

Table of Contents

1 Getting Started	3
1.1 Introduction	3
1.2 Key Features	3
1.3 What's Included	3
1.4 First Step: Out-of-Box Experience	3
1.5 Next Steps: Looking Into the Provided Code	4
2 Hardware	5
2.1 Jumper Map	6
2.2 Block Diagram	7
2.3 Hardware Features	7
2.4 Power	13
2.5 External Power Supply and BoosterPack Plug-in Module	13
2.6 Measure Current Draw of the MSPM0 MCU	14
2.7 Clocking	14
2.8 BoosterPack Plug-in Module Pinout	14
3 Software Examples	15
4 Resources	16
4.1 Integrated Development Environments	16
4.2 MSPM0 SDK and TI Resource Explorer	16
4.3 MSPM0G3507 MCU	17
4.4 Community Resources	17
5 Schematics	18
6 Revision History	24

Trademarks

LaunchPad™, EnergyTrace™, BoosterPack™, Code Composer Studio™, and TI E2E™ are trademarks of Texas Instruments.

IAR Embedded Workbench™ is a trademark of IAR Systems AB.

Arm®, Cortex®, Keil®, and µVision® are registered trademarks of Arm Limited.

All trademarks are the property of their respective owners.

1 Getting Started

1.1 Introduction

The MSPM0G3507 is Arm® 32-bit Cortex®-M0+ CPU with a memory protection unit and frequency up to 80MHz. The device features 128KB of embedded flash memory combined with 32KB of on-chip RAM. The integrated high-performance analog peripherals include two 12-bit 4-Msps SAR ADCs, two zero-drift chopper op-amps (OPA), and one general-purpose amplifier (GPAMP) to help users design their system.

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

Free software development tools are also available such as TI's [Code Composer Studio™ IDE](#), [IAR Embedded Workbench™ IDE](#), and [Keil®µVision® IDE](#). Code Composer Studio IDE supports EnergyTrace™ technology when paired with the MSPM0G3507 LaunchPad™ development kit. Developer can easily measure the power consumption of their applications. More information about the LaunchPad development kit, the supported BoosterPack™ plug-in modules, and the available resources can be found at [TI's LaunchPad development kit portal](#). To get started quickly and find available resources in the MSPM0 software development kit (SDK), visit the [TI Cloud Developer Zone](#). MSPM0 MCUs are also supported by extensive online collateral, training with MSP Academy, and online support through the [TI E2E™ support forums](#).

See the [MSPM0 Quick Reference Guide](#) for more resources surrounding MSPM0 devices.

1.2 Key Features

- Onboard XDS110 debug probe
- EnergyTrace technology available for ultra-low-power debugging
- 2 buttons, 1 LED and 1 RGB LED for user interaction
- Temperature sensor circuit
- Light sensor circuit
- External OPA2365 (default buffer mode) for ADC (up to 4 Msps) evaluation
- Onboard 32.768kHz and 40MHz crystals
- RC filter for ADC input (unpopulated by default)

1.3 What's Included

1.3.1 Kit Contents

- MSPM0G3507 LaunchPad development kit
- 1 Micro USB cable
- 1 Quick Start Guide

1.3.2 Software Examples

- [Out-of-Box Software Example](#)
- [SDK Examples](#)

1.4 First Step: Out-of-Box Experience

An easy way to get started with the EVM is by using the preprogrammed out-of-box code. This code demonstrates some key features of the EVM and is meant to be used with the LP-MSPM0G3507 out-of-box demo GUI.

1.4.1 Connecting to the Computer

Connect the LaunchPad development kit using the included USB cable to a computer. A green power LED illuminates if power connects. For proper operation, drivers are needed. TI recommends that you get the drivers by installing an IDE such as TI's Code Composer Studio IDE or IAR Embedded Workbench IDE. Standalone drivers are also available.

1.4.2 Running the Out-of-Box Experience

Access the GUI Composer powered GUI for the Out-of-Box Experience (OoBE) through the TI Cloud Gallery. Alternatively, the GUI is provided with the MSPM0 SDK as well for offline operation. [Figure 1-1](#) shows the welcome screen for the GUI. The GUI runs through three main examples:

- Blink LED – Change the rate of LED blinking
- Light Sensor – Read light sensor and plot results, and also modulate RGB brightness
- Thermistor – Read thermistor voltage and plot converted temperature over time. The RGB LED changes colors to signify differences of temperature measured.
- Function Generator – Generate sine, square, and sawtooth waveforms, sample with ADC, and plot on graph.

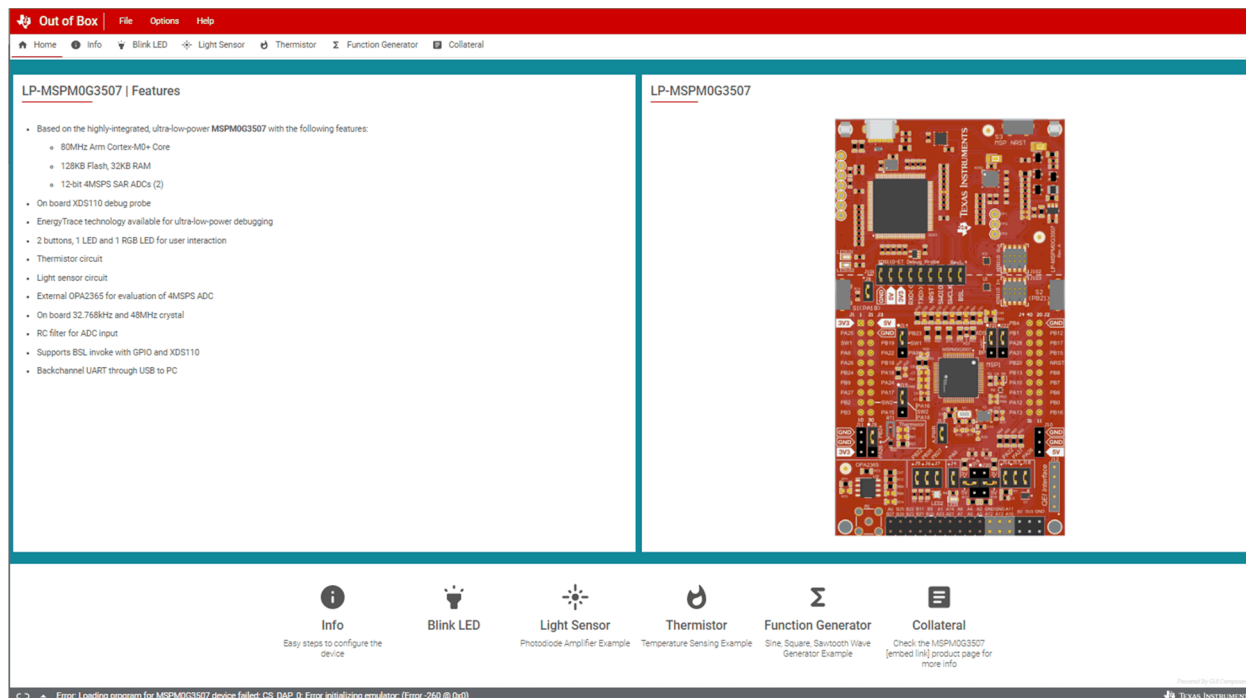


Figure 1-1. Out-of-Box Experience Welcome Screen

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 MSPM0 SDK. Code Composer Studio Cloud is a simple Cloud-based IDE that enables developing and running applications on the LaunchPad development kit. SysConfig for MSPM0 is another graphical tool that can be utilized to easily and quickly setup your MSPM0G3507 device, pins, and peripherals to fit your development needs. 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 1.3.2](#) describes all functions in detail and provides a project structure to help familiarize you with the code. With the XDS110 debug probe, debugging and downloading new code is simple. A connection between the EVM and a PC through the provided USB cable is all that is needed.

2 Hardware

Figure 2-1 shows an overview of MSPM0G3507 LaunchPad development kit hardware.

