MSP430FR5994 LaunchPad™ Development Kit (MSP-EXP430FR5994)

The MSP-EXP430FR5994 LaunchPad™ development kit is an easy-to-use evaluation module (EVM) for the MSP430FR5994 microcontroller (MCU). It contains everything needed to start developing on the ultra-low-power MSP430FRx FRAM microcontroller platform, including onboard debug probe for programming, debugging, and energy measurements. Figure 1 shows the development kit.

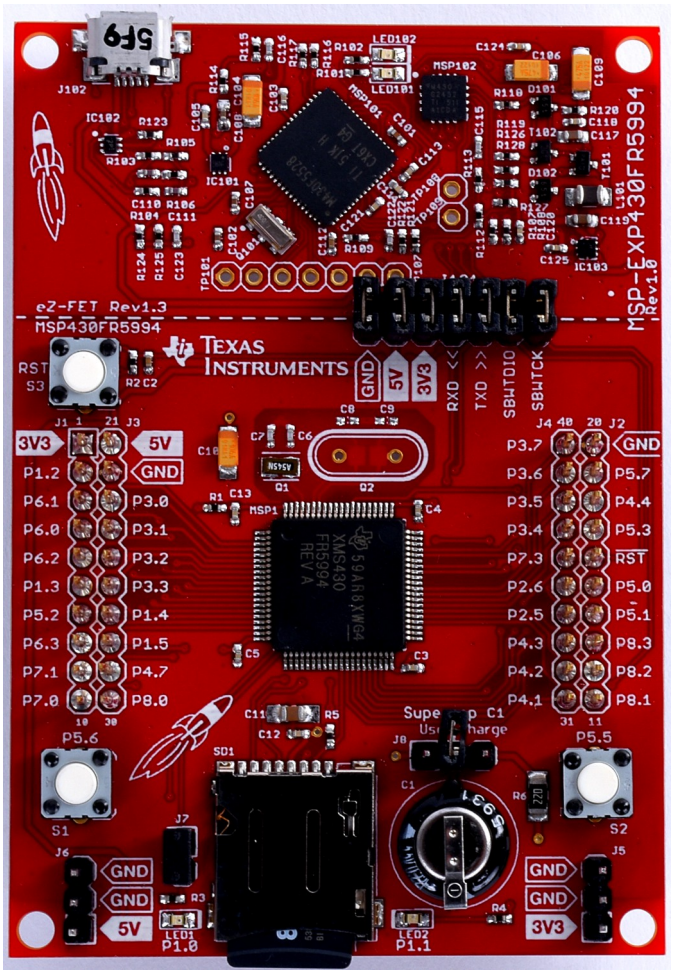

Figure 1. MSP-EXP430FR5994 LaunchPad Development Kit

Contents

List of Figures

Trademarks

1 Getting Started

1.1 Introduction

The MSP-EXP430FR5994 LaunchPad development kit is an easy-to-use evaluation module (EVM) for the MSP430FR5994 microcontroller (MCU). The LaunchPad development kit contains everything needed to start developing on the ultra-low-power MSP430FRx FRAM microcontroller platform, including onboard debug probe for programming, debugging and energy measurements. The board features onboard buttons and LEDs for quick integration of a simple user interface, a microSD card port to interface with microSD cards, and a super capacitor (super cap) to enable stand-alone applications without an external power supply.

The MSP430FR5994 MCU features 256KB of embedded FRAM (Ferroelectric Random Access Memory), a nonvolatile memory known for its ultra-low power, high endurance, and high speed write access. The device also features 8KB of SRAM, supports CPU speeds of up to 16 MHz and has integrated peripherals for communication, ADC, timers, AES encryption, low-energy accelerator (LEA) (a new hardware module for the FRAM family that is designed for fast, efficient, and low-power vector math), and more—plenty to get you started in your development.

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

Free software development tools are also available, such as the TI Eclipse-based Code Composer Studio™ IDE (CCS) and the IAR Embedded Workbench® for MSP430™ IDE (EW430). Both of these IDEs support EnergyTrace++™ technology for real-time power profiling and debugging when paired with the MSP430FR5994 LaunchPad kit.

1.2 Key Features

- MSP ULP FRAM technology based MSP430FR5994 16-bit MCU
- EnergyTrace++ Technology available for ultra-low-power debugging
- 40-pin LaunchPad development kit standard leveraging the BoosterPack plug-in module ecosystem
- Onboard eZ-FET debug probe
- Two buttons and two LEDs for user interaction
- Onboard microSD card
- Super capacitor (0.22 F)

1.3 What's Included

1.3.1 Kit Contents

- MSP-EXP430FR5994 LaunchPad development kit
- Micro-USB cable
- Quick start guide

1.3.2 Software Examples

- Out-of-box software
- Blink LED
- Audio BoosterPack plug-in module record and playback
- Low-energy accelerator for signal processing
- EEPROM emulation

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. It 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 drivers by installing an IDE such as TI CCS or IAR EW430. Drivers are also available at ti.com/MSPdrivers.

1.4.2 Running the Out-of-Box Demo

When connected to the computer, the LaunchPad development kit powers up. Press and hold the S1 and S2 buttons simultaneously to select a new mode. See Section 3.1 for a detailed explanation of each mode.

1.4.2.1 Live Temperature Mode

This mode provides live temperature data streaming to the PC GUI. You can influence the temperature of the device and see changes on the GUI.

1.4.2.2 FRAM Data Log Mode

This mode shows the FRAM data logging capabilities of the MSP430FR5994. After starting this mode, the LaunchPad development kit wakes up every five seconds from sleep mode (indicated by LED blink) to log both temperature and input voltage values. After reconnecting to the GUI, these values can be uploaded and graphed in the GUI.

1.4.2.3 SD Card Data Log Mode

This mode shows the data logging capabilities of the MSP430FR5994 while interfacing with an SD card. After starting this mode, the LaunchPad development kit wakes up every five seconds from sleep mode (indicated by LED blink) to log both temperature and input voltage values. After reconnecting to the GUI, these values can be uploaded and graphed in the GUI.

1.5 Next Steps: Looking Into the Provided Code

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

The quickest way to get started using the LaunchPad development kit is to use TI's Cloud Development Tools. The cloud-based Resource Explorer provides access to all of the examples and resources in 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-EXP430FR5994 hardware.

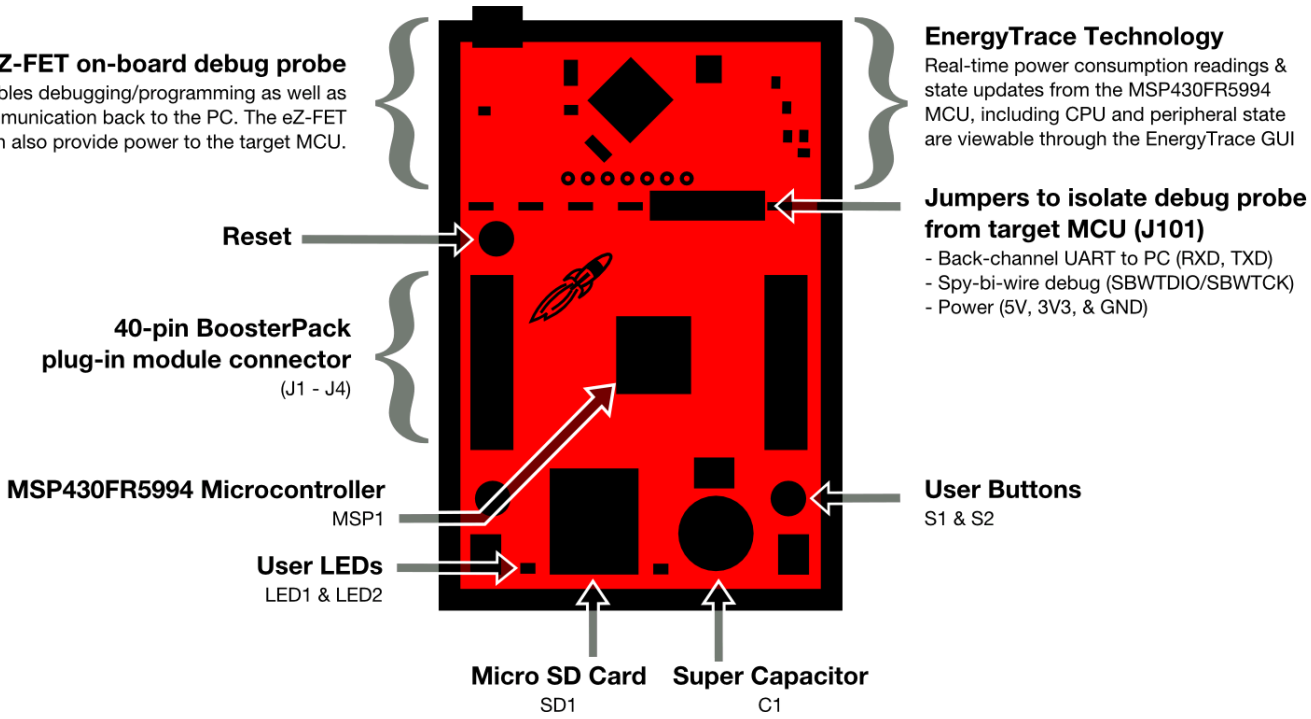

Figure 2. MSP-EXP430FR5994 Overview

2.1 Block Diagram

Figure 3 shows the block diagram.

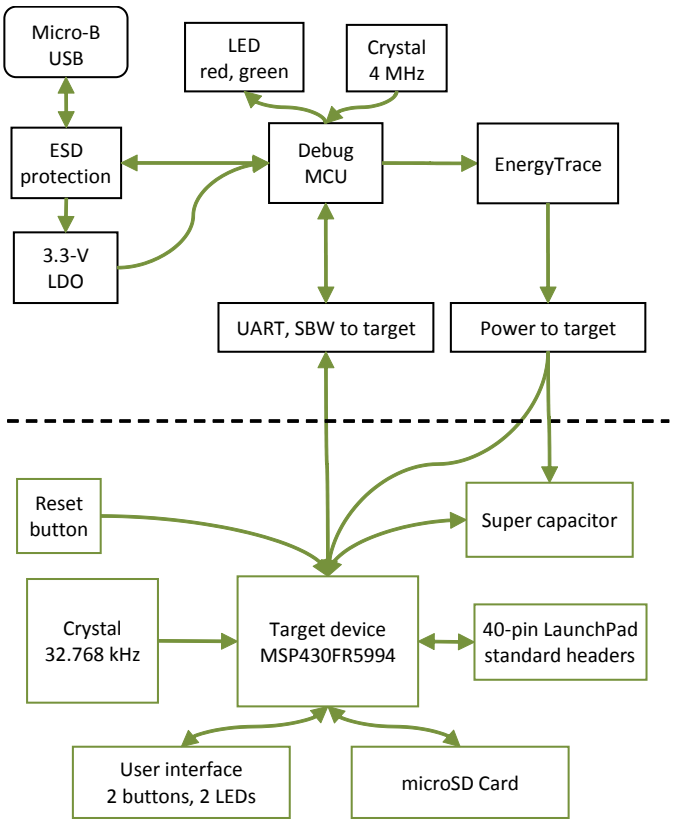

Figure 3. MSP-EXP430FR5994 Block Diagram

2.2 Hardware Features

2.2.1 MSP430FR5994 MCU

The MSP430FR5994 is the next device in TI's new ULP FRAM technology platform. FRAM is a cutting-edge memory technology that combines the best features of flash and RAM into one nonvolatile memory. For more information on FRAM, see www.ti.com/fram.

Device features include:

- 1.8-V to 3.6-V operation
- 16-bit RISC architecture up to 16-MHz system clock and 8-MHz FRAM access
- 256KB of FRAM and 8KB of SRAM
- 16-channel 12-bit ADC
- 16-channel analog comparator
- Six 16-bit timers with seven capture/compare registers each
- 6-channel direct memory access (DMA)
- 128-bit or 256-bit AES
- 32-bit hardware multiplier (MPY)
- 68 GPIOs

Figure 4 shows the pinout of the MSP430FR5994 MCU in the 80-pin PN package.

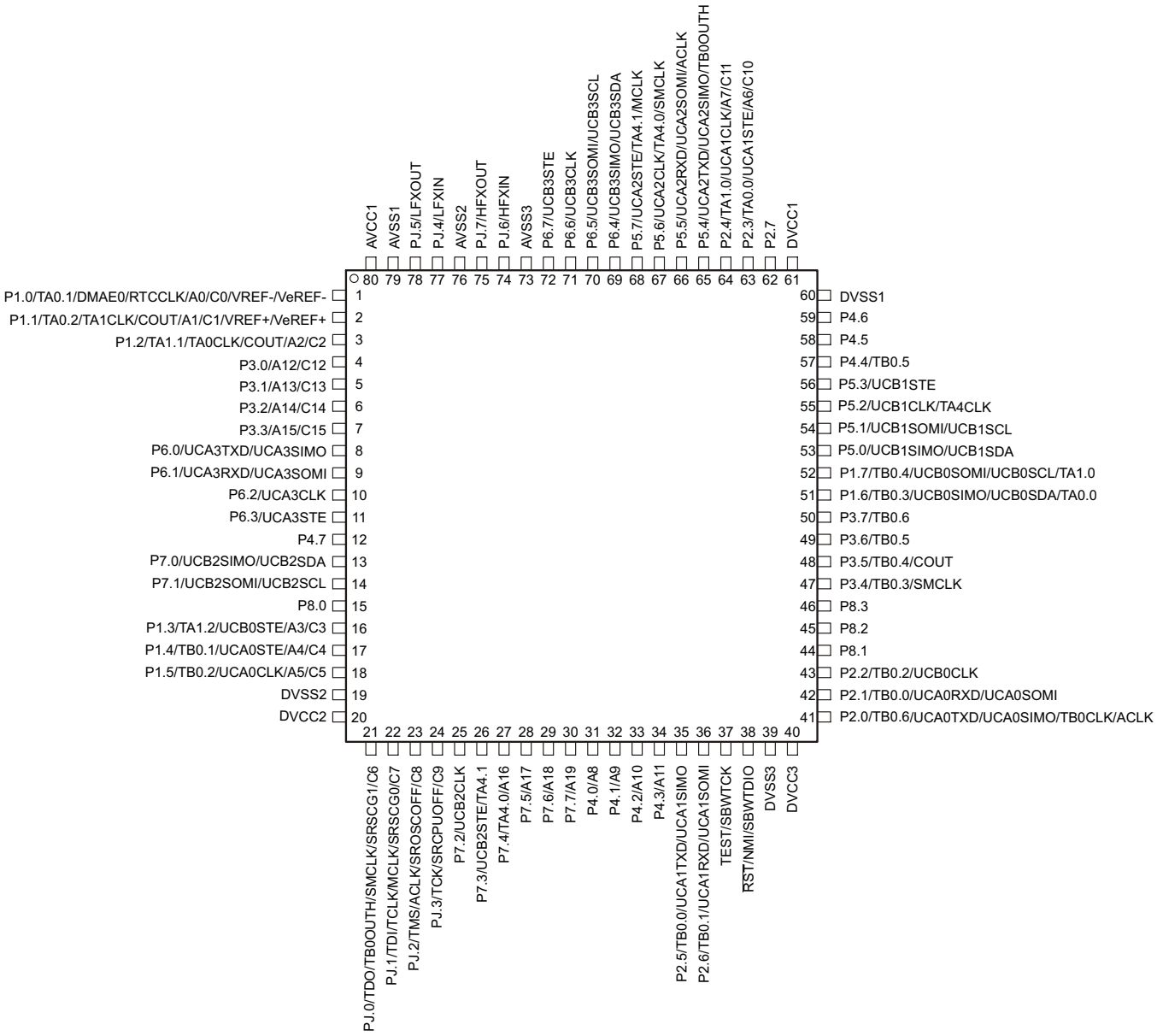

Figure 4. MSP430FR5994 Pinout

2.2.2 eZ-FET Onboard Debug Probe With EnergyTrace++ Technology

To keep development easy and cost effective, TI's LaunchPad development kits integrate an onboard debug probe, which eliminates the need for expensive programmers. The MSP-EXP430FR5994 has the eZ-FET debug probe (see Figure 5), which is a simple and low-cost debugger that supports all MSP430 device derivatives.

