

MSP430FR2433 LaunchPad™ Development Kit (MSP-EXP430FR2433)

The MSP-EXP430FR2433 LaunchPad™ Development Kit is an easy-to-use evaluation module (EVM) based on the [MSP430FR2433](#) Value Line Sensing microcontroller (MCU). It contains everything needed to start developing on the ultra-low-power MSP430FR2x Value Line Sensing MCU platform, including onboard debug probe for programming, debugging, and energy measurements. The board includes two buttons and two LEDs for creating a simple user interface. It also supports using a supercapacitor (must be purchased and installed by the user) that acts like a rechargeable battery, enabling stand-alone applications without an external power supply.

Figure 1 shows the MSP-EXP430FR2433 LaunchPad development kit.

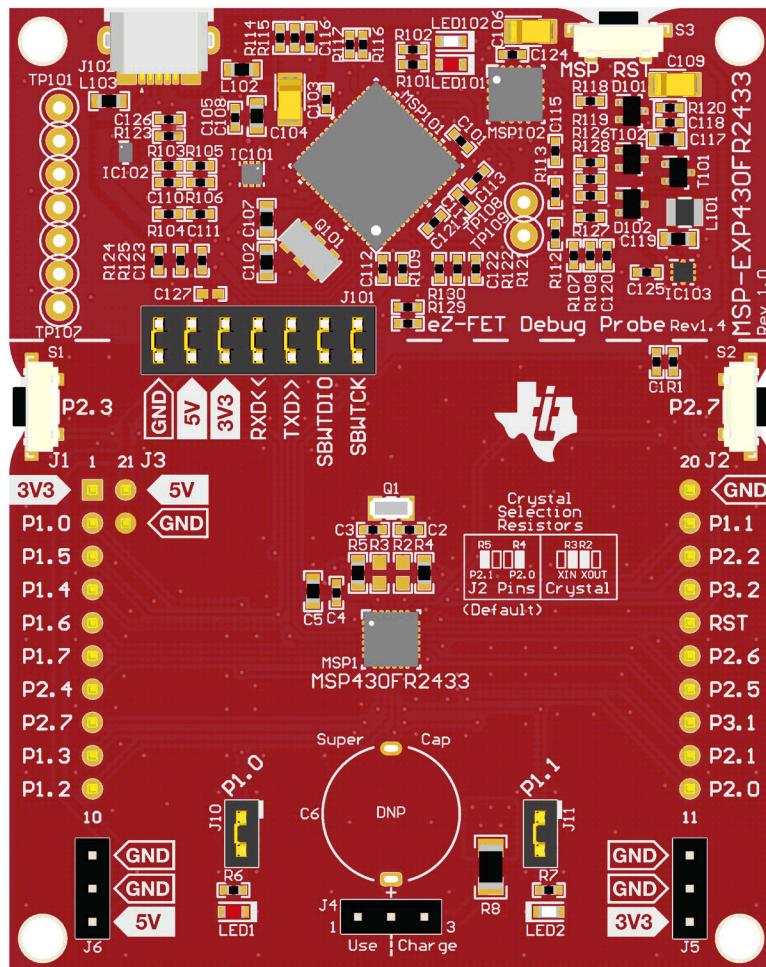


Figure 1. MSP-EXP430FR2433 LaunchPad Development Kit

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	4
1.5	Next Steps: Looking Into the Provided Code	4
2	Hardware.....	5
2.1	Block Diagram.....	5
2.2	Hardware Features	6
2.3	Power	10
2.4	Measure Current Draw of the MSP430 MCU.....	12
2.5	Clocking	12
2.6	Using the eZ-FET Debug Probe With a Different Target	12
2.7	BoosterPack Pinout	13
2.8	Design Files	14
2.9	Hardware Change log	14
3	Software Examples	14
3.1	Out-of-Box Software Example	15
3.2	Blink LED Example.....	16
4	Resources	16
4.1	Integrated Development Environments.....	16
4.2	LaunchPad Websites	19
4.3	MSPWare and TI Resource Explorer.....	20
4.4	FRAM Utilities.....	20
4.5	MSP430FR2433 MCU	21
4.6	Community Resources	21
5	FAQ	21
6	Schematics	22

List of Figures

1	MSP-EXP430FR2433 LaunchPad Development Kit.....	1
2	MSP-EXP430FR2433 Overview	5
3	MSP-EXP430FR2433 Block Diagram.....	5
4	MSP430FR2433 Pinout.....	6
5	eZ-FET Debug Probe	7
6	eZ-FET Isolation Jumper Block Diagram	8
7	Application Backchannel UART in Device Manager	9
8	MSP-EXP430FR2433 Power Block Diagram.....	10
9	MSP-EXP430FR2433 Supercap Power Block Diagram	11
10	BoosterPack Checker Tool	13
11	LaunchPad Kit to BoosterPack Module Connector Pinout.....	14
12	TI Resource Explorer Cloud	17
13	CCS Cloud	17
14	Directing the Project>Import Function to the Demo Project	18
15	When CCS Has Found the Project	19
16	Using TI Resource Explorer to Browse MSP-EXP430FR2433 in MSPWare	20
17	Schematics (1 of 2)	22
18	Schematics (2 of 2)	23

List of Tables

1	EnergyTrace Technology	7
---	------------------------------	---

2	Isolation Block Connections	8
3	Hardware Change Log	14
4	Software Examples	14
5	IDE Minimum Requirements for MSP-EXP430FR2433	14
6	Source File and Folders	15
7	Source File and Folders	16
8	How MSP Device Documentation is Organized	21

Trademarks

LaunchPad, BoosterPack, Code Composer Studio, EnergyTrace, MSP430, E2E are trademarks of Texas Instruments.

IAR Embedded Workbench is a registered trademark of IAR Systems.

All other trademarks are the property of their respective owners.

1 Getting Started

1.1 *Introduction*

The 16-MHz MSP430FR2433 device features 15.5KB of embedded FRAM (ferroelectric random access memory), a nonvolatile memory known for its ultra-low power, high endurance, and high-speed write access. Combined with the 4KB of on-chip SRAM, users have access to 15.5KB of memory to split between their program and data as required. For example, a data logging application may require a large data memory with a relatively small program memory, so the memory may be allocated as required between program and data memory.

Rapid prototyping is simplified by the 20-pin BoosterPack™ plug-in module headers, which support a wide range of available BoosterPack 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 MSP430FR2433 LaunchPad development kit.

1.2 **Key Features**

- MSP ULP FRAM technology based MSP430FR2433 16-bit MCU
- EnergyTrace Technology available for ultra-low-power debugging
- 20-pin LaunchPad development kit standard leveraging the BoosterPack plug-in module ecosystem
- Onboard eZ-FET debug probe
- 2 buttons and 2 LEDs for user interaction

1.3 **What's Included**

1.3.1 **Kit Contents**

- 1 MSP-EXP430FR2433 LaunchPad development kit
- 1 Micro USB cable
- 1 Quick start guide
- The supercapacitor is not included and must be provided by the user

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 to a computer using the included USB cable. A green power LED should illuminate. For proper operation, drivers are needed. It is recommended to get drivers by installing an IDE such as TI's CCS or IAR EW430. Drivers are also available at www.ti.com/MSPdrivers.

1.4.2 Running the Out-of-Box Demo

The out-of-box (OOB) demo for the MSP-EXP430FR2433 LaunchPad development kit demonstrates how to setup a periodic temperature data logger, by using a ring buffer inside the FRAM memory of the MSP430FR2433 MCU. Alternatively, the demo also implements a real-time temperature sensor.

