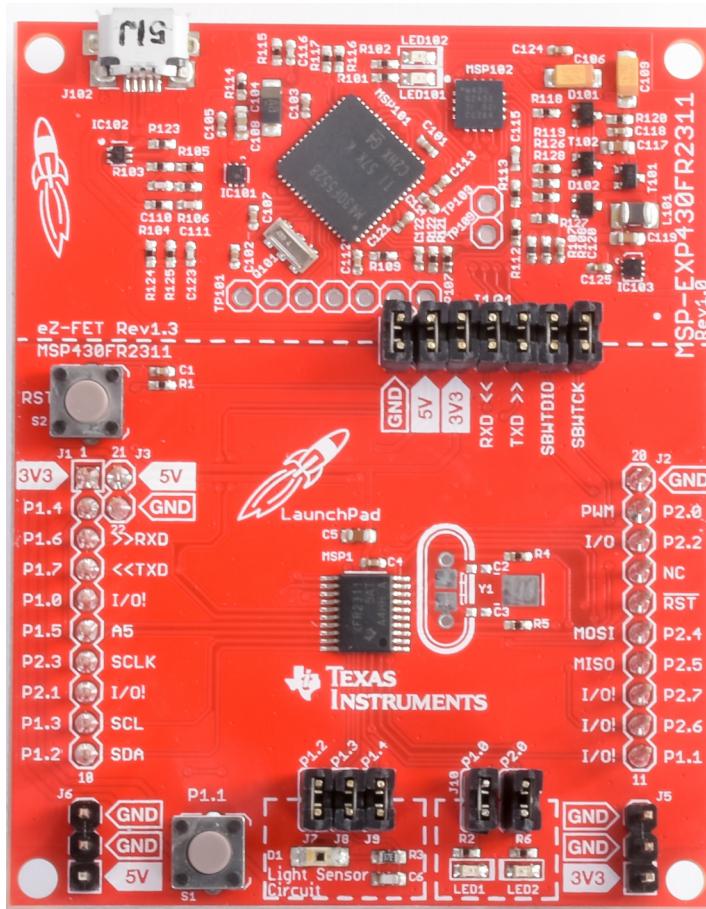


**MSP430FR2311 LaunchPad™ Development Kit  
(MSP-EXP430FR2311)**

Figure 1 shows the MSP430FR2311 LaunchPad™ development kit.



**Figure 1. MSP430FR2311 LaunchPad Development Kit**

**Contents**

1	Getting Started .....	3
2	Hardware.....	5
3	Software Examples .....	14
4	Resources.....	17
5	FAQ .....	22
6	Schematics .....	23

**List of Figures**

1	MSP430FR2311 LaunchPad Development Kit .....	1
2	MSP-EXP430FR2311 Overview .....	5
3	MSP-EXP430FR2311 Block Diagram.....	5
4	MSP430FR2311 MCU PW20 Pinout.....	6
5	eZ-FET Emulator .....	7
6	eZ-FET Isolation Jumper Block Diagram .....	9
7	Application Backchannel UART in Device Manager .....	9
8	MSP-EXP430FR2311 Power Block Diagram.....	10
9	LaunchPad Development Kit To Boosterpack Plug-in Module Connector Pinout .....	13
10	TI Resource Explorer Cloud .....	17
11	CCS Cloud .....	18
12	Directing the Project>Import Function to the Demo Project .....	19
13	When CCS Has Found the Project .....	19
14	Using TI Resource Explorer to Browse MSPWare .....	21
15	Schematics (1 of 6) .....	23
16	Schematics (2 of 6) .....	24
17	Schematics (3 of 6) .....	25
18	Schematics (4 of 6) .....	26
19	Schematics (5 of 6) .....	27
20	Schematics (6 of 6) .....	28

**List of Tables**

1	EnergyTrace Software .....	7
2	Isolation Block Connections .....	8
3	Hardware Change Log .....	13
4	Software Examples .....	14
5	IDE Minimum Requirements for MSP-EXP430FR2311 .....	14
6	Source File and Folders.....	14
7	Source File and Folders.....	15
8	Source File and Folders .....	16
9	How MSP Device Documentation is Organized.....	21

**Trademarks**

LaunchPad, BoosterPack, Code Composer Studio, EnergyTrace, MSP430, MSPWare, ControlSUITE, TivaWare, 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 [MSP430FR2311 LaunchPad™ Development Kit \(MSP-EXP430FR2311\)](#) is an easy-to-use evaluation module (EVM) for the MSP430FR2311 MCU. It contains everything needed to start developing on the [ultra-low-power MSP430FRx FRAM microcontroller platform](#), including onboard emulation for programming, debugging, and energy measurements. The board features onboard buttons and LEDs for quick integration of a simple user interface as well as an analog photodiode to showcase the integrated analog peripherals. The MSP430FR2311 MCU device is the world's first MCU with a configurable low-leakage current sense amplifier, with 50-pA current leakage and 4KB of embedded FRAM (ferroelectric random access memory), a nonvolatile memory known for its ultra-low power, high endurance, and high-speed write access.

Rapid prototyping is simplified by the 20-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.

The out-of-box experience provided with the [MSP-EXP430FR2311](#) MCU LaunchPad development kit features the onboard analog photodiode. The current from the photodiode is converted by an integrated operational amplifier to a voltage. That voltage is then sensed by an analog-to-digital converter and fed to a timer module to pulse width modulate LED2. As more light hits the photodiode, LED2 brightens, and as less light hits the photodiode, LED2 dims.

Free software development tools are also available, such as [TI's Eclipse-based Code Composer Studio™](#) integrated development environment (IDE) and [IAR Embedded Workbench®](#) IDE. Both of these IDEs support [EnergyTrace™ technology](#) for real-time power profiling and debugging when paired with the MSP430FR2311 MCU LaunchPad development kit. More information about the LaunchPad development kit, the supported BoosterPack plug-in modules, and available resources can be found on [TI's LaunchPad ecosystem portal](#).

The MSP-EXP430FR2311 is based on Experimental Silicon. The MSP430FR2311IPW20 device used in this version of the MSP-EXP430FR2311 is a prototype device and has not met or completed Texas Instruments internal reliability qualification requirements. The device used in this development kit is for prototyping purposes only and is not intended to be used in production systems. Therefore, the device is used on an "as is" basis and Texas Instruments makes no warranty of fitness for a specific purpose.

### 1.2 Key Features

- MSP430™ ultra-low-power FRAM technology based the MSP430FR2311 16-bit MCU
- EnergyTrace software available for ultra-low-power debugging
- 20 pin LaunchPad development kit standard leveraging the BoosterPack ecosystem
- Onboard eZ-FET emulation
- Two buttons and two LEDs for user interaction

### 1.3 What's Included

#### 1.3.1 Kit Contents

- 1x [MSP-EXP430FR2311](#) MCU LaunchPad development kit
- 1x micro USB cable
- 1x quick start guide

#### 1.3.2 Software Examples

- Out-of-box software
- Blink LED
- Software I<sup>2</sup>C

## 1.4 First Steps: Out-of-Box Experience

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

### 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 installing the TI Code Composer Studio IDE or IAR Embedded Workbench for MSP430 IDE. Drivers are also available at [ti.com/MSPdrivers](http://ti.com/MSPdrivers).

### 1.4.2 Running the Out-of-Box Demo

When connected to your computer, the LaunchPad development kit powers up and LED2 dims based on the amount of ambient light hitting the photo diode, D1. Use a light source such as a flashlight or desk lamp to illuminate D1 and the brightness of LED2 should increase depending on the intensity of the light source.

## 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. Refer to [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](#), [dev.ti.com](http://dev.ti.com). The cloud-based Resource Explorer provides access to all of the examples and resources in MSPWare™ software. Code Composer Studio IDE 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 for download from [www.ti.com/tool/msp-exp430fr2311](http://www.ti.com/tool/msp-exp430fr2311). Code is licensed under BSD, and TI encourages reuse and modifications to fit specific needs.

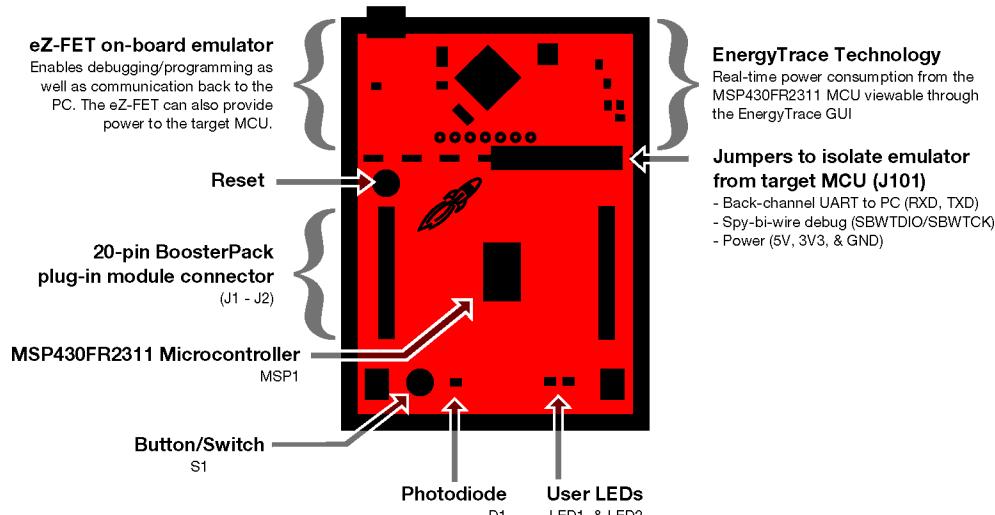
[Section 3](#) describes all functions in detail and provides a project structure to help familiarize you with the code.

Using the onboard eZ-FET emulator, 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-EXP430FR2311 hardware.

