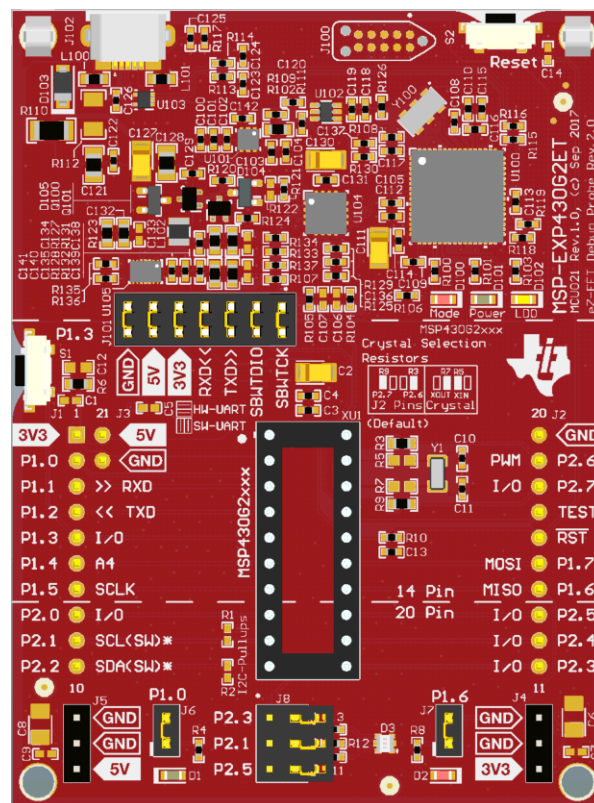


## MSP430G2553 LaunchPad™ Development Kit (MSP-EXP430G2ET)

The **MSP430G2553 LaunchPad™ Development Kit** is an inexpensive and easy-to-use evaluation module (EVM) for the MSP430G2xx entry-level series of microcontrollers (MCUs). It contains everything needed to start developing on the ultra-low-power MSP430™ microcontroller platform, including an onboard debug probe for programming, debugging and energy measurements. The board also features a push button and three LEDs for creating a simple user interface.

Figure 1 shows the MSP-EXP430G2ET LaunchPad development kit.



**Figure 1. MSP-EXP430G2ET LaunchPad Development Kit**

## Contents

1	Getting Started .....	4
	1.1 Introduction .....	4
	1.2 Key Features .....	4
	1.3 What's Included .....	4
	1.4 First Steps: Out-of-Box Experience .....	4
	1.5 Next Steps: Looking Into the Provided Code .....	5
2	Hardware.....	6
	2.1 Block Diagram.....	6
	2.2 Hardware Features .....	7
	2.3 Power .....	10
	2.4 Measure MSP430 Current Draw .....	11
	2.5 Clocking .....	11
	2.6 Using the eZ-FET Debug Probe With a Different Target .....	11
	2.7 BoosterPack Plug-in Module Pinout .....	12
	2.8 20-Pin DIP Socket.....	13
	2.9 Supported Devices .....	14
	2.10 Design Files.....	15
	2.11 Hardware Change log .....	15
3	Software Examples .....	15
	3.1 Out-of-Box Software Example .....	16
	3.2 Blink LED Example.....	17
4	Resources .....	17
	4.1 Integrated Development Environments.....	17
	4.2 LaunchPad Development Kit Websites.....	20
	4.3 MSPWare and TI Resource Explorer.....	21
	4.4 MSP430G2553 MCU .....	21
	4.5 Community Resources .....	22
5	FAQ .....	23
6	Schematics .....	24

## List of Figures

1	MSP-EXP430G2ET LaunchPad Development Kit .....	1
2	MSP-EXP430G2ET Overview .....	6
3	MSP-EXP430G2ET Block Diagram .....	6
4	MSP430G2553 20-Pin N Package (Top View) .....	7
5	eZ-FET Debug Probe .....	8
6	eZ-FET Isolation Jumper Block Diagram .....	9
7	Application Backchannel UART in Device Manager.....	10
8	MSP-EXP430G2ET Power Block Diagram .....	10
9	BoosterPack Checker Tool .....	13
10	Pinout of Connector From LaunchPad Development Kit to BoosterPack Plug-in Module .....	13
11	Insert Device Into Target Socket .....	14
12	TI Resource Explorer Cloud .....	18
13	CCS Cloud .....	18
14	Directing the Project>Import Function to the Demo Project .....	19
15	When CCS Has Found the Project .....	20
16	Using TI Resource Explorer to Browse MSP-EXP430G2ET in MSPWare .....	21
17	Schematics (1 of 3) .....	24
18	Schematics (2 of 3) .....	25
19	Schematics (3 of 3) .....	26

**List of Tables**

1	EnergyTrace Technology .....	8
2	Isolation Block Connections .....	9
3	Supported Devices .....	14
4	Hardware Change Log .....	15
5	Software Examples .....	16
6	IDE Minimum Requirements for MSP-EXP430G2ET .....	16
7	Source File .....	16
8	Source File and Folders .....	17
9	How MSP Device Documentation is Organized .....	21

**Trademarks**

LaunchPad, MSP430, BoosterPack, Code Composer Studio, EnergyTrace, 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 MSP430G2553 16-bit MCU has 16KB of flash, 512 bytes of RAM, up to 16-MHz CPU speed, an 8-channel 10-bit ADC, capacitive-touch enabled I/Os, a universal serial communication interface, and more – plenty to get started in your development.

Rapid prototyping is simplified by the 20-pin BoosterPack™ plug-in module headers that 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. You can either design your own BoosterPack plug-in module or choose among many already available from TI and third-party developers.

The LaunchPad development kit features an integrated DIP target socket that supports up to 20 pins, allowing MSP430 entry-level MCUs to be plugged into the LaunchPad development kit. The MSP-EXP430G2ET LaunchPad development kit comes with an MSP430G2553 MCU by default. The MSP430G2553 MCU has the most memory available of the compatible entry-level MCUs.

Free software development tools are also available, such as TI's Eclipse-based [Code Composer Studio™ IDE \(CCS\)](#) and [IAR Embedded Workbench® for MSP430 IDE \(IAR EW430\)](#). Both of these IDEs support EnergyTrace™ technology for real-time power profiling and debugging when paired with the MSP430G2553 LaunchPad development kit. More information about the LaunchPad development kit, including documentation and design files, can be found on the [MSP430G2553 LaunchPad development kit tool page](#).

### 1.2 Key Features

- High-quality 20-pin DIP socket for an easy plug-in or removal of the target MCU
- Supports MSP430G2xx1, MSP430G2xx2, MSP430G2xx3, and MSP430F20xx MCUs in PDIP14 or PDIP20 packages (see [Section 2.9](#) for a list of supported MCUs)
- 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
- 1 button and 3 LEDs for user interaction

### 1.3 What's Included

#### 1.3.1 Kit Contents

- 1x MSP-EXP430G2ET LaunchPad development kit
- 1x micro USB-B cable
- 1x 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. It 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. The green power and yellow LDO LEDs should illuminate. For proper operation, drivers are needed. TI recommends installing the drivers by installing an IDE such as TI CCS or IAR EW430. Drivers are also available at <http://www.ti.com/MSPdrivers>.

## 1.4.2 Running the Out-of-Box Experience

The LaunchPad development kit includes a pre-programmed MSP430G2553 MCU already installed in the target socket. When the LaunchPad development kit is connected through USB, the Out-of-Box Experience (OOBE) demo starts with a two LED toggle sequence.

Press button P1.3 to switch the application to the Live Temperature mode. The LaunchPad development kit should start streaming live temperature data to the PC to be visualized in the [MSP-EXP430G2ET OOBE GUI](#) or displayed in a serial terminal. A reference temperature is taken at the beginning of this mode, and the LEDs of the LaunchPad development kit signal a rise or fall in temperature by varying the brightness of the on-board red or green LED, respectively. The reference temperature can also be recalibrated to the ambient temperature with another button press on P1.3.