Both the logged and the real-time temperature data can be transmitted to the PC and visualized using the accompanying cloud GUI (visit [MSP-EXP430FR2433 OOB GUI](#)). If access to the cloud GUI is not available, the data can still be observed using any serial terminal application (application UART settings: 115200, 8, 1, n).

By default after power up, the LaunchPad development kit enters the FRAM Data Log mode. The red LED1 blinks periodically (approximately every 5 seconds), which indicates that the device is waking up to log the temperature and going back to sleep. Press the left user button S1 to transfer the stored temperature to the PC.

Next, try pressing the S1 and S2 buttons simultaneously to enter the Live Temperature mode. The LaunchPad development kit should start streaming live temperature data to the PC to be visualized in the MSP-EXP430FR2433 OOB GUI or displayed in a serial terminal. In this mode, the application also keeps track of a temperature threshold (the default is 25°C), and when a new temperature data is acquired, it is compared against that threshold. If the current temperature is above the threshold, the red LED1 illuminates, and if the current temperature is below the threshold, the green LED2 illuminates. Pressing S1 or S2 independently increases or decreases the temperature threshold in this mode.

The user can influence the temperature of the device by blowing hot or cold air and observing the changes in the user LED brightness or see data changes on the GUI.

1.5 Next Steps: Looking Into the Provided Code

After the EVM features have been explored, the fun can begin. It is 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 MSPWare 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 USB 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 MSP-EXP430FR2433 hardware.

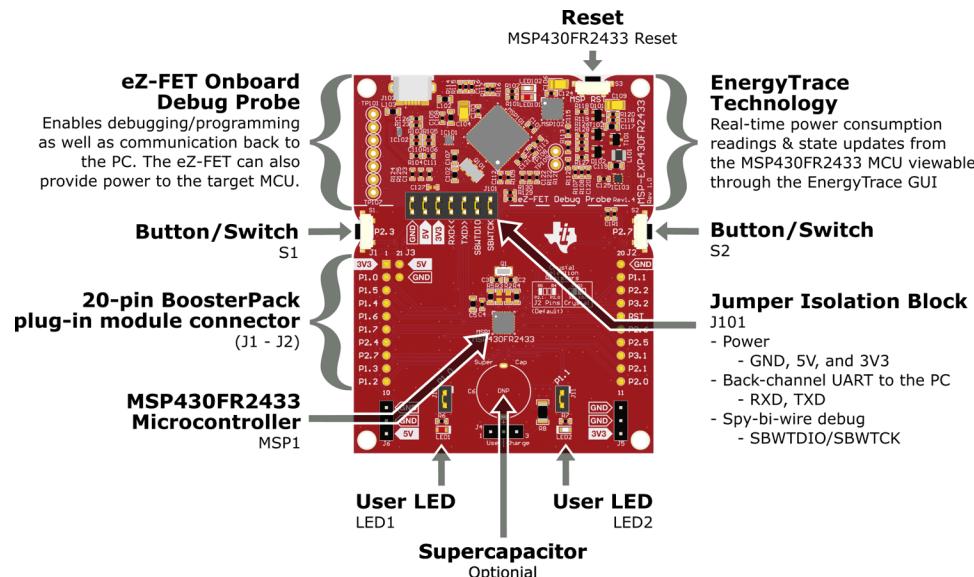


Figure 2. MSP-EXP430FR2433 Overview

2.1 Block Diagram

Figure 3 shows the block diagram.

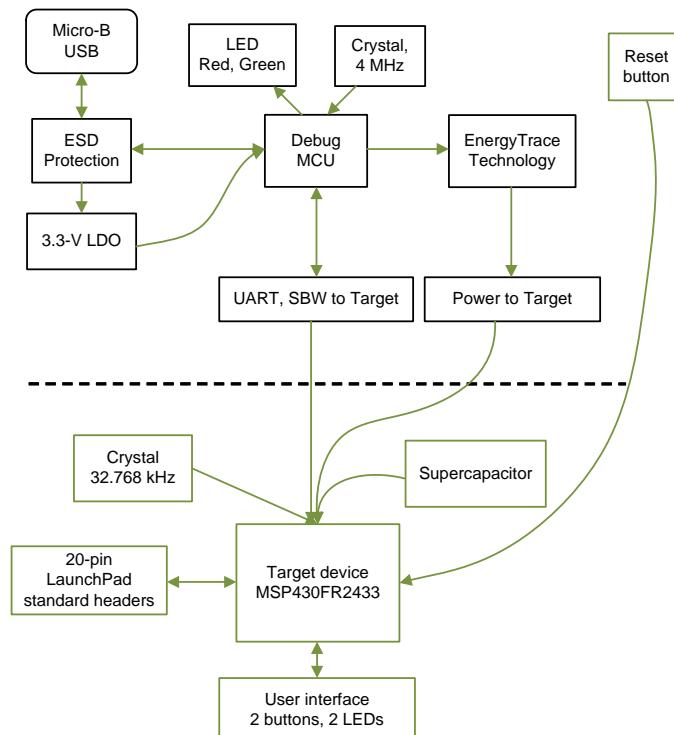


Figure 3. MSP-EXP430FR2433 Block Diagram

2.2 Hardware Features

2.2.1 MSP430FR2433 MCU

The MSP430FR2433 is an ultra-low-power MSP430FRx FRAM-based microcontroller (MCU), which offer extended data logging and security capabilities. The MSP430FR2433 offers the small VQFN package (4 mm x 4 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. More information on FRAM can be found at 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
- 15KB of program FRAM, 512B of information FRAM, and 4KB of RAM
- 8-channel 10-bit ADC
- Four 16-Bit timers
 - Two timers with three capture/compare registers each (Timer_A3)
 - Two timers with two capture/compare registers each (Timer_A2)
- 32-bit hardware multiplier (MPY)
- 19 GPIOs
- Two enhanced universal serial communication interfaces (eUSCI_A) support UART, IrDA, and SPI
- One eUSCI (eUSCI_B) supports SPI and I²C