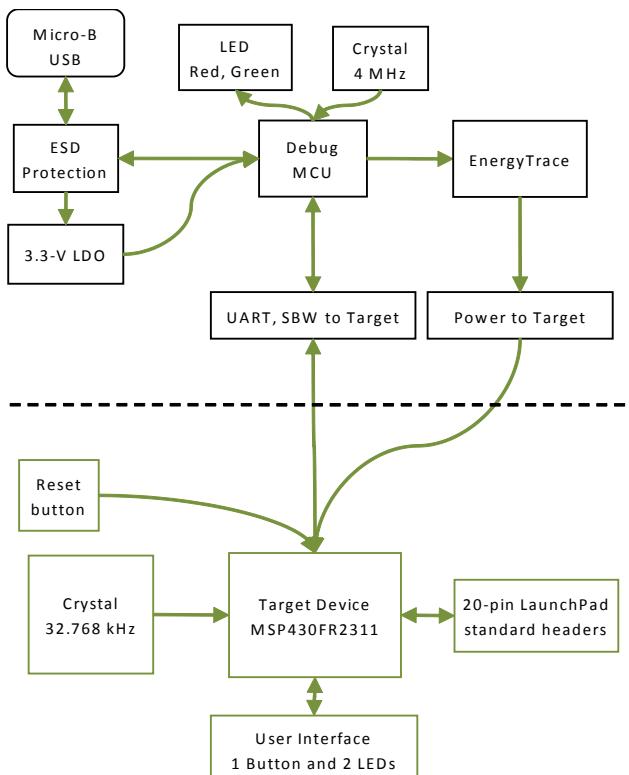
# MSP-EXP430FR2311 Overview



**Figure 2. MSP-EXP430FR2311 Overview**

### 2.1 Block Diagram

Figure 3 shows the block diagram.



**Figure 3. MSP-EXP430FR2311 Block Diagram**

## 2.2 Hardware Features

### 2.2.1 MSP430FR2311 MCU

The MSP430FR2311 MCU is the next device in TI's new ultra-low-power FRAM technology platform. FRAM is a cutting edge memory technology, combining the best features of flash and RAM into one nonvolatile memory.

Device features include:

- 1.8-V to 3.6-V operation
- 16-bit RISC architecture with up to 16-MHz system clock and 8-MHz FRAM access
- Up to 3.75KB of nonvolatile FRAM
- 126- $\mu$ A/MHz active mode and 1- $\mu$ A standby with RTC
- 8-channel 10-bit ADC
- Operational amplifier that supports multiple modes
- Enhanced comparator with integrated 6-bit DAC as reference voltage
- Transimpedance amplifier
- Three timers
- Direct memory access
- 16 GPIOs

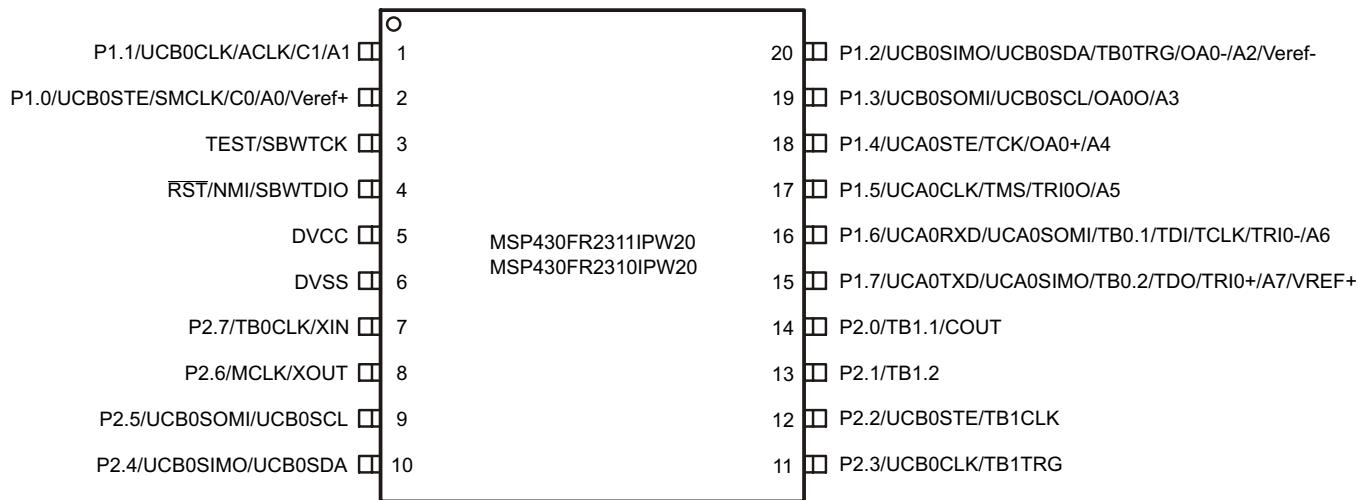
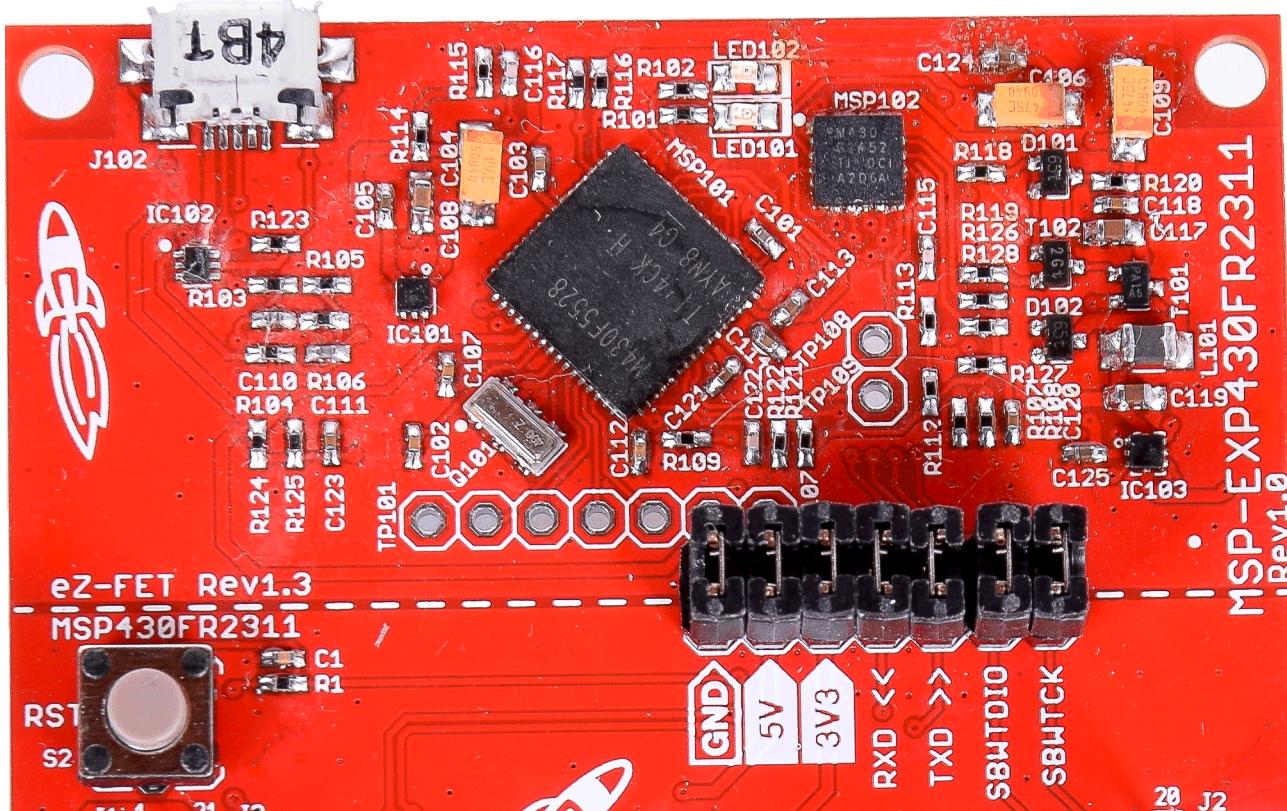


Figure 4. MSP430FR2311 MCU PW20 Pinout

## 2.2.2 eZ-FET Onboard Emulator With EnergyTrace™ Software

To keep development easy and cost effective, TI's LaunchPad development kits integrate an onboard emulator, which eliminates the need for expensive programmers. The [MSP-EXP430FR2311](#) has the eZ-FET emulator (see [Figure 5](#)), which is a simple and low-cost debugger that supports all MSP430 MCU derivatives.



**Figure 5. eZ-FET Emulator**

The [MSP-EXP430FR2311](#) MCU LaunchPad development kit features EnergyTrace software. The EnergyTrace software functionality varies across the MSP portfolio.

**Table 1. EnergyTrace Software**

Features	EnergyTrace Software	EnergyTrace++ Software
Current monitoring	✓	✓
CPU State		✓
Peripheral and system state		✓
Devices supported	All MSP430 MCUs	MSP430FR59xx and FR69xx MCUs
Development tool required	MSP-FET or eZ-FET	MSP-FET or eZ-FET

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. The provided UART supports hardware flow control (RTS and CTS), although by default these signals are not connected to the target.

The dotted line shown in [Figure 5](#) divides the eZ-FET emulator from the target area. The signals that cross this line can be disconnected by jumpers on J101, the isolation jumper block. More details on the isolation jumper block are in [Section 2.2.3](#).

The eZ-FET hardware can be found in the schematics in [Section 6](#) and in the accompanying [hardware design files](#). The software and more information about the debugger can be found on the [eZ-FET wiki](#).

### 2.2.3 Emulator 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 MSP430FR2311 MCU target domain. This includes eZ-FET Spy-Bi-Wire signals, application UART signals, and 3.3-V and 5-V power.