You can influence the temperature of the MCU by blowing hot or cold air onto it, and you can then see changes in the LED brightness or data changes on the GUI.

This GUI is created with [GUI Composer 2.0](#) with the source available for customization, imported from the [TI Cloud Gallery](#). The serial communication port on the PC must be configured with 9600 bps, one stop bit, and no flow control to display the values correctly.

---

**NOTE:** The OOB cloud GUI is supported in only the latest version of Chrome, Firefox, and Safari browsers. An installer for the offline standalone GUI can also be downloaded from the [TI Cloud Gallery](#).

---

## 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. 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-EXP430G2ET hardware.

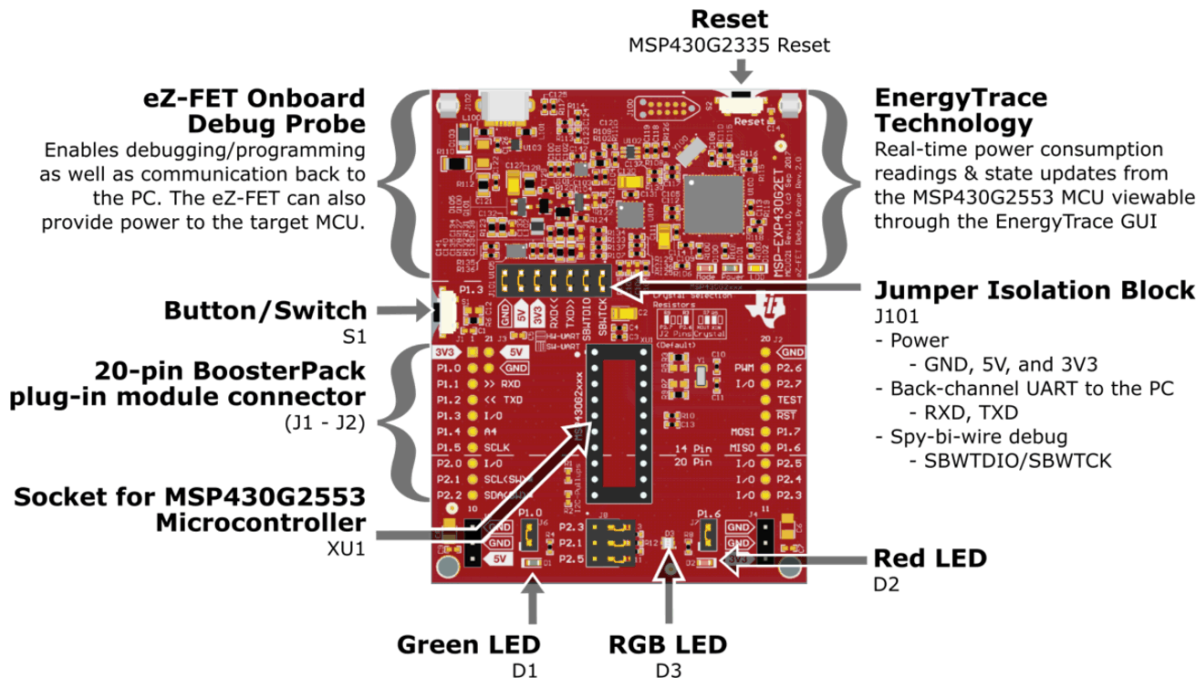


Figure 2. MSP-EXP430G2ET Overview

### 2.1 Block Diagram

Figure 3 shows the block diagram.

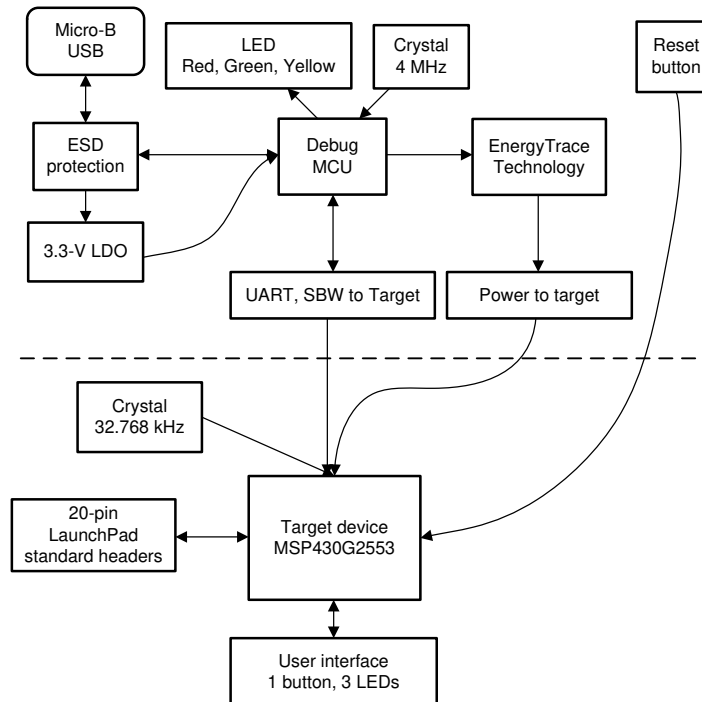


Figure 3. MSP-EXP430G2ET Block Diagram

## 2.2 Hardware Features

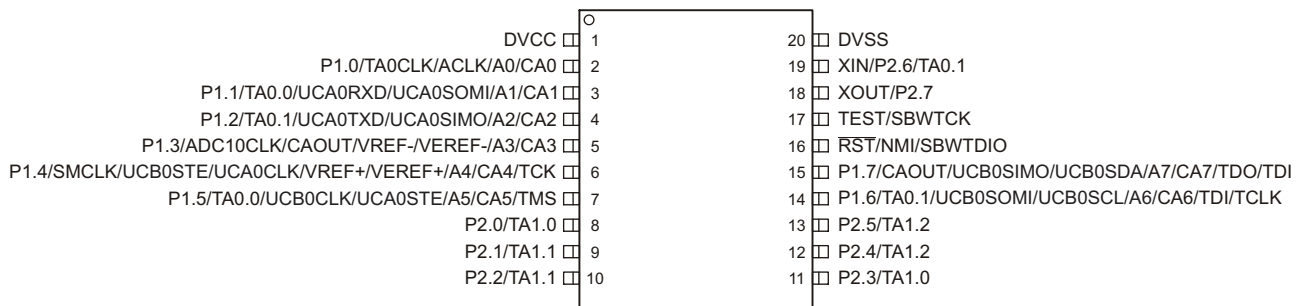
### 2.2.1 MSP430G2553 MCU

The MSP430G2553 is a member of the MSP430 family of ultra-low-power MCUs. MSP430 MCUs features different sets of peripherals targeted for various applications. The MCU architecture, combined with five low-power modes, is optimized to achieve extended battery life in portable measurement applications.

Device features include:

- 1.8-V to 3.6-V operation
- 16-bit RISC architecture up to 16-MHz system clock
- 16KB of flash memory and 512 bytes of SRAM
- 8-channel 10-bit ADC
- 8-channel comparator
- Two 16-bit timers with three capture/compare registers (Timer\_A)
- 24 GPIOs
- One universal serial communication interface (USCI\_A) supports UART, IrDA, and SPI
- One USCI (USCI\_B) supports SPI and I<sup>2</sup>C

Figure 4 shows the pinout of the MSP430G2553 20-pin N (PDIP) package.