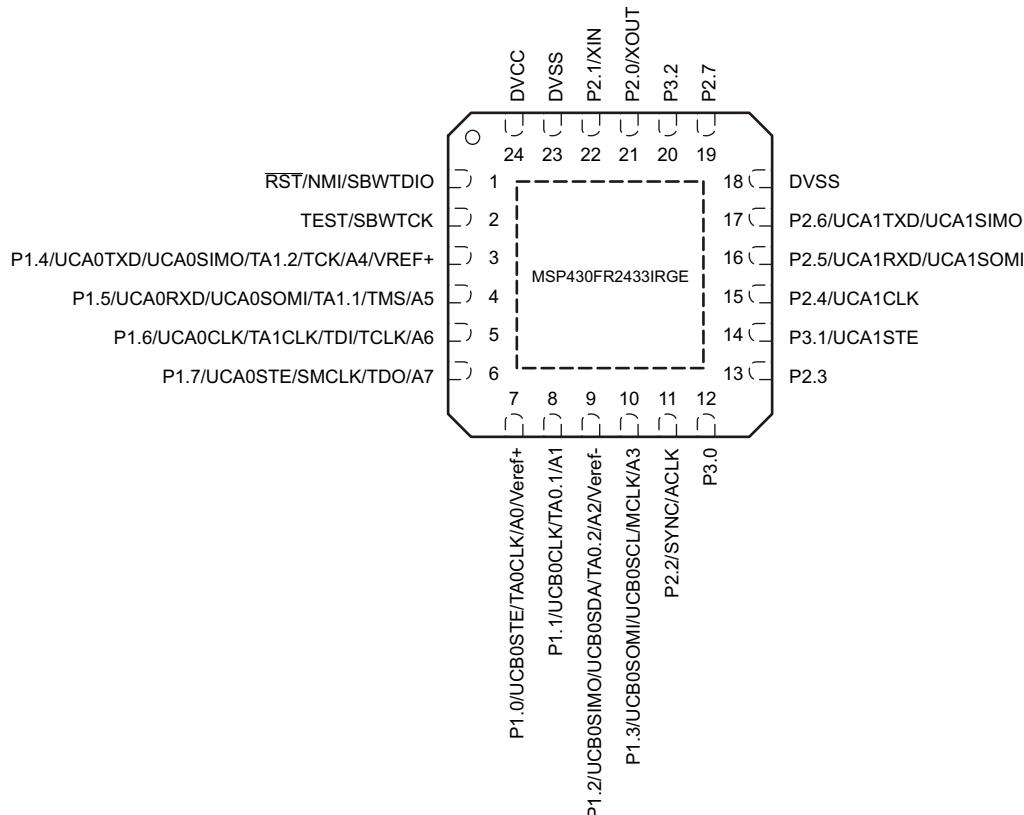


Figure 4. MSP430FR2433 Pinout

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 MSP-EXP430FR2433 has the eZ-FET debug probe (see [Figure 5](#)), which is a simple and low-cost debugger that supports all MSP430™ device derivatives.

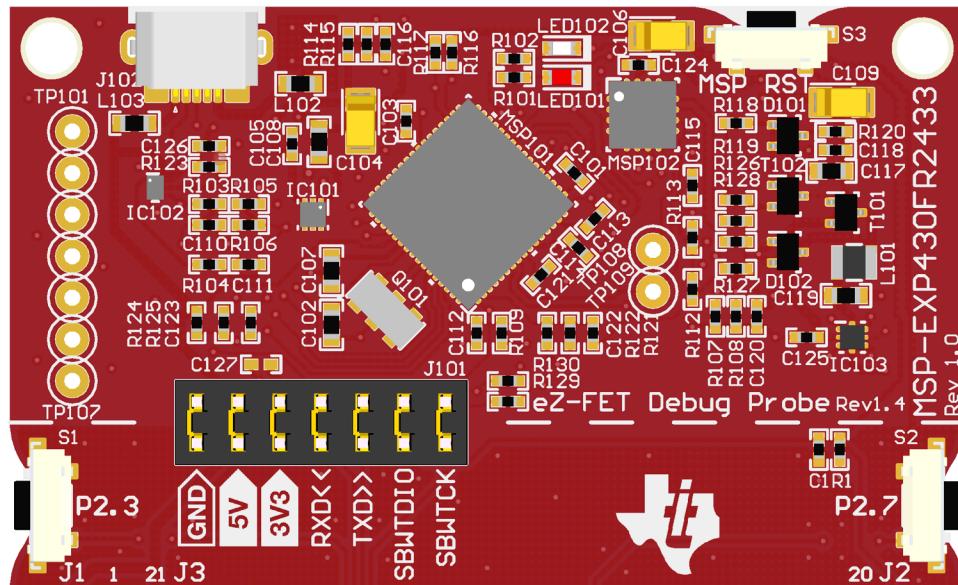


Figure 5. eZ-FET Debug Probe

The MSP-EXP430FR2433 LaunchPad development kit features EnergyTrace technology but does not have support for EnergyTrace++ technology. The EnergyTrace functionality varies across the MSP portfolio.

Table 1. EnergyTrace Technology

Features	EnergyTrace Technology	EnergyTrace++ Technology
Current monitoring	Yes	Yes
CPU state	No	Yes
Peripheral and system state	No	Yes
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, see [Section 2.2.4](#).

The eZ-FET hardware can be found in the schematics in [Section 6](#) and in the [MSP-EXP430FR2433 design files download page](#). The software and more information about the debugger can be found on the eZ-FET wiki. More information on the eZ-FET can be found in the [MSP Debuggers User's Guide](#).

2.2.3 Debug Probe Connection: Isolation Jumper Block

The isolation jumper block at jumper J101 allows the user to connect or disconnect signals that cross from the eZ-FET domain into the MSP430FR2433 target domain. This includes eZ-FET Spy-Bi-Wire signals, application UART signals, and 3.3-V and 5-V power. [Table 2](#) describes these connections.

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 other use than onboard debugging and application UART communication
- To expose the programming and UART interface of the eZ-FET so that it can be used for devices other than the onboard MCU. See [Section 2.6](#) for how to use the eZ-FET Debug Probe with a different target.

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
RXD <<	Backchannel UART: The target FR2433 receives data through this signal. The arrows indicate the direction of the signal.
TXD >>	Backchannel UART: The target FR2433 sends data through this signal. The arrows indicate the direction of the signal.
SBW RST	Spy-Bi-Wire debug: SBWTDO data signal. This pin also functions as the RST signal (active low).
SBW TST	Spy-Bi-Wire debug: SBWTCK clock signal. This pin also functions as the TST signal.

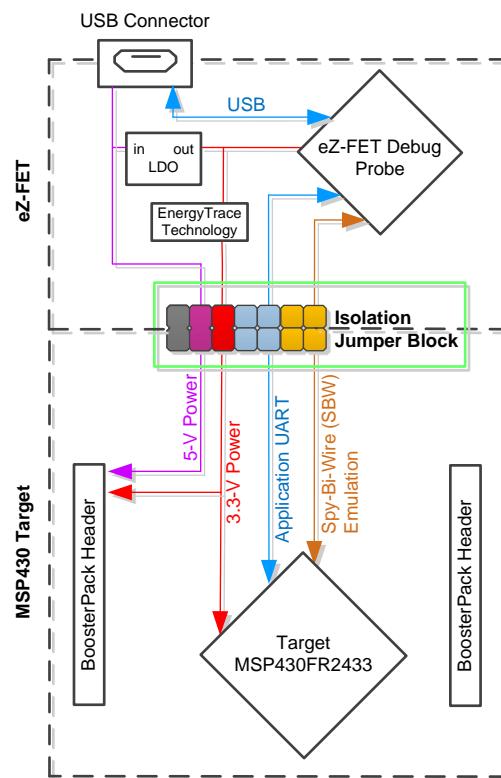


Figure 6. eZ-FET Isolation Jumper Block Diagram

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 PC host side. This can be used to create graphical user interfaces (GUIs) and other programs on the PC that communicate with the LaunchPad.

[Figure 6](#) shows the pathway of the backchannel UART. The backchannel UART is the UART on eUSCI_A0.

On the host side, a virtual COM port for the application backchannel UART is generated when the LaunchPad 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

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 MSP430FR2433 side, the backchannel is connected to the eUSCI_A0 module. The eZ-FET has a configurable baud rate; therefore, it is important that the PC application configures the baud rate to be the same as what is configured on the eUSCI_A0.

2.2.5 Optional Features

2.2.5.1 Supercapacitor

A through-hole component footprint is available on the board and allows user to populate a supercapacitor to power the system without any external power. The recommended part is the Panasonic EEC-S0HD224H 220 mF (0.22 F) supercapacitor and can be purchased from major electronic component distributors.

NOTE: A supercapacitor is not included in the kit and must be supplied by the user.