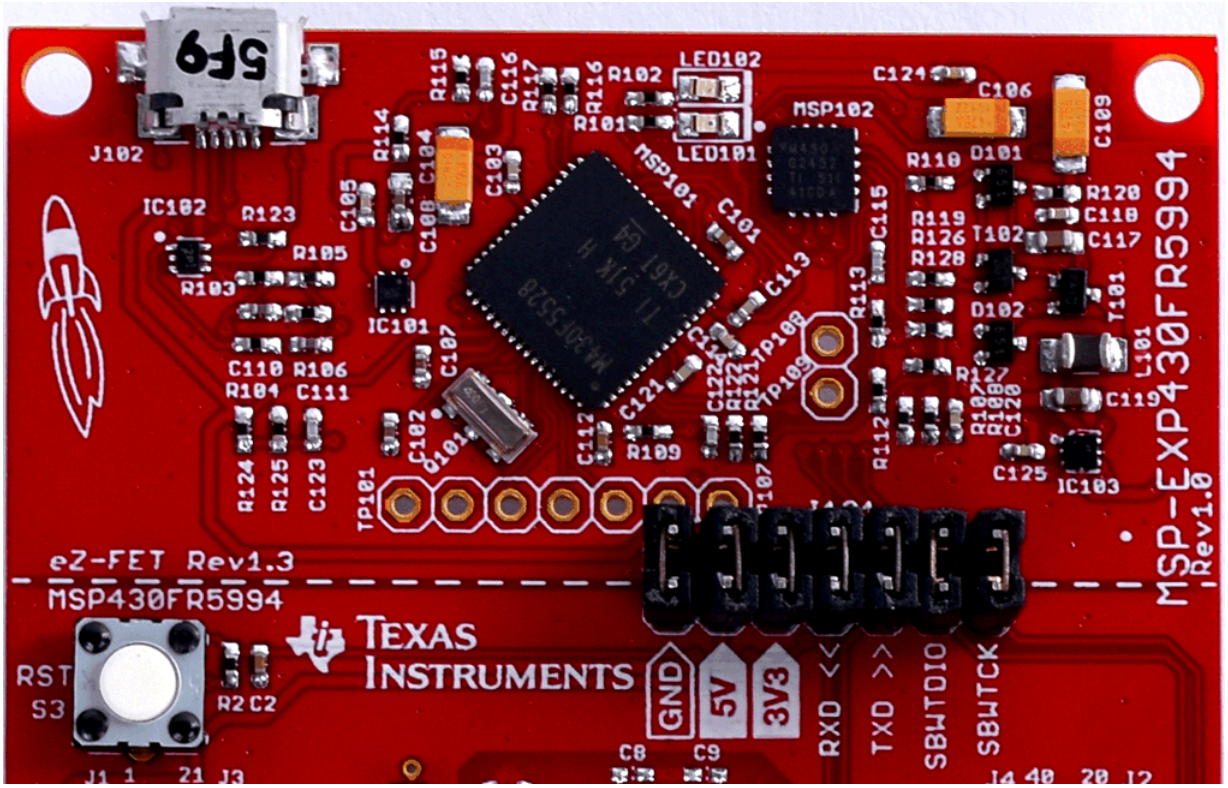

Figure 5. eZ-FET Debug Probe

The MSP-EXP430FR5994 LaunchPad development kit features full EnergyTrace++ technology. The EnergyTrace™ functionality varies across the MSP portfolio (see Table 1).

Table 1. EnergyTrace Technology

Features	EnergyTrace™ Technology	EnergyTrace++™ Technology
Current Monitoring	✓	✓
CPU State		✓
Peripheral and System State		✓
Devices Supported	All MSP430 MCUs	FR59xx and FR69xx MCUs
Development Tool Required	MSP-FET or eZ-FET	MSP-FET or eZ-FET

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

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

Details of the eZ-FET hardware can be found in the schematics in Section 6 and in the hardware design files download page. The software and more information about the debugger can be found on the eZ-FET wiki.

2.2.3 Debug Probe Connection: Isolation Jumper Block

The isolation jumper block at jumper J101 can connect or disconnect signals that cross from the eZ-FET domain into the MSP430FR5994 target domain. This includes eZ-FET Spy-Bi-Wire signals, application UART signals, and 3.3-V and 5-V power (see Table 2 and Figure 6).

Reasons to open these connections:

- To remove any and all influence from the eZ-FET debug probe for high accuracy target power measurements
- To control 3-V and 5-V power flow between the eZ-FET and target domains
- To expose the target MCU pins for other use than onboard debugging and application UART communication
- To expose the programming and UART interface of the eZ-FET so that it can be used for devices other than the onboard MCU.

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 MSP430FR5994 receives data through this signal. The arrows indicate the direction of the signal.
TXD >>	Backchannel UART: The target MSP430FR5994 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.

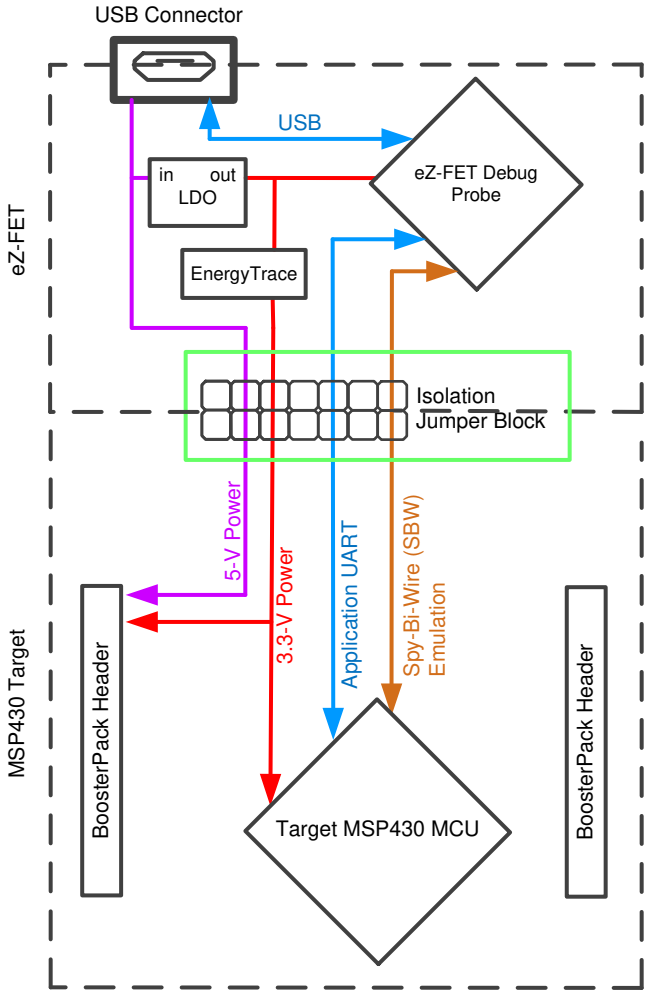

Figure 6. eZ-FET Isolation Jumper Block Diagram

2.2.4 Application (or Backchannel) UART

The backchannel UART allows communication with the USB host that is not part of the 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. This UART channel is separate from the UART on the 40-pin BoosterPack plug-in module connector (eUSCI_A3).

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 (see Figure 7).

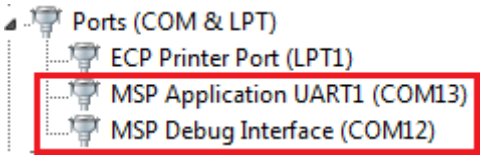

Figure 7. Application Backchannel UART in Device Manager

The backchannel UART is the *MSP Application UART1 (COM13)* port. In this case, Figure 7 shows COM13, but this port can vary from one host PC to the next. Identify the correct COM port and configure it in the host application according to its documentation. Then open the port and begin communication to it from the host.

On the target MSP430FR5994 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 microSD Card

The MSP430FR5994 LaunchPad development kit features an onboard microSD card. With an SD card, there is another method of cheap data logging available for users. The Out-Of-Box experience comes with an SD Card Library that helps users interface the MSP430FR5994 with the SD card. The library lets users store data in files with a name of their choice, so that the data can be used later in conjunction with a PC.

The slot can detect if a card is present. If a card is inserted while the MSP430FR5994 is on, an interrupt for the MCU is generated. Table 3 lists the SD Card Detect pin and the rest of the pin assignments that are used to communicate with the SD card.

Table 3. microSD Card to MCU Connections

microSD Card Function	MSP430FR5994 Pin
SD Card Detect	P7.2
SD SPI MOSI	P1.6
SD SPI MISO	P1.7
SD SPI CS	P4.0
SD SPI CLK	P2.2

R7 on the MSP430FR5994 LaunchPad development kit is not populated on the board to ensure accurate LPM current measurements. Use the internal MSP pullup resistor in software, or populate R7.

R5 is also not populated. In the SD Card library, the SPI CS line is driven high or low. Often why a pullup resistor like R5 is included is for during system startup. Before the MSP430FR5994 fully starts and outputs a high signal on the CS line, it is possible that CS may be floating, and the SD card may interpret other floating SPI lines as communication. It is a good practice to populate this resistor in an application where the exact startup conditions are not controlled. For the LaunchPad development kit, this resistor is removed for precise current measurements when all GPIO are set low.

2.2.5.2 220-mF Super Capacitor

A 220-mF (0.22-F) capacitor is mounted onboard and allows powering the system without any external power. The super cap can be used in the following ways: charging, using (direct connection to 3V3 rail), and disconnected. For more details on these use modes and how to use them, see Section 2.3.

2.3 Power

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

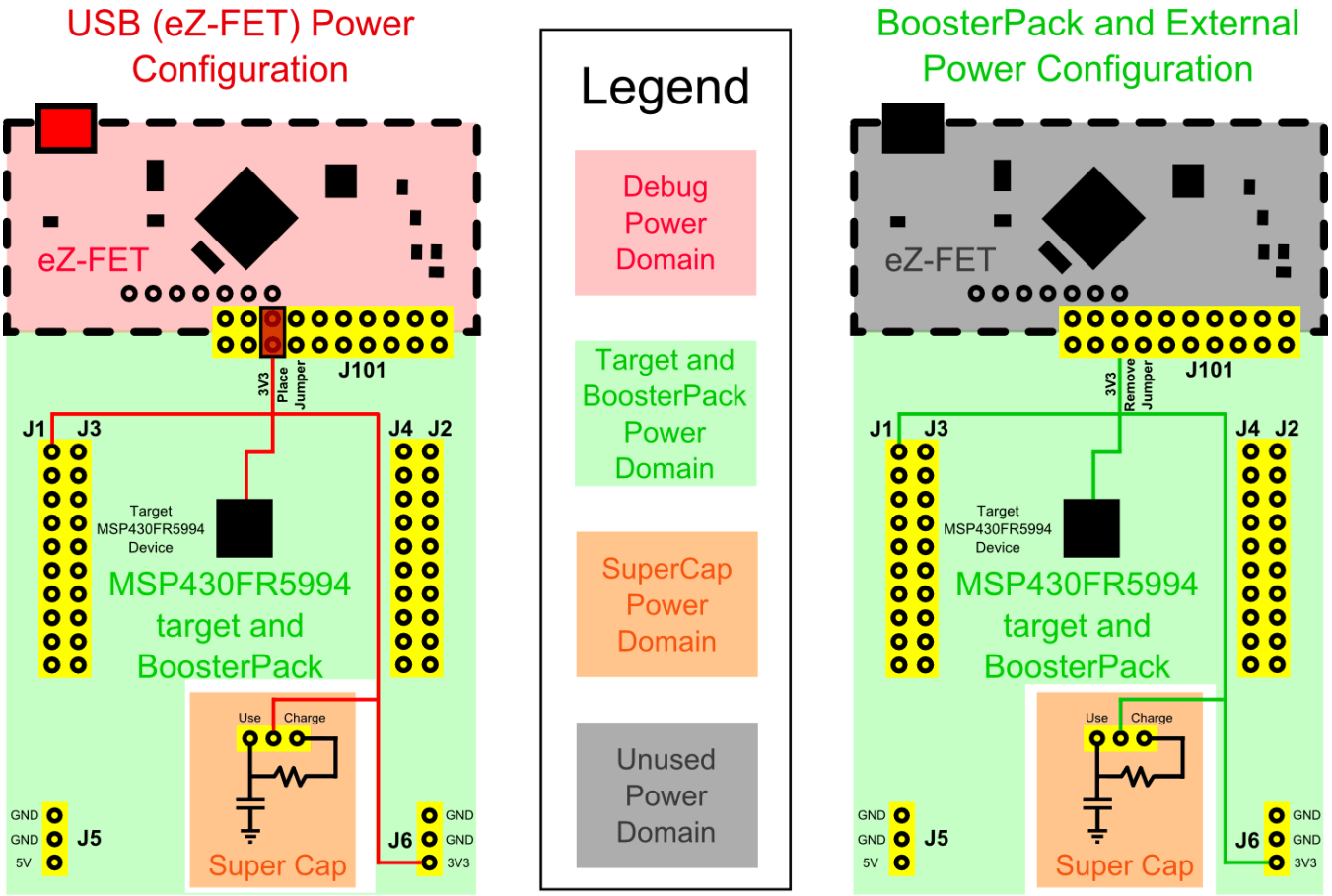

Figure 8. MSP-EXP430FR5994 Power Block Diagram

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 J6 on the board supplies external power directly. Comply with the device voltage operation specifications when supplying external power. The MSP430FR5994 has an operating range of 1.8 V to 3.6 V. More information can be found in the MSP430FR5994 data sheet.