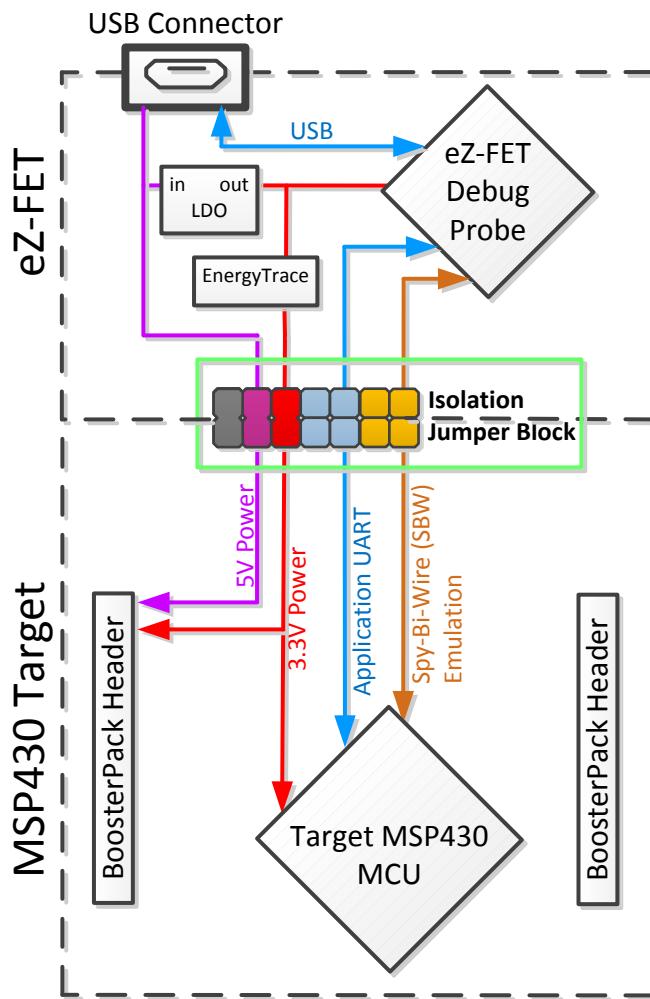
Reasons to open these connections:

- To remove any and all influence from the eZ-FET emulator 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 microcontroller 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 microcontroller

**Table 2. Isolation Block Connections<sup>(1)</sup>**

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 FR2311 receives data through this signal. The arrows indicate the direction of the signal.
TXD >>	Backchannel UART: The target FR2311 sends data through this signal. The arrows indicate the direction of the signal.
SBW RST	Spy-Bi-Wire emulation: SBWTDO data signal. This pin also functions as the RST signal (active low).
SBW TST	Spy-Bi-Wire emulation: SBWTCK clock signal. This pin also functions as the TST signal.

<sup>(1)</sup> Also see [Figure 6](#).



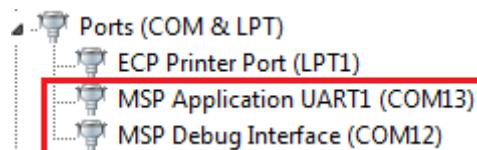
**Figure 6. eZ-FET Isolation Jumper Block Diagram**

#### 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 development kit.

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 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**

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 MSP430FR2311 MCU 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 Special Features

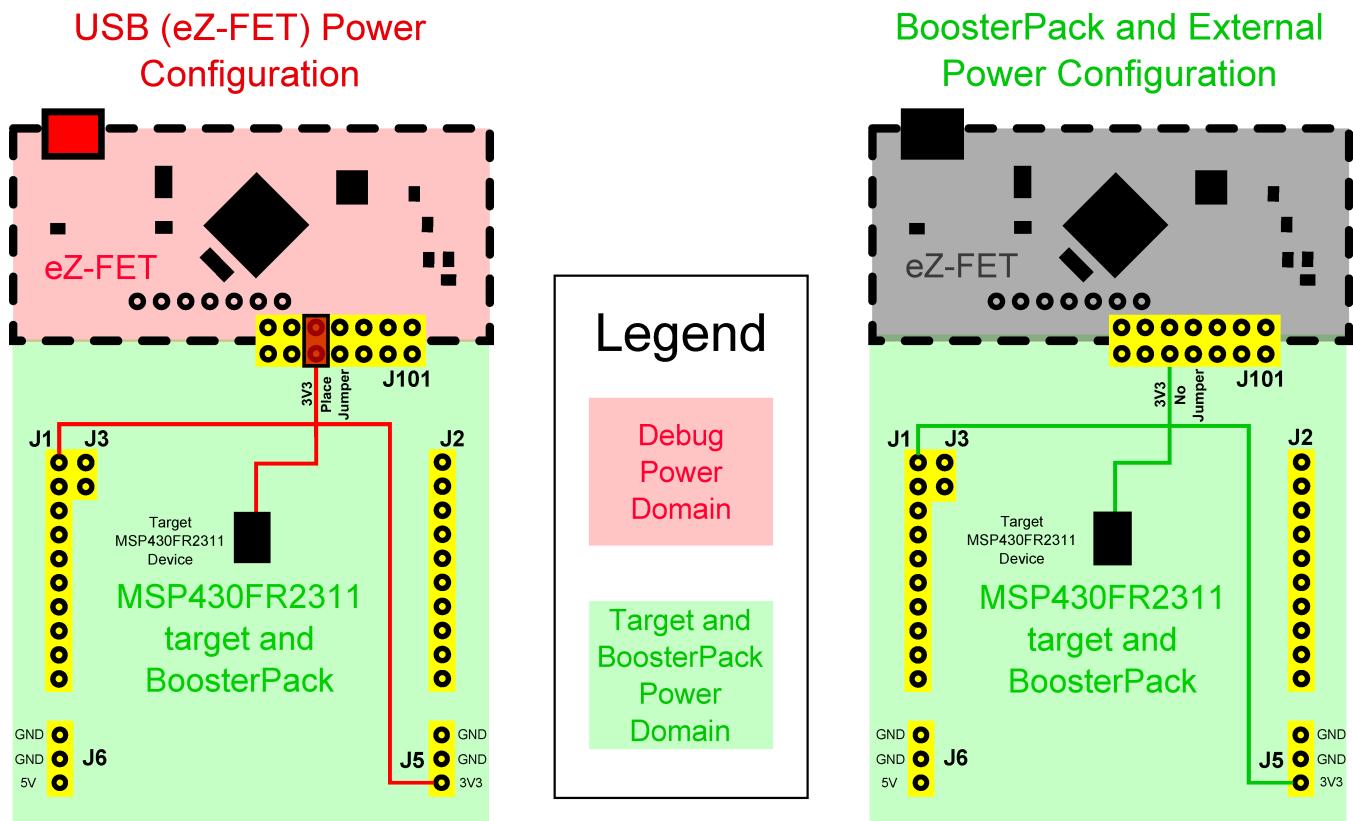
### 2.2.5.1 Smart Analog Combo (SAC)

The MSP430FR2311 MCU LaunchPad development kit features pins to access the smart analog combo on the device. These pins are accessed on jumpers J7, J8 and J9 where the smart analog combo operational amplifier is connected to the onboard photodiode circuit. These pins are also connected to the BoosterPack plug-in module header pins. For applications that use the SAC and a connected BoosterPack plug-in module, be sure to check for any pin conflicts. The photodiode circuit can be disconnected from the SAC using jumpers J7, J8 and J9. The user can connect an external analog sensor circuit to the SAC by removing jumpers J7, J8 and J9 and connecting their circuit to the appropriate header pins through the BoosterPack plug-in module headers or the jumpers near the light sensor circuit.

To adjust the light sensor circuits' sensitivity the user can adjust either the software or the hardware. By changing the value of R3 and C6 the user can change the operational amplifiers low-pass filter and sensitivity to the photodiodes current. The user can also adjust the calculations performed in the software to adjust the sensitivity of the circuit to their desired level.

## 2.3 Power

The board was designed to accommodate various powering methods, including through the onboard eZ-FET as well as external or BoosterPack power (see [Figure 8](#)).



**Figure 8. MSP-EXP430FR2311 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.

### 2.3.2 BoosterPack Plug-In-Module 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 MSP430FR2311 MCU has an operating range of 1.8 V to 3.6 V. More information can be found in [MSP430FR231x Mixed-Signal Microcontrollers](#).

## 2.4 Measure MSP430 MCU Current Draw

To measure the current draw of the MSP430FR2311 MCU using a multi-meter, use the 3V3 jumper on the 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 isolation block, and attach an ammeter across this jumper.
2. Consider the effect that the backchannel UART and any circuitry attached to the MSP430FR2311 MCU 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). 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 software allows you to compare various current profiles and better optimize your energy performance!

## 2.5 Clocking

The [MSP-EXP430FR2311](#) provides an input for an external clock in addition to the internal clocks in the device. No crystal is populated by default to allow all pins to connect to the BoosterPack header. The user can choose to populate any 32.768 kHz crystal that will fit onto the provided footprint. One option is recommended below:

- Y1: 32-kHz Epson FC-135R 12.5pF crystal, Part #: X1A0001410014

The 32-kHz crystal allows for lower LPM3 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. If using the external Y1 crystal footprint the user will need to remove the 0-ohm resistors R4 and R5 and populate C2 and C3 with the appropriate tuning capacitors for the crystal that is 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 Emulator With a Different Target

The eZ-FET emulator on the LaunchPad development kit can interface to most MSP430 MCU derivative devices, not just the onboard MSP430FR2311 MCU target device.

To do this, disconnect every jumper in the isolation jumper block. This is necessary, because the emulator 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. Note that to be compatible with SBW, the capacitor on RST/SBWTDO cannot be greater than 2.2 nF. The documentation for designing MSP430 MCU JTAG interface circuitry is the [MSP430 Hardware Tools User's Guide](#).

