identify the COM port for the backchannel. On Windows PCs, Device Manager can assist.

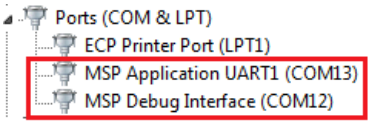


Figure 7. Application Backchannel UART in Device Manager

Copyright © 2019, Texas Instruments Incorporated

The backchannel UART is the "MSP Application UART1" port. In this case, Figure 7 shows COM13, but this port can vary from one host PC to the next. After you identify the correct COM port, configure it in your host application according to its documentation. You can then open the port and begin communication to it from the host.

On the target MSP430FR2476 MCU, the backchannel is connected to the eUSCI_A0 module. The eZ-FET has a configurable baud rate, and it is important to configure the PC application with the same baud rate as the eUSCI_A0.

2.2.5 Special Features

2.2.5.1 TMP235 Temperature Sensor

The MSP430FR2476 MCU LaunchPad development kit features a TMP235A2 temperature sensor that is accurate to ±0.5°C (typical) from –40°C to +150°C. The analog output of the TMP235 is connected to analog input A1 (P1.1) on the MSP430FR2476 through a capacitor (C7) and a pair of resistors (R12 and R13). The capacitor and resistors can be replaced if a custom RC circuit is needed. For applications that use the temperature sensor and a connected BoosterPack plug-in module, determine if there are any pin conflicts. The 3.3-V power rail can be disconnected from the TMP235 device by removing the shunt jumper at J9. This is useful in low-power applications that do not use the temperature sensor.

2.2.5.2 CR2032 Coin Cell Battery

On the back of the LP-MSP430FR2476 board is a holder for a CR2032 coin cell battery (battery not included). A CR2032 battery can be added to demonstrate battery-powered applications. See Section 2.3.2 for additional details on how to power the LaunchPad development kit with the coin cell battery.

2.3 Power

The board accommodates various powering methods, including through the onboard eZ-FET, using the CR2032 battery, or using external or BoosterPack plug-in module power (see Figure 8).

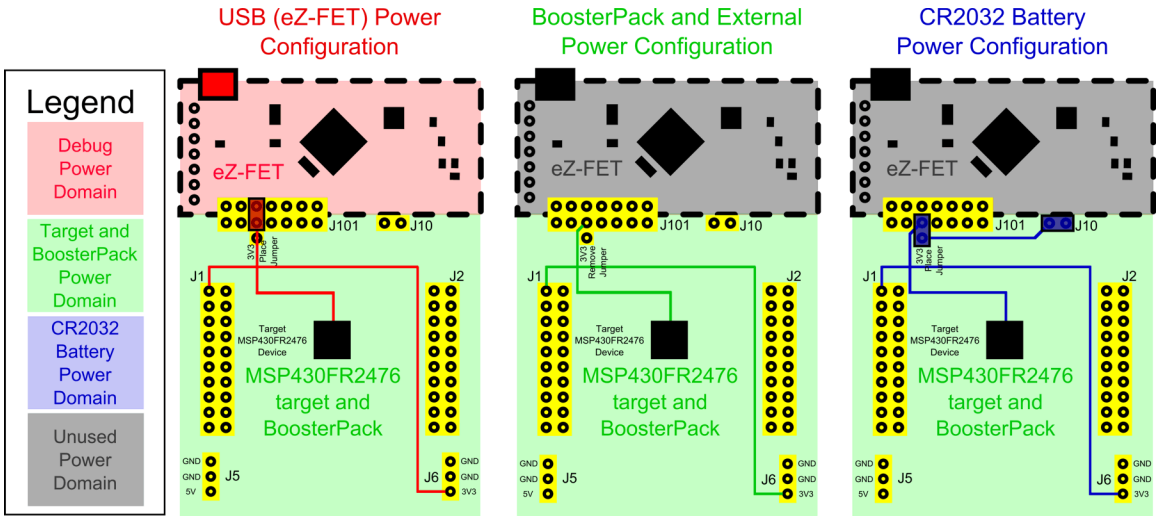


Figure 8. LP-MSP430FR2476 Power Block Diagram

2.3.1 eZ-FET USB Power

The most common power-supply scenario is from USB through the eZ-FET debugger. This provides 5-V power from the USB and also regulates this power rail to 3.3 V for eZ-FET operation and 3.3 V to the target side of the LaunchPad development kit. Power from the eZ-FET is controlled by jumper J101. For 3.3 V, make sure that a jumper is connected across the J101 3V3 terminal.

Copyright © 2019, Texas Instruments Incorporated

2.3.2 CR2032 Battery Power

A CR2032 battery holder is on the back of the LP-MSP430FR2476 board to power the LaunchPad development kit from a single coin battery. To power the LaunchPad kit from the CR2032 coin cell, insert the battery securely and in the correct orientation, and remove all of the shunt jumpers from J101 except the 3V3 jumper. Place the 3V3 jumper in the lower configuration to connect the battery output to the target side 3.3-V rail. This placement is also shown on the silkscreen on the PCB. Place a shunt jumper across J10. Alternatively, place an ammeter across the pins of J10 to measure current out of the battery.

Do not use the eZ-FET emulation circuit when the LP-MSP430FR2476 is powered from the CR2032 coin cell battery. Remove all of the shunt jumpers on J101 except the 3V3 (which is in the lower position) to disconnect the eZ-FET circuit from the target side of the LaunchPad development kit. The CR2032 voltage level will decrease with use and can cause voltage level violations on the eZ-FET portion of the design if left connected. To debug the target device, the LaunchPad development kit should be configured for eZ-FET USB Power as shown in Section 2.3.1

When not using the CR2032 battery to power the board, be sure the 3V3 shunt jumper on J101 is not placed in the battery configuration. It is also useful to remove the jumper from J10 to disconnect the battery from the rest of the power domains. This ensures there is no risk of applying an external voltage source to the battery.

2.3.3 BoosterPack Plug-in Module and External Power Supply

Headers J6 is present on the board to supply external power directly. It is important to comply with the device voltage operation specifications when supplying external power. The MSP430FR2476 has an operating range of 1.8 V to 3.6 V. More information can be found in the MSP430FR247x Mixed-Signal Microcontrollers data sheet.

2.4 Measure Current Draw of the MSP430 MCU

To measure the current draw of the MSP430FR2476 MCU using a multimeter, use the 3V3 jumper on the J101 jumper isolation block. The current measured includes the target device and any current drawn through the BoosterPack plug-in module headers.