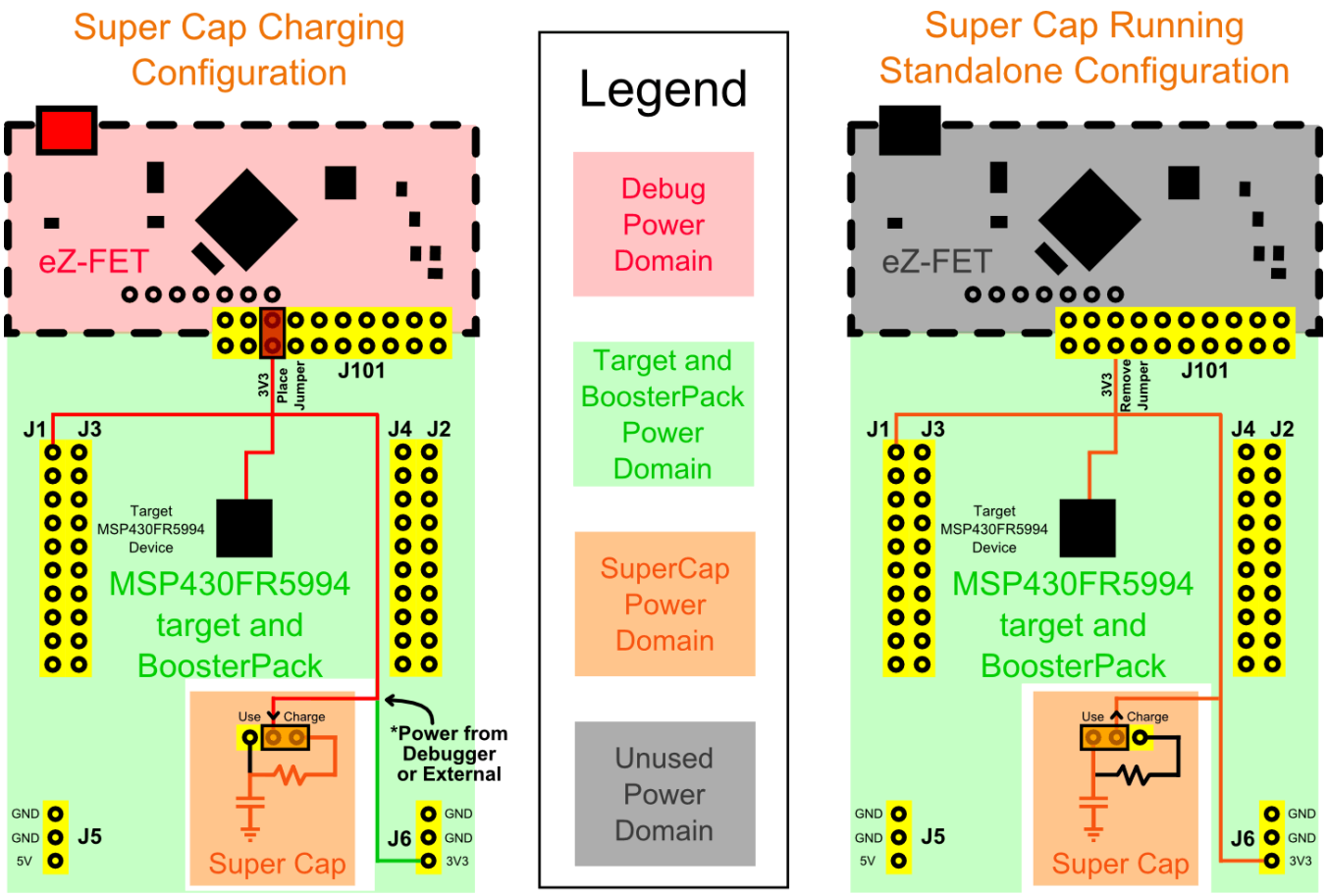

Figure 9. MSP-EXP430FR5994 Super Cap Power Block Diagram

2.3.3 Super Cap (C1)

A 220-mF (0.22-F) super cap is mounted onboard and allows powering the system without any external power. This demonstrates the ultra-low power of the MSP430FR5994 target MCU. See how long you can run your application on the super cap alone!

2.3.3.1 Charging the Super Cap

The super cap can be charged when the EVM is plugged into the PC or when the board is externally powered. During charging, set J8 to the Charge setting, which adds a current limiting resistor for charging.

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

2.3.3.2 Using the Super Cap

After charging of the super cap, move the J8 jumper to the Use setting and unplug power. This connects the super cap to the 3V3 rail without the charging resistor. Now, the LaunchPad development kit is being powered completely by the C1 super capacitor.

For the lowest-power operation, make sure to disconnect the J101 jumpers and remove the microSD card if it is not in use. Removing J101 jumpers prevents the super cap from powering the eZ-FET circuitry and consuming additional power. The microSD card has approximately 100 µA of current draw just being plugged into the system, even when not in use. Taking these steps allows your application to be powered longer from only the super cap.

2.3.3.3 Disabling the Super Cap

The super cap can be completely decoupled from the board by removing the J8 jumper. Hang this jumper off only one pin to prevent losing the jumper.

2.4 Measure MSP430 Current Draw

To measure the current draw of the MSP430FR5994 using a multimeter, use the 3V3 jumper on the J101 jumper isolation block. The current measured includes the target device and any current drawn through the BoosterPack plug-in module headers.

To measure ultra-low power, follow these steps:

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 MSP430FR5994 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 or outputs (I/Os) on the MSP430FR5994. 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.

Alternatively, EnergyTrace++ technology can be used to measure the same current, and see energy profiles through integrated GUI in CCS and IAR. EnergyTrace allows you to compare various current profiles and better optimize the energy performance.

2.5 Clocking

The MSP-EXP430FR5994 provides external clocks in addition to the internal clocks in the device.

- Q1: 32-kHz Epson crystal (FC-135R)
- Q2: DNP high-frequency crystal footprint

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.

The high-frequency crystal is not populated by default, but the footprint for a crystal is provided. Populate a high-frequency crystal for applications that need more precise high-frequency clock sources than the internal DCO.

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

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

For more information about configuring internal clocks and using the external oscillators, see the *MSP430FR58xx, MSP430FR59xx, and MSP430FR6xx 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 derivative devices, not just the onboard MSP430FR5994 target device.

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

Next, make sure the target board has proper connections for SBW. To be compatible with SBW, the capacitor on RST/SBWTDIO cannot be greater than 2.2 nF. The documentation for designing MSP430 JTAG interface circuitry is the *MSP430 Hardware Tools User's Guide*.

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

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

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

2.7 BoosterPack Plug-in Module Pinout

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

The 40-pin standard is compatible with the 20-pin standard that is used by other LaunchPad development kit like the MSP-EXP430FR4133. This allows some subset of functionality of 40-pin BoosterPack plug-in 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-EXP430FR5994 LaunchPad development kit is compatible with all 40-pin BoosterPack plug-in module that comply with the standard. If the reseller or owner of the BoosterPack plug-in module does not explicitly indicate compatibility with the MSP-EXP430FR5994 LaunchPad development kit, compare the schematic of the candidate BoosterPack plug-in module with the LaunchPad development kit to ensure compatibility. Keep in mind that sometimes conflicts can be resolved by changing the MSP430FR5994 MCU pin function configuration in software.

To check the compatibility of your desired BoosterPack plug-in module for your design, with a LaunchPad development kit of your choice, you can use the BoosterPack Checker tool (see Figure 10). This allows you to select any LaunchPad development kit we offer and determine its compatibility with any number of BoosterPack plug-in module that we offer. You can also add your own BoosterPack plug-in module to check its compatibility as you prototype that next design.

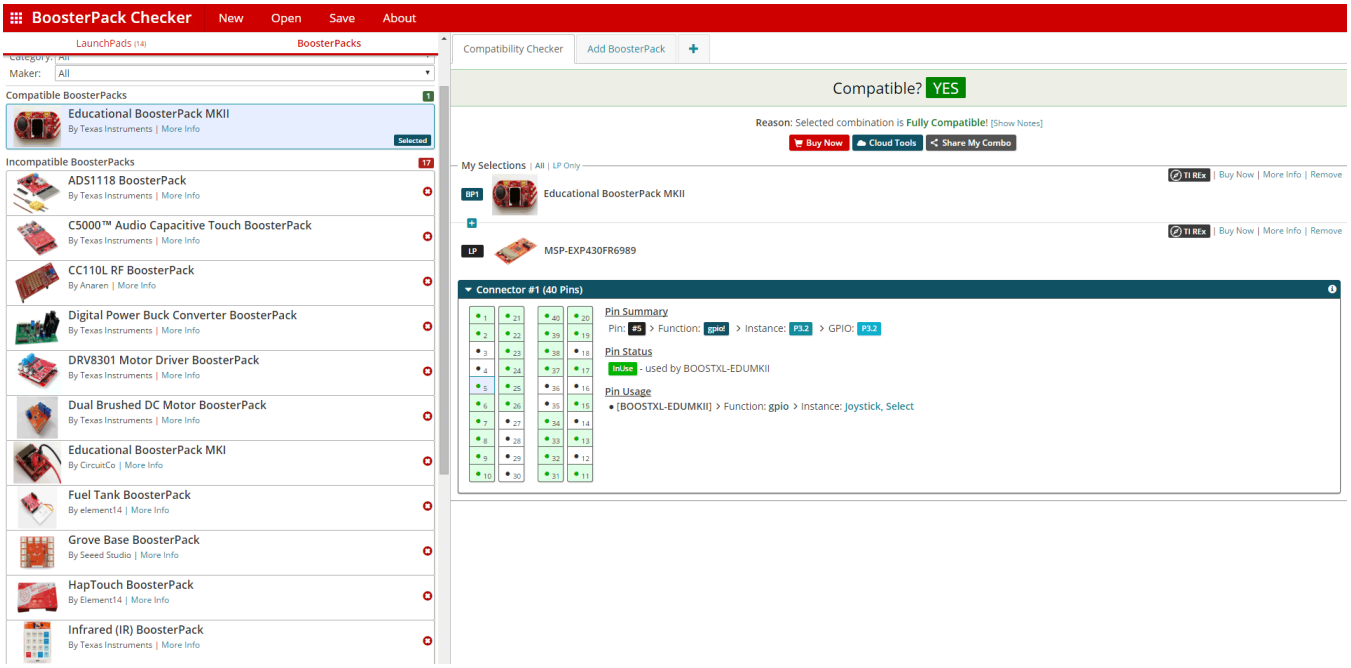

Figure 10. BoosterPack Plug-in Module Checker Tool

Figure 11 shows the 40-pin pinout of the MSP430FR5994 LaunchPad development kit.

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 MSP430FR5994 data sheet for more details on individual pin functions.

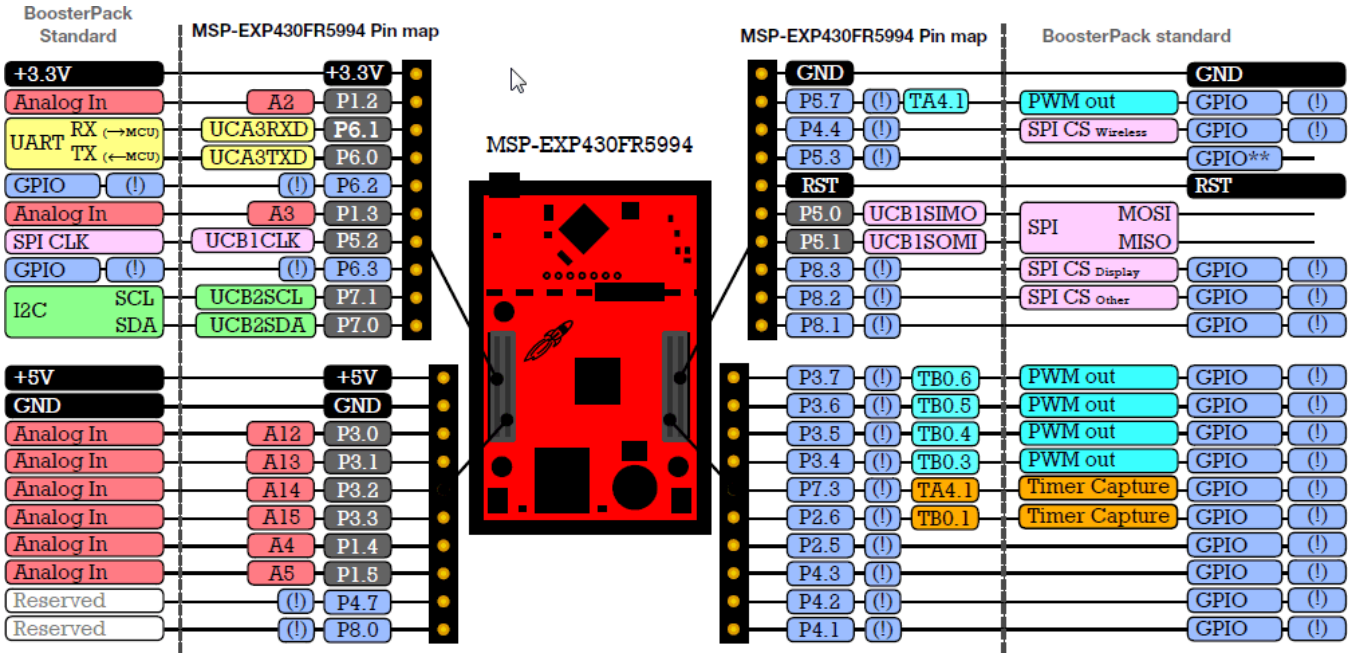

Figure 11. LaunchPad Development Kit to BoosterPack Plug-in Module Connector Pinout

2.8 Design Files

2.8.1 Hardware

See Section 6 for schematics. All design files including schematics, layout, bill of materials (BOM), Gerber files, and documentation are available on the MSP-EXP430FR5994 Design File Download Page.

2.8.2 Software

All design files including TI-TXT object-code firmware images, software example projects, and documentation are available on the MSP-EXP430FR5994 Design File Download Page.

2.9 Hardware Change Log

Table 4 lists the change history for all released hardware revisions.

Table 4. Hardware Change Log

PCB Revision	Description
Rev 1.1	Initial Release

3 Software Examples

Software examples are included with the MSP430FR5994 LaunchPad development kit (see Table 5), and can be found in the MSP430FR5994 LaunchPad development kit Download Page and are also available in MSP430Ware for MSP Microcontrollers.