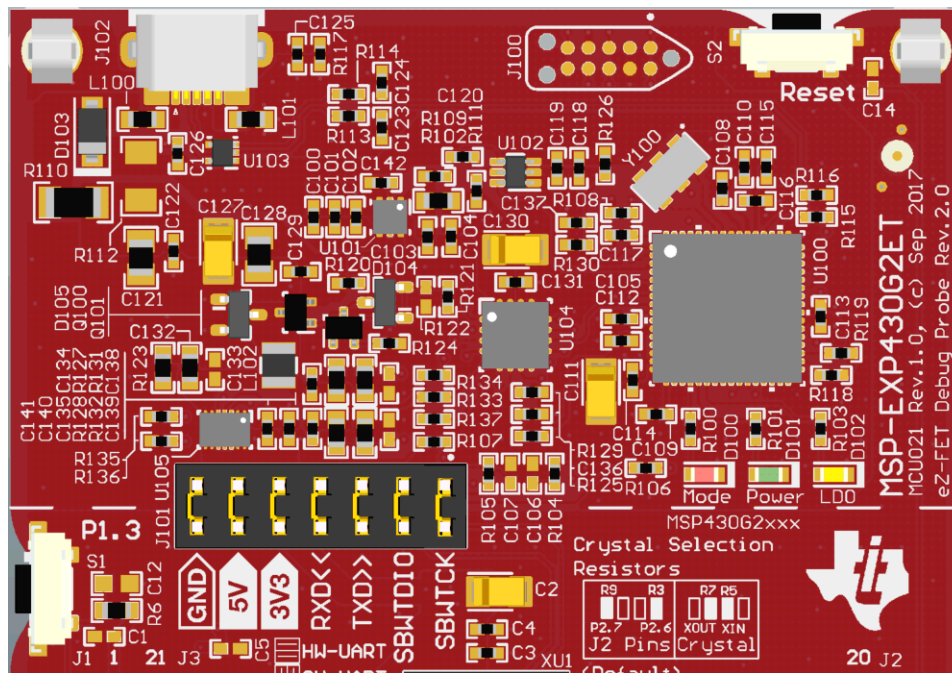


**Figure 4. MSP430G2553 20-Pin N Package (Top View)**

### 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-EXP430G2ET has the eZ-FET debug probe (see Figure 5), which is a simple and low-cost debugger that supports all MSP430 MCUs.




**Figure 5. eZ-FET Debug Probe**

The MSP-EXP430G2ET LaunchPad development kit features EnergyTrace technology but does not have support for EnergyTrace++ technology (see [Table 1](#)). The EnergyTrace technology functionality varies across the MSP430 portfolio.

**Table 1. EnergyTrace Technology**

Features	EnergyTrace Technology	EnergyTrace++ Technology
Current monitoring	✓	✓
CPU state		✓
Peripheral and system states		✓
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 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 details on the backchannel connection, see [Section 2.2.4](#).

For more information about the eZ-FET hardware, see the schematics in [Section 6](#) and the [Hardware Design Files](#). For more information about the software and the debugger, see the [eZ-FET wiki](#).

### 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 MSP430G2553 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:

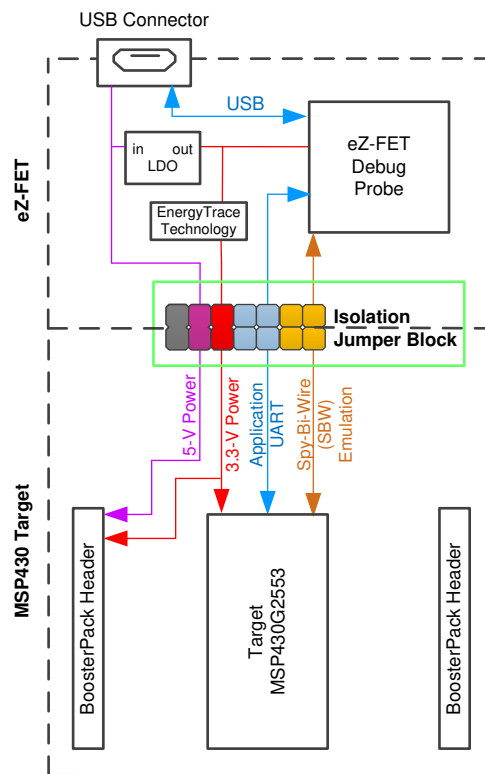
- To remove any and all influence from the eZ-FET debug probe for high-accuracy target power measurements



- 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 so that it can be used for MCUs other than the onboard MCU.

**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 G2553 receives data through this signal. The arrows indicate the direction of the signal.
TXD >>	Backchannel UART: The target G2553 sends data through this signal. The arrows indicate the direction of the signal.
SBW RST	Spy-Bi-Wire debug: SBWTDIO 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.



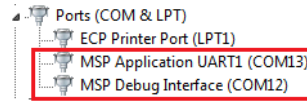
**Figure 6. eZ-FET Isolation Jumper Block Diagram**

### 2.2.4 Application (or Backchannel) UART

The backchannel UART allows communication with the USB host that is not part of the main functionality of the target application. This is useful during development and also provides a communication channel to the PC host. This communication 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 USCI\_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**

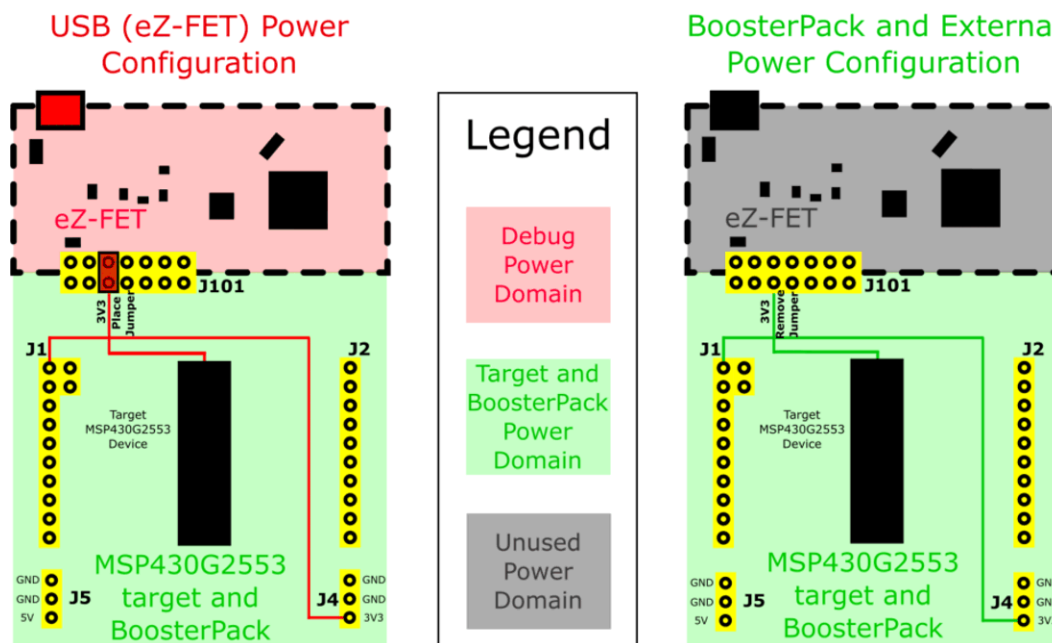
The backchannel UART is the "MSP Application UART1" port. In this case, Figure 7 shows COM13, but this port can vary from one host PC to the next. After you identify the correct COM port, configure it in your host application according to its documentation. You can then open the port and begin communication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

On the target MSP430G2553 side, the backchannel is connected to the USCI\_A0 module. The eZ-FET has a configurable baud rate; therefore, it is important to configure the baud rate of the PC application to the same rate as the USCI\_A0.

**NOTE:** The backchannel UART connection in the isolation jumper block can also be configured to use a SW UART. The SW UART is done through the TimerA0 module located on pins P1.1 and P1.2. If the jumpers are in the horizontal position, the HW UART should be used. If the jumpers are in the vertical position, the SW UART should be used. The orientation guidelines are also printed in silkscreen on the board. It is recommended to use the HW UART configuration.

### 2.3 Power

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



**Figure 8. MSP-EXP430G2ET 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 J4 is present on the board to supply external power directly. It is important to comply with the MCU voltage operation specifications when supplying external power. The MSP430G2553 has an operating range of 1.8 V to 3.6 V. For more information, see the [MSP430G2x53, MSP430G2x13 Mixed-Signal Microcontrollers data sheet](#).