Using the onboard jumper headers, the supercapacitor can be configured in the following ways: charging, using (direct connection to 3V3 rail), or disconnected. For more details on these use modes and how to use them, see [Section 2.3](#).

2.3 Power

The board was designed to accommodate various powering methods, including through the onboard eZ-FET as well as from external or BoosterPack power. Figure 8 shows power from the eZ-FET and power from external source through header J5 or the BoosterPack module.

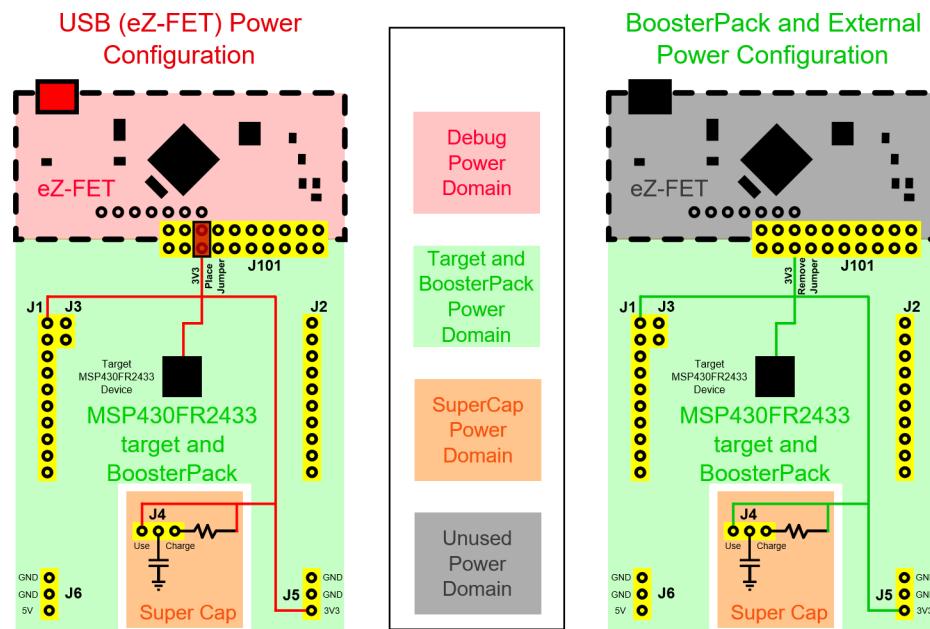


Figure 8. MSP-EXP430FR2433 Power Block Diagram

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. 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.

2.3.2 BoosterPack and External Power Supply

Header J5 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 MSP430FR2433 has an operating range of 1.8 V to 3.6 V. More information can be found in the [MSP430FR2433 Mixed-Signal Microcontroller](#) data sheet.

2.3.3 Supercap (C6)

If the supercapacitor is populated, it can be used to evaluate the ultra-low power feature of the MSP430FR2433 target device. See how long you can run your application on the supercap alone.

NOTE: A supercapacitor is not included in the kit and must be supplied by the user.

Figure 9 shows charging the supercap and powering directly from it.

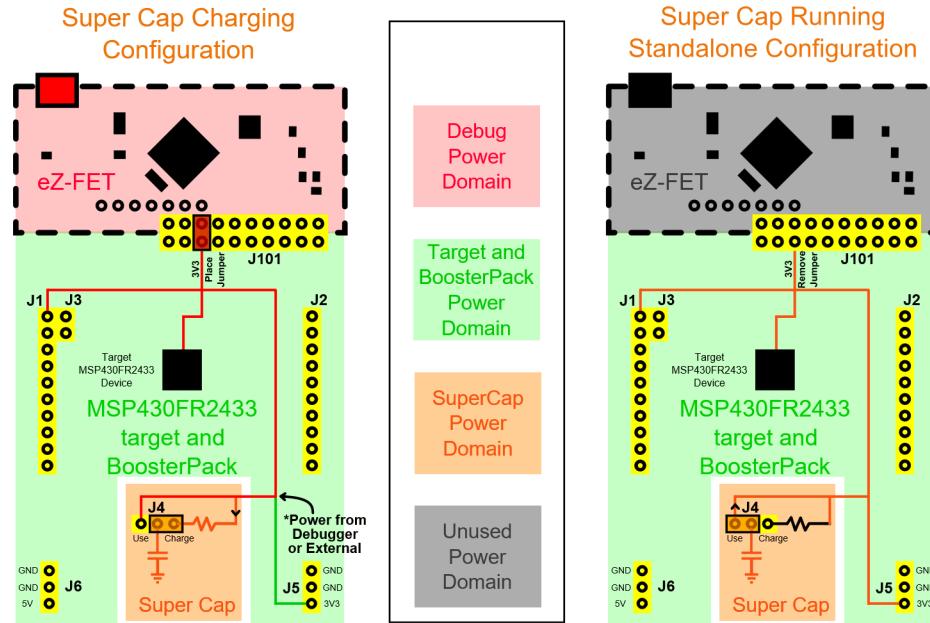


Figure 9. MSP-EXP430FR2433 Supercap Power Block Diagram

2.3.3.1 Charging the Supercap

The supercapacitor can be charged when the EVM is plugged into the PC or when the board is externally powered. During charging, set J4 to the “Charge” setting, this adds in a current limiting resistor for charging.

To charge the supercap, power must be coming from the eZ-FET debug probe, external power through J5, or a BoosterPack module powering through J1. Allow two to three minutes for the supercap to charge (time may vary depending on initial charge of the supercap and your power source) to full V_{CC} .

2.3.3.2 Using the Supercap

After charging of the supercapacitor, you can move the J4 jumper to the *Use* setting and then unplug power. This connects the supercapacitor to the 3V3 rail without the charging resistor in between. At this point, the LaunchPad kit is being powered completely by the C6 supercapacitor.

For lowest power operation, make sure to disconnect the J101 jumpers so that the eZ-FET is not also powered by the C6 supercap.

2.3.3.3 Disabling the Supercap

The supercapacitor can be completely decoupled from the board by removing the J4 jumper. Hanging this jumper off only one pin can prevent losing the jumper.

2.4 Measure Current Draw of the MSP430 MCU

To measure the current draw of the MSP430FR2433 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 headers.

To measure ultra-low power, follow these steps:

1. Remove the 3V3 jumper in the J101 isolation block, and attach an ammeter across this jumper.
2. Consider the effect that the backchannel UART and any circuitry attached to the MSP430FR2433 may 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.
3. Make sure there are no floating inputs/outputs (I/Os) on the MSP430FR2433. 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.
4. Begin target execution.
5. 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 can also be used to compare various current profiles and better optimize your energy performance!

2.5 Clocking

The MSP-EXP430FR2433 provides external clocks in addition to the internal clocks in the device.

- Q1: 32-kHz Epson crystal (FC-135R) 12.5-pF crystal, part number: X1A0001410014

The 32-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.

By default, the crystal is not connected to the MSP430FR2433 because the target pins are multiplexed with 2 BoosterPack header pins. $0\text{-}\Omega$ resistors R4 and R5 need to be removed, while R2 and R3 must be shorted across, to connect the external crystal to the MSP430FR2433. See the onboard crystal selection resistors silkscreen for how to configure the resistors to select between the crystal or the BoosterPack pins.

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

- MCLK: DCO 1 MHz
- SMCLK: DCO 1 MHz
- ACLK: REFO 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 can interface to most MSP430 derivative devices, not just the onboard MSP430FR2433 target device.