To measure ultra-low power, follow these steps:

- Remove the 3V3 jumper in the J101 isolation block, and attach an ammeter across this jumper. Be sure to use the correct pair of pins as outlined in Section 2.3 to select between eZ-FET power and CR2032 power.
- Consider the effect that the backchannel UART and any circuitry attached to the MSP430FR2476 can have on current draw. Consider disconnecting these at the isolation jumper block, or at least consider their current sinking and sourcing capability in the final measurement.
- If using the CR2032 to supply power, remove all shunt jumpers at J101, except 3V3 in the lower position, and connect J10 with a shunt jumper.
- Make sure there are no floating inputs/outputs (I/Os) on the MSP430FR2476. These cause unnecessary extra current draw. Every I/O should either be driven out or, if it is an input, should be pulled or driven to a high or low level.
- Begin target execution.
- Measure the current. Keep in mind that if the current levels are fluctuating, it may be difficult to get a stable measurement. It is easier to measure quiescent states.

EnergyTrace technology can also be used to compare various current profiles and better optimize your energy performance.

Copyright © 2019, Texas Instruments Incorporated

2.5 Clocking

The LP-MSP430FR2476 provides an external clock in addition to the internal clocks in the device.

- Q1: 32.768-kHz 12.5-pF crystal

The 32.768-kHz crystal allows for lower LPM sleep currents than do the other low-frequency clock sources. Therefore, the presence of the crystal allows the full range of low-power modes to be used.

The internal clocks in the device default to the following configuration:

- MCLK: DCO at 1 MHz
- SMCLK: DCO at 1 MHz
- ACLK: REFO at 32.768 kHz

For more information about configuring internal clocks and using the external oscillators, see the MSP430FR4xx and MSP430FR2xx Family User's Guide.

2.6 Using the eZ-FET Debug Probe With a Different Target

The eZ-FET debug probe on the LaunchPad development kit can interface to most MSP430 MCUs, not just the onboard MSP430FR2476 target device.

To interface to another MCU, disconnect all jumpers in the isolation jumper block. This is necessary because the debug probe cannot connect to more than one target at a time over the Spy-Bi-Wire (SBW) connection.

Next, make sure the target board has proper connections for SBW. To be compatible with SBW, the capacitor on $\overline{\text{RST}}$/SBWTDIO must be 2.2 nF or less. For documentation on designing MSP430 JTAG interface circuitry, see the MSP430 Hardware Tools User's Guide.

Finally, wire together the following signals from the debug probe side of the isolation jumper block to the target hardware:

- 5 V (if 5 V is needed)
- 3.3 V
- GND
- SBWTDIO
- SBWTCK
- TXD (if the UART backchannel is to be used)
- RXD (if the UART backchannel is to be used)

This wiring can be done either with jumper wires or by designing the board with a connector that plugs into the isolation jumper block.

2.7 BoosterPack Plug-in Module Pinout

The LaunchPad development kit adheres to the 40-pin LaunchPad development kit pinout standard. This standard was created to aid compatibility between LaunchPad development kits and BoosterPack plug-in modules across the TI ecosystem.

While most BoosterPack plug-in modules are compliant with the standard, some are not. The LP-MSP430FR2476 LaunchPad development kit is compatible with all 40-pin BoosterPack plug-in modules that are compliant with the standard. If the reseller or owner of the BoosterPack plug-in module does not explicitly indicate compatibility with the LP-MSP430FR2476 LaunchPad development kit, compare the schematic of the candidate BoosterPack plug-in module with the LaunchPad development kit to ensure compatibility. Keep in mind that sometimes conflicts can be resolved by changing the MSP430FR2476 device pin function configuration in software.

Copyright © 2019, Texas Instruments Incorporated

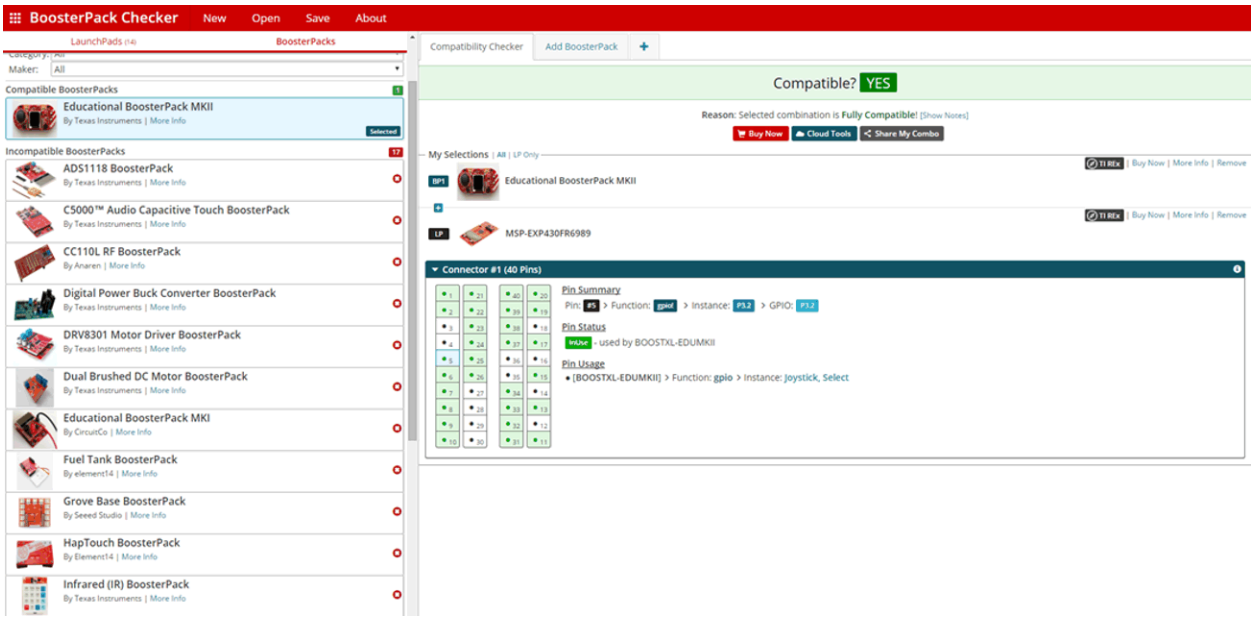


Figure 9. BoosterPack Checker Tool

To check the compatibility of a BoosterPack plug-in module with the LaunchPad development kit of your choice, use the BoosterPack Checker Tool. This allows you to select any LaunchPad development kit that TI offers and determine its compatibility with any number of BoosterPack plug-in modules that we offer. You can also add your own BoosterPack plug-in module to check its compatibility as you prototype that next design.

Figure 10 shows the 40-pin pinout of the MSP430FR2476 LaunchPad development kit.

Note that software's configuration of the pin functions plays a role in compatibility. The LaunchPad development kit side of the dashed line shows only the applicable function for conforming to the standard. However, each pin has other functionality that can be configured by the software. See the MSP430FR2476 device data sheet for more details on individual pin functions.