## 2.4 Measure MSP430 Current Draw

To measure the current draw of the MSP430G2553 using a multi-meter, use the 3V3 jumper on the J101 jumper isolation block. The current measured includes the target MCU and any current drawn through the BoosterPack plug-in module 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 MSP430G2553 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 MSP430G2553. 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. 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.

## 2.5 Clocking

The MSP-EXP430G2ET provides an external clock in addition to the internal clocks in the MCU.

- Y1: 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.

By default, the crystal is not connected to the MSP430G2553 because the target pins are multiplexed with two BoosterPack plug-in module header pins. 0-Ω resistors R3 and R9 must be removed, while R5 and R7 must be shorted across to connect the external crystal to the MSP430G2553. See the onboard crystal selection resistors silkscreen for how to configure the resistors to select between the crystal or the BoosterPack plug-in module pins.

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

- MCLK: DCO at 1 MHz
- SMCLK: DCO at 1 MHz
- ACLK: LFXT1 at 32.768 kHz

For more information about configuring internal clocks and using the external oscillators, see the [MSP430x2xx 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 MSP430G2553 target MCU.

1. 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.
2. Make sure the target board has proper connections for SBW. Note that to be compatible with SBW, the capacitor on RST/SBWDIO cannot be greater than 2.2 nF. The documentation for designing MSP430 JTAG interface circuitry is the [MSP430 Hardware Tools User's Guide](#).
3. Connect 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
  - SBWDIO
  - 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 development kit pinout standard. A standard was created to aid compatibility between LaunchPad development kit and BoosterPack plug-in module tools across the TI ecosystem.

While most BoosterPack plug-in modules are compliant with the standard, some are not. The MSP-EXP430G2ET LaunchPad development kit is compatible with all 20-pin BoosterPack plug-in modules that comply with the standard. If the reseller or owner of the BoosterPack plug-in module does not explicitly indicate compatibility with the MSP-EXP430G2ET LaunchPad development kit, compare the schematic of the candidate BoosterPack plug-in module with the LaunchPad development kit to ensure compatibility. Sometimes conflicts can be resolved by changing the MSP430G2553 pin function configuration in software.

To check the compatibility of your desired BoosterPack plug-in modules for your design, with a LaunchPad development kit of your choice, you can use the [BoosterPack Checker tool](#) (see [Figure 9](#)). This allows you to select any LaunchPad development kit we offer 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.

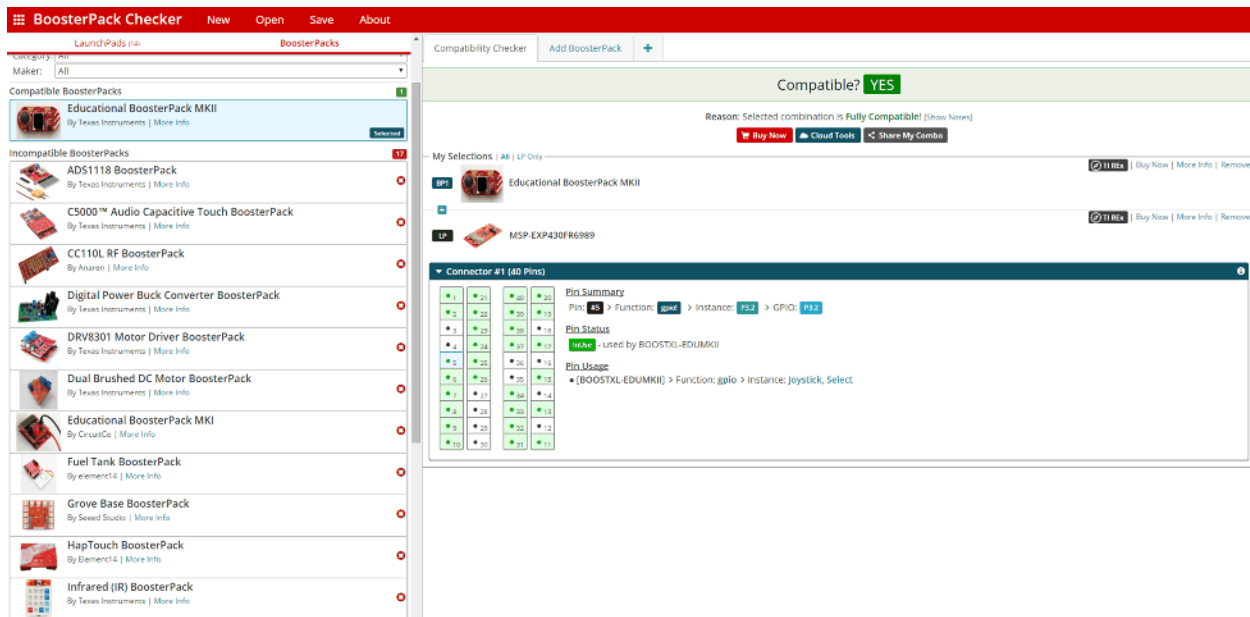


Figure 9. BoosterPack Checker Tool

Figure 10 shows the 20-pin pinout of the connector from the LaunchPad development kit to a BoosterPack plug-in module.

Software 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 MSP430G2553 data sheet for more details on individual pin functions.

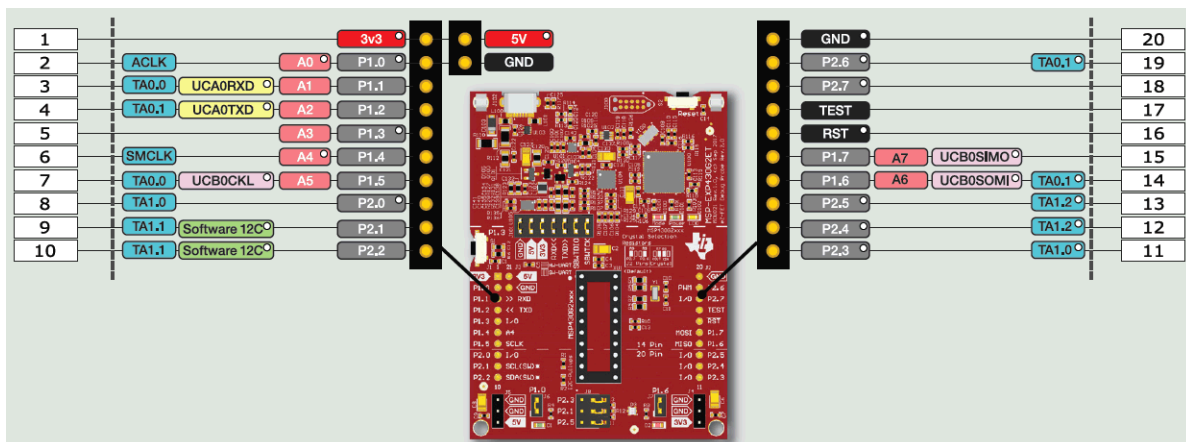


Figure 10. Pinout of Connector From LaunchPad Development Kit to BoosterPack Plug-in Module

## 2.8 20-Pin DIP Socket

The MSP-EXP430G2ET comes with the MSP430G2553 MCU plugged into the DIP target socket. However, both PDIP14 and PDIP20 packages of the MSP430G2xx entry-level MCUs and the MSP430F20xx MCUs can be inserted into the DIP socket aligned to pin 1 (see Figure 11). For a complete list of supported MCUs, see Section 2.9.