Finally, wire together these signals from the emulator 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 Plug-in Module Pinout

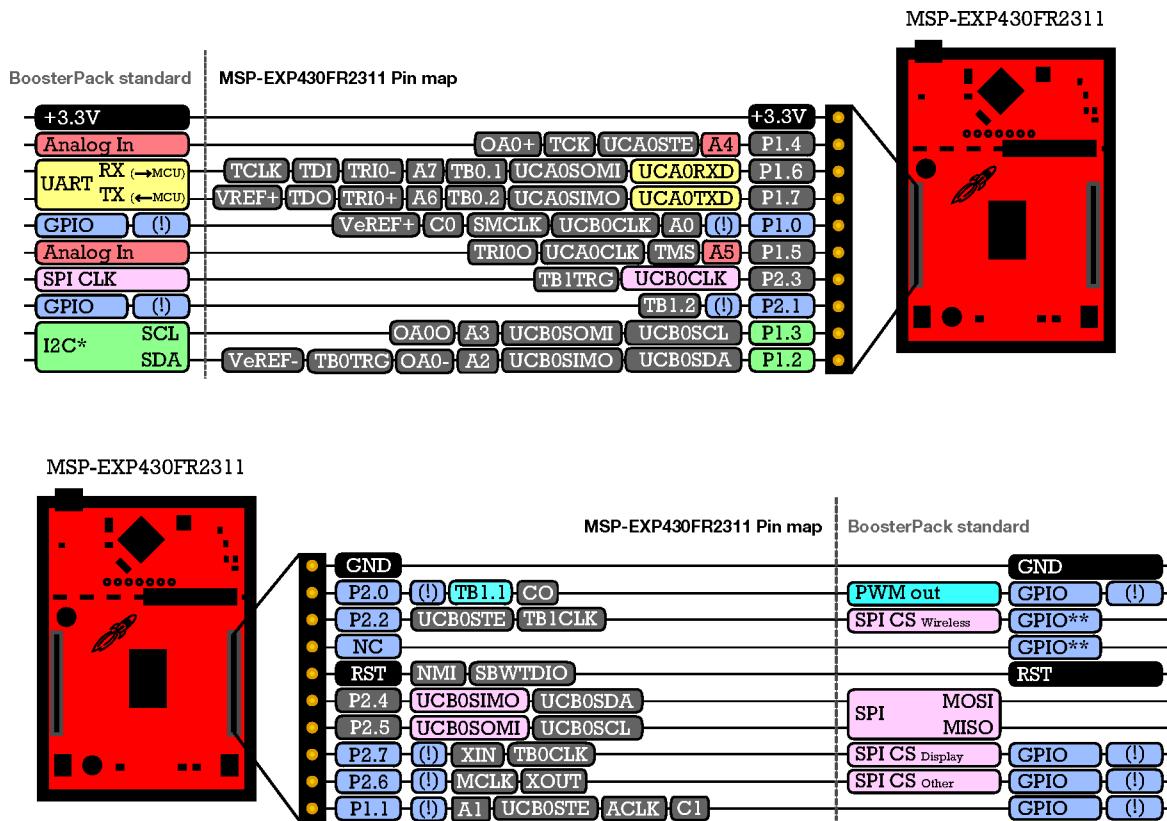
The LaunchPad development kit adheres to the 20-pin LaunchPad pinout standard. A standard was created to aid compatibility between the LaunchPad development kit and BoosterPack plug-in module tools across the TI ecosystem.

The 40-pin standard is compatible with the 20-pin standard that is used by other LaunchPad development kit's like the [MSP-EXP430FR6989](#). This allows some subset of functionality of 40-pin BoosterPack modules to be used with 20-pin LaunchPad development kits.

While most BoosterPack plug-in modules are compliant with the standard, some are not. The [MSP-EXP430FR2311](#) MCU LaunchPad development kit is compatible with all 20-pin BoosterPacks 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 [MSP-EXP430FR2311](#) MCU LaunchPad development kit, compare the schematic of the candidate BoosterPack with the LaunchPad to ensure compatibility. Keep in mind that sometimes conflicts can be resolved by changing the MSP430FR2311 MCU pin function configuration in software. More information about compatibility can also be found at [www.ti.com/launchpad](http://www.ti.com/launchpad).

[Figure 9](#) shows the 20-pin pinout of the MSP430FR2311 MCU LaunchPad development kit.

Note that software configuration of the pin functions plays a role in compatibility. The LaunchPad development kit side of the dashed line in [Figure 9](#) shows all of the functions for which the MSP430FR2311 MCU's pins can be configured. This can also be seen in the MSP430FR2311 MCU data sheet. The BoosterPack plug-in module side of the dashed line shows the standard. The LaunchPad development kit function whose color matches the BoosterPack plug-in module function shows the specific software-configurable function by which the MSP430FR2311 LaunchPad development kit adheres to the standard.



**Figure 9. LaunchPad Development Kit To Boosterpack Plug-in 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 in the [MSP-EXP430FR2311 Hardware Design Files](#).

## **2.8.2 Software**

All design files including TI-TXT object-code firmware images, software example projects, and documentation are available in the [MSP-EXP430FR2311 Software Examples](#).

## **2.9    *Hardware Change log***

**Table 3. Hardware Change Log**

PCB Revision	Description
1.0	Initial release
1.1	Updated rear silkscreen to current LaunchPad standards. Added CE marking to silkscreen.

### 3 Software Examples

Three software examples are included with the MSP430FR2311 MCU LaunchPad development kit (see [Table 4](#)), which can be found in the zip source folder in the [MSP-EXP430FR2311 Software Examples](#) and are also available inside [MSPWare](#).

**Table 4. Software Examples**

Demo Name	BoosterPack Plug-In Module Required	Description	More Details
OutOfBox_MSP430FR2311	None	The out-of-box demo preprogrammed on the LaunchPad development kit from the factory. Demonstrates features of MSP430FR2311 MCU device.	<a href="#">Section 3.1</a>
BlinkLED_MSP430FR2311	None	Blinks an LED on the LaunchPad development kit at a fixed interval.	<a href="#">Section 3.2</a>
BOOSTXL- EDUMKII_SoftwareI2C_- MSP430FR2311	BOOSTXL- EDUMKII	Reads digital ambient light sensor utilizing software I <sup>2</sup> C master example.	<a href="#">Section 3.3</a>

To use any of the software examples with the LaunchPad development kit, you must have an integrated development environment (IDE) that supports the MSP430FR2311 microcontroller (see [Table 5](#)).

**Table 5. IDE Minimum Requirements for MSP-EXP430FR2311**

Code Composer Studio™ IDE	IAR Embedded Workbench® IDE
Version 6.1 or later	Version 6.40.2 or later

For more details on how to get started quickly, and where to download the latest Code Composer Studio and IAR Embedded Workbench IDEs, see [Section 4.1](#).

#### 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.

This demo shows how to use the op-amp module, combined with the timer, ADC, and LED, to implement a simple light sensor circuit for ultra-low-power applications.

Upon powering up the out-of-box demo, LED1 will toggle at startup and then LED2 will dim based on the amount of ambient light in the area. If the user covers the photo diode, D1, with a piece of paper or their hand the brightness of LED2 should dim based on the amount of light. If the user illuminates the photo diode with a light source LED2 should increase in brightness until the ADC saturates and the LED reaches full brightness. The current ADC reading is saved to FRAM memory before the device re-enters LPM3.

##### 3.1.1 Source File Structure

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

**Table 6. Source File and Folders**

Name	Description
main.c	The out-of-box demo main function, initializations, shared ISRs, and so on.
Library: Driverlib	Device driver library

### 3.1.2 Power Measurement

The MSP430FR2311 MCU is ultra-low-power and its out-of-box software provides an example of an ultra-low-power application. To measure the applications power we will use EnergyTrace software integrated in Code Composer Studio IDE v6.1 or later.

After importing the out-of-box project into Code Composer Studio IDE and connecting the LaunchPad development kit, follow the instructions below. For more information on how to import a project, see [Section 4.1.2](#).



1. Click Debug to program the LaunchPad development kit and start the Debug session.
2. Click Proceed on the ULP Advisor to continue to the Debug session.
3. Select View → Other → EnergyTrace → EnergyTrace software from the menu bar at the top and then click OK.  

4. Click the Set measurement duration icon and select "10 sec" from the drop-down menu.
5. Remove the following jumpers from the LaunchPad development kit:  
J11 at LED2  
RXD at J101  
TXD at J101
6. Select Run → Free Run (Ctrl+F8) from the menu bar at the top.
7. When finished, review the power profile of the application and note the battery life of more than 400 days when using a CR2032.

### 3.2 Blink LED Example

This very simple 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	<a href="#">Device driver library</a>

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

### 3.3 Software I<sup>2</sup>C Example

This simple software example shows how to implement software controlled I<sup>2</sup>C master communication on the LaunchPad development kit.

#### 3.3.1 Source File Structure

The project is split into multiple files (see [Table 8](#)). This makes it easier to navigate and reuse parts of it for other projects.

**Table 8. Source File and Folders**

Name	Description
main.c	The out-of-box demo main function, initializations, shared ISR's, etc
MSP430_SWI2C_Master.c	Main function file for software I <sup>2</sup> C master communication
opt3001.c	Ambient light sensor library
Library: Driverlib	<a href="#">Device driver library</a>