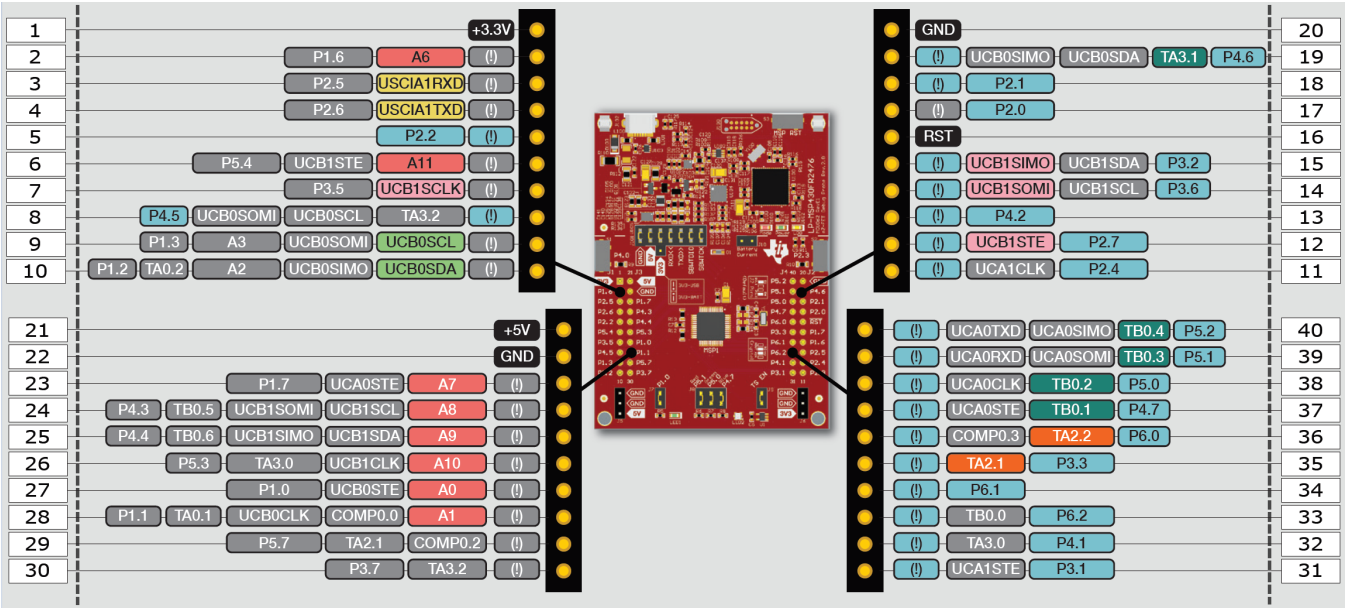


Figure 10. LaunchPad Development Kit to BoosterPack Plug-in Module Connector Pinout

Copyright © 2019, Texas Instruments Incorporated

2.8 Design Files

2.8.1 Hardware

Schematics can be found in Section 6. All design files including schematics, layout, bill of materials (BOM), Gerber files, and documentation are available in the LP-MSP430FR2476 design files.

2.8.2 Software

All software files including TI-TXT object-code firmware images, software example projects, and documentation are available on the LP-MSP430FR2476 software download page.

2.9 Hardware Change Log

Table 3 lists the hardware revision history.

Table 3. Hardware Change Log

PCB Revision	Description
Rev A	Initial release

3 Software Examples

Two software examples are included with the MSP430FR2476 LaunchPad development kit (see Table 4), which can be found in the MSP430FR2476 LaunchPad development kit download page and are also available in MSP430Ware software.

Table 4. Software Examples

Demo Name	BoosterPack Plug-in Module Required	Description	More Details
OutofBox_LP-MSP430FR2476	None	The out-of-box demo that is factory-programmed on the LaunchPad development kit. Demonstrates features of MSP430FR2476 device.	Section 3.1
BlinkLED_LP-MSP430FR2476	None	Blinks an LED on the LaunchPad development kit at a fixed interval.	Section 3.2

To use any of the software examples with the LaunchPad development kit, you must have an integrated development environment (IDE) that supports the MSP430FR2476 device (see Table 5).

Table 5. IDE Minimum Requirements for LP-MSP430FR2476

Code Composer Studio IDE	IAR Embedded Workbench for Texas Instruments MSP430 IDE
v8.0 or later	v7.12.1 or later

For more details on how to get started quickly and where to download the latest CCS and IAR IDEs, see Section 4.

3.1 Out-of-Box Software Example

This section describes the functionality and structure of the Out-of-Box software that is preloaded in the EVM.

The Out-of-Box Experience (OOBE) of the LP-MSP430FR2476 LaunchPad development kit demonstrates how to implement a simple thermometer by using the internal ADC of the MSP430FR2476 to measure the output voltage from an analog temperature sensor.

Copyright © 2019, Texas Instruments Incorporated

3.1.1 Source File Structure

The project is split into multiple files (see Table 6). This makes it easier to navigate and reuse parts of it for other projects.

Table 6. Source File and Folders

Name	Description
main.c	Out-of-Box demo main function
Library: driverlib	Device driver library

3.1.2 Overview

When powering up the Out-of-Box demo, the board performs an initial temperature calibration and illuminates RGB LED2 in a green color. Try to heat the temperature sensor U1 by blowing hot air on it to see if you can get LED2 to turn red or using a compressed air can to blow cold air onto the sensor to see LED2 turn blue. LED2 indicates the relative difference between the current temperature and the calibrated measurement.

Press button S1 to recalibrate the temperature to the most recent measurement and send a UART message (at 9600 baud) containing the new calibrated temperature through the backchannel UART. Press button S2 at any time to send a UART message containing the current ambient temperature measurement in Celsius.

3.2 Blink LED Example

This software example shows how to software toggle a GPIO to blink an LED on the LaunchPad development kit.

3.2.1 Source File Structure

The project is split into multiple files (see Table 7). This makes it easier to navigate and reuse parts of it for other projects.

Table 7. Source File and Folders

Name	Description
main.c	The Blink LED main function
Library: Driverlib	Device driver library

The main code uses the MSP430 Driver Library to halt the watchdog timer and to configure and toggle the P1.0 GPIO pin connected to the LED inside a software loop.

Copyright © 2019, Texas Instruments Incorporated

4 Resources

4.1 Integrated Development Environments

Although the source files can be viewed with any text editor, more can be done with the projects if they're opened with a development environment like Code Composer Studio IDE (CCS) or IAR Embedded Workbench IDE.

4.1.1 TI Cloud Development Tools

TI's Cloud-based software development tools provide instant access to MSP430Ware content and a web-based IDE.

4.1.1.1 TI Resource Explorer Cloud

TI Resource Explorer Cloud provides a web interface for browsing examples, libraries and documentation found in MSP430Ware without having to download files to your local drive.

Find out more about TI Resource Explorer Cloud at dev.ti.com.