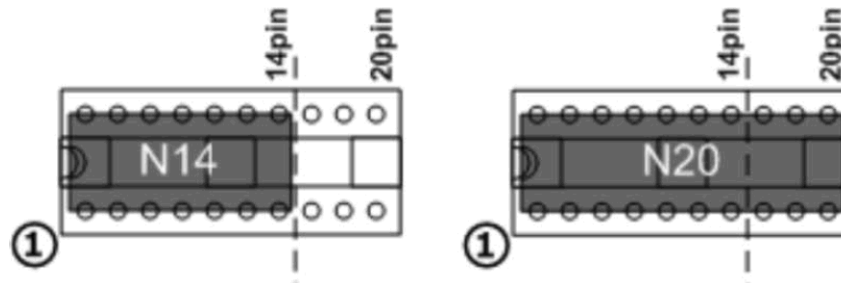


Figure 11. Insert Device Into Target Socket

## 2.9 Supported Devices

TI offers several MSP430 MCUs in a PDIP package that are compatible with this LaunchPad development kit. [Table 3](#) lists the supported MCUs.

Table 3. Supported Devices

Part Number	Family	Description
<a href="#">MSP430F2001</a>	F2xx	16-bit Ultra-Low-Power Microcontroller, 1KB Flash, 128B RAM, Comparator
<a href="#">MSP430F2002</a>	F2xx	16-bit Ultra-Low-Power Microcontroller, 1KB Flash, 128B RAM, 10-Bit SAR A/D, USI for SPI/I <sup>2</sup> C
<a href="#">MSP430F2003</a>	F2xx	16-bit Ultra-Low-Power Microcontroller, 1KB Flash, 128B RAM, 16-Bit Sigma-Delta A/D, USI for SPI/I <sup>2</sup> C
<a href="#">MSP430F2011</a>	F2xx	16-bit Ultra-Low-Power Microcontroller, 2KB Flash, 128B RAM, Comparator
<a href="#">MSP430F2012</a>	F2xx	16-bit Ultra-Low-Power Microcontroller, 2KB Flash, 128B RAM, 10-Bit SAR A/D, USI for SPI/I <sup>2</sup> C
<a href="#">MSP430F2013</a>	F2xx	16-bit Ultra-Low-Power Microcontroller, 2KB Flash, 128B RAM, 16-Bit Sigma-Delta A/D, USI for SPI/I <sup>2</sup> C
<a href="#">MSP430G2001</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 512B Flash, 128B RAM
<a href="#">MSP430G2101</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 1KB Flash, 128B RAM
<a href="#">MSP430G2111</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 1KB Flash, 128B RAM, Comparator
<a href="#">MSP430G2121</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 1KB Flash, 128B RAM, USI for SPI/I <sup>2</sup> C
<a href="#">MSP430G2131</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 1KB Flash, 128B RAM, 10-Bit SAR A/D, USI for SPI/I <sup>2</sup> C
<a href="#">MSP430G2201</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 2KB Flash, 128B RAM
<a href="#">MSP430G2211</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 2KB Flash, 128B RAM, Comparator
<a href="#">MSP430G2221</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 2KB Flash, 128B RAM, USI for SPI/I <sup>2</sup> C
<a href="#">MSP430G2231</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 2KB Flash, 128B RAM, 10-Bit SAR A/D, USI for SPI/I <sup>2</sup> C
<a href="#">MSP430G2102</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 1KB Flash, 256B RAM, USI for SPI/I <sup>2</sup> C
<a href="#">MSP430G2202</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 2KB Flash, 256B RAM, USI for SPI/I <sup>2</sup> C
<a href="#">MSP430G2302</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 4KB Flash, 256B RAM, USI for SPI/I <sup>2</sup> C
<a href="#">MSP430G2402</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 8KB Flash, 256B RAM, USI for SPI/I <sup>2</sup> C
<a href="#">MSP430G2112</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 1KB Flash, 256B RAM, Comparator, USI for SPI/I <sup>2</sup> C
<a href="#">MSP430G2212</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 2KB Flash, 256B RAM, Comparator, USI for SPI/I <sup>2</sup> C
<a href="#">MSP430G2312</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 4KB Flash, 256B RAM, Comparator, USI for SPI/I <sup>2</sup> C
<a href="#">MSP430G2412</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 8KB Flash, 256B RAM, Comparator, USI for SPI/I <sup>2</sup> C
<a href="#">MSP430G2132</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 1KB Flash, 256B RAM, 10-Bit SAR A/D, USI for SPI/I <sup>2</sup> C
<a href="#">MSP430G2232</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 2KB Flash, 256B RAM, 10-Bit SAR A/D, USI for SPI/I <sup>2</sup> C
<a href="#">MSP430G2332</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 4KB Flash, 256B RAM, 10-Bit SAR A/D, USI for SPI/I <sup>2</sup> C
<a href="#">MSP430G2432</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 8KB Flash, 256B RAM, 10-Bit SAR A/D, USI for SPI/I <sup>2</sup> C
<a href="#">MSP430G2152</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 1KB Flash, 256B RAM, 10-Bit SAR A/D, Comparator, USI for SPI/I <sup>2</sup> C
<a href="#">MSP430G2252</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 2KB Flash, 256B RAM, 10-Bit SAR A/D, Comparator, USI for SPI/I <sup>2</sup> C
<a href="#">MSP430G2352</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 4KB Flash, 256B RAM, 10-Bit SAR A/D, Comparator, USI for SPI/I <sup>2</sup> C



**Table 3. Supported Devices (continued)**

Part Number	Family	Description
<a href="#">MSP430G2452</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 8KB Flash, 256B RAM, 10-Bit SAR A/D, Comparator, USI for SPI/I <sup>2</sup> C
<a href="#">MSP430G2153</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 1KB Flash, 256B RAM, 10-Bit SAR A/D, Comparator, USCI for I <sup>2</sup> C/SPI/UART
<a href="#">MSP430G2203</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 2KB Flash, 256B RAM, Comparator, USCI for I <sup>2</sup> C/SPI/UART
<a href="#">MSP430G2313</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 2KB Flash, 256B RAM, Comparator, USCI for I <sup>2</sup> C/SPI/UART
<a href="#">MSP430G2333</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 2KB Flash, 256B RAM, 10-Bit SAR A/D, Comparator, USCI for I <sup>2</sup> C/SPI/UART
<a href="#">MSP430G2353</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 2KB Flash, 256B RAM, 10-Bit SAR A/D, Comparator, USCI for I <sup>2</sup> C/SPI/UART
<a href="#">MSP430G2403</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 8KB Flash, 512B RAM,, Comparator, USCI for I <sup>2</sup> C/SPI/UART
<a href="#">MSP430G2413</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 8KB Flash, 512B RAM, Comparator, USCI for I <sup>2</sup> C/SPI/UART
<a href="#">MSP430G2433</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 8KB Flash, 512B RAM, 10-Bit SAR A/D, Comparator, USCI for I <sup>2</sup> C/SPI/UART
<a href="#">MSP430G2453</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 8KB Flash, 512B RAM, 10-Bit SAR A/D, Comparator, USCI for I <sup>2</sup> C/SPI/UART
<a href="#">MSP430G2513</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 16KB Flash, 512B RAM, Comparator, USCI for I <sup>2</sup> C/SPI/UART
<a href="#">MSP430G2533</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 16KB Flash, 512B RAM, 10-Bit SAR A/D, Comparator, USCI for I <sup>2</sup> C/SPI/UART
<a href="#">MSP430G2553</a>	G2xx	16-bit Ultra-Low-Power Microcontroller, 16KB Flash, 512B RAM, 10-Bit SAR A/D, Comparator, USCI for I <sup>2</sup> C/SPI/UART

## 2.10 Design Files

### 2.10.1 Hardware

See [Section 6](#) for the schematics. All design files including schematics, layout, bill of materials (BOM), Gerber files, and documentation are available in the [MSP-EXP430G2ET Hardware Design Files](#).

### 2.10.2 Software

All design files including TI-TXT object-code firmware images, software example projects, and documentation are available in the [MSP-EXP430G2ET LaunchPad development kit download page](#).

## 2.11 Hardware Change log

[Table 4](#) lists the revision history of the MSP430G2553 LaunchPad development kit hardware.