## 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, IAR Embedded Workbench IDE, or Energia.

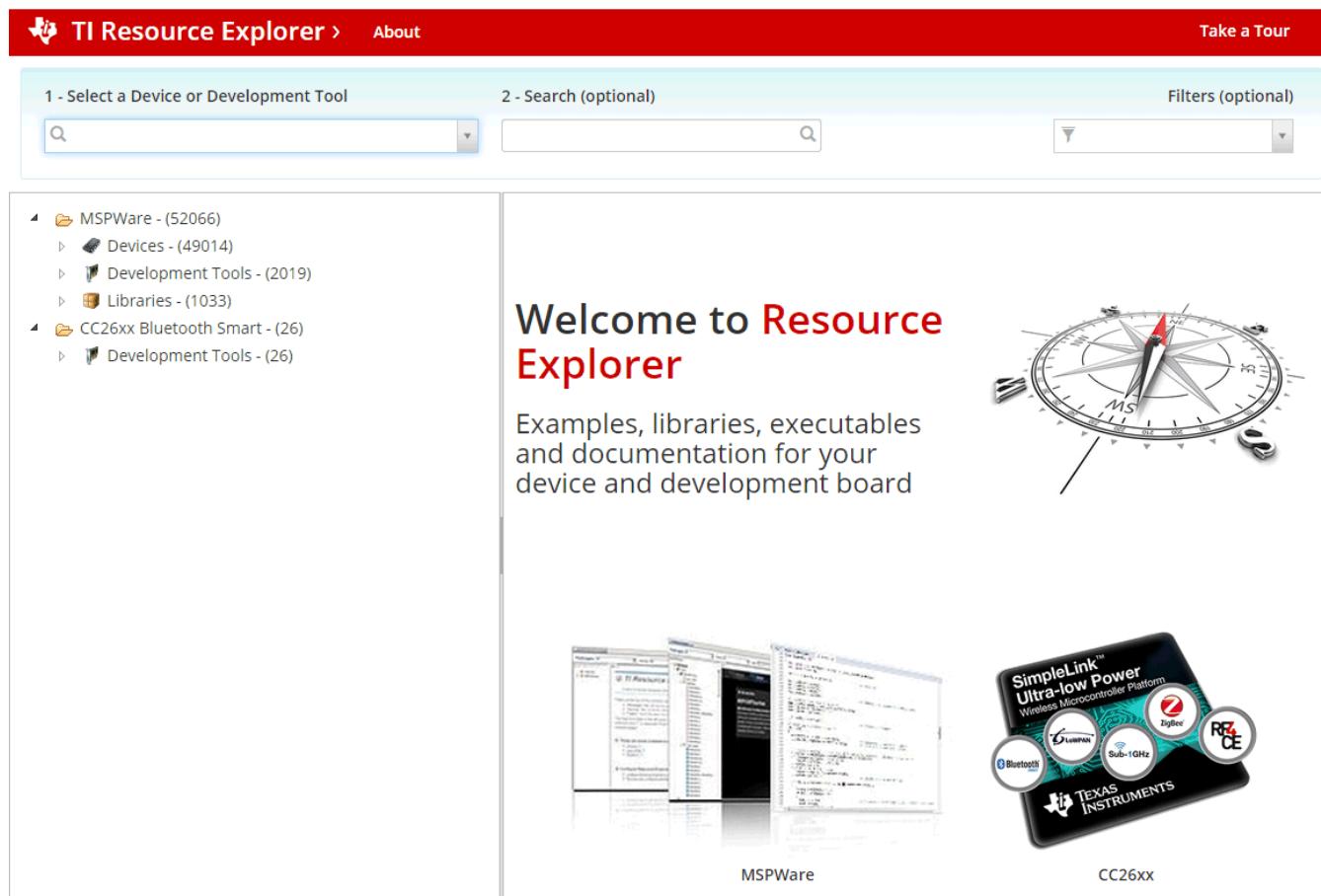
#### 4.1.1 TI Cloud Development Tools

TI's Cloud-based software development tools provide instant access to MSPWare software 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 software without having to download files to your local drive (see [Figure 10](#)).

Go check out TI Resource Explorer Cloud now at <https://dev.ti.com/>.



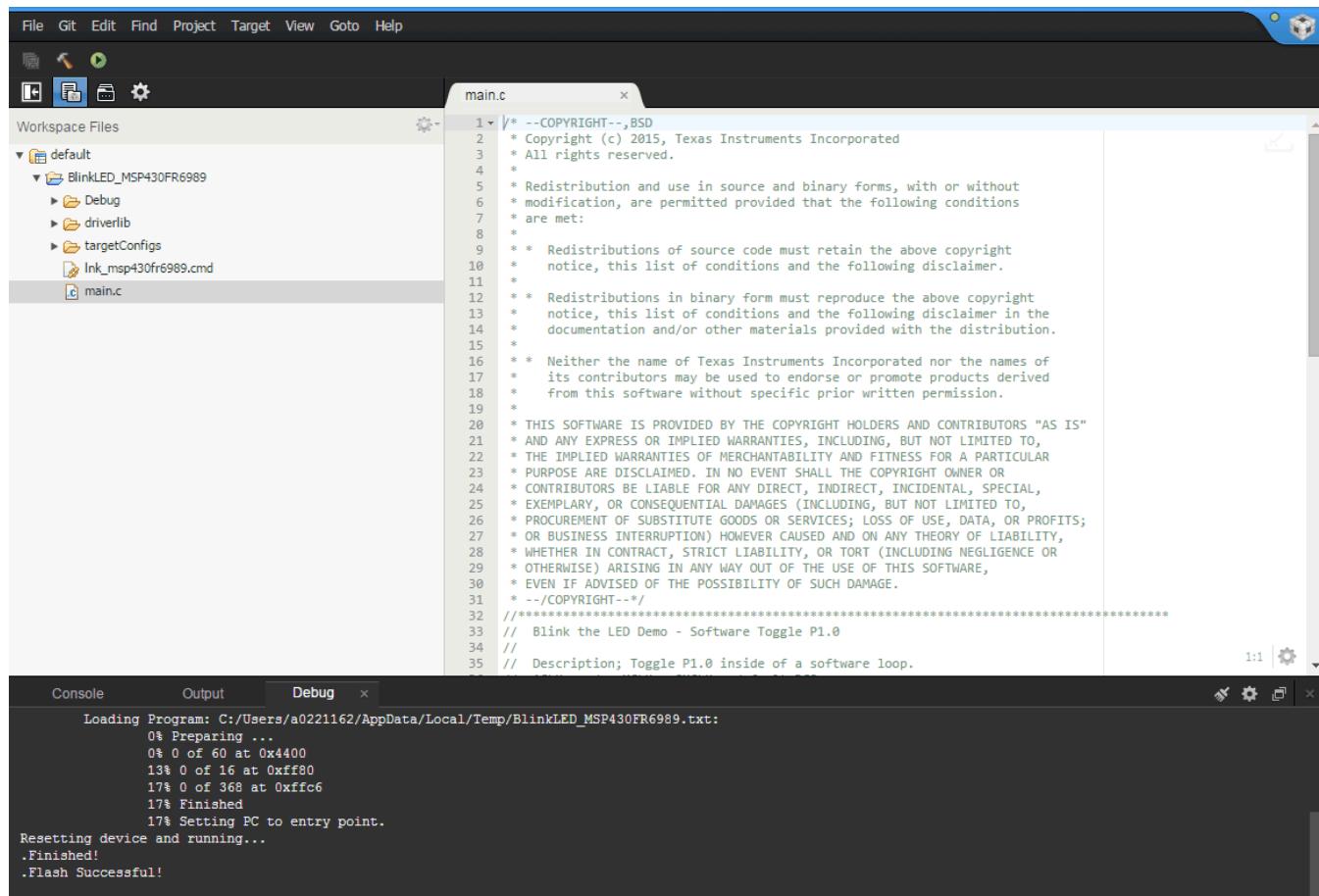
**Figure 10. TI Resource Explorer Cloud**

#### 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 the LaunchPad development kit (see [Figure 11](#)). No need to download and install large software packages, simply connect the LaunchPad development kit and begin. 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.

A full comparison between CCS Cloud and CCS Desktop is available at <https://dev.ti.com/about#cloudordesktop>.

Go check out Code Composer Studio Cloud now at <https://dev.ti.com/>.



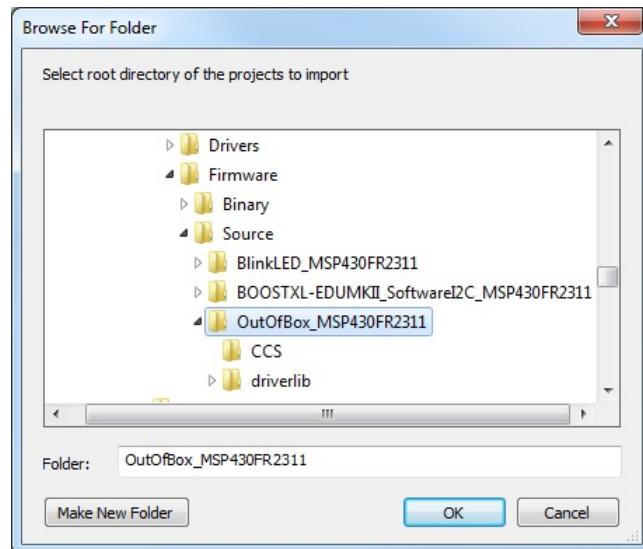
**Figure 11. CCS Cloud**

#### 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 IDE 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 IDE and download it at [www.ti.com/tool/ccstudio](http://www.ti.com/tool/ccstudio).

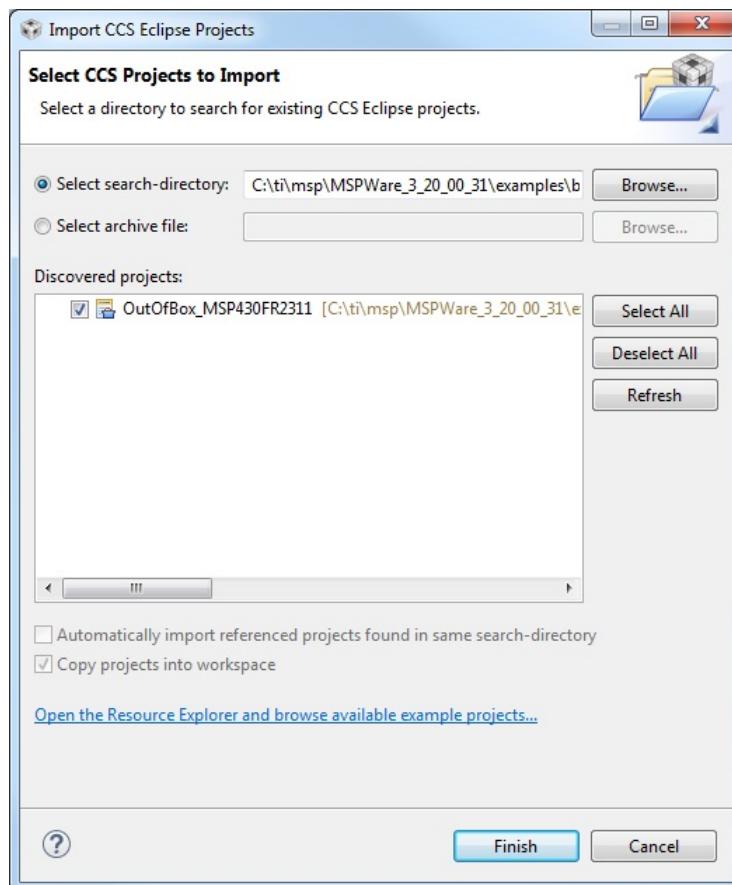
CCS v6.1 or higher is required. When CCS has been launched, and a workspace directory chosen, use Project>Import Existing CCS Eclipse Project. Direct it to the desired demo's project directory that contains main.c (see [Figure 12](#)).



**Figure 12. 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, Code Composer Studio should recognize the project and allow you to import it. The indication that Code Composer Studio has found it is that the project appears in the box shown in [Figure 13](#), and it has a checkmark to the left of it.