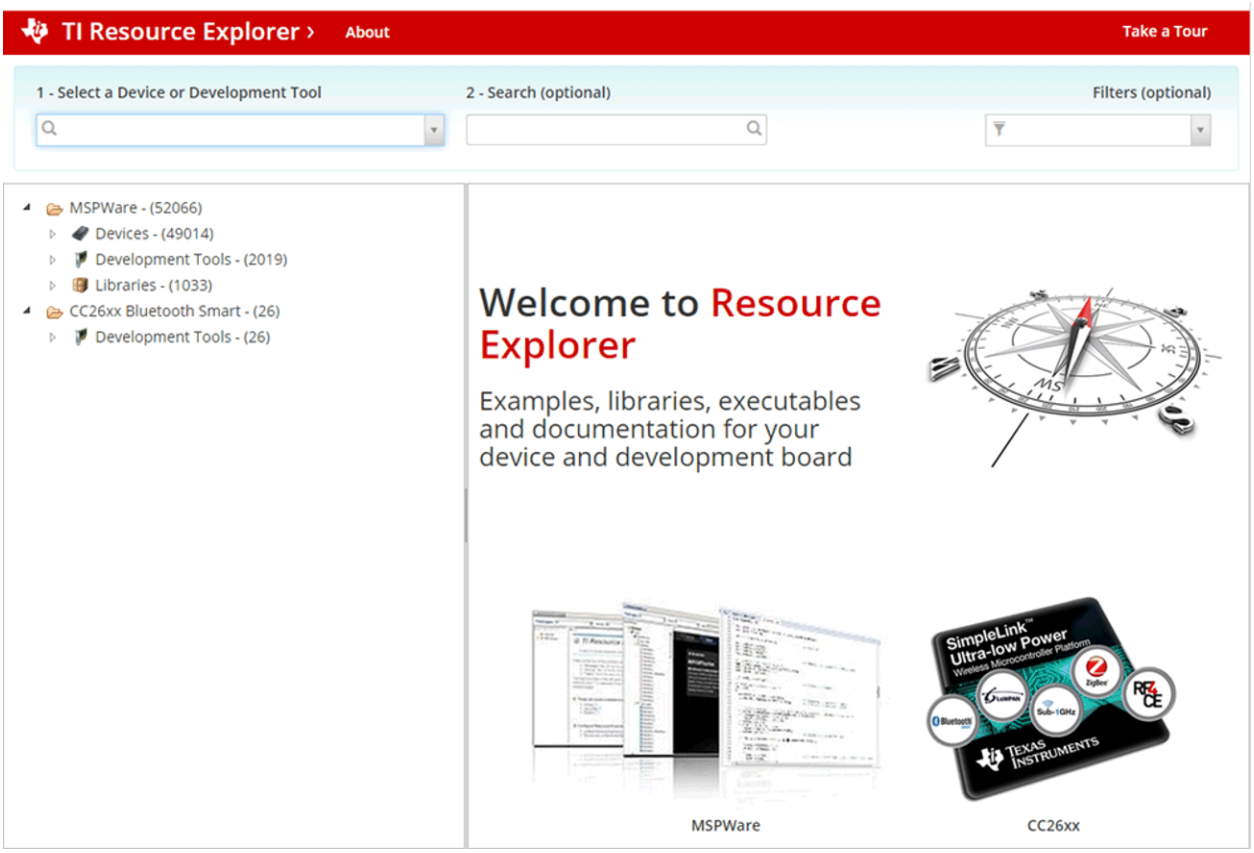


Figure 11. TI Resource Explorer Cloud

Copyright © 2019, Texas Instruments Incorporated

4.1.1.2 Code Composer Studio Cloud

Code Composer Studio Cloud (CCS Cloud) is a web-based IDE that lets you quickly create, edit, build, and debug applications for a LaunchPad development kit. No need to download and install large software packages, simply connect your LaunchPad development kit and begin. You can choose to select from a large variety of examples in MSP430Ware software and Energia or develop your own application. CCS Cloud supports debug features such as execution control, breakpoints and viewing variables.

For more information, see the full comparison between CCS Cloud and CCS Desktop.

Try Code Composer Studio Cloud now at dev.ti.com.

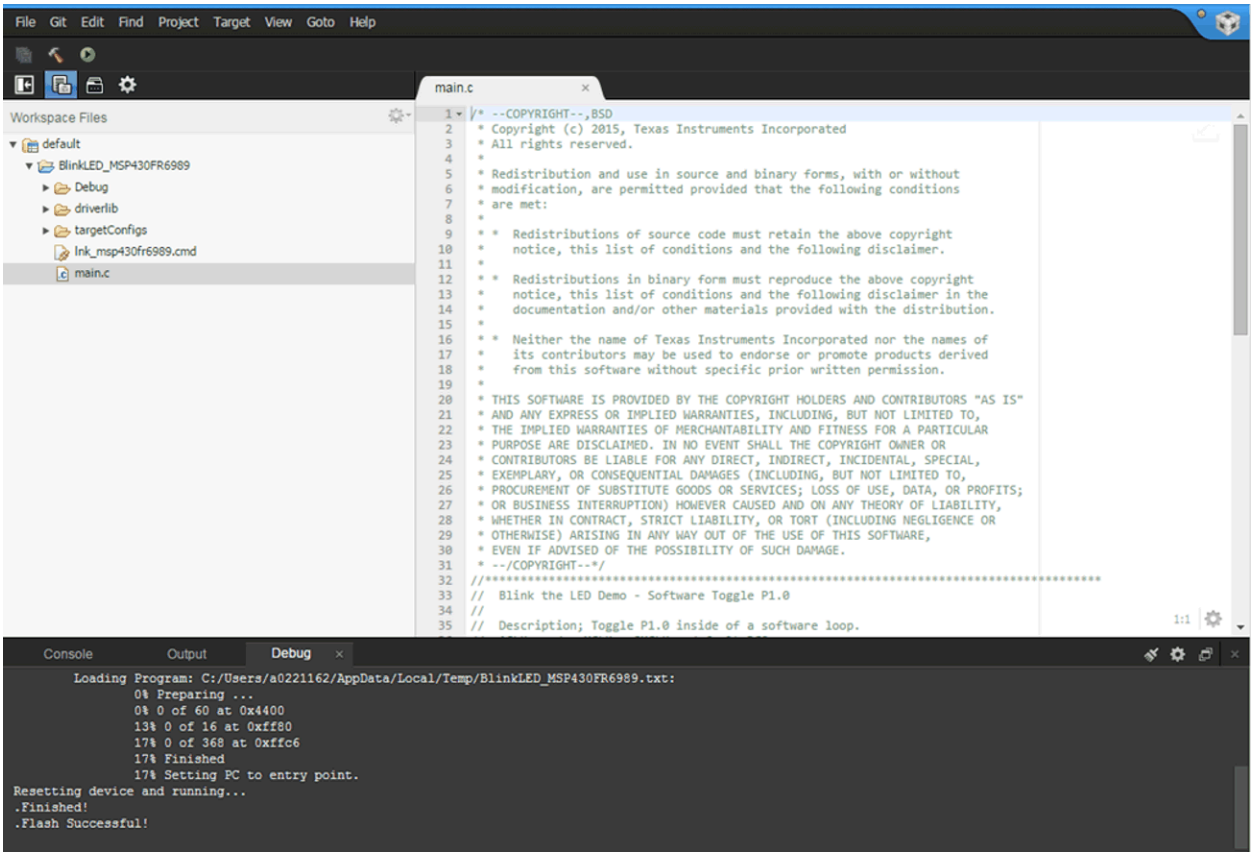


Figure 12. CCS Cloud

4.1.2 Code Composer Studio IDE

Code Composer Studio Desktop is a professional integrated development environment that supports TI's microcontroller and embedded processors portfolio. Code Composer Studio Desktop comprises a suite of tools used to develop and debug embedded applications. It includes an optimizing C/C++ compiler, source code editor, project build environment, debugger, profiler, and many other features.