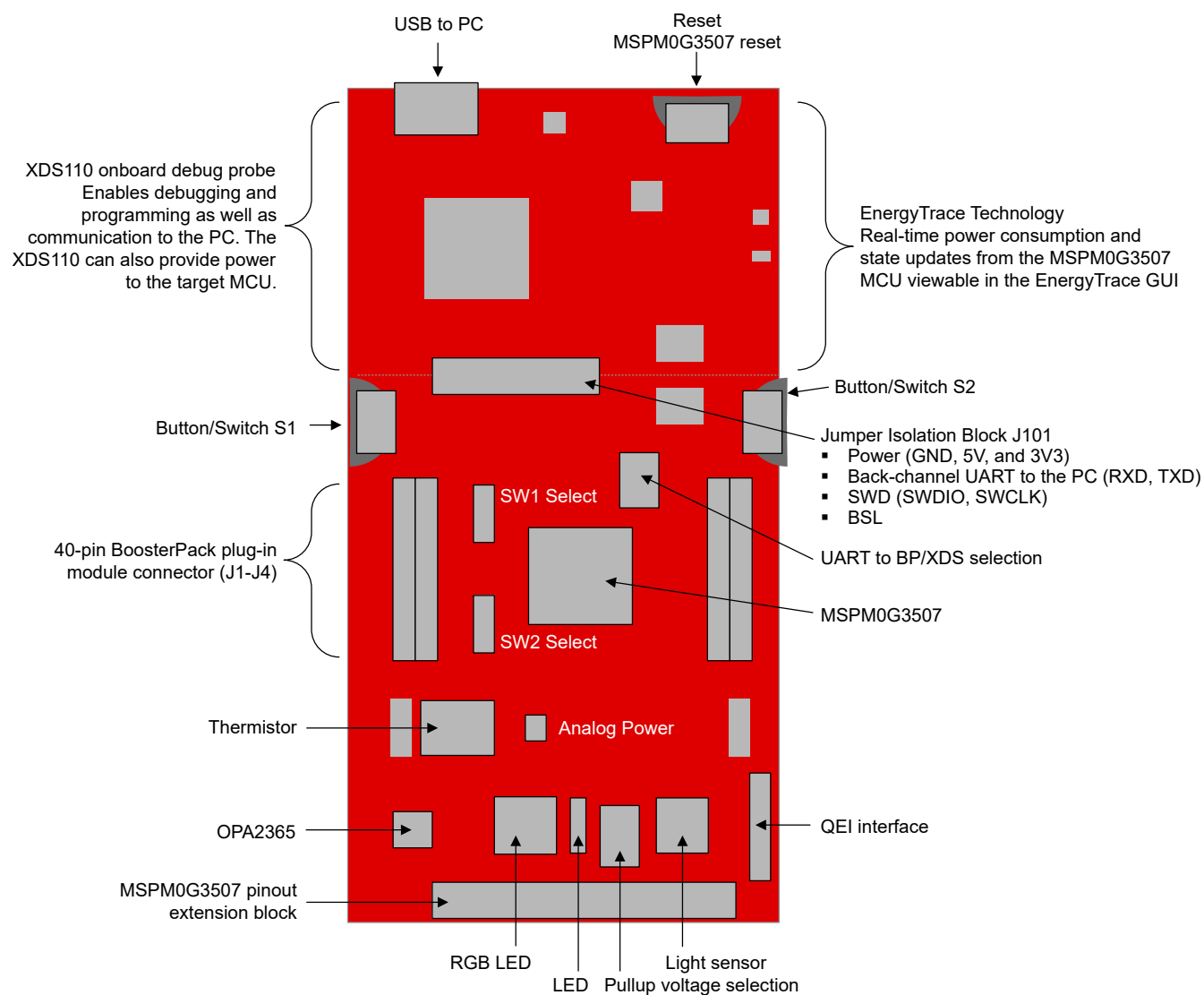


Figure 2-1. MSPM0G3507 LaunchPad Development Kit Hardware

2.1 Jumper Map

Table 2-1 lists the default installation of the jumpers.

Table 2-1. LaunchPad Kit Header Description and Jumper Shunt Installation

Jumper	Description	Default setting	Connected Pin or Signal	Low Power Measurement Recommendation
J101	XDS110-ET isolation block	Installed	GND, 5 V, 3V3, RXD, TXD, NRST, SWDIO, SWCLK, BSL	OFF to disconnect from XDS-110 circuitry
J102	XDS110-ET OUT	NA	GND, 3V3, SWDIO, SWCLK, NRST	No external connection
J103	XDS110-ET IN	NA	GND, 3V3, SWDIO, SWCLK	No external connection
J1/J3	BoosterPack pin header	NA	See schematic for details	No external connection
J2/J4	BoosterPack pin header	NA	See schematic for details	No external connection
J4	Red LED1	Installed	PA0 to LED	OFF to disconnect pin from LED
J5	RGB LED2 – blue channel	Installed	PB22 to RGB LED	OFF to disconnect pin from RGB LED
J6	RGB LED2 – red channel	Installed	PB26 to RGB LED	OFF to disconnect pin from RGB LED
J7	RGB LED2 – green channel	Installed	PB27 to RGB LED	OFF to disconnect pin from RGB LED
J8	S1 Button and BSL invoke	Installed	PA18	OFF to avoid current from external pulldown depending on pin configuration
J9	Thermistor signal selection	(1) short to (2) PB.24	PB24 to thermistor circuit	OFF to disconnect from thermistor circuit
J10	5V power header	NA	5V, GND	No external connection
J11	3V3 power header	NA	3V3, GND	No external connection
J12	QEI interface header	NA	PA29, PA30, PB14, 3V3, GND	No external connections
J13	Analog power – power to thermistor and OPA2365	Installed	3V3	OFF to disconnect power to thermistor, and OPA2365 circuits
J14	SW1 selection to BP header – PA9/PB23	(1) short to (2) PB.23	PB23 to J1.3	Don't care if not used in BoosterPack connector
J15	SW2 selection to BP header – PA16/PA18	(1) short to (2) PA16	PA16 to J3.29	Don't care if not used in BoosterPack connector
J16	OPA0_OUT for light sensor circuit	Installed	PA22 to light sensor	OFF to disconnect from photodiode circuit
J17	OPA0_IN0- for light sensor circuit	Installed	PA27 to light sensor	OFF to disconnect from photodiode circuit
J18	OPA0_IN0+ for light sensor circuit	Installed	PA26 to light sensor	OFF to disconnect from photodiode circuit
J19	PA0 Open-drain IO pullup	(1) Short to (2) 3.3V	PA0 to 3V3	OFF if pin initialized as output low or input with pullup or pulldown
J20	PA1 Open-drain IO pullup	(1) Short to (2) 3.3V	PA1 to 3V3	OFF if pin initialized as output low or input with pullup or pulldown
J21	UART0_TX selection	(1) short to (2) XDS_UART function	PA10 to XDS	OFF to disconnect pin
J22	UART0_RX selection	(1) short to (2) XDS_UART function	PA11 to XDS	OFF to disconnect pin
J23-J28	MSPM0G3507 pin extension header (unlabeled – bottom of board)	NA	See schematic for details	No external connection

2.2 Block Diagram

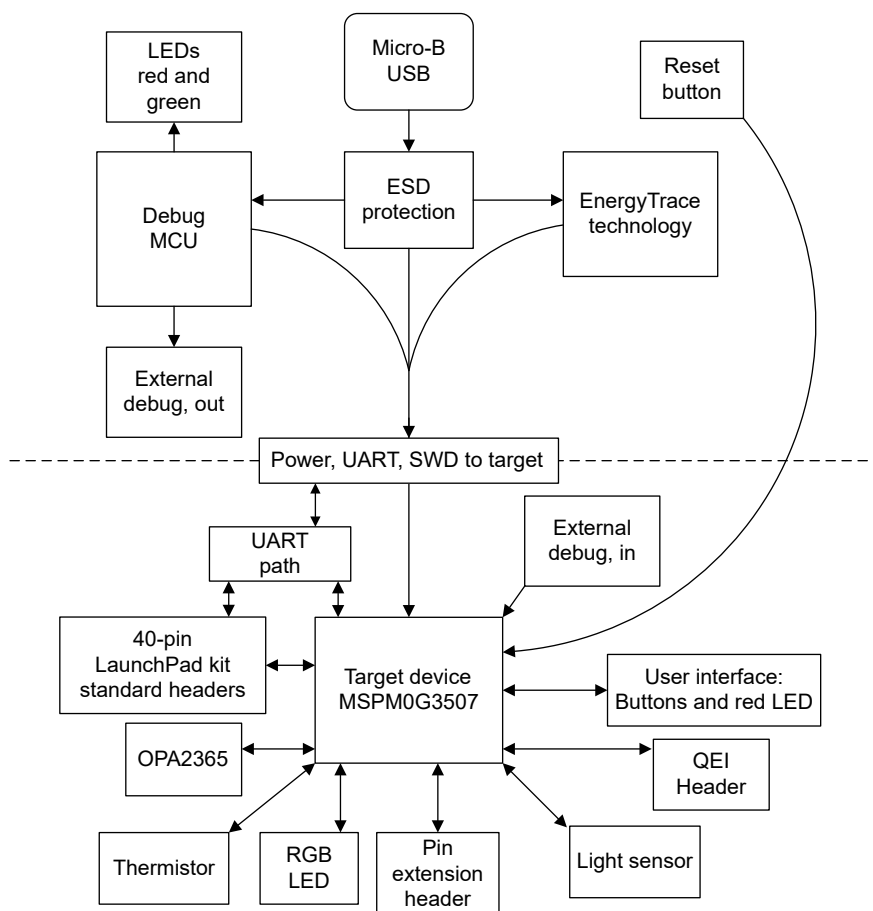


Figure 2-2. Block Diagram

2.3 Hardware Features

2.3.1 MSPM0G3507 MCU

The MSPM0G3507 devices provide 128KB of embedded flash program memory with built-in error correction code (ECC) and 32KB of SRAM with hardware parity. The devices also incorporate a memory protection unit, 7-channel DMA, math accelerator, and a variety of high-performance analog peripherals such as two 12-bit 4-Msps ADCs, a configurable internal shared voltage reference, one 12-bit DAC, three high speed comparators with built-in reference DACs, two zero-drift op-amps with programmable gain, and one general-purpose amplifier. The devices also offer intelligent digital peripherals such as two 16-bit advanced control timers, three 16-bit general purpose timers, one 32-bit high resolution timer, two windowed-watchdog timers, and one RTC with alarm and calendar mode. These devices provide data integrity and encryption peripherals (AES, CRC, TRNG) and enhanced communication interfaces (four UART, two I2C, two SPI, and one CAN 2.0/FD).

Device feature include:

- 1.62V to 3.6V operation
- Arm 32-bit Cortex-M0+ with memory protection unit, up to 80MHz
- 128KB of flash with built-in ECC and 32KB of SRAM with hardware parity
- Two 12-bit 4-Msps ADCs
- 12-bit DAC
- Two zero-drift zero-crossover chopper op-amps
- Two 16-bit advanced control timers
- Three 16-bit general-purpose timers
- One 32-bit high-resolution time
- 60 GPIO

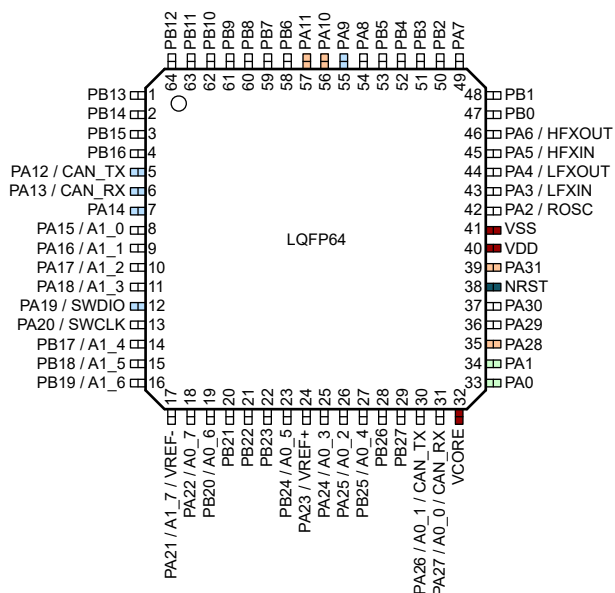
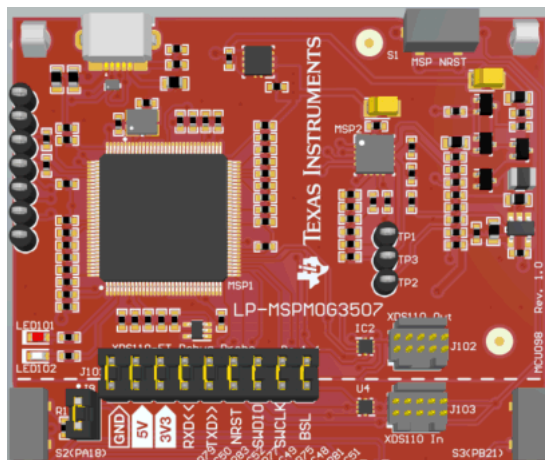


Figure 2-3. LQFP64 Pinout (Top View)

2.3.2 XDS110-ET 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 MSPM0G3507 has the XDS110 debug probe (see [Figure 2-4](#)), which is a simple and low-cost debugger that supports all MSPM0 device derivatives.



The XDS110 ET 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. More details can be found in [Section 2.3.4](#).

2.3.3 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 XDS110-ET domain into the MSPM0G3507 target domain. This includes XDS110-ET SWD signals, application UART signals, and 3.3V and 5V power.

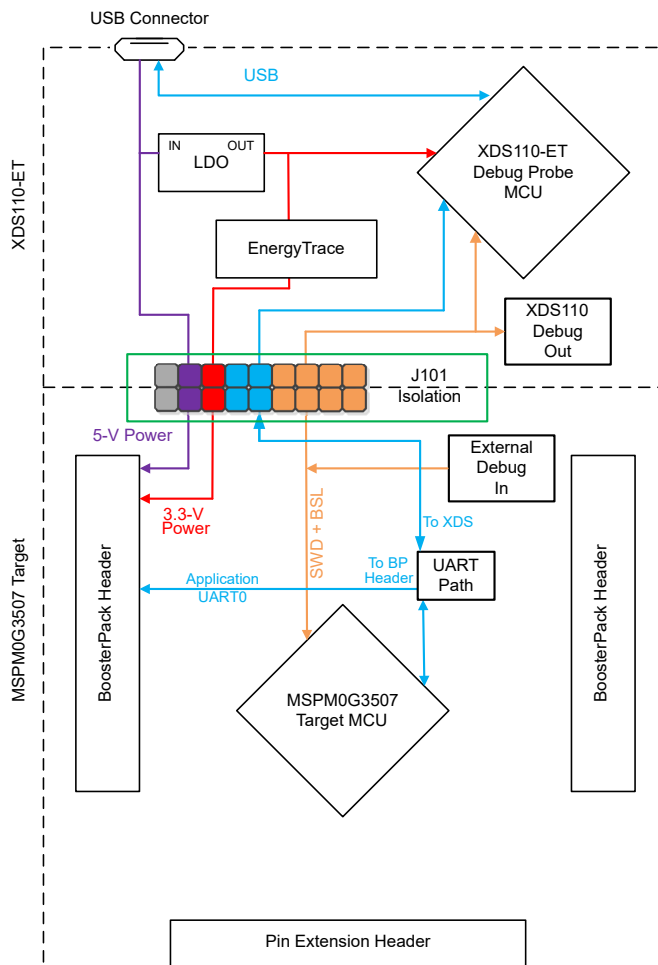


Figure 2-5. Isolation Jumper Block

Reasons to open these connections:

- To remove any and all influence from the XDS110-ET debug probe for high accuracy target power measurements
- To control 3.3V and 5V power flow between the XDS110-ET 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 XDS110-ET so that the programming and UART interface can be used for devices other than the onboard MCU.

Jumper	Description
GND	Ground
5 V	5V VBUS from USB
3V3	3.3V rail, derived from VBUS in the XDS110-ET domain
RXD<<	Backchannel UART: The target M0G3507 receives data through this signal. The arrows indicate the direction of the signal
TXD>>	Backchannel UART: The target M0G3507 sends data through this signal. The arrows indicate the direction of the signal
NRST	RST signal
SWDIO	Serial wire debug: SWDIO data signal.
SWCLK	Serial wire debug: SWDCLK clock signal
BSL	Bootstrap loader signal

2.3.4 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 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 2-5 shows the pathway of the backchannel UART. The backchannel UART is the UART on UART0 (PA10, PA11) depending on the jumper settings on headers J21 and J22.

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.

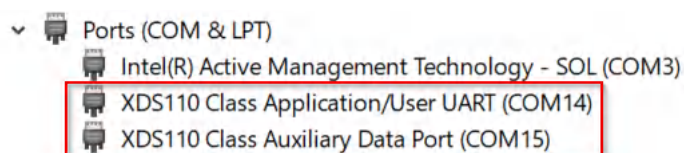


Figure 2-6. Application Backchannel UART in Device Manager

The backchannel UART is the "XDS110 Class Application/User UART" port. In this case, Figure 2-6 shows COM14, but this port can vary from one host PC to the next. After identifying the correct COM port, configure in your host application according to the documentation. The user can then open the port and begin communication from the host.

On the target MSPM0G3507 side, the backchannel is connected to the UART0 module. The XDS110-ET has a configurable baud rate; therefore, the PC application configuring the baud rate to be the same as what is configured on the UART0 is important.

2.3.5 Using an External Debug Probe Instead of the Onboard XDS110-ET

Many users have a preferred external debug probe and want to bypass the XDS110-ET debug probe to program the MSPM0 target MCU. The bypass is enabled by jumpers on isolation block J101 and the connector J103. Using an external debug probe is simple, and full JTAG access is provided through J103.

1. Remove jumpers on the JTAG signals on the J101 isolation block, including NRST, SWDIO and SWCLK.
2. Plug any Arm debug probe into J103.
 - a. J103 follows the Arm Cortex Debug Connector standard outlined in [Cortex-M Debug Connectors](#).
3. Plug USB power into the LaunchPad development kit, or power the kit externally.
 - a. Make sure that the jumpers across 3V3 and GND are connected if using USB power.
 - b. External debug probes do not provide power, the VCC pin is a power sense pin.
 - c. For more details on powering the LaunchPad development kit, see [Section 2.4](#)

2.3.6 Using the XDS110-ET Debug Probe With a Different Target

The XDS110-ET debug probe on the LaunchPad development kit can interface to most Arm Cortex-M devices, not just the onboard target MSPM0G3507 device. This functionality is enabled by the J102 10-pin Cortex-M JTAG connector and a 10-pin cable. Header J102 follows the Cortex-M Arm standard, however, pin 1 is not a voltage sense pin. The XDS110-ET outputs only 3.3V JTAG signals. If another voltage level is needed, the user must provide level shifters to translate the JTAG signal voltages. Additionally, 3.3V of output power can be sourced from the XDS110-ET when jumper JP102 is connected. This allows the XDS110-ET to power the external target at 3.3V through pin 1. EnergyTrace functionality is not available when programming an external target.

1. Remove jumpers on the JTAG signals on the J101 isolation block, including RST, TMS, TCK, TDO, and TDI.
2. Plug the 10-pin cable into J102, and connect to an external target. J102 follows the Arm Cortex Debug Connector standard outlined in [Cortex-M Debug Connectors](#).
3. Plug USB power into the LaunchPad development kit, or power the kit externally. JTAG levels are 3.3V ONLY.

2.3.7 Special Features

2.3.7.1 Up to 4-Msps High-Speed Analog-to-Digital Converter

MSPM0 G-series device includes an up to 4-Msps high-speed ADC module. To help evaluate the best performance of the ADC, an external OPA is also included on the LaunchPad kit. This external OPA can be used as a buffer or to amplify the input signal. With the built-in ADC module, the device can achieve a maximum rate of 4 Msps.

The LaunchPad kit uses the OPA2365 in the ADC input channel in buffer mode by default (see [Figure 2-7](#)). The OPA2365 has 2 OPAs inside, and footprints for external passives are provided to allow users to customize the OPA configuration.

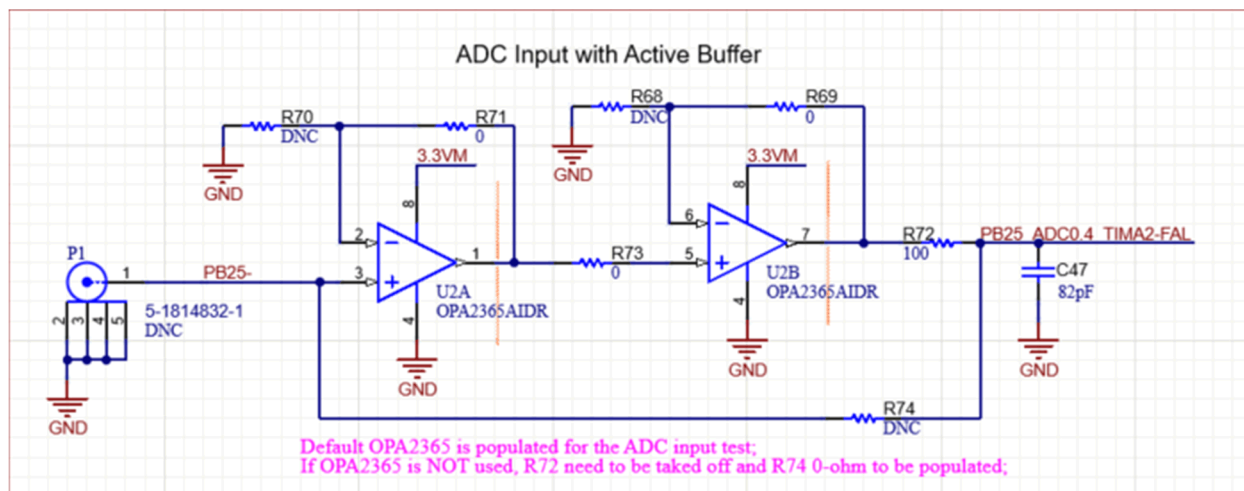


Figure 2-7. External OPA Circuit

For the signal input, a coaxial connector can be populated at P1 to reduce environmental noise. You can also access the OPA input through the pin extension header at the bottom of the board.

The OPA output connects to the ADC0.4 channel, and the example code helps to evaluate ADC performance quickly.

2.3.7.2 Thermistor

The LaunchPad kit includes a 10k linear thermistor (PTC), the TMP6131. [Figure 2-8](#) shows the circuit. The PTC is in a low-side configuration with a 10k Ω 10ppm pullup resistor. The default position of the jumper in the header (1-2 position) connects this circuit directly to the ADC. In this mode, the output voltage is approximately 1.6V at room temperature.

The jumper can also be set to the 2-3 position to allow an OPA connection to the thermistor. The OPA output can be connected internally, through software, to the negative OPA terminal. This allows the OPA to be used as a buffer for the thermistor with an internal connection to an ADC channel for sampling. If an external connection or additional biasing or filtering is wanted, then the passives shown in [Figure 2-8](#) need to be populated.

Note

The capacitor and resistor network associated with the OPA is not populated by default. This allows the user to populate with passive values to fit the specific application needs.

Note

When using the OPA and the passive network, be sure to check for conflicts with other pins on the LaunchPad kit because pins are multiplexed with several functions.

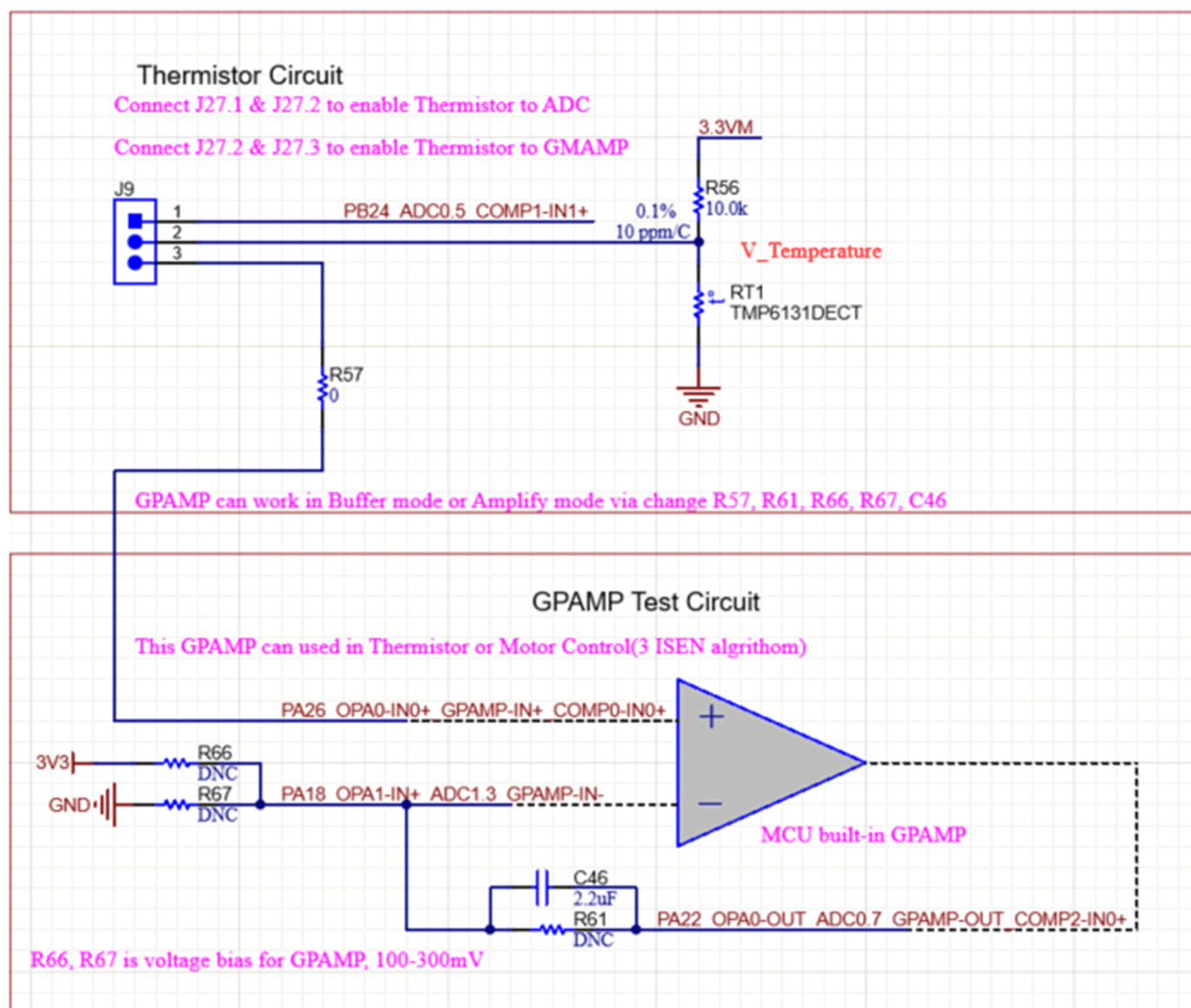


Figure 2-8. Thermistor Circuit

2.3.7.3 Light Sensor

A light sensor circuit is provided on the board using the photodiode in the configuration shown in [Figure 2-9](#). The jumper connections between this circuit and the MSPM0 are populated by default so the photodiode is in circuit. The jumpers can be removed to place the photodiode out of circuit so the MSPM0 connections can be used for other evaluation purposes.

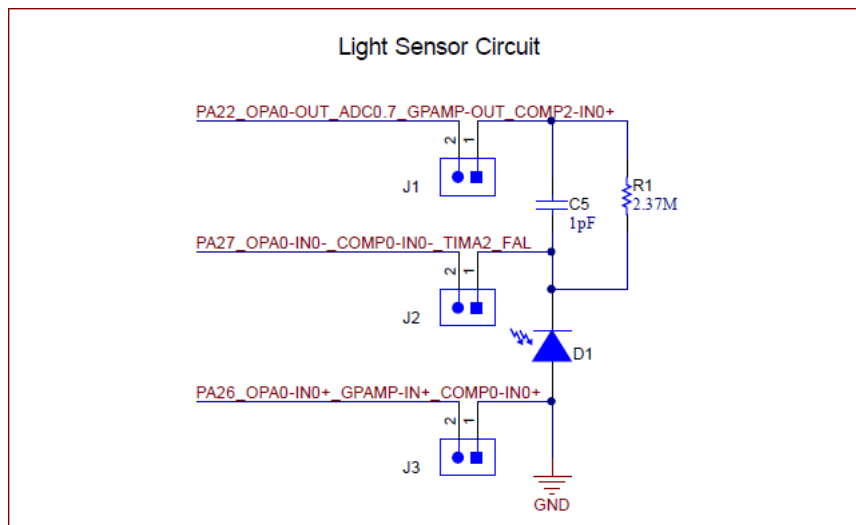


Figure 2-9. Light Sensor Circuit

This circuit is intended to be used with the internal OPA of the MSPM0 in an transimpedance configuration to drive the circuit. The OPA output is internally set to an ADC channel for sampling.

2.4 Power

The board accommodates various powering methods, including through the onboard XDS110-ET as well as external or BoosterPack plug-in module power.

2.4.1 XDS110-ET USB Power

The most common power-supply scenario is from USB through the XDS110-ET debugger. This provides 5V power from the USB and also regulates this power rail to 3.3V for XDS110-ET operation and 3.3V to the target side of the LaunchPad kit. Power from the XDS110-ET is controlled by jumper J101. For 3.3V, make sure that a jumper is connected across the J101 3V3 terminal.

2.5 External Power Supply and BoosterPack Plug-in Module

Header J11 is present on the board to supply external power directly. The importance is to comply with the device voltage operation specifications when supplying external power. The MSPM0G3507 has an operating range of 1.62V to 3.6V. More information can be found in the [MSPM0G3507 data sheet](#).

2.6 Measure Current Draw of the MSPM0 MCU

To measure the current draw of the MSP430FR2355 MCU using a multimeter, use the 3V3 jumper on the J101 jumper isolation block. The current measured includes the target device and any current drawn through the BoosterPack plug-in module headers.

To measure ultra-low power, follow these steps:

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 MSPM0G3507 can have on the current draw. Consider disconnecting these at the isolation jumper block, or at least consider the current sinking and sourcing capability in the final measurement.
 - a. Disconnect unnecessary jumpers as outlined in the jumper map of [Table 2-1](#). In particular, Analog Power jumper J13 must be removed if not using the thermistor or OPA2365 circuits.
3. Make sure there are no floating inputs or outputs on the MSPM0G3507. This causes unnecessary extra current draw. Every I/O is either driven out or, if there is an input, then the I/O is 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, then getting a stable measurement is difficult. Measuring quiescent states is easier.

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

2.7 Clocking

The MSPM0G3507 provides external clocks in addition to the internal clocks in the device.

- Y1: 32.768kHz 12.5pF crystal
- Y2: 40MHz 12.5pF crystal

The internal SYSOSC is 32 MHz by default at an accuracy of 2.5%. For higher accuracy, connect a 0.1% resistor at ROSC pin. The MCLK is sourced by 32MHz SYSOSC at default. CPUCLK is sourced directly from MCLK in RUN mode and disabled in other modes. The low-power clock (ULPCLK) can be sourced by MCLK and active in RUN and SLEEP mode by configuration. For more clock tree details, see Section 2.3 *Clock Module (CKM)* in the [MSPM0 G-Series 80MHz Microcontrollers Technical Reference Manual](#).

2.8 BoosterPack Plug-in Module Pinout

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

While most BoosterPack plug-in modules are compliant with the standard, some are not. The MSPM0G3507 LaunchPad development kit is compatible with all 40-pin BoosterPack plug-in modules that are compliant with the standard. If the reseller or owner of the BoosterPack plug-in module does not explicitly indicate compatibility with the MSPM0G3507 LaunchPad development kit, compare the schematic of the candidate BoosterPack plug-in module with the LaunchPad development kit to verify compatibility. Keep in mind that sometimes conflicts can be resolved by changing the MSPM0G3507 device pin function configuration in software.

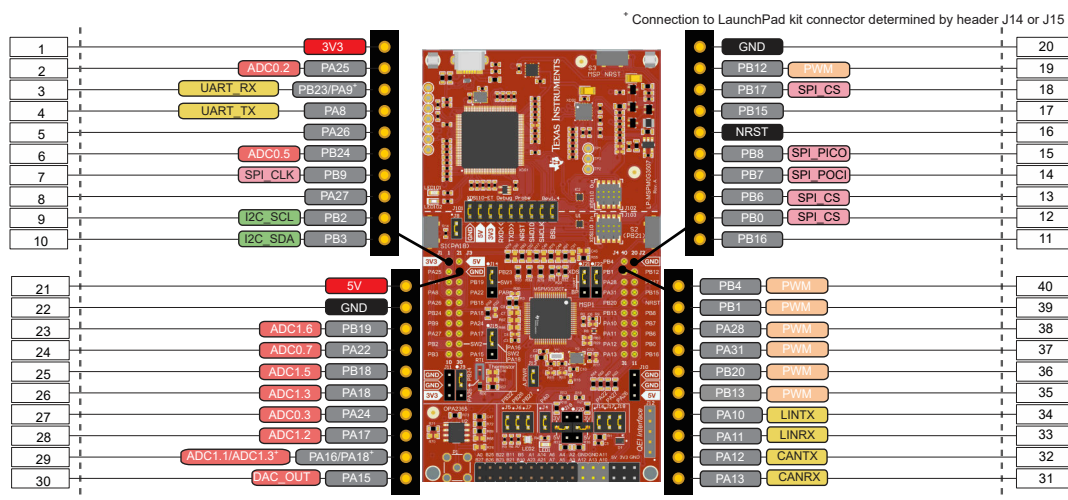


Figure 2-10. LaunchPad Development Kit to BoosterPack Plug-in Module Connector Pinout

3 Software Examples

See the [MSPM0 SDK documentation](#) for more details about available software.

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 the source files 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 MSPM0_SDK content and a web-based IDE.

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

4.1.3 Code Composer Studio Cloud

Code Composer Studio Cloud (CCS Cloud) is a web-based IDE that lets you quickly create, edit, build, and debug applications for the LaunchPad development kit. No need to download and install large software packages, simply connect your LaunchPad development kit and begin. Choose 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 more information, see the [full comparison between CCS Cloud and CCS Desktop](#).

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

4.1.4 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. Code Composer Studio includes an optimizing C/C++ compiler, source code editor, project build environment, debugger, profiler, and many other features.

To learn more about CCS and view user's guides, visit the [CCS tool page](#).

CCS v11.1 or higher is required. When CCS has been launched, and a workspace directory chosen, use Project>Import Existing CCS Eclipse Project. Direct to the desired demo project directory that contains main.c.

4.2 MSPM0 SDK and TI Resource Explorer

TI Resource Explorer is a tool integrated into CCS that allows you to browse through available design resources. TI Resource Explorer helps you quickly find what you need inside packages. 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 4.1.2](#) for more information.

4.3 MSPM0G3507 MCU

4.3.1 Device Documentation

More information about the MSPM0G3507 device is available. For every MSP device, the documentation is organized as shown in [Table 4-1](#).

Table 4-1. Device Documentation

Document	For MSPM0L1306	Description
Device family TRM	MSPM0 G-Series 80MHz Microcontrollers Technical Reference Manual	Architectural information about the device, including all modules and peripherals such as clocks, timers, ADC, and so on
Device-specific data sheet	MSPM0G350x Mixed-Signal Microcontrollers With CAN-FD Interface MSPM0G310x Mixed-Signal Microcontrollers With CAN-FD Interface MSPM0G150x Mixed-Signal Microcontrollers MSPM0G110x Mixed-Signal Microcontrollers	Device-specific information and all parametric information for this device

4.3.2 MSPM0G3507 Code Examples

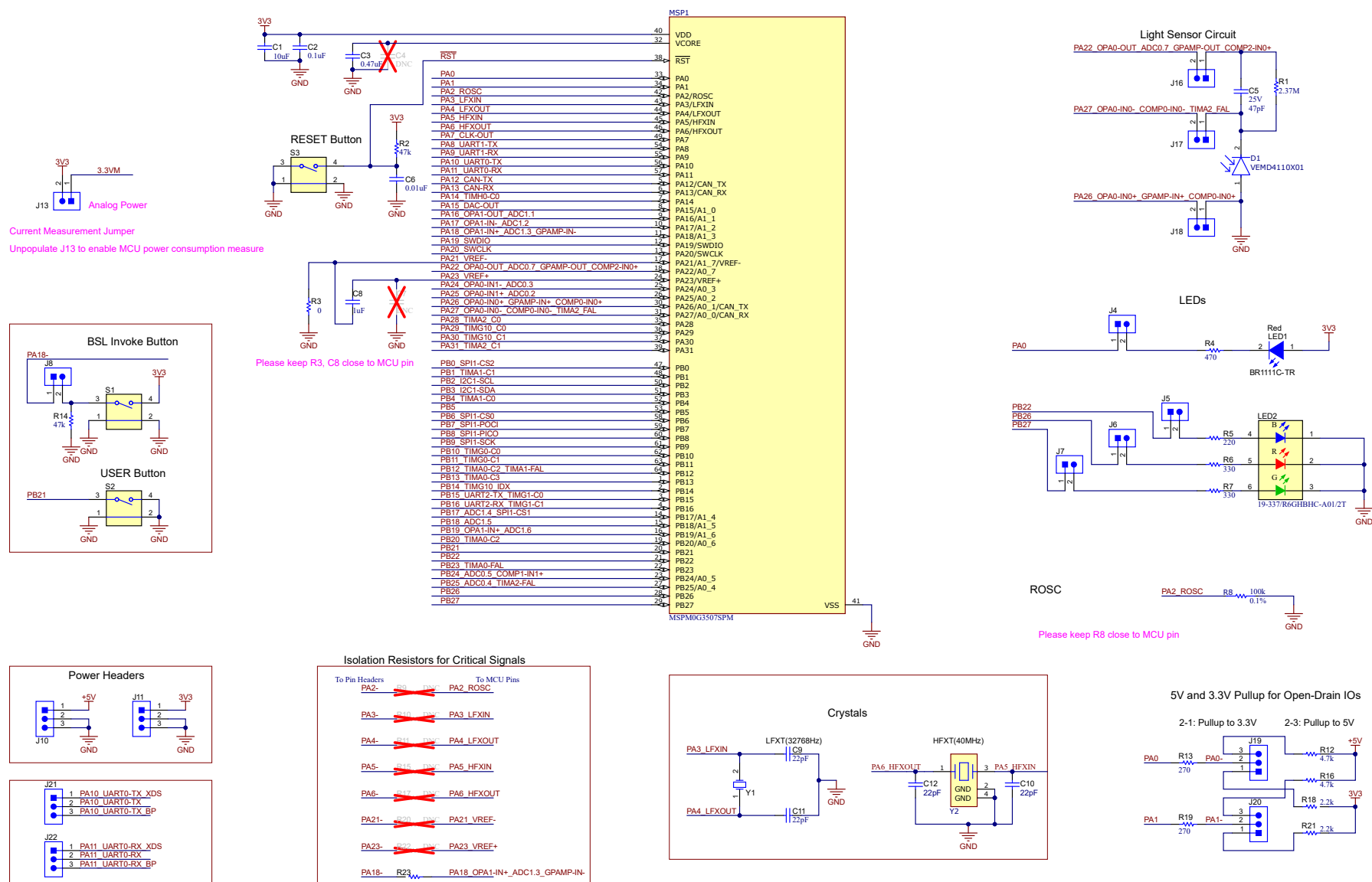
[MSPM0_SDK](#) has a set of simple C examples that demonstrate how to use the entire set of peripherals on the MSPM0G3507 MCU. Every MSP derivative has a set of these code examples. When starting a new project or adding a new peripheral, these examples serve as a great starting point.

4.4 Community Resources

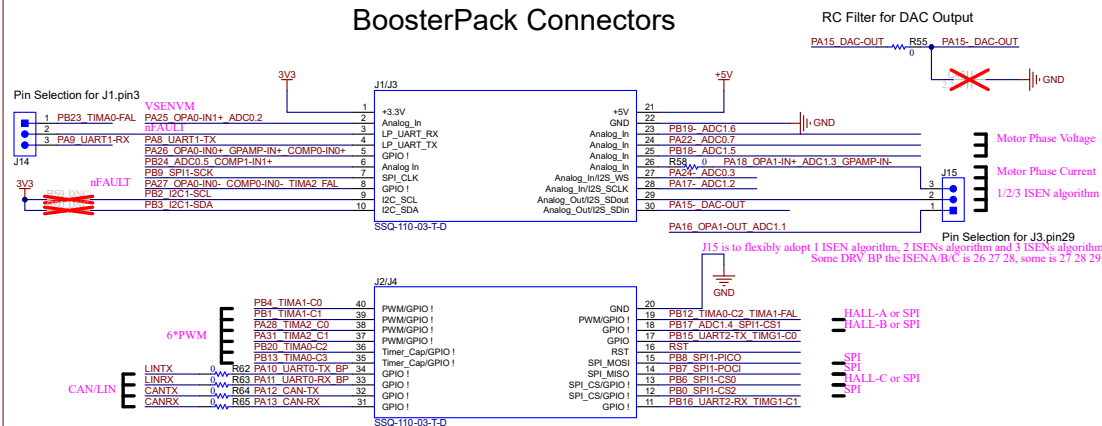
4.4.1 TI E2E™ Forums

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

5 Schematics

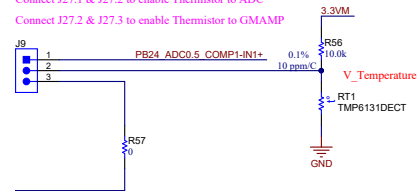


BoosterPack Connectors



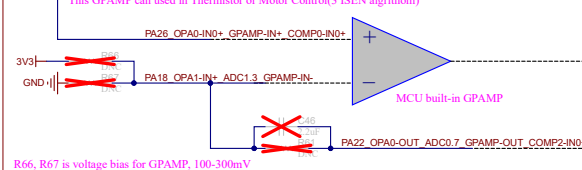
Thermistor Circuit

Connect J27.1 & J27.2 to enable Thermistor to ADC
Connect J27.2 & J27.3 to enable Thermistor to GPAMP

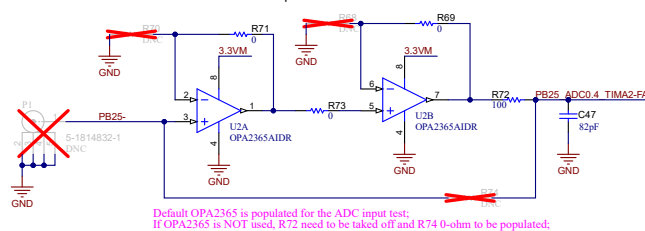


GPAMP Test Circuit

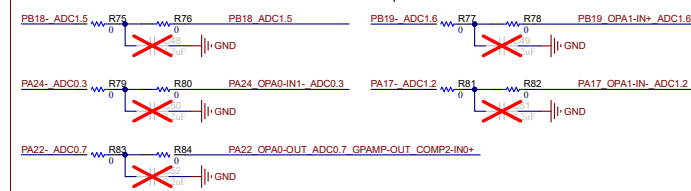
This GPAMP can be used in Thermistor or Motor Control(3 ISEN algorithm)

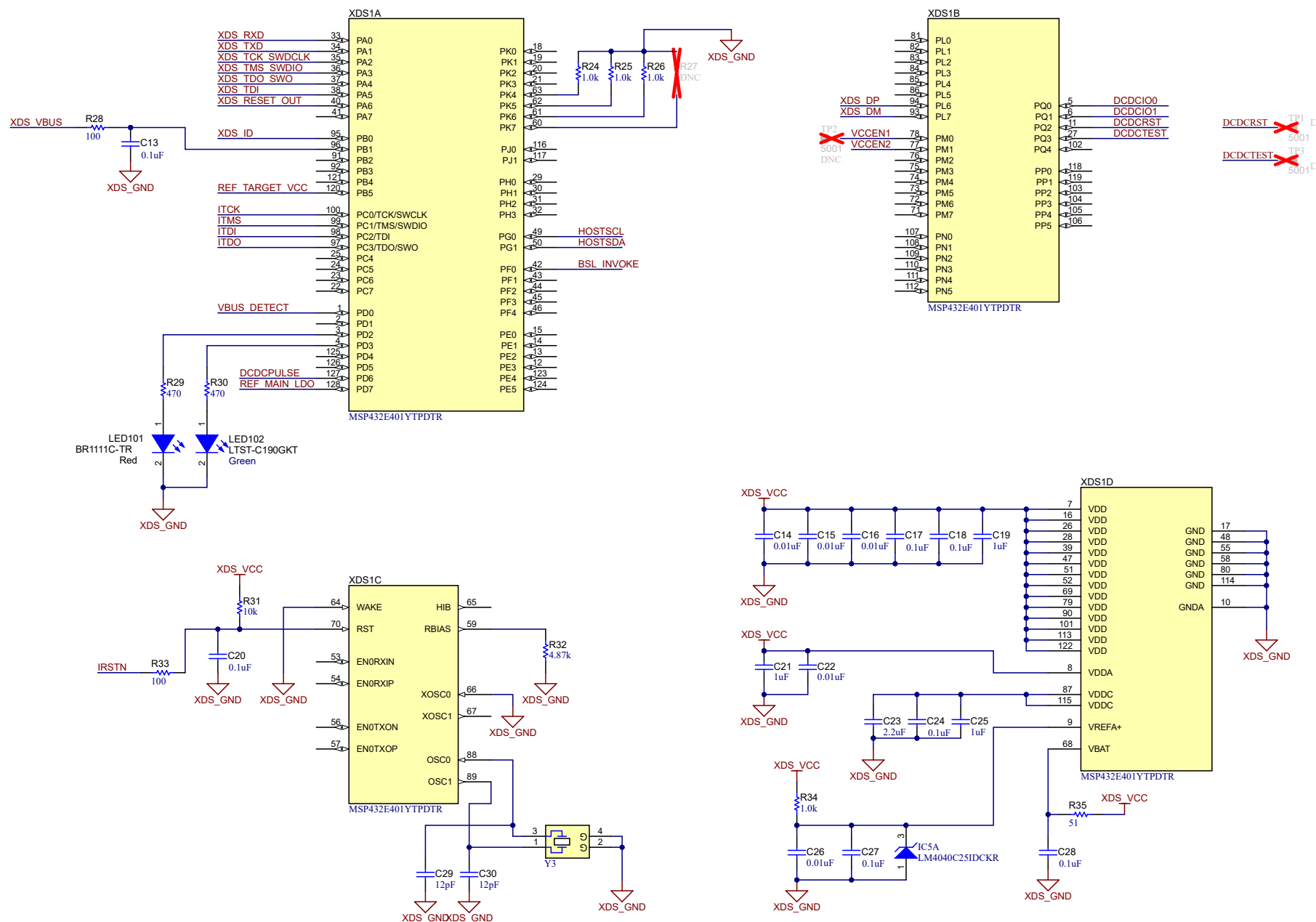


ADC Input with Active Buffer



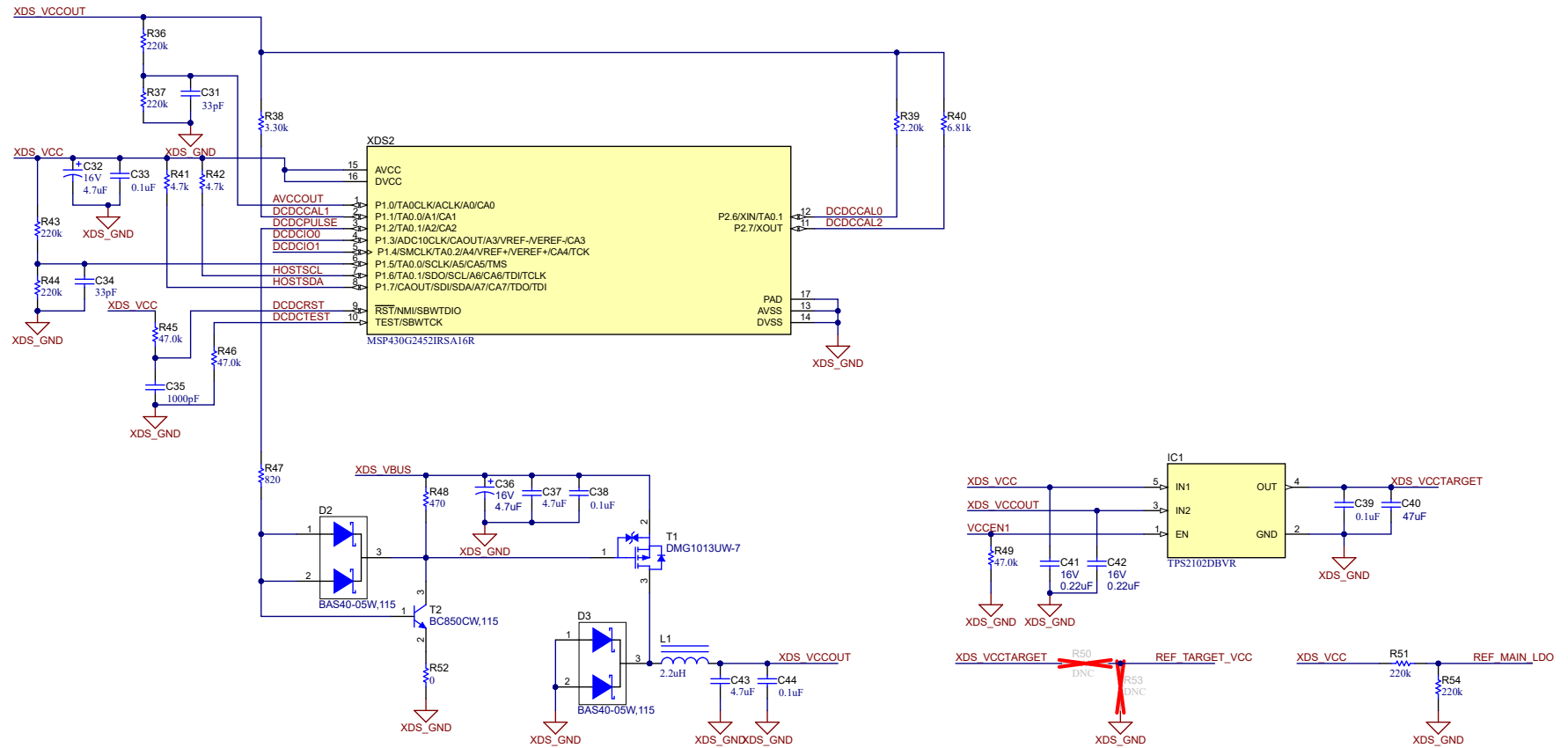
RC Filter for ADC Input

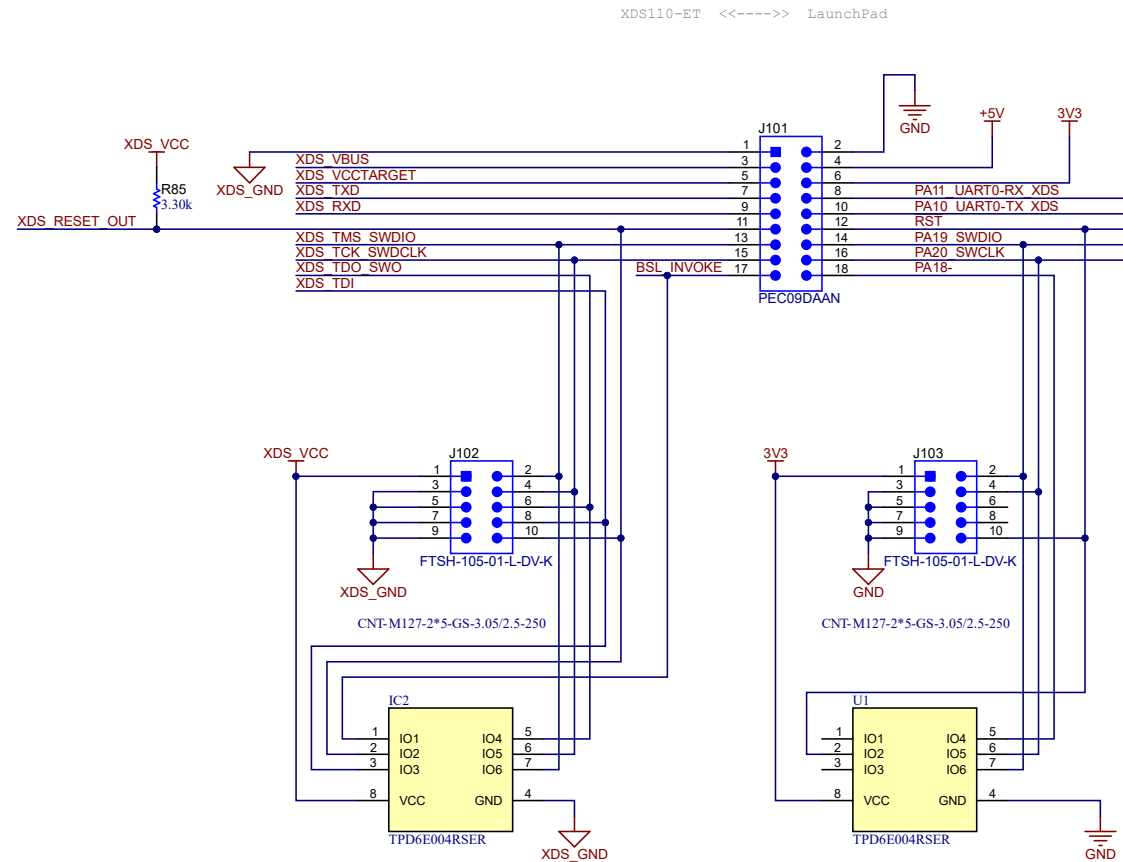


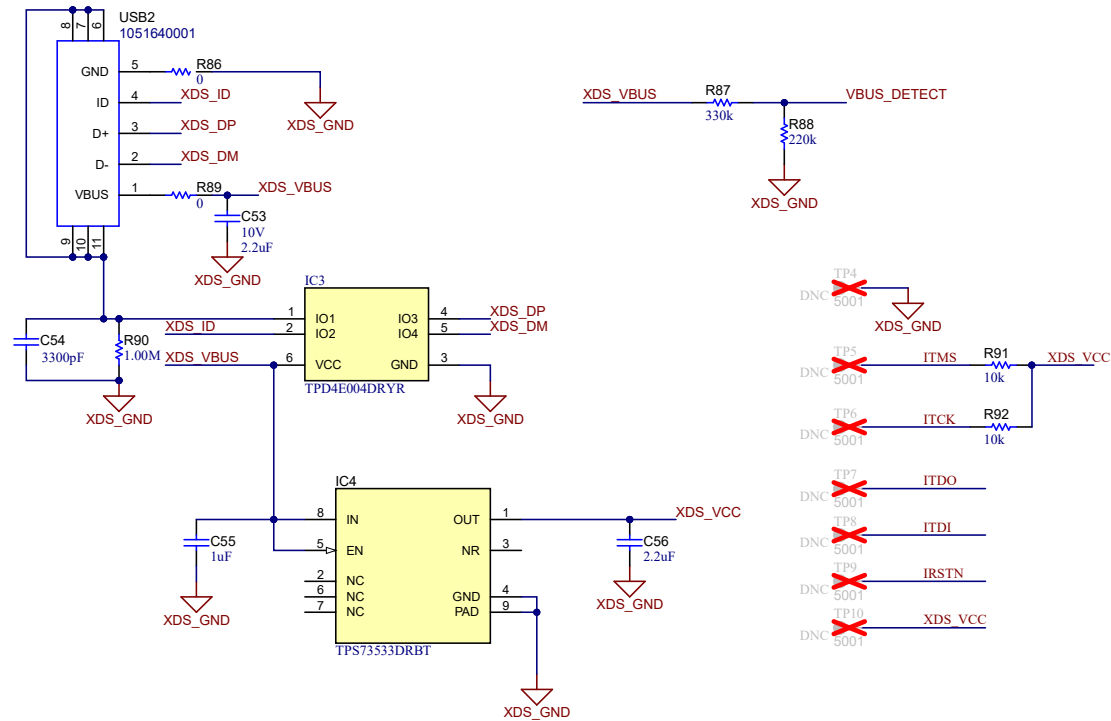


Software-controlled DCDC converter

Energy measurement method protected under U.S. Patent Application 13/329,073 and subsequent patent applications







6 Revision History

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

Changes from Revision B (March 2023) to Revision C (November 2023) Page

- Updated all images in [Section 5 Schematics](#) 18

Changes from Revision A (March 2023) to Revision B (November 2023) Page

- Changed HF Xtal to 40-MHz crystal..... 3
- Deleted *Design Files* 5
- Changed HF Xtal to 40-MHz 12-pF crystal..... 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](#) 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 © 2024, Texas Instruments Incorporated