Table 5. Software Examples

Demo Name	BoosterPack Required	Description	More Details
OutOfBox_FR5994	None	The out-of-box demo pre-programmed on the LaunchPad from the factory. Demonstrates features of MSP430FR5994 MCU.	Section 3.1
BlinkLED_FR5994	None	Blinks an LED on the LaunchPad at a fixed interval.	Section 3.2
BOOSTXL-AUDIO_ RecordPlayback_ MSP430FR5994	• MSP-EXP430FR5994 • BOOSTXL-AUDIO	Demonstrates how to record and playback audio from FRAM memory using DMA.	Section 3.3
BOOSTXL-AUDIO_ LEA_MSP430FR5994	• MSP-EXP430FR5994 • BOOSTXL-AUDIO • 430BOOST-SHARP96 or BOOSTXL-SHARP128	Demonstrates the performance of the MSP low-energy accelerator (LEA) by performing FFT and FIR.	Section 3.4
EEPROM Emulation reference design	• MSP-EXP430FR5994	Emulates EEPROM using FRAM technology on MSP supporting both I^2C and SPI	Section 3.5

To use any of the software examples with the LaunchPad kit, you must have an integrated development environment (IDE) that supports the MSP430FR5994 MCU. Table 6 lists the minimum requirements for IDEs.

Table 6. IDE Minimum Requirements for MSP-EXP430FR5994

Code Composer Studio™ IDE	IAR Embedded Workbench® IDE
Code Composer Studio IDE v6.1.3 or later	IAR Embedded Workbench for MSP430 v6.40.2 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.

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.

3.1.1 Source File Structure

The project is organized in multiple files. This organization makes it easier to navigate and to reuse parts of it for other projects. Table 7 describes each file in the project

Table 7. Source File and Folders

Name	Description
main.c	The out-of-box demo main function, initializations, shared ISRs, and more
LiveTempMode.c	Contains functions for the live temperature streaming mode
FRAMLogMode.c	Contains functions for the FRAM data logging mode
SDCardLogMode.c	Contains function for the SD card data logging mode
Library: Driverlib	Device driver library
Library: FatFs	Generic FAT file system module for small embedded systems (FatFs)
HAL/HAL_SDCard.c	Hardware abstraction layer for board/device to SD card connection
Library: SDCardLib	Wrapper library to provide higher-level SD card APIs

3.1.2 Out-of-Box Demo GUI

The out-of-box demo GUI (see Figure 12) is required to control the out-of-box application running on the MSP-EXP430FR5994 LaunchPad development kit. The GUI can be found in the latest MSPWare installation or in MSP-EXP430FR5994_Software_Examples.zip, available on the MSP-EXP430FR5994 Design File Download Page.

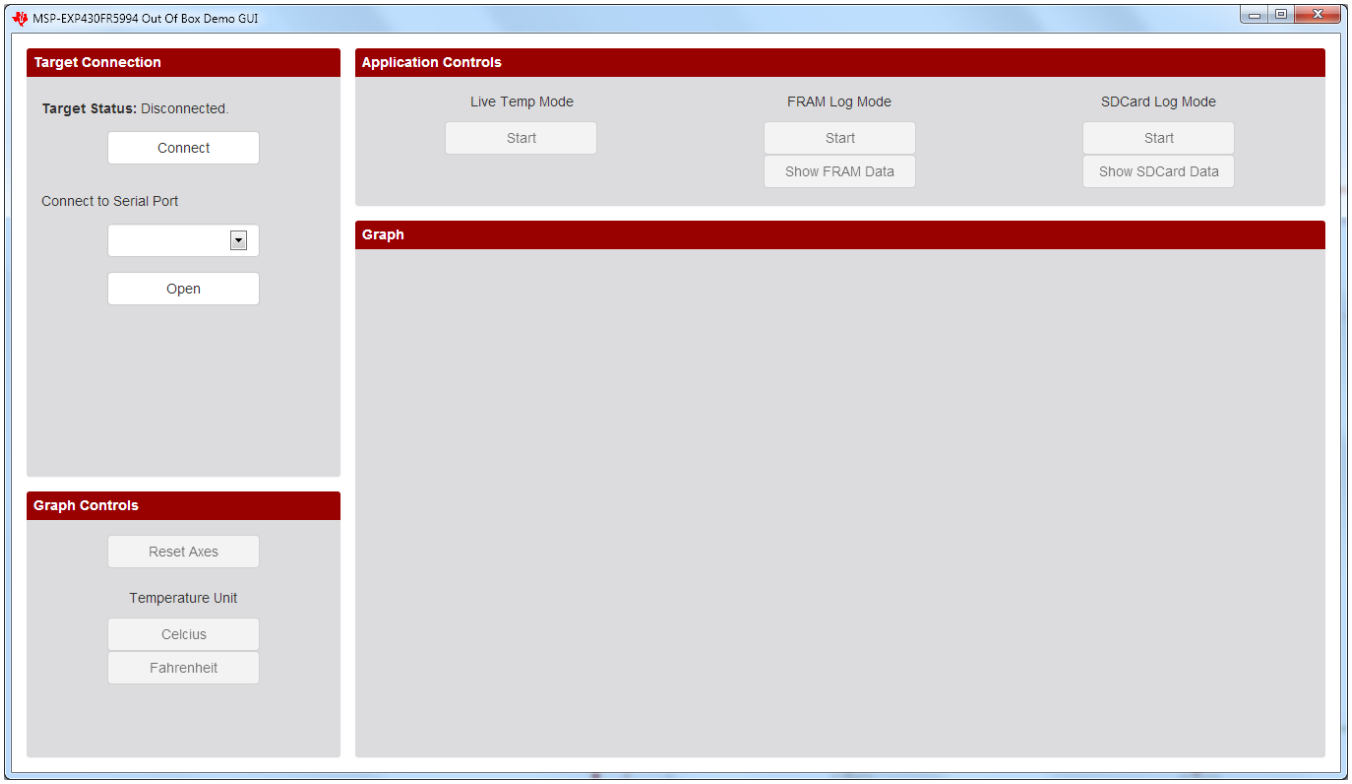

Figure 12. MSP-EXP430FR5994 Out-of-Box Demo GUI

Establish connection to the LaunchPad development kit by first clicking the Connect button, followed by selecting the correct Serial COM Port (MSP Application UART1) and clicking the Open button. On Windows, open Device Manager → Ports (COM & LPT) to verify the corresponding COM port of the backchannel UART.

After connection has been established, the GUI pings the LaunchPad development kit every few seconds to make sure that it is still present and to keep the serial port open. If no response is received from the LaunchPad development kit, the GUI automatically closes the serial port connection.

3.1.3 Power Up and Idle

When the LaunchPad development kit powers up after being connected to a computer, the red and green LEDs toggle several times to indicate that the out-of-box demo is running. The MSP430FR5994 then enters low-power mode 3 to wait for UART commands from the PC GUI.

3.1.4 Live Temperature Mode

To enter the live temperature mode, click the Start button below Live Temp Mode in the GUI Application Controls panel (see Figure 13).

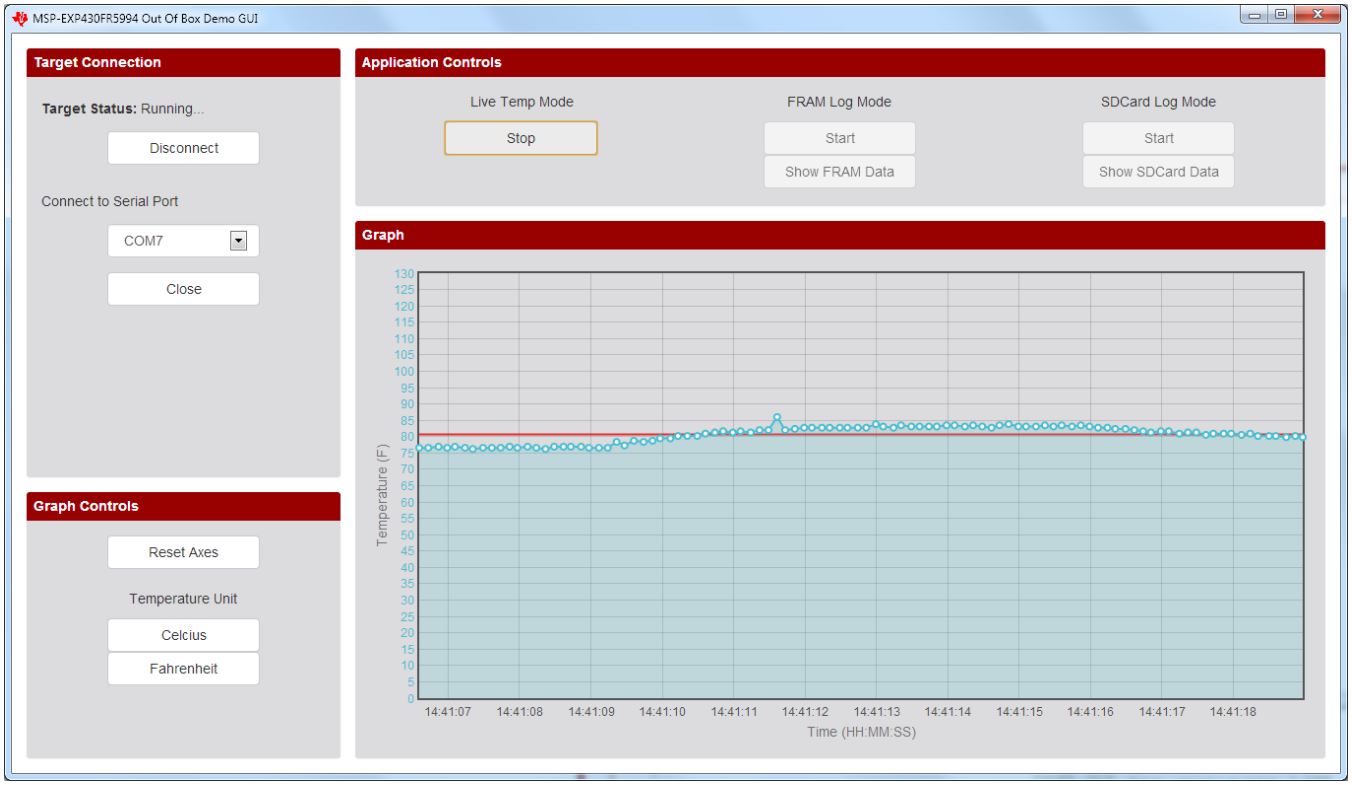

Figure 13. Live Temperature Mode

The MSP430FR5994 first sends two ADC calibration constants for the temperature sensor to the PC GUI. It then sets up its 12-bit ADC for sampling and converting the signals from its internal temperature sensor. A hardware timer is also configured to trigger the ADC conversion every 0.125 seconds before the device enters low-power mode 3 to conserve power. As soon as the ADC sample and conversion is complete, the raw ADC data is sent the through the UART backchannel to the PC GUI.

As the raw ADC data is received by the PC GUI, Celsius and Fahrenheit units are calculated first. The PC GUI keeps a buffer of the most recent 100 temperature measurements, which are graphed against the current time of the PC on the Incoming Data panel.

A red horizontal line is drawn across the data plot to indicate the moving average of the incoming data.

To exit this mode, click the Stop button under Live Temp Mode. You must exit this mode before starting the other modes.

3.1.5 FRAM Log Mode

To enter the FRAM Log Mode, click the Start button under FRAM Log Mode in the GUI Application Controls panel. The PC GUI also sends its current timestamp over UART to be stored in the LaunchPad development kit. This timestamp is later used to extrapolate the X-axis time values when the FRAM logged data are transferred to the GUI (see Figure 14).

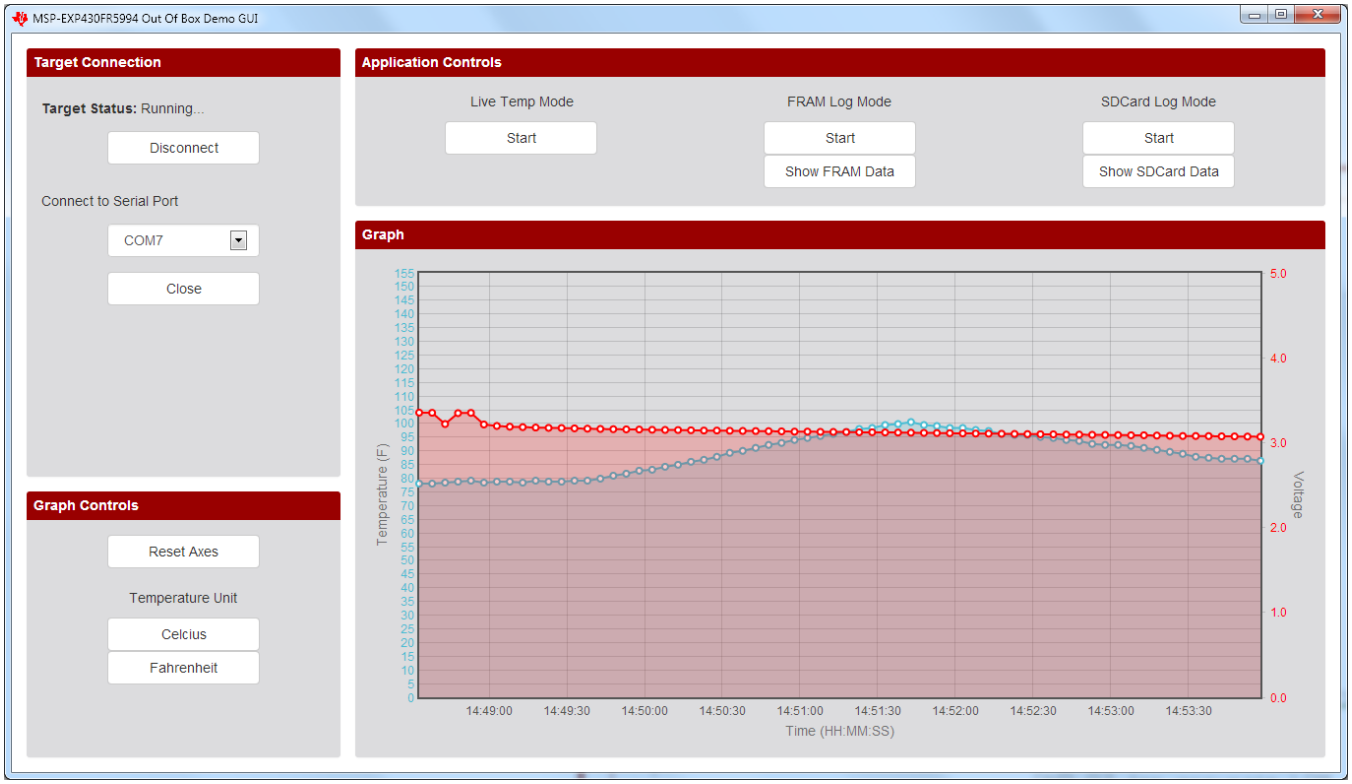

Figure 14. FRAM Log Mode

When the MSP430FR5994 receives the UART command from the GUI, it starts the entry sequence by initializing the Real-Time Clock to trigger an interrupt every 5 seconds. The red LED blinks three times to indicate successful entry into FRAM Log Mode.