**Table 4. Hardware Change Log**

PCB Revision	Description
Rev 1.0	Initial Release

## 3 Software Examples

Two software examples are included with the MSP430G2553 LaunchPad development kit (see [Table 5](#)), which can be found in the [MSP-EXP430G2ET LaunchPad development kit download page](#) and are also available inside [MSPWare](#).



**Table 5. Software Examples**

Demo Name	BoosterPack Plug-in Module Required	Description	More Details
OutOfBox_MSP-EXP430G2ET	None	The out-of-box demo preprogrammed on the LaunchPad development kit from the factory. Demonstrates features of the MSP430G2553 MCU.	<a href="#">Section 3.1</a>
BlinkLED_MSP-EXP430G2ET	None	Blinks an LED on the LaunchPad development kit at a fixed interval.	<a href="#">Section 3.2</a>

To use any of the software examples with the LaunchPad development kit, you must have an integrated development environment (IDE) that supports the MSP430G2553 MCU.

**Table 6. IDE Minimum Requirements for MSP-EXP430G2ET**

Code Composer Studio IDE	IAR Embedded Workbench for MSP430 IDE
v7.0 or later	v7.11.1 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 on the EVM.

The Out-of-Box demo for the MSP-EXP430G2ET LaunchPad development kit starts with a two LED toggle sequence. The demo also implements a real-time temperature sensor.

#### 3.1.1 Source File

The OOBE project includes one main.c source file (see [Table 7](#)).

**Table 7. Source File**

Name	Description
main.c	Contains the main Out-of-Box demo and auxiliary functions

#### 3.1.2 Overview

The online cloud-based [MSP-EXP430G2ET OOBE 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: 9600, 8, 1, n).

When powering up the Out-of-Box demo, the MSP-EXP430G2ET LaunchPad development kit starts with a two LED toggle sequence. At any time, press S1 to switch to the Live Temperature mode.

#### 3.1.3 Live Temperature Mode

In this mode, the LaunchPad development kit repeatedly measures the MSP430G2553 MCU internal temperature and transfers the data to the PC through UART.

A reference temperature is taken at the beginning of this mode, and the LEDs of the LaunchPad development kit signal a rise or fall in temperature by varying the brightness of the on-board red or green LED, respectively. The reference temperature can also be recalibrated with another button press on P1.3.

The application keeps track of the temperature threshold, and when a new temperature data is acquired, it is compared against the threshold. If measured temperature is below the threshold, the red LED will illuminate, and if the measured temperature is above the threshold, the green LED will illuminate.

Pressing S1 will recalibrate the temperature threshold in this mode. The further the recorded temperature is from the threshold, the brighter the corresponding LEDs will illuminate.

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

The demo application uses the on-chip peripherals of the MSP430G2553 MCU such as the 10-bit ADC, which samples the internal temperature sensor, and 16-bit timers, which drive the PWM to vary brightness of the LEDs and enable software UART for communication with the PC. The MSP430G2553 offers a USCI interface that is capable of communicating through UART at up to 2 MBaud, but to be aligned with all the other MSP430G2xx MCUs, the demo uses the Timer UART implementation, which can be used on all the other MCUs. This way the demo can be used with any other MSP430G2xx MCU with an integrated ADC, without any change in the program.

## 3.2 **Blink LED Example**

This 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 Blink LED project includes one main.c source file (see [Table 8](#)).

**Table 8. Source File and Folders**

Name	Description
main.c	The Blink LED main function

The main code uses register level access code 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 (CCS) or 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.

Try TI Resource Explorer Cloud now at <https://dev.ti.com/>.

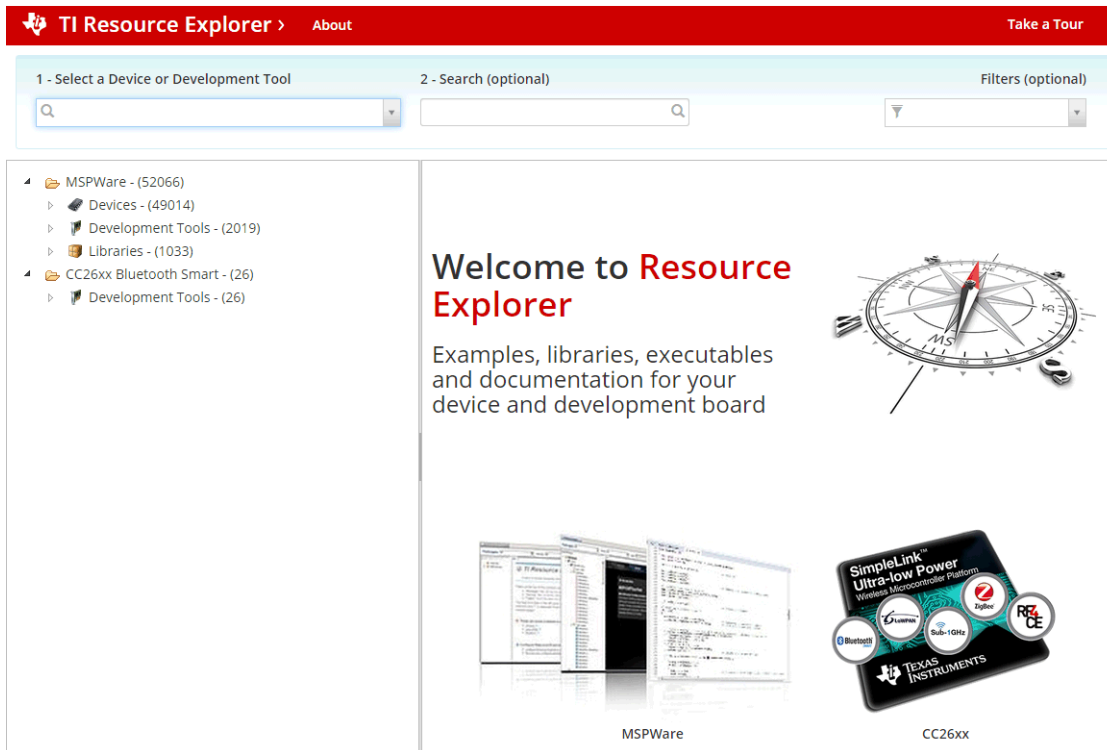


Figure 12. TI Resource Explorer Cloud

#### 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 comparison of CCS Cloud and CCS Desktop, visit [Should I use CCS Cloud or CCS Desktop](#).

Visit Code Composer Studio Cloud now at <http://dev.ti.com>.

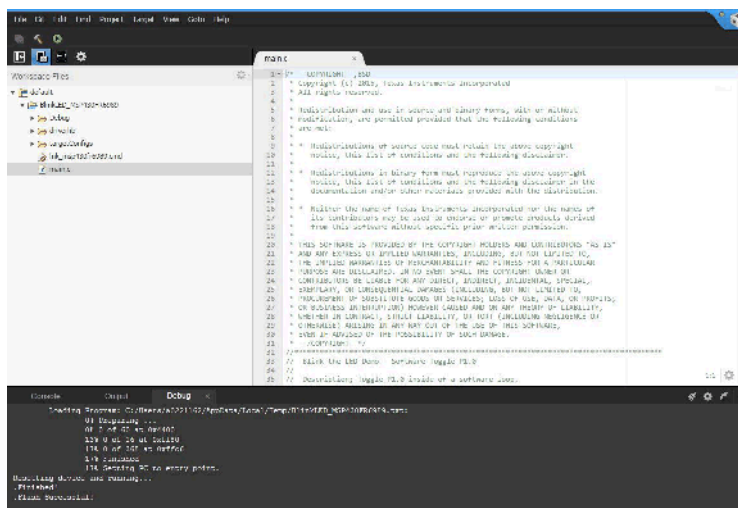


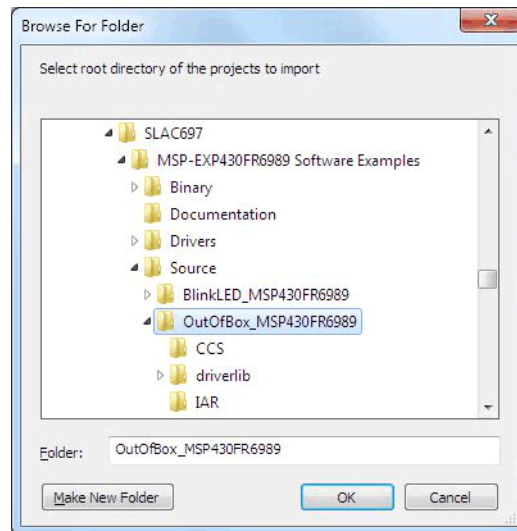
Figure 13. CCS Cloud

### 4.1.2 Code Composer Studio IDE

Code Composer Studio (CCS) 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.