To do this, disconnect every jumper 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 RST/SBWTDO cannot be greater than 2.2 nF. The documentation for designing MSP430 JTAG interface circuitry is the [MSP430 Hardware Tools User's Guide](#).

Finally, wire together these 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
- SBWTDO

- 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 Pinout

The LaunchPad kit adheres to the 20-pin LaunchPad pinout standard. A standard was created to aid compatibility between LaunchPad kits and BoosterPack modules across the TI ecosystem.

While most BoosterPack modules are compliant with the standard, some are not. The MSP-EXP430FR2433 LaunchPad kit is compatible with all 20-pin BoosterPack modules that are compliant with the standard. If the reseller or owner of the BoosterPack module does not explicitly indicate compatibility with the MSP-EXP430FR2433 LaunchPad development kit, compare the schematic of the candidate BoosterPack module with the LaunchPad kit to ensure compatibility. Keep in mind that sometimes conflicts can be resolved by changing the MSP430FR2433 device pin function configuration in software.

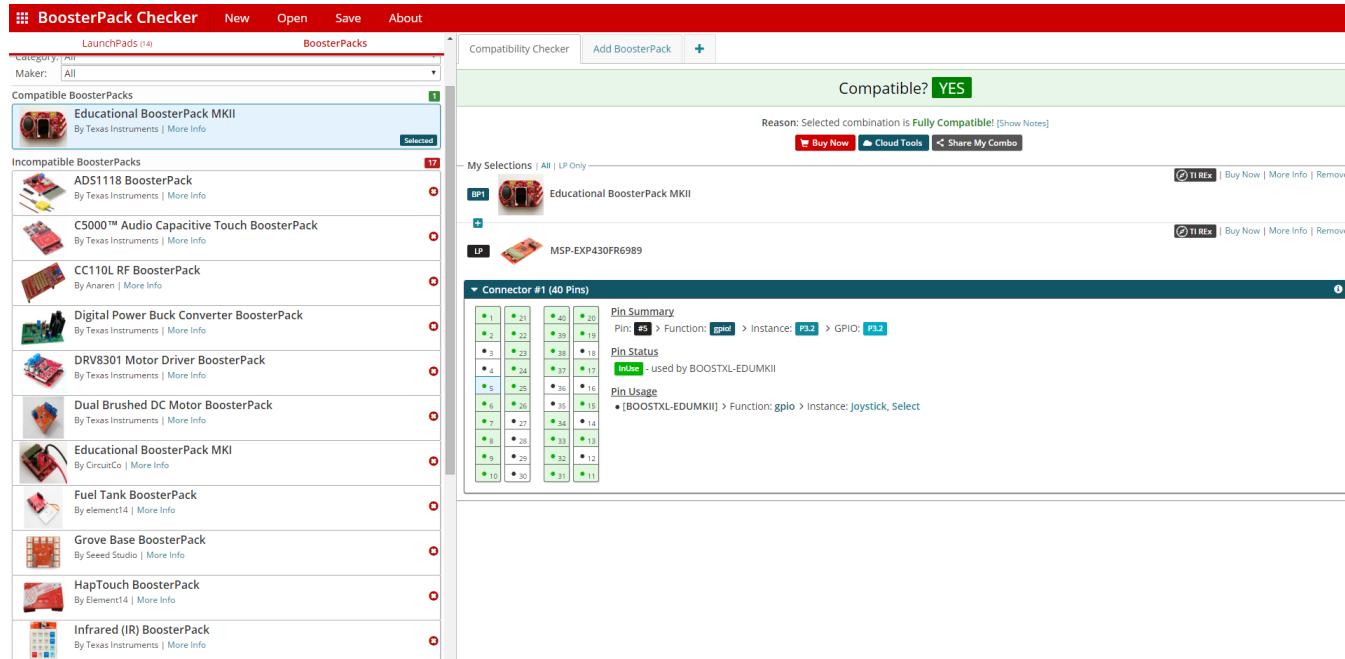


Figure 10. BoosterPack Checker Tool

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

Figure 11 shows the 20-pin pinout of the MSP430FR2433 LaunchPad development kit.

The inner side of the dashed line shows some of the software selectable functions on each pin, including the functions that conform to the standard. However, each pin may have other functionalities that can be configured by the software. See the [MSP430FR2433 data sheet](#) for more details on individual pin functions.

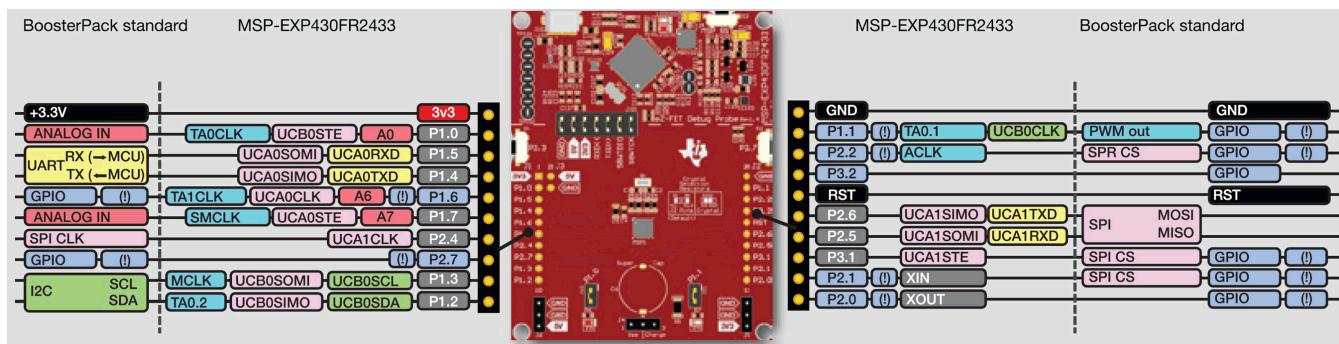


Figure 11. LaunchPad Kit to BoosterPack Module Connector Pinout

2.8 Design Files

2.8.1 Hardware

Section 6 shows the schematics. All design files including schematics, layout, bill of materials (BOM), Gerber files, and documentation are available on the [MSP-EXP430FR2433 design files download page](#).

2.8.2 Software

All design files including TI-TXT object-code firmware images, software example projects, and documentation are available on the [MSP-EXP430FR2433 design files download page](#).

2.9 Hardware Change Log

Table 3 lists the revision history of the MSP-EXP430FR2433 hardware.

Table 3. Hardware Change Log

PCB Revision	Description
Rev 1.0	Initial release

3 Software Examples

Four software examples are included with the MSP430FR2433 LaunchPad development kit (see Table 4), which can be found in the [MSP430FR2433 LaunchPad development kit download page](#) and are also available inside the MSPWare software.

Table 4. Software Examples

Demo Name	BoosterPack Required	Description	More Details
OutOfBox_FR2433	None	The out-of-box demo preprogrammed on the LaunchPad kit from the factory. Demonstrates features of MSP430FR2433 device	Section 3.1
BlinkLED_FR2433	None	Blinks an LED on the LaunchPad 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 MSP430FR2433 device (see Table 5).

Table 5. IDE Minimum Requirements for MSP-EXP430FR2433

Code Composer Studio IDE	IAR Embedded Workbench for Texas Instruments MSP430 IDE
Version 6.1.3 or later	Version 6.30 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 on the EVM.