Learn more about CCS and download it at http://www.ti.com/tool/ccstudio.

CCS v8.0 or higher is required for this development kit. To load an existing project, launch CCS, choose a workspace directory, and click Project>Import Existing CCS Eclipse Project. Browse to the desired demo project directory that contains main.c (see Figure 13).

Copyright © 2019, Texas Instruments Incorporated

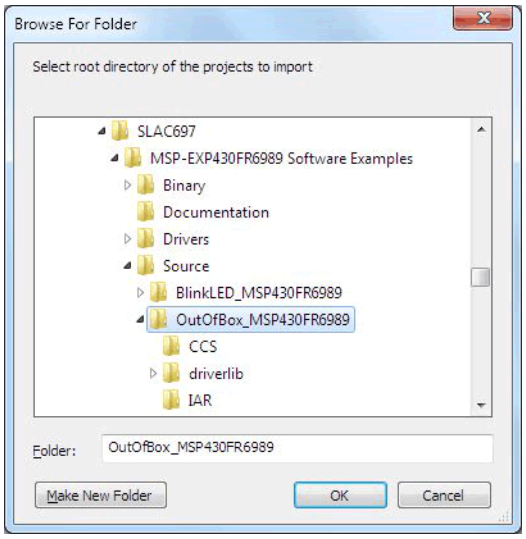


Figure 13. Directing the Project>Import Function to the Demo Project

Selecting the \CCS subdirectory also works. The CCS-specific files are located there.

When you click OK, CCS should recognize the project and allow you to import it. The indication that CCS has found it is that the project appears in the box shown in Figure 14, and it has a checkmark to the left of it.

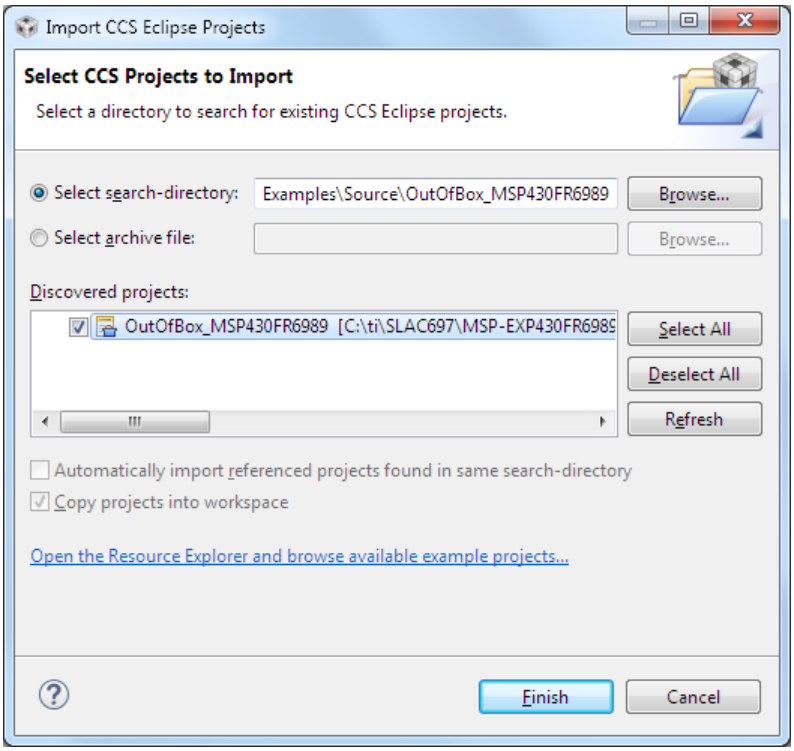


Figure 14. When CCS Has Found the Project

Sometimes CCS finds the project but does not show a checkmark; this might mean that your workspace already has a project by that name. You can resolve this by renaming or deleting that project. (Even if you do not see it in the CCS workspace, look in the workspace directory on the file system.)

Copyright © 2019, Texas Instruments Incorporated

4.1.3 IAR Embedded Workbench for MSP430 IDE

IAR Embedded Workbench for MSP430 is another powerful integrated development environment that lets you develop and manage complete embedded application projects. It integrates the IAR C/C++ compiler, IAR assembler, IAR ILINK linker, editor, project manager, command line build utility, and IAR C-SPY® debugger.

Learn more about IAR Embedded Workbench for MSP430 and download it at the IAR Systems website.

IAR v7.12.1 or higher is required for this development kit. To open the demo in IAR, click File>Open>Workspace…, and browse to the *.eww workspace file inside the \IAR subdirectory of the desired demo. All workspace information is contained within this file.

The subdirectory also has an *.ewp project file. This file can be opened into an existing workspace by clicking Project>Add-Existing-Project….

Although the software examples have all of the code required to run them, IAR users can download and install MSP430Ware, which contains MSP430 libraries and the TI Resource Explorer. These are already included in a CCS installation (unless the user selected otherwise).

4.2 LaunchPad Development Kit Websites

More information about the LaunchPad development kit, supported BoosterPack plug-in modules, and available resources can be found at:

- LP-MSP430FR2476 tool folder: resources specific to this particular LaunchPad development kit
- TI's LaunchPad development kit portal: information about all LaunchPad development kits from TI

4.3 MSP430Ware and TI Resource Explorer

TI Resource Explorer is a tool integrated into CCS that allows you to browse through available design resources. TI Resource Explorer will help you quickly find what you need inside packages including MSP430Ware, ControlSuite, TivaWare and more. TI Resource Explorer is well organized to find everything that you need quickly, and you can import software projects into your workspace in one click.

TI Resource Explorer Cloud is one of the TI Cloud Development tools, and is tightly integrated with CCS Cloud. See Section 4.1.1 for more information.

MSP430Ware is a collection of code examples, software libraries, data sheets and other design resources for all MSP430 devices delivered in a convenient package – essentially everything developers need to become MSP experts.

In addition to providing a complete collection of existing MSP430 design resources, MSP430Ware also includes a high-level API called MSP Driver Library. This library makes it easy to talk to MSP430 hardware. For more information, visit MSP430Ware for MSP Microcontrollers.