The MSP430FR5994 wakes up every 5 seconds from low-power mode 3 to perform data logging before going back to low-power mode 3. The GUI automatically disconnects from the LaunchPad development kit after entering FRAM Log Mode. Each time the device wakes up, the green LED lights up to indicate a data point is stored. Two 10000 long FRAM array buffers are allocated to store the raw ADC output data.

Because the device can be powered solely with the onboard Super Cap, the 12-bit ADC is set up to sample and convert the signals from its internal temperature sensor and battery monitor (super cap voltage).

The board allows powering the application with the USB cable or the onboard super cap. See Section 2.3.3 for more detail on the super cap. To switch to the super cap:

1. While board is powered through USB, configure the jumper to Charge on J8. Wait 2 to 3 minutes.
2. Start FRAM Logging.
3. Switch the jumper on J8 to Use.
4. Disconnect the SBWTDIO and 3v3 jumpers on J101 (to prevent back powering the eZ-FET).
5. Disconnect the USB.

NOTE: Remove the SD card from the holder to reduce power consumption and extend application run time when using the super cap.

To exit the FRAM Log Mode, press the S2 (right) push button on the LaunchPad development kit. The red LED turns on briefly to indicate successful exit and return to the Power up and Idle state. Reattach the jumpers to the default positions and connect USB. Re-open the serial port to the LaunchPad development kit in the GUI. Click the Transfer FRAM Data button to transmit the logged temperature and voltage data from the device FRAM to the PC.

3.1.6 SD Card Log Mode

The SD card mode works similarly to the FRAM Log Mode, except that the temperature and voltage data are stored into .txt files on the SD card. Each time the SD card log mode is started, a new LOG_#.TXT (# increments for the next file) is created under /root/DATA_LOG/.

Enter and exit SD card log mode the same way that you enter and exit FRAM log mode. Click Show SD Card Data to transfer the data from the most recently created LOG_#.TXT to the PC.

NOTE: The super cap cannot power the SD card log mode for long periods of time, because the SD card consumes significantly more power.

3.2 Blink LED Example

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

3.2.1 Source File Structure

The project is split into multiple files (see Table 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 Blink LED main function
Library: Driverlib	Device driver library

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

3.3 BOOSTXL-AUDIO Audio Record and Playback Example

This section describes the functionality and structure of the BOOSTXL-AUDIO_RecordPlayback_MSP430FR5994 demo that is included in the MSP-EXP430FR5994 Software Examples download, or that is more easily accessible through MSPWare (see Section 4.3).

3.3.1 Source File Structure

The project is split into multiple files (see Table 9). This makes it easier to navigate and reuse parts of it for other projects.

Table 9. Source File and Folders

Name	Description
main.c	The demo's clock, GPIO, DAC and interrupt configurations.
application/application.c	Main application loop and interrupt service routines
application/audio_collect.c	Setup, start, stop and shutdown audio collect functions
application/audio_playback.c	Setup, start and stop playback functions and interrupt service routines
application/dac8311.c	Operating modes/functions of the onboard SPI DAC
application/global.h	Global variables definitions
Library: driverlib	Device driver library

3.3.2 Operation

This demo uses the built-in ADC12 on the MSP430FR5994 MCU to sample from the output of the analog microphone on the Audio Signal Processing BoosterPack plug-in module. Using direct memory access (DMA), the 12-bit microphone data is stored and retrieved from FRAM memory. During playback, the microphone data is sent through SPI to the onboard DAC to drive the audio output of the onboard speaker or headphones.

To begin recording an audio sample, press switch S1 on the MSP-EXP430FR5994 (see Figure 15). LED1 turns on while audio is being recorded and turns off when the recording phase is complete. Headphones with an inline microphone can be used to record audio. The BoosterPack plug-in module automatically detects the inline microphone when the headphones are plugged into the provided jack (J6) and records from it instead of the onboard microphone.

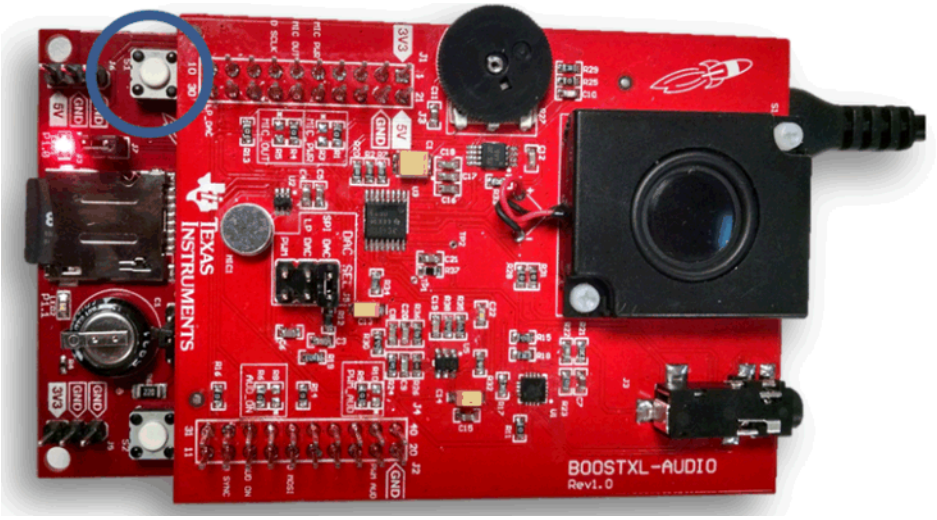

Figure 15. Record

To play back the recorded audio sample, press switch S2 on the MSP-EXP430FR5994 LaunchPad development kit (see Figure 16). LED2 turns on during playback and turns off when the playback phase is complete. To use headphones to listen to the audio playback, plug headphones into the provided jack J6.

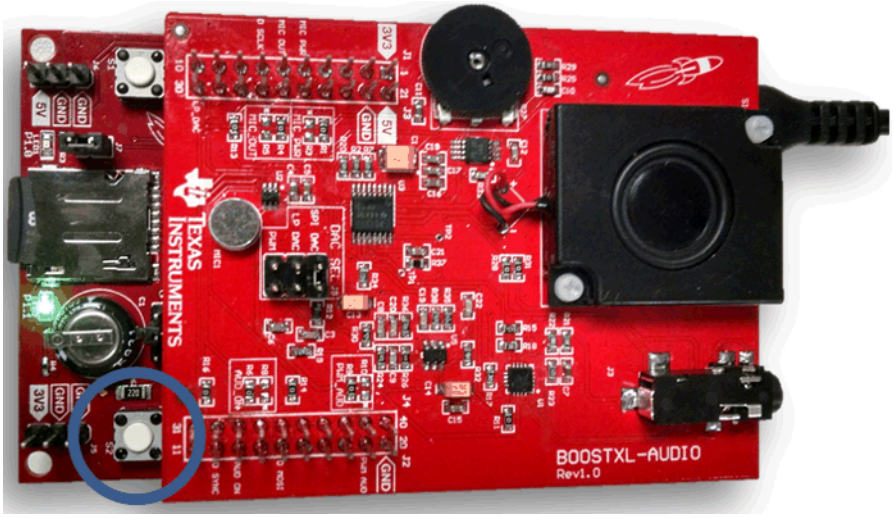

Figure 16. Playback

3.4 Filtering and Signal Processing With LEA Reference Design Example

This section describes the functionality and structure of the Filtering and Signal Processing With LEA reference design. Its software can be downloaded from TIDM-FILTERING-SIGNALPROCESSING-LEA Software.

3.4.1 Source File Structure

The project is split into multiple files (see Table 10). This makes it easier to navigate and reuse parts of it for other projects.

Table 10. Source File and Folders

Name	Description
main.c	The demo's clock, GPIO, display and interrupt configurations.
application/application.c	Main application loop and interrupt service routines
application/audio_collect.c	Setup, start, stop and shutdown audio collect functions
application/audio_playback.c	Setup, start and stop playback functions and interrupt service routines
application/dac8311.c	Operating modes/functions of the onboard SPI DAC
application/global.h	Global variables definitions
application/fir.c	FIR filtering functions
application/FFT.c	Fast Fourier Transform filtering functions
application/FFT_430.asm	MSP430 Fast Fourier Transform filtering functions in assembly
application/benchmark.c	Performance benchmark timer and interrupt service routines
application/fir_coefficient	FIR coefficient definitions
Library: DSPLib	MSP430 DSP Library
Library: grlib	MSP430 Graphics Library
Library: driverlib	Device driver library

3.4.2 Operation

This demo is a TI reference design that highlights the signal processing capabilities and performance of the MSP430FR5994 MCU and its integrated low-energy accelerator (LEA). This example also uses the 430BOOST-SHARP96 or the BOOSTXL-SHARP128 BoosterPack plug-in modules to display the filtered output of the audio signal and act as a user interface. To use this code example user's must configure the Audio BoosterPack plug-in module to use its alternate microphone power and output pins by moving the 0-ohm resistor on R1 to R3 and R4 to R5 as shown in Figure 17. For more information on this example, visit the reference design page.

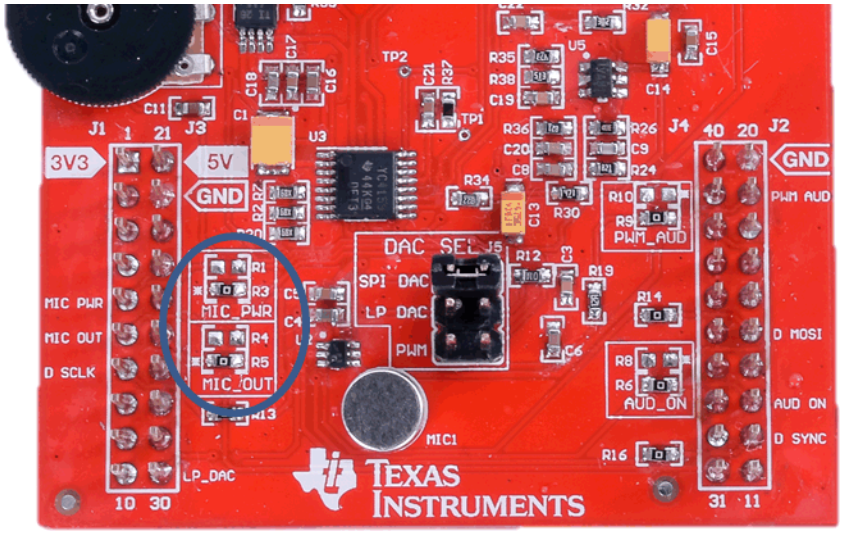

Figure 17. Alternate Microphone Configuration

3.5 Emulating EEPROM Reference Design Example

This section describes the functionality and structure of the Emulating EEPROM reference design. Its software can be downloaded from TIDM-FRAM-EEPROM Software.

3.5.1 Source File Structure

The project is split into multiple files (see Table 11). This makes it easier to navigate and reuse parts of it for other projects.

Table 11. Source File and Folders

Name	Description
main.c	The demo's clock, GPIO, EEPROM initialization and interrupt configurations.
eeprom_interface/eeprom_i2c.c	EEPROM I^2C interface initialization and functions
eeprom_interface/eeprom_spi.c	EEPROM SPI interface initialization and functions
eeprom_definitions.h	Global variables definitions
eeprom.c	EEPROM standard functions
sensing_proc.c	Functions for sampling temperature and voltage
Library: driverlib	Device driver library

3.5.2 Operation

The EEPROM emulation is configured to use I²C or SPI protocol in Slave mode as indicated by Figure 18 and Figure 19. It would typically be connected to a host processor which would act as the master. This implementation, unlike traditional EEPROM, requires no caching after several hundred bytes. The host could continuously write data to memory once the communication is initiated. And the data is immediately written to memory. This means that the application could continuously stream data with much higher throughput. The SPI operation also includes DMA.

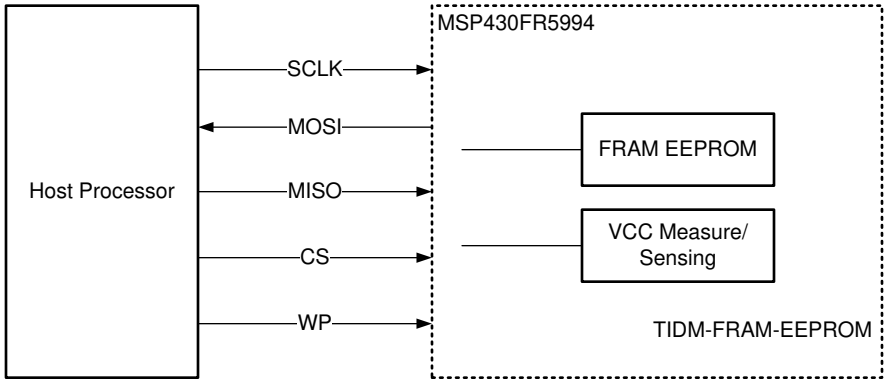

Figure 18. EEPROM SPI Interface Block Diagram

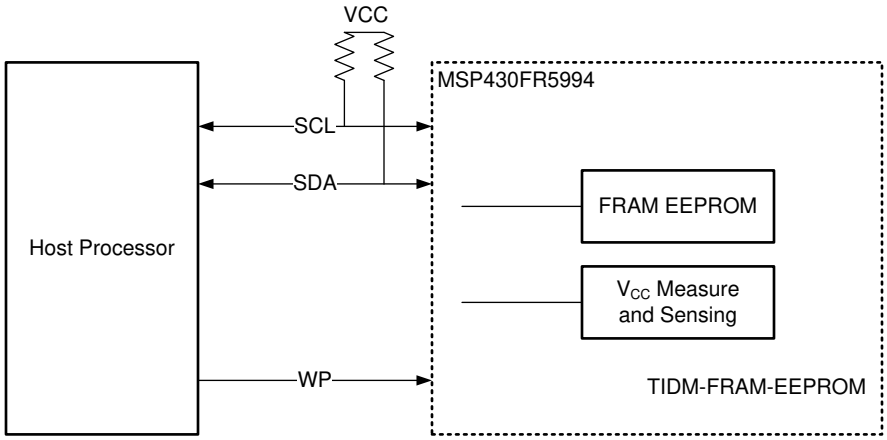

Figure 19. EEPROM I²C Interface Block Diagram

This reference design also emulates industry standard EEPROM protocols such as I²C and SPI, as well as a write protection pin to ensure that the device is protected from any writes. On top of EEPROM emulation, the reference design periodically samples the ADC for the latest VCC and temperature and stores it in FRAM at a low priority. When the host application requests the data, it is immediately available. The sensor data is currently configured to periodically sample every second and can be custom tailored for the application. The sensor reading does not block the EEPROM emulation. The EEPROM emulation is the highest priority function. For more information on this example, visit the reference design page.

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 and IAR Embedded Workbench IDE.