The Out-of-Box demo for the MSP-EXP430FR2433 LaunchPad development kit demonstrates how to setup a periodic temperature data logger, by using a ring-buffer inside the FRAM memory of the MSP430FR2433 MCU. The demo also implements a real-time temperature sensor.

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
FRAMLogMode.c	Contains functions for the FRAM data logging mode
LiveTempMode.c	Contains function for the live temperature streaming mode
Library: driverlib	Device driver library
Library: fram-utilities	Contains the CTPL and NVS software libraries from the FRAM Utilities
Library: iqmathlib	Fixed Point Math Library for MSP
Library: jsmn	Third-party library for parsing JSON formatted strings

3.1.2 Overview

An online cloud GUI ([MSP-EXP430FR2433 OOB GUI](#)) can be used to download this demo to your board and visualize the temperature data. A serial terminal can also be used to display the data being sent from the demo to the PC (application UART settings: 115200, 8, 1, n).

Upon powering up the Out-of-Box demo, the green LED2 lights up for 1 second, and then the board enters FRAM data logging mode to measure and record the internal temperature of the MSP430FR2433 MCU every 5 seconds.

By default after power up, the LaunchPad development kit enters the FRAM data logging mode. At any time, press the S1 and S2 buttons simultaneously to switch between the live temperature mode and the FRAM data logging mode.

3.1.3 FRAM Data Logging Mode

This mode shows the FRAM data logging capabilities of the MSP430FR2433. After starting this mode, the LaunchPad wake up from LPM3.5 approximately every 5 seconds (indicated by short red LED1 blink) to log temperature values. Data are stored into a ring buffer located in the FRAM memory. When the ring buffer fills up to its capacity, the oldest temperature data is discarded and replaced with the newest data.

- Press the left button S1 to transfer all data stored inside the ring buffer to the PC.
- Press the right button S1 to clear all data and return to an empty ring buffer.
- The compute through power loss (CTPL) and nonvolatile storage (NVS) libraries from the FRAM Utilities are used in this demo to simplify the code to setup this simple data logging application.

3.1.4 Live Temperature Mode

In this mode, the LaunchPad kit repeatedly measures the internal temperature of the MSP430FR2433 MCU and transfers the data to the PC through UART.

The application also keeps track of a temperature threshold in this mode (default is 25°C), and when a new temperature data is acquired, it is compared against the threshold. If measured temperature is below the threshold, the green LED2 illuminates, and if the measured temperature is above the threshold, the red LED1 illuminates.

Independently pressing S1 or S2 increases or decreases, respectively, the temperature threshold in this mode. The further the recorded temperature is from the threshold, the brighter the corresponding LEDs illuminate.

The user can influence the temperature of the device by blowing hot or cold air and observing the changes in the user LED brightness or see data changes on the GUI.

3.2 **Blink LED Example**

This very simple software example shows how to software toggle a GPIO to blink an LED on the LaunchPad 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 simply 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.

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 are opened with a development environment like Code Composer Studio IDE and IAR Embedded Workbench IDE.

4.1.1 **TI Cloud Development Tools**

TI's Cloud-based software development tools provide instant access to MSPWare 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 MSPWare without having to download files to your local drive (see [Figure 12](#)).

Visit TI Resource Explorer Cloud now at dev.ti.com.

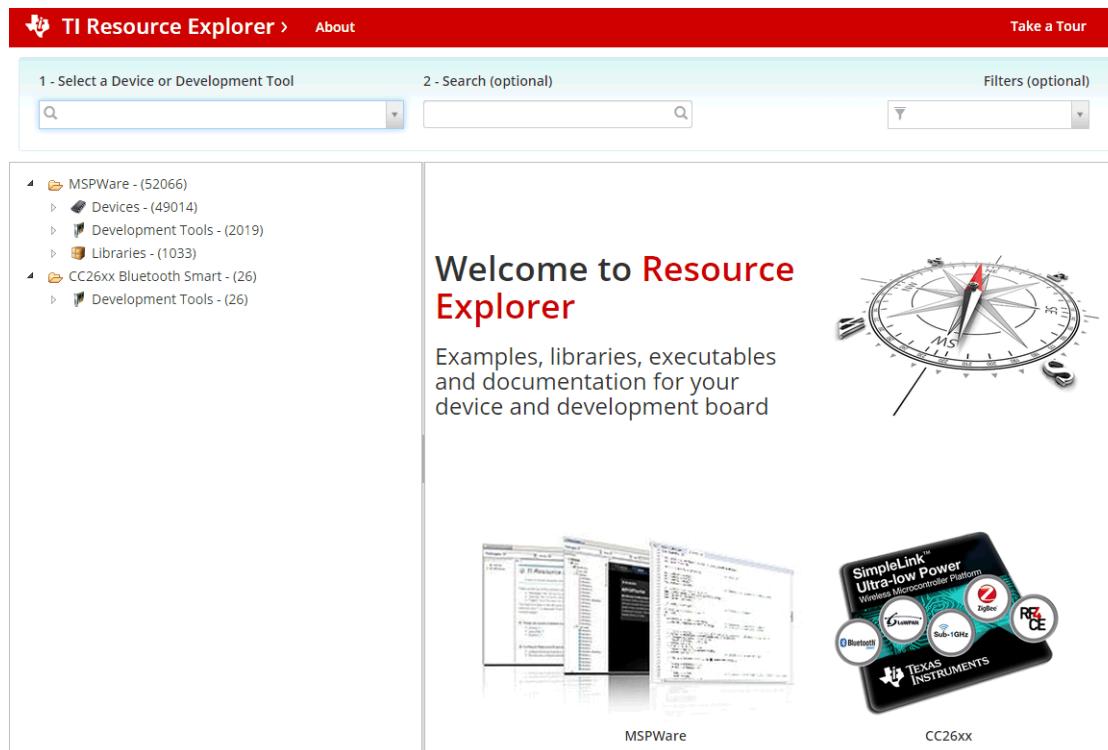


Figure 12. TI Resource Explorer Cloud

4.1.1.2 Code Composer Studio Cloud

Code Composer Studio Cloud (CCS Cloud) is a web-based IDE that enables you to quickly create, edit, build and debug applications for your 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 MSPWare software and Energia or develop your own application. CCS Cloud supports debug features such as execution control, breakpoints and viewing variables.

For a full comparison between CCS Cloud and CCS Desktop, visit the [TI Cloud Tools page](#).

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

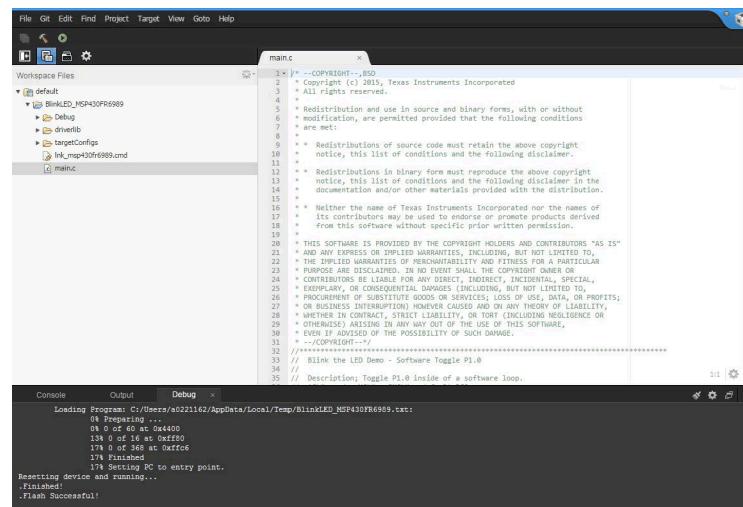


Figure 13. CCS Cloud

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 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 www.ti.com/tool/ccstudio.