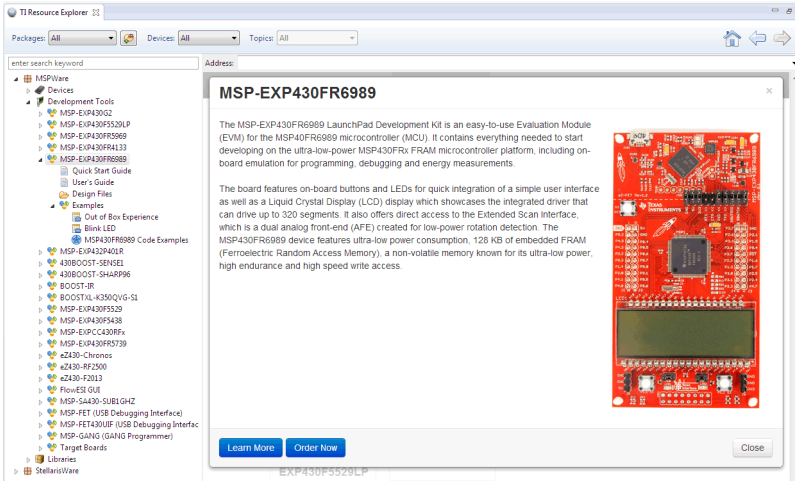


Figure 15. Using TI Resource Explorer to Browse LP-MSP430FR2476 in MSP430Ware

Copyright © 2019, Texas Instruments Incorporated

In TI Resource Explorer, you can find these examples and many more and can easily import them into CCS with one click.

4.4 FRAM Utilities

The Texas Instruments FRAM Utilities is a collection of embedded software utilities that leverage the ultra-low-power and virtually unlimited write endurance of FRAM. The utilities are available for MSP430FRxx FRAM microcontrollers and provide example code to help start application development.

4.4.1 Compute Through Power Loss

Compute Through Power Loss is a utility API set that enables ease of use with LPMx.5 low-power modes and a powerful shutdown mode that allows an application to save and restore critical system components when a power loss is detected.

4.4.2 Nonvolatile Storage (NVS)

The nonvolatile storage (NVS) library makes handling of nonvolatile data easy and robust against intermittent power loss or asynchronous device resets. To keep data storage constant, the nonvolatile storage library contains functions that store data in a way that is guaranteed to recover the last valid entry without data corruption.

4.5 MSP430FR2476 MCU

4.5.1 Device Documentation

At some point, you will probably want more information about the MSP430FR2476 device. For every MSP device, the documentation is organized as shown in Table 8.

Table 8. How MSP Device Documentation is Organized

Document	For MSP430FR2476	Description
Device family user's guide	MSP430FR4xx and MSP430FR2xx Family User's Guide	Architectural information about the device, including all modules and peripherals such as clocks, timers, ADC, and so on
Device-specific data sheet	MSP430FR247x Mixed-Signal Microcontrollers Data Sheet	Device-specific information and all parametric information for this device

4.5.2 MSP430FR2476 Code Examples

MSP430FR267x, MSP430FR247x code examples is a set of simple C examples that demonstrate how to use the entire set of peripherals on the MSP430FR2476 MCU, including serial communication, ADC10, timers, and others, through direct register access. Every MSP derivative has a set of these code examples. When starting a new project or adding a new peripheral, these examples serve as a great starting point.

4.5.3 MSP430 Application Notes and TI Designs

Many application notes can be found at www.ti.com/msp430. Also see TI reference designs for practical design examples and topics.

4.6 Community Resources

4.6.1 TI E2E™ Community

Search the forums at e2e.ti.com. If you cannot find your answer, post your question to the community.

4.6.2 Community at Large

Many online communities focus on the LaunchPad development kit—for example, www.43oh.com. You can find additional tools, resources, and support from these communities.

Copyright © 2019, Texas Instruments Incorporated

5 FAQ

Q: I can’t get the backchannel UART to connect. What’s wrong?

A: Check the following:

- Do the baud rate in the host terminal application and the eUSCI settings match?
- Are the appropriate jumpers in place on the isolation jumper block?
- Probe on RXD and send data from the host. If you don’t see data, it might be a problem on the host side.
- Probe on TXD while sending data from the MSP430 MCU. If you don’t see data, it might be a configuration problem with the eUSCI module.
- Consider the use of the hardware flow control lines (especially for higher baud rates).

Q: The MSP430G2 LaunchPad development kit had a socket, allowing me change the target device. Why doesn’t this LaunchPad development kit use one?

A: The target device on this LaunchPad development kit does not come in a dual in-line package. Sockets for the available device package are too expensive for this kit’s target price.