You can learn more about CCS and download it at <http://www.ti.com/tool/ccstudio>.

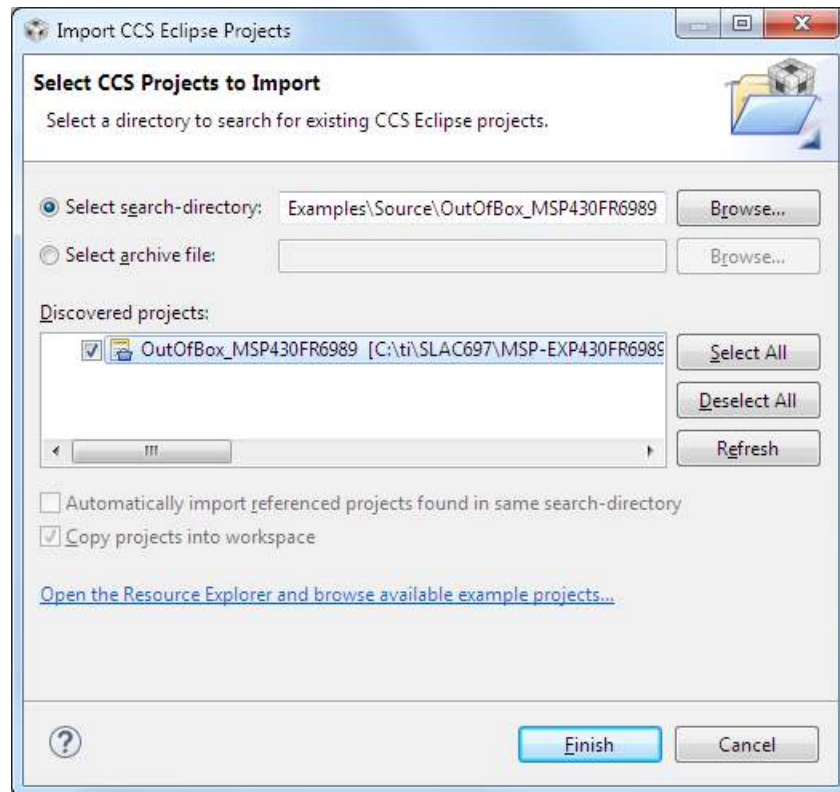
CCS v7.0 or higher is required. When CCS has been launched, and a workspace directory chosen, use Project>Import Existing CCS Eclipse Project. Direct it to the desired demo project directory that contains main.c.



**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.



**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 430 IDE

IAR Embedded Workbench for MSP430 (IAR EW430) 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.

You can learn more about IAR Embedded Workbench for MSP430 and download it at the [IAR website](#).

IAR EW430 v7.11.1 or higher is required. To open the demo in IAR EW430, 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 EW430 users can 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 Development Kit Websites

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

- [MSP-EXP430G2ET 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 will help you quickly find what you need inside packages including MSPWare, ControlSuite™ libraries, 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 for more information.

MSPWare is a collection of code examples, software libraries, data sheets and other design resources for all MSP430 MCUs 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 talk to MSP hardware. More information can be found at <http://www.ti.com/tool/mspware>.

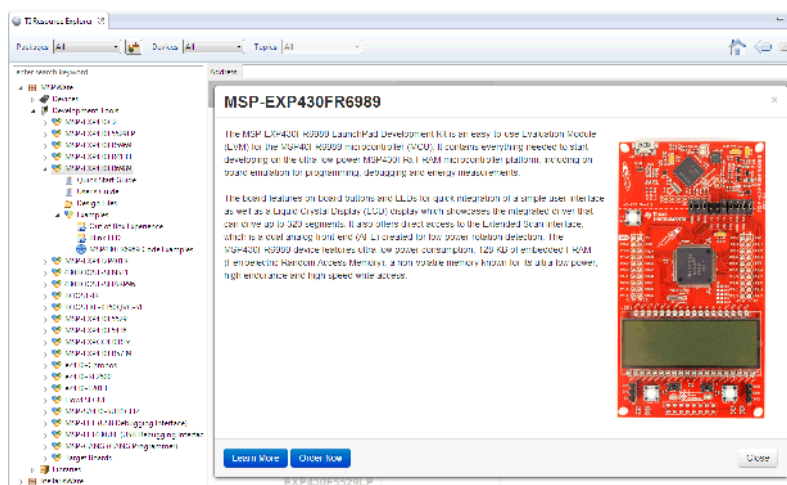


Figure 16. Using TI Resource Explorer to Browse MSP-EXP430G2ET in MSPWare

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

### 4.4 MSP430G2553 MCU

#### 4.4.1 Device Documentation

At some point, you will probably want more information about the MSP430G2553 MCU. For every MSP430 MCU, the documentation is organized as shown in Table 9.

Table 9. How MSP Device Documentation is Organized

Document	For MSP430G2553	Description
Device family user's guide	<a href="#">MSP430x2xx Family User's Guide</a>	Architectural information about the MCU, including all modules and peripherals such as clocks, timers, ADC, and so on.
Device-specific data sheet	<a href="#">MSP430G2x53, MSP430G2x13 Mixed-Signal Microcontrollers data sheet</a>	Device-specific information and all parametric information for this MCU

#### 4.4.2 MSP430G2553 Code Examples

[MSP430G2x53](#), [MSP430G2x33](#), [MSP430G2x13](#), [MSP430G2x03 Code Examples](#) is a set of simple C examples that demonstrate how to use the entire set of peripherals on the MSP430G2553 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.

#### 4.4.3 MSP430 Application Notes and TI Designs

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

### 4.5 Community Resources

#### 4.5.1 TI E2E™ Community

Search the forums at <https://e2e.ti.com>. If you cannot find your answer, post your question to the community.

#### 4.5.2 Community at Large

Many online communities focus on the LaunchPad development kit – for example, <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 terminal application and the USCI 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 USCI module.

### **Q: The device is not answering to any communication, JTAG or UART.**

A: If you are experiencing difficulties in communicating to the attached MSP430 target MCU, even though all the communication drivers for the MSP-EXP430G2ET are loaded correctly, the emulator is probably set to a wrong communication state. This can be fixed by reconnecting the LaunchPad development kit and restarting the communicating application. Also make sure that all the jumpers on J101 are connected properly between the emulator and the target MCU.