CCS v7.0.0 or higher is required. When CCS has been launched, and a workspace directory chosen, use Project>Import Existing CCS Eclipse Project. Direct it to the desired demo's project directory that contains main.c.

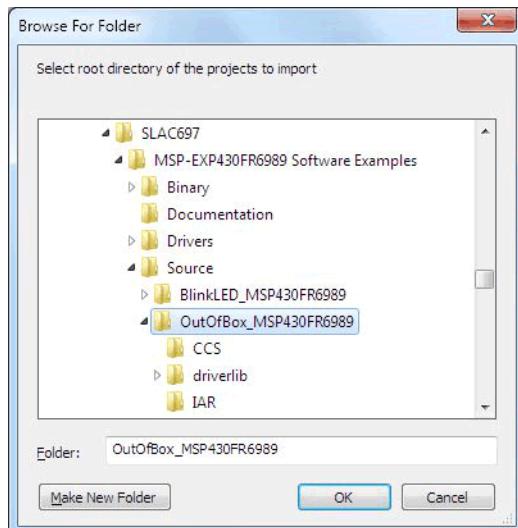


Figure 14. Directing the Project>Import Function to the Demo Project

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 15](#), and it has a checkmark to the left of it.

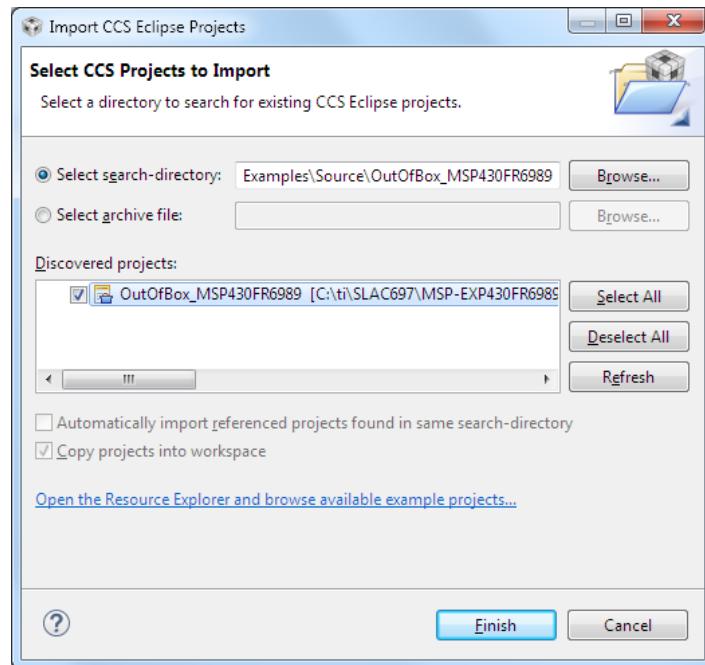


Figure 15. When CCS Has Found the Project

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, be sure to check the workspace directory on the file system.)

4.1.3 IAR Embedded Workbench for Texas Instruments 430

IAR Embedded Workbench for MSP430 is another very powerful integrated development environment that allows you to 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 supp.iar.com/Download/SW/?item=EW430-EVAL.

IAR 6.30 or higher is required. 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 may download and install MSPWare, 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 Websites

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

- [MSP-EXP430FR2433 Tool Folder](#): Resources specific to this particular LaunchPad development kit
- [TI's LaunchPad portal](#): Information about all LaunchPad development kits from TI

4.3 MSPWare 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 helps you quickly find what you need inside packages including MSPWare, 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.

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

In addition to providing a complete collection of existing MSP design resources, MSPWare also includes a high-level API called MSP Driver Library. This library makes it easy to program MSP hardware. For more information, see www.ti.com/tool/mspware.

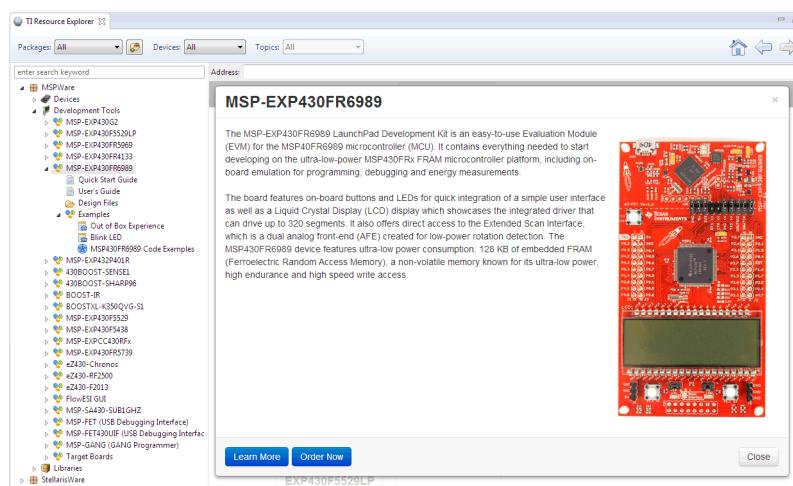


Figure 16. Using TI Resource Explorer to Browse MSP-EXP430FR2433 in MSPWare

Inside TI Resource Explorer, these examples and many more can be found, and easily imported into CCS with one click.

4.4 FRAM Utilities

The TI 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 (CTPL)

CTPL 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 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 ensures recovery of the last valid entry without data corruption.

4.5 MSP430FR2433 MCU

4.5.1 Device Documentation

At some point, you will probably want more information about the MSP430FR2433 device. For every MSP device, the documentation is organized as shown in [Table 8](#).

Table 8. How MSP Device Documentation is Organized

Document	For MSP430FR2433	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	MSP430FR2433 Mixed-Signal Microcontroller data sheet	Device-specific information and all parametric information for this device

4.5.2 MSP430FR2433 Code Examples

[MSP430FR243x, MSP430FR253x, MSP430FR263x Code Examples](#) is a set of simple C examples that demonstrate how to use the entire set of peripherals on the MSP430FR2433 MCU, including serial communication, ADC10, Timer, 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, along with TI Designs with 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 kits – for example, www.43oh.com. You can find additional tools, resources, and support from these communities.

5 FAQ

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

A: Check the following:

- See Section 5.8.2 of the [MSP Debuggers User's Guide](#) for supported baud rates when using the eZ-FET.
- Do the baud-rate settings in the terminal application on the host PC and the eUSCI 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 MSP. 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 MSP G2 LaunchPad kit has a socket that allows me change the target device. Why doesn't this LaunchPad kit use one?