Copyright © 2019, Texas Instruments Incorporated

6 Schematics

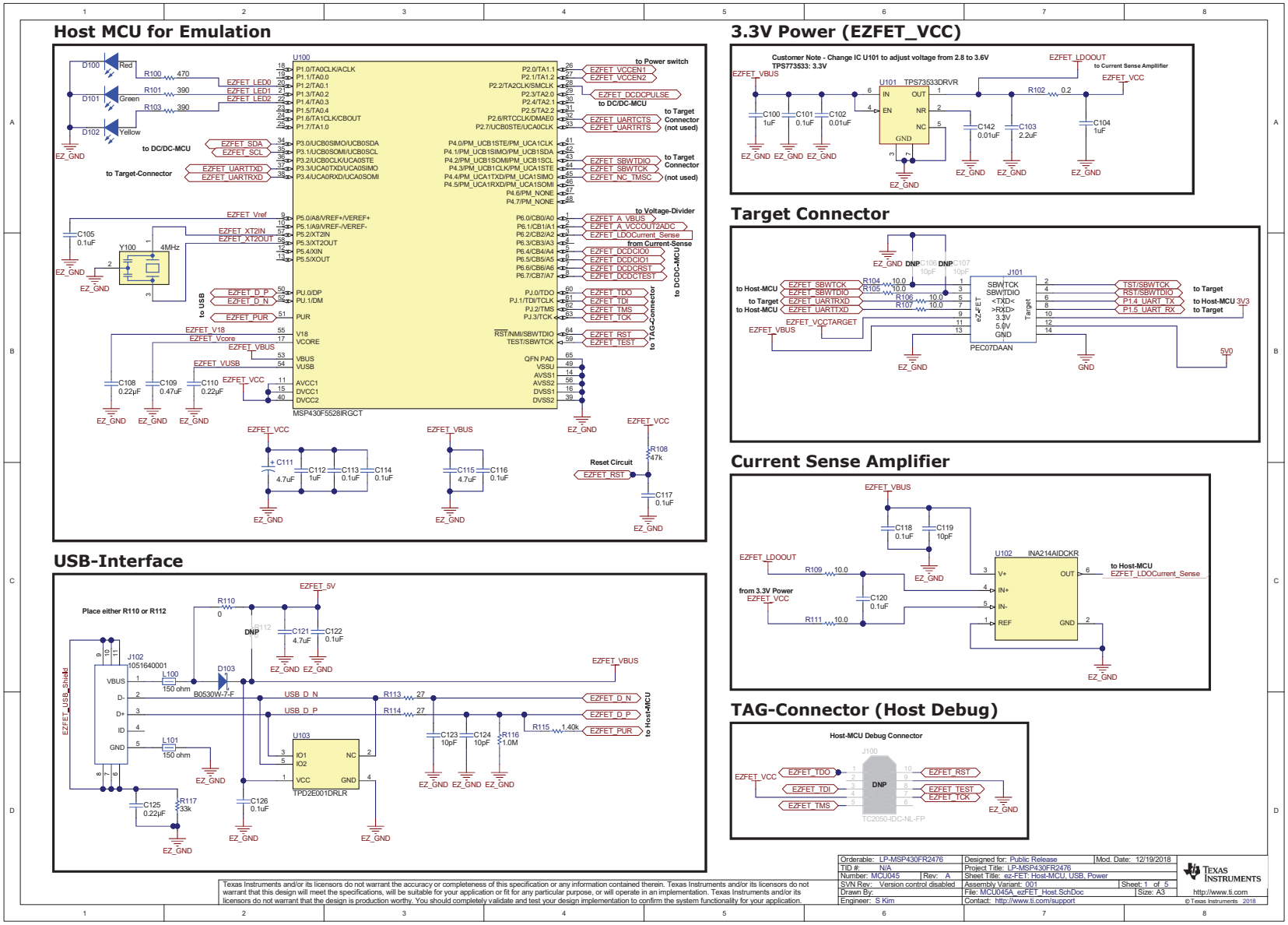


Figure 16. Schematics (1 of 4)

Copyright © 2019, Texas Instruments Incorporated

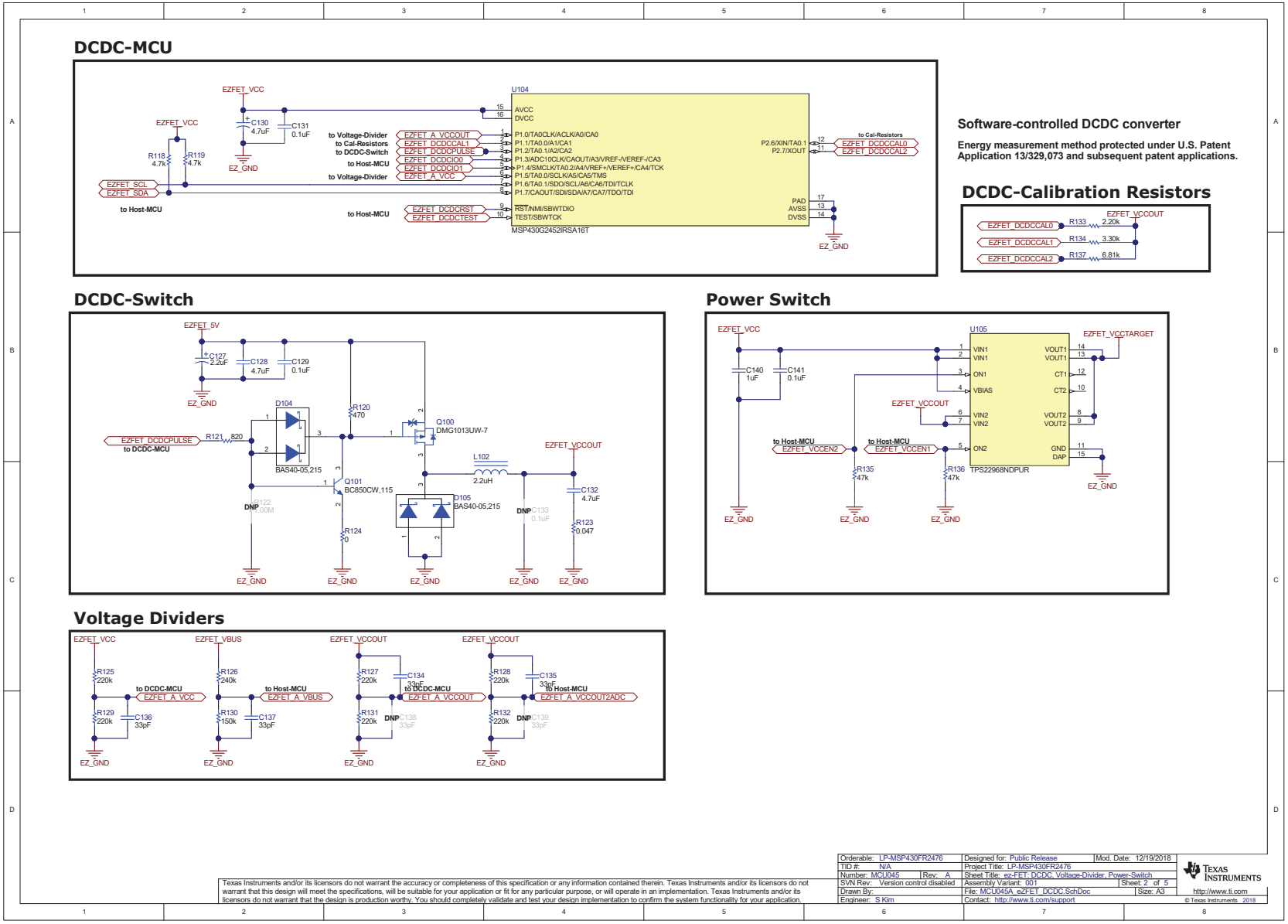


Figure 17. Schematics (2 of 4)