**Figure 13. When CCS Has Found the Project**

Sometimes Code Composer Studio IDE 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 Code Composer Studio IDE workspace, be sure to check the workspace directory on the file system.)

#### 4.1.3 IAR Embedded Workbench for Texas Instruments MSP430

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](http://supp.iar.com/Download/SW/?item=EW430-EVAL).

IAR 6.10 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 software, which contains MSP430 MCU libraries and the TI Resource Explorer. These are already included in a Code Composer Studio IDE installation (unless the user selected otherwise).

#### 4.1.4 Energia

Energia is a simple, open-source, and community-driven code editor that is based on the [Wiring](#) and [Arduino](#) framework. Energia provides unmatched ease of use through very high-level APIs that can be used across hardware platforms. Energia is a light-weight IDE that does not have the full feature set of Code Composer Studio IDE or IAR. However, Energia is great for anyone who wants to get started very quickly or who does not have significant coding experience.

Learn more about Energia and download it at [www.energia.nu](http://www.energia.nu).

### 4.2 LaunchPad Websites

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

- [MSP-EXP430FR2311 Tool Folder](#): Resources specific to this particular LaunchPad development kit
- [TI's LaunchPad development kit portal](#): Information about all LaunchPad development kits from TI

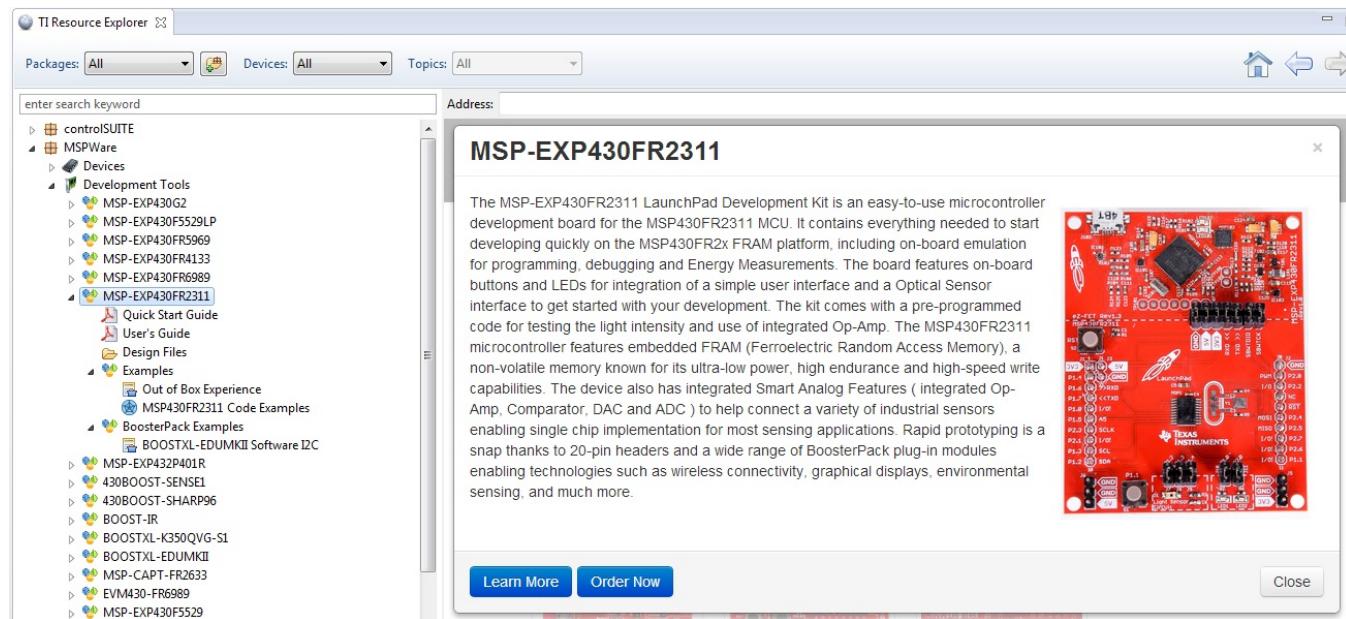
### 4.3 MSPWare Software and TI Resource Explorer

TI Resource Explorer is a tool integrated into Code Composer Studio IDE that allows you to browse through available design resources (see [Figure 14](#)). TI Resource Explorer will help you quickly find what you need inside packages including MSPWare software, ControlSUITE™ software, TivaWare™ software, 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 4.1.1.2](#) for more information.

MSPWare software 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 software also includes a high level API called MSP Driver Library. This library makes it easy to talk to MSP hardware. More information can be found at [www.ti.com/tool/mspware](http://www.ti.com/tool/mspware).



**Figure 14. Using TI Resource Explorer to Browse MSPWare**

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

#### 4.4 FRAM Utilities

The [TI MSP430 FRAM MCU 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.

#### 4.5 MSP430FR2311MCU

##### 4.5.1 Device Documentation

At some point, you will probably want more information about the MSP430FR2311 microcontrollers. [Table 9](#) summarizes how the documentation is organized for every MSP device.

**Table 9. How MSP Device Documentation is Organized**

Document	For MSP430FR2311 MCU	Description
Device family user's guide	<a href="#">MSP430FR4xx and MSP430FR2xx Family User's Guide</a>	Architectural information about the device, including all modules and peripherals such as clocks, timers, ADC, and so on.
Device-specific data sheet	<a href="#">MSP430FR231x Mixed-Signal Microcontrollers data sheet</a>	Device-specific information and all parametric information for this device.

##### 4.5.2 MSP430FR2311 MCU Code Examples

[MSP430FR231x Code Examples](#) is a set of very simple C examples that demonstrate how to use the entire set of peripherals on the MSP430 MCU, including serial communication, ADC10, SAC, TRI, Timer\_B, 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 MCU Application Notes and TI Designs

There are many application notes that can be found at [www.ti.com/msp430](http://www.ti.com/msp430), as well as [TI Designs](#) with practical design examples and topics.

### 4.6 Community Resources

#### 4.6.1 TI E2E™ Online Community

Search the forums at [e2e.ti.com](http://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](http://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:

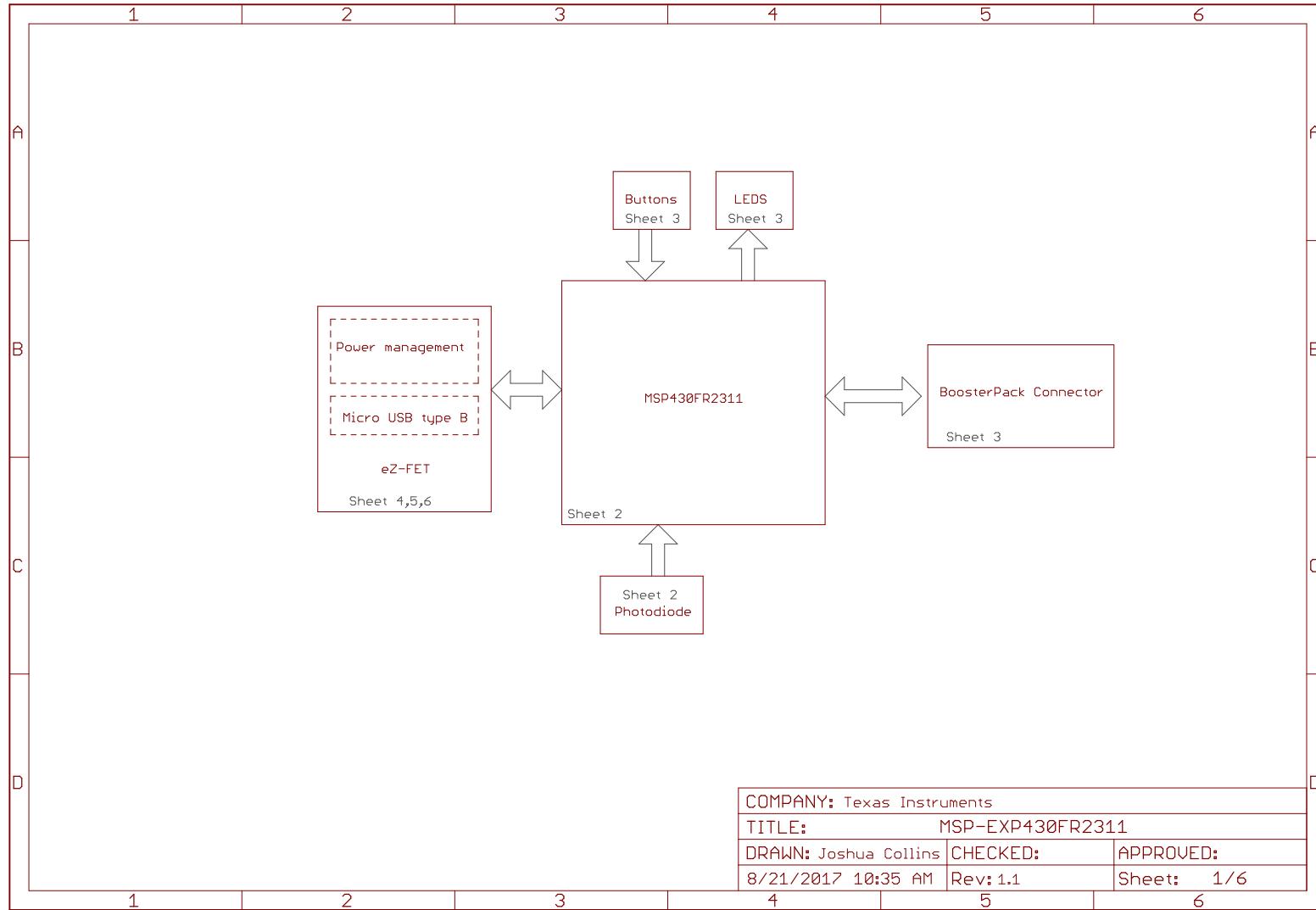
- Do the baud rate in the host's terminal application and the eUSCI settings match?
- Are the appropriate jumpers in place on the isolation jumper block?
- Probe on RXD and send data from the host. If you don't see data, it might be a problem on the host side.
- Probe on TXD while sending data from the 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 development kit had a socket, allowing me change the target device. Why doesn't this LaunchPad development kit use one?