4.1.1 TI Cloud Development Tools

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

4.1.1.1 TI Resource Explorer Cloud

TI Resource Explorer Cloud provides a web interface for browsing examples, libraries and documentation found in MSPWare without having to download files to your local drive (see Figure 20).

Learn more about TI Resource Explorer Cloud now at dev.ti.com.

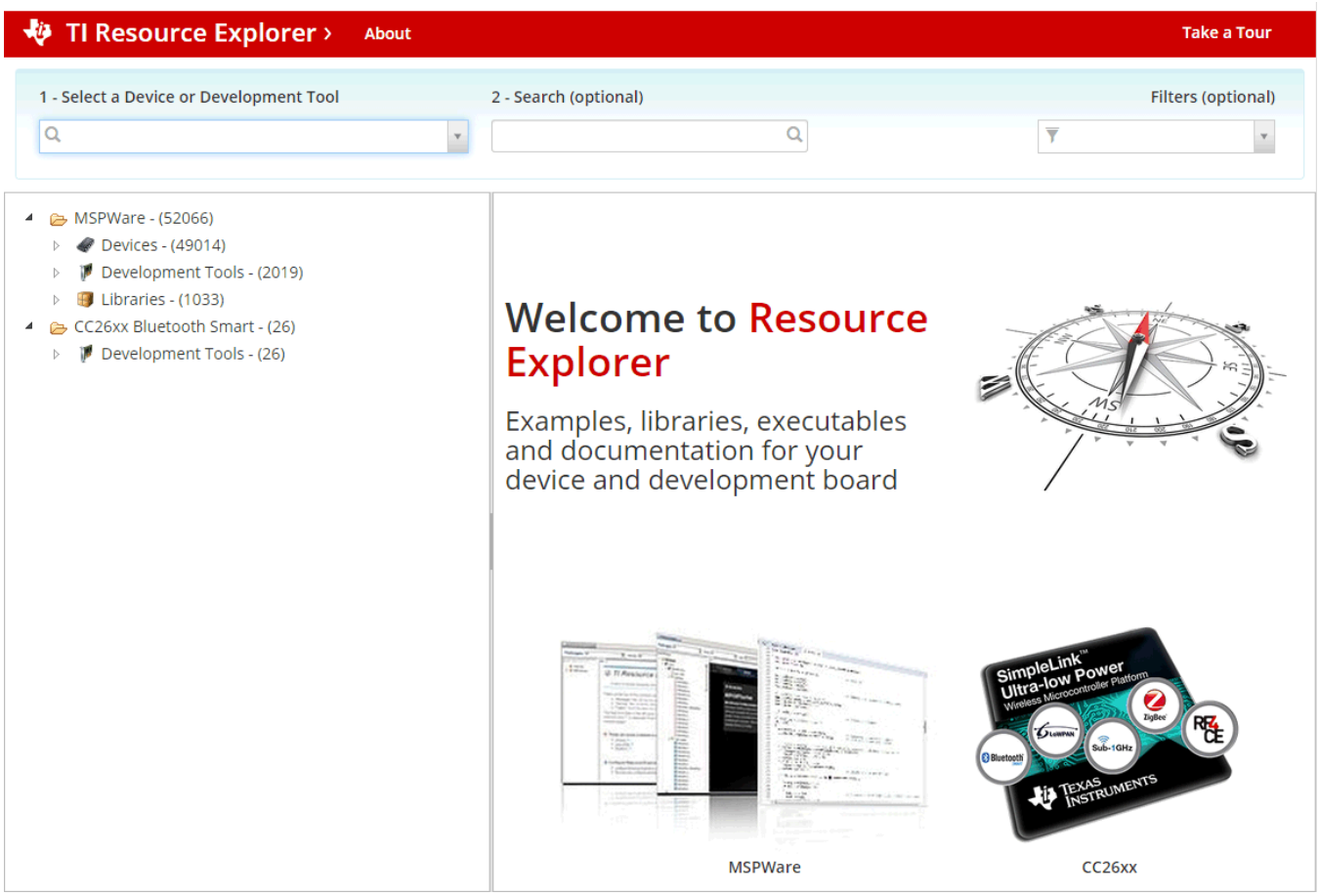

Figure 20. TI Resource Explorer Cloud

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 (see Figure 21). 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.

A full comparison between CCS Cloud and CCS Desktop is available here.

Learn more about Code Composer Studio Cloud now at dev.ti.com.

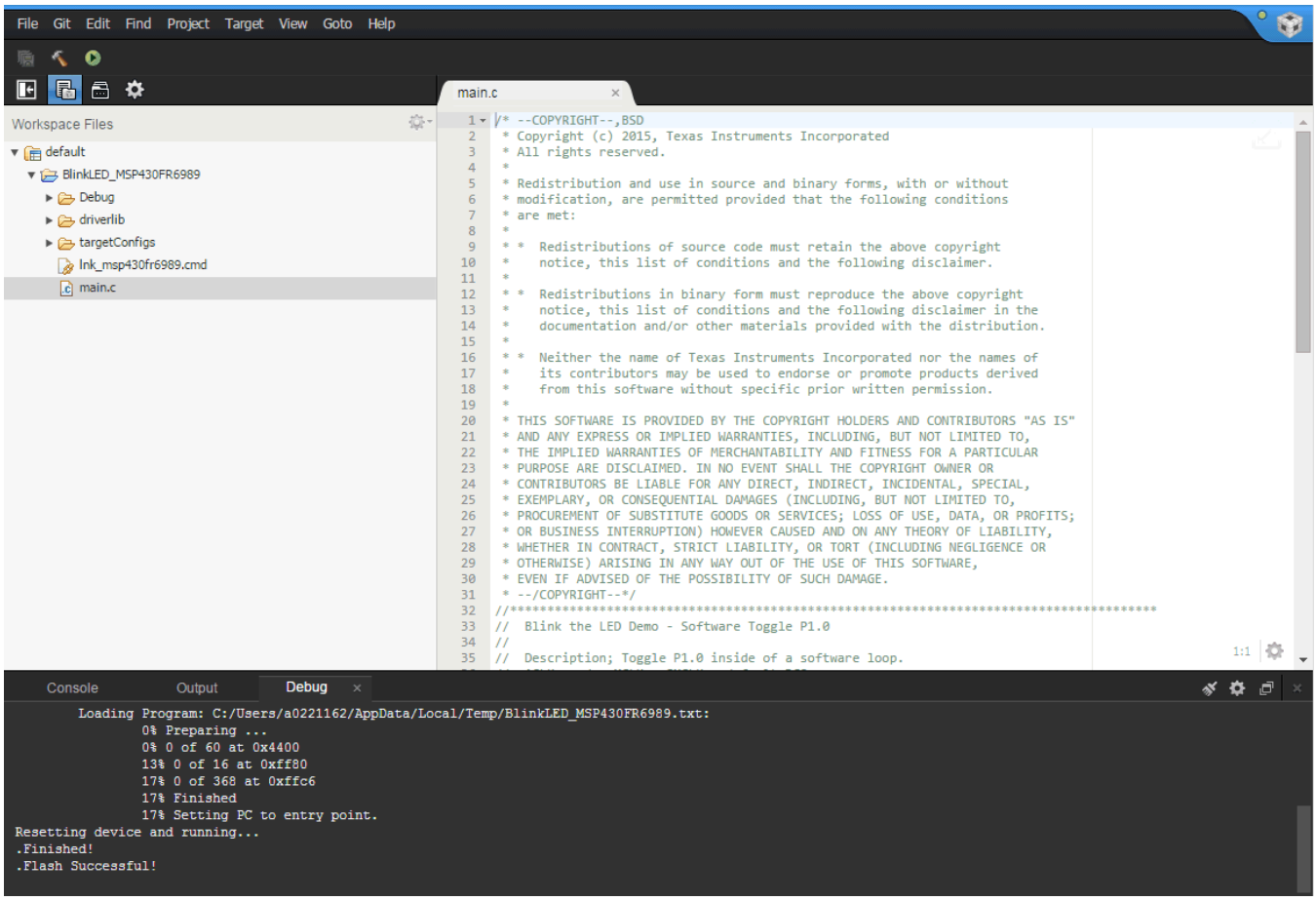

Figure 21. CCS Cloud

4.1.2 Code Composer Studio™ IDE

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

Learn more about CCS and download it from the Code Composer Studio (CCS) Integrated Development Environment (IDE) page.

CCS v6.1.3 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 22).

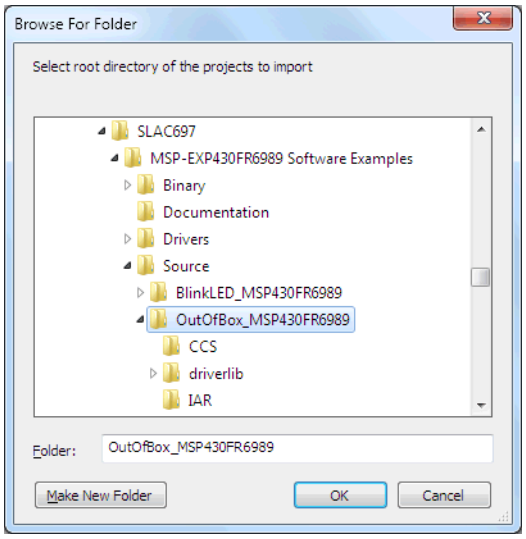

Figure 22. Directing the Project>Import Function to the Demo Project

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 19, and it has a checkmark to the left of it.

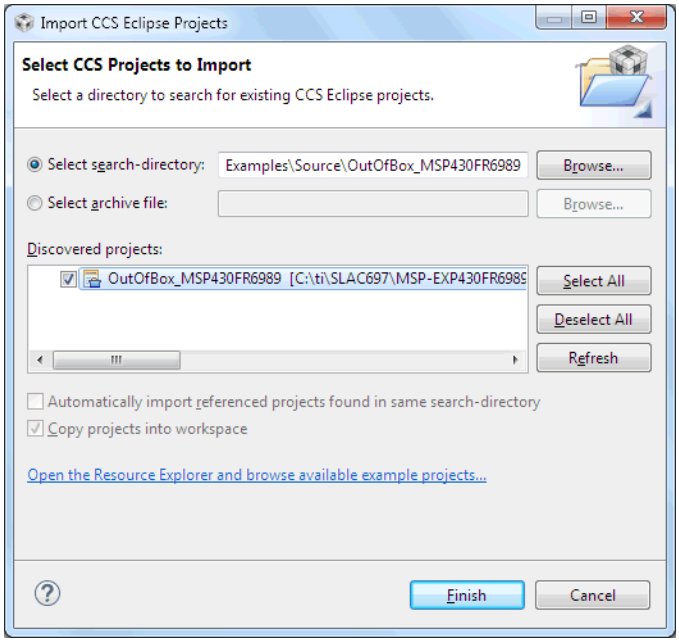

Figure 23. When CCS Has Found the Project

Sometimes CCS finds the project but does not show a checkmark. This might mean that the workspace already has a project by that name. Resolve this conflict by renaming or deleting that project. Even if you do not see it in the CCS workspace, check the workspace directory on the file system.

4.1.3 IAR Embedded Workbench for 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 http://supp.iar.com/Download/SW/?item=EW430-EVAL.

IAR 6.30 or higher is required. To open the demo in IAR, click File>Open>Workspace…, and browse to the *.eww workspace file inside the \IAR subdirectory of the desired demo. All workspace information is contained within this file.

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

Although the software examples have all of the code required to run them, IAR users may download and install MSPWare, which contains MSP430 libraries and the TI Resource Explorer. By default, these are already included in a CCS installation.

4.2 LaunchPad Websites

For more information about the LaunchPad development kit, supported BoosterPack plug-in modules, and available resources, visit:

- MSP-EXP430FR5994 tool folder: Resources specific to this particular LaunchPad development kit
- TI LaunchPad portal: Information about all LaunchPad 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 (see Figure 24). TI Resource Explorer helps you quickly find what you need inside packages including MSPWare, ControlSuite, TivaWare, and more. TI Resource Explorer is well organized to find everything 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 it is tightly integrated with CCS Cloud. See Section 4.1.1 for more information.

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

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

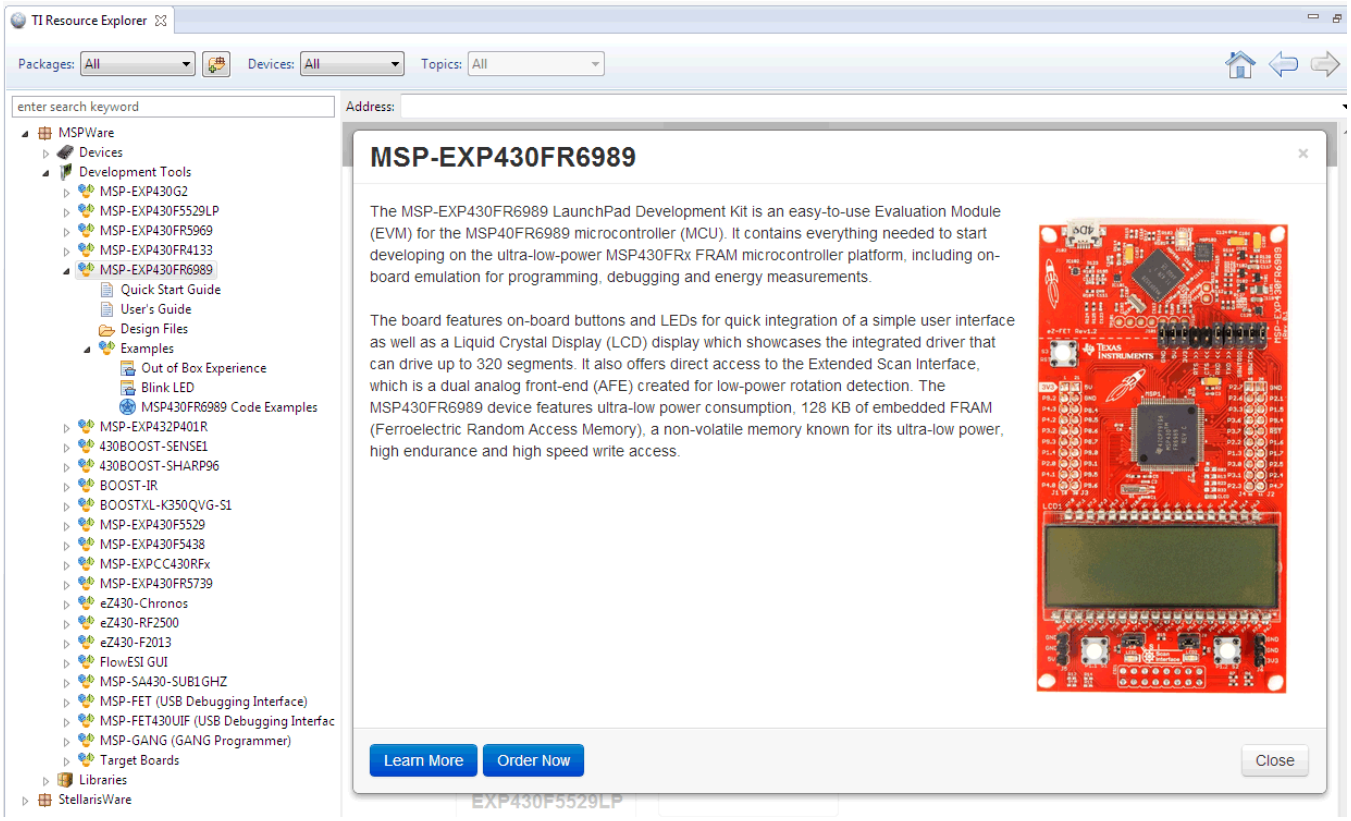

Figure 24. Using TI Resource Explorer to Browse MSP-EXP430FR5994 in MSPWare

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

4.4 FRAM Utilities

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

4.4.1 Compute Through Power Loss (CTPL)

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

4.5 MSP430FR5994 MCU

4.5.1 Device Documentation

At some point, you will probably need more information about the MSP430FR5994 MCU. For every MSP device, the documentation is organized as shown in Table 12.