6 Schematics

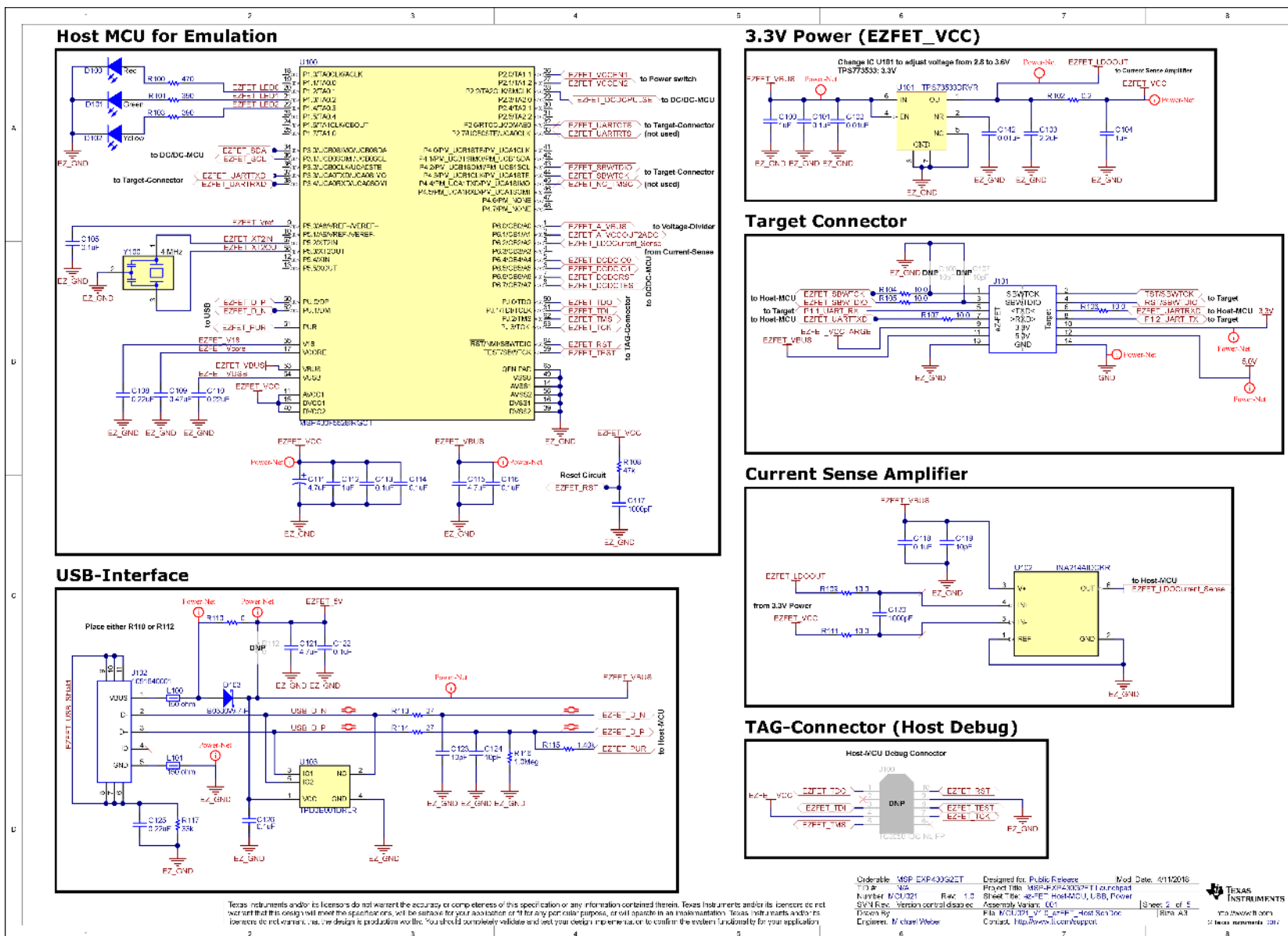
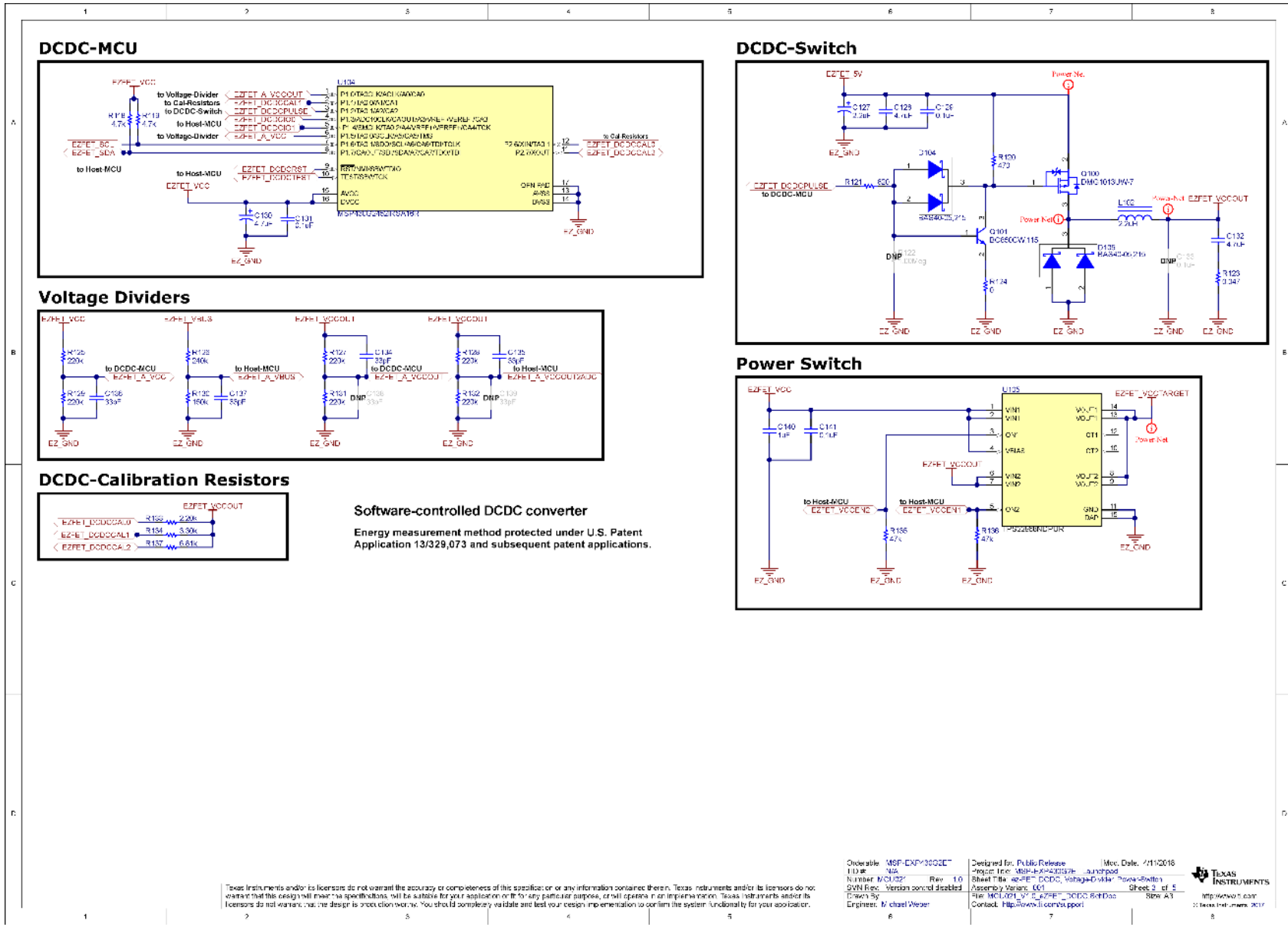


Figure 17. Schematics (1 of 3)



**Software-controlled DCDC converter**  
 Energy measurement method protected under U.S. Patent Application 13/329,073 and subsequent patent applications.

Texas Instruments and/or its licensors do not warrant the accuracy or completeness of this specification or any information contained therein. Texas Instruments and/or its licensors do not warrant that this design will meet the specifications, will be suitable for your application or fit for any particular purpose, or will operate in an impervious or "Texas Instruments and/or its licensors do not warrant that the design is production-ready. You should completely validate and test your design implementation to confirm the system functionality for your application.

Figure 18. Schematics (2 of 3)



## Revision History

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

<b>Changes from May 1, 2018 to March 20, 2020 (from * Revision (20180501) to A Revision)</b>	<b>Page</b>
• Added a Note to Section 2.2.4 .....	10
• Removed all mentions of 16 or 24 Capacitive-Touch Enabled I/O Pins.....	14

## IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), 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 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](#) or other applicable terms available either on [ti.com](http://ti.com) or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

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

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2022, Texas Instruments Incorporated