A: This LaunchPad development kit provides more functionality, and this means using a device with more pins. Sockets for devices with this many pins are too expensive for the target price of the LaunchPad development kit.

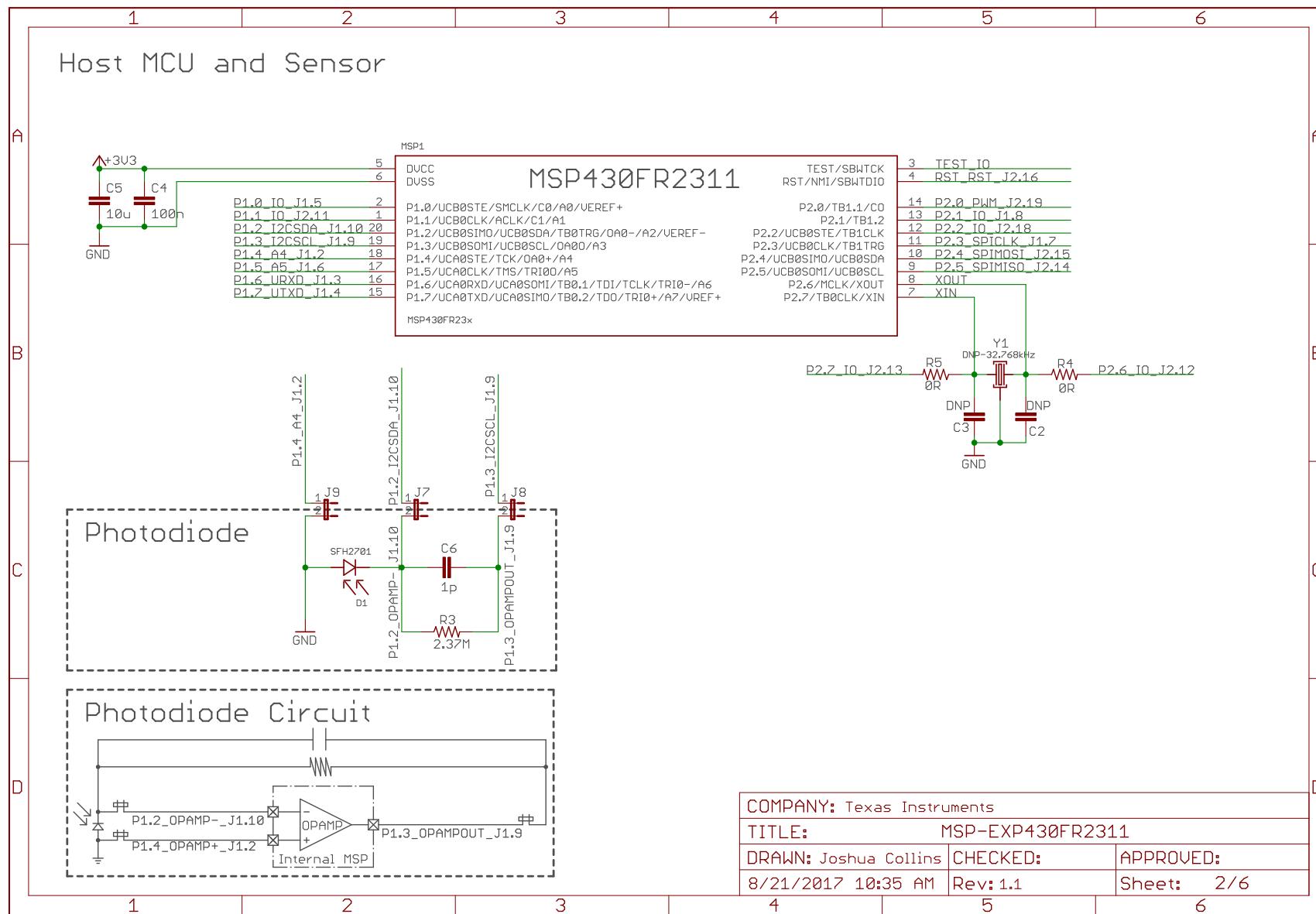
## 6 Schematics

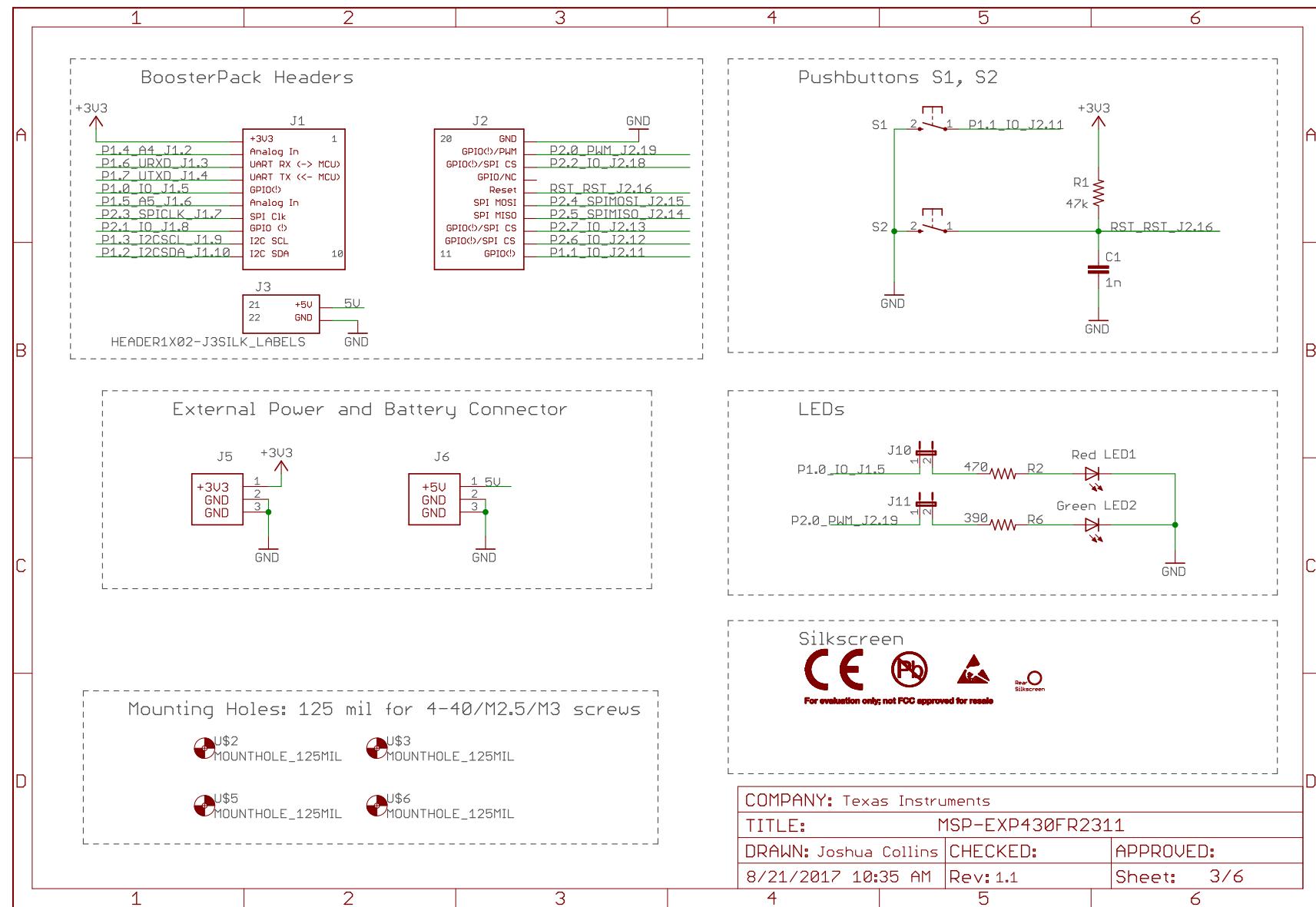
Figure 15 to Figure 20 show the schematics for the MSP-EXP430FR2311, which are also available in the [hardware design files](#).



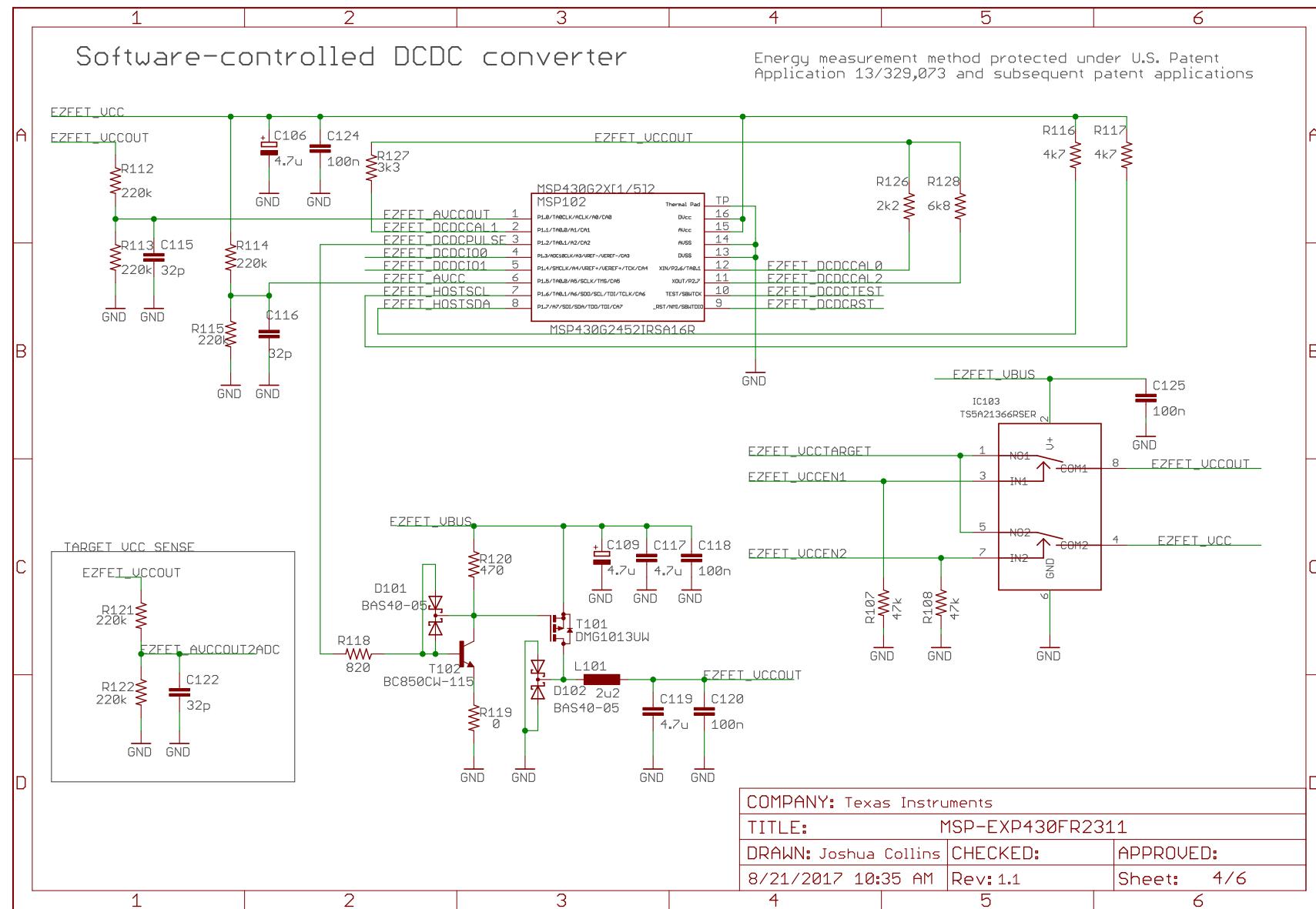
Copyright © 2017, Texas Instruments Incorporated