Table 12. How MSP Device Documentation is Organized

Document	For MSP430FR5994	Description
Device family user's guide	*MSP430FR58xx, MSP430FR59xx, and MSP430FR6xx Family User's Guide*	Architectural information about the device, including all modules and peripherals such as clocks, timers, ADC, and so on.
Device-specific data sheet	*MSP430FR599x, MSP430FR596x Mixed-Signal Microcontrollers*	Device-specific information and all parametric information for this device

4.5.2 MSP430FR5994 Code Examples

MSP430FR599x, MSP430FR596x Code Examples is a set of simple C examples that demonstrate how to use the entire set of MSP430 peripherals (including serial communication, ADC12, LCD_C, Timer_A, 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. There are also MSP Driver Library based code examples available in MSPWare.

4.5.3 MSP430 Application Notes and TI Reference Designs

Visit www.ti.com/msp430 for many application notes and reference designs with practical design examples and topics.

4.6 Community Resources

4.6.1 TI E2E™ Support Forums

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

4.6.2 Community at Large

Many online communities focus on the LaunchPad development kits (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 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 had a socket, allowing me change the target device. Why doesn't this LaunchPad 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 kits.

6 Schematics

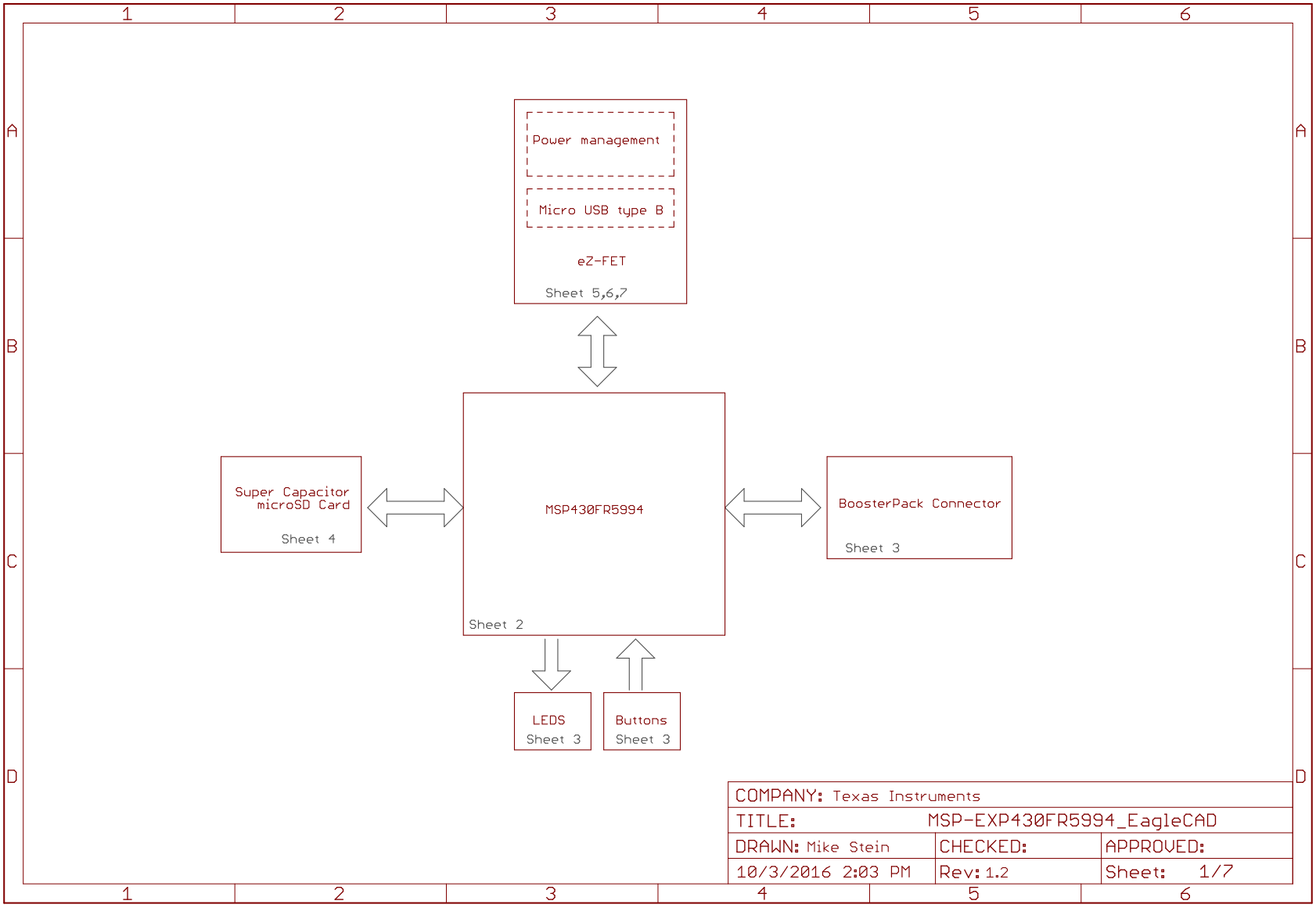

Figure 25. Schematics (1 of 7)

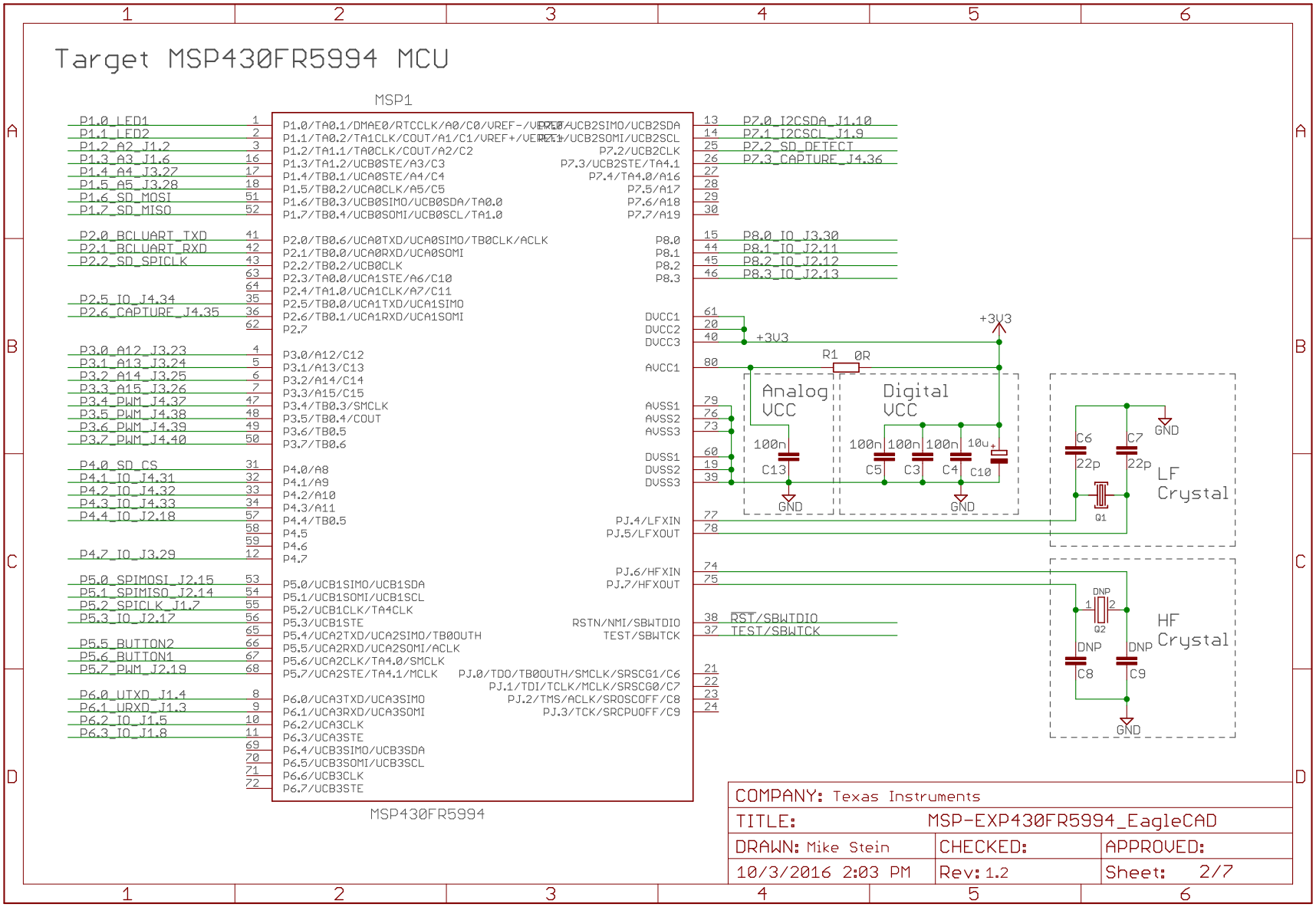

Figure 26. Schematics (2 of 7)

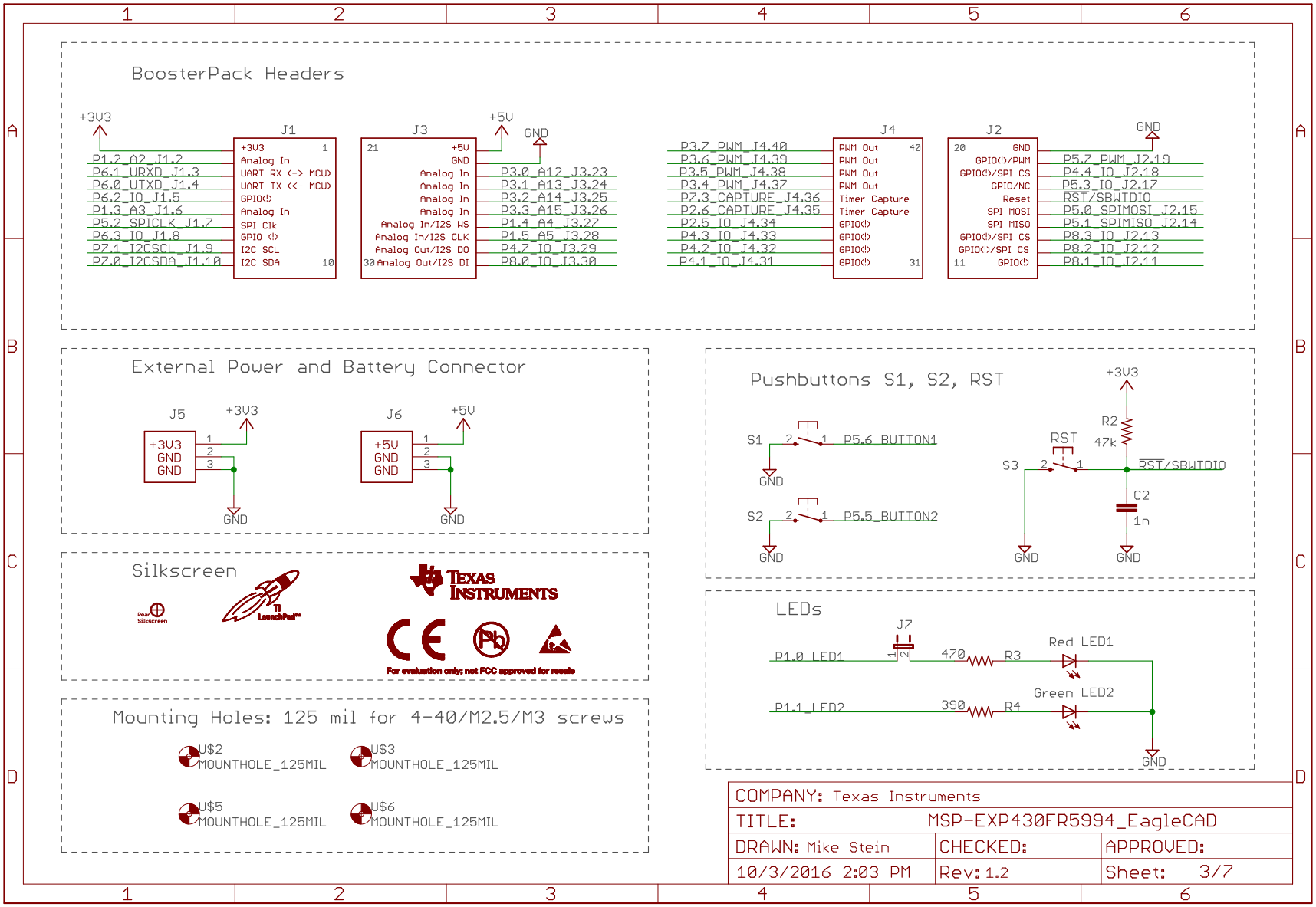

Figure 27. Schematics (3 of 7)