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

6 Schematics

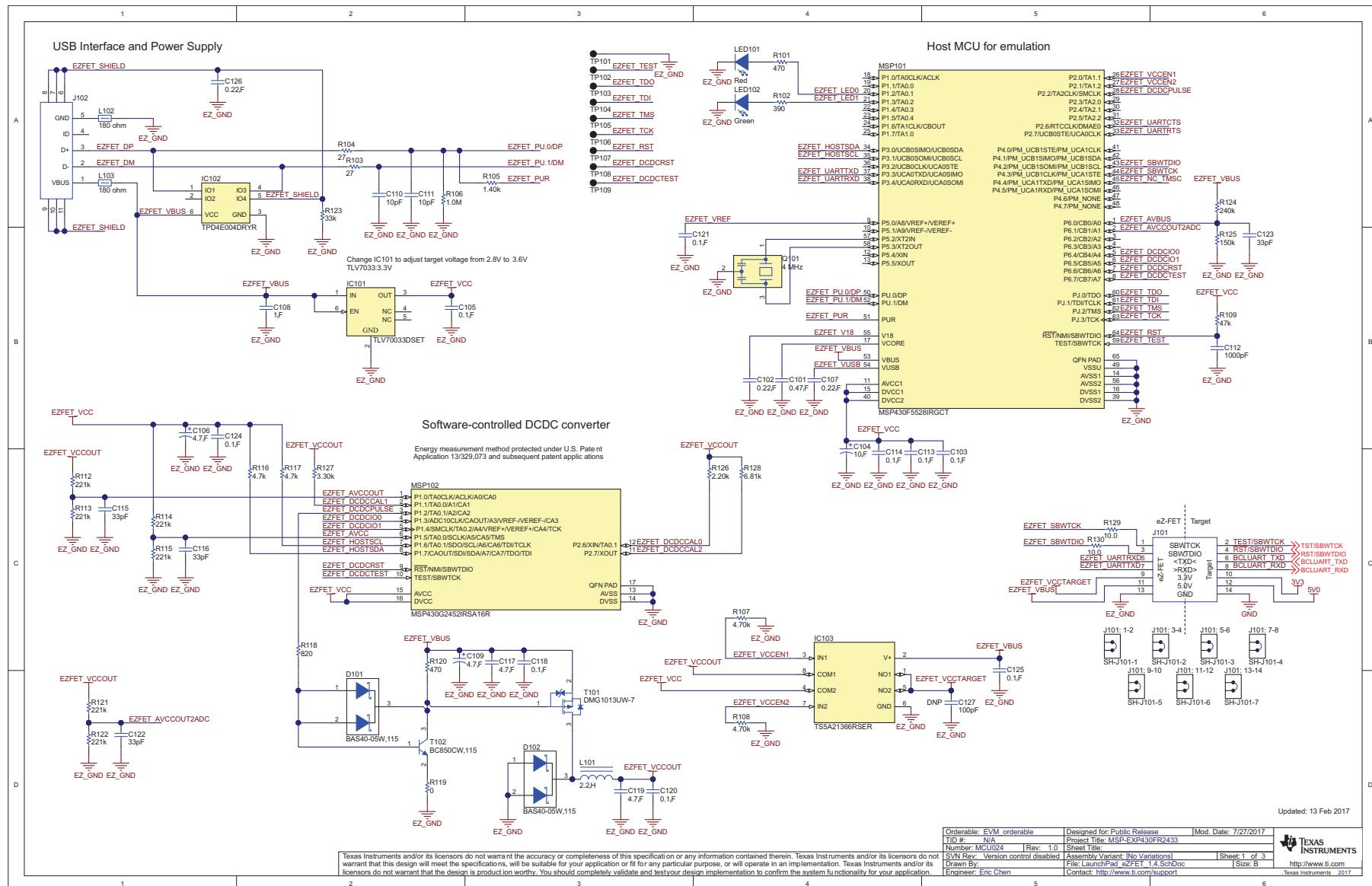
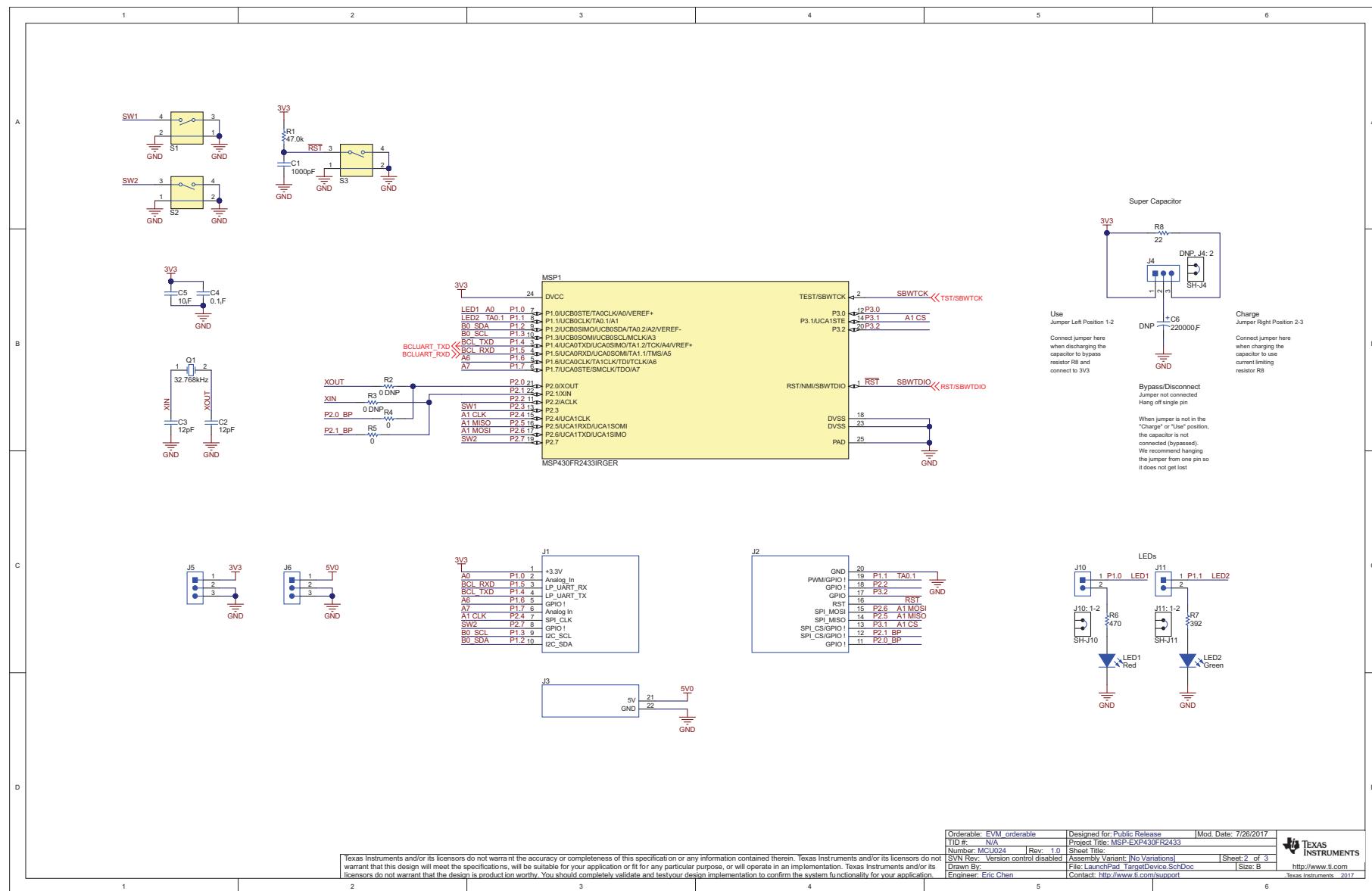


Figure 17. Schematics (1 of 2)



Copyright © 2016, Texas Instruments Incorporated

Figure 18. Schematics (2 of 2)

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, regulatory 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 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](#), [TI's General Quality Guidelines](#), 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. Unless TI explicitly designates a product as custom or customer-specified, TI products are standard, catalog, general purpose devices.

TI objects to and rejects any additional or different terms you may propose.

Copyright © 2026, Texas Instruments Incorporated

Last updated 10/2025