**Figure 15. Schematics (1 of 6)**

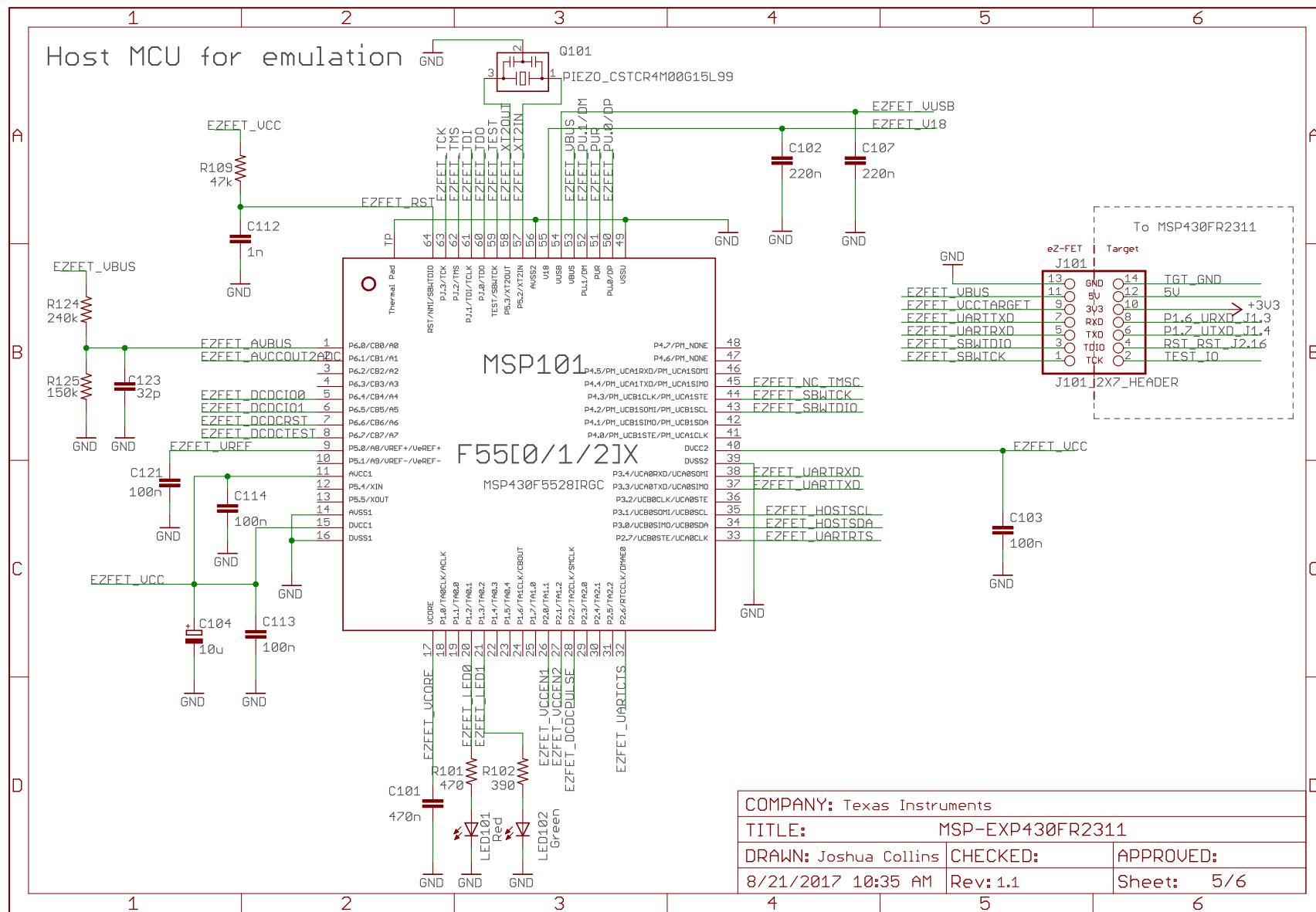




Copyright © 2017, Texas Instruments Incorporated



**Figure 18. Schematics (4 of 6)**


**Figure 19. Schematics (5 of 6)**

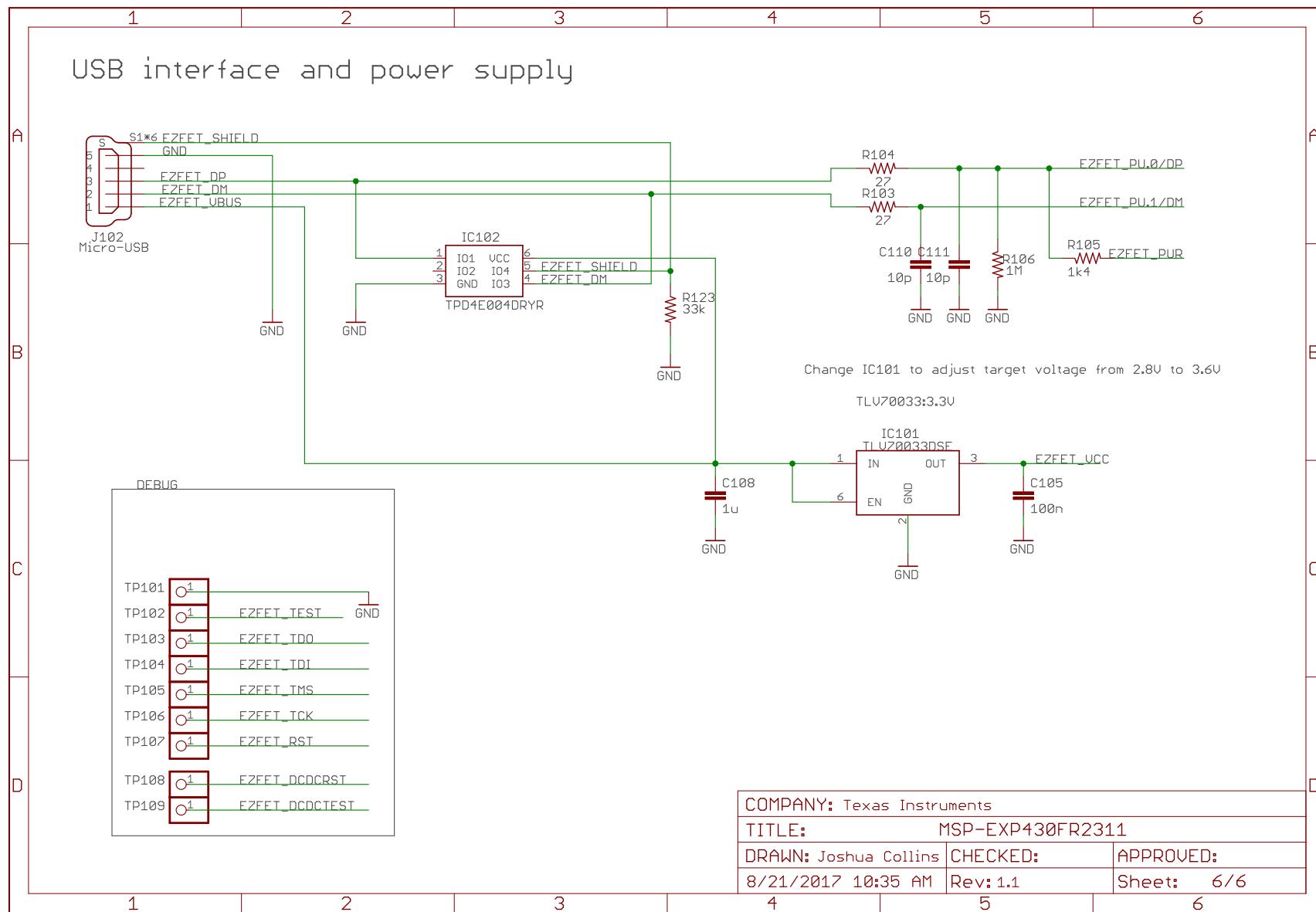


Figure 20. Schematics (6 of 6)

## Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from May 18, 2016 to August 30, 2017	Page
• Added rev 1.1 to <a href="#">Table 3, Hardware Change Log</a> .....	13
• Changed required IAR Embedded Workbench IDE version to v6.40.2 in <a href="#">Table 5, IDE Minimum Requirements for MSP-EXP430FR2311</a> .....	14
• Updated all schematics figures in <a href="#">Section 6, Schematics</a> .....	23

## **IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES**

Texas Instruments Incorporated ("TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products (<http://www.ti.com/sc/docs/stdterms.htm>), evaluation modules, and samples (<http://www.ti.com/sc/docs/samptersms.htm>).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2018, Texas Instruments Incorporated