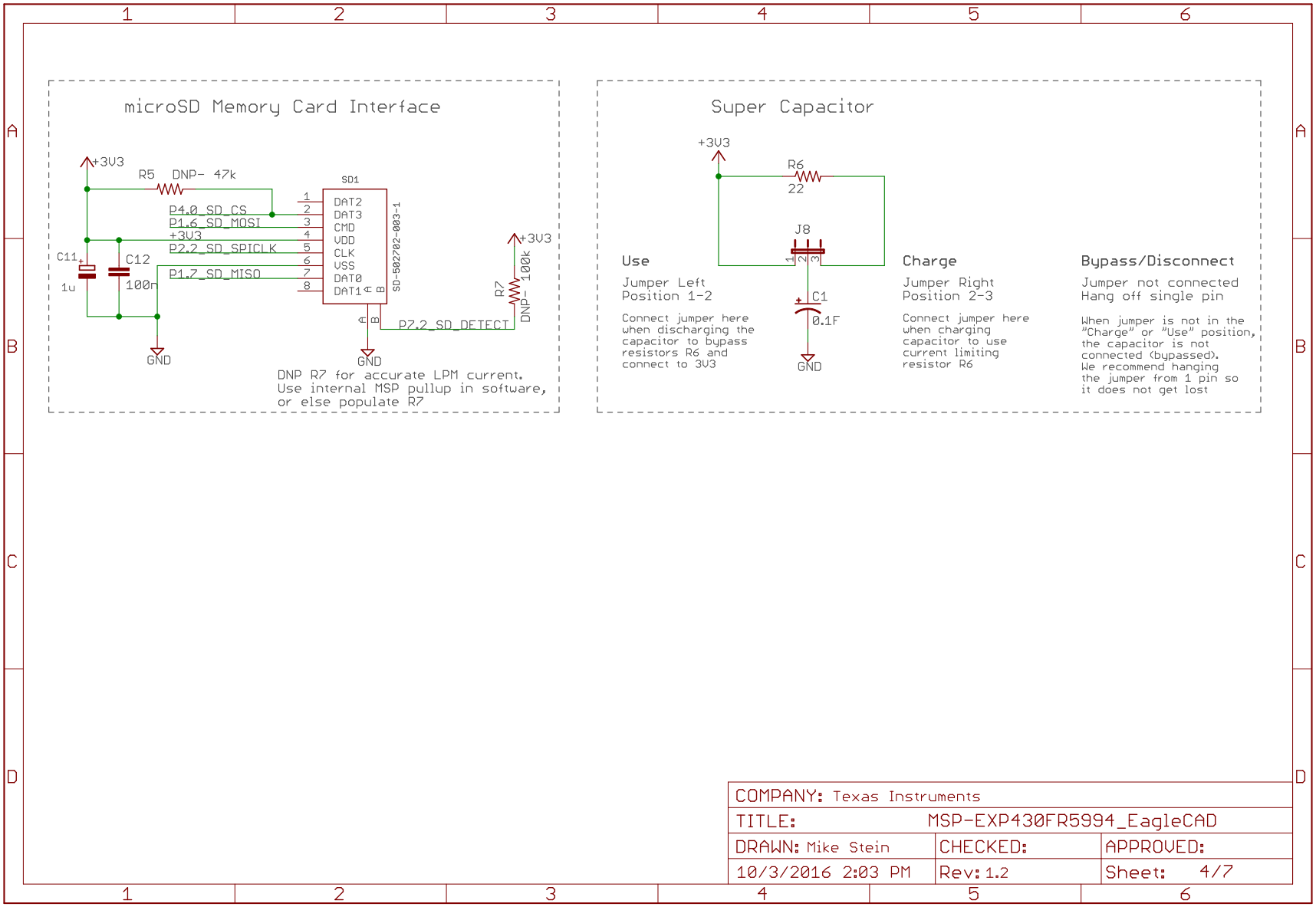

Figure 28. Schematics (4 of 7)

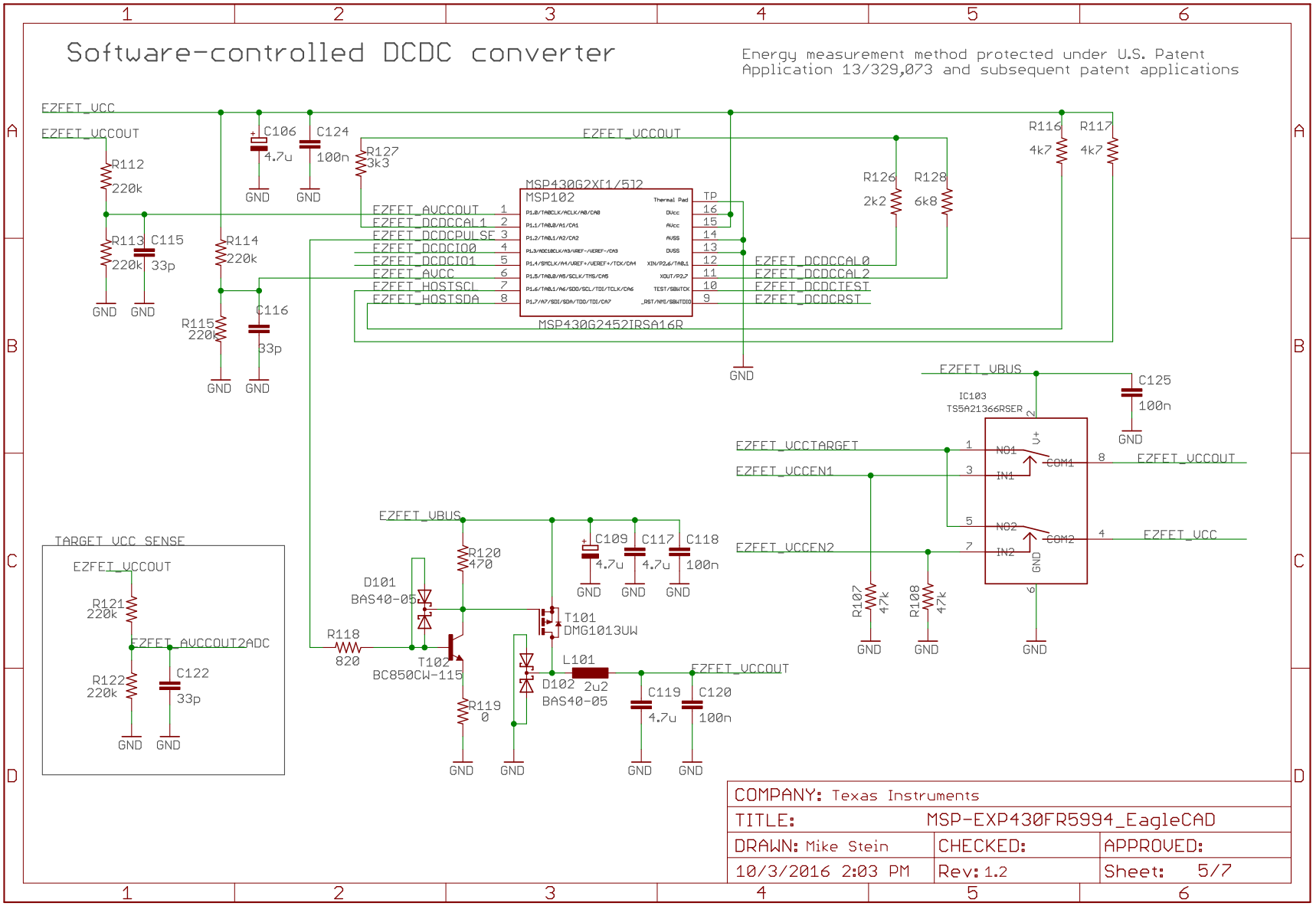

Figure 29. Schematics (5 of 7)

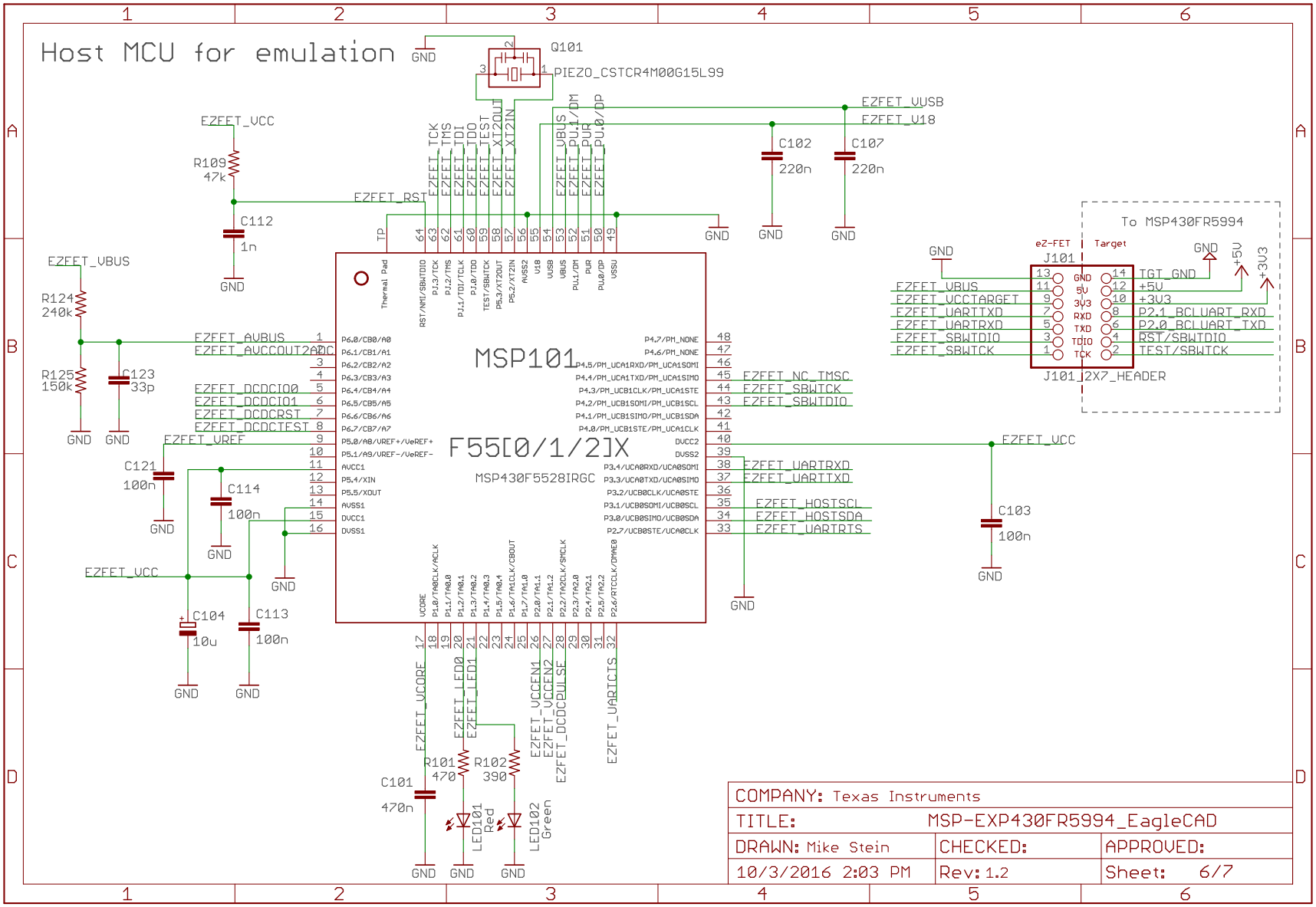

Figure 30. Schematics (6 of 7)

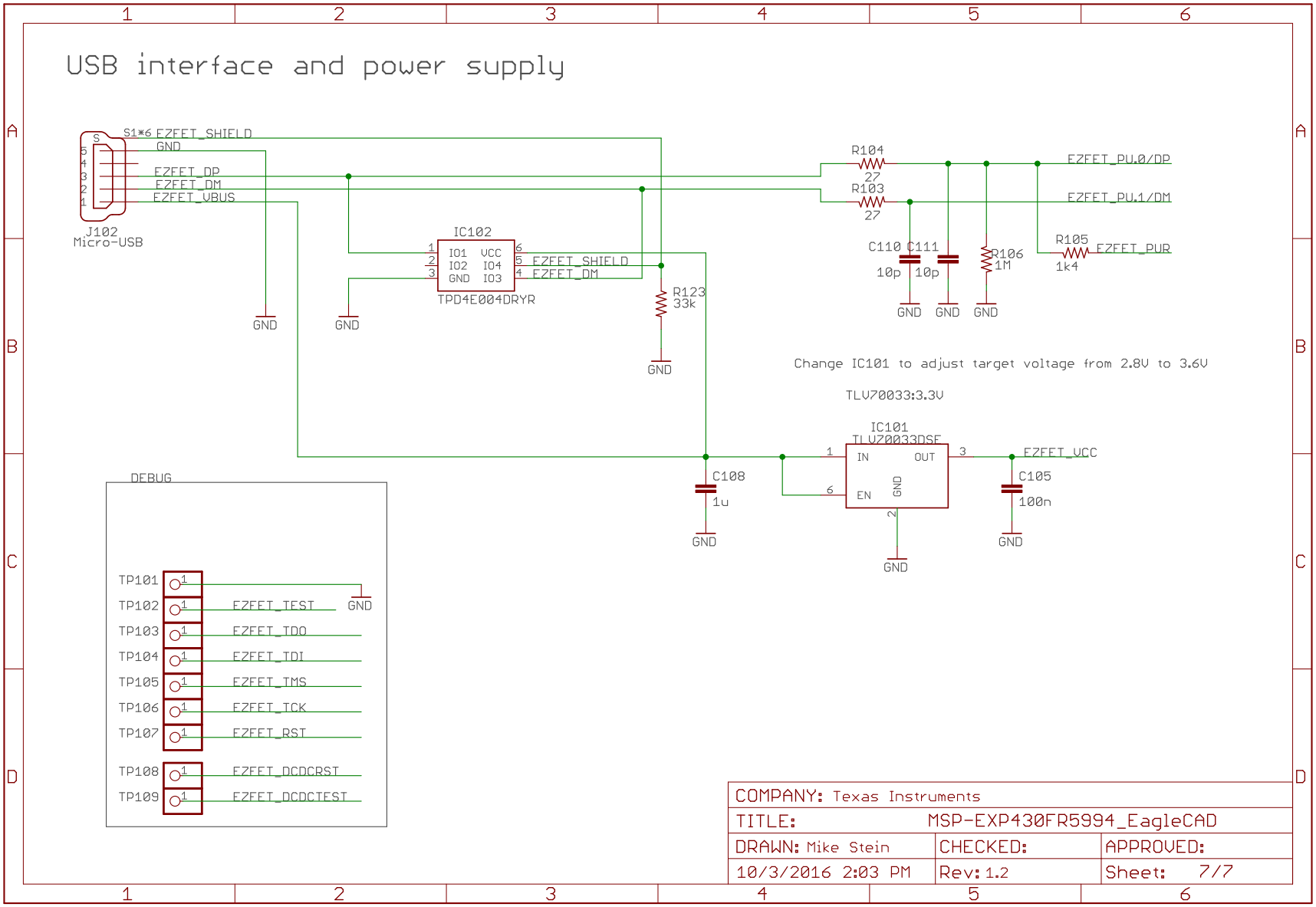

Figure 31. Schematics (7 of 7)

Revision History

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

Changes from April 27, 2016 to September 3, 2019	Page
• Added BOOSTXL-SHARP128 as a supported BoosterPack module in Table 5, *Software Examples*	17
• Changed required IAR Embedded Workbench IDE version to v6.40.2 in Table 6, *IDE Minimum Requirements for MSP-EXP430FR5994*	17
• Added BOOSTXL-SHARP128 as a supported BoosterPack module in Section 3.4.2, *Operation*	24
• Updated all figures in Section 6, *Schematics*	33

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 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