Copyright © 2019, Texas Instruments Incorporated

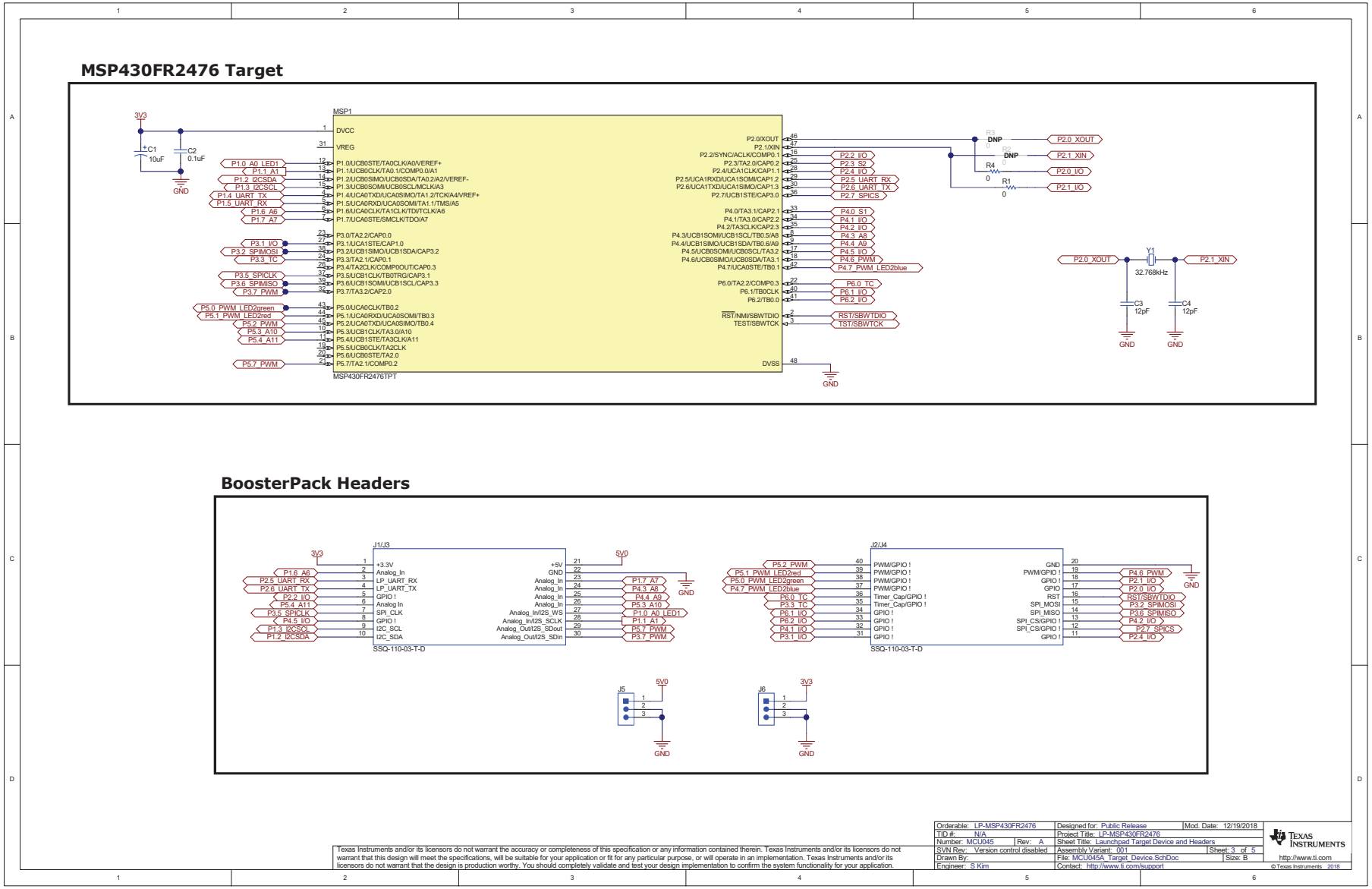


Figure 18. Schematics (3 of 4)

Copyright © 2019, Texas Instruments Incorporated

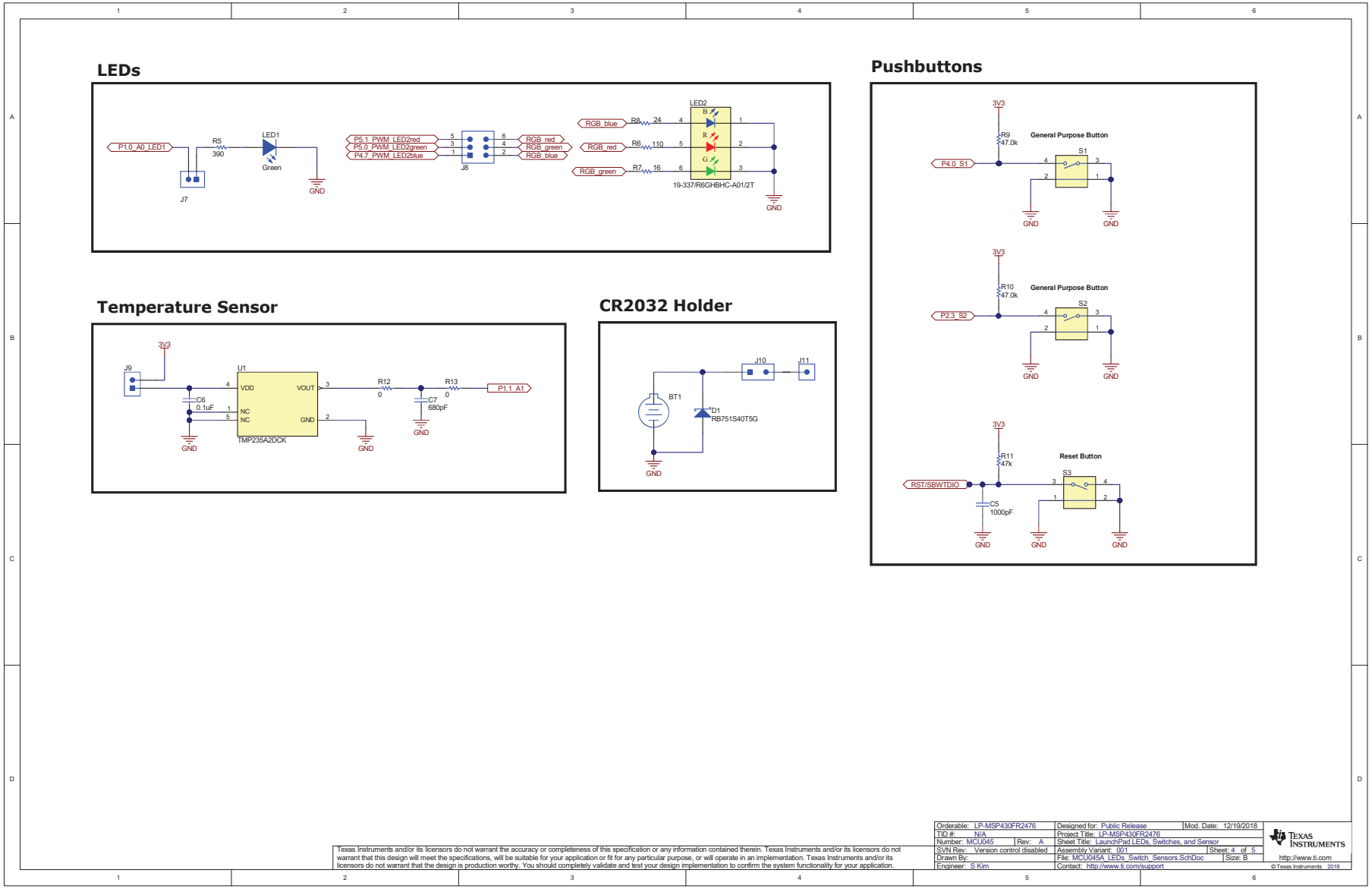


Figure 19. Schematics (4 of 4)

Copyright © 2019, Texas Instruments Incorporated

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) or other applicable terms available either on 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