TI Designs
Fast Prototyping 24-Bit Sigma-Delta ADC With Energia Ecosystem

TI Designs

This TI Design showcases a Sigma-Delta ADC BoosterPack™ for TI LaunchPad™ and how to proceed to a fast prototyping evaluation with Energia leading analog performances evaluation within minutes.

Design Resources

TIDA-00647	Design Folder
ADS1220	Product Folder
TIDA-00648	Tools Folder
TIDA-00339	Tools Folder

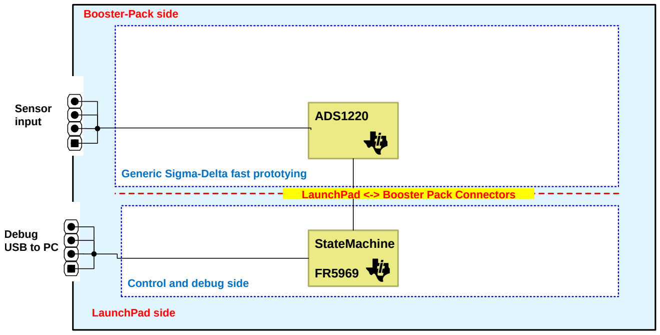
TI E2E™ Community   ASK Our E2E Experts

Design Features

- Energia Control of ADS1220
- Energia Libraries to Convert Data
- Energia Library for ADC Code to Convert Temperature
- 24-Bit ΔΣ ADC: ADS1220
- Physical Form Factor in BoosterPack for Compatibility With all TI LaunchPad Devices
- Scalable From Generic to Specific Evaluation Within Minutes

Featured Applications

- Factory Automation And Process Control
- Sensors And Field Transmitters



An IMPORTANT NOTICE at the end of this TI reference design addresses authorized use, intellectual property matters and other important disclaimers and information.

1 System Overview

1.1 System Introduction

For signals with relatively low bandwidth, up to 100 kHz, Delta-Sigma converters offer higher performances than other topologies [1] [2]. This TI Design leverages Energia and its rapid prototyping capabilities to evaluate hardware, such as a high-performance ADC, within minutes.

Table 1. Design Solutions to Industry Problems

INDUSTRY CHALLENGE	TIDA-00468 SOLUTION
Reducing time to market windows	Fast prototyping solution for low power 24-bit SD ADC
Be close to end application for relevant evaluation	Show case how to use the platform for universal temperature input evaluation

1.2 Key System Specifications

Table 2. Key System Specifications

PARAMETER	DETAILS
2-wire RTD input	Section 7.1
3-wire RTD input	Section 7.2
4-wire RTD input	Section 7.3
Thermocouple input	Section 7.4
Universal temperature input	Section 7.5

2 System Design Theory

2.1 Delta-Sigma

The DS converter is made of three main blocks: the DS modulator, the digital filter, and decimation. The internal DS modulator shown in Figure 1 coarsely samples the input signal at a very high rate into a 1-bit stream. The digital and decimation filters then take this sampled data and convert it into a high-resolution, slower digital code. While most converters have one sample rate, the DS converter has two—the input sampling rate (f_S) and the output data rate (f_D).

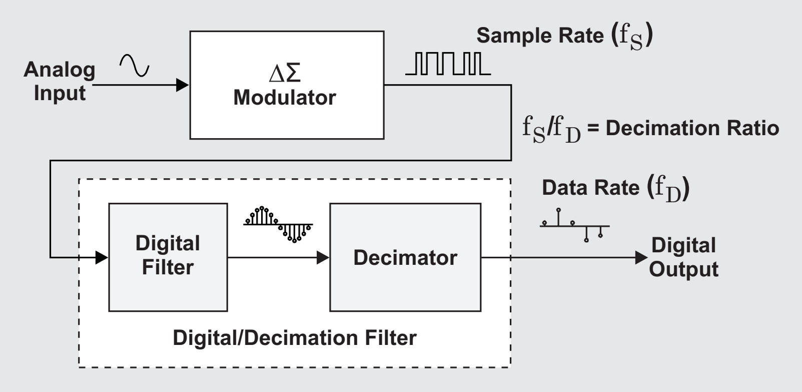


Figure 1. Block Diagram of DS ADC

2.1.1 ΔΣ Modulator

The modulator can be modeled either in a time domain (Figure 2) or in a frequency domain.

In the time domain, the input signal (xi) and the output of a 1-bit DAC (x4) are differentiated to generate an intermediate analog voltage (x2), which is integrated into another analog value (x3) and compared to a threshold value. The output value (x4) is clocked out of the ADC as well as to the DAC for differentiation against the input value (keeping in mind that the input value is considered almost constant against changes of x4).

After a certain time, the value x3 reaches the reference and x4 changes sign, as does x2 and the slope of x3. Such a modulation scheme proves the quantization noise is as described in Figure 2. Higher order of modulations with better noise shaping performances can be designed, but these designs usually come at the expense of the response.

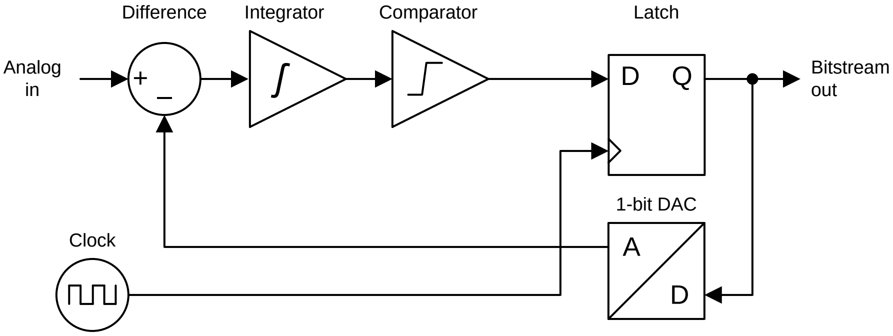


Figure 2. Block Diagram of First-Order Sigma-Delta in Time Domain

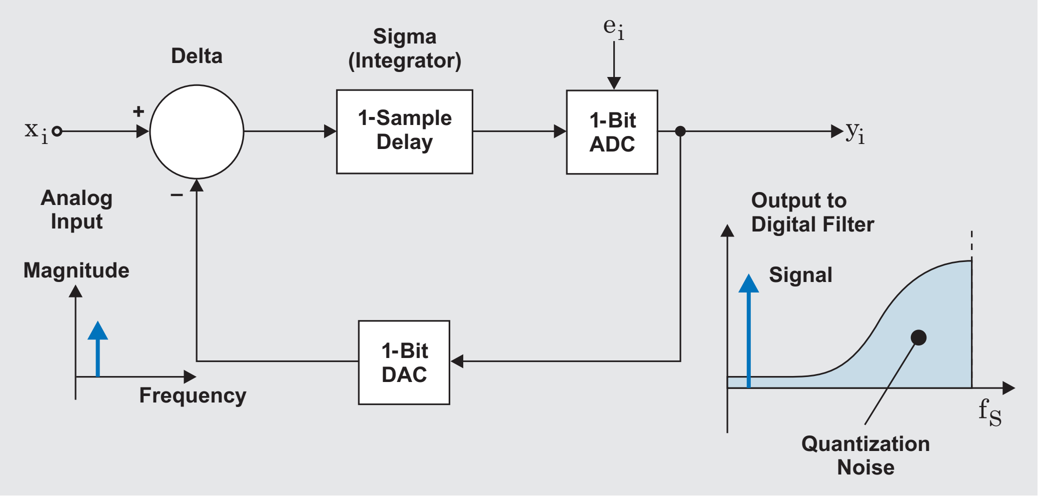


Figure 3. First-Order DS Modulator in Frequency Domain

2.1.2 Digital Filter

The digital filter mentioned in Figure 1 can be implemented as shown on Figure 4; alternatively, more advanced filters such as sync filters can be used as well depending on the desired outcome.

The main function of the filter is to remove the quantization noise in the higher frequencies present at the output of the 1-bit ADC (see Figure 4). Since the noise is not flat, removing the higher frequency content, will increase the signal to noise ratio.

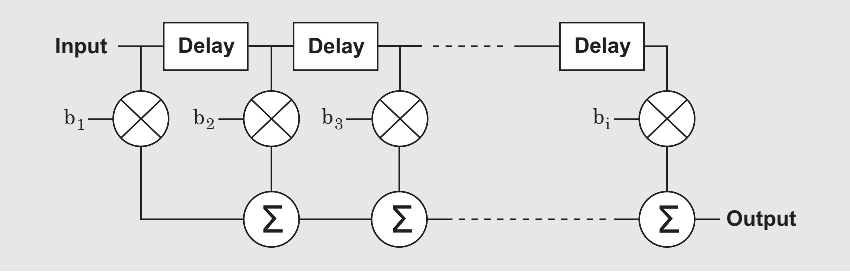


Figure 4. First-Order, Low-Pass Averaging Filter

2.1.3 Decimator

The decimator role mostly provides a manageable data output rate.

2.1.4 Summary

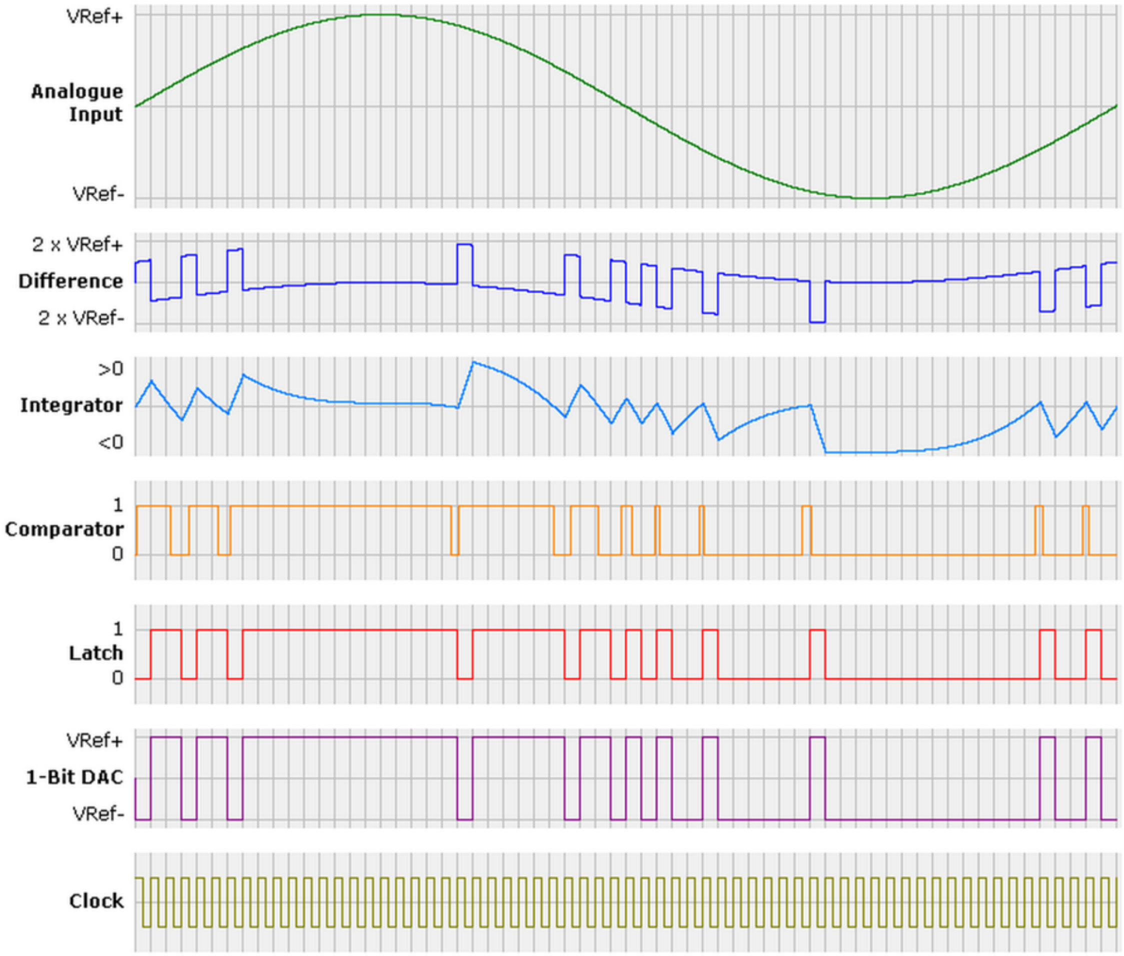


Figure 5. Overview of Signals at Different Stages of Sigma-Delta

2.2 TI Rapid Prototyping Environment

The TI rapid prototyping environment is made of the following three key components:

- Energia: the graphical user interface running on all main operating systems
- Sketch: the name that Energia uses for a program; it is the unit of code that is uploaded to and runs on a LaunchPad board
- LaunchPad: the board on which resides the microcontrollers from Texas Instruments
- BoosterPack: the add-on boards connected to the LaunchPad, extending the functionalities of the MCU (and exercised thanks to a sketch)

CAUTION

While simplifying the design, this rapid prototyping environment does not compromise on performances: the program running on the LaunchPad has been processed by Energia then compiled using the standard C++ compiler before being loaded.

2.2.1 Energia

Energia is an open-source integrated development environment (IDE) for rapid prototyping bringing the wiring framework (the framework underlying other rapid prototyping environment) to the Texas Instruments LaunchPad evaluation kit.

The wiring framework is an open-source programming framework for microcontrollers. This framework allows cross-platform software to control devices attached to a wide range of microcontrollers.

The wiring-based Energia IDE is an application written in Java, derived from the IDE made for the processing programming language (for instance, the name "sketch" is derived from the processing IDE).

The Energia IDE comes with a C/C++ library called "Energia", which makes common input/output operations much easier.

Energia programs are written in C/C++, though most of the complexity of C/C++ code is hidden by the Energia IDE. For instance, only two functions need to be defined for a program to run (though more functions can be defined):

- Setup(): a function that runs once at the start of the program, which can be used to define initial environment settings
- Loop(): a function called repeatedly until the board is powered off

2.2.2 LaunchPad Ecosystem

LaunchPads are microcontroller development kits from Texas Instruments that focus easy evaluations:

- Simple USB interface to PC (no more need for additional debugger to connect to the JTAG port as the logic is already on the LaunchPad board)
- Standardized interface for extension boards (called BoosterPacks)

LaunchPads comply to the electrical interface specifications shown in Figure 6, which is available as a PDF here: http://www.ti.com/ww/en/launchpad/dl/boosterpack-pinout-v2.pdf

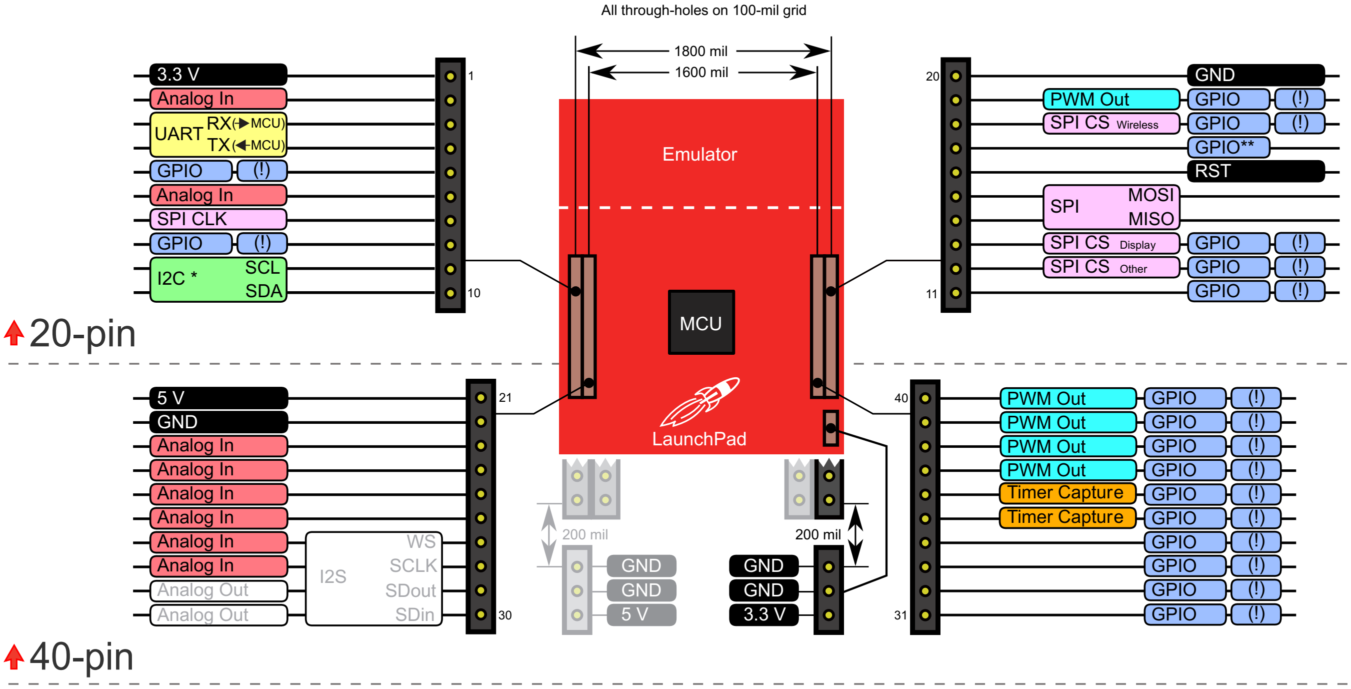


Figure 6. Electrical Interface Specifications

2.2.3 BoosterPack Ecosystem

BoosterPacks are plug-in modules that fit on top of LaunchPads. These innovative tools plug into a consistent and standardized connector on the LaunchPad and allow developers to explore different applications enabled by any TI microcontroller.

BoosterPacks are available from Texas Instruments, from third parties, and from the community. BoosterPacks include functions such as capacitive touch, wireless communication, sensor readings, LED lighting control, and more. BoosterPacks are available in 20- and 40-pin variants, and multiple BoosterPacks can plug into a LaunchPad to enhance the functionality of a design.

2.2.4 LaunchPad, BoosterPack, and Energia Abstraction

BoosterPacks can be mechanically and electrically connected as described in Figure 7 and Figure 8.

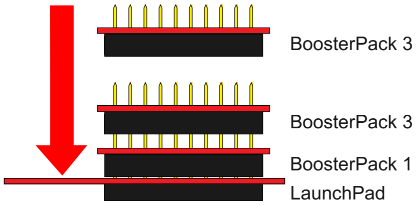


Figure 7. Mechanical and Electrical Connections Between LaunchPad and BoosterPack

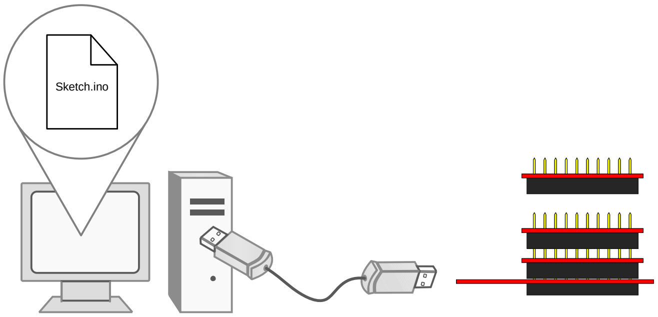


Figure 8. Simple USB Connection From LaunchPad to PC

The Energīa framework binds the entire rapid prototyping environment together:

- Editing and saving the source code (called sketch)
- Compiling and linking the source to binaries
- Loading the binaries over USB to the target (LaunchPad)
- Displaying the messages (serial port over USB) from the LaunchPad

2.2.5 IDE

The IDE provides a simplified development window that is ideal for engineers who are not familiar with software development tools such as Code Composer Studio™ (CCS). Engineers (should they be hardware engineers, system architects, or so on) can then develop the logic in small files called a "sketch", which are then loaded on the LaunchPad for execution.

Energīa provides lot of the more basic and core functionalities needed for introductory applications and projects. Once proof of concept applications are realized, development will likely leverage CCS, a full-blown professional development environment that introduces more capabilities and functionality such as full debug.

2.2.6 Compiling and Loading

Figure 9 summarizes what Energīa does in the background before loading the compiled code on the LaunchPad.

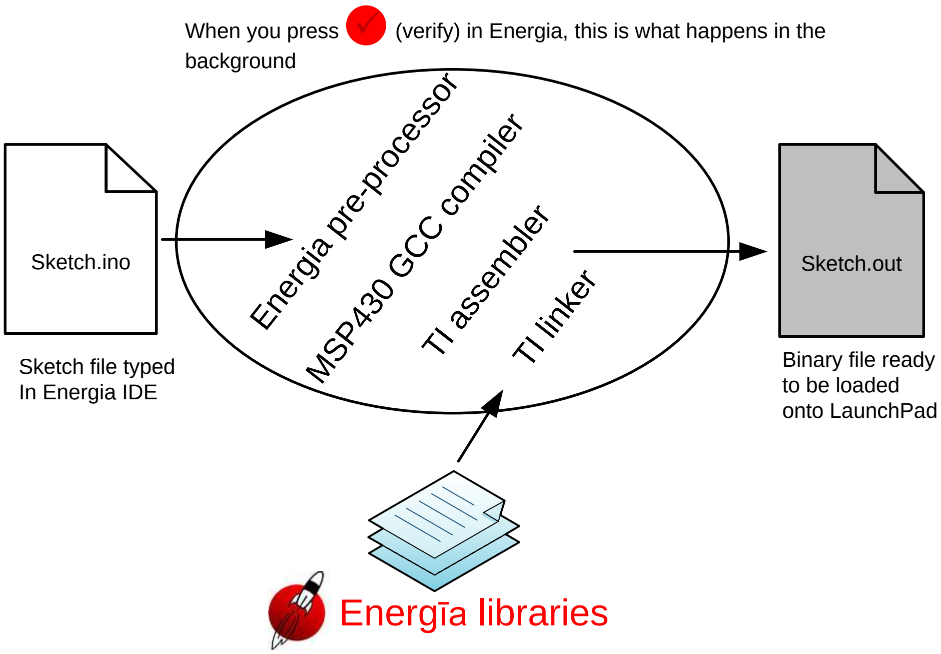


Figure 9. Energia Build Process

More specifically, the following events must happen for the Energīa code to get onto the LaunchPad board:

First, the Energīa environment performs some small transformations to make sure the code is correct in C or C++ (two common programming languages) including the "Energia libraries" (#include “Energia.h”), which contain all the definitions needed for the standard Energīa core. If a sketch contains multiple files (tabs), all tabs without an extension are concatenated together to form a main one. Tabs with .c, .cpp, or .h will be compiled separately while others will be renamed. Then Energīa searches for function definitions and creates prototypes for them. Then Energīa adds the target main.cxx file to the bottom of the sketch, which includes some of the target specific settings (a part of what allows the rapid prototyping as the code can start from a known good setup).

The object files then get passed to a compiler (MSPGCC), which turns the human readable code into machine readable instructions (or object files). Then, the code gets combined with (linked against) the standard Energīa libraries that provide basic functions like digitalWrite() or Serial.print(). The result is a single Intel hex file, which contains the specific bytes that need to be written to the program memory of the chip on the LaunchPad.

The .cxx file can then be uploaded to the board. It is transmitted over the USB or serial connection through the bootloader already on the chip or with external programming hardware due to the mspflasher.

2.2.7 Running

Once the firmware has been uploaded and stored in the non-volatile memory (NVM), which can be Flash memory or FRAM memory, the debug logic on the LaunchPad will issue a hardware reset, which will then launch the newly programmed firmware.

2.2.8 Debugging

With Energīa, the default debug consists mostly in UART traces going over the USB into the terminal of the Energīa GUI.

For more advanced features, it is recommended to import sketches into CCS (see Section 2.2.9).

2.2.9 Importing to CCSv6

Energīa sketches can be imported to CCS.

Importing sketches allows users to reuse and scale the work done during the early prototyping stages by the software teams.

For more details, see the Energia site [8] or Section 11 where the tutorial was captured for future. The only code change is to include the Energia library to be able to leverage the Energia API.

2.2.10 Scalability and Reuse

Sketches as well as the binaries generated with Energia can be loaded on boards other than LaunchPads with CCS in conjunction with an emulator.

The only constraint is that the allocation of pins remains the same than on the original LaunchPad on which the code was initially developed and validated.

Beyond binary reuse, source code can also be reused and integrated in software projects to allow reuse of the functionalities validated by the prototyping team by the software team.

2.2.10.1 Accessing Map File for Memory Footprint Evaluation

Another aspect which can be of interest when transitioning the code from a fast prototyping environment to a software production environment is related to the memory footprint used by the framework. While the tools do not keep the actual .map file, it is possible to re-run the tool chain and by adding the relevant flags to keep the map file. in the log window during the compile, find the following output (often line 8 and 9 from the bottom):

```
<Energia install folder>hardware\tools\msp430\bin\msp430-gcc -Os -Wl,-gc-sections,-
u,main -mmcu=msp430f5529 -o
<User folder>\AppData\Local\Temp\build2484947497227615311.tmp\sketch_aug18a.cpp.elf
<User folder>\AppData\Local\Temp\build2484947497227615311.tmp\sketch_aug18a.cpp.o
<User folder>\AppData\Local\Temp\build2484947497227615311.tmp\core.a -L
<User folder>\AppData\Local\Temp\build2484947497227615311.tmp -lm
```

Take this and add: `-Wl,-Map "x.map"`

or run in the output folder (for example, <User folder>\AppData\Local\Temp\build2484947497227615311.tmp):

```
<Energia install folder>\hardware\tools\msp430\bin\msp430-size.exe -t
sketch_aug18a.cpp.elf
```

3 System Description

This design focuses on bringing a fast prototyping solution for low power ΣΔ ADC to the Energia, LaunchPad, or BoosterPack ecosystem.

It consists of an ADS1220 mounted on a BoosterPack mechanical and electrical board as described in Figure 10:

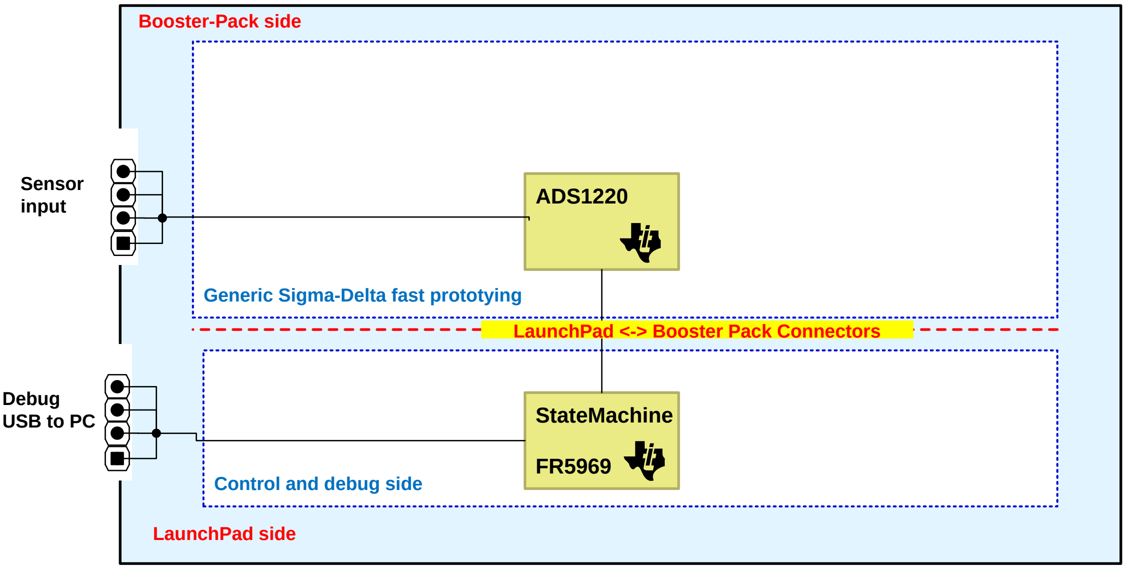


Figure 10. Block Diagram

3.1 ADS1220

Features:

- Low current consumption: As low as 120 µA (typ) in Duty-Cycle mode
- Wide supply range: 2.3 V to 5.5 V
- Programmable gain: 1 V/V to 128 V/V
- Programmable data rates: Up to 2 kSPS
- Up to 20-bits effective resolution
- Simultaneous 50-Hz and 60-Hz rejection at 20 SPS with single-cycle settling digital filter
- Two differential or four single-ended inputs
- Dual matched programmable current sources: 10 µA to 1.5 mA
- Internal 2.048-V reference: 5 ppm/°C (typ) drift
- Internal 2% accurate oscillator
- Internal temperature sensor: 0.5°C (typ) accuracy
- SPI-compatible interface (mode 1)
- Package: 3.5×3.5×0.9-mm VQFN™

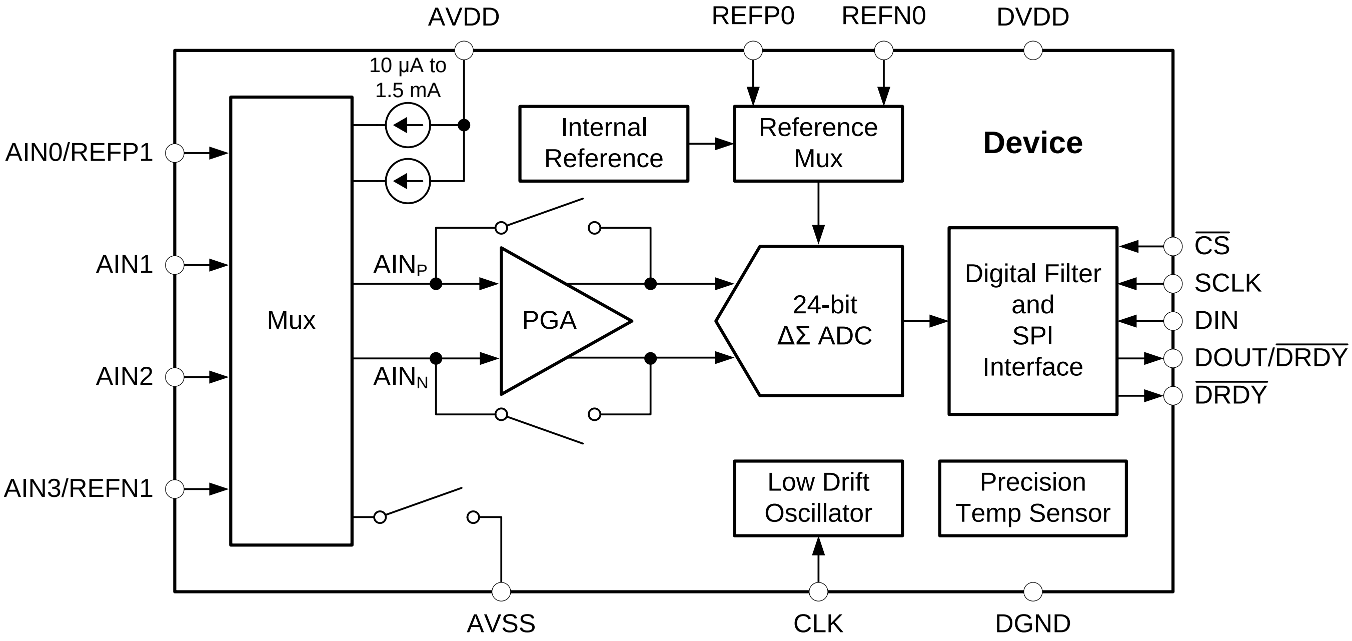


Figure 11. ADS1220 Block Diagram

This component was selected for the following reasons:

- Low current consumption: As low as 120 µA (typ) in Duty-Cycle mode
- 24-bit ADC

3.2 TI BoosterPack

When designing a BoosterPack, consult the latest design guidelines [5], which include electrical specifications as well as template files in the main design tools format [6].

3.3 MSP430FR5969 LaunchPad: MSP-EXP430FR5969

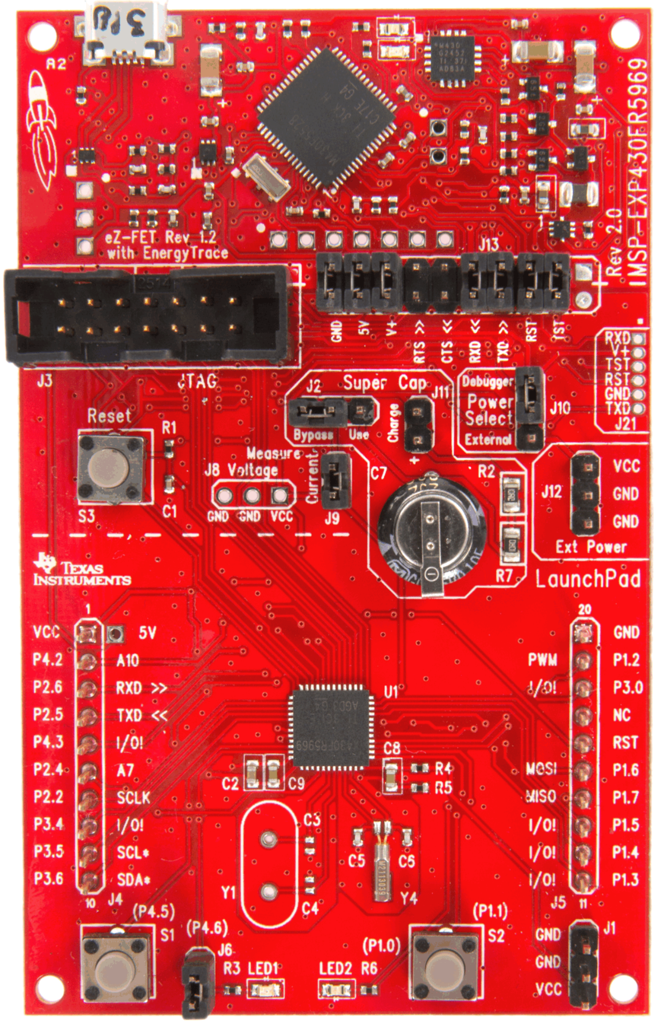


Figure 12. MSP-EXP430FR5969

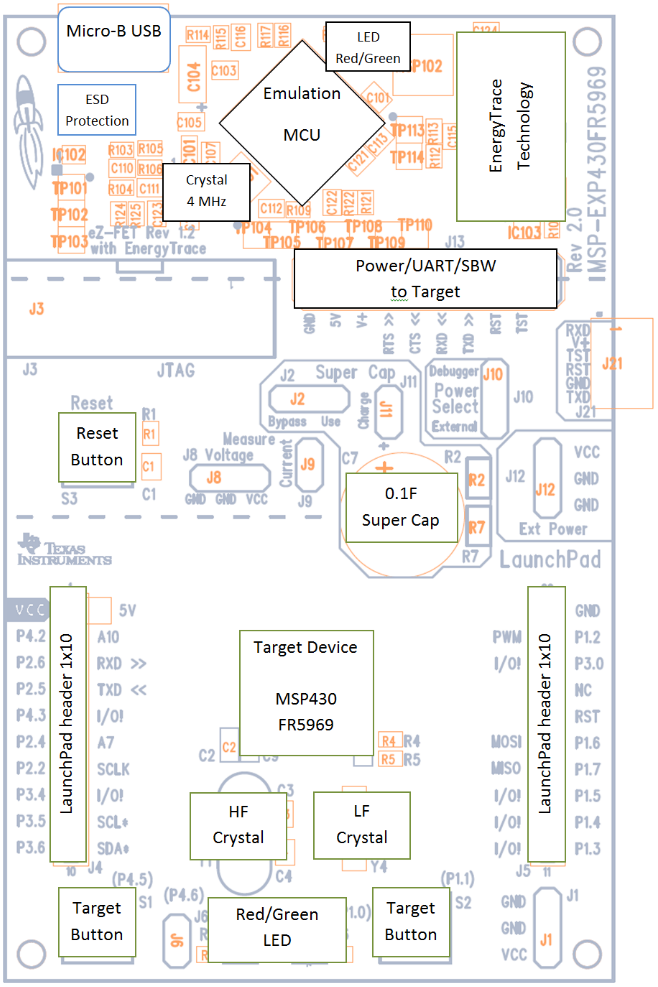


Figure 13. EVM Overview

3.3.1 Introduction

MSP430™ ultra-low-power (ULP) MCUs with embedded Ferroelectric Random Access Memory (FRAM) technology now join the MCU LaunchPad Development Kit ecosystem. The MSP-EXP430FR5969 (or the "FR5969 LaunchPad") is an easy-to-use evaluation module (EVM) for the MSP430FR5969 microcontroller. It contains everything needed to start developing on the MSP430 FRAM platform, including on-board emulation for programming, debugging, and energy measurements. The board features buttons and LEDs for quick integration of a simple user interface as well as a super capacitor (super cap) that enables standalone applications without an external power supply.

3.3.2 Key Features

- MSP430 ultra-low-power FRAM technology based MSP430FR5969 16-bit MCU
- 20-pin LaunchPad standard that leverages the BoosterPack ecosystem
- 0.1-F super capacitor for standalone power
- Onboard eZ-FET emulation with EnergyTrace++™ Technology
- Two buttons and two LEDs for user interaction
- Backchannel UART through USB to PC

3.4 CC3200 LaunchPad: CC3200-LAUNCHXL

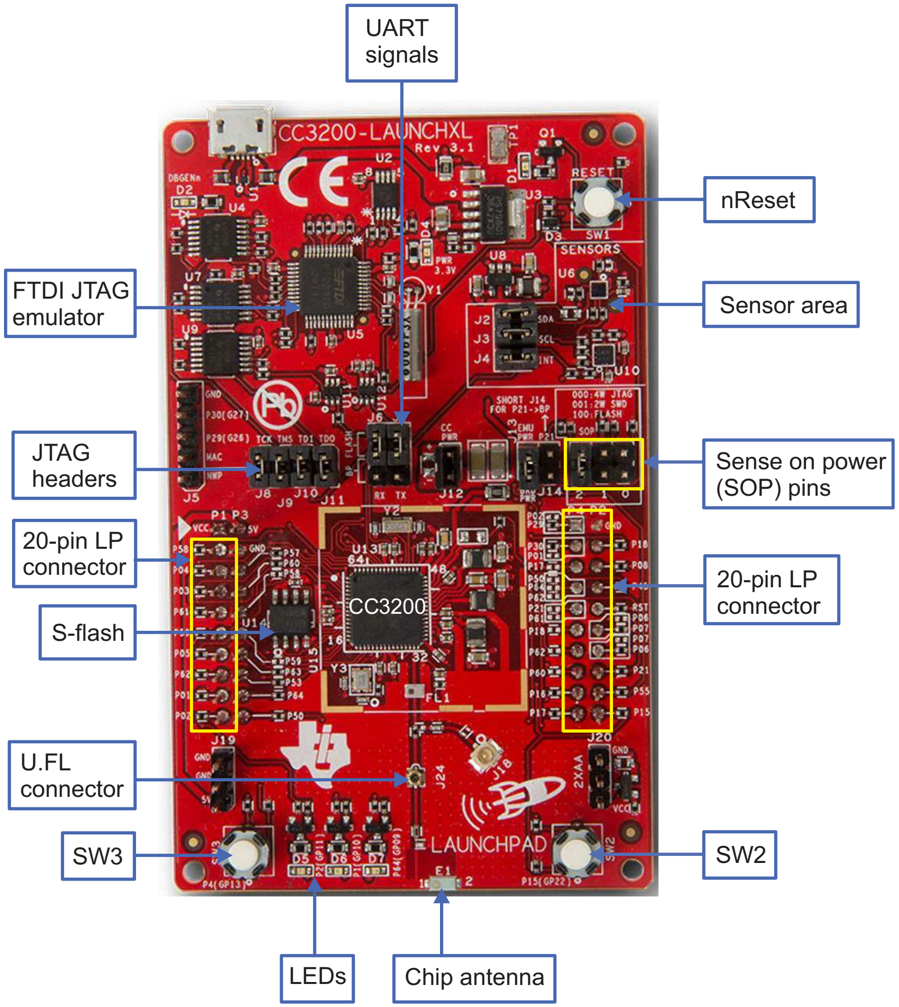


Figure 14. CC3200 LaunchPad EVM Overview

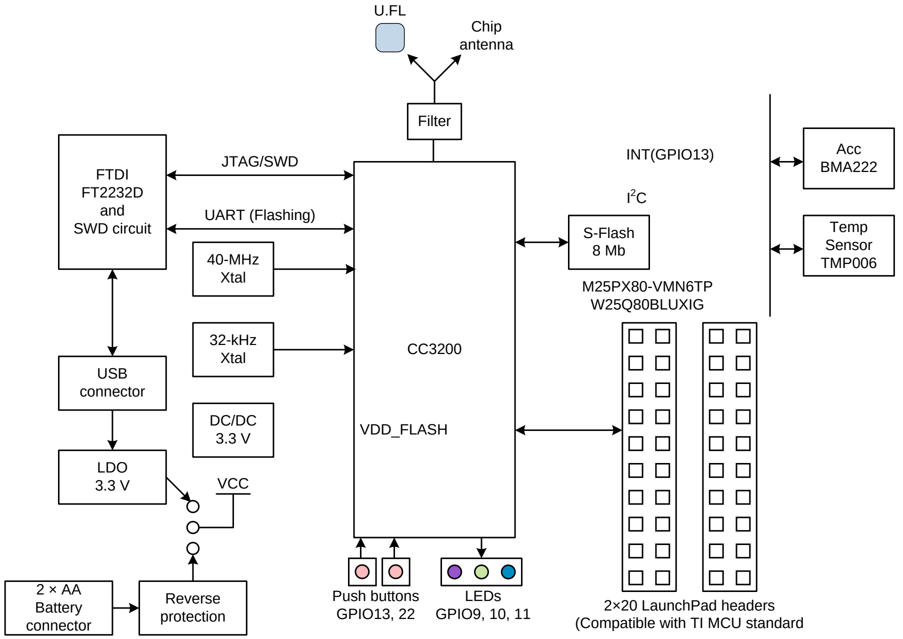


Figure 15. CC3200 LaunchPad EVM Diagram

3.4.1 Introduction

The high performance CC3200 is the industry's first single-chip Microcontroller (MCU) with built-in Wi-Fi connectivity for the LaunchPad ecosystem. Created for the Internet of Things (IoT), the SimpleLink Wi- Fi CC3200 device is a wireless MCU that integrates a high-performance ARM® Cortex®-M4 MCU allowing customers to develop an entire application with a single IC. With on-chip Wi-Fi, internet and robust security protocols, no prior Wi-Fi experience is needed for faster development.

The CC3200 LaunchPad is a low-cost evaluation platform for ARM® Cortex™-M4F-based microcontrollers. The LaunchPad design highlights the CC3200 Internet-on-a-chip™ solution and WiFi capabilities. The CC3200 LaunchPad also features programmable user buttons, RGB LED for custom applications and onboard emulation for debugging. The stackable headers of the CC3200 LaunchPad XL interface demonstrate how easy it is to expand the functionality of the LaunchPad when interfacing with other peripherals on many existing BoosterPack add-on boards such as graphical displays, audio codec, antenna selection, environmental sensing, and much more. Figure 1 shows a photo of the CC3200 LaunchPad.

Free software development tools are also available, including TI’s Eclipse-based CCS and IAR Embedded Workbench®. More information about the LaunchPad, the supported BoosterPacks, and the available resources can be found at TI’s LaunchPad portal. Also visit the CC3200 Wiki page for design resources and example projects.

NOTE: The antennas used for this transmitter must be installed to provide a separation distance of at least 20 cm from all persons and must not be co-located or operating in conjunction with any other antenna or transmitter

NOTE: All figures and references in this document apply to the Rev3.2. Most of the document also applies to the Rev4.1, unless otherwise stated.

3.4.2 Key Features

- CC3200, SimpleLink Wi-Fi, internet-on-a-chip™ solution with integrated MCU
- 40-pin LaunchPad standard that leverages the BoosterPack ecosystem
- FTDI based JTAG emulation with serial port for Flash programming
- Two buttons and three LEDs for user interaction
- Back-channel universal asynchronous receiver/transmitter (UART) through USB to PC
- Onboard chip antenna with U.FL for conducted testing
- On-board accelerometer and temperature sensor for out-of-box demo
- Micro USB connector for power and debug connections

4 Design Considerations

This section details the hardware and software design considerations.

4.1 *Hardware*

4.1.1 ADS1220 Design Notes

4.1.1.1 ADC Operation (3-Wire RTD Measurement)

The first heart of this reference design is ADS1220. ADS1220 integrates all required features (such as dual-matched programmable current sources, buffered reference inputs, and PGA) to ease the implementation of ratiometric 2-, 3-, and 4-wire RTD measurements.

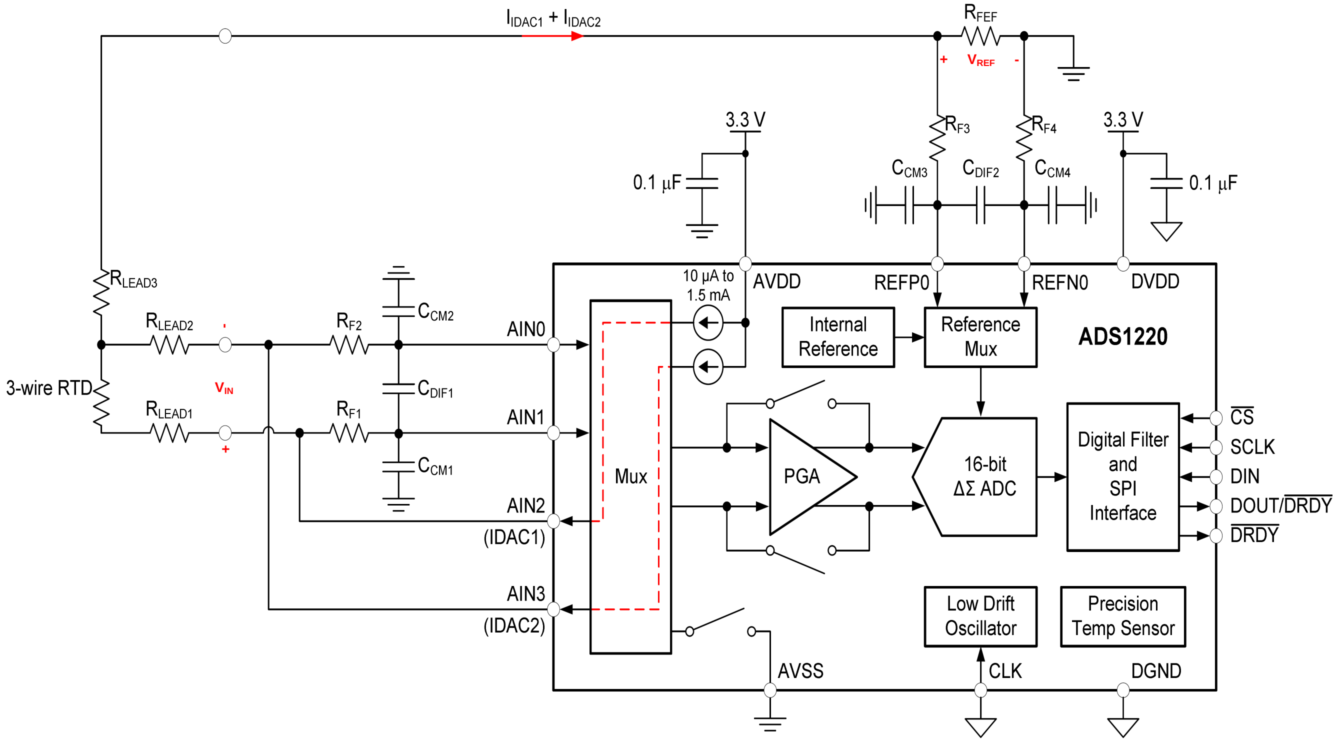


Figure 16. 3-Wire RTD Measurement

The circuit in Figure 16 employs a ratiometric measurement approach. In other words, the sensor signal (the voltage across the RTD) and the reference voltage for the ADC are derived from the same excitation source. Therefore, errors resulting from temperature drift or noise cancel out because these errors are common to both the sensor signal and the reference.

In order to implement a ratiometric 3-wire RTD measurement, ADS1220 provides two integrated digital-analog converter (IDAC) current sources capable of outputting currents from 10 µA to 1.5 mA. IDAC1 is routed to one of the excitation leads of the RTD. IDAC2 is routed to the second excitation lead as shown in Figure 15 by appropriate setting of I1MUX [2:0]: IDAC1 routing configuration bits and I2MUX [2:0]: IDAC2 routing configuration bits in the configuration register 3. Both currents have the same value, which is programmable by bits IDAC [2:0]: IDAC current setting in configuration register 2.

The design of the ADS1220 ensures that both IDAC values are closely matched, even across temperature. The excitation current of 1 mA or less is generally preferred for industrial applications in order to minimize the error due to self-heating. In this application, IDAC1 and IDAC2 programmed for 250 µA.

The sum of both currents, which is 500 µA, flows through a reference resistor, R_{REF}. The voltage developed across RREF is fed into the positive and negative reference pins (REFP0 and REFN0) of the ADS1220 which is configured to use this external reference voltage V_{REF} for the analog-to-digital conversions. The R_{REF} resistor and excitation current are sized to produce an external reference of mid-supply (AVDD + AVSS) / 2.

$$\text{ADC CODE } \alpha \frac{V_{RTD} \text{ x } PGA_{GAIN}}{V_{REF}} \tag{1}$$

$$\text{ADC CODE } \alpha \frac{R_{RTD} \text{ x } PGA_{GAIN}}{2 \times R_{REF}} \tag{2}$$

From Equation 2, the accuracy of the measurement depends on the reference resistor. Therefore, the absolute accuracy and temperature drift of the excitation current does not matter. However, because the value of the reference resistor directly impacts the measurement result, the accuracy can be set to a higher level by choosing a high-precision, low-drift reference resistor R_{REF}.

4.1.1.1.1 Selecting R_{REF}

$$V_{REF} = (IDAC1 + IDAC2) \times R_{REF} \tag{3}$$

Because, IDAC1 = IDAC2 = IDAC = 250 µA

$$IDAC1 = IDAC2 = IDAC = 250\ \mu A \tag{4}$$

If R_{REF} = 3.24 kΩ, ±0.1% tolerance or better, 25 PPM/°C drift or better, then

$$V_{REF} = 2 \times IDAC \times R_{REF} = 2 \times 250\ \mu A \times 3.24\ k\Omega = 1.62\ V \tag{5}$$

which is close to mid-supply voltage.

4.1.1.1.2 Selecting PGA Gain

The voltage produced across the RTD needs to be amplified by the PGA before it reaches ΔΣ ADC for conversion. The PGA gain should be chosen so that ADC input signal is still less than V_{REF} at maximum RTD temperature. The largest possible gain will yield the best resolution per °C.

R_{RTD} at 850°C = 390.48 Ω,

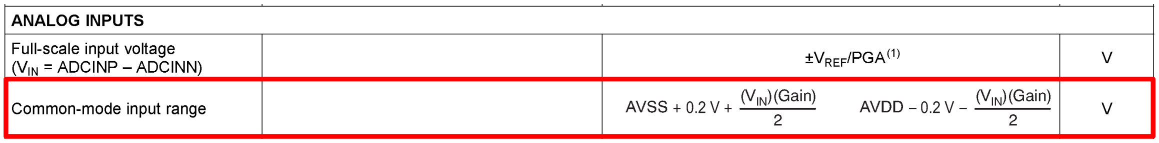
V_{RTD} at 850°C = 390.48 Ω × 250 µA = 97.62 mV,

$$PGA\_GAIN = \frac{V_{REF}}{V_{RTD} \text{ at } 850°C} = \frac{1.62\ V}{97.62\ mV} = 16.6\ V / V \tag{6}$$

The closest programmable gain option is 16 V/V.

4.1.1.1.3 Common-Mode Voltage Compliance Check

The allowed common-mode input voltage range is as highlighted in Figure 17.



ANALOG INPUTS				
Full-scale input voltage (V_{IN} = ADCINP – ADCINN)		±V_{REF}/PGA[1]		V
Common-mode input range		$AVSS + 0.2\,V + \frac{(V_{IN})(Gain)}{2}$	$AVDD - 0.2\,V - \frac{(V_{IN})(Gain)}{2}$	V

Figure 17. Specification for Common-Mode Input Range

$$0\,V + 0.2 + \frac{0.09762\,V \times 16\,V/V}{2} \le V_{CM} \le 3.3\,V - 0.2\,V - \frac{0.09762\,V \times 16\,V/V}{2} \tag{7}$$

$0.981\,V \le V_{CM} \le 2.32\,V$

The common-mode input voltage actually set by the design can be given as shown in Equation 8.

$$V_{CMI} = (IDAC \times R_{LEAD}) + \frac{IDAC \times R_{RTD}}{2} + 2 \times IDAC \times (R_{LEAD} + R_{REF}) \tag{8}$$

Assuming R_{LEAD} = 10 Ω,

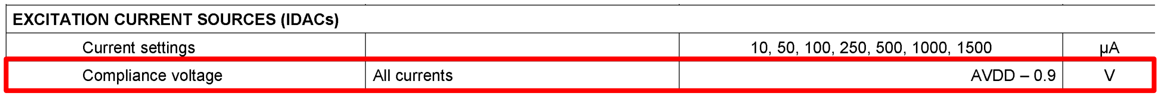
V_{CMI} at R_{RTD} at –200°C ≤ VCMI ≤ V_{CMI} at R_{RTD} at 850°C (9)

$1.63\,V \le V_{CMI} \le 1.68\,V$

This result is well within the maximum allowed common-mode input voltage range.

4.1.1.1.4 Excitation Current Compliance Voltage Check

When designing the circuit, care should be taken to meet the compliance-voltage requirement of the IDACs. The IDACs require a minimum headroom of (AVDD – 0.9 V) in order to operate accurately.



EXCITATION CURRENT SOURCES (IDACs)			
Current settings		10, 50, 100, 250, 500, 1000, 1500	µA
Compliance voltage	All currents	AVDD – 0.9	V

Figure 18. Specification for Excitation Current Source Compliance Voltage

For IDAC1:

$AVSS + [IDAC1 \times (R_{LEAD1} + R_{RTD})] + [(IDAC1+IDAC2) \times (R_{LEAD3} + R_{REF})] + V_{DIODE} \le AVDD - 0.9\,V$ (10)

$2.12512\,V \le 2.4\,V$

For IDAC2:

$AVSS + [IDAC2 \times (R_{LEAD2} + R_{RTD})] + [(IDAC1+IDAC2) \times (R_{LEAD3} + R_{REF})] + V_{DIODE} \le AVDD - 0.9\,V$ (11)

$1.628\,V \le 2.4\,V$

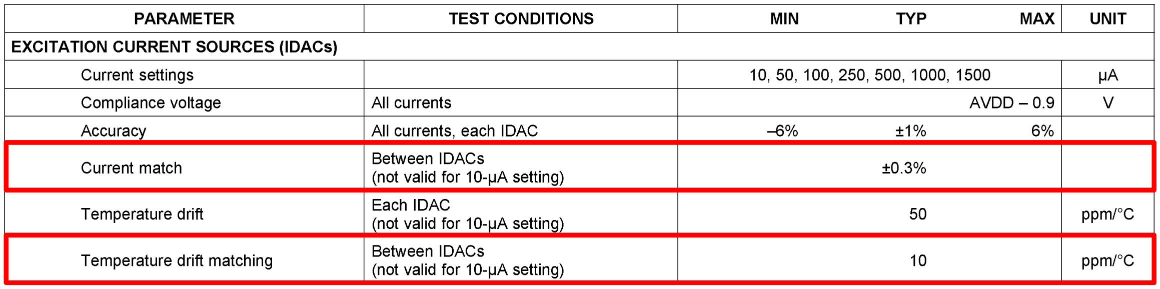
This means that the excitation current compliance voltages for IDAC1 and IDAC2 are met.

4.1.1.1.5 Analog Input and Reference Low-Pass Filter Design

RTD voltage output signal is typically in the millivolt range, which makes the signal susceptible to noise. First-order differential and common-mode RC filters (R_{F1}, R_{F2}, C_{DIF1}, C_{CM1}, and C_{CM2}) are placed on the ADC inputs, as well as on the reference inputs (R_{F3}, R_{F4}, C_{DIF2}, C_{CM3}, and, C_{CM4}) to eliminate high-frequency noise in RTD measurements. The differential cut-off frequency of the analog-input filter must be chosen at least ten times higher than the ADC bandwidth at the selected data rate. The cut-off frequency chosen for this design is higher to account for a faster sampling rate. If the corner frequencies are not matched, noise appearing on analog input will be different from noise on the reference, which will not get cancelled in ratiometric measurement. For best performance, it is recommended to match the corner frequencies of the analog-input and reference low-pass filters. For more details on how to match corner frequencies of analog input and reference filters, refer to the application report *RTD Ratiometric Measurements and Filtering Using the ADS1148 and ADS1248 Family of Devices* (SBAA201).

4.1.1.1.6 Excitation Current Chopping

The two current sources must be perfectly matched to successfully cancel the lead resistance of the RTD wires. While initial matching of the current sources is important, any remaining mismatch in the two sources can be minimized by swapping (or chopping) the two current sources between the two inputs. Taking a measurement in both configurations and averaging the two readings will greatly reduce the effects of mismatched current sources (see Equation 12). Realizing current chopping technique could become possible by the use of an integrated, digitally-controlled, very-flexible input multiplexer inside ADS1220.



PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
EXCITATION CURRENT SOURCES (IDACs)					
Current settings			10, 50, 100, 250, 500, 1000, 1500		µA
Compliance voltage	All currents			AVDD – 0.9	V
Accuracy	All currents, each IDAC	–6%	±1%	6%	
Current match	Between IDACs (not valid for 10-µA setting)		±0.3%		
Temperature drift	Each IDAC (not valid for 10-µA setting)		50		ppm/°C
Temperature drift matching	Between IDACs (not valid for 10-µA setting)		10		ppm/°C

Figure 19. Specification for Excitation Current Source Mismatch and Temperature Drift

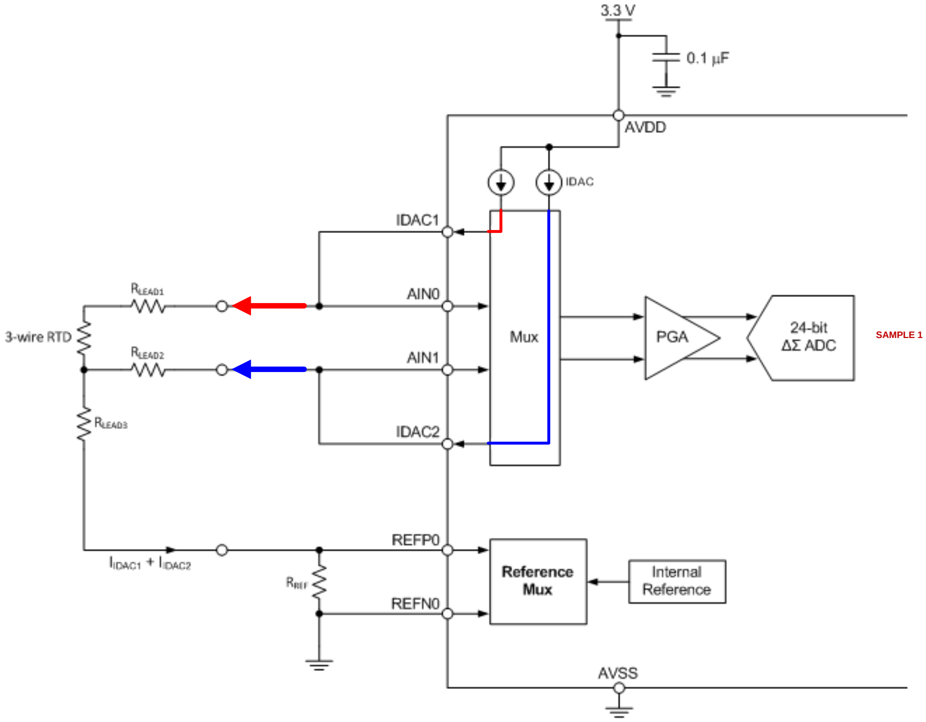


Figure 20. Measurement Before Excitation Current Swap

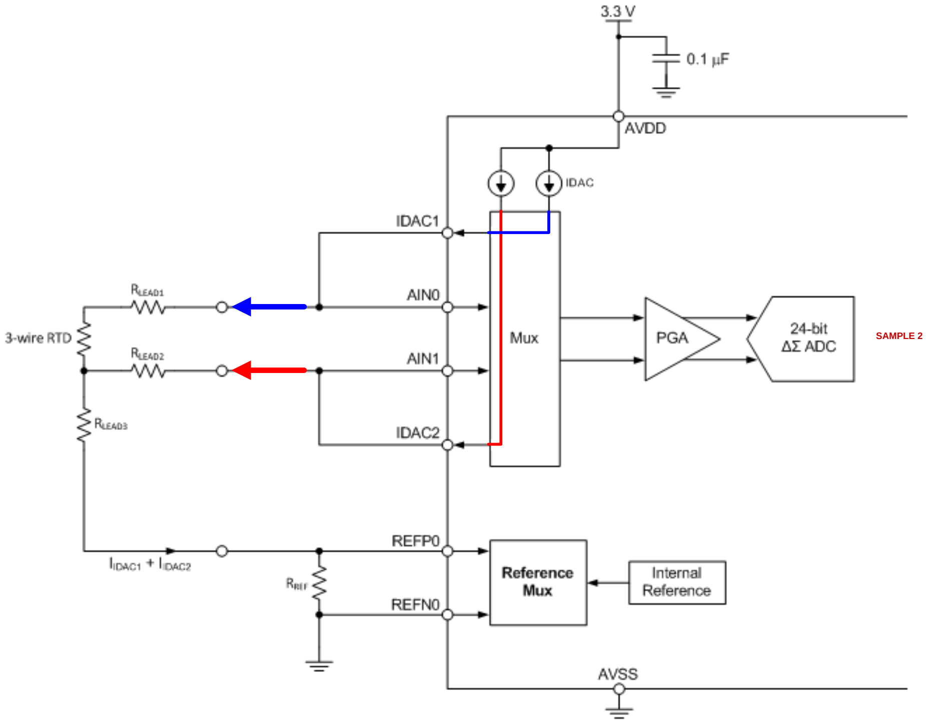


Figure 21. Measurement After Excitation Current Swap

$$\text{IDAC Mismatch Corrected Reading} = \frac{\text{SAMPLE 1} + \text{SAMPLE 2}}{2} \qquad (12)$$

4.1.2 BoosterPack Design Notes

The BoosterPack is designed with connectors whose numbering is different than the BoosterPack documentation; Figure 22 is an extract from the schematics (on the left the board schematics and on the right the connector with the signals highlighting the associated BoosterPack signals).

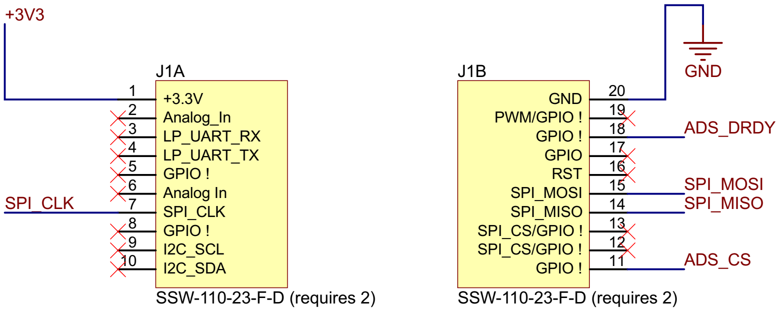


Figure 22. BoosterPack Connections

4.2 Software

4.2.1 MSP-EXP430FR5969

When connecting a BoosterPack to a LaunchPad, the key point is to identify which signals are used:

Section 4.1.2 showed that the ADS1220 chip select is on BP11, consult the LaunchPad to see which pin is connected and then define it in the sketch as:

```
#define RDY P3_0
#define CS P1_3
```

To run in continuous mode, map an interrupt to the RDY pin:

```
    //Configure the microcontroller GPIO connected to the DRDY pin as a falling
edge triggered interrupt input;
    pinMode(RDY, INPUT_PULLUP);
    attachInterrupt(RDY, counting, FALLING); // Interrupt is fired whenever button
is pressed
```

To define a global flag as well as the interrupt handler:

```
volatile int
count = 0;
…
void counting()
{
  count+=1;
}
```

For the initialization, put the pseudo-code from the ADS1220 datasheet in the setup function of the sketch (see the source code in the referenced zip file). Once the initialization is over, the sketch will run forever in the loop function.

The main code will loop until the interrupt handler has fired and reads the data. Once the data is read, it will display through UART over USB on the PC console. The library includes two small functions as an example of how to read the resistor or the internal temp sensor (ADS1220_next_read_internal_temp and ADS1220_restore_after_internal_temp_read, respectively).

For more options on how to leverage this generic ΣΔ platform, see Section 5.

5 Getting Started Hardware

This section describes how to get started with EVM hardware.

5.1 Connect the ADS1220 BoosterPack to MSP430FR5969 LaunchPad

To connect the generic SD BoosterPack, ensure that the connector index match on the LaunchPad and the BoosterPack.

Once connected:

1. Install Energia if not done it yet (see Section 11.1).
2. Connect the micro-USB of the LaunchPad to the PC.
3. Ensure in the Windows® Device Manager (or alternative naming for other OS) that the board is recognized.
4. Load one of the demo sketches from Section 6.
5. Open HyperTerminal and monitor that the displayed values match expectations.

5.2 Application Specific Details—Universal Temperature Input

This section details how to use the generic SD ADC BoosterPack into a universal temperature input solution. For more details on the setup for universal input, see Section 7.5.

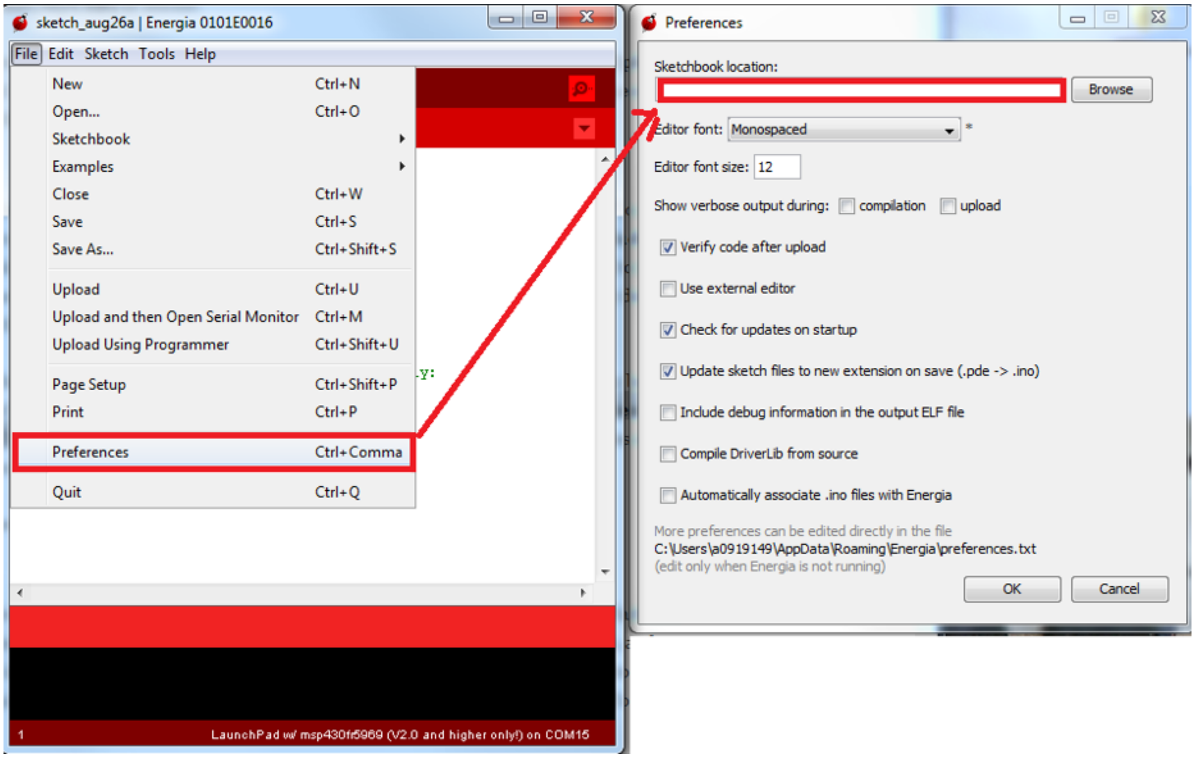
6 Getting Started Firmware

This section describes how to get started with the provided firmware.

6.1 Generic Setup

To install the ADS1220 library, create a directory called libraries within the sketchbook directory. Then unzip the library there.

The "sketchbook" directory is configured by click on File → Preferences. Then select the "Sketchbook location" field.


sketch_aug26a | Energia 0101E0016
File Edit Sketch Tools Help
New Ctrl+N
Open... Ctrl+O
Sketchbook
Examples
Close Ctrl+W
Save Ctrl+S
Save As... Ctrl+Shift+S
Upload Ctrl+U
Upload and then Open Serial Monitor Ctrl+M
Upload Using Programmer Ctrl+Shift+U
Page Setup Ctrl+Shift+P
Print Ctrl+P
Preferences Ctrl+Comma
Quit Ctrl+Q
1
LaunchPad w/ msp430fr5969 (V2.0 and higher only!) on COM15
Preferences
Browse
OK
Cancel

Sketchbook location:

Editor font: Monospaced *

Editor font size: 12

Show verbose output during: □ compilation □ upload

☑ Verify code after upload

□ Use external editor

☑ Check for updates on startup

☑ Update sketch files to new extension on save (.pde -> .ino)

□ Include debug information in the output ELF file

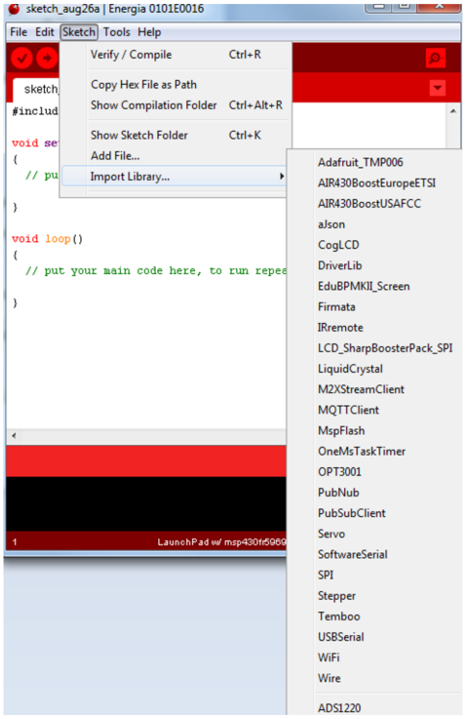
□ Compile DriverLib from source

□ Automatically associate .ino files with Energia

More preferences can be edited directly in the file
C:\Users\a0919149\AppData\Roaming\Energia\preferences.txt
(edit only when Energia is not running)

Figure 23. Energia Initial Configuration

Once installed, the library is visible from Sketch → Import Library → ADS1220.



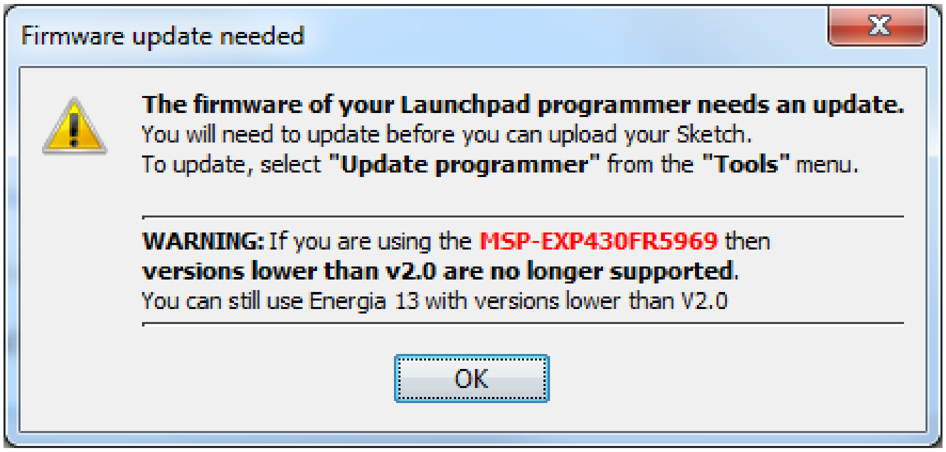
```
#includ

void se
{
  // pu

}

void loop()
{
  // put your main code here, to run repee

}
```

Figure 24. Adding ADS1220 Library to the Sketch

6.2 Error Messages

6.2.1 Firmware Update Needed

Figure 25 appears when the LaunchPad firmware needs an update.



Firmware update needed

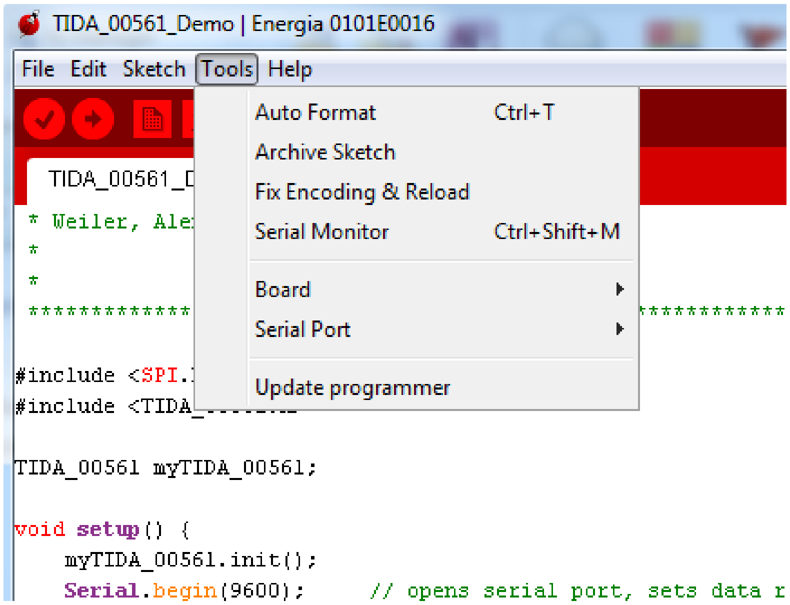
The firmware of your Launchpad programmer needs an update.
You will need to update before you can upload your Sketch.
To update, select **"Update programmer"** from the **"Tools"** menu.

WARNING: If you are using the MSP-EXP430FR5969 then
versions lower than v2.0 are no longer supported.
You can still use Energia 13 with versions lower than V2.0

Figure 25. Possible Error Message Requiring a Firmware Upgrade

This notice means the firmware in the debug interface of the LaunchPad does not have a version matching the version that Energia expects.

1. Ensure the latest Energia version is installed (recommended not mandatory)
2. Proceed with the update of debug firmware:
   (a) Click "OK" on the window.
   (b) Go to the *Tools* menu and update the programmer.



TIDA_00561_Demo | Energia 0101E0016

```
 * Weiler, Ale
 *
 *
 *************                                    ************

#include <SPI.]
#include <TIDA_

TIDA_00561 myTIDA_00561;

void setup() {
    myTIDA_00561.init();
    Serial.begin(9600);      // opens serial port, sets data r
```

Figure 26. Configuring Energia's Serial Port

3. The Energia console will then say "Updating programmer (this may take a minute)…"
4. Once the console says "Done updating programmer…", start again from where you were (click on button *Upload*).

7 Test Setup and Data — Universal Temperature Input

This section covers how to use the Energia library in conjunction with the ADS1220 BoosterPack to build a universal temperature input by walking through the different temperature inputs and then showing how to merge them all in one universal temperature input fast prototyping solution.

7.1 2-Wire RTD Setup

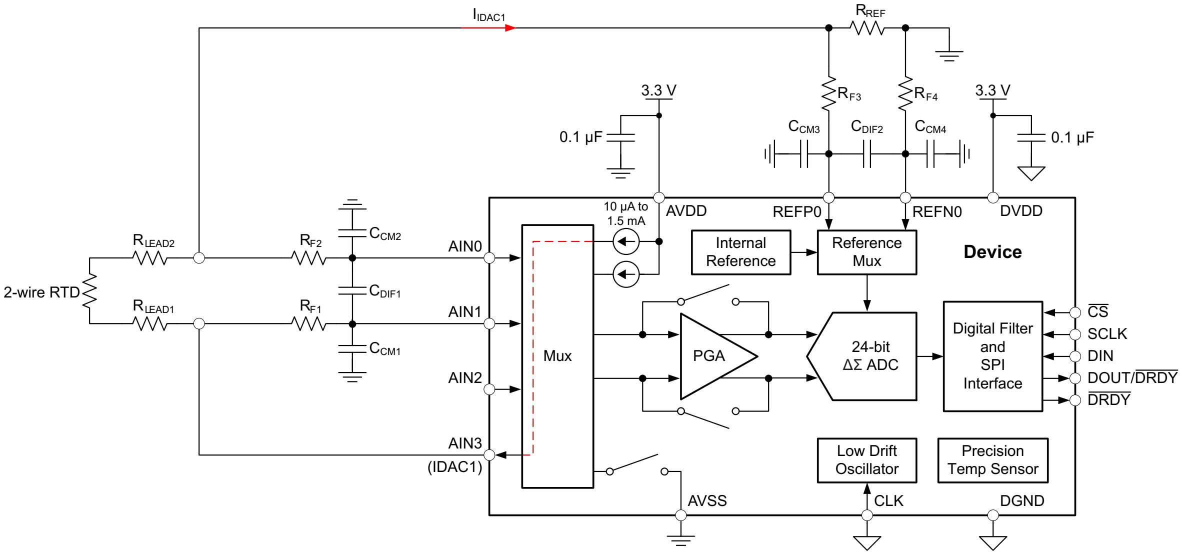


Figure 27. 2-Wire RTD Setup

To enable the 2-wire RTD setup, the ADS1220 needs to be configured as the following Energia sketch (select the library example ADS1220_2wire):

```
myADS1220_647.ADS1220_initfull(ADS1220_MUX_AIN0_AIN1, ADS1220_GAIN_16,
ADS1220_USE_PGA,
                ADS1220_DATA_RATE_20SPS, ADS1220_OP_MODE_NORMAL,
                ADS1220_CONVERSION_CONTINUOUS,  ADS1220_TEMP_SENSOR_OFF,
                ADS1220_BURN_OUT_CURRENT_OFF,
                ADS1220_FIR_50_60, ADS1220_VREF_EXT_REF0_PINS,
                ADS1220_LOW_SIDE_POWER_OPEN, ADS1220_IDAC_CURRENT_250_UA,
                ADS1220_IDAC1_AIN3, ADS1220_IDAC2_AIN2, ADS1220_DRDY_ON_DRDY_ONLY);
```

`ADS1220_MUX_AIN0_AIN1:` selects the input AIN0 and AIN1

`ADS1220_USE_PGA:` uses the PGA

`ADS1220_GAIN_16:` PGA gain of 16

`ADS1220_IDAC_CURRENT_250_UA:` IDAC set for 250 µA

`ADS1220_IDAC1_AIN3:` IDAC1 is MUXED on AIN3

7.2 3-Wire RTD Setup

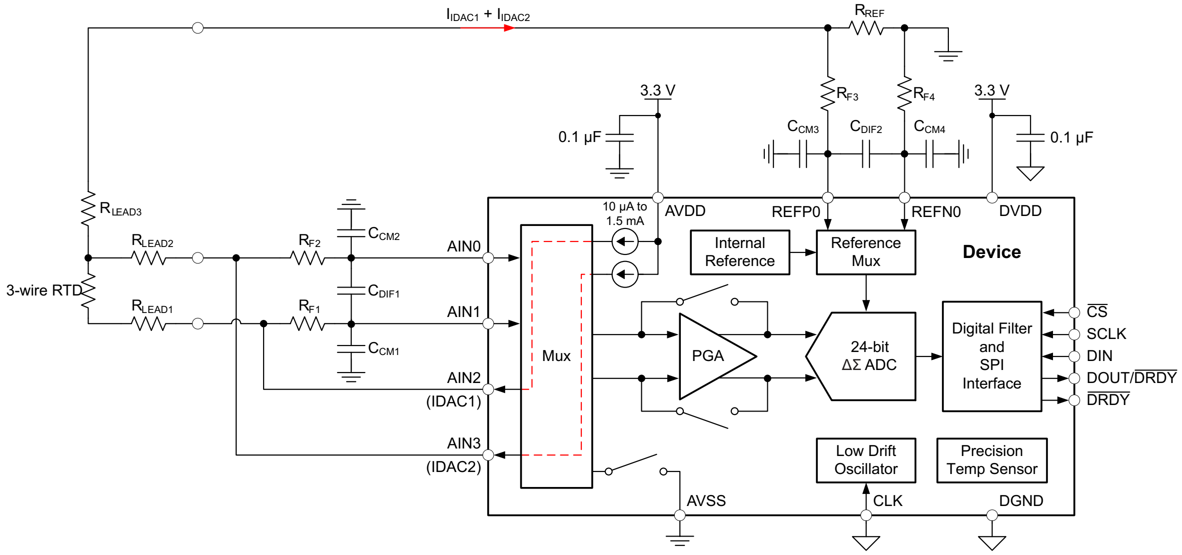


Figure 28. 3-Wire RTD Setup

To enable the 3-wire RTD setup, the ADS1220 needs to be configured as the following Energia sketch (select the library example ADS1220_3wires):

```
myADS1220_647.ADS1220_initfull(ADS1220_MUX_AIN0_AIN1, ADS1220_GAIN_16,
ADS1220_USE_PGA,
                    ADS1220_DATA_RATE_20SPS, ADS1220_OP_MODE_NORMAL,
                    ADS1220_CONVERSION_CONTINUOUS,  ADS1220_TEMP_SENSOR_OFF,
                    ADS1220_BURN_OUT_CURRENT_OFF,
                    ADS1220_FIR_50_60, ADS1220_VREF_EXT_REF0_PINS,
                    ADS1220_LOW_SIDE_POWER_OPEN, ADS1220_IDAC_CURRENT_250_UA,
                    ADS1220_IDAC1_AIN3, ADS1220_IDAC2_AIN2, ADS1220_DRDY_ON_DRDY_ONLY);
```

ADS1220_MUX_AIN0_AIN1: selects the input AIN0 and AIN1

ADS1220_USE_PGA: uses the PGA

ADS1220_GAIN_16: PGA gain of 16

ADS1220_IDAC_CURRENT_250_UA: IDAC set for 250 µA

ADS1220_IDAC1_AIN3: IDAC1 is MUXED on AIN3

ADS1220_IDAC2_AIN2: IDAC1 is MUXED on AIN2 (this allows to compensate for the wire lead in 3-wire mode)

7.3 4-Wire RTD Setup

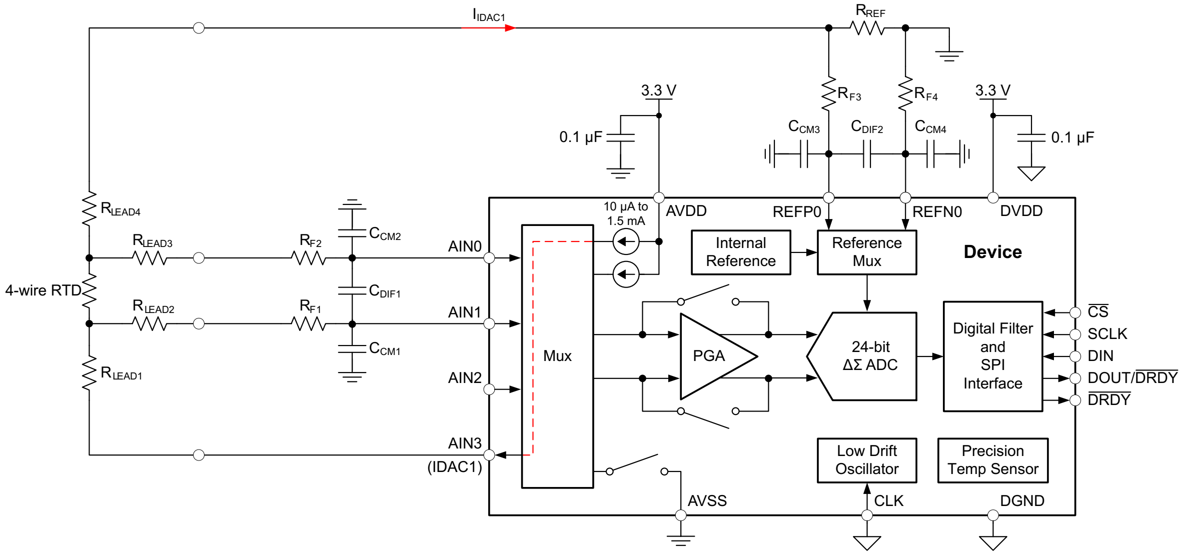


Figure 29. 4-Wire RTD Setup

To enable the 4-wire RTD setup, the ADS1220 needs to be configured as the following Energia sketch (select the library example ADS1220_4wire):

```
myADS1220_647.ADS1220_initfull(ADS1220_MUX_AIN0_AIN1, ADS1220_GAIN_16,
ADS1220_USE_PGA,
                ADS1220_DATA_RATE_20SPS, ADS1220_OP_MODE_NORMAL,
                ADS1220_CONVERSION_CONTINUOUS,  ADS1220_TEMP_SENSOR_OFF,
                ADS1220_BURN_OUT_CURRENT_OFF,
                ADS1220_FIR_50_60, ADS1220_VREF_EXT_REF0_PINS,
                ADS1220_LOW_SIDE_POWER_OPEN, ADS1220_IDAC_CURRENT_250_UA,
                ADS1220_IDAC1_AIN3, ADS1220_IDAC2_DISABLED,
                ADS1220_DRDY_ON_DRDY_ONLY);
```

ADS1220_MUX_AIN0_AIN1: selects the input AIN0 and AIN1
ADS1220_USE_PGA: uses the PGA
ADS1220_GAIN_16: PGA gain of 16
ADS1220_IDAC_CURRENT_250_UA: IDAC set for 250 µA
ADS1220_IDAC1_AIN3: IDAC1 is MUXED on AIN3
ADS1220_IDAC2_DISABLED: IDAC2 disabled

7.4 Thermocouple Setup (Type K)

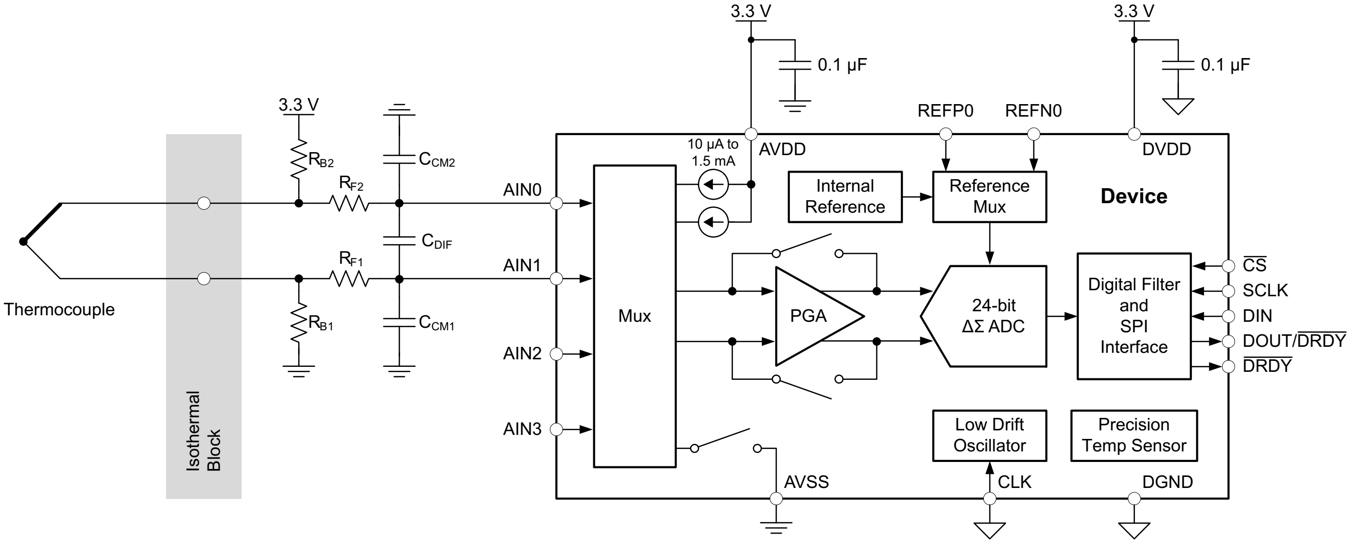


Figure 30. Thermocouple Setup

To enable the TC setup, the ADS1220 needs to be configured as the following Energia sketch (select the library example ADS1220_TC):

```
myADS1220_647.ADS1220_initfull(ADS1220_MUX_AIN0_AIN1, ADS1220_GAIN_32,
ADS1220_USE_PGA,
                          ADS1220_DATA_RATE_20SPS, ADS1220_OP_MODE_NORMAL,
                          ADS1220_CONVERSION_CONTINUOUS,  ADS1220_TEMP_SENSOR_ON,
                          ADS1220_BURN_OUT_CURRENT_OFF,
                          ADS1220_FIR_50_60, ADS1220_VREF_AVDDADS1220_LOW_SIDE_POWER_OPEN,
                          ADS1220_IDAC_CURRENT_OFF,
                          ADS1220_IDAC1_DISABLED, ADS1220_IDAC2_DISABLED,
                          ADS1220_DRDY_ON_DRDY_ONLY);
```

ADS1220_MUX_AIN0_AIN1: selects the input AIN0 and AIN1
ADS1220_USE_PGA: uses the PGA
ADS1220_GAIN_16: PGA gain of 32
ADS1220_IDAC_CURRENT_OFF: IDAC disabled
ADS1220_IDAC1_ DISABLED: IDAC1 is MUX is left open
ADS1220_IDAC2_ DISABLED: IDAC1 is MUX is left open
ADS1220_VREF_AVDD: uses the internal reference
ADS1220_TEMP_SENSOR_ON: uses the internal temp sensor for the CJC

In this setup, being able to bias the PGA input common mode voltage is critical; however, the ADS1220 does not have a built-in feature for this.

Due to the rapid prototyping approach of this platform, use some of the GPIO of the MCU on the LaunchPad to provide the bias by wiring them to the connector J4 and J3:

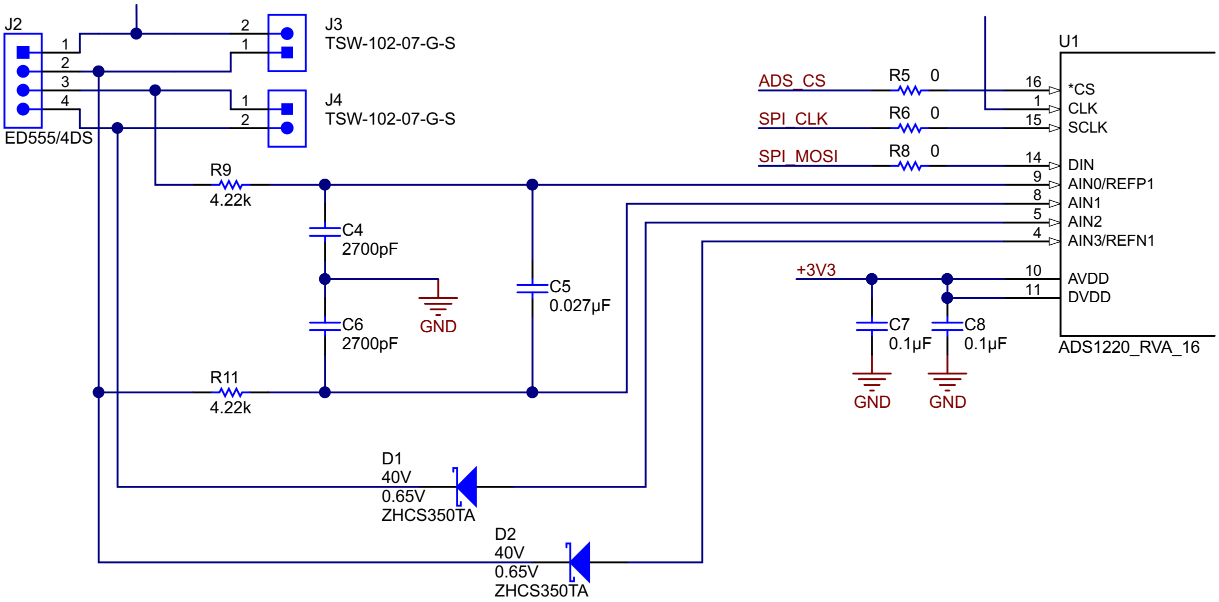


Figure 31. Connections for ADS1220 Pins to Screw Terminals

One of the GPIO should be connected through 2 MΩ to J2_1 and turned to high:

```
int VDD_pull = **;                // select here the pin on the BP header
int GND_pull = ****;                // select here the pin on the BP header which
you want to use to connect the thermocouple to GND through 2MR

(in your setup function)
pinMode(VDD_pull, OUTPUT);               // configure the pin to digital output
digitalWrite(VDD_pull, HIGH);   // set the pin to output VDD

pinMode(GND_pull, OUTPUT);               // configure the pin to digital output
digitalWrite(GND_pull, LOW);   // set the pin to output VDD
```

7.5 Universal Temperature Input

From the previous sections, it becomes possible to build a universal temperature input hardware fast prototyping solution.

Make sure that in the RTD mode the common mode input voltage bias needed by the thermocouple is not provided. For this, turn the digital output of the microcontroller into Hi-Z.

Using MSP-EXP430FR5969 as a LaunchPad, the analog inputs of the comparator can be configured in Hi-Z inputs (trying to configure digital inputs in Hi-Z will lead to excessive power consumption). This can be done in two ways:

1. Create a "Comparator" library with Energia like API’s.
2. Use direct register access like in the MSP430 code examples.

For example:

```
void setup()
{
  // Configure GPIO
  P3DIR |= BIT5;                            // P3.5 output direction
  P3SEL1 |= BIT5;                           // Select CEOUT function on P3.5/CEOUT

  // Disable the GPIO power-on default high-impedance mode to activate
  // previously configured port settings
  PM5CTL0 &= ~LOCKLPM5;

  // Setup Comparator_E
  CECTL0 = CEIPEN | CEIPSEL_1;              // Enable V+, input channel CE1
  CECTL1 = CEPWRMD_1;                       // normal power mode
  CECTL2 = CEREFL_2 | CERS_3 | CERSEL;      // VREF is applied to -terminal
                                            // R-ladder off; bandgap ref voltage
                                            // supplied to ref amplifier to get
Vcref=2.0V
  CECTL3 = BIT1;                            // Input Buffer Disable @P1.1/CE1
  CECTL1 |= CEON;                           // Turn On Comparator_E

  __delay_cycles(75);                       // delay for the reference to settle

  __bis_SR_register(LPM4_bits);             // Enter LPM4
  __no_operation();
}

void loop()
{
  // put your main code here, to run repeatedly:
}
```

8 Design Files

8.1 Schematics

To download the schematics, see the design files at TIDA-00647.

8.2 Bill of Materials

To download the bill of materials (BOM), see the design files at TIDA-00647.

8.3 Layer Plots

To download the layer plots, see the design files at TIDA-00647.

8.4 Altium Project

To download the Altium project files, see the design files at TIDA-00647.

8.5 Gerber Files

To download the Gerber files, see the design files at TIDA-00647.

9 Software Files

To download the software files, see the design files at TIDA-00647.

10 References

1. Texas Instruments, *Choose the right A/D converter for your application* (PDF)
2. Texas Instruments, *SAR and Delta-Sigma ADC Fundamentals*, TI Training (https://training.ti.com/sar-and-delta-sigma-adc-fundamentals)
3. Texas Instruments, *Continuous-Time Sigma-Delta ADCs*, Application Note (SNAA098)
4. Texas Instruments, *SAR ADC's vs Delta Sigma ADC's: Different Architectures for Different Application Needs*, TI Training (PDF)
5. Texas Instruments, *Build Your Own BoosterPack*, TI LaunchPad Webpage (http://www.ti.com/ww/en/launchpad/byob.html)
6. Texas Instruments, *BoosterPack Pinout Standard* (PDF)
7. Texas Instruments, *Altium Template Files for BoosterPacks* (SLAC629)
8. Energia, *Import Energia Sketch to Code Composer Studio v6* (http://energia.nu/guide/import-energia-sketch-to-ccsv6/)

11 Annexes

11.1 Install Energia

For more details, see http://energia.nu/Guide_index.html.

For a quick starter’s guide, visit the downloads page (http://energia.nu/download/), select the binaries for the specified OS, and download.

11.2 Energia Sketch Import Feature in CCS

For more information on the Energia Sketch Import feature, see the following link: http://energia.nu/guide/import-energia-sketch-to-ccsv6/

12 Credits

The author wishes to acknowledge key contributions from colleagues that enabled this project to happen on time and in conformance with design specifications and market needs: Stefan Schauer, whose help was invaluable to connect the dots between all the different software components, and Robert Wessels, who started back in January 2012 with the goal to bring Wiring and Arduino framework to Texas Instruments’ MSP430.

It is only thanks to Stefan and Robert work and support that this fast prototyping platform could be released.

13 About the Author

MATTHIEU CHEVRIER is a systems architect at Texas Instruments, where he is responsible for defining and developing reference design solutions for the industrial segment. Matthieu brings to this role his extensive experience in embedded system designs in both hardware (power management, mixed signal, and so on) and software (such as low level drivers, RTOS, and compilers). Matthieu earned his master of science in electrical engineering (MSEE) from Supélec, an Ivy League university in France. Matthieu holds patents from IPO, EPO, and USPTO.

Revision History

Changes from Original (September 2015) to A Revision	Page
• Changed from preview page	1

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

IMPORTANT NOTICE FOR TI REFERENCE DESIGNS

Texas Instruments Incorporated ("TI") reference designs are solely intended to assist designers (“Buyers”) who are developing systems that incorporate TI semiconductor products (also referred to herein as “components”). Buyer understands and agrees that Buyer remains responsible for using its independent analysis, evaluation and judgment in designing Buyer’s systems and products.

TI reference designs have been created using standard laboratory conditions and engineering practices. **TI has not conducted any testing other than that specifically described in the published documentation for a particular reference design.** TI may make corrections, enhancements, improvements and other changes to its reference designs.

Buyers are authorized to use TI reference designs with the TI component(s) identified in each particular reference design and to modify the reference design in the development of their end products. HOWEVER, NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY THIRD PARTY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT, IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI REFERENCE DESIGNS ARE PROVIDED "AS IS". TI MAKES NO WARRANTIES OR REPRESENTATIONS WITH REGARD TO THE REFERENCE DESIGNS OR USE OF THE REFERENCE DESIGNS, EXPRESS, IMPLIED OR STATUTORY, INCLUDING ACCURACY OR COMPLETENESS. TI DISCLAIMS ANY WARRANTY OF TITLE AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, QUIET ENJOYMENT, QUIET POSSESSION, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS WITH REGARD TO TI REFERENCE DESIGNS OR USE THEREOF. TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY BUYERS AGAINST ANY THIRD PARTY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON A COMBINATION OF COMPONENTS PROVIDED IN A TI REFERENCE DESIGN. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, SPECIAL, INCIDENTAL, CONSEQUENTIAL OR INDIRECT DAMAGES, HOWEVER CAUSED, ON ANY THEORY OF LIABILITY AND WHETHER OR NOT TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, ARISING IN ANY WAY OUT OF TI REFERENCE DESIGNS OR BUYER’S USE OF TI REFERENCE DESIGNS.

TI reserves the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques for TI components are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers’ products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers’ products and applications, Buyers should provide adequate design and operating safeguards.

Reproduction of significant portions of TI information in TI data books, data sheets or reference designs is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards that anticipate dangerous failures, monitor failures and their consequences, lessen the likelihood of dangerous failures and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in Buyer’s safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed an agreement specifically governing such use.

Only those TI components that TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components that have ***not*** been so designated is solely at Buyer's risk